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 pre-defined set of anchor boxes, FCOS completely avoids the complicated computation related to anchor boxes such as calculating overlapping during training and significantly reduces the training memory footprint. 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), our detector FCOS outperforms previous anchor-based 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 [20], SSD [15] and YOLOv2, v3 [19] 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 [12, 20], detection performance is sensitive to the sizes, aspect ratios and number of anchor boxes. For example, in RetinaNet [12], varying these hyper-parameters affects the performance up to 4% in AP on the COCO benchmark [13]. 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) [11] 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. An excessively large number of anchor boxes also significantly increase the amount of computation and

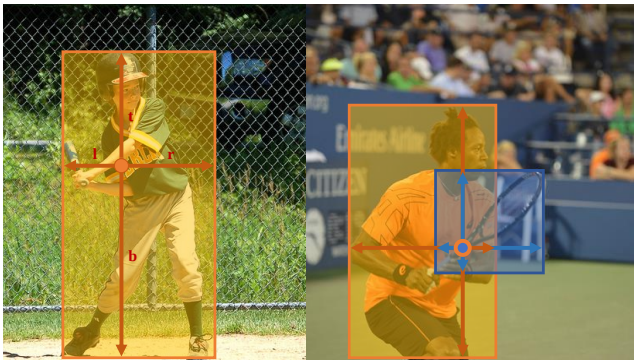

Figure 1 – As shown in the left image, FCOS works by predicting a 4D vector (l, t, r, b) encoding the location of a bounding box at each foreground pixel (supervised by ground-truth bounding box information during training). The right plot shows that when a location residing in multiple bounding boxes,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

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

memory footprint when computing the intersection-over-union (IOU) scores between all anchor boxes and ground-truth boxes during training.

Recently, fully convolutional networks (FCNs) [16] have achieved tremendous success in dense prediction tasks such as semantic segmentation [16], depth estimation [14], key-point detection [2], and counting [1]. As one of high-level vision tasks, object 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 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 simper 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 [9] and UnitBox [24]. 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 [9] 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. Beside, more significantly, these methods are mainly used in special domain objection detection such as scene text detection [25] or face detection [24, 9], 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 during training: 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 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, particular its training, *considerably* simpler. Moreover, by eliminating the anchor boxes, our new detector completely avoids the complex IOU computation and matching between anchor boxes and ground-truth boxes during training and reduces the total training memory footprint by a factor of 2 or so.
- 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 [5]. 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 convolutional neural networks (CNNs) and avoid repeated feature computation, speeding up detection

