ReFPN-FCOS: One-stage Object Detection for Feature Learning and Accurate Localization

JIEXIAN ZENG1,3, JIALE XIONG1,2, XIANG FU1,2 and LU LENG1,2,4*(Member, IEEE)

1School of software, Nanchang Hangkong University, Nanchang 330063, China
2Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China
3Science and Technology College, Nanchang Hangkong University, Gongqingcheng 332020, China
4School of Electrical and Electronic Engineering, College of Engineering, Yonsei University,Seoul 120749, Republic of Korea

Corresponding author: Lu Leng (leng@nchu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61763033, Grant 61662049, Grant 61866028, Grant 61663031, and Grant 61866025, the Key Program Project of Research and Development (Jiangxi Provincial Department of Science and Technology) under Grant 20171ACE50024.

ABSTRACT One-stage object detectors are simple and efficient; however, they cannot extract sufficient object features due to simplistic structures. At the same time, the classification score cannot reflect the actual positioning of the candidate box. Therefore, it is not accurate to use classification score only as the candidate box position score in non-maximum suppression (NMS) stage. These two shortcomings degrade the detection accuracy. In this paper, a novel feature pyramid architecture named refined feature pyramid network (ReFPN) is introduced to obtain better object features. ReFPN designs a refined module which is parallel with feature pyramid network (FPN) to extract the semantic features of objects, and then the extraction of features are used to optimize the features of FPN by summation. In addition, we design the refined center-ness (RCenter-ness) branch that predicts the position score of each point on the feature map to improve the localization accuracy. The predicted position score is multiplied by the classification score to obtain the final position score that has a stronger correlation with localization accuracy. The average precisions of this method achieve 1.1% and 1.3 % higher than those of FCOS, when using ResNet50 and ResNet101 as backbone respectively. Code download link: https://github.com/xjl-le/mmdete

INDEX TERMS Refined center-ness branch; Refined FPN; Fusion classification and location

I. INTRODUCTION

In recent years, artificial intelligence has been hot all the time. Under the support of deep learning theory, object detection is developing rapidly. A large number of researchers have put forward many excellent detection models by analyzing the problems in object detection. These models can be classified into two categories: multi-stage object detectors [1,2,3] and one-stage object detectors [4,5,6,7]. Multi-stage object detectors use the networks to detect the objects for many times. In this process, a large number of invalid prediction boxes are eliminated, and the prediction boxes of the positive samples are optimized for many times. Multi-stage object detectors can obtain more detailed classification and positioning of object, so most of the current multi-stage object detectors are superior to the current one-stage object detectors in terms of detection accuracy. However, multi-stage object detectors are typically too large, such as Faster R-CNN [8], Cascade R-CNN [9] and Hybrid task cascade (HTC) [10]. They all require high performance device environment, and the detection time is very long, so multi-stage object detectors cannot meet the current requirements of real-time object detection. One-stage object detectors can be implemented in two ways: the anchor-based methods [11,12] and the anchor-free methods [13,14,15,16]. The current one-stage object detectors include SSD [17], YOLOv1, v2, v3 [18,19,20] and FCOS [21]. They directly use full convolutional network (FCN) to classify and locate objects. The main advantages of one-stage object detectors
are light weight and fast speed, which avoid the problem of long detection time of multi-stage object detectors.

Although the detection accuracies of one-stage object detectors are lower than those of multi-stage object detectors, one-stage object detectors have more extensive applications in real-time object detection. Therefore, it is significant to construct a high performance one-stage object detector. In order to explore the gap between one-stage object detectors and multi-stage object detectors, we tested and compared the current state-of-the-art one-stage detector FCOS and multi-stage detector Cascade R-CNN respectively. As shown in Figure 1, Cascade R-CNN detected more objects especially small objects, and its detection result for occluded objects is also better than that of FCOS. Thus one-stage object detectors have great improvement space in detecting small objects and occluded objects.

In order to improve the accuracy of one-stage object detectors, two approaches are employed. 1) To obtain more representative features of the object; 2) To obtain more accurate position score of the predicted box. Most of the object detectors use feature pyramid network (FPN) [22] structure to optimize the features of different levels. FPN distributed the objects of different sizes to different feature layers for detection. With the deepening of the network, the semantic information of the feature map is richer, while the shallow features have richer location information. In order to enable the lower features to obtain more semantic information, FPN proposed top-down structure, which transmitted the feature map from the high level to the low level to enhance the semantic information of the feature map of the shallow level. FPN alleviated the difficulty of multi-scale detection and obtained better object features.

However, the structure of FPN is not optimal, and it has two deficiencies: 1) FPN used 1x1 convolution to reduce the channel number in the feature map. Figure 2 shows the feature channel number on each layer of the backbone. The channel numbers of the feature maps C3, C4 and C5 are 512, 1024 and 2048, respectively. However, all the channel numbers uniformly become 256 after 1x1 convolution, so the features on the layers lose a lot of useful information. 2) The top-down structure of FPN paid more attention to the features of adjacent layers, so there was an obstacle to the propagation of features from high level to low level. Thus FPN-based detectors cannot get the optimal results of the small objects by using low level feature.

A practical and effective feature pyramid structure, refined feature pyramid Network (ReFPN), is designed in this paper to alleviate the deficiencies of FPN. ReFPN is mainly composed of two modules.

Module 1: First, the features of different layers of the backbone are convolved with 3x3 convolution kernels to obtain the feature maps of different sizes. Then linear interpolation scales up the high-level features to the size of the lowest-level feature. Finally, the features are added by the element-wise way to obtain refined feature. Compared with 1x1 convolution of FPN, 3x3 convolution is adopted in this paper, which effectively increases the receptive field of the network, thus this module obtains additional semantic features of different feature layers, and these features are different from those of FPN.

