FCOS: A Simple and Strong Anchor-free Object Detector

Zhi Tian, Chunhua Shen, Hao Chen, Tong He

Abstract—In computer vision, object detection is one of most important tasks, which underpins a few instance-level recognition tasks and many downstream applications. Recently one-stage methods have gained much attention over two-stage approaches due to their simpler design and competitive performance. Here we propose a fully convolutional one-stage object detector (FCOS) to solve object detection in a per-pixel prediction fashion, analogue to other dense prediction problems such as 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 pre-defined set of anchor boxes, FCOS completely avoids the complicated computation related to anchor boxes such as calculating the intersection over union (IoU) scores during training. More importantly, we also avoid all hyper-parameters related to anchor boxes, which are often sensitive to the final detection performance. With the only post-processing non-maximum suppression (NMS), 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: `git.io/AdelaiDet`

Index Terms—Object detection, fully convolutional one-stage object detection, anchor box, deep learning.

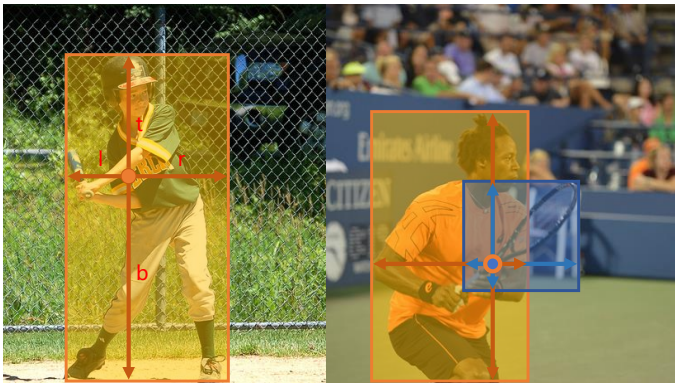

Fig. 1. **Overall concept of FCOS**. As shown in the left image, FCOS works by predicting a 4D vector (l, t, r, b) encoding the location of a bounding box at each foreground pixel (supervised by ground-truth bounding box information during training). The right plot shows that when a location residing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

1 Introduction

Object detection requires an algorithm to predict a bounding box location and a category label for each instance of interest in an image. Prior to deep learning, the sliding-window approach was the main method [7], [35], [43], which exhaustively classifies every possible location, thus requiring feature extraction and classification evaluation to be very fast. With deep learning, detection has been largely shifted to the use of fully convolutional networks (FCNs) since the invention of Faster R-CNN [32]. All current mainstream detectors such as Faster R-CNN [32], SSD [25] and YOLOv2, v3 [31] 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 modern detectors' success*. Despite their great success, it is important to note that anchor-based detectors suffer some drawbacks:

- As shown in Faster R-CNN and RetinaNet [22], detection performance is sensitive to the sizes, aspect ratios and number of anchor boxes. For example, in RetinaNet, varying these hyper-parameters affects the performance up to 4% in AP on the COCO benchmark [23]. As a result, these hyper-parameters need to be carefully tuned in anchor-based detectors.
- 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.
- 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) [21] for an image with its shorter side being 800). Most of these anchor boxes are labeled as negative samples during training. The excessive number of negative samples aggravates the imbalance between positive and negative samples in training.
- Anchor boxes also involve complicated computation such as calculating the intersection-over-union (IoU) scores with ground-truth bounding boxes.

Recently, FCNs [28] have achieved tremendous success in dense prediction tasks such as semantic segmentation [15], [28], [40], depth estimation [24], [48], keypoint detection [3] and counting. As one of high-level vision tasks, ob-

Authors are with The University of Adelaide, Australia. C. Shen is the corresponding author (e-mail: chunhua.shen@adelaide.edu.au).

ject detection might be the only one deviating from the neat fully convolutional per-pixel prediction framework mainly due to the use of anchor boxes.

It is natural 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 in this work that the answer is affirmative. Moreover, we demonstrate that, the much simpler FCN-based detector can surprisingly achieve 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 [18]. 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 [18] 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 [16], [53] or face detection [18], [50], 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 similar or even better 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. It is easy to see that the locations near the center of its target bounding box can make more reliable predictions. As a result, we introduce a novel "center-ness" score to depict the deviation of a location to the center, as defined in Eq. (3), which is used to down-weigh low-quality detected bounding boxes and thus helps to suppress these low-quality detections in NMS. The center-ness score is predicted by a branch (only one layer) in parallel with the bounding box regression branch, as shown in Fig. 2. The simple yet effective center-ness branch remarkably improves the detection performance with a negligible increase in computational time.

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. An example is shown in [27], where a structured knowledge distillation method was developed for dense prediction tasks. Thanks to the standard FCN framework of FCOS, the developed technique can be immediately applied to FCOS based object detection.
- 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 than its anchor-based counterpart.
- Without bells and whistles, we achieve state-of-the-art results among one-stage detectors. Given its improved accuracy 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 designing detection methods.
- With considerably reduced design complexity, our proposed detector outperforms previous strong baseline detectors such as Faster R-CNN [32], RetinaNet [22], YOLOv3 [31] and SSD [25]. More importantly, due to its simple design, FCOS can be easily extended to solve other instance-level recognition tasks with minimal modification, as already evidenced by instance segmentation [2], [20], [47], [51], keypoint detection [39], text spotting [26], and tracking [13], [44]. We expect to see more instance recognition methods built upon FCOS.

2 Related Work

Here we review some work that is closest to ours.

Anchor-based Detectors. Anchor-based detectors inherit the ideas from traditional sliding-window and proposal based detectors such as Fast R-CNN [11]. 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 [32], SSD [25] and YOLOv2 [30], 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 hyper-parameters to label each anchor box as a positive, ignored or negative sample. In previous works, they often

