FCOS: Fully Convolutional One-Stage Object Detection

Zhi Tian Chunhua Shen* Hao Chen Tong He
The University of Adelaide, Australia

Abstract

We propose a fully convolutional one-stage object detector (FCOS) to solve object detection in a per-pixel prediction fashion, analogue to semantic segmentation. Almost all state-of-the-art object detectors such as RetinaNet, SSD, YOLOv3, and Faster R-CNN rely on pre-defined anchor boxes. In contrast, our proposed detector FCOS is anchor box free, as well as proposal free. By eliminating the predefined set of anchor boxes, FCOS completely avoids the complicated computation related to anchor boxes such as calculating overlapping during training. More importantly, we also avoid all hyper-parameters related to anchor boxes, which are often very sensitive to the final detection performance. With the only post-processing non-maximum suppression (NMS), FCOS with ResNeXt-64x4d-101 achieves 44.7% *in AP with single-model and single-scale testing, surpassing previous one-stage detectors with the advantage of being much simpler. For the first time, we demonstrate a much simpler and flexible detection framework achieving improved detection accuracy. We hope that the proposed FCOS framework can serve as a simple and strong alternative for many other instance-level tasks. Code is available at:* `tinyurl.com/FCOSv1`

1. Introduction

Object detection is a fundamental yet challenging task in computer vision, which requires the algorithm to predict a bounding box with a category label for each instance of interest in an image. All current mainstream detectors such as Faster R-CNN [24], SSD [18] and YOLOv2, v3 [23] rely on a set of pre-defined anchor boxes and *it has long been believed that the use of anchor boxes is the key to detectors' success.* Despite their great success, it is important to note that anchor-based detectors suffer some drawbacks: 1) As shown in [15, 24], detection performance is sensitive to the sizes, aspect ratios and number of anchor boxes. For example, in RetinaNet [15], varying these hyper-parameters affects the performance up to 4% in AP on the COCO benchmark [16]. As a result, these hyper-parameters need to be carefully tuned in anchor-based detectors. 2) Even with careful design, because the scales and aspect ratios of anchor boxes are kept fixed, detectors encounter difficulties to deal with object candidates with large shape variations, particularly for small objects. The pre-defined anchor boxes also hamper the generalization ability of detectors, as they need to be re-designed on new detection tasks with different object sizes or aspect ratios. 3) In order to achieve a high recall rate, an anchor-based detector is required to densely place anchor boxes on the input image (*e.g.*, more than 180K anchor boxes in feature pyramid networks (FPN) [14] for an image with its shorter side being 800). Most of these anchor boxes are labelled as negative samples during training. The excessive number of negative samples aggravates the imbalance between positive and negative samples in training. 4) Anchor boxes also involve complicated computation such as calculating the intersection-over-union (IoU) scores with ground-truth bounding boxes.

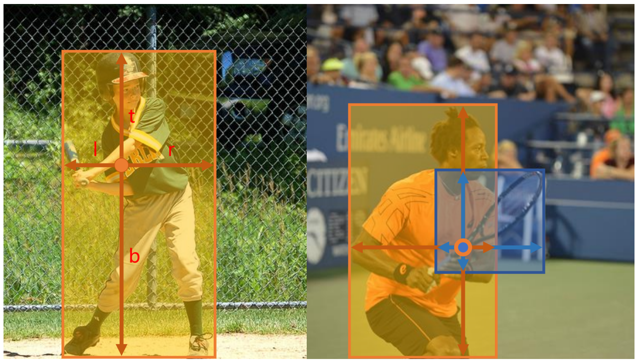

Figure 1 – As shown in the left image, FCOS works by predicting a 4D vector (l, t, r, b) encoding the location of a bounding box at each foreground pixel (supervised by ground-truth bounding box information during training). The right plot shows that when a location residing in multiple bounding boxes, it can be ambiguous in terms of which bounding box this location should regress.

Recently, fully convolutional networks (FCNs) [20] have achieved tremendous success in dense prediction tasks such as semantic segmentation [20, 28, 9, 19], depth estimation [17, 31], keypoint detection [3] and counting [2]. As one of high-level vision tasks, object detection might be the only one deviating from the neat fully convolutional per-

[*] Corresponding author, email: `chunhua.shen@adelaide.edu.au`

pixel prediction framework mainly due to the use of anchor boxes. It is nature to ask a question: *Can we solve object detection in the neat per-pixel prediction fashion, analogue to FCN for semantic segmentation, for example?* Thus those fundamental vision tasks can be unified in (almost) one single framework. We show that the answer is affirmative. Moreover, we demonstrate that, for the first time, the much simpler FCN-based detector achieves even better performance than its anchor-based counterparts.