Module 2: To integrate multi-level features and obtain rich semantic feature, the refined feature is resized to the sizes of multi-level features with max-pooling. Then the obtained features are used to strengthen the original features of FPN. Therefore, the feature maps of the objects are transmitted among multiple feature layers, avoiding the obstacle of feature transmission. Through the above processing, ReFPN can effectively avoid the deficiencies of FPN and obtain better object feature than FPN.

Object detection is mainly composed of two tasks: classification task and localization task. The classification task is to identify the objects, it tends to obtain semantic
information from the network. The richer the semantic information is, the more accurate the object recognition is. The localization task determines the positions of the objects and uses the appropriate boxes to locate the objects. The positioning information is more inclined to the context information of the network. The richer the context information is, the more accurate the positioning of the object is. From the above analysis, it can be seen that the information for classification and localization is different, so the correlation between classification and localization is low. However, during inference stage, most of the current object detectors still use classification scores as the position scores in the non-maximum suppression (NMS) [23] stage, which degrades the accuracy of the model. To solve this problem, this paper designs the refined center-ness (RCenter-ness) branch, which takes the object context information as input to predict the position score of the prediction box.

Our main contributions are summarized as follows: 1) We review the FPN structure as well as the correlation between classification and location confidence in detail, and point out their shortcomings. Then we propose a novel one-stage detector, ReFPN-FCOS, as a practical and efficient framework that combines two novel modules: ReFPN and RCenter-ness branch. 2) ReFPN is mainly used to fuse high-level and low-level object features as well as enhance the semantic information of low-level features. 3) In RC Center-ness branch, the predicted object position scoring branch is parallel to the regression branch, and the two branches share the parameters. In order to obtain the better object position information, the regression result is taken as the input of the object position scoring. 4) A super parameter is used to control and adjust the contribution of the positioning score.

ReFPN-FCOS is evaluated on MS COCO with various backbones and obtains significant improvements over state-of-the-art one-stage detectors. The average accuracy of ReFPN-FCOS on ResNet50 is 37.7%, which is higher than 36.6% of the original FCOS. Meanwhile, the average accuracy on ResNet101 is 41.8%, which is higher than 40.5% of the original FCOS.

II. RELATED WORK
A. OPTIMAL FPN FEATURE
At present, both the one-stage object detectors and the two-stage object detectors have outstanding achievements. In order to obtain better object features, object detectors use some modules to aggregate the features of different layers to generate distinctive feature representation and enhance the detection performance of the detectors. FPN spread high-level semantic information to low-level feature map through a top-down structure, enriching the feature information of each layer. It improved the performance of the detector. Hu et al. [24] used a ResNet50 [25] with FPN as the backbone for feature extraction, which achieved a learning-based image detection method for printed circuit board defect detection.

FPN is not the optimal structure, so many object detectors improved on FPN by aggregating multi-scale features. PANet [26] pointed out that low-level features were of great help to instance segmentation, but low-level features need to go through a large number of convolutional layers to obtain high-level features, which hindered the transmission of low-level location information. In order to reduce the path of features in information transmission, PANet added a bottom-up structure on the basis of FPN. First, features were transmitted from the high level to the low level, and then the obtained features were transmitted from the low level to the high level through the bottom-up structure. Object features were extracted by fusing multi-layer convolution feature maps [27]. Libra R-CNN [28] aggregated the features of all layers of the feature pyramid, then used the aggregated features to refine the features of FPN. NAS-FPN [29] firstly used neural architecture to search the optimal FPN structure. AugFPN [30] designed residual feature augmentation to enhance high-level features. Mask Refined R-CNN [31] improved the precision of Mask R-CNN [32] for instance segmentation by combining multi-scale information. Since fusion can improve the performance of detector, the analysis on correlation of fusion features showed that fusion of low-correlation feature can avoid redundancy and achieve more obvious improvement [33]. Guo et al. [34] designed a balanced attention FPN by fusing the multi-level feature maps of the FPN to balance the original features. Li et al. [35] designed a dense path aggregation feature pyramid network to obtain rich semantic features and context features.

In addition, some advanced and latest works were developed to enhance the robustness of object features, such as rotation-invariant property. Wu et al. [36] developed ORSIm detector to extract the rotation-invariant channel features in the frequency domain and the original spatial channel features. Wu et al. [37] proposed a multi-source active fine-tuning vehicle detection (Ms-AFt) framework to obtain vehicle feature. Xu et al. [38] used rotation-invariant features and channel features to construct more robust feature representation.

All of these works have different degrees of optimization for the feature maps. Unlike them, this paper designs a novel structure ReFPN, it takes into account both information loss caused by the reduction of feature channels and information transmission hindrance from high level to low level. Thus ReFPN can obtain better object feature to improve the accuracy.

B. FUSION OF CLASSIFICATION AND LOCATION
In recent years, many scholars have pointed out that there is a big difference between classification and localization, so it is very important to design a reasonable detection head or optimization strategy for object detector. Double-head RCNN [39] found that using FC-head and Conv-head to predict classification and regression could achieve better results. IoU-Net [40] designed Intersection-over-Union (IoU)
prediction head that predicted the IoU of each candidate box. The predicted IoU guided the NMS to better remove repeated prediction boxes. IoU-Aware single-stage object detection [41] designed a branch of IoU prediction that shared a detection head with regression branch to predict IoU between the proposal box and the ground-truth box. During inference, the predicted IoU was multiplied by the classification confidence to obtain the final position score that was used in the NMS. The fusion also required reasonable optimization strategy [42,43]. FCOS proposed the center-ness that was used for the position score of the prediction box. During inference, the predicted center-ness computed the weights of the classification scores to obtain the final position scores. In this paper, the RCenter-ness branch is proposed, which is similar to center-ness of FCOS. However, RCenter-ness uses the context information and the result of regression as inputs to obtain the position score. This modification brings obvious improvement to the accuracy.