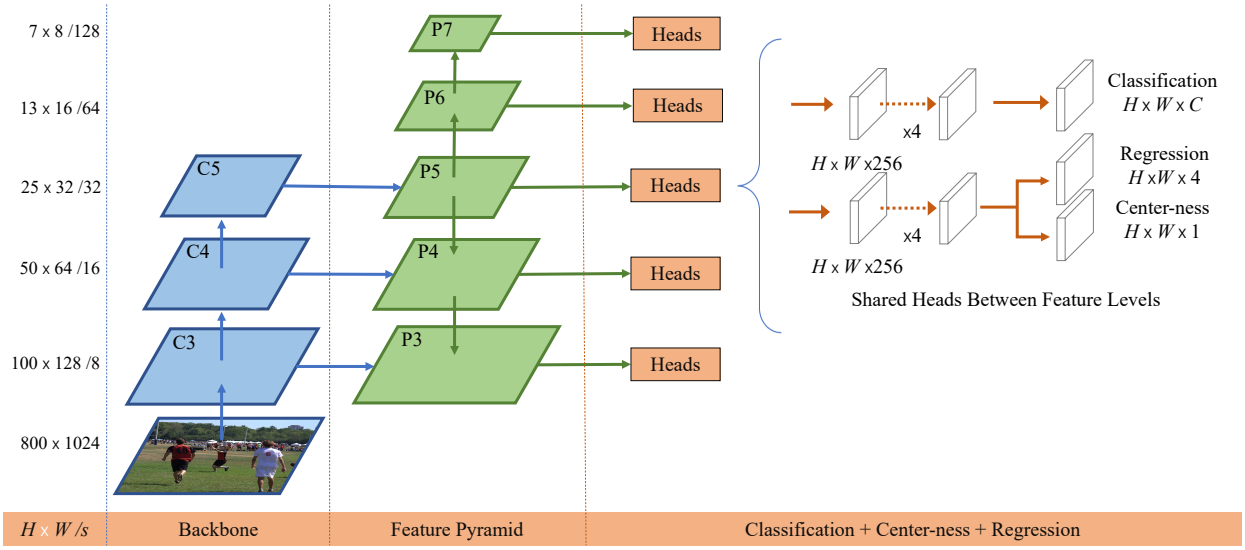

Fig. 2. **The network architecture of FCOS**, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. '/s' ($s = 8, 16, ..., 128$) is the down-sampling ratio of the feature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

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 [29]. 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 [30]. As a result, YOLOv2 [30] employs anchor boxes as well. Compared to YOLOv1, FCOS can take advantages of all points in a ground truth bounding box to predict the bounding boxes and the low-quality detected bounding boxes can be 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 [19] 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 [50] are based on DenseBox [18]. 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 much better detection performance than its anchor-based counterparts. Recently, FSAF [54] was proposed to employ an anchor-free detection branch as a complement to an anchor-based detection branch since they consider that a totally anchor-free detector cannot achieve good performance. They also make use of a feature selection module to improve the performance of the anchor-free branch, making the anchor-free detector have a comparable performance to its anchor-based counterpart. However, in this work, we surprisingly show that the totally anchor-free detector can actually obtain better performance than its anchor-based counterpart, without the need for the feature selection module in FSAF. Even more surprisingly, it can outperform the combination of anchor-free and anchor-based detectors in FSAF. As a result, the long-standing anchor-boxes can be completely eliminated, making detection significantly simpler.

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 $B_i = (x_0^{(i)}, y_0^{(i)}, x_1^{(i)} y_1^{(i)}, c^{(i)}) \in \mathbb{R}^4 \times \{1, 2 \ ... \ C\}$. 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

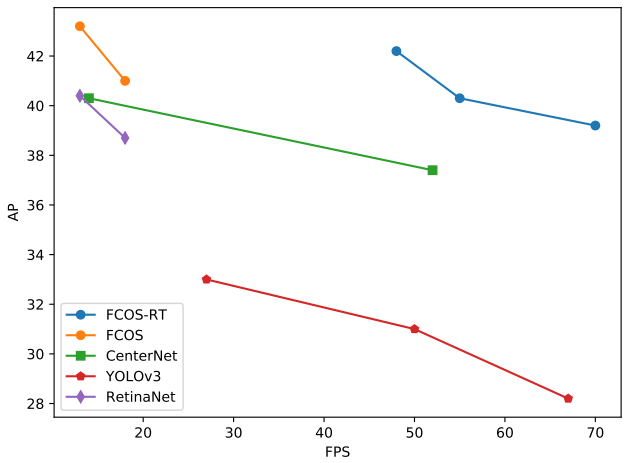

Fig. 3. **Speed/accuracy trade-off between FCOS and several recent methods**: CenterNet [52], YOLOv3 [31] and RetinaNet [22]. Speed is measured on a NVIDIA 1080Ti GPU. For fair comparison, we only measure the network latency for all detectors. RetinaNet results are from `Detectron2`. FCOS achieves competitive performance compared with recent methods including anchor-based ones.

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 the 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 [28].

Specifically, location (x, y) is considered as a positive sample if it falls into the center area of any ground-truth box, by following [1]. The center area of a box centered at (c_x, c_y) is defined as the sub-box $(c_x - rs, c_y - rs, c_x + rs, c_y + rs)$, where s is the total stride until the current feature maps and r is a hyper-parameter being 1.5 on COCO. The sub-box is clipped so that it is not beyond the original box. Note that this is different from our original conference version [42], where we consider the locations positive as long as they are in a ground-truth box. 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 the center area of 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,

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

where s is the total stride until the feature maps F_i, which is used to scale down regression targets and prevents the gradients from exploding during training. Together with these designs, FCOS can detect objects in an anchor-free way and everything is learned by the networks without the need for any pre-defined anchor-boxes. *It is worth noting that this is not identical to an anchor-based detector with one anchor-box per location*, the crucial difference is the way we define positive and negative samples. The single-anchor detector still uses pre-defined anchor-boxes as a prior and uses IoUs between the anchor-boxes and ground-truth boxes to determine the labels for these anchor-boxes. In FCOS, we remove the need for the prior and the locations are labeled by their inclusion in ground-truth boxes. In experiments, we will show that using a single anchor can only achieve inferior performance.

Network Outputs. Corresponding to the training targets, the final layer of our networks predicts an 80D vector $\boldsymbol{p}$ for classification and a 4D vector $\boldsymbol{t} = (l, t, r, b)$ encoding bounding-box coordinates. Following [22], instead of training a multi-class classifier, we train C binary classifiers. Similar to [22], we add two branches, respectively with four convolutional layers (exclude the final prediction layers) after the feature maps produced by FPNs for classification and regression tasks, respectively. Moreover, since the regression targets are always positive, we employ $\mathrm{ReLU}(x)$ 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 [22], [32] with 9 anchor boxes per location*, which is of great importance when FCOS is applied to keypoint detection [39] or instance segmentation [41].

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 [22] and L_{reg} is the GIoU loss [33]. As shown in experiments, the GIoU loss has better performance than the IoU loss in UnitBox [50], which is used in our preliminary version [42]. 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 [22], 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 [21].