In the literature, some works attempted to leverage the FCNs-based framework for object detection such as DenseBox [12]. Specifically, these FCN-based frameworks directly predict a 4D vector plus a class category at each spatial location on a level of feature maps. As shown in Fig. 1 (left), the 4D vector depicts the relative offsets from the four sides of a bounding box to the location. These frameworks are similar to the FCNs for semantic segmentation, except that each location is required to regress a 4D continuous vector. However, to handle the bounding boxes with different sizes, DenseBox [12] crops and resizes training images to a fixed scale. Thus DenseBox has to perform detection on image pyramids, which is against FCN's philosophy of computing all convolutions once. Besides, more significantly, these methods are mainly used in special domain objection detection such as scene text detection [33, 10] or face detection [32, 12], since it is believed that these methods do not work well when applied to generic object detection with highly overlapped bounding boxes. As shown in Fig. 1 (right), the highly overlapped bounding boxes result in an intractable ambiguity: it is not clear w.r.t. which bounding box to regress for the pixels in the overlapped regions.

In the sequel, we take a closer look at the issue and show that with FPN this ambiguity can be largely eliminated. As a result, our method can already obtain comparable detection accuracy with those traditional anchor based detectors. Furthermore, we observe that our method may produce a number of low-quality predicted bounding boxes at the locations that are far from the center of an target object. In order to suppress these low-quality detections, we introduce a novel "center-ness" branch (only one layer) to predict the deviation of a pixel to the center of its corresponding bounding box, as defined in Eq. (3). This score is then used to down-weight low-quality detected bounding boxes and merge the detection results in NMS. The simple yet effective center-ness branch allows the FCN-based detector to outperform anchor-based counterparts under exactly the same training and testing settings.

This new detection framework enjoys the following advantages.

- Detection is now unified with many other FCN-solvable tasks such as semantic segmentation, making it easier to re-use ideas from those tasks.
- Detection becomes proposal free and anchor free, which significantly reduces the number of design parameters. The design parameters typically need heuristic tuning and many tricks are involved in order to achieve good performance. Therefore, our new detection framework makes the detector, particularly its training, *considerably* simpler.
- By eliminating the anchor boxes, our new detector completely avoids the complicated computation related to anchor boxes such as the IOU computation and matching between the anchor boxes and ground-truth boxes during training, resulting in faster training and testing as well as less training memory footprint than its anchor-based counterpart.
- Without bells and whistles, we achieve state-of-the-art results among one-stage detectors. We also show that the proposed FCOS can be used as a Region Proposal Networks (RPNs) in two-stage detectors and can achieve significantly better performance than its anchor-based RPN counterparts. Given the even better performance of the much simpler anchor-free detector, *we encourage the community to rethink the necessity of anchor boxes in object detection*, which are currently considered as the *de facto* standard for detection.
- The proposed detector can be immediately extended to solve other vision tasks with minimal modification, including instance segmentation and key-point detection. We believe that this new method can be the new baseline for many instance-wise prediction problems.

2. Related Work

Anchor-based Detectors. Anchor-based detectors inherit the ideas from traditional sliding-window and proposal based detectors such as Fast R-CNN [6]. In anchor-based detectors, the anchor boxes can be viewed as pre-defined sliding windows or proposals, which are classified as positive or negative patches, with an extra offsets regression to refine the prediction of bounding box locations. Therefore, the anchor boxes in these detectors may be viewed as *training samples*. Unlike previous detectors like Fast RCNN, which compute image features for each sliding window/proposal repeatedly, anchor boxes make use of the feature maps of CNNs and avoid repeated feature computation, speeding up detection process dramatically. The design of anchor boxes are popularized by Faster R-CNN in its RPNs [24], SSD [18] and YOLOv2 [22], and has become the convention in a modern detector.

However, as described above, anchor boxes result in excessively many hyper-parameters, which typically need to be carefully tuned in order to achieve good performance. Besides the above hyper-parameters describing anchor shapes, the anchor-based detectors also need other

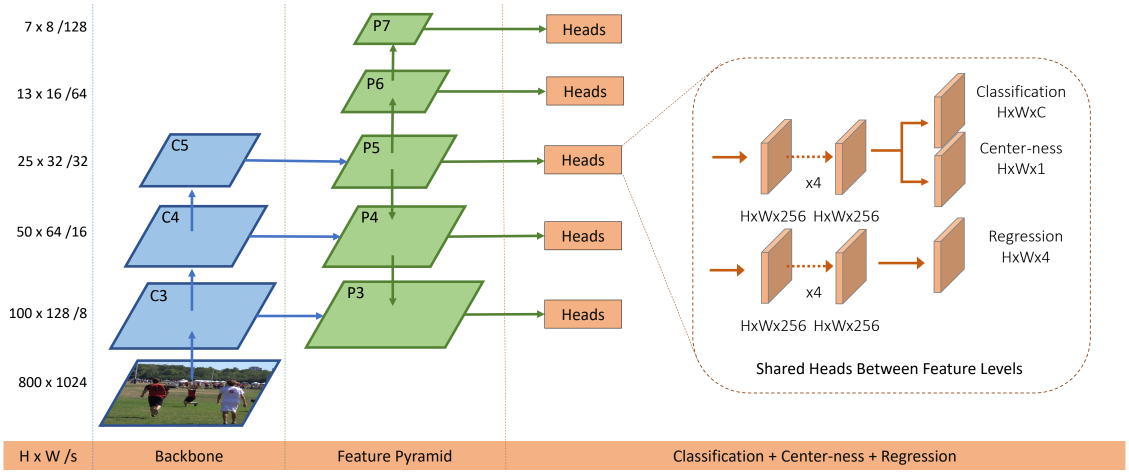

Figure 2 – The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. '/*s*' (s = 8, 16, ..., 128) is the down-sampling ratio of the feature maps at the level to the input image. As an example, all the numbers are computed with an 800 × 1024 input.

