POLARNET: LEARNING TO OPTIMIZE POLAR KEYPOINTS FOR KEYPOINT BASED OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

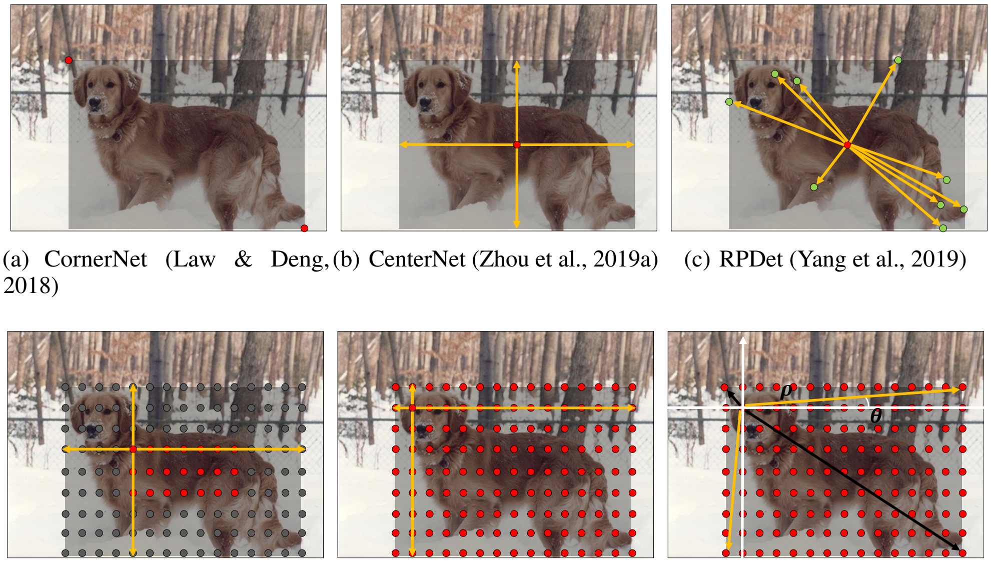
A variety of anchor-free object detectors have been actively proposed as possible alternatives to the mainstream anchor-based detectors that often rely on complicated design of anchor boxes. Despite achieving promising performance at par with anchor-based detectors, the existing anchor-free detectors such as FCOS or CenterNet predict objects based on standard Cartesian coordinates, which often yield poor quality keypoints. Further, the bounding box regression is also scale-sensitive. In this paper, we propose a new anchor-free keypoint based detector "PolarNet", where keypoints are represented as a set of Polar coordinates instead of Cartesian coordinates. The "PolarNet" detector learns offsets pointing to the corners of objects in order to learn high quality keypoints. Additionally, PolarNet uses corner points to localize objects, making the localization scale-insensitive. Finally in our experiments, we show that PolarNet, an anchor-free detector, outperforms the existing anchor-free detectors, and is able to achieve highly competitive result on COCO test-dev benchmark (48.0% AP under the single-model single-scale testing) which is at par with the state-of-the-art two-stage anchor-based object detectors.

1 INTRODUCTION

Deep learning based object detection techniques have achieved remarkable success in many real-world applications (Krizhevsky et al., 2012; He et al., 2016; Goodfellow et al., 2016). The mainstream state-of-the-art detectors are often based on the anchor-based detection methods (Ren et al., 2015; Redmon & Farhadi, 2016; Girshick, 2015; Lin et al., 2017b), which heavily rely on the design and selection of appropriate *anchor boxes*, namely a set of predefined bounding boxes of a certain height and width, to capture various scales and aspect ratios of different object classes for detection. Unlike the anchor-based detectors, the anchor-free detectors have emerged recently as a promising direction for object detection that eliminates the need of manually designing anchor boxes (Zhu et al., 2019; Tian et al., 2019; Law & Deng, 2018; Duan et al., 2019).

In literature, a variety of anchor-free object detectors have been proposed based on different object modeling strategies. Figure 1 (a)-(e) gives examples comparing five popular anchor-free detectors from the perspective of object modeling. For example, CornerNet (Law & Deng, 2018) was proposed for detecting objects using a pair of corner points. Instead of using two corners, CenterNet (Zhou et al., 2019a) proposed modeling an object as one center point of its bounding box. Besides these, there are also a number of other anchor-free detectors that extend these ideas of Corner-based or Centerness-based or various other keypoint design strategies to improve the detection performance. FSAF (Zhu et al., 2019) and FCOS (Tian et al., 2019) predict objects by learning the offsets to the boundary from sampled keypoints. FCOS (Tian et al., 2019) uses many keypoints by treating every pixel as a keypoint, while FSAF (Zhu et al., 2019) samples a set of multiple keypoints from the center region to eliminate points near the boundary.

Among keypoint based object detection, two different strategies are commonly adopted. One is keypoint position based (determining the bounding box by the position of the keypoints) such CornerNet (Law & Deng, 2018) and CenterNet (Duan et al., 2019). The other is keypoint offsets based(determining the bounding box by learning offsets from keypoints) such as FSAF (Zhu et al., 2019), CenterNet (Zhou et al., 2019a) FCOS (Tian et al., 2019). Compared to the keypoint position based methods, keypoint offset based methods are able to better capture contextual information about the objects, and are thus more commonly used to design detectors.