First, 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 IOU score requirements 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, FCOS is still able to produce a good BPR, and it can even better than the BPR of the anchor-based detector RetinaNet [22] in the official implementation Detectron [12] (refer to Table 1). Therefore, the BPR is actually not a problem of FCOS. Moreover, with multi-level FPN prediction [21], the BPR can be improved further to match the best BPR the anchor-based RetinaNet can achieve.

Second, as shown in Fig. 1 (right), 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. In this work, we show that the ambiguity can be greatly resolved with multi-level prediction, and FCOS can obtain *on par*, sometimes even better, performance compared with anchor-based ones.

Specifically, following FPN [21], we detect different size objects on different feature map levels. we make use of five levels of feature maps defined as $\{P_3, P_4, P_5, P_6, P_7\}$. As shown in Fig. 2, P_3, P_4 and P_5 are produced by the backbone CNNs' feature maps C_3, C_4 and C_5 with the top-down connections as in [21]. P_6 and P_7 are produced by applying one 3×3 convolutional layer with the stride being 2 on P_5 and P_6, respectively. Note that this is different from the original RetinaNet, which obtain P_6 and P_7 from the backbone feature maps C_5. We find both schemes achieve similar performance but the one we use has fewer parameters. Moreover, the feature levels P_3, P_4, P_5, P_6 and P_7 have strides 8, 16, 32, 64 and 128, respectively.

Anchor-based detectors assign different scale anchor boxes to different feature levels. Since anchor boxes and ground-boxes are associated by their IoU scores, this enables different FPN feature levels to handle different scale objects. However, *this couples the sizes of anchor boxes and the target object sizes of each FPN level*, which is problematic. The anchor box sizes should be data-specific, which might be changed from one dataset to another. The target object sizes of each FPN level should depend on the receptive field of the FPN level, which depends on the network architecture. FCOS removes the coupling as we only need focus on the target object sizes of each FPN level and need not design the anchor box sizes. Unlike anchor-based detectors, in FCOS, we directly limit the range of bounding box regression for each level. More specifically, we first compute the regression targets l^*, t^*, r^* and b^* for each location on all feature levels. Next, if a location at feature level i satisfies $\max(l^*, t^*, r^*, b^*) \leq m_{i-1}$ or $\max(l^*, t^*, r^*, b^*) \geq m_i$, it is set as a negative sample and 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. We argue that bounding the maximum distance is a better way to determine the range of target objects for each feature level because this makes sure that the complete objects are always in the receptive field of each feature level. Moreover, since objects of different sizes are assigned to different feature levels and overlapping mostly happens between objects with considerably different sizes, the aforementioned ambiguity can be largely alleviated. 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, with the multi-level prediction, both anchor-free and anchor-based detectors can achieve the same level performance.

Finally, following [21], [22], 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 may not be the optimal design to make use of identical heads for different feature levels. In our preliminary version [42], this issue is addressed by multiplying a learnable scalar to the convolutional layer's outputs. In this version, since the regression targets are scaled down by the stride of FPN feature levels, as shown in Eq. (1), the scalars become less important. However, we still keep them for compatibility.

3.3 Center-ness for FCOS

After using multi-level prediction, FCOS can already achieve better performance than its anchor-based counterpart RetinaNet. Furthermore, we observed that there are a lot of low-quality detections produced by the locations far away from the center of an object.

We propose a simple yet effective strategy to suppress these low-quality detections. Specifically, we add *a single-layer branch*, in parallel with the regression branch (as shown in Fig. 2) to predict the "center-ness" of a location[2]. 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. 4. Given the regression targets l^*, t^*, r^* and b^* for a location, the center-ness target is defined as,

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

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 $\boldsymbol{s}_{x,y}$ (used for ranking the detections in NMS) is the square

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

2. This is different from our conference version which positions the center-ness on the classification branch, but it has been shown that positioning it on the regression branch can obtain better performance.

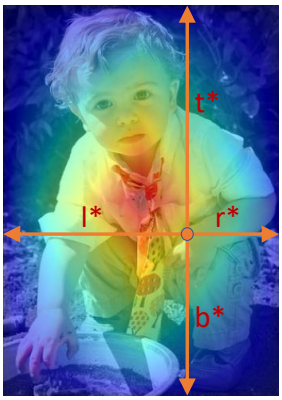

Fig. 4. **Center-ness.** Red, blue, and other colors denote 1, 0 and the values between them, respectively. Center-ness is computed using Eq. (3) and decays from 1 to 0 as the location deviates from the center of the object. During testing, the center-ness predicted by the network is multiplied with the classification score for NMS, thus being able to down-weight the low-quality bounding boxes predicted by a location far from the center of an object.

root of the product of the predicted center-ness $o_{x,y}$ and the corresponding classification score $\boldsymbol{p}_{x,y}$. Formally,

$$\boldsymbol{s}_{x,y} = \sqrt{\boldsymbol{p}_{x,y} \times o_{x,y}}, \tag{4}$$

where sqrt is used to calibrate the order of magnitude of the final score and has no effect on average precision (AP).

Consequently, 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*.

4 Experiments

Our experiments are conducted on the large-scale detection benchmark COCO [23]. Following the common practice [21], [22], [32], we use the COCO `train2017` split (115K images) for training and `val2017` 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, we use the following implementation details. ResNet-50 [14] is used as our backbone networks and the same hyper-parameters with RetinaNet [22] are used. Specifically, our 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 [6]. For the newly added layers, we initialize them as in [22]. 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 the predicted class scores. The next post-processing of FCOS exactly follows that of RetinaNet [22]. The post-processing hyper-parameters are also the same except that we use NMS threshold 0.6 instead of 0.5 in RetinaNet. Experiments will be conducted to show the effect of the NMS threshold. Moreover, we use the same sizes of input images as in training.

Method	w/ FPN	Low-quality matches	BPR (%)
RetinaNet	✓	Not used	88.16
RetinaNet	✓	≥ 0.4	91.94
RetinaNet	✓	All	**99.32**
FCOS		-	96.34
FCOS	✓	-	98.95

TABLE 1
The best possible recall (BPR) of anchor-based RetinaNet under a variety of matching rules and the BPR of FCOS on the COCO `val2017` split. FCOS has very similar BPR to the best anchor-based one and has much higher recall than the official implementation in `Detectron` [12], where only low-quality matches with IOU ≥ 0.4 are considered.