hyper-parameters to label each anchor box as a positive, ignored or negative sample. In previous works, they often employ intersection over union (IOU) between anchor boxes and ground-truth boxes to determine the label of an anchor box (*e.g.*, a positive anchor if its IOU is in $[0.5, 1]$). These hyper-parameters have shown a great impact on the final accuracy, and require heuristic tuning. Meanwhile, these hyper-parameters are specific to detection tasks, making detection tasks deviate from a neat fully convolutional network architectures used in other dense prediction tasks such as semantic segmentation.

Anchor-free Detectors. The most popular anchor-free detector might be YOLOv1 [21]. Instead of using anchor boxes, YOLOv1 predicts bounding boxes at points near the center of objects. Only the points near the center are used since they are considered to be able to produce higher-quality detection. However, since only points near the center are used to predict bounding boxes, YOLOv1 suffers from low recall as mentioned in YOLOv2 [22]. As a result, YOLOv2 [22] employs anchor boxes as well. Compared to YOLOv1, FCOS takes advantages of all points in a ground truth bounding box to predict the bounding boxes and the low-quality detected bounding boxes are suppressed by the proposed "center-ness" branch. As a result, FCOS is able to provide comparable recall with anchor-based detectors as shown in our experiments.

CornerNet [13] is a recently proposed one-stage anchor-free detector, which detects a pair of corners of a bounding box and groups them to form the final detected bounding box. CornerNet requires much more complicated post-processing to group the pairs of corners belonging to the same instance. An extra distance metric is learned for the purpose of grouping.

Another family of anchor-free detectors such as [32] are based on DenseBox [12]. The family of detectors have been considered unsuitable for generic object detection due to difficulty in handling overlapping bounding boxes and the recall being relatively low. In this work, we show that both problems can be largely alleviated with multi-level FPN prediction. Moreover, we also show together with our proposed center-ness branch, the much simpler detector can achieve even better detection performance than its anchor-based counterparts.

3. Our Approach

In this section, we first reformulate object detection in a per-pixel prediction fashion. Next, we show that how we make use of multi-level prediction to improve the recall and resolve the ambiguity resulted from overlapped bounding boxes. Finally, we present our proposed "center-ness" branch, which helps suppress the low-quality detected bounding boxes and improves the overall performance by a large margin.

3.1. Fully Convolutional One-Stage Object Detector

Let $F_i \in \mathbb{R}^{H \times W \times C}$ be the feature maps at layer i of a backbone CNN and s be the total stride until the layer. The ground-truth bounding boxes for an input image are defined as $\{B_i\}$, where $B_i = (x_0^{(i)}, y_0^{(i)}, x_1^{(i)} y_1^{(i)}, c^{(i)}) \in \mathbb{R}^4 \times \{1, 2 \ldots C\}$. Here $(x_0^{(i)}, y_0^{(i)})$ and $(x_1^{(i)} y_1^{(i)})$ denote the coordinates of the left-top and right-bottom corners of the bounding box. $c^{(i)}$ is the class that the object in the bounding box belongs to. C is the number of classes, which is 80 for MS-COCO dataset.

For each location (x, y) on the feature map F_i, we can map it back onto the input image as $(\lfloor \frac{s}{2} \rfloor + xs, \lfloor \frac{s}{2} \rfloor + ys)$, which is near the center of the receptive field of the location (x, y). Different from anchor-based detectors, which consider the location on the input image as the center of (multiple) anchor boxes and regress the target bounding box with these anchor boxes as references, we directly regress the target bounding box at the location. In other words, our detector directly views locations as *training samples* instead of anchor boxes in anchor-based detectors, which is the same as FCNs for semantic segmentation [20].

Specifically, location (x, y) is considered as a positive sample if it falls into any ground-truth box and the class label c^* of the location is the class label of the ground-truth box. Otherwise it is a negative sample and $c^* = 0$ (background class). Besides the label for classification, we also have a 4D real vector $\boldsymbol{t}^* = (l^*, t^*, r^*, b^*)$ being the regression targets for the location. Here l^*, t^*, r^* and b^* are the distances from the location to the four sides of the bounding box, as shown in Fig. 1 (left). If a location falls into multiple bounding boxes, it is considered as an *ambiguous sample*. We simply choose the bounding box with minimal area as its regression target. In the next section, we will show that with multi-level prediction, the number of ambiguous samples can be reduced significantly and thus they hardly affect the detection performance. Formally, if location (x, y) is associated to a bounding box B_i, the training regression targets for the location can be formulated as,

$$\begin{aligned} l^* &= x - x_0^{(i)}, \quad t^* = y - y_0^{(i)}, \\ r^* &= x_1^{(i)} - x, \quad b^* = y_1^{(i)} - y. \end{aligned} \tag{1}$$

It is worth noting that FCOS can leverage as many foreground samples as possible to train the regressor. It is different from anchor-based detectors, which only consider the anchor boxes with a highly enough IOU with ground-truth boxes as positive samples. We argue that it may be one of the reasons that FCOS outperforms its anchor-based counterparts.