C. ANCHOR-FREE DETECTION

In recent years, anchor-free detection methods have attracted more and more attention, they avoid the super parameters selection of anchor-based methods and reduce the number of prediction boxes. FoveaBox [44] took the points in the object center area as positive samples, and designed a regression strategy that mapped the location of feature points to the corresponding prediction box. PolarMask [45] predicted 36 distances from the point on the feature map to the object contour, and it proposed Polar center-ness for inhibiting low quality positive samples. OYOLOv2_FTD [46] considered the entire image area for strong object detection and used loss function to enhance effective learning. Sparse-YOLO [47] proposed a hardware/software (HW/SW) co-design methodology targeting CPU+FPGA-based heterogeneous platforms to achieve real-time object detection.

Some Anchor-free methods are based on key point detection. Hei et al. [48] designed the corner pooling module which predicted the two key points of object: the upper left corner and the lower right corner, and then used predicted embedding vector to combine of the two points at the upper left corner and the lower right corner. Zhou et al. [49] modeled an object as a single point, the center point of the ground-truth box, and used the network to learn the length and width of the object. In order to get a better heat map of key point output, Gaussian kernel function was applied to
output the key point of the object to the heat map. These models have satisfactory detection performance. Therefore, ReFPN-FCOS is implemented in anchor-free way. It takes the ground-truth boxes of the objects as prior information to generate prediction boxes, so the number of prediction boxes is less than that of the anchor-based methods.

III. METHODOLOGY

The ReFPN-FCOS framework is shown in Figure 2. ReFPN designs a refined module based on FPN to fuses the features of multi-layer feature map, this module is in parallel with FPN structure and optimizes the original FPN feature maps. In addition, we modify center-ness branch, which is parallel with the regression branch and shares the same input. The branch is used to predict the position scores of points in the ground-truth box. ReFPN-FCOS alleviates the problem of features loss and improves localization accuracy. The details of each module are specified as follows.

A. REFPN

Inspired by AugFPN, as the network layer deepens, the channel number in the feature map is twice that in the previous layer. In order to reduce the number of network parameters, FPN used 1x1 convolution to reduce the channel numbers in all feature maps of the backbone network to 256, which hindered the network from extracting sufficient object features. As the channel number was greatly reduced, a large amount of semantic information was lost in the deep feature map. While the channel number in the shallow feature was already small, after 1x1 convolution, a large amount of information was also lost. At the same time, FPN used the top-down structure to transfer the deep features to the shallow features, so as to increase the semantic information of the lower features. However, the top-down structure of FPN paid more attention to the information of adjacent feature maps, which hindered the top-down feature transmission and decreased the accuracy of small objects.

In order to solve these two problems, we propose ReFPN that has two main functions: 1) solve the problem of feature loss caused by the decrease of channel number in deep feature maps; 2) increase the semantic information of shallow feature maps and improve the detection accuracy of small objects. As shown in Figure 2, ReFPN is an additional branch on the basis of FPN, which makes up for the deficiency of FPN and facilitates the transfer of features at all levels.

At present, most object detectors adopt FPN to extract object features, and the objects of different sizes are dispersed to different feature layers of FPN for detection. The deep feature maps are used to detect large objects, while the shallow feature maps are used to detect small objects. As shown in Figure 2, the feature map on the backbone network is C1, where I represents the layer index of the feature map. For simple representation, the lowest feature map and highest feature map are respectively denoted as \( l_{\text{min}} \) and \( l_{\text{max}} \). The backbone network only retains \{C3, C4, C5\} as effective feature maps to reduce the number of network parameters and improve the detection speed. As shown in Figure 3, in order to make up for the information loss caused by the reduction of channel number, the refined module performs 3x3 convolution on C3, C4 and C5, respectively, and obtains the corresponding feature maps, M3, M4 and M5, with the channel number of 256. Then linear interpolation increases the sizes of M4 and M5 to the same size as M3. Finally, the three feature maps are summed to get the refined feature:

\[
\text{Re Feature} = \sum_{l_{\text{min}}}^{l_{\text{max}}} M_i
\]  

As shown in Figure 2, ReFPN keeps the original structure of FPN, and the feature maps \{P3, P4, P5\} are obtained from the \{C3, C4, C5\} after FPN. Note that there are no feature maps \{P6, P7\} at this time. In order to obtain a better feature representation, the refined feature obtains a feature map with the same size of P4 by maximum pooling, and then the
obtained feature map is fused with P4 by the element-wise way to obtain R4. The R5 is also obtained in this way. Since P3 has the same size and channels as refined feature, so R3 integrates refined feature and P3 without any processing to reduce parameters. These operations address two shortcomings of FPN. R6 is obtained by convolving feature map R5 with 3x3 convolution kernels and the stride is equal to 2, while R7 is obtained by convolving feature map R6 with 3x3 convolution kernels and the stride is equal to 2, so R6 and R7 can obtain more useful information.