4.1 Analysis of FCOS

4.1.1 Best Possible Recall (BPR) of FCOS

We first address the concern that is FCOS might not provide a good best possible recall (BPR) (*i.e.*, upper bound of the recall rate). In the section, we show that the concern is not necessary by comparing BPR of FCOS and that of its anchor-based counterpart RetinaNet on the COCO `val2017` split. The following analyses are based on the FCOS implementation in `AdelaiDet`. Formally, BPR is defined as the ratio of the number of ground-truth boxes that a detector can recall at the most to the number of all ground-truth boxes. A ground-truth box is considered recalled if the box is assigned to at least one training sample (*i.e.*, a location in FCOS or an anchor box in anchor-based detectors), and a training sampling can be associated to *at most* one ground-truth box. As shown in Table 1, both with FPN, FCOS and RetinaNet obtain similar BPR (98.95 vs 99.32)[3]. Due to the fact that the best recall of current detectors are much lower than 90%, the small BPR gap (less than 0.5%) between FCOS and the anchor-based RetinaNet will not actually affect the performance of a detector. It is also confirmed in Table 3, where FCOS achieves better or similar AR than RetinaNet under the same training and testing settings. Even more

3. One might think that the BPR of RetinaNet should be 1 if all the low-quality matches are used. However, this is not true in some cases as each anchor can only be associated to the ground-truth box with the highest IOU to it. For example, if two boxes A and B, both of which are small and contained in the common of all the anchor boxes at the same location. Clearly, for all these anchor boxes, the box with larger area has higher IOU scores and thus all the anchor boxes will be associated to it. Another one will be missing.

w/ ctr. sampling	w/ FPN	1	2	≥ 3
		76.60%	20.05%	3.35%
	✓	92.58%	6.97%	0.45%
✓		96.52%	3.34%	0.14%
✓	✓	97.34%	2.59%	0.07%

TABLE 2
The ratios of the ambiguous samples to all the positive samples in FCOS. $1, 2$ and ≥ 3 denote the number of ground-truth boxes a location should be associated to. If the number is greater than 1, the location is defined as an "ambiguous sample" in this work. As shown in the table, with center sampling and FPN, the ratio of ambiguous samples is low (*i.e.*, $< 3\%$).

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
RetinaNet (#A=9)	35.9	55.8	38.4	20.6	39.8	46.6	31.0	49.8	53.0	33.8	57.4	67.9
RetinaNet (#A=1) w/ imprv.	35.2	55.6	37.0	19.9	39.2	45.2	30.4	49.9	53.5	33.6	57.7	68.2
RetinaNet (#A=9) w/ imprv.	37.6	56.6	40.6	21.5	42.1	48.0	32.1	52.2	56.4	35.5	60.2	72.7
FCOS w/o ctr.-ness	38.0	57.2	40.9	21.5	42.4	49.1	32.1	52.4	56.2	36.6	60.6	71.9
FCOS w/ ctr.-ness	**38.9**	*57.5*	**42.2**	**23.1**	**42.7**	**50.2**	**32.4**	**53.8**	**57.5**	**38.5**	**62.1**	**72.9**

TABLE 3

FCOS vs. RetinaNet on `val2017` split with ResNet-50-FPN as the backbone. All experiments use the same training settings. The proposed anchor-free FCOS achieves even better performance than anchor-based RetinaNet. #A: the number of anchors per location. RetinaNet (#A=9): the original RetinaNet from `Detectron2` [46]. RetinaNet w/ imprv. RetinaNet with the universal improvements in FCOS including Group Normalization (GN) [45], GIoU loss [1] and scalars in regression, using P_5 instead of C_5 and NMS threshold 0.6 instead of 0.5. We have tried our best to make all the details consistent. As shown the table, even without the center-ness branch, the much simpler FCOS already outperforms "RetinaNet (#A=9) w/ imprv" by 0.6% in AP. With the center-ness branch, the performance is further improved to 38.9% in AP.

surprisingly, only with feature level P_4 with stride being 16 (*i.e.*, no FPN), FCOS can obtain a decent BPR of 96.34%. The BPR is much higher than the BPR of 91.94% of the RetinaNet in the official implementation `Detectron` [12], where only the low-quality matches with IOU ≥ 0.4 are used. Therefore, the concern about low BPR may not be necessary.

4.1.2 Ambiguous Samples in FCOS

Another concern about the FCN-based detector is that it may have a large number of *ambiguous samples* due to the overlap in ground-truth boxes, as shown in Fig. 1 (right). In Table 2, we show the ratios of the ambiguous samples to all positive samples on `val2017` split. If a location should be associated to multiple ground-truth boxes without using the rule of choosing the box with minimum area, the location is defined as an "ambiguous sample". As shown in the table, there are indeed a large amount of ambiguous samples (23.40%) if FPN is not used (*i.e.*, only P_4 used). However, with FPN, the ratio can be significantly reduced to only 7.42% since most of overlapped objects are assigned to different feature levels. Furthermore, if the center sampling is used, the ambiguous samples can be significantly reduced. As shown in Table 2, even without FPN, th ratio is only 3.48%. By further applying FPN, the ratio is reduced to 2.66%. Note that it does not imply that there are 2.66% locations where FCOS makes mistakes. As mentioned before, these locations are associated with the smallest one among the ground-truth boxes associated to the same location. Therefore, these locations only take the risk of missing some larger objects. In other words, it may harm the recall of FCOS. However, as shown in Table 1, the recall gap between FCOS and RetinaNet is negligible, which suggests that the ratio of the missing objects is extremely low.

4.1.3 The Effect of 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 38.2% to 38.9%. Compared to our conference version [42], the gap is relatively smaller since we make use of the center sampling by default and it already eliminates a large number of false positives. However, the improvement is still impressive as the center-ness branch only adds negligible computational time. Moreover, we will show later that the center-ness can bring a large improvement in crowded scenarios. One

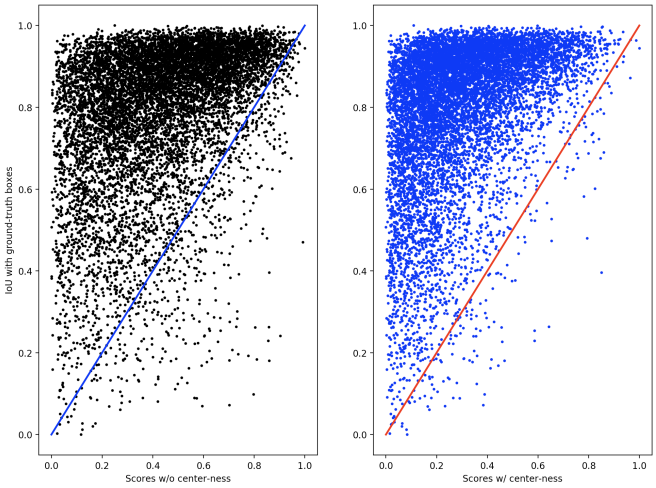

Fig. 5. **Quantitative results of applying the center-ness scores to classification scores**. A point in the figure denotes a bounding box. The dashed line is the line $y = x$. As shown in the right figure, after applying the center-ness scores, the boxes with low IoU scores but high confidence scores (*i.e.*, under the line $y = x$) are reduced substantially.