Network Outputs. Corresponding to the training targets, the final layer of our networks predicts an 80D vector $\boldsymbol{p}$ of classification labels and a 4D vector $\boldsymbol{t} = (l, t, r, b)$ bounding box coordinates. Following [15], instead of training a multi-class classifier, we train C binary classifiers. Similar to [15], we add four convolutional layers after the feature maps of the backbone networks respectively for classification and regression branches. Moreover, since the regression targets are always positive, we employ $\exp(x)$ to map any real number to $(0, \infty)$ on the top of the regression branch. *It is worth noting that FCOS has* $9\times$ *fewer network output variables than the popular anchor-based detectors [15, 24] with 9 anchor boxes per location.*

Loss Function. We define our training loss function as follows:

$$\begin{aligned} L(\{\boldsymbol{p}_{x,y}\}, \{\boldsymbol{t}_{x,y}\}) &= \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(\boldsymbol{p}_{x,y}, c^*_{x,y}) \\ &+ \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{\{c^*_{x,y} > 0\}} L_{\text{reg}}(\boldsymbol{t}_{x,y}, \boldsymbol{t}^*_{x,y}), \end{aligned} \tag{2}$$

where L_{cls} is focal loss as in [15] and L_{reg} is the IOU loss as in UnitBox [32]. N_{pos} denotes the number of positive samples and λ being 1 in this paper is the balance weight for L_{reg}. The summation is calculated over all locations on the feature maps F_i. $\mathbb{1}_{\{c^*_i > 0\}}$ is the indicator function, being 1 if $c^*_i > 0$ and 0 otherwise.

Inference. The inference of FCOS is straightforward. Given an input images, we forward it through the network and obtain the classification scores $\boldsymbol{p}_{x,y}$ and the regression prediction $\boldsymbol{t}_{x,y}$ for each location on the feature maps F_i. Following [15], we choose the location with $p_{x,y} > 0.05$ as positive samples and invert Eq. (1) to obtain the predicted bounding boxes.

3.2. Multi-level Prediction with FPN for FCOS

Here we show that how two possible issues of the proposed FCOS can be resolved with multi-level prediction with FPN [14]. 1) The large stride (*e.g.*, 16×) of the final feature maps in a CNN can result in a relatively low *best possible recall (BPR)*[1]. For anchor based detectors, low recall rates due to the large stride can be compensated to some extent by lowering the required IOU scores for positive anchor boxes. For FCOS, at the first glance one may think that the BPR can be much lower than anchor-based detectors because it is impossible to recall an object which no location on the final feature maps encodes due to a large stride. Here, we empirically show that even with a large stride, FCN-based FCOS is still able to produce a good BPR, and it can even better than the BPR of the anchor-based detector RetinaNet [15] in the official implementation Detectron [7] (refer to Table 1). Therefore, the BPR is actually not a problem of FCOS. Moreover, with multi-level FPN prediction [14], the BPR can be improved further to match the best BPR the anchor-based RetinaNet can achieve. 2) Overlaps in ground-truth boxes can cause intractable ambiguity , *i.e.*, which bounding box should a location in the overlap regress? This ambiguity results in degraded performance of FCN-based detectors. In this work, we show that the ambiguity can be greatly resolved with multi-level prediction, and the FCN-based detector can obtain *on par*, sometimes even better, performance compared with anchor-based ones.

Following FPN [14], we detect different sizes of objects on different levels of feature maps. Specifically, we make use of five levels of feature maps defined as

[1] Upper bound of the recall rate that a detector can achieve.

$\{P_3, P_4, P_5, P_6, P_7\}$. P_3, P_4 and P_5 are produced by the backbone CNNs' feature maps C_3, C_4 and C_5 followed by a 1×1 convolutional layer with the top-down connections in [14], as shown in Fig. 2. P_6 and P_7 are produced by applying one convolutional layer with the stride being 2 on P_5 and P_6, respectively. As a result, the feature levels P_3, P_4, P_5, P_6 and P_7 have strides 8, 16, 32, 64 and 128, respectively.

Unlike anchor-based detectors, which assign anchor boxes with different sizes to different feature levels, we directly limit the range of bounding box regression for each level. More specifically, we firstly compute the regression targets l^*, t^*, $r*$ and b^* for each location on all feature levels. Next, if a location satisfies $\max(l^*, t^*, r^*, b^*) > m_i$ or $\max(l^*, t^*, r^*, b^*) < m_{i-1}$, it is set as a negative sample and is thus not required to regress a bounding box anymore. Here m_i is the maximum distance that feature level i needs to regress. In this work, m_2, m_3, m_4, m_5, m_6 and m_7 are set as 0, 64, 128, 256, 512 and ∞, respectively. Since objects with different sizes are assigned to different feature levels and most overlapping happens between objects with considerably different sizes. If a location, even with multi-level prediction used, is still assigned to more than one ground-truth boxes, we simply choose the ground-truth box with minimal area as its target. As shown in our experiments, the multi-level prediction can largely alleviate the aforementioned ambiguity and improve the FCN-based detector to the same level of anchor-based ones.

