CentripetalNet: Pursuing High-quality Keypoint Pairs for Object Detection

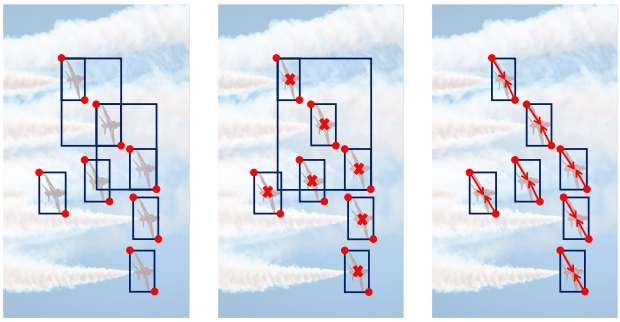
Zhiwei Dong[1,2] Guoxuan Li[3] Yue Liao[4] Fei Wang[2*] Pengju Ren[1] Chen Qian[2]
[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]SenseTime Research [3]University of Chinese Academy of Sciences [4]Beihang University
kivee@foxmail.com; liguoxuan18@mails.ucas.ac.cn; liaoyue.ai@gmail.com;
{wangfei, qianchen}@sensetime.com; pengjuren@xjtu.edu.cn

Abstract

Keypoint-based detectors have achieved pretty-well performance. However, incorrect keypoint matching is still widespread and greatly affects the performance of the detector. In this paper, we propose CentripetalNet which uses centripetal shift to pair corner keypoints from the same instance. CentripetalNet predicts the position and the centripetal shift of the corner points and matches corners whose shifted results are aligned. Combining position information, our approach matches corner points more accurately than the conventional embedding approaches do. Corner pooling extracts information inside the bounding boxes onto the border. To make this information more aware at the corners, we design a cross-star deformable convolution network to conduct feature adaption. Furthermore, we explore instance segmentation on anchor-free detectors by equipping our CentripetalNet with a mask prediction module. On MS-COCO test-dev, our CentripetalNet not only outperforms all existing anchor-free detectors with an AP of 48.0% *but also achieves comparable performance to the state-of-the-art instance segmentation approaches with a* 40.2% *MaskAP. Code is available at* `https://github.com/KiveeDong/CentripetalNet`.

1. Introduction

Object detection is a fundamental topic in various applications of computer vision, such as automatic driving, mobile entertainment, and video surveillance. It is challenging in large appearance variance caused by scale deformation and occlusion. With the development of deep learning, object detection has achieved great progress [10, 9, 30, 27, 25, 21, 11, 22, 1, 17, 34, 33, 36, 20, 19, 16, 39]. The anchor-based methods [9, 30, 25] have led the fashion in the past few years, but it is difficult to manually design a set of suitable anchors. Additionally, the anchor-based methods suffer from the significant imbalance between negative and positive anchor boxes. To improve it, the CornerNet [17] proposes a novel method to represent a bounding box as a pair of corners, i.e, top-left corner and bottom-right corner. Based on this idea, lots of corner-based methods [17, 7] have emerged. The corner-based detection framework has been leading the new trends in the object detection area gradually. The corner-based detection framework can be divided into two steps including corner points prediction and corner matching. In this paper, we focus on the second step.



(a) CornerNet (b) CenterNet (c) CentripetalNet

Figure 1. (a) CornerNet generates some false corner pairs because of similar embeddings caused by similar appearance. (b) CenterNet removes some false corner pairs through center prediction, but it naturally can not handle some dense situation. (c) CentripetalNet avoids the drawbacks of CornerNet and CenterNet.

The conventional methods [17, 7] mainly use an associative embedding method to pair corners, where the network is required to learn an additional embedding for each corner to identify whether two corners belong to the same bounding-box. In this manner, if two corners are from the same box, they will have similar embeddings, otherwise, their embeddings will be quite different. Associative embedding-based detectors have achieved pretty-well performance in object detection, but they also have some limitations. Firstly, the training process employs push and pull loss to learn the embedding of each point. Push loss will

*Corresponding author

be calculated between points that do not belong to the same object to push them away from each other. While the pull loss is only considered between points from the same object. Thus, during training, the network is actually trained to find the unique matching point within all potential points of the diagonal. It is highly sensitive to outliers and the training difficulty will increase dramatically when there are multiple similar objects in one training sample. Secondly, the embedding prediction is based on the appearance feature without using position information, thus as shown in Figure 1, if two objects have a similar appearance, the network tends to predict the similar embeddings for them even if they are far apart.