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.

We visualize the effect of applying the center-ness in Fig. 4. As shown in the figure, after applying the center-ness scores to the classification scores, the boxes with low IoU scores but high confidence scores are largely eliminated

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o ctr.-ness	38.0	57.2	40.9	21.5	42.4	49.1
w/ ctr.-ness†	37.5	56.5	40.2	21.6	41.5	48.5
w/ ctr.-ness (L1)	**38.9**	**57.6**	42.0	23.0	42.3	**51.0**
w/ ctr.-ness	**38.9**	57.5	**42.2**	**23.1**	**42.7**	**50.2**

TABLE 4

Ablation study for the proposed center-ness branch on the `val2017` split. ctr.-ness†: using the center-ness computed from the predicted regression vector when testing. "ctr.-ness" is that using center-ness predicted from the proposed center-ness branch. The center-ness branch improves the detection performance. On the contrary, using "ctr.-ness†" even degrades the performance, which suggests that the separate center-ness branch is necessary. w/ ctr.-ness (L1): using L1 instead of BCE as the loss to optimize the center-ness.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Baseline	**38.9**	**57.5**	**42.2**	**23.1**	**42.7**	**50.2**
w/o GN	37.9	56.4	40.9	22.1	41.8	48.8
w/ IoU	38.6	57.2	41.9	22.4	42.1	49.8
w/ C_5	38.5	57.4	41.7	22.8	42.1	49.3

TABLE 5
Ablation study for design choices in FCOS. w/o GN: without using Group Normalization (GN) for the convolutional layers in heads. w/ IoU: using IoU loss in [50] instead of GIoU. w/ C_5: using C_5 instead of P_5.

r	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1.0	38.5	57.2	41.5	22.6	42.3	49.7
1.5	**38.9**	57.5	**42.2**	**23.1**	**42.7**	**50.2**
2.0	38.8	**57.7**	41.7	22.7	42.6	49.9

TABLE 6
Ablation study for the radius r of positive sample regions (defined in Section 3.1).

(*i.e.*, the points under the line $y = x$ in the Fig. 4), which are potential false positives.

4.1.4 Other Design Choices

Other design choices are also investigated. As shown Table 5, removing group normalization (GN) [45] in both the classification and regression heads drops the performance by 1% AP. By replacing GIoU [33] with the origin IoU loss in [50], the performance drops by 0.3% AP. Using C_5 instead of P_5 also degrades the performance. Moreover, using P_5 can reduce the number of the network parameters. We also conduct experiments for the radius r of positive sample regions. As shown in Table 6, $r = 1.5$ has the best performance on COCO `val` split.

We also conduct experiments with different strategies of assigning objects to FPN levels. First, we experiment with the assigning strategy when FPN [21] assigns the object proposals (*i.e.*, ROIs) to FPN levels. It assigns the objects according to the formulation $k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor$, where $k \in \{3, 4, 5, 6, 7\}$ is the target FPN level, w and h are the ground-truth box's width and height, respectively, and k_0 is the target level which an object with scale 224 should be mapped into. We use $k_0 = 5$. As shown in Table 7, this strategy results in degraded performance (37.7% AP). We conjecture that it may be because the strategy cannot make sure the complete object be within the receptive field of the target FPN level.

Similarly, $\sqrt{(h^* \times w^*)}/2$ and $\max(h^*, w^*)/2$ also deteriorate the performance. Eventually, $\max(l^*, t^*, r^*, b^*)$ achieves the best performance as the strategy makes sure that the complete target objects are always in the effective receptive field of the FPN level. Moreover, this implies that the range hyper-parameters of each FPN level (*i.e.*, m_i) is mainly related to the network architecture (which determines the receptive fields). This is a desirable feature since it eliminates the hyper-parameter tuning when FCOS is applied to different datasets.

Strategy	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FPN	37.7	56.6	40.6	22.2	40.9	49.7
$\sqrt{(h^* \times w^*)}/2$	37.6	56.5	40.6	22.4	41.6	47.3
$\max(h^*, w^*)/2$	38.1	57.0	41.3	22.5	41.8	48.7
$\max(l^*, t^*, r^*, b^*)$	**38.9**	**57.5**	**42.2**	**23.1**	**42.7**	**50.2**

TABLE 7
Ablation study for different strategies of assigning objects to FPN levels. FPN: the strategy of assigning object proposals (*i.e.*, ROIs) to FPN levels in the original FPN, described in the text. h^* and w^* are the height and width of a ground-truth box, respectively. l^*, t^*, r^* and b^* are the distances from a location to the four boundaries of a ground-truth box. "$\max(l^*, t^*, r^*, b^*)$" (used by FCOS) has the best performance.

4.2 FCOS vs. Anchor-based Counterparts

Here, we compare FCOS with its anchor-based counterpart RetinaNet on the challenging benchmark COCO, demonstrating that the much simpler anchor-free FCOS is superior.

In order to make a fair comparison, we add the universal improvements in FCOS to RetinaNet. The improved RetinaNet is denoted as "*RetinaNet w/ imprv.*" in Table 3. As shown the table, even without the center-ness branch, FCOS achieves 0.4% better AP than "RetinaNet (#A=9) w/ imprv." (38.0% vs 37.6% in AP). The performance of FCOS can be further boosted to 38.9% with the help of the proposed center-ness branch. Moreover, it is worth noting that FCOS achieves much better performance than the RetinaNet with a single anchor per location "RetinaNet (#A=1) w/ imprv." (38.0% vs 35.2%), which suggests that FCOS is not equivalent to the single-anchor RetinaNet. The major difference is FCOS does not employ IoU scores between anchor boxes and ground-truth boxes to determine the training labels.

Given the superior performance and merits of the anchor-free detector (*e.g.*, much simpler and fewer hyper-parameters), we encourage the community to rethink the necessity of anchor boxes in object detection.