(a) CornerNet (Law & Deng, 2018)
(b) CenterNet (Zhou et al., 2019a)
(c) RPDet (Yang et al., 2019)

(d) FSAF (Zhu et al., 2019)
(e) FCOS (Tian et al., 2019)
(f) PolarNet (ours)

Figure 1: Comparison of different anchor-free object detection methods. Red dots denote positive keypoints and grey dots denote negative keypoints.

However, existing object modeling strategies for keypoint offset based methods may be sub-optimal. Most existing anchor-free detectors such as FCOS (Tian et al., 2019) are based on Cartesian coordinates, which learn offsets to the boundary of objects. However, this kind of design yields a lot of poor quality keypoints. These points are near the boundary with extremely large variance of offsets (See Figure 1(e)). Besides, the bounding box prediction is based on the prediction of these offsets, which is scale-sensitive. Our goal is to have an anchor-free detector, that can avoid poor quality keypoints, and is able to simultaneously learn scale-insensitive feature for bounding box regression. To achieve this goal, in this paper, we propose a new keypoint based object detector named "PolarNet", which learns keypoints based on polar coordinates. Figure 1 (f) illustrates the idea of the proposed PolarNet compared to the other anchor-free detectors. The set of keypoints is represented by polar coordinates to avoid large variance of offsets. And the corner points in PolarNet are also used to localize objects, which is scale-invariant.

The key contributions of this work are:

- We introduce a unified view of keypoint based object detection for understanding popular anchor-free object detectors, in which many popular anchor-free object detectors can be viewed as a special case of keypoint based detectors with different object modeling strategies;
- We propose a new anchor-free object detector named "PolarNet" which present keypoints based on polar coordinates which enables learning of better quality keypoints by reducing the variance of learned offsets compared to the existing approaches.
- We conduct experiments to evaluate the performance of our PolarNet detector on the COCO benchmark, in which our results show that PolarNet outperforms all the existing anchor-free detectors, and is able to achieve highly competitive results better or on par with the state-of-the-art two-stage anchor-based detectors on COCO test-dev (48.0% AP with DCNv2-ResNeXt-101 backbone on COCO test-dev under single-model single-scale settings). Our source code and models will be released publicly upon acceptance.

The rest of this paper is organized as follows. Section 2 reviews two major categories of related work in deep-learning based object detection: popular anchor-based detectors and recent anchor-free detectors. Section 3 presents the proposed PolarNet detector in detail. Section 4 discusses our experimental results and analysis, and Section 5 concludes this paper.

2 Related Work

In this section, we briefly review two major groups of related work in the literature of deep-learning based object detection approaches: the mainstream family of anchor-based object detection methods and the emerging family of anchor-free object detection methods. We refer readers to more extensive surveys of deep-learning based object detection studies in (Liu et al., 2020; Wu et al., 2020).

2.1 Anchor-based Object Detection

The methods in this group represent the mainstream detectors widely used in many real-world applications. They can be broadly categorized into two groups: two-stage detectors (Ren et al., 2015; Girshick et al., 2014; Girshick, 2015) and one-stage detectors (Liu et al., 2016; Zhang et al., 2018; Lin et al., 2017b). Two-stage detectors consist of two stages: (i) region proposal generation, and (ii) region proposal classification and regression. For example, one of most popoular two-stage detectors is Faster R-CNN (Ren et al., 2015) that uses a Region Proposal Network (RPN) to generate regions of interest in the first stage and then sends the region proposals down the pipeline for object classification and bounding-box regression. Faster R-CNN has resulted in many various extensions and improvements in literature (Lin et al., 2017a; Kong et al., 2016; Zhu et al., 2017; Dai et al., 2017; 2016). Single-stage detectors, also known as single-shot detectors, do not need the proposal generation and simply treat object detection as a simple classification and regression problem by taking an input image and directly learning the class probabilities and bounding box coordinates of objects via convolutional networks. Popular single-stage detectors include YOLO (Redmon et al., 2016), SSD (Liu et al., 2016), and RetinaNet (Lin et al., 2017b). Typically, single-stage detectors are less accurate than two-stage detectors, but are much faster and thus more amenable to real-time inference needs. In general, anchor-based detectors suffer from some critical limitations, including requiring heuristic design of anchors, poor alignment of anchors with ground truth objects, and incurring a large number of false positives when anchors are not carefully designed.

2.2 Anchor-free Object Detection

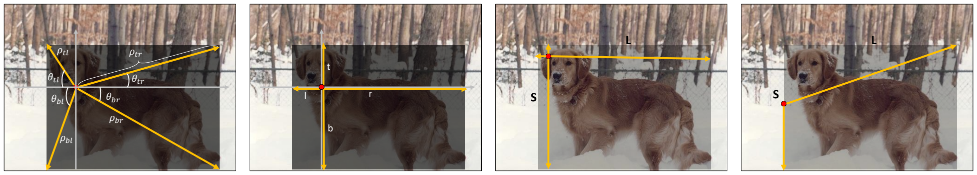
Here, we view most anchor-free detectors from as keypoint based detectors. We can categorize the existing anchor-free object detection methods into two groups according to different bounding box prediction strategies: 1) Keypoint Position and 2) Keypoint Offsets based detectors. We review some representative works in each group below.