Finally, following [14, 15], we share the heads between different feature levels, not only making the detector parameter-efficient but also improving the detection performance. However, we observe that different feature levels are required to regress different size range (*e.g.*, the size range is $[0, 64]$ for P_3 and $[64, 128]$ for P_4), and therefore it is not reasonable to make use of identical heads for different feature levels. As a result, instead of using the standard $\exp(x)$, we make use of $\exp(s_i x)$ with a trainable scalar s_i to automatically adjust the base of the exponential function for feature level P_i, which slightly improves the detection performance.

3.3. Center-ness for FCOS

After using multi-level prediction in FCOS, there is still a performance gap between FCOS and anchor-based detectors. We observed that it is due to a lot of low-quality predicted bounding boxes produced by locations far away from the center of an object.

We propose a simple yet effective strategy to suppress these low-quality detected bounding boxes without introducing any hyper-parameters. Specifically, we add *a single-layer branch*, in parallel with the classification branch (as shown in Fig. 2) to predict the "center-ness" of a location[2].

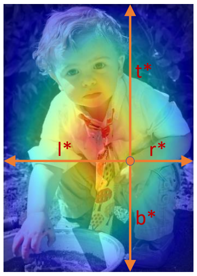

Figure 3 – Center-ness. Red, blue, and other colors denote 1, 0 and the values between them, respectively. Center-ness is computed by Eq. (3) and decays from 1 to 0 as the location deviates from the center of the object. When testing, the center-ness predicted by the network is multiplied with the classification score thus can down-weight the low-quality bounding boxes predicted by a location far from the center of an object.

The center-ness depicts the normalized distance from the location to the center of the object that the location is responsible for, as shown Fig. 3. Given the regression targets l^*, t^*, r^* and b^* for a location, the center-ness target is defined as,

$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}. \tag{3}$$

We employ sqrt here to slow down the decay of the center-ness. The center-ness ranges from 0 to 1 and is thus trained with binary cross entropy (BCE) loss. The loss is added to the loss function Eq. (2). When testing, the final score (used for ranking the detected bounding boxes) is computed by multiplying the predicted center-ness with the corresponding classification score. Thus the center-ness can down-weight the scores of bounding boxes far from the center of an object. As a result, with high probability, these low-quality bounding boxes might be filtered out by the final non-maximum suppression (NMS) process, improving the detection performance *remarkably*.

An alternative of the center-ness is to make use of only the central portion of ground-truth bounding box as positive samples with the price of one extra hyper-parameter, as shown in works [12, 33]. After our submission, it has been shown in [1] that the combination of both methods can achieve a much better performance. The experimental results can be found in Table 3.

4. Experiments

Our experiments are conducted on the large-scale detection benchmark COCO [16]. Following the common practice [15, 14, 24], we use the COCO `trainval35k` split (115K images) for training and `minival` split (5K images) as validation for our ablation study. We report our main results on the `test_dev` split (20K images) by uploading our detection results to the evaluation server.

Training Details. Unless specified, ResNet-50 [8] is used as our backbone networks and the same hyper-parameters with RetinaNet [15] are used. Specifically, our

[2] After the initial submission, it has been shown that the AP on MS-COCO can be improved if the center-ness is parallel with the regression branch instead of the classification branch. However, unless specified, we still use the configuration in Fig. 2.

Method	w/ FPN	Low-quality matches	BPR (%)
RetinaNet	✓	None	86.82
RetinaNet	✓	≥ 0.4	90.92
RetinaNet	✓	All	**99.23**
FCOS		-	95.55
FCOS	✓	-	98.40

Table 1 – The BPR for anchor-based RetinaNet under a variety of matching rules and the BPR for FCN-based FCOS. FCN-based FCOS has very similar recall to the best anchor-based one and has much higher recall than the official implementation in Detectron [7], where only low-quality matches with IOU ≥ 0.4 are considered.

w/ FPN	Amb. samples (%)	Amb. samples (diff.) (%)
	23.16	17.84
✓	**7.14**	**3.75**

Table 2 – Amb. samples denotes the ratio of the ambiguous samples to all positive samples. Amb. samples (diff.) is similar but excludes those ambiguous samples in the overlapped regions but belonging to the same category as the kind of ambiguity does not matter when inferring. We can see that with FPN, this percentage of ambiguous samples is small (3.75%).

network is trained with stochastic gradient descent (SGD) for 90K iterations with the initial learning rate being 0.01 and a mini-batch of 16 images. The learning rate is reduced by a factor of 10 at iteration 60K and 80K, respectively. Weight decay and momentum are set as 0.0001 and 0.9, respectively. We initialize our backbone networks with the weights pre-trained on ImageNet [4]. For the newly added layers, we initialize them as in [15]. Unless specified, the input images are resized to have their shorter side being 800 and their longer side less or equal to 1333.