4.3 Comparison with State-of-the-art Detectors on COCO

We compare FCOS with other state-of-the-art object detectors on `test-dev` split of MS-COCO benchmark. For these experiments, following previous works [22], [25], we make use of multi-scale training. To be specific, during training, the shorter side of the input image is sampled from $[640, 800]$ with a step of 32. Moreover, we 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 38.9% on `val2017` in Table 3.

As shown in Table 8, with ResNet-101-FPN, FCOS outperforms the original RetinaNet with the same backbone by 4.1% AP (43.2% vs 39.1%). Compared to other one-stage detectors such as SSD [25] and DSSD [10], we also achieve much better performance. Moreover, FCOS also surpasses the classical two-stage anchor-based detector Faster R-CNN by a large margin (43.2% vs. 36.2%). To our knowledge, it is the first time that an anchor-free detector, without any bells and whistles, outperforms anchor-based detectors by a large margin. Moreover, FCOS also outperforms the previous anchor-free detector CornerNet [19] and CenterNet [9] while being much simpler since they 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. Quantitative results are shown in

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods:							
Faster R-CNN+++ [14]	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w/ FPN [21]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [37]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [36]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [30]	DarkNet-19 [30]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [25]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 608×608 [31]	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9
DSSD513 [10]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [22]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [19]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FSAF [54]	ResNeXt-64x4d-101-FPN	42.9	63.8	46.3	26.6	46.2	52.7
CenterNet511 [9]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
FCOS	ResNet-101-FPN	43.2	62.4	46.8	26.1	46.2	52.8
FCOS	ResNeXt-32x8d-101-FPN	44.1	63.7	47.9	27.4	46.8	53.7
FCOS	ResNeXt-64x4d-101-FPN	44.8	64.4	48.5	27.7	47.4	55.0
FCOS w/ deform. conv. v2 [55]	ResNeXt-32x8d-101-FPN	46.6	65.9	50.8	28.6	49.1	58.6
FCOS	ResNet-101-BiFPN [38]	45.0	63.6	48.7	27.0	47.9	55.9
FCOS	ResNeXt-32x8d-101-BiFPN	46.2	65.2	50.0	28.7	49.1	56.5
FCOS w/ deform. conv. v2	ResNeXt-32x8d-101-BiFPN	47.9	66.9	51.9	30.2	50.3	59.9
w/ test-time augmentation:							
FCOS	ResNet-101-FPN	45.9	64.5	50.4	29.4	48.3	56.1
FCOS	ResNeXt-32x8d-101-FPN	47.0	66.0	51.6	30.7	49.4	57.1
FCOS	ResNeXt-64x4d-101-FPN	47.5	66.4	51.9	31.4	49.7	58.2
FCOS w/ deform. conv. v2	ResNeXt-32x8d-101-FPN	49.1	68.0	53.9	31.7	51.6	61.0
FCOS	ResNet-101-BiFPN	47.9	65.9	52.5	31.0	50.7	59.7
FCOS	ResNeXt-32x8d-101-BiFPN	49.0	67.4	53.6	32.0	51.7	60.5
FCOS w/ deform. conv. v2	ResNeXt-32x8d-101-BiFPN	**50.4**	**68.9**	**55.0**	**33.2**	**53.0**	**62.7**

TABLE 8
FCOS vs. other state-of-the-art two-stage or one-stage detectors (*single-model results*). FCOS outperforms a few recent anchor-based and anchor-free detectors by a considerable margin.

Fig. 6. It appears that FCOS works well with a variety of challenging cases.

We also introduce some complementary techniques to FCOS. First, deformable convolutions are used in stages 3 and 4 of the backbone, and also replace the last convolutional layers in the classification and regression towers (*i.e.*, the $4\times$ convolutions shown in Fig. 2). As shown in Table 8, by applying deformable convolutions [5], [55] to ResNeXt-32x8d-101-FPN based FCOS, the performance is improved from 44.1% to 46.6% AP, as shown in Table 8. In addition, we also attempt to replace FPN in FCOS with BiFPN [38]. We make use of BiFPN in D3 model in [38]. To be specific, the single cell of BiFPN is repeated 6 times and the number of its output channels is set to 160. Note that unlike the original BiFPN, we do not employ depthwise separable convolutions in it. As a result, BiFPN generally improves all FCOS models by $\sim 2\%$ AP and pushes the performance of the best model to 47.9%.

We also report the result of using test-time data augmentation. Specifically, in inference, the input image is respectively resized to $[400, 1200]$ pixels with step 100. At each scale, the original image and its horizontal flip are evaluated. The results from these augmented images are merged by NMS. As shown in Table 8, the test-time augmentation improves the best performance to 50.4% AP.

4.4 Real-time FCOS

We also design a real-time version FCOS-RT. In the real-time settings, we reduce the shorter side of input images from 800 to 512 and the maximum longer size from 1333 to 736, which decreases the inference time per image by $\sim 50\%$. With the smaller input size, the higher feature levels P_6 and P_7 become less important. Thus, following BlendMask-RT [2], we remove P_6 and P_7, further reducing the inference time. Moreover, in order to boost the performance of the real-time version, we employ a more aggressive training strategy. Specifically, during training, multi-scale data augmentation is used and the shorter size of input image is sampled from 256 to 608 with interval 32. Synchronized batch normalization (SyncBN) is used. We also increase the training iterations to $360K$ (*i.e.*, $4\times$). The learning rate is decreased by a factor of 10 at iteration $300K$ and $340K$.

The resulting real-time models are shown in Table 9.

Method	FPS	AP	AP `test-dev`
YOLOv3 (Darknet-53) [31]	26	–	33.0
CenterNet (DLA-34) [52]	**52**	37.4	–
FCOS-RT (R-50)	38	40.2	40.2
FCOS-RT (DLA-34-BiFPN)	43	**42.1**	**42.2**
FCOS-RT (DLA-34)	46	40.3	40.3
FCOS-RT w/ shtw. (DLA-34)	**52**	39.1	39.2

TABLE 9
Real-time FCOS (FCOS-RT) models. AP is on COCO `val` split. "shtw.": sharing towers (*i.e.*, $4\times$ conv. layers shown in Fig. 2) between the classification and regression branches. The inference time is measured with a single 1080Ti or Titan XP GPU (these two GPUs' speeds are close). FPS of all FCOS models is measured on a 1080Ti GPU. We measure the FPS of YOLOv3 on 1080Ti using the code released by [31]. For FCOS and YOLOv3, we measure the end-to-end inference time here (*i.e.*, from prepossessing to the final output boxes).