Based on the above considerations, we propose CentripetalNet using centripetal shift to match corners, along with a cross-star deformable convolution module for better prediction of centripetal shift. Given a pair of corners, we define a 2-D vector, i.e., centripetal shift, for each corner, where the centripetal shift encodes the spatial offset from the corner to the center point of the box. In this way, each corner can generate a center point based on the centripetal shift, thus if two corners belong to the same bounding-box, the center points generated by them should be close. The quality of the match may be represented by the distance between two centers and the geometric center of this match. Combined with position information of each corner point, the method is robust to outliers compared to the associative embedding approach. Furthermore, we propose a novel component, namely cross-star deformable convolution, to learn not only a large receptive field but also the geometric structure of 'cross star'. We observe that there are some 'cross stars' in the feature map of the corner pooling output. The border of the 'cross star' contains context information of the object because corner pooling uses max and sum operations to extend the location information of the object to the corner along the 'cross star' border. Thus, we embed the object geometric and location information into the offset field of the deformable convolution explicitly. Equipped with the centripetal shift and cross-star deformable convolution, our model has achieved a significant performance gain compared to CornerNet, from 42.1% AP to 47.8% AP on MS-COCO test-dev2017. Moreover, motivated by the benefits of multi-task learning in object detection, we add instance mask branch to further improve the accuracy. We apply the RoIAlign to pool features from a group of predicted regions of interests (RoIs) and feed the pooled features into a mask head to generate the final segmentation prediction. To demonstrate the effectiveness of our CentripetalNet, we evaluate the method on the challenging MS-COCO benchmark [23]. CentripetalNet not only outperforms all existing anchor-free detectors with an AP of 48.0% but also achieves comparable performance with the state-of-the-art instance segmentation methods.

2. Related Work

Anchor-based Approach: Anchor-based detectors set anchor boxes in each position of the feature map. The network predicts the probability of having objects in each anchor box and adjusts the size of the anchor boxes to match the object.

Two-stage methods are derived from R-CNN series of methods [10, 12, 9] which first extract RoIs using a selective search method [32] then classify and regress them. Faster R-CNN [30] employs a region proposal network(RPN) to generate RoIs by modifying preset anchor boxes. Mask R-CNN [11] replaces the RoIPool layer with the RoIAlign layer using bilinear interpolation. Its mask head uses a top-down method to obtain instance segmentations.

Without extracting RoIs, one-stage methods directly classify and regress the preset anchor boxes. SSD [25] utilizes features maps from multiple different convolution layers to classify and regress anchor boxes with different strides. Compared with YOLO [27], YOLOv2 [28] uses preset anchors. However, the above methods are bothered by the imbalance between negative and positive samples. RetinaNet [22] uses focal loss to mitigate classification imbalance problem. RefineDet [41] refines the FPN structure by introducing the anchor refinement module to filter and eliminate negative samples.

Anchor-free Approach: For anchor-based methods, the shape of anchor boxes should be carefully designed to fit the target object. Compared to the anchor-based approach, anchor-free detectors no longer need to preset anchor boxes. Mainly two types of anchor-free detectors are proposed.

The first type of detectors directly predict the center of an object. Yolov1 [27] predicts the size and shape of the object at the points near the center of the object. DenseBox [14] introduces a fully convolutional neural network framework to gain high efficiency. UnitBox [40] uses IoU loss to regress the four bounds as a whole unit. Since the number of positive samples is relatively small, these detectors suffer from a quite low recall. To cope with this problem, FCOS [31] treats all the points inside the bounding box of the object as positive samples. It detects all the positive points and the distance from the point to the border of the bounding box.