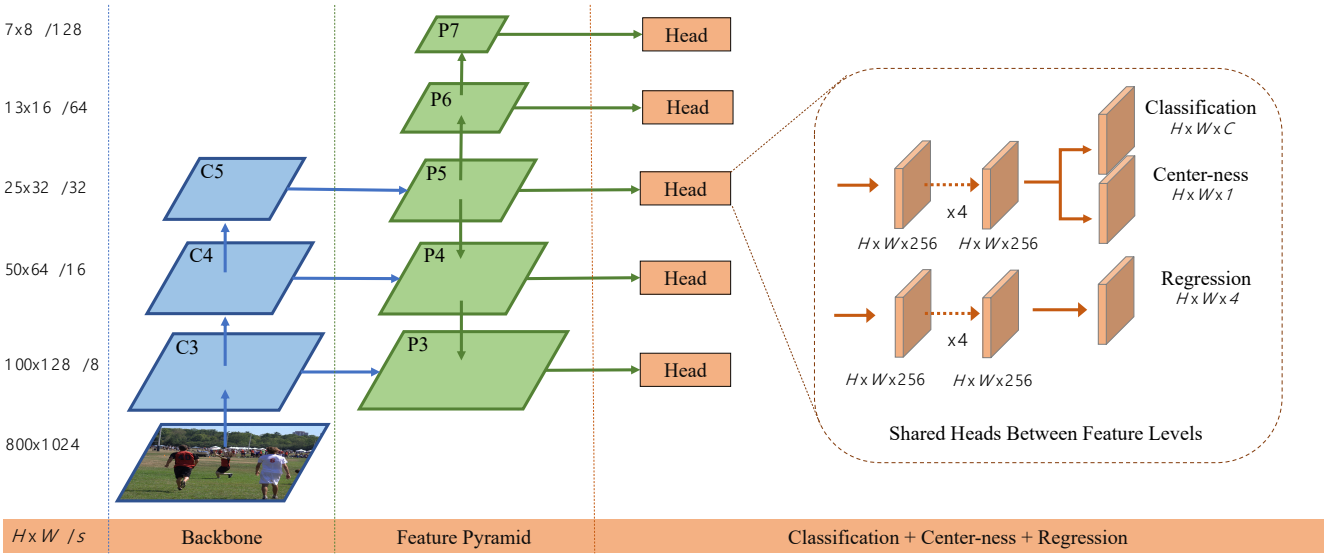

Figure 2 – The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. '$/s$' ($s = 8, 16, ..., 128$) is the down-sampling ratio of the level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

process dramatically. The design of anchor boxes are popularized by Faster R-CNN in its RPNs [20], SSD [15] and YOLOv2 [18], 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 hyper-parameters of anchor shapes described above, 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 employ intersection over union (IOU) between anchor boxes and ground-truth boxes to label them (*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 as used in other dense prediction tasks such as semantic segmentation.

Anchor-free Detectors. The most popular anchor-free detector might be YOLOv1 [17]. 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 [18]. As a result, YOLOv2 [18] makes use of 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 [10] 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 [24] are based on DenseBox [9]. The family of detectors have been considered unsuitable for generic object detection due to difficulty in handling overlapping bounding boxes and the recall being 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 in training. Finally, we present our proposed "center-

ness" branch, which helps suppress the low-quality detected bounding boxes and improve 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 before 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 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 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 anchor boxes and regress the target bounding box for these anchor boxes, we directly regress the target bounding box for each 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 in FCNs for semantic segmentation [16].

Specifically, location (x, y) is considered as a positive sample if it falls into any ground-truth bounding box and the class label c^* of the location is the class label of B_i. 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 target for each sample. 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*. For now, 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. 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 [12], instead of training a multi-class classifier, we train C binary classifiers. Similar to [12], 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 [12, 20] 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 the focal loss as in [12] and L_{reg} is the IOU loss as in UnitBox [24]. 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 [12], 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 [11]. 1) The large stride (*e.g.*, $16\times$) 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 for 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

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

it can even better than the BPR of the anchor-based detector RetinaNet [12] in the official implementation Detectron [6] (refer to Table 1). Therefore, the BPR is actually not a problem of FCOS. Moreover, with multi-level FPN prediction [11], 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 during training, *i.e.*, w.r.t. which bounding box should a location in the overlap to 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 [11], we detect different sizes of objects on different levels of feature maps. Specifically, we make use of five levels of feature maps defined as $\{P_3, P_4, P_5, P_6, P_7\}$. 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 lateral connections in [11], 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. 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, the multi-level prediction can largely alleviate the aforementioned ambiguity and improve the FCN-based detector to the same level of anchor-based ones, as shown in our experiments.

Finally, following [11, 12], 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 empirically improves the detection performance.

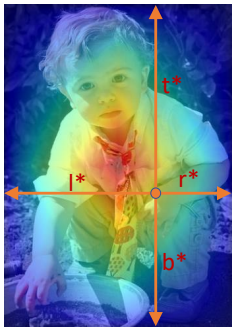

Figure 3 – Center-ness. Red, blue, and other colors denote 1, 0 and the values between them, respectively. Center-ness is computed by Eq. (3) and decays from 1 to 0 as the location deviates from the center of the object. When testing, the center-ness predicted by the network is multiplied with the classification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

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 to predict the "center-ness" of a location (*i.e.*, the distance from the location to the center of the object that the location is responsible for), as shown in Fig. 2. 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 (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*.