Inference Details. We firstly forward the input image through the network and obtain the predicted bounding boxes with a predicted class. Unless specified, the following post-processing is exactly the same with RetinaNet [15] and we directly make use of the same post-processing hyper-parameters of RetinaNet. We use the same sizes of input images as in training. We hypothesize that the performance of our detector may be improved further if we carefully tune the hyper-parameters.

4.1. Ablation Study

4.1.1 Multi-level Prediction with FPN

As mentioned before, the major concerns of an FCN-based detector are *low recall* rates and *ambiguous samples* resulted from overlapping in ground-truth bounding boxes. In the section, we show that both issues can be largely resolved with multi-level prediction.

Best Possible Recalls. The first concern about the FCN-based detector is that it might not provide a good best possible recall (BPR). In the section, we show that the concern is not necessary. Here BPR is defined as the ratio of the number of ground-truth boxes a detector can recall at the most divided by all ground-truth boxes. A ground-truth box is considered being recalled if the box is assigned to at least one sample (*i.e.*, a location in FCOS or an anchor box in anchor-based detectors) during training. As shown in Table 1, only with feature level P_4 with stride being 16 (*i.e.*, no FPN), FCOS can already obtain a BPR of 95.55%. The BPR is much higher than the BPR of 90.92% of the anchor-based detector RetinaNet in the official implementation Detectron, where only the low-quality matches with IOU ≥ 0.4 are used. With the help of FPN, FCOS can achieve a BPR of 98.40%, which is very close to the best BPR that the anchor-based detector can achieve by using all low-quality matches. Due to the fact that the best recall of current detectors are much lower than 90%, the small BPR gap (less than 1%) between FCOS and the anchor-based detector will not actually affect the performance of detector. It is also confirmed in Table 3, where FCOS achieves even better AR than its anchor-based counterparts under the same training and testing settings. Therefore, the concern about low BPR may not be necessary.

Ambiguous Samples. Another concern about the FCN-based detector is that it may have a large number of *ambiguous samples* due to the overlapping in ground-truth bounding boxes, as shown in Fig. 1 (right). In Table 2, we show the ratios of the ambiguous samples to all positive samples on `minival` split. As shown in the table, there are indeed a large amount of ambiguous samples (23.16%) if FPN is not used and only feature level P_4 is used. However, with FPN, the ratio can be significantly reduced to only 7.14% since most of overlapped objects are assigned to different feature levels. Moreover, we argue that the ambiguous samples resulted from overlapping between objects of the same category do not matter. For instance, if object A and B with the same class have overlap, no matter which object the locations in the overlap predict, the prediction is correct because it is always matched with the same category. The missed object can be predicted by the locations only belonging to it. Therefore, we only count the ambiguous samples in overlap between bounding boxes with different categories. As shown in Table 2, the multi-level prediction reduces the ratio of ambiguous samples from 17.84% to 3.75%. In order to further show that the overlapping in ground truth boxes is not a problem of our FCN-based FCOS, we count that when inferring how many detected bounding boxes come from the ambiguous locations. We found that only 2.3% detected bounding boxes are produced by the ambiguous locations. By further only considering the overlap between different categories, the ratio is reduced to 1.5%. Note that it does not imply that there are 1.5% locations where FCOS cannot work. As mentioned before, these locations are associated with the ground-truth boxes with minimal area. Therefore, these locations only take the risk of missing some larger objects. As shown in the following experiments, they do not

Method	C_5/P_5	w/ GN	nms thr.	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR$_1$	AR$_{10}$	AR$_{100}$
RetinaNet	C_5		.50	35.9	56.0	38.2	20.0	39.8	47.4	31.0	49.4	52.5
FCOS	C_5		.50	36.3	54.8	38.7	20.5	39.8	47.8	31.5	50.6	53.5
FCOS	P_5		.50	36.4	54.9	38.8	19.7	39.7	48.8	31.4	50.6	53.4
FCOS	P_5		.60	36.5	54.5	39.2	19.8	40.0	48.9	31.3	51.2	54.5
FCOS	P_5	✓	.60	37.1	55.9	39.8	21.3	41.0	47.8	31.4	51.4	54.9
Improvements												
+ ctr. on reg.	P_5	✓	.60	37.4	56.1	40.3	21.8	41.2	48.8	31.5	51.7	55.2
+ ctr. sampling [1]	P_5	✓	.60	38.1	56.7	**41.4**	**22.6**	41.6	**50.4**	32.1	52.8	56.3
+ GIoU [1]	P_5	✓	.60	38.3	57.1	41.0	21.9	42.4	49.5	32.0	52.9	56.5
+ Normalization	P_5	✓	.60	**38.6**	**57.4**	**41.4**	22.3	**42.5**	49.8	**32.3**	**53.4**	**57.1**