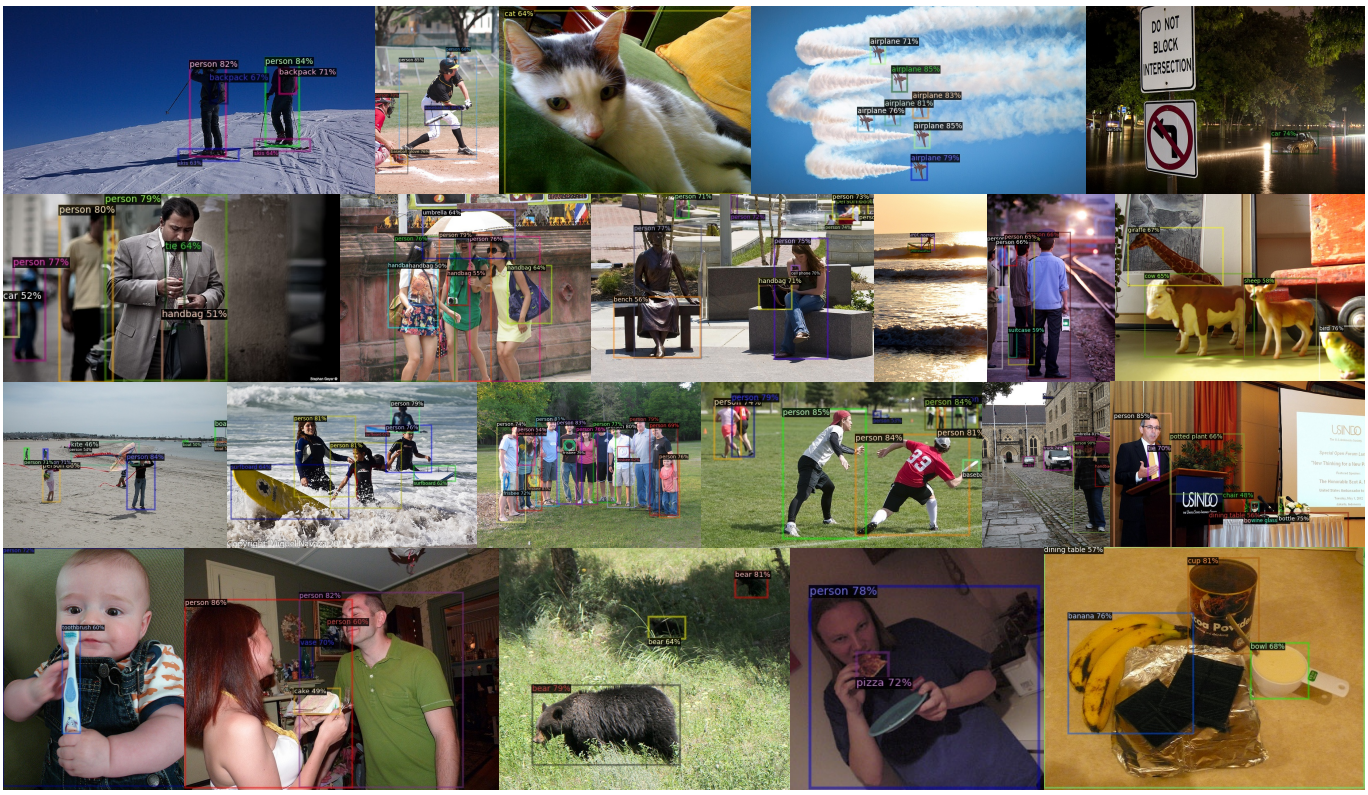

Fig. 6. Qualitative results. As shown in the figure, FCOS works well with a wide range of objects including crowded, occluded, extremely small and very large objects.

With ResNet-50, FCOS-RT can achieve 40.2% AP at 38 FPS on a single 1080Ti GPU card. We further replace ResNet-50 with the backbone DLA-34 [49], which results in a better speed/accuracy trade-off (40.3% AP at 46 FPS). In order to compare with CenterNet [52], we share the towers (*i.e.*, $4\times$ conv. layers shown in Fig. 2) between the classification and regression branches, which improves the speed from 46 FPS to 52 FPS but deteriorate the performance by 1.2% AP. However, as shown in Table 9, the model still outperforms CenterNet [52] by 1.7% AP at the same speed. For the real-time models, we also replace FPN with BiFPN as in Section 4.3, resulting 1.8% AP improvement (from 40.3% to 42.1%) at similar speed. A speed/accuracy comparison between FCOS and a few recent detection methods is shown in Fig. 3.

4.5 FCOS on CrowdHuman

We also conduct experiments on the highly crowded dataset CrowdHuman [34]. CrowdHuman consists of $15k$ images for training, $4,370$ for validation and $5,000$ images for testing. Following previous works on crowded benchmark [4], [34], we use AP, *long-average Miss Rate on False Positive Per Image in* $[10^{-2}, 100]$ (MR^{-2}) [8] and *Jaccard Index* (JI) as the evaluation metrics. Note that lower MR^{-2} is better. Following [4], all experiments here are trained on the `train` split for 30 epochs with batch size 16 and then evaluated on the `val` split. Two some changes are made when FCOS is applied to the benchmark. First, the NMS threshold is set as 0.5 instead of 0.6. We find that it has large impact on MR^{-2} and JI. Second, when a location is supposed to be associated to multiple ground-truth boxes, on COCO, we choose the object with minimal area as the target for the location. On CrowdHuman, we instead choose the target with minimal distance to the location. The distance between a location and an object is defined as the distance from the location to the center of the object. On COCO, both schemes result in similar performance. However, the latter has much better performance than the former on the highly crowded dataset. Other settings are the same as that of COCO.