From the perspective of anchor-based detectors, which use two IOU thresholds T_{low} and T_{high} to label the anchor

boxes as negative, ignored and positive samples, the centerness can be viewed as a soft threshold. It is learned during the training of networks and does not need to be tuned. Moreover, with the strategy, our detector can still view any locations falling into a ground box as positive samples, except for the ones set as negative samples in the aforementioned multi-level prediction, so as to use as many training samples as possible for the regressor.

4. Experiments

Our experiments are conducted on the large-scale detection benchmark COCO [13]. Following the common practice [12, 11, 20], 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 [7] is used as our backbone networks and the same hyper-parameters with RetinaNet [12] 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 [3]. For the newly added layers, we initialize them as in [12]. 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. The following post-processing is exactly the same with RetinaNet [12] and we directly make use of the same post-processing hyper-parameters (such as the threshold of NMS) of RetinaNet. We argue that the performance of our detector can be improved further if the hyper-parameters are optimized for it. We use the same sizes of input images as in training.

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 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 in precision-recall curve (shown in supplementary material) of these detectors are much lower than 90%, the small BPR gap between FCOS and the anchor-based detector will not actually affect the performance of detector. It is also confirmed in Table 4, where FCOS achieves even better AR than its anchor-based counterparts. Therefore, the concern about low BPR may not be necessary.

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 [6], 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%).

w/ FPN	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
	18.5	34.2	18.0	5.5	20.7	30.6
✓	**33.8**	**54.5**	**35.0**	**20.6**	**38.1**	**43.2**

Table 3 – With or without multi-level prediction in FCOS. The multi-level prediction almost doubles the accuracy in AP.

Ambiguous Samples. Another concern about the FCN-based detector is that it may have a large number of *ambigu-*

Method	C_5/P_5	# samples	Mem (GB)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}
RetinaNet	C_5	∼180K	54.4	35.7	54.7	38.5	19.5	39.9	47.5	30.7	49.1	52.0
RetinaNet w/ GN	C_5	∼180K	54.8	36.1	**56.3**	**38.9**	20.6	**40.5**	47.5	31.1	49.4	**52.5**
FCOS	C_5	∼20K	29.5	36.4	55.7	38.7	20.7	40.2	**48.0**	**31.6**	**50.1**	**52.5**
FCOS	P_5	∼20K	**29.3**	**36.6**	56.0	**38.9**	**20.9**	40.3	47.2	31.5	50.0	52.4

Table 4 – FCOS vs. RetinaNet on the `minival` split. Our FCN-based FCOS with $9\times$ fewer samples and $\sim 2\times$ less training memory footprint achieves even better performance than anchor-based RetinaNet both in AP and AR, directly using the exactly same training and testing settings from RetinaNet. The results of RetinaNet are obtained with the models provided in the official Detectron. RetinaNet with GN (Group Normalization) is trained by ourselves with the provided official code in Detectron.

ous 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 the feature level P_4 is used. However, if we use all feature levels, 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 when inferring since the bounding box predicted these samples can always be matched with a correct category regardless of w.r.t. which object the sample regresses. 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%. As shown in the following experiments, the extremely low overlapping ratio does not make our FCOS inferior to anchor-based detectors.

Detection Performance. So far we have shown that the BPR of FCOS is enough and the multi-level prediction can not only improve the BPR but also significantly reduce the ambiguity during training in terms of w.r.t. which bounding box to regress. As shown in Table 3, with the help of multi-level prediction, the FCN-based FCOS can already achieve the same level performance as the anchor-based RetinaNet with the multi-level prediction (33.8% vs. 35.7%). Compared the one with only one feature level P_4, the AP is almost doubled.