For the second type, detectors predict keypoints and group them to get bounding boxes. CornerNet [17] detects top-left and bottom-right corners of the object and embeds them into an abstract feature space. It matches corners of the same object by computing distance between embeddings of each pair of points. ExtremeNet [42] detects the top-, left-, bottom-, rightmost, and center points of the object. Combined with Deep Extreme Cut [26], the extreme points can be used for instance segmentation. These detectors need some specific grouping methods to obtain bounding boxes. RepPoints [38] uses deformable convolutional networks(DCN) [6] to get sets of points used to represent

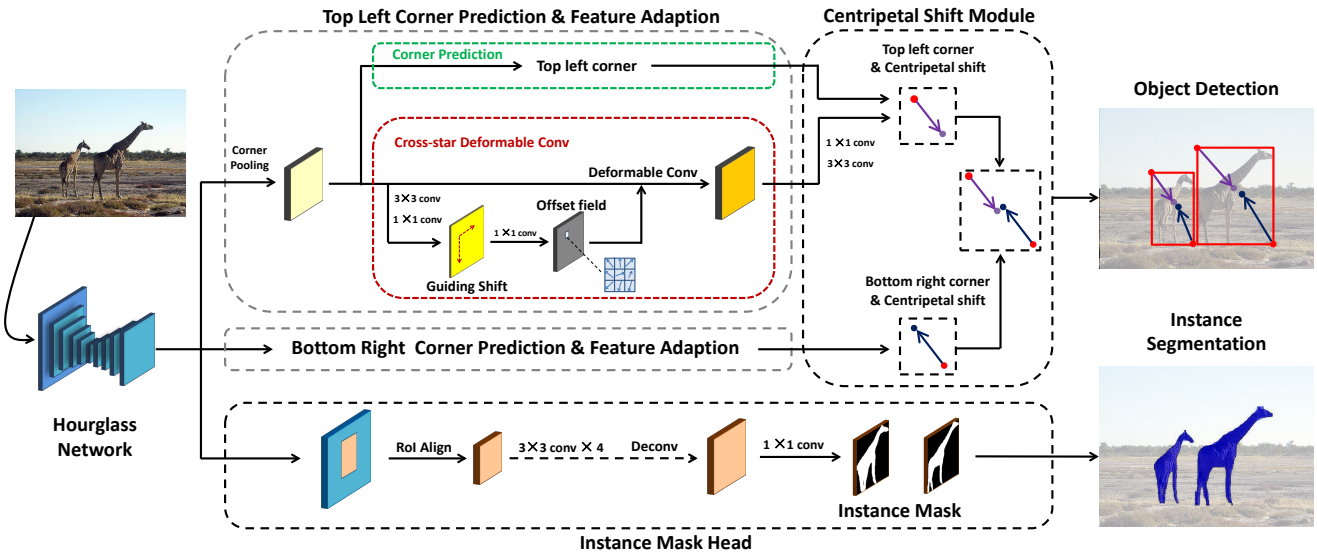


Figure 2. An overview of CentripetalNet. As the corner prediction and feature adaption of top-left corner and bottom-right corner are similar, we only draw top-left corner module for simplicity. Centripetal shift module gets predicted corners and adapted features, then it predicts the centripetal shift of each corner and performs corner matching based on the predicted corners and centripetal shifts. During matching, if the positions of the shifted corners are close enough, they form a bounding box with a high score.

objects. The converting functions are carefully designed to convert point sets to bounding boxes. CenterNet [7] adds a center detection branch into CornerNet and largely improves the performance by center point validation.

These methods usually achieve high recall with quite many false detections. The main challenge resides in the approach to match keypoints of the same object. In this work, we propose a centripetal shift which encodes the relationship between corners and gets their corresponding centers by predicted spatial information, thus we can build the connection between the top-left and bottom-right corners through their sharing center.

3. CentripetalNet

We first provide an overview of the approach. As shown in Figure 2, CentripetalNet consists of four modules, namely corner prediction module, centripetal shift module, cross-star deformable convolution, and instance mask head. We first generate corner candidates based on the CornerNet pipeline. With all the corner candidates, we then introduce a centripetal shift algorithm to pursue high-quality corner pairs and generate final predicted bounding boxes. Specifically, the centripetal shift module predicts the centripetal shifts of the corner points and matches corner pairs whose shifted results decoded from their locations and centripetal shifts are aligned. Then, we propose a novel cross-star deformable convolution, whose offset field is learned from the shifts from corners to their corresponding centers, to conduct feature adaption for enriching the visual features of the corner locations, which is important to improve the accuracy of the centripetal shift module. Finally, we add an instance mask module to further improve the detection performance and extend our method to the instance segmentation area. Our method takes the predicted bounding boxes of centripetal shift module as region proposals, uses RoIAlign to extract the region features and applies a small convolution network to predict the segmentation masks. Overall, our CentripetalNet is trained end-to-end and can inference with or without the instance segmentation module.