Keypoint Position: This kind of anchor-free detectors predict bounding box via keypoint positions, such as corners. A representative detector is CornerNet (Law & Deng, 2018). CornerNet models each object by a pair of corner keypoints, which eliminates the need of anchor boxes and is perhaps the first anchor-free detector that achieved the state-of-the-art single-stage object detection accuracy. There have been some extensions of CornerNet to improve its efficiency towards real-time applications such as CornerNet-Lite (Law et al., 2019). Based on CornerNet, CenterNet (Duan et al., 2019), models an object by a triplet of keypoints, including one center and two corner keypoints. ExtremeNet (Zhou et al., 2019b) models an object by a set of five keypoints, including one center and four extreme points ((top-most, leftmost, bottom-most, right-most) of an object based on a standard keypoint estimation network. RPDet (Yang et al., 2019) represents an object by a fixed set of 9 keypoints learned from the center of an object which are refined progressively during the training process.

Keypoint Offsets: This kind of anchor-free detectors predict bounding box via learning offsets based on keypoints. A representative anchor-free detector is CenterNet (Zhou et al., 2019a) that models an object by the center point of its bounding box, and uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and even pose. Some approaches such as FSAS (Zhu et al., 2019) and FoveaBox (Kong et al., 2019) sample many keypoints from the center region of an object and learn to predict objects. Some approaches such as FCOS (Tian et al., 2019) and SAPD (Zhu et al., 2019) treat all the pixels within an object as candidate keypoints during training and improve them by a separate post-processing strategy. These keypoint based detectors all learn offsets to the boundary of the objects based on Cartesian coordinates, which yields a lot of poor quality keypoints. Some strategies are adopted to remove these bad points such as center sampling (Kong et al., 2019; Zhu et al., 2019) or centerness score (Tian et al., 2019). Unlike the above keypoint based detectors that learn keypoints based on Cartesian coordinates, our approach learns a set of keypoints based on polar coordinates which makes it easier to learn high quality keypoints.

3 PolarNet Detector

3.1 Polar Coordinates vs Cartesian Coordinates



(a) Polar (b) Cartesian (c) FCOS worst case (d) PolarNet worst case

Figure 2: (a) and (b) show the comparison of Polar coordinates and Cartesian coordinates. (c) and (d) show the worst case in training FCOS and PolarNet. In (c), top offset is close to 0 but right offset is close to L (and same for bottom and left offsets), indicating the large variance of learned pair of offsets if the points lie near the boundary. In (d), there are two pairs of offsets pointing to different corner pairs, and at least one pair is guaranteed to have low variance.

Many popular keypoint based detectors represent keypoints based on Cartesian coordinates. This strategy yields a lot of keypoints with poor quality (see Figure 1(e)), making it difficult to optimize without manually designed strategies (such as centerness (Tian et al., 2019) or center sampling (Zhu et al., 2019)). A better approach for keypoints representation is thus required to avoid poor quality keypoints (such as points near the boundary). In order to overcome this challenge, we represent the keypoints with polar coordinates instead of vanilla Cartesian coordinates. We learn the offsets of each keypoints pointing to the corners, and select the optimal corner pair for detector training. Furthermore, the corners are used for scale-insensitive localization.

For each keypoint, we have to predict offsets based on this point to generate the bounding box (bbx). Vanilla keypoint based detectors learn keypoints based on Cartesian coordinates, which learns 4D vectors (t, l, b, r), corresponding to (top,bottom, left, right) pointing to the boundary of the objects (See Figure 2 (b)). In PolarNet, for each object, we have two pairs of corners(top left(tl), bottom right(br) and top right(tr), bottom left(bl)). And thus for each point (x, y), we predict two pair of 4D vectors: $(\theta_{\text{tr}}, \rho_{\text{tr}}, \theta_{\text{bl}}, \rho_{\text{bl}})$ and $(\theta_{\text{tl}}, \rho_{\text{tl}}, \theta_{\text{br}}, \rho_{\text{br}})$, which point to the four corners of the object (See Figure 2(a)). Here ρ and θ denote the distance and angle of point (x, y) to corner (corner $\in \{$br, tl, bl, tr$\}$ and $\theta \in (0, \frac{\pi}{2})$). We defer the discussion for training the keypoint predictor later.

For keypoint offsets based detectors, learning offsets with large variance yields unsatisfactory results. And thus in many existing methods, poor quality points will be removed (Zhu et al., 2019) or be suppressed (Tian et al., 2019).Here we analyze the worst case in FCOS (by Cartesian coordinates) and PolarNet (by Polar coordinates) when training the detectors.