Table 3 – FCOS vs. RetinaNet on the `minival` split with ResNet-50-FPN as the backbone. Directly using the training and testing settings of RetinaNet, our anchor-free FCOS achieves even better performance than anchor-based RetinaNet both in AP and AR. With Group Normalization (GN) in heads and NMS threshold being 0.6, FCOS can achieve 37.1 in AP. After our submission, some almost cost-free improvements have been made for FCOS and the performance has been improved by a large margin, as shown by the rows below "**Improvements**". "ctr. on reg.": moving the center-ness branch to the regression branch. "ctr. sampling": only sampling the central portion of ground-truth boxes as positive samples. "GIoU": penalizing the union area over the circumscribed rectangle's area in IoU Loss. "Normalization": normalizing the regression targets in Eq. (1) with the strides of FPN levels. Refer to our code for details.

	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
None	33.5	52.6	35.2	20.8	38.5	42.6
center-ness†	33.5	52.4	35.1	20.8	37.8	42.8
center-ness	**37.1**	**55.9**	**39.8**	**21.3**	**41.0**	**47.8**

Table 4 – Ablation study for the proposed center-ness branch on `minival` split. "None" denotes that no center-ness is used. "center-ness†" denotes that using the center-ness computed from the predicted regression vector. "center-ness" is that using center-ness predicted from the proposed center-ness branch. The center-ness branch improves the detection performance under all metrics.

make our FCOS inferior to anchor-based detectors.

4.1.2 With or Without Center-ness

As mentioned before, we propose "center-ness" to suppress the low-quality detected bounding boxes produced by the locations far from the center of an object. As shown in Table 4, the center-ness branch can boost AP from 33.5% to 37.1%, making anchor-free FCOS outperform anchor-based RetinaNet (35.9%). Note that anchor-based RetinaNet employs two IoU thresholds to label anchor boxes as positive/negative samples, which can also help to suppress the low-quality predictions. The proposed center-ness can eliminate the two hyper-parameters. However, after our initial submission, it has shown that using both center-ness and the thresholds can result in a better performance, as shown by the row "+ ctr. sampling" in Table 3. One may note that center-ness can also be computed with the predicted regression vector without introducing the extra center-ness branch. However, as shown in Table 4, the center-ness computed from the regression vector cannot improve the performance and thus the separate center-ness is necessary.

4.1.3 FCOS vs. Anchor-based Detectors

The aforementioned FCOS has two minor differences from the standard RetinaNet. 1) We use Group Normalization (GN) [29] in the newly added convolutional layers except for the last prediction layers, which makes our training more stable. 2) We use P_5 to produce the P_6 and P_7 instead of C_5 in the standard RetinaNet. We observe that using P_5 can improve the performance slightly.

To show that our FCOS can serve as an simple and strong alternative of anchor-based detectors, and for a fair comparison, we remove GN (the gradients are clipped to prevent them from exploding) and use C_5 in our detector. As shown in Table 3, with exactly the same settings, our FCOS still compares favorably with the anchor-based detector (36.3% vs 35.9%). Moreover, it is worth to note that we directly use all hyper-parameters (*e.g.*, learning rate, the NMS threshold and etc.) from RetinaNet, which have been optimized for the anchor-based detector. We argue that the performance of FCOS can be improved further if the hyper-parameters are tuned for it.

It is worth noting that with some almost cost-free improvements, as shown in Table 3, the performance of our anchor-free detector can be improved by a large margin. Given the superior performance and the merits of the anchor-free detector (*e.g.*, much simpler and fewer hyper-parameters than anchor-based detectors), we encourage the community to rethink the necessity of anchor boxes in object detection.

4.2. Comparison with State-of-the-art Detectors

We compare FCOS with other state-of-the-art object detectors on $\texttt{test}-\texttt{dev}$ split of MS-COCO benchmark. For these experiments, we randomly scale the shorter side of images in the range from 640 to 800 during the training and double the number of iterations to 180K (with the learning rate change points scaled proportionally). Other settings are exactly the same as the model with AP 37.1% in Table 3. As shown in Table 5, with ResNet-101-FPN, our FCOS outperforms the RetinaNet with the same backbone ResNet-101-FPN by 2.4% in AP. To our knowledge, it is

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods:							
Faster R-CNN w/ FPN [14]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [11]	Inception-ResNet-v2 [27]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [25]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [22]	DarkNet-19 [22]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [18]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [5]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [15]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [13]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FSAF [34]	ResNeXt-64x4d-101-FPN	42.9	63.8	46.3	26.6	46.2	52.7
FCOS	ResNet-101-FPN	41.5	60.7	45.0	24.4	44.8	51.6
FCOS	HRNet-W32-5l [26]	42.0	60.4	45.3	25.4	45.0	51.0
FCOS	ResNeXt-32x8d-101-FPN	42.7	62.2	46.1	26.0	45.6	52.6
FCOS	ResNeXt-64x4d-101-FPN	43.2	62.8	46.6	26.5	46.2	53.3
FCOS w/ improvements	ResNeXt-64x4d-101-FPN	**44.7**	**64.1**	**48.4**	**27.6**	**47.5**	**55.6**

Table 5 – FCOS vs. other state-of-the-art two-stage or one-stage detectors (*single-model and single-scale results*). FCOS outperforms the anchor-based counterpart RetinaNet by 2.4% in AP with the same backbone. FCOS also outperforms the recent anchor-free one-stage detector CornerNet with much less design complexity. Refer to Table 3 for details of "improvements".