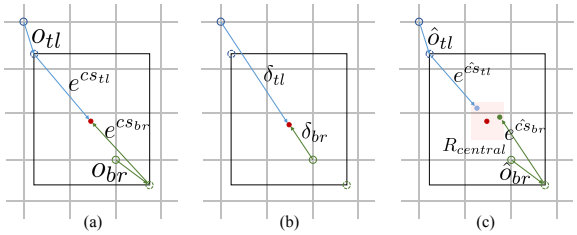


Figure 3. (a) When mapping the ground truth corner to the heatmap, local offset O_{tl}(or O_{br}) is used to compensate the precision loss as in [17]. (b) The guiding shift δ is the shift from ground truth corner on the heatmap to center of bounding box. (c) $R_{central}$ is the central region we use to match the corners.

3.1. Centripetal Shift Module

Centripetal Shift. For $bbox^i = (tlx^i, tly^i, brx^i, bry^i)$, its geometric center is $(ctx^i, cty^i) = (\frac{tlx^i + brx^i}{2}, \frac{tly^i + bry^i}{2})$. We define the centripetal shifts for its top-left corner and

bottom-right corner separately as

$$\begin{aligned} cs_{tl}^{i} &= (log(\tfrac{ctx^{i}-tlx^{i}}{s}), log(\tfrac{cty^{i}-tly^{i}}{s})) \\ cs_{br}^{i} &= (log(\tfrac{brx^{i}-ctx^{i}}{s}), log(\tfrac{bry^{i}-cty^{i}}{s})) \end{aligned} \tag{1}$$

Here we use log function to reduce the numerical range of centripetal shift and make the learning process easier.

During training, we apply smooth L1 loss at the locations of ground truth corners

$$L_{cs} = \frac{1}{N}\sum_{k=1}^{N}[\mathcal{L}_1(cs_{tl}^{k}, \hat{cs}_{tl}^{k}) + \mathcal{L}_1(cs_{br}^{k}, \hat{cs}_{br}^{k})] \tag{2}$$

where $\mathcal{L}_1$ is SmoothL1 loss and N is the number of ground truths in a training sample.

Corner Matching. To match the corners, we design a matching method using their centripetal shifts and their locations. It is intuitive and reasonable that a pair of corners belonging to the same bounding box should share the center of that box. As we can decode the corresponding center of a predicted corner from its location and centripetal shift, it is easy to compare whether the centers of a pair of corners are close enough and close to the center of the bounding box composed of the corner pair, as shown in Figure 3(c). Motivated by the above observations, our method goes as follows. Once the corners are obtained from corner heatmaps and local offset feature maps, we group the corners that are of the same category and satisfy $tlx < brx \wedge tly < bry$ to construct predicted bounding boxes. For each bounding box $bbox^{j}$, we set its score as the geometric mean of its corners' scores, which are obtained by applying $softmax$ on predicted corner heatmaps.

Then, as shown in Figure 3 we define a central region for each bounding box as Equation 3 to compare the proximity of decoded centers and the bounding box center.

$$R_{central} = \{(x, y) | x \in [ctlx, cbrx], y \in [ctly, cbry]\} \tag{3}$$

and the corners of $R_{central}$ are computed as

$$\begin{cases} ctlx = \frac{tlx+brx}{2} - \frac{brx-tlx}{2}\mu \\ ctly = \frac{tly+bry}{2} - \frac{bry-tly}{2}\mu \\ cbrx = \frac{tlx+brx}{2} + \frac{brx-tlx}{2}\mu \\ cbry = \frac{tly+bry}{2} + \frac{bry-tly}{2}\mu \end{cases} \tag{4}$$

where $0 < \mu \leq 1$ indicates that width and height of central region are μ times of the bounding box's width and height. With the centripetal shift, we can decode the center (tl_{ctx}, tl_{cty}) and (br_{ctx}, br_{cty}) for top-left corner and bottom-right corner separately.