Assume the height and width of input object as h and w, we define L and S as $L = \max(h, w)$ and $S = \min(w, h)$. The range r of learned offsets in FCOS is:

$$r \in (0, L) \tag{1}$$

In Figure 2(c), when the point is close to the boundary, the variance of learned offsets is very large (shorter offset is close to 0 while longer offset is close to L).

In PolarNet, we will learn two pair of offsets for each keypoint, pointing to different corner pairs (TL-BR and TR-BL). We will keep the pair with close offsets for training PolarNet. And thus, the range r of learned offsets is:

$$r \in (\frac{S}{2}, \sqrt{\frac{S^2}{4} + L^2}) \tag{2}$$

The worst case in PolarNet is when the keypoints lie in the center of short line of the objects, the ratio of longer offsets with shorter offsets equals to $\sqrt{1 + \frac{4L^2}{S^2}}$ (See Figure 2(d)), which significantly reduces the variance.

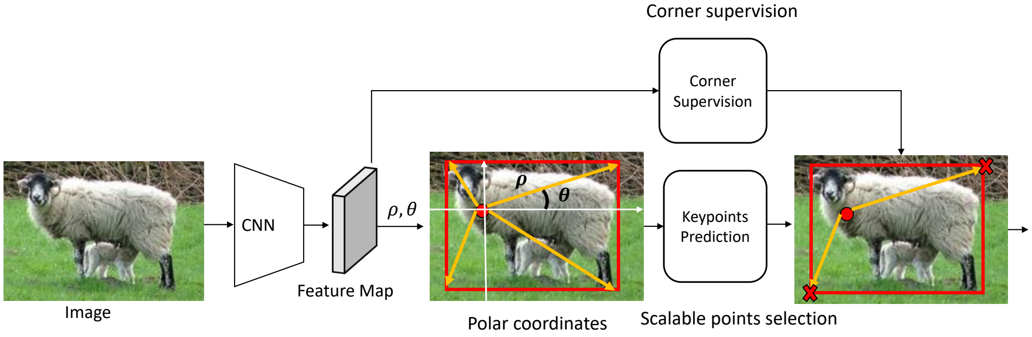


Figure 3: The architecture of our PolarNet detector. Following the CNN backbone, we also apply the Feature Pyramid Network (FPN). The objectness prediction module is a pixel-wise prediction to predict the likelihood of a relevant object in the location of the pixel. Each keypoint is represented by polar coordinates which points to corners. The optimal coordinates is selected for training and the corners are also used for bounding box regression training.

3.2 Overview

We now present the proposed keypoint based detection network (PolarNet), a new anchor-free detector that is capable of better representing keypoints and localizing the objects. Figure 3 gives an overview of the architecture of the proposed PolarNet. An input image is passed through a CNN network to produce a set of feature maps. Note that following the CNN backbone, we also apply the Feature Pyramid Network (FPN) to help deal with scale variances. Based on the feature maps, a Fully Convolutional layer is applied to perform the pixel-wise objectness prediction, which predicts the objectness likelihood of a pixel with respect to a particular object category. Then for each keypoint within the objects, we will learn a set of polar coordinates pointing to the corners of the objects. The keypoints predictor will be used here to select the optimal coordinates for detector training. And the corners will also be used here to supervise bounding box regression learning. Finally, after the selected keypoints are obtained, the keypoints' bounding boxes together with their objectness and likelihood scores are passed through NMS to obtain the final detection result.

Next we will discuss several key modules of the proposed framework, including polar coordinates, pixel-wise objectness prediction, overall detector training, inferences, and other details.

3.3 Pixel-wise Objectness Prediction

The input to this module is a set of feature maps from a CNN backbone, denoted by $F_i \in \mathbb{R}^{H\times W\times C}$ the feature map at layer i of a CNN with a total of C classes. We view any pixel/location (x, y) of the feature map as a potential candidate for keypoints, and the objectness prediction module aims to predict how likely a particular location of the feature map is relevant to some particular class of objects. This idea follows the principle of fully convolutional networks (FCN) for semantic segmentation and has also been used previously in FCOS (Tian et al., 2019).

Specifically, we predict the objectness of a particular location (x, y) by a real vector $(\mathbf{c}_{x,y}, \mathbf{b}_{x,y})$, where $\mathbf{c}_{x,y} \in (0, 1)^C$ denotes a C-dimensional vector of object class prediction scores for (x, y).

3.4 Detector Training

3.4.1 Training losses for keypoints selection

We train the objectness prediction using both classification loss and bounding box regression loss. We define $\mathcal{Q} = (x, y)$ is the set of candidate locations that fall into the ground-truth bounding boxes. The classification loss is defined as

$$L_{\text{cls}} = \frac{1}{|\mathcal{Q}|}\sum_{(x,y)} L_{\text{focal}}(\mathbf{c}_{x,y}, \mathbf{c}^*_{x,y}) \tag{3}$$

where L_{focal} is based on the focal loss (Lin et al., 2017b) and $\mathbf{c}^*_{x,y} \in [0, 1]^C$ denotes the ground-truth class labels at location (x, y).