4.1.2 With or Without Center-ness

We have shown that the FCN-based FCOS is able to achieve a comparable performance with the anchor-based detector RetinaNet. However, there is still a performance gap $\sim 2\%$ in AP. We believe that this gap may be due to the fact that there are a few low-quality detected bounding boxes produced by the locations far from the center of an object. It is easy to see that a location closer to the center is more likely to produce more accurate predictions. Thus the detections produced by far-away locations should be assigned a low confidence score. To this end we make use of a center-ness branch to suppress the low-quality detected bounding boxes. As shown in Table 5, the center-ness branch can boost AP from 33.8% to 36.6%, which outperforms the performance of the anchor-based detector (35.7%). 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 5, the center-ness computed from the regression vector cannot improve the performance and thus the separately learned center-ness is necessary.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
None	33.8	54.5	35.0	20.6	38.1	43.2
center-ness†	33.1	53.1	34.3	20.4	36.9	42.4
center-ness	**36.6**	**56.0**	**38.9**	**20.9**	**40.3**	**47.2**

Table 5 – 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.

To further demonstrate the usefulness of center-ness, we carry out one more experiment. We assume that we had an oracle which provides the ground-truth center-ness score during inference. With keeping all the other settings exactly the same, *ground-truth center-ness for inference significantly improves the AP to* 42.1, meaning that there is much room for further improving our current accuracy of 36.6 AP as shown in Table 5, as long as we improve the prediction accuracy of the center-ness.

As a result, we make the center-ness branch deeper,[2]

[2]Recall that we have used only one layer for the center-ness branch.

Method	C_5/P_5	AP	AP_{50}	AP_{75}
RetinaNet	C_5	33.7	52.9	36.2
RetinaNet w/ GN	C_5	34.2	**54.4**	36.5
FCOS	C_5	34.4	53.9	36.1
FCOS	P_5	**34.7**	54.3	**36.6**

Table 6 – FCOS vs. RetinaNet on `minival_overlapped` split, which only consists of images containing overlapped bounding boxes. FCOS still achieves even better performance than the anchor-based counterpart.

having the same architecture as the classification and regression branches, which improves the AP from 36.6 to 36.8.

In theory we may even train a separate deep network, which does not share any weight with the main detector, with its only purpose being to predict the center-ness score. This is only possible due to the fact that the center-ness score is solely used in inference. Therefore we are able to decouple the training of the center-ness predictor from the training of the detector. This decouple allows us to design the best possible center-ness predictor with the price of extra computation complexity. We also hypothesize that all other detectors, if NMS is needed for post-processing, may be able to benefit from such an accurate center-ness score predictor. We leave this topic for future work.

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) [23] 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 add GN into RetinaNet and use C_5 in our detector as well. As shown in Table 4, with exactly the same settings, our FCOS still compares favorably against the anchor-based detector. Since our FCN-based detector has a lot of merits (*e.g.*, much less design complexity and using only half of memory footprint for training as shown in Table 4) than the anchor-based ones, we encourage the community to rethink the necessity of anchor boxes in object detection. 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.

One may still worry that the overlapping in bounding boxes would result in a degraded performance. In order to further show that the overlapping is not a problem of FCOS, we construct a subset of `minival`, named `minival_overlapped`. It consists of 3986 images, each of which includes at least one overlapped bounding box. The subset contains 35, 058 bounding boxes in total, in which 30, 625 bounding boxes (up to 87%) overlap with other bounding boxes. On the subset, our FCOS still achieves even better performance than the anchor-based RetinaNet, which suggests that FCOS can work well with overlapped bounding boxes.

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

In the ablation study, in order to make a fair comparison with anchor-based counterparts and demonstrate that our framework can serve as a strong and simple alternative of anchor-based detectors, we directly make use of all hyper-parameters of RetinaNet. We argue that the performance can be much improved if the hyper-parameters are tuned for our detector. For our main results on $\texttt{test} - \texttt{dev}$ split, we make use of scale jitter as in RetinaNet during the training and double the number of iterations. Other settings are exactly the same as in ablation study. As shown in Table 7, with ResNet-101-FPN and ResNet-32x8d-101-FPN as the backbone, our FCOS outperforms RetinaNet with the same backbone by 1.9% and 1.3% in AP, respectively. 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. FCOS also outperforms other classical two-stage anchor-based detectors such as Faster R-CNN by a large margin.