Method	# samples	AR^{100}	AR^{1k}
RPN w/ FPN & GN (ReImpl.)	~200K	44.7	56.9
FCOS w/ GN w/o center-ness	~66K	48.0	59.3
FCOS w/ GN	~66K	**52.8**	**60.3**

Table 6 – FCOS as Region Proposal Networks vs. RPNs with FPN. ResNet-50 is used as the backbone. FCOS improves AR^{100} and AR^{1k} by 8.1% and 3.4%, respectively. GN: Group Normalization.

the first time that an anchor-free detector, without any bells and whistles outperforms anchor-based detectors by a large margin. FCOS also outperforms other classical two-stage anchor-based detectors such as Faster R-CNN by a large margin. With ResNeXt-64x4d-101-FPN [30] as the backbone, FCOS achieves 43.2% in AP. It outperforms the recent state-of-the-art anchor-free detector CornerNet [13] by a large margin while being much simpler. Note that CornerNet requires to group corners with embedding vectors, which needs special design for the detector. Thus, we argue that FCOS is more likely to serve as a strong and simple alternative to current mainstream anchor-based detectors. Moreover, FCOS with the improvements in Table 3 achieves 44.7% in AP with single-model and single scale testing, which surpasses previous detectors by a large margin.

5. Extensions on Region Proposal Networks

So far we have shown that in a one-stage detector, our FCOS can achieve even better performance than anchor-based counterparts. Intuitively, FCOS should be also able to replace the anchor boxes in Region Proposal Networks (RPNs) with FPN [14] in the two-stage detector Faster R-CNN. Here, we confirm that by experiments.

Compared to RPNs with FPN [14], we replace anchor boxes with the method in FCOS. Moreover, we add GN into the layers in FPN heads, which can make our training more stable. All other settings are exactly the same with RPNs with FPN in the official code [7]. As shown in Table 6, even without the proposed center-ness branch, our FCOS already improves both AR^{100} and AR^{1k} significantly. With the proposed center-ness branch, FCOS further boosts AR^{100} and AR^{1k} respectively to 52.8% and 60.3%, which are 18% relative improvement for AR^{100} and 3.4% absolute improvement for AR^{1k} over the RPNs with FPN.

6. Conclusion

We have proposed an anchor-free and proposal-free one-stage detector FCOS. As shown in experiments, FCOS compares favourably against the popular anchor-based one-stage detectors, including RetinaNet, YOLO and SSD, but with much less design complexity. FCOS completely avoids all computation and hyper-parameters related to anchor boxes and solves the object detection in a per-pixel prediction fashion, similar to other dense prediction tasks such as semantic segmentation. FCOS also achieves state-of-the-art performance among one-stage detectors. We also show that FCOS can be used as RPNs in the two-stage detector Faster R-CNN and outperforms the its RPNs by a large margin. Given its effectiveness and efficiency, we hope that FCOS can serve as a strong and simple alternative of current mainstream anchor-based detectors. We also believe that FCOS can be extended to solve many other instance-level recognition tasks.

Acknowledgments. We would like to thank the author of [1] for the tricks of center sampling and GIoU. We also thank Chaorui Deng for HRNet based FCOS and his suggestion of positioning the center-ness branch with box regression.

References

[1] https://github.com/yqyao/FCOS_PLUS, 2019.

[2] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proc. ACM Int. Conf. Multimedia*, pages 640–644. ACM, 2016.

[3] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial PoseNet: A structure-aware convolutional network for human pose estimation. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255. IEEE, 2009.

[5] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander Berg. DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[6] Ross Girshick. Fast R-CNN. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1440–1448, 2015.

[7] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016.

[9] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2019.

[10] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5020–5029, 2018.

[11] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7310–7311, 2017.

[12] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[13] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proc. Eur. Conf. Comp. Vis.*, pages 734–750, 2018.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2117–2125, 2017.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2980–2988, 2017.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755. Springer, 2014.

[17] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *Proc. Eur. Conf. Comp. Vis.*, pages 21–37. Springer, 2016.

[19] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2019.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015.

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 779–788, 2016.

[22] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7263–7271, 2017.

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 91–99, 2015.

[25] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[26] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[27] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. National Conf. Artificial Intell.*, 2017.

[28] Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3126–3135, 2019.

[29] Yuxin Wu and Kaiming He. Group normalization. In *Proc. Eur. Conf. Comp. Vis.*, pages 3–19, 2018.

[30] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1492–1500, 2017.

[31] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[32] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proc. ACM Int. Conf. Multimedia*, pages 516–520. ACM, 2016.

[33] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5551–5560, 2017.

[34] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2019.