As shown in Figure 2, the single-layer convolution branch RCenter-ness is parallel to the regression branch and shares the positioning feature map with the regression branch. To keep the efficiency, the RCenter-ness prediction head only uses a single 3x3 convolution to obtain the position score feature map, the size of feature map is \([N, 1, W, H]\). To better predict the position score of each point within the ground-truth box, we also take regression distances as the input of RCenter-ness. As shown in Figure 5, regression distances are the distances between the points and the four sides of the ground-truth of each object, then these regression distances convolved with 1x1 to obtain the position score feature map, the size of which is \([N, 1, W, H]\). Notice that the regression distances are the predicted distance transformed by the scale function. The two position score feature maps are added to get the final position score \(RCENTERNESS\). The \(RCENTERNESS\) target of each point within the ground-truth box is computed by regression distances as:

\[
RCENTERNESS = \frac{\min(l^*, r^*)}{\max(l^*, r^*)} = \frac{\min(l^*, b^*)}{\max(l^*, b^*)}
\]  

Where \(l^*, l^*, b^*\) and \(r^*\) respectively represent the distances between the positive sample point and the four sides of the corresponding ground-truth box. We also employ positive square root here to slow down the decay of the \(RCENTERNESS\). The value range of \(RCENTERNESS\) is \([0, 1]\). Therefore, cross entropy loss (BCE) is used as the loss function during training and only the losses for the positive examples are computed as:

\[
L_{RCENTERNESS} = \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} BCE \left( RCENTERNESS, \hat{RCENTERNESS} \right)
\]  

\(RCENTERNESS\) is the position score predicted by the network. In the training stage, we take \(RCENTERNESS\) as the weight of regression loss, which effectively prompts the gradient from RCenter-ness branch to be back-propagate to the regression head. The prediction boxes closer to the ground-truth center area get higher weights, which effectively inhibits the low-quality prediction boxes that are far away from the target center. In the inference stage, ReFPN-FCOS produces a large number of prediction boxes, so we use NMS to eliminate the repeated prediction boxes. Traditional NMS uses the classification score as the position score of the prediction box, which is unreasonable. Therefore, we use the products of the classification scores and \(RCENTERNESS\) as the position scores:

\[
S_{pos} = Scores \times RCENTERNESS
\]  

The parameter \(\epsilon\) controls the contribution of \(RCENTERNESS\). \(S_{pos}\) is used in the subsequent NMS. The ablation study in the Section IV. D will determine the value of \(\epsilon\).

IV. EXPERIMENTS

A. DATASET AND EVALUATION METRICS

All our experiments are carried out on MS COCO data set which contains a large number of complex daily scenes and covers common objects in daily life. MS COCO is a universally recognized data set for current object detection,
TABLE 1. Comparison with the state-of-the-art methods.

| Method     | Backbone       | Schedule | AP  | AP 50 | AP 75 | AP S | AP M | AP L |
|------------|----------------|----------|-----|-------|-------|------|------|------|
| YOLOv2 [20]| DarkNet-19     | -        | 21.6| 44.0  | 19.2  | 5.0  | 22.4 | 35.5 |
| SSD512 [18]| ResNet-101     | -        | 31.2| 50.4  | 33.3  | 10.2 | 34.5 | 49.8 |
| RetainNet [4]| ResNet-101   | -        | 39.1| 59.1  | 42.3  | 21.8 | 42.7 | 50.2 |
| Faster R-CNN [8]| ResNet-101-FPN | -    | 36.2| 59.1  | 42.3  | 21.8 | 42.7 | 50.2 |
| Deformable R-FCN [1]| Inception-ResNet-v2 | -  | 37.5 | 58.0  | 40.8  | 19.4 | 40.1 | 52.5 |
| Mask R-CNN [32]| ResNet-101-FPN | -       | 38.2| 60.3  | 41.7  | 20.1 | 41.1 | 50.2 |
| FoveaBox [41]# | ResNet-50-FPN   | 1x       | 36.5| 56.0  | 38.6  | 20.5 | 39.9 | 47.7 |
| FoveaBox# | ResNet-101-FPN   | 1x | 38.6 | 57.9  | 41.1  | 21.6 | 42.5 | 50.4 |
| FCOS [22]# | ResNet-50-FPN   | 1x | 36.6 | 55.7  | 38.8  | 20.7 | 40.1 | 47.4 |
| FCOS# | ResNet-101-FPN   | 1x | 39.2 | 58.8  | 42.1  | 22.9 | 42.8 | 51.6 |
| FCOS# | ResNet-101-FPN-MS | 2x | 40.9 | 60.3  | 44.1  | 24.7 | 45.0 | 52.3 |
| Faster R-CNN* | ResNet-50-FPN   | 1x | 36.2 | 58.5  | 38.9  | 21.0 | 38.9 | 45.3 |
| Faster R-CNN* | ResNet-101-FPN | 1x | 38.8 | 60.9  | 42.1  | 22.6 | 42.4 | 48.5 |
| Faster R-CNN* | ResNetXt-32x8d-101-FPN | 1x | 40.3 | 62.7  | 44.0  | 24.4 | 43.7 | 49.8 |
| RetainNet* | ResNet-50-FPN   | 1x | 35.9 | 55.8  | 38.4  | 19.9 | 38.8 | 45.0 |
| RetainNet* | ResNet-101-FPN | 1x | 38.1 | 58.5  | 40.8  | 21.2 | 41.5 | 48.2 |
| FCOS* | ResNet-50-FPN   | 1x | 36.6 | 55.8  | 39.0  | 20.6 | 40.4 | 47.8 |
| FCOS* | ResNet-101-FPN | 1x | 38.6 | 57.7  | 41.0  | 21.6 | 42.3 | 50.8 |
| FCOS* | ResNet-101-FPN-MS | 2x | 40.5 | 61.3  | 43.5  | 23.4 | 44.7 | 53.2 |
| ReFPN-FCOS | ResNet-50-FPN | 1x | 37.7 | 56.0  | 40.3  | 21.6 | 41.6 | 48.2 |
| ReFPN-FCOS | ResNet-101-FPN | 1x | 39.7 | 58.2  | 42.7  | 23.1 | 43.5 | 51.8 |
| ReFPN-FCOS | ResNet-101-FPN-MS | 2x | 41.8 | 60.6  | 45.0  | 25.3 | 46.1 | 53.9 |

The symbol * represents the results that re-implement on mmdetection. The symbol # represents the results given by mmdetection. Schedule represents the number of iterations during training, the same as Detection, 1x and 2x respectively represent the model training 12 and 24 epochs on the training data set. MS shows the results of the multi-scale training.