Compared to the recent state-of-the-art one-stage detector CornerNet [10], our FCOS also has 0.5% gain in AP. Maybe the gain is relatively small, but our detector enjoys the following advantages over CornerNet. 1) We achieve the performance with a faster and simpler backbone ResNet-101 instead of Hourglass-104 in CornerNet. 2) Except for the standard post-processing NMS in the detection task, our detector does not need any other post-processing. In contrast, CornerNet requires grouping pairs of corners with embedding vectors, which needs special design for the detector. 3) Compared to CornerNet, we argue that our FCOS is more likely to serve as a strong and simple alternative for current mainstream anchor-based detectors.

5. Extensions on Region Proposal Networks

Thus far, we have shown that in 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 [11] in the two-stage detector Faster R-CNN. In the section, we confirm that by experiments.

Compared to RPNs with FPN [11], 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

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods:							
Faster R-CNN+++ [7]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w/ FPN [11]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [8]	Inception-ResNet-v2 [22]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [21]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [18]	DarkNet-19 [18]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [15]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [4]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [12]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [12]	ResNeXt-32x8d-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
CornerNet [10]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	**53.9**
FCOS (ours)	ResNet-101-FPN	41.0	60.7	44.1	24.0	44.1	51.0
FCOS (ours)	ResNeXt-32x8d-101-FPN	**42.1**	**62.1**	**45.2**	**25.6**	**44.9**	52.0

Table 7 – 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 1.9% and 1.3% in AP with ResNet-101-FPN and ResNeXt-32x8d-101-FPN, respectively. FCOS also outperforms the recent anchor-free one-stage detector CornerNet with much less design complexity.

Method	# samples	AR^{100}	AR^{1k}
RPN w/ FPN (Detectron)	$\sim$200K	43.5	57.3
RPN w/ FPN (ReImpl.)	$\sim$200K	43.3	56.9
RPN w/ FPN & GN (ReImpl.)	$\sim$200K	44.7	56.9
FCOS w/o center-ness	$\sim$66K	48.0	59.3
FCOS	$\sim$66K	**52.8**	**60.3**

Table 8 – FCOS as Region Proposal Networks vs. RPNs with FPN. ResNet-50 is used as the backbone. FCOS improves AR^{100} and AR^{1k} by more than 9% and 3%, respectively. GN: Group Normalization.

with FPN in the official code [6]. As shown in Table 8, even without 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 21% relative improvement for AR^{100} and 3% absolute improvement for AR^{1k} over the official 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.

Appendix

7. Class-agnostic Precision-recall Curves

Method	AP	AP_{50}	AP_{75}	AP_{90}
Orginal RetinaNet [12]	39.5	63.6	41.8	10.6
RetinaNet w/ GN [23]	40.0	64.5	42.2	10.4
FCOS	**40.5**	**64.7**	**42.6**	**13.1**
		+0.2	+0.4	+2.7

Table 9 – The class-agnostic detection performance for RetinaNet and FCOS. FCOS has better performance than RetinaNet. Moreover, the improvement over RetinaNet becomes larger with a stricter IOU threshold. The results are obtained with the same models in Table 4 of our main paper.

In Fig. 4, Fig. 5 and Fig. 6, we present class-agnostic precision-recall curves on split `minival` at IOU thresholds being 0.50, 0.75 and 0.90, respectively. Table 9 shows APs corresponding to the three curves.

As shown in Table 9, our FCOS achieves better performance than its anchor-based counterpart RetinaNet. Moreover, it worth noting that with a stricter IOU threshold, FCOS enjoys a larger improvement over RetinaNet, which suggests that FCOS has a better bounding box regressor to detect objects more accurately. One of the reasons should be that FCOS has the ability to leverage more foreground samples to train the regressor as mentioned in our main paper.