For bounding box regression, we only consider a location that falls into any ground-truth box. Unlike vanilla keypoint detectors which yield poor quality keypoints due to large variance of learned offsets, in our model, each keypoint has two target boxes which point to different corner pairs. And we assume, the corner pair with similar distance ρ to the keypoints is more suibtale for training the detector. More specifically, we learn a keypoints selector to select suitable corner pair as:

$$t^*_{(x,y)} = \arg\min(|\frac{\rho_{\text{tl}}}{\rho_{\text{br}}} - 1|, |\frac{\rho_{\text{bl}}}{\rho_{\text{tr}}} - 1|) \tag{4}$$

where $t^*_{(x,y)}$ is the label of selected box in (x, y), and $t = 0$ if TL-BR pair is selected otherwise $t = 1$. After removing the undesirable box, we guarantee for each keypoint, at least one box is available for training the detector. The selected box will be used for training the models, and here for each point we learn a binary classifier to indicate which corner pair to use as

$$L_{\text{sel}} = \frac{1}{|\mathcal{Q}|} \sum_{(x,y)} L_{\text{CE}}(\mathbf{t}_{x,y}, \mathbf{t}^*_{x,y}) \tag{5}$$

where $\mathbf{t}_{x,y}$ is the predicted score in point (x, y).

3.4.2 Training loss for corner supervision

In vanilla keypoint detector, the learned offsets will first be converted into boxes, and then a IoU loss is followed to learn the bbox regressor. However, the learned offsets are sensitive to scales, and in IoU loss, the coordinates of bounding box are not explicitly optimized. Inspired by the scale-invariant property of corner features, we aim to enhance the original feature by corner features. We first learn the corner features supervised by corner annotation, and then integrate them into the original feature map to make it robust to scale variance. Here we train the bbox regressors based on the new generated features maps. Suppose the selected polar coordinates in (x, y) is $\mathbf{B_{x,y}}$ as $(\theta_{\text{br}}, \rho_{\text{br}}, \theta_{\text{tl}}, \rho_{\text{tl}})$, and we can easily obtain the corner coordinates $\mathbf{B^*_{x,y}}$ as $(\theta^*_{\text{br}}, \rho^*_{\text{br}}, \theta^*_{\text{tl}}, \rho^*_{\text{tl}})$, and thus we determine the supervision from corners as:

$$L_{\text{corner}} = \frac{1}{|\mathcal{Q}|} \sum_{(x,y)} L_{\text{reg}}(\mathbf{B_{x,y}}, \mathbf{B^*_{x,y}}) \tag{6}$$

$$L_{\text{reg}}(\mathbf{B_{x,y}}, \mathbf{B^*_{x,y}}) = |\rho_{\text{br}} - \rho^*_{\text{br}}| + |\rho_{\text{tl}} - \rho^*_{\text{tl}}| + \tan|\theta_{\text{br}} - \theta^*_{\text{br}}| + \tan|\theta_{\text{tl}} - \theta^*_{\text{tl}}| \tag{7}$$

3.4.3 IoU loss for bounding boxes regression

In addition to corner supervision, here we also keep the original GIoU loss in training bounding boxes. We can easily to obtain the bounding box based on the selected polar coordinate. And we define the bounding box regression loss as:

$$L_{\text{loc}} = \frac{1}{|\mathcal{Q}|} \sum_{(x,y)\in\mathcal{Q}} L_{\text{GIoU}}(\mathbf{B}_{x,y}, \mathbf{B}^*_{x,y}) \tag{8}$$

where L_{GIoU} is based on the GIoU loss (Rezatofighi et al., 2019). We transform the learned offsets into bounding box and optimize the GIoU loss.

3.4.4 Overall training loss and adaptive keypoint training

Combining the above training losses for both objectness prediction and polar coordinates modules, we can define the overall training loss function as follows:

$$L_{\text{overall}} = L_{\text{cls}} + L_{\text{loc}} + L_{\text{corner}} + L_{\text{sel}} \tag{9}$$

3.5 Inference

During the inference stage, PolarNet takes an image as an input and passes it through the CNN network followed by the FPN module to obtain the feature maps. Based on the resulting feature maps, we then predict the objectness scores $\mathbf{c}_{x,y}$ and bounding box offsets $\mathbf{b}_{x,y}$ of each point/location (x, y) on the feature map. After that, we select top k points/locations with the highest class objectness scores as the candidate points to the keypoint prediction module. For each candidate keypoint, we predict $\mathbf{t}_{x,y}$ to determine optimal corner pair and output the optimal box from $\mathbf{b}_{x,y}$ as the final output of the predicted box. Finally, we pass the predicted boxes to a soft-version of Non-Maximum Suppression (NMS) to obtain the final detection result.

4 Experiments

4.1 Experimental Dataset and Setup

We conducted experiments on MSCOCO dataset, which has 80 categories in three splits: train (115k images), val (5k images), and test-dev (20k images). Following common practice, we used the train set to train our model and the val set for ablation studies, and finally report the results on test-dev set for comparison. In our experiments, only bounding box level annotations are used. We consider three types of backbones: ResNet-50 (He et al., 2016), ResNet-101 (He et al., 2016) and ResNeXt-101-DCNv2(Xizhou Zhu & Dai, 2019). For efficiency, ResNet-50 and ResNet-101 is used in our ablation study.