TABLE 2. Recall rate AR.

| Method     | Backbone       | ARmax=1 | ARmax=10 | ARmax=100 | ARsmall | ARmedium | ARlarge |
|------------|----------------|---------|----------|------------|---------|----------|--------|
| FCOS* | ResNet50-FPN   | 31.4    | 50.9     | 53.9       | 33.6    | 58.5     | 69.0   |
| FCOS* | ResNet101-FPN | 32.8    | 52.6     | 55.6       | 34.9    | 60.8     | 70.7   |
| ReFPN-FCOS | ResNet50-FPN | 31.7    | 51.6     | 54.6       | 35.0    | 59.0     | 69.7   |
| ReFPN-FCOS | ResNet101-FPN | 33.5    | 53.7     | 56.7       | 36.4    | 61.7     | 73.0   |

The symbol * represents the results that reappear on MMdetection.

which has strong verifiability for current detection algorithms. It introduces three new challenges: (1) MS COCO contains a large number of objects of different sizes and scales, among which small objects account for a large proportion; (2) With complex background, MS COCO generally has more categories or instances than PASCAL and ImageNet data sets, and a large number of occlusion phenomena exist between the objects; (3) Evaluation standard is more inclined to accurate object positioning. MS COCO database consists of three data sets: training, test and validation. In MS COCO2017, there are 115K training data, 5K verification data and 20K with no disclosed labels for test data. All experiments follow the evaluation criteria of MS COCO dataset. We use AP (average AP at IoUs from 0.5 to 0.95 with an interval of 0.05), AP50 (AP at IoU= 0.5) and AP75 (AP at IoU= 0.75). We also use APS, AP50 and AP75 to measure the detector’s average accuracy of small, medium and large objects.

B. TRAINING AND TESTING DETAILS

ReFPN-FCOS is modified based on FCOS. During training, as the same as FCOS, we use Focal Loss and IoU loss as the loss functions of classification and regression respectively. For fair comparisons, all experiments are implemented on Pytorch and mmdetection [50]. There have been no artificial changes to the FCOS configuration file, and all related configurations are the same as FCOS detection. However, since we only have two GPUs, the batch size is set to 10 (5 images per GPU) for Resnet50, and the learning rate is set as 0.01 according to the linear scaling rule. The batch size is set to 6 (3 images per GPU) for Resnet101, and the learning rate is set as 0.005. For the main results, all the modules are evaluated on COCO. The converged models provided by mmdetection are evaluated as the baselines. All of our experiments are carried out in strict accordance with the standard operation, without adding any tricks. In inference...
phase, we evaluate ReFPN-FCOS using the image scale of [1333, 800] on COCO.

C. MAIN RESULTS
We use different backbone networks to verify our algorithm. The experimental results are shown in Table 1. ReFPN-FCOS is compared with the state-of-the-art object detection models, and exceeds the current state-of-the-art anchor-free algorithms. In the case of different batch sizes, ReFPN-FCOS exceeds the baseline by 0.5%~1.1% on different backbone networks, the improvement on ResNet50 is more remarkable than that on ResNet101. This result is because that the batch size of ResNet50 is larger than ResNet101. To make a fair comparison, we have re-implemented the baseline. With the same batch size, ReFPN-FCOS can improve AP by 1.1%~1.3% compared with the baselines on different backbones. The performance of APs is largely improved by 1%~1.9%, while the performance of APM is improved by 1.2%~1.4%. Thus ReFPN-FCOS has better performance on small and medium objects. In addition, the performance of ReFPN-FCOS surpasses that of anchor-free detector FoveaBox with the same backbone by 1.1% and 1.2% in terms of AP, it indicates that ReFPN-FCOS is better fit the objects.

To further illustrate the effectiveness of our approach, we make a comparative analysis of recall rates. As shown in Table 2, ReFPN-FCOS significantly improves the recall rate of the proposal box with the same backbone. The performances for $AP_{max=1}$ are improved by 0.3% and 0.7% compared with the baselines, and the performances for $AP_{max=10}$ are improved by 0.7% and 1.1%. In addition, the $AP_{max=100}$ of ReFPN-FCOS are 0.7% and 1.1% higher than those of the baselines. It shows that ReFPN-FCOS can enhance the average recall of proposal. We find that the improvements of $AP_{max}$ on ResNet101 are more remarkable than that on ResNet50, because larger backbone can achieve better fitting objectives.