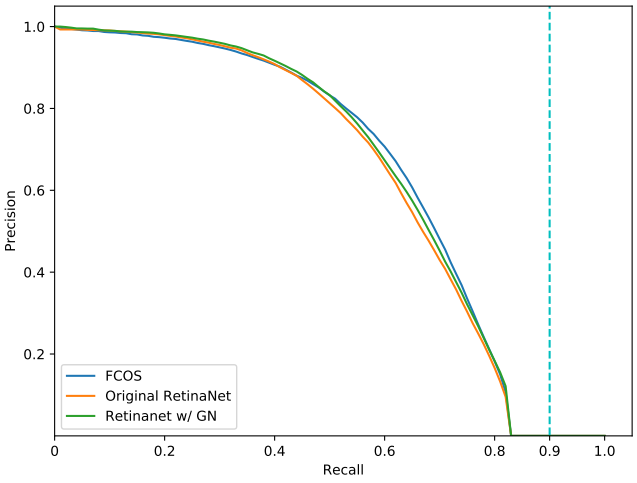

Figure 4 – Class-agnostic precision-recall curves at IOU $= 0.50$.

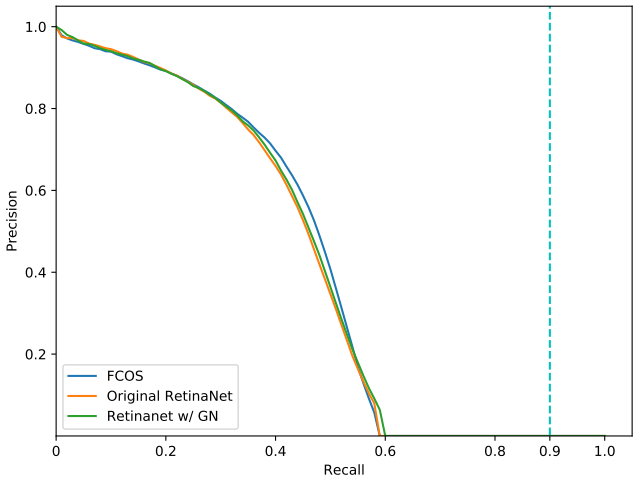

Figure 5 – Class-agnostic precision-recall curves at IOU $= 0.75$.

Finally, as shown in all precision-recall curves, the best recalls of these detectors in the precision-recall curves are much lower than 90%. It further suggests that the small gap (98.40% vs. 99.23%) of *best possible recall (BPR)* between FCOS and RetinaNet hardly harms the final detection performance.

8. Visualization for Center-ness

As mentioned in our main paper, by suppressing low-quality detected bounding boxes, the proposed center-ness branch improves the detection performance by a large margin. In this section, we confirm this.

We expect that the center-ness can down-weight the scores of low-quality bounding boxes such that these bounding boxes can be filtered out in following post-processing such as non-maximum suppression (NMS). A detected bounding box is considered as a low-quality one if it has a low IOU score with its corresponding ground-truth bounding box. A bounding box with low IOU but a high confidence score is likely to become a false positive and harm the precision.

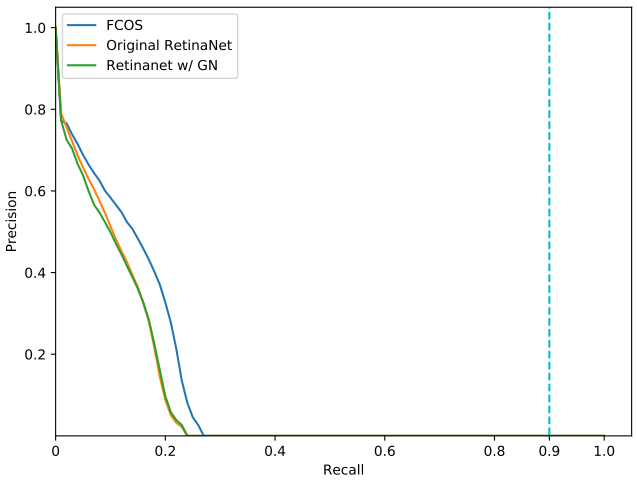

Figure 6 – Class-agnostic precision-recall curves at IOU $= 0.90$.