4.2 Experimental Results

Table 1 shows the results on the COCO val set by comparing our method with other popular anchor-free detectors mostly with ResNet-101 (for CornerNet and CenterNet we add results on Hourglass-104 since they are designed based on Hourglass backbones). For comparison purposes, we also include two anchor-based detectors, including the popular RetinaNet(Lin et al., 2017b) and the state-of-the-art ATSS (Zhang et al., 2020) .

Object Detectors	Anchor-free	Backbone	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
RetinaNet(Lin et al., 2017b)	Anchor-based	R-101	39.1	59.1	42.3	21.8	42.7	50.2
ATSS (Zhang et al., 2020)	Anchor-based	R-101	43.6	62.1	47.4	26.1	47.0	53.6
CornerNet(Law & Deng, 2018)	Keypoint Position	R-101	30.2	44.1	32.0	13.3	33.3	42.7
CornerNet(Law & Deng, 2018)	Keypoint Position	HG-104	40.5	56.5	43.1	19.4	42.7	53.9
RPDet (Yang et al., 2019)	Keypoint Position	R-101	41.0	62.9	44.3	23.6	44.1	51.7
CenterNet(Zhou et al., 2019a)	Keypoint Offsets	R-101	34.6	53.0	36.9	-	-	-
CenterNet(Zhou et al., 2019a)	Keypoint Offsets	HG-104	42.1	61.1	45.9	24.1	45.5	52.8
FSAF(Zhu et al., 2019)	Keypoint Offsets	R-101	40.9	61.5	44.0	24.0	44.2	51.3
FoveaBox(Kong et al., 2019)	Keypoint Offsets	R-101	40.6	60.1	43.5	23.3	45.2	54.5
FCOS (Tian et al., 2019)	Keypoint Offsets	R-101	41.5	60.7	45.0	24.4	44.8	51.6
PolarNet (ours)	Keypoint Offsets	R-101	**43.8**	62.3	47.1	26.6	47.1	53.6

Table 1: Performance evaluation of popular keypoint based detectors and two anchor-baesd detectors. The models are trained on MSCOCO train with 115k images, and validated on MSCOCO val set with 5k images. “R-101” denotes ResNet-101 backbone and “HG-104” denotes Hourglass-104 backbone.

From the results, we can see that all anchor-free/keypoint-based methods outperform RetinaNet which uses predefined anchors and IoU matching methods. This confirms the advantage of keypoint-based detection methods over heuristic anchor-based designs. However, the existing anchor-free detectors are worse than ATSS, which is a recent state-of-the-art anchor-based method by borrowing and adapting some advanced strategies from anchor-free methods. By examining the results of our PolarNet, we found that it outperforms all the existing keypoint-based detectors. Compared with Keypoint Position methods such as CornerNet, our model is able to capture more contextual information of the objects. Compared with other Keypoint Offsets methods such as FCOS, our method represents keypoints by polar coordinates which eliminate poor keypoints and learn box via corner supervision. This merit significantly boost the performance of our method. Finally, our method is also better than the state-of-the-art anchor-based ATSS, but eliminates the need of manually designed anchors.

4.3 Ablation Study

This ablation study aims to examine the effeciveness of polar coordinates and corner supervision in our PolarNet. Table 2 shows the results of our ablation study and our baseline FCOS. From the results in Table 2, we can see that original FCOS produce poor results without Centerness due to the poor points. The results of FCOS improves from 34.3% to 38.5% by polar coordinates and it also outperforms FCOS with Centerness (37.1%), which indicates the effectiveness of polar coordinates. We further add corner supervision in FCOS, and the results improve from 37.1% to 38.3%, which proves that our corner supervision is effective.

Methods	Backbone	AP	AP_{50}	AP_{75}
FCOS	R-50	34.3	52.0	35.9
FCOS + Centerness	R-50	37.1	55.9	39.8
FCOS + Centerness + Corner	R-50	38.3	57.0	41.8
FCOS + Polar	R-50	38.5	57.3	41.4
PolarNet	R-50	39.3	57.6	42.6

Table 2: Ablation study on different components in PolarNet. Models are trained on COCO train2017 and tested on COCO val2017 with ResNet-50.

4.4 Comparison with State-of-the-art Detectors

We compare PolarNet with other state-of-the-art detectors on COCO test-dev set. Unlike the previous experiments, we train our models on backbone ResNeXt-101-DCNv2. Table 3 shows the results on COCO test-dev under the single-model single-scale settings. PolarNet outperforms all single-stage detectors by a substantial margin (except ATSS), also outperform a variety of two-stage/multi-stage detectors, and achieves the best results among all the keypoint based detectors. This promising result validates our hypothesis of learning keypoints based on polar coordinates with corner supervision is able to better predict the objects. Specifically, compared with other center-based methods such as CenterNet or FSFA, our method not only extracts features from the central region, but also encodes features from the whole bounding boxes. Besides, our corner supervision is scale-invariant and thus shows better results in localization. Finally, compared with the SOTA anchor-based ATSS, our method eliminates the need of using anchors and thus avoids heuristic anchor design.