D. ABLATION EXPERIMENTS
ReFPN-FCOS designs two modules on the basis of FCOS. In order to analyze the importance of the two modules, experiments are conducted on each module. The experimental results are shown in Table 3. ReFPN and RCenterness have improved the original FCOS to different degrees. Compared with the original FCOS, ReFPN has increased AP by 0.4%, which is not a big improvement. However, ReFPN yields 1.1% higher $AP_S$ compared with the original FCOS. This result shows that ReFPN extracts more representative feature of small objects by aggregating high

---

| ReFPN          | RCenterness | AP  | AP50 | AP75 | APS | APM | APL |
|----------------|-------------|-----|------|------|-----|-----|-----|
| -              | -           | 36.6| 55.8 | 39.0 | 20.6| 40.4| 47.8|
| ✓              | -           | 37.0| 55.8 | 39.5 | 21.7| 40.7| 48.2|
| -              | ✓           | 37.1| 55.9 | 39.6 | 21.0| 40.9| 48.3|
| ✓              | ✓           | 37.7| 56.0 | 40.3 | 21.6| 41.6| 48.2|

| Setting        | AP  | AP50 | AP75 | APS | AM  | APL |
|----------------|-----|------|------|-----|-----|-----|
| RCenterness_I  | 37.4| 56.6 | 39.9 | 21.8| 41.3| 47.9|
| RCenterness_II | 37.7| 56.0 | 40.3 | 21.6| 41.6| 48.2|

| $\epsilon$ | AP  | AP50 | AP75 | APS | AM  | APL |
|------------|-----|------|------|-----|-----|-----|
| 0.0        | 38.9| 58.5 | 41.4 | 23.9| 44.1| 49.2|
| 0.1        | 40.2| 59.3 | 42.9 | 24.3| 45.0| 51.3|
| 0.2        | 40.8| 55.9 | 43.7 | 24.7| 45.6| 52.3|
| 0.3        | 41.2| 60.3 | 44.2 | 24.8| 45.9| 52.8|
| 0.4        | 41.5| 60.5 | 44.6 | 24.9| 46.1| 53.4|
| 0.5        | 41.7| 60.7 | 44.8 | 25.1| 46.1| 53.7|
| 0.6        | 41.8| 60.7 | 45.0 | 25.1| 46.1| 53.9|
| 0.7        | 41.8| 60.7 | 45.0 | 25.2| 46.1| 53.9|
| 0.8        | 41.8| 60.6 | 45.0 | 25.3| 46.1| 53.9|
| 0.9        | 41.8| 60.5 | 45.0 | 25.3| 45.9| 54.0|
| 1.0        | 41.8| 60.4 | 45.0 | 25.3| 45.9| 53.9|
| 1.1        | 41.8| 60.2 | 45.0 | 25.3| 45.8| 53.9|
| 1.2        | 41.7| 60.1 | 44.9 | 25.3| 45.7| 53.8|
| 1.5        | 41.5| 59.6 | 44.7 | 25.2| 45.4| 53.5|
and low level features. The RCenter-ness branch brings 0.5% higher AP than the FCOS baseline, and it has different degrees of improvement to the large, medium and small objects. Thus the RCenter-ness branch can obtain better position scores, indicating that the context feature of object is more suitable for predicting the position information of the object box. From the above data, the two modules designed in this paper are effective. After combining the two modules, the average accuracy of ReFPN-FCOS is 1.1% higher than that of the FCOS baseline. It indicates that the two modules of ReFPN-FCOS are complementary and jointly improve the performance of detector.

Ablation experiments of RCenter-ness are shown in Figure 4. We design two implementation schemes for the RCenter-ness branch. In Figure 4(a), RCenter-ness_I uses context feature as input, while RCenter-ness_II uses context feature and the regression result as input in Figure 4(b). The main difference between the two design options is whether to add regression result. The experimental results of the two types are shown in Table 4, the RCenter-ness_II obtains a better performance, which improves AP by 0.3% compared with RCenter-ness_I. The result indicates that adding regression result could better promote the detector to predict position score. In addition, for medium and large objects, the performance of RCenter-ness_II is better than that of RCenter-ness_I. However, regression distances of small objects are too small, the advantage of RCenter-ness_II is not remarkable for small objects.
Ablation experiments on the relation between RCenterness and classification score are shown in Table 5. Under the influence of YOLO V1, most of the current detection algorithms with IoU branch or center-ness branch generally use the product of the classification score and Centerness or IoU as the final position score of the prediction box in the inference stage, and then use it to remove redundant boxes in NMS. This strategy is also adopted in this work for fair comparison. In order to verify the influence of RCenterness, the super parametric $\varepsilon$ is set to different values. The values of $\varepsilon$ gradually increase from zero, the average accuracy gets better. This is because when $\varepsilon$ is small, the value of RCenterness becomes large, resulting in a large number of low-quality predicted bounding boxes that are retained. When parameter $\varepsilon$ is between [0.6, 1], the average accuracy does not change. The result indicates that RCenter-ness can accurately predict the object position score.

Visual detection results are shown in Figure 6. To further evaluate the performance of ReFPN-FCOS and its generalization ability, we visualize the images detected by ReFPN-FCOS and FCOS. The images include multiple scenes and are randomly selected from the Internet. The background of these images are complex, and there are a lot of occlusion phenomena. Moreover, the small objects take up a large proportion. We also use a photograph of elk to verify the false detection. As shown in Figure 6, the first and second columns show the detection results of ReFPN-FCOS and FCOS, respectively. As shown in the first and third rows, FCOS obviously misses the detection of many objects, while ReFPN-FCOS can detect more objects and is good at the detection of the objects with severe occlusion. However, in both ReFPN-FCOS and FCOS, the confidences of small objects are very low due to inadequate features of small objects, so using feature alignment can alleviate this shortage. In the second row, FCOS misses the detection of persons in the bottom right corner of the image, but identifies the sheep. ReFPN-FCOS identifies the person in the bottom right corner, but mistakenly detects the sheep in the bottom right corner as cow. The images in fourth row are particularly interesting. We use the network to identify a category that does not exist in the COCO data set. ReFPN-FCOS has fewer error detection than FCOS, and its positioning is more accurate than FCOS. However, in both ReFPN-FCOS and FCOS, the confidences of small objects are very low due to inadequate features of small objects, so using feature alignment can alleviate this shortage.