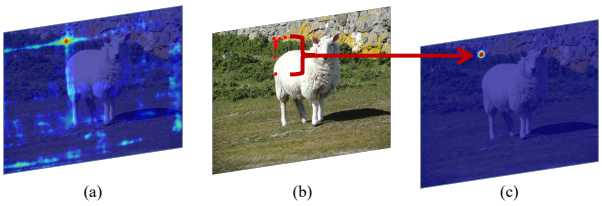
Then we calculate the score weight w^{j} for each predicted bounding box that satisfies $(tl_{ctx}^{j}, tl_{cty}^{j}) \in R_{central}^{j} \wedge (br_{ctx}^{j}, br_{cty}^{j}) \in R_{central}^{j}$ as follows

$$w^{j} = e^{-\frac{|br_{ctx}^{j} - tl_{ctx}^{j}||br_{cty}^{j} - tl_{cty}^{j}|}{(cbrx^{j} - ctlx^{j})(cbry^{j} - ctly^{j})}} \tag{5}$$

which means that the regressed centers are closer, the predicted box has a higher scoring weight. For other bounding boxes, we set $w^{j} = 0$. Finally we can re-score the predicted bounding boxes by multiplying the score weights.

3.2. Cross-star Deformable Convolution

Due to corner pooling, there are some 'cross stars' in the feature map as shown in Figure 4(a). The border of the 'cross star' maintains abundant context information of the object because corner pooling uses max and sum operations to extend the location information of the object to the corner along the 'cross star' border. To capture the context information on 'cross stars', a common approach is to use deformable convolution to shift the receptive field. However, as shown in Figure 7(a), standard deformable convolution cannot well align the sampling points with the 'cross star' border which indicates it is hard to learn the geometric structure of cross-star without any prior information. Following the above observation, we proposed the cross-star deformable convolution, a novel convolution operation to enhance the visual features at corners.



(a) (b) (c)

Figure 4. (a) 'Cross star' caused by corner pooling. (b) The sampling points of the cross-star deformable convolution at the corner. (c) Top-left corner heatmap from corner prediciton module.

Our proposed cross-star deformable convolution is depicted in Figure 2. Firstly, we feed the feature map of the corner pooling into the cross-star deformable convolution module. Since the geometric structure of the 'cross star' is relative to the scale of the object and the direction of corners relative to the center, we embed a guiding shift, the shift from corner to center as shown in Figure 3(b), to guide the offset field branch explicitly. Formally, guiding shift is a vector $(\delta_x, \delta_y) = (\frac{ctx}{s} - \lfloor\frac{x}{s}\rfloor, \frac{cty}{s} - \lfloor\frac{y}{s}\rfloor)$. The value of the vector on each axis represents the scale and the signal of the vector indicates the direction. Note that the supervision of direction information guides the network to pay attention to the cross star border near the object. Specifically, the offset field is carried out on three convolution layers. The first two convolution layers embed the corner pooling output into the

feature map, which is supervised by the following loss:

$$L_\delta = \frac{1}{N}\sum_{k=1}^{N}[L_1(\delta_{tl}, \hat{\delta}_{tl}) + L_1(\delta_{br}, \hat{\delta}_{br})] \tag{6}$$

where δ means the guiding shift and is defined as

$$\delta_{tl}^i = (\frac{ctx^i}{s} - \lfloor\frac{tlx^i}{s}\rfloor, \frac{cty^i}{s} - \lfloor\frac{tly^i}{s}\rfloor) \tag{7}$$

The second convolution layer maps the above feature into the offset field, which contains the context and geometric information explicitly. Guiding shift embeds the scale and direction geometric information into the offset field branch and reduces the learning difficulty. By visualizing the learned offset field as shown in Figure 7c, our cross-star deformable convolution can efficiently learn the geometric information of 'cross star' and extract information of 'cross star' border.

3.3. Instance Mask Head

We treat the detection results before soft-NMS as region proposals and use a fully convolutional neural network to predict the mask on top of them. To make sure that the detection module could produce proposals, we first pretrain CentripetalNet for a few epochs.We select top k scored proposals and perform RoIAlign on top of the feature map from the backbone network to get their features. We set the size of RoIAlign to 14×14 and predict a mask of 28×28.