Method	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Anchor-based								
Multi-stage								
FRCN-FPN (Lin et al., 2017a)	R-101	9.9	36.2	59.1	39.0	18.2	39.0	48.2
Cascade R-CNN(Cai & Vasconcelos, 2018)	R-101	8.0	42.8	62.1	46.3	23.7	45.5	55.2
Libra R-CNN (Pang et al., 2019)	X-101-64x4d	5.6	43.0	64.0	47.0	25.3	45.6	54.6
TridentNet (Li et al., 2019)	R-101-DCN	1.3	46.8	67.6	51.5	28.0	51.2	60.5
FreeAnchor (Zhang et al., 2019)	X-101-32x8d	5.4	44.8	64.3	48.4	27.0	47.9	56.0
Fitness-NMS (Tychsen-Smith & Petersson, 2018)	R-101	5.0	41.8	60.9	44.9	21.5	45.0	57.5
Single-stage								
RefineDet(Zhang et al., 2018)	R-101	-	36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet (Lin et al., 2017b)	R-101	8.0	39.1	59.1	42.3	21.8	42.7	50.2
AB+FSAF (Zhu et al., 2019)	R-101	7.1	40.9	61.5	44.0	24.0	44.2	51.3
AB+FSAF (Zhu et al., 2019)	X-101-64x4d	4.2	42.9	63.8	46.3	26.6	46.2	52.7
M2Det (Zhao et al., 2019)	VGG-16	11.8	41.0	59.7	45.0	22.1	46.5	53.8
ATSS (Zhang et al., 2020)	X-101-64x4d-DCNv2	7.1	47.7	66.5	51.9	29.7	50.8	59.4
Anchor-free								
GA-FRCN (Wang et al., 2019)	R-50	9.4	39.8	59.2	43.5	21.8	42.6	50.7
GA-RetinaNet (Wang et al., 2019)	R-50	10.8	37.1	56.9	40.0	20.1	40.1	48.0
CornerNet (Law & Deng, 2018)	HG-104	3.1	40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet (Zhou et al., 2019b)	HG-104	2.8	40.2	55.5	43.2	20.4	43.2	53.1
FoveaBox (Kong et al., 2019)	R-101	11.2	40.6	60.1	43.5	23.3	45.2	54.5
FoveaBox (Kong et al., 2019)	X-101	-	42.1	61.9	45.2	24.9	46.8	55.6
FCOS (Tian et al., 2019)	R-101	9.3	41.5	60.7	45.0	24.4	44.8	51.6
FCOS w/ imprv(Tian et al., 2019)	X-101-64x4d	5.4	44.7	64.1	48.4	27.6	47.5	55.6
CenterNet (Zhou et al., 2019a)	HG-104	7.8	42.1	61.1	45.9	24.1	45.5	52.8
CenterNet (Duan et al., 2019)	HG-104	3.3	44.9	62.4	48.1	25.6	47.4	57.4
RPDet (Yang et al., 2019)	R-101-DCN	8.0	45.0	66.1	49.0	26.6	48.6	57.5
SAPD (Zhu et al., 2019)	X-101-64x4d-DCN	4.5	47.4	67.4	51.1	28.1	50.3	61.5
PolarNet (ours)	X-101-64x4d-DCNv2	8.1	**48.0**	67.1	52.3	30.0	51.2	59.5

Table 3: Comparison of PolarNet with other state-of-the-art two-stage or one-stage detectors (*single-model and single-scale results*). All models were trained on COCO train set and tested on test-dev.

5 Conclusion

In this paper, we presented a view of the popular anchor-free object detectors from the object modeling based detection perspective. We argue that the existing keypoint based detectors yields a lot of poor quality keypoints based on Cartesian coordinates and the bounding box regression learning is also based on scale-variant features. To overcome the limitation, we proposed PolarNet, a new keypoint based detector which represents keypoints by polar coordinates which avoids poor quality keypoints as well as learning scale-invariant corner features for object localization. Our new detector obtained the state-of-the-art results on the MS COCO test-dev set under the single-model single-scale settings.

REFERENCES

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.

Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *CVPR*, 2016.

Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *arXiv preprint arXiv:1904.03797*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.

Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019.

Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017b.

Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019.

Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *arXiv preprint arXiv:1612.08242*, 2016.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pp. 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

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 *CVPR*, 2019.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, 2018.

Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *CVPR*, 2019.

Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. Recent advances in deep learning for object detection. *Neurocomputing*, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Stephen Lin Xizhou Zhu, Han Hu and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.

Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, 2019.

Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018.

Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *NeurIPs*, 2019.

Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, 2019.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019a.

Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019b.

Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. *arXiv preprint arXiv:1911.12448*, 2019.