In Fig. 7, we consider a detected bounding box as a 2D point (x, y) with x being its score and y being the IOU with its corresponding ground-truth box. As shown in Fig. 7 (left), before applying the center-ness, there are a large number of low-quality bounding boxes but with a high confidence score (i.e., the points under the line $y = x$). Due to their high scores, these low-quality bounding boxes cannot be eliminated in post-processing and result in lowering the precision of the detector. After multiplying the classification score with the center-ness score, these points are pushed to the left side of the plot (i.e., their scores are reduced), as shown in Fig. 7 (right). As a result, these low-quality bounding boxes are much more likely to be filtered out in post-processing and the final detection performance can be improved.

9. Qualitative Results

Some qualitative results are shown in Fig. 8. As shown in the figure, our proposed FCOS can detect a wide range of objects including crowded, occluded, highly overlapped, extremely small and very large objects.

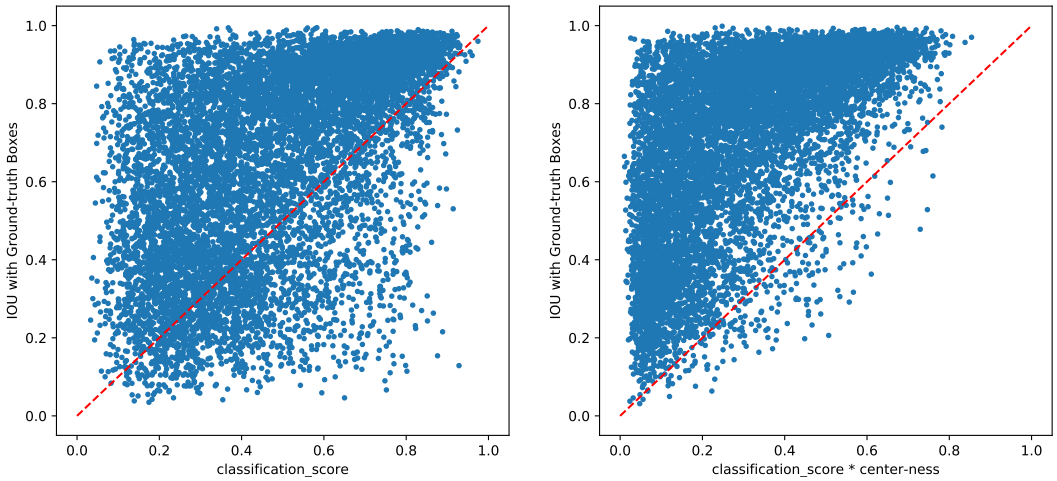

Figure 7 – Without (left) or with (right) the proposed center-ness. A point in the figure denotes a detected bounding box. The dashed line is the line $y = x$. As shown in the figure (right), after multiplying the classification scores with the center-ness scores, the low-quality boxes (under the line $y = x$) are pushed to the left side of the plot. It suggests that the scores of these boxes are reduced substantially.

References

[1] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proc. ACM Int. Conf. Multimedia*, pages 640–644. ACM, 2016.

[2] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. Adversarial PoseNet: A structure-aware convolutional network for human pose estimation. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.

[4] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. Berg. DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

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

[6] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7310–7311, 2017.

[9] L. Huang, Y. Yang, Y. Deng, and Y. Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

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

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2117–2125, 2017.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2980–2988, 2017.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755. Springer, 2014.

[14] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proc. Eur. Conf. Comp. Vis.*, pages 21–37. Springer, 2016.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015.

Figure 8 – Some detection results on `minival` split. ResNet-50 is used as the backbone. As shown in the figure, FCOS works well with a wide range of objects including crowded, occluded, highly overlapped, extremely small and very large objects.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 779–788, 2016.

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

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

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

[21] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. National Conf. Artificial Intell.*, 2017.

[23] Y. Wu and K. He. Group normalization. In *Proc. Eur. Conf. Comp. Vis.*, pages 3–19, 2018.

[24] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proc. ACM Int. Conf. Multimedia*, pages 516–520. ACM, 2016.

[25] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5551–5560, 2017.