After getting the features from RoIs, we apply four consecutive 3×3 convolution layers, then use a transposed convolution layer to upsample the feature map to a 28×28 mask map $\hat{m}$. During training, we apply cross entropy loss for each region proposal

$$L_{mask} = \frac{1}{N}\sum_{k=1}^{N} CE(m_i, \hat{m}_i) \tag{8}$$

4. Experiments

4.1. Experimental Setting

Dataset We train and validate our method on the MS-COCO 2017 dataset. We train our model on the train2017 split with about 115K annotated images and validate our method on the val2017 split with 5K images. We also report the performance of our model on test-dev2017 for the comparison with other detectors.

Multi-task Training Our final objective function is

$$L = L_{det} + L_{off} + \alpha L_\delta + L_{cs} + L_{mask} \tag{9}$$

where L_{det} and L_{off} are defined as CornerNet. We set α to 0.05, as we find that large α degrades the performance of the network. As in CornerNet, we add intermediate supervision when we use Hourglass-104 as the backbone network. However, for the instance segmentation mask, we only use the feature from the last layer of the backbone to get proposals and calculate L_{mask}.

Implementation Details We train our model on 16 32GB NVIDIA V100 GPUs with a batch size of 96(6 images per GPU), and we use Adam optimizer with an initial learning rate of 0.0005. To compare with other state-of-the-art models, we train our model for 210 epochs and decay the learning rate by 10 at the 180th epoch. In the ablation study, we use Hourglass-52 as the backbone and train 110 epochs, decaying the learning rate at the 90th epoch if not specified. During training, we randomly crop the input images and resize them to 511×511, and we also apply some usual data augmentations, such as color jitter and brightness jitter.

During testing, we keep the resolution of input images and pad them with zeros before feeding them into the network. We use flip augmentation by default, and report both of single-scale and multi-scale test results on MS-COCO test-dev2017. To get the corners, we follow the steps of CornerNet. We firstly apply $softmax$ and 3×3 max pooling on the predicted corner heatmaps and select the $top100$ scored top-left corners and $top100$ bottom-right corners, then refine their locations using the predicted local offsets. Next, we can group and re-score corner pairs as described in section 3.2. In detail, we set $\mu = \frac{1}{2.1}$ for those bounding boxes with an area larger than 3500, and $\mu = \frac{1}{2.4}$ for others. Finally, we apply soft-NMS then keep the $top100$ results in the remaining bounding boxes whose scores are above 0.

4.2. Comparison with state-of-the-art models

Object detection As shown in Table 1, CentripetalNet with Hourglass-104 as the backbone network achieves an AP of 46.1% at single-scale and 48.0% at multi-scale on MS-COCO test-dev 2017, which are the best performance in all anchor-free detectors. Compared to the second-best anchor-free detector, CenterNet(hourglass-104), our model achieves 1.2% and 1.0% AP improvement at single-scale and multi-scale separately. Compared to CenterNet, the improvement of CentripetalNet comes from large and medium object detection, which is just the weakness of CenterNet, as centers of large objects are more difficult to be located than those of small objects, from the perspective of probability. Compared with the two-stage detectors(without ensemble), our model is competitive as its performance is close to the state-of-the-art $48.4\%AP$ of TridentNet [18].

Moreover, as presented in Table 2 the AR metric of CentripetalNet outperforms all other anchor-free detectors on all sizes of objects. We suppose that the advantages of CentripetalNet's recall lie in two aspects. Firstly, the corner matching strategy based on centripetal shift can eliminate many high-scored false detections compared to CornerNet.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage:							
Faster R-CNN w/FPN [21]	ResNet-101 [13]	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN [11]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
HTC [2]	ResNeXt-101	47.1	63.9	44.7	22.8	43.9	54.6
PANet(multi-scale) [24]	ResNeXt-101	47.4	67.2	51.8	30.1	**51.7**	60.0
TridentNet(multi-scale) [18]	ResNet-101-DCN	**48.4**	**69.7**	**53.5**	**31.8**	51.3	**60.3**
Single-stage anchor-based:							
SSD513 [25]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [29]	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9
RetinaNet800 [22]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
Single-stage anchor-free:							
ExtremeNet(single-scale) [42]	Hourglass-104	40.2	55.5	43.2	20.4	43.2	53.1
CornerNet511(multi-scale) [17]	Hourglass-104	42.1	57.8	45.3	20.8	44.8	56.7
FCOS [31]	ResNeXt-101	42.1	62.1	45.2	25.6	44.9	52.0
ExtremeNet(multi-scale) [42]	Hourglass-104	43.7	60.5	47.0	24.1	46.9	57.6
CenterNet511(single-scale) [7]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
RPDet(single-scale) [38]	ResNet-101-DCN	45.0	66.1	49.0	26.6	48.6	57.5
RPDet(multi-scale) [38]	ResNet-101-DCN	46.5	**67.4**	50.9	**30.3**	49.7	57.1
CenterNet511(multi-scale) [7]	Hourglass-104	47.0	64.5	50.7	28.9	49.9	58.9
CentripetalNet w.o/mask(single-scale)	Hourglass-104	45.8	63.0	49.3	25.0	48.2	58.7
CentripetalNet w.o/mask(multi-scale)	Hourglass-104	47.8	65.0	51.5	28.9	50.2	59.4
CentripetalNet(single-scale)	Hourglass-104	46.1	63.1	49.7	25.3	48.7	59.2
CentripetalNet(multi-scale)	Hourglass-104	**48.0**	65.1	**51.8**	29.0	**50.4**	**59.9**