V. CONCLUSIONS

In this paper, we review one-stage and two-stage detection algorithms and find that the potential of the current one-stage detection is not fully exerted, which is mainly caused by two reasons. The first reason is insufficient feature extraction. The second reason is that using the classification score as the positioning score of the prediction box is unreasonable in inference stage. Thus a more efficient detection algorithm ReFPN-FCOS is proposed. ReFPN-FCOS employs two novel modules, namely ReFPN and RCenter-ness, to alleviate the two aforementioned problems, so it achieves substantial improvement. The object features extracted by ReFPN can be further optimized, and there is still a gap between the predicted score and the real position score. In the future, we will use feature alignment to get better object features and explore a better way to predict object position scores.

REFERENCES

[1] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in: Advances in neural information processing systems. Dec. 2016, pp. 379–387.
[2] R. Girshick, “Fast R-cnn,” in: Proceedings of the IEEE international conference on computer vision, Dec. 2015, pp. 1440–1448.
[3] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchiesfor accurate object detection and semantic segmentation,” in: Proceedings of the IEEE conference on computer vision and pattern recognition, Jun. 2014, pp. 580–587.
[4] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in: Proceedings of the IEEE international conference on computer vision, Oct. 2017, pp. 2980–2988.
[5] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2018, pp. 4203–4212.
[6] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille, “Single-shot object detection with enriched semantics,” in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2018, pp. 5813–5821.
[7] B. Li, Y. Liu, and X. Wang, “Gradient harmonized single-stage detector,” in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 8577–8584, Sep. 2019.
[8] 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.
[9] 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.
[10] K. Chen, J. M. Pang, J. Q. Wang, Y. Xiong, X. Li, S. Sun, and C. C. Loy, “Hybrid task cascade for instance segmentation,” in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2019, pp. 4974–4983.
[11] K. He, X. Zhang, S Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, Jan. 2015.
[12] S. Wu, and X. Li, “IoU-balanced Loss Functions for Single-stage Object Detection,” 2019, arXiv: 1908.05641. [Online]. Available: https://arxiv.org/abs/1908.05641.
[13] 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: https://arxiv.org/abs/1509.04874.
[14] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “UnitBox: An Advanced Object Detection Network,” in: Proceedings of the 24th ACM international conference on Multimedia, Oct. 2016, pp. 516–520.
[15] X. Zhou, J. Zhuo, and P. Krahenbuhl, “Bottom-Up Object Detection by Grouping Extreme and Center Points,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2019, pp. 850-859.
[16] C. Zhu, Y. He, and M. Savvides, “Feature Selective Anchor-Free Module for Single-Shot Object Detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun.
2019, pp. 840-849.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. F. Yu, and A. C. Berg, “Ssd: Single shot multibox detector,” in: European conference on computer vision, Springer, Sep. 2016, pp. 21–37.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in: Proceedings of the IEEE conference on computer vision and pattern recognition, Jun. 2016, pp. 779–788.

[19] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” In Proceedings of the IEEE conference on computer vision and pattern recognition, Jul. 2017, pp. 7263–7271.

[20] J. Redmon, and A. Farhadhi, “YOLOv3: An Incremental Improvement,” 2018, arxiv: 1804.02767. [Online]. Available: https://arxiv.org/abs/1804.02767.

[21] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” In Proceedings of the IEEE international conference on computer vision. Oct. 2019, pp. 9627-9636.

[22] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in: Proceedings of the IEEE conference on computer vision and pattern recognition, Jul. 2017, pp. 2117–2125.

[23] J. Chu, Y. Q. Zhang, L. Leng, and Jun Miao, “Syncretic-NMS: A merging non-maximum suppression algorithm for instance segmentation,” IEEE Access, vol. 8, pp. 114705-114714, Jun. 2020.

[24] B. Hu, and J. Wang, “Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network,” IEEE Access, vol. 8, pp. 108335-108345, Jun. 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” In Proceedings of the IEEE conference on computer vision and pattern recognition. Jun. 2016, pp. 770–778.

[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” In Proceedings of the IEEE conference on computer vision and pattern recognition, Jun. 2018, pp. 8759-8768.

[27] J. Chu, Z. X. Guo, and L. Leng, “Object detection based on multi-layer convolution feature fusion and online hard example mining,” IEEE Access, vol. 6, pp. 19959-19967, Mar. 2018.

[28] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra R-CNN: Towards Balanced Learning for Object Detection”. In Proceedings of the IEEE conference on computer vision and pattern recognition, vol. 57, no. 7, pp. 821-830, Jun. 2019.

[29] G. Ghiasi, Y. T. Lin, and V. Q. Le, “NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection,” In Proceedings of the IEEE conference on computer vision and pattern recognition, Jun. 2019, pp. 7036-7045.

[30] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, “ Augfpn: Improving multi-scale feature learning for object detection,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Jun. 2020, pp. 12595-12604.

[31] Y. Q. Zhang, J. Chu, L. Leng, and J. Miao, “Mask R-CNN: A network for refining object details in instance segmentation,” Sensors, vol. 22, no. 4, pp. 1010.1-1010, Feb. 2020.