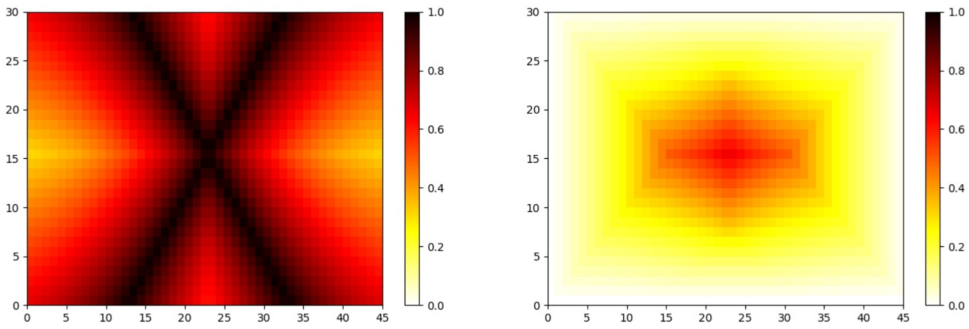
Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019.

Yousong Zhu, Chaoyang Zhao, Jinqiao Wang, Xu Zhao, Yi Wu, and Hanqing Lu. Couplenet: Coupling global structure with local parts for object detection. In *ICCV*, 2017.

A More Ablation Study

A.1 Scale Variance of FCOS and PolarNet

In this section, we explore the scale variance of offsets learned by FCOS and PolarNet. Each keypoint predicts a set of offsets $\{X\}$ (For FCOS, $\{X\} = \{t, r, b, l\}$, etc.), and we obtain the scale ratio r as $r = \frac{\min\{X\}}{\max\{X\}}$. If the variance of the learned offsets is small, r is close to 1 and this point is suitable for predict offsets. In Figure 4 we show the heatmaps of scale ratio of FCOS and PolarNet. For FCOS, only the keypoints within central region reports low variance, and for PolarNet, the region of low variance area is much larger than FCOS, which means we have more high quality keypoints for training.



(a) Polar (b) Cartesian

Figure 4: (a) and (b) show the comparison of heatmaps of Polar coordinates and Cartesian coordinates.

A.2 Importance of GIoU loss

In this section, we explore the importance of the GIoU loss. We can directly predict objects by learned corners. However, this way fails to differentiate the quality of the positive predictions (quality means the IoU with objects). If the GIoU loss is removed, the performance drops from 39.3% to 36.5% (See Table 4), indicating the importance of the GIoU loss.

Methods	Backbone	AP	AP_{50}	AP_{75}
FCOS	R-50	37.1	55.9	39.8
FCOS-Center	R-50	38.1	56.7	41.4
PolarNet w/o GIoU	R-50	36.5	55.1	39.0
PolarNet	R-50	39.3	57.6	42.6

Table 4: Ablation study on GIoU in PolarNet. Models are trained on COCO train2017 and tested on COCO val2017 with ResNet-50.

A.3 Compare with Center Sampling

From Figure 4, we know the variance of central region of FCOS is much smaller than other regions. In this section, we compare our PolarNet with FCOS trained with points within only central region. The results are shown in Table 4. The results if FCOS-Center is much better than original FCOS (37.1% vs 38.1%), which indicates keypoints with large variance of learned offsets hurt the detection performance. And our PolarNet is still better than this FCOS variant.

B Implementation Details

B.1 Network Backbones

Following the recent state-of-the-art object detectors such as FCOS (Tian et al., 2019) and RetinaNet (Lin et al., 2017b), we adopt the ResNet (He et al., 2016) and ResNeXt (Xie et al., 2017) CNN network as our backbone architecture. ResNet and ResNeXt are two fully convolutional networks, which are composed of a sequence of residual modules and were first used for image classification. Residual module first encodes the input feature by a sequence of convolution and normalization layers, and then aggregates the generated feature map with the original input features. In order to predict objects with large scale variance, we also apply the Feature Pyramid Network (FPN) (Lin et al., 2017a) in our approach, which combines the shallow layer features with deep layer features by the latent connection. To learn a scale-robust detector, each level of FPN is responsible for a certain scale of objects, making it very suitable for object detection. Specifically, we use 5 FPN levels to make prediction, with stride 8, 16, 32, 64 and 128 compared with the original image, and each of the level is responsible for a certain scale of the objects: (0, 64], (64, 128], (128, 256], (256, 512] and (512, INF]. We adopt ResNet-50 (He et al., 2016), ResNet-101 (He et al., 2016) and DCN2-ResNeXt-101-64x4 (Xizhou Zhu & Dai, 2019) as our backbone architecture in our experiments.

B.2 Initialization and Optimization

We train the model from weights pre-trained on ImageNet classification task and other parameters are initialized by the same methods as RetinaNet (Lin et al., 2017b). The model is trained with SGD optimization methods with 180k iterations with 16 images per mini-batch. The initial learning rate is set to 1e-2 and is reduced 10 times at 120k and 160k iterations. We re-scale the the input images into 800x1333 pixels before training. We use the same data augmentation strategy presented in (Tian et al., 2019) when training the model, and for each image, the top-50 predictions are produced