First, we count the ambiguous sample ratios on CrowdHuman `val` set. With FPN-based FCOS, there are 84.47% unambiguous positive samples (with one ground-truth box), 13.63% with two ground-truth boxes, 1.69% with three ground-truth boxes and the rest ($< 0.3\%$) with more than three ground-truth boxes. Given the much higher ambigu-

	AP	MR^{-2}	JI
RetinaNet w/ imprv.	81.60	57.36	72.88
FCOS w/o ctr.-ness	83.16	59.04	73.09
FCOS w/ ctr.-ness	85.0	51.34	74.97
+ MIP ($K = 2$)	85.19	51.60	75.14
+ Set NMS [4]	**87.28**	**51.21**	**77.34**

TABLE 10
FCOS for crowded object detection on the CrowdHuman dataset. Even on the highly crowded benchmark, FCOS still attains even better performance than anchor-based RetinaNet. Note that lower MR^{-2} is better. "MIP w. set NMS": Multiple Instance Prediction, which predicts multiple instances from a single location as proposed by [4]. Note that we are not pursuing the state-of-the-art performance on the benchmark. We only show that the anchor boxes are not necessary even on the highly-crowded benchmark.

Fig. 7. Qualitative results on the CrowdHuman `val` set with the ResNet-50-FPN backbone.

ous sample ratio than COCO, it is expected that FCOS will have inferior performance on the highly crowded dataset.

We compare FCOS without center-ness with the the improved RetinaNet (*i.e.*, "RetinaNet w/ imprv."). To our surprise, even without center-ness, FCOS can already achieve decent performance. As shown in Table 10, FCOS compares favorably with its anchor-based counterpart RetinaNet on two out of three metrics (AP and JI), *which suggests that anchor-based detectors have no large advantages even under the highly crowded scenario.* The higher MR^{-2} of FCOS denotes that FCOS might have a large number of false positives with high confidence. By using the center-ness, MR^{-2} can be significantly reduced from 59.04% to 51.34%. As a result, FCOS can achieve better results under all the three metrics.

Furthermore, as shown in [4], it is more reasonable to let one proposal make multiple predictions under the highly crowded scenario (*i.e.*, multiple instance prediction (MIP)). After that, these predictions are merged by Set NMS [4], which skips the suppression for the boxes from the same location. A similar idea can be easily incorporated into FCOS. To be specific, if a location should be associated to multiple objects, instead of choosing a single target (*i.e.*, the closest one to the location), the location's targets are set as the K-closest objects. Accordingly, the network is required to make K predictions per location. Moreover, we do not make use of the earth mover's distance (EMD) loss for simplicity. Finally, the results are merged by Set NMS [4]. As shown in Table 10, with MIP and Set NMS, improved performance is achieved under all the three metrics.

5 Conclusion

In this work, we have proposed an anchor-free and proposal-free one-stage detector FCOS. Our experiments demonstrate that FCOS compares favourably against the widely-used 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 present some real-time models of our detector, which have state-of-the-art performance and inference speed. Given its effectiveness and efficiency, we hope that FCOS can serve as a strong and simple alternative of current mainstream anchor-based detectors.

Acknowledgments

This work was in part supported by ARC DP grant #DP200103797. We would like to thank the author of [1] for the tricks of center sampling and GIoU. We also thank Chaorui Deng for his suggestion of positioning the center-ness branch with box regression.

References

[1] https://github.com/yqyao/FCOS_PLUS, 2019.
[2] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. "BlendMask: Top-down meets bottom-up for instance segmentation,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.
[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] Xuangeng Chu, Anlin Zheng, Xiangyu Zhang, and Jian Sun. "Detection in crowded scenes: One proposal, multiple predictions,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.
[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable convolutional networks,". In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 764–773, 2017.
[6] 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.
[7] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. "Fast feature pyramids for object detection,". *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.
[8] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. "Pedestrian detection: An evaluation of the state of the art,". *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):743–761, 2011.

[9] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. "CenterNet: Keypoint triplets for object detection,". In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 6569–6578, 2019.

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

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

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

[13] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. "SiamCAR: Siamese fully convolutional classification and regression for visual tracking,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

[20] Youngwan Lee and Jongyoul Park. "CenterMask: Real-time anchor-free instance segmentation,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. "ABCNet: Real-time scene text spotting with adaptive Bezier-curve network,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

[27] Yifan Liu, Changyong Shun, Jingdong Wang, and Chunhua Shen. "Structured knowledge distillation for dense prediction,". *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[28] 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.

[29] 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.

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

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

[32] 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.

[33] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. "Generalized intersection over union: A metric and a loss for bounding box regression,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 658–666, 2019.

[34] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. "Crowdhuman: A benchmark for detecting human in a crowd,". *arXiv preprint arXiv:1805.00123*, 2018.

[35] Chunhua Shen, Peng Wang, Sakrapee Paisitkriangkrai, and Anton van den Hengel. "Training effective node classifiers for cascade classification,". *Int. J. Comp. Vis.*, 103(3):326–347, 2013.

[36] 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.

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

[38] Mingxing Tan, Ruoming Pang, and Quoc Le. "EfficientDet: Scalable and efficient object detection,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

[39] Zhi Tian, Hao Chen, and Chunhua Shen. "DirectPose: Direct end-to-end multi-person pose estimation,". *arXiv preprint arXiv:1911.07451*, 2019.

[40] 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.

[41] Zhi Tian, Chunhua Shen, and Hao Chen. "Conditional convolutions for instance segmentation,". *arXiv preprint arXiv:2003.05664*, 2020.

[42] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. "FCOS: Fully convolutional one-stage object detection,". In *Proc. IEEE Int. Conf. Comp. Vis.*, 2019.

[43] Paul Viola and Michael Jones. "Robust real-time object detection,". 2001.

[44] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. "Tracking by instance detection: A meta-learning approach,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

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

[46] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. "Detectron2,". https://github.com/facebookresearch/detectron2, 2019.

[47] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. "PolarMask: Single shot instance segmentation with polar representation,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020. arxiv.org/abs/1909.13226.

[48] 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.

[49] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. "Deep layer aggregation,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2403–2412, 2018.

[50] 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.

[51] Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. "Mask encoding for single shot instance segmentation,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020.

[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as points,". In *arXiv preprint arXiv:1904.07850*, 2019.

[53] 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.

[54] 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.

[55] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. "Deformable convnets v2: More deformable, better results,". In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9308–9316, 2019.

Zhi Tian is a PhD student at School of Computer Science, The University of Adelaide, Australia. He received his BSc degree in Computer Science from Sichuan University, China. He was awarded a Google PhD fellowship in 2019.

Chunhua Shen is a Professor at School of Computer Science, The University of Adelaide, Australia.

Hao Chen is a PhD student at School of Computer Science, The University of Adelaide, Australia. He received his BSc and master degrees from Zhejiang University, China.

Tong He is a PhD student at School of Computer Science, The University of Adelaide, Australia. He received his master degree from Wuhan University, China; and BSc degree from Tianjin University of Science and Technology, China.