[32] K. He, G. Gkioxari, P. Doll, and R. Girshick, “Mask r-cnn,” in: Proceedings of the IEEE international conference on computer vision, Jun. 2017, pp. 2961–2969.

[33] L. Leng, and J. S. Zhang, “PalmHash Code vs. PalmPhasor Code,” Neurocomputing, vol.108, no. 2, pp. 108:1-12, Aug. 2013.

[34] Q. Guo, L. Liu, W. Xu, Y. Gong, X. Zhang, and W. Jing, “An Improved Faster R-CNN for High-Speed Railway Dropper Detection,” IEEE Access, vol. 8, pp. 105622-105633, Jun. 2020.

[35] Y. Li, X. Pei, and Q. Huang, “Anchor-Free Single Stage Detector in Remote Sensing Images Based on Multiscale Dense Path Aggregation Feature Pyramid Network,” IEEE Access, vol. 8, pp. 63121-63133, Mar. 2020.

[36] X. Wu, D. F. Hong, J. J. Tian, J. Chanussot, and W. Li, “ORSIm Detector: A Novel Object Detection Framework in Optical Remote Sensing Imagery Using Spatial-Frequency Channel Features,” IEEE Transactions on Geoscience and Remote Sensing, July, 2019, pp. 5146-5158.

[37] X. Wu, W. Li, D. F. Hong, J. J. Tian, R. Tao, and Q Du, “Vehicle detection of multi-source remote sensing data using active fine tuning network,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 167, pp. 39-53, Sep. 2020.

[38] X. Wu, D. F. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, “Fourier-Based Rotation-Invariant Feature Boosting: An Efficient Framework for Geospatial Object Detection,” IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 2, pp. 302-306, Jun. 2020.

[39] Y. Wu, Y. Chen, and L. Yuan, “Double-head rcmn: Rethinking classification and localization for object detection,” 2019, arXiv: 1904.06493. [Online]. Available: https://arxiv.org/abs/1904.06493v2.

[40] B. R. Jiang, R. X. Luo, J. Y. Mao, T. T. Xiao, and Y. N. Jiang, “Acquisition of localization confidence for accurate object detection,” In Proceedings of the European Conference on Computer Vision (ECCV), Sep. 2018, pp. 784–799.

[41] S. Wu, X. Li, and X. Wang, “IoI-aware single-stage object detector for accurate localization,” Image and Vision Computing, Mar. 2020, pp. 103911.

[42] L. Leng, and A. B. J. Teoh, “Alignment-free row-co-occurrence cancelable palmprint Fuzzy Vault,” Pattern Recognition, vol. 48, no. 7, pp. 2290-2303, Jul. 2015.

[43] L. Leng, M. Li, C. Kim, and X. Bi, “Dual-source discrimination power analysis for multi-instance contactless palmprint recognition,” Multimedia Tools and Applications, vol. 1, no. 76, pp. 333-354, Nov. 2017.

[44] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, “FoveaBox: Beyond Anchor-Based Object Detection,” IEEE Transactions on Image Processing, Jun. 2020, pp. 7389-7398.

[45] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, and P. Luo, “Polarmask: Single shot instance segmentation with polar representation,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Jun. 2020, pp. 12193-12202.

[46] H. R. Alsanad, N. O. Ucan, and M. Ilyas M, “Real-Time Fuel Truck Detection Algorithm Based on Deep Convolutional Neural Network,” IEEE Access, vol. 8, pp. 118808-118817, Jun. 2020.

[47] Z. Wang, K. Xu, S. Wu, L. Liu, L. Liu, and D. Wang, “Sparse-YOLO: Hardware/Software Co-Design of an FPGA Accelerator for YOLOv2,” IEEE Access, vol. 8, pp. 116569-116585, Jun. 2020.

[48] H. Law, and J. Deng, “CornerNet: Detecting Objects as Paired Keypoints,” european conference on computer vision, Sep. 2018, pp. 765-781.

[49] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as Points,” 2019. arXiv:1904.07850. [Online]. Available: https://arxiv.org/abs/1904.07850.

[50] J. Pang, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, and Z. Zhang, “Mmdetection: Open mmlab detection toolbox and benchmark,” 2019, arXiv:1906.07155. [Online]. Available: https://arxiv.org/abs/1906.07155.
SECOND B. AUTHOR, JIALE XIONG
received the bachelor’s degree from Nanchang Hangkong University, Jiangxi province, China. He is currently pursuing the master’s degree with Nanchang Hangkong University. His research interests include computer vision, image processing and pattern recognition.

THIRD C. AUTHOR, XIANG FU
received his Ph.D degree from Xidian University in 2008. Now he is an associate professor at Nanchang Hangkong University, his interested research fields are computer vision, image processing and pattern recognition.

FORTH D. AUTHOR, LU LENG
received his Ph.D degree from Southwest Jiaotong University, Chengdu, P. R. China, in 2012. He did his post-doctoral research at Yonsei University, Seoul, Republic of Korea, and Nanjing University of Aeronautics and Astronautics, Nanjing, P. R. China. He was a visiting scholar of West Virginia University, USA from 2015 to 2016, and Yonsei University from 2019 to 2020. Currently he is an associate professor of Nanchang Hangkong University.

He has published more than 90 international journal and conference papers, and been granted several scholarships and funding projects in his academic research. He is the reviewer of several international journals and conferences. His research interests include computer vision, biometric template protection, and biometric recognition.

Dr. Leng is a member of Institute of Electrical and Electronics Engineers (IEEE), Association for Computing Machinery (ACM), China Society of Image and Graphics (CSIG), China Computer Federation (CCF), and China Association of Artificial Intelligence (CAAI).