Table 1. Object detection performance comparison on MS-COCO test-dev.

Secondly, our corner matching strategy does not depend on the center detection, thus CentripetalNet can preserve those correct bounding boxes which are mistakenly removed in CenterNet because of the missed detection of centers.

Method	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
CornerNet511-104 [17]	36.4	55.7	60.0	38.5	62.7	77.4
CenterNet511-104 [7]	37.5	60.3	64.8	45.1	68.3	79.7
CentripetalNet-104	**37.7**	**63.9**	**68.7**	**48.8**	**71.9**	**84.0**

Table 2. Comparison of AR metric of multi-scale test on MS-COCO test-dev2017.

Instance segmentation On MS-COCO test-dev 2017, as Table 3 shows, our best model achieves 38.8% AP in single-scale test, while Mask R-CNN with ResNeXt-101-FPN achieves 37.5% AP. ExtremeNet can be used for instance segmentation, with another network, DEXTR, which can convert extreme points to instance masks. However, with the same backbone, CentripetalNet achieves 4.2% AP higher than ExtremeNet, and it can be trained end-to-end with the mask prediction module. Compared with the top-ranked methods, our model achieves comparable performance with a MaskAP of 40.2%.

4.3. Ablation study

Centripetal Shift To verify the effectiveness of our proposed centripetal shift, we conduct a series of experiments based on the corner matching methods used in previous corner-based detectors including CornerNet and CenterNet. CornerNet uses associative embedding to match corner pairs. To prove our centripetal shift's effectiveness, we replace the associative embedding of CornerNet with our centripetal shift and use our matching strategy. To be fair, we do not use the cross-star deformable convolution and expand the dimension of associative embedding to 2, the same as our centripetal shift. As shown in Table 4, our method based on centripetal shift brings great performance improvement for CornerNet. As centripetal shift encodes the relationship between corner and center, direct regression to the center should have a similar effect. However, during implementation, it is sometimes impossible to apply the logarithm to the offset between the ground truth corners on heatmap and precise center locations, as the offsets sometimes may be negative because of the rounding operation when mapping the corners from original image to the heatmap. We replace the associative embedding with center regression and find that it also performs much better than CornerNet, but still worse than our centripetal shift as Table 4 shows. CenterNet directly predicts the center heatmap and matches the corners according to the centers and associative embedding. So we add the center prediction module to CornerNet and use the matching strategy of CenterNet, but our method still performs better, especially for large objects.

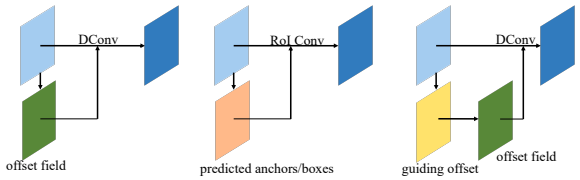
Cross-star Deformable Convolution Our cross-star deformable convolution is a kind of feature adaption method. Feature adaption has recently been studied in anchor-based detectors [35] [5], but our work is the first to discuss the

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
PolarMask [37]	ResNeXt-101	32.9	55.4	33.8	15.5	35.1	46.3
ExtremeNet [42]+DEXTR [26]	Hourglass-104	34.6	54.9	36.6	16.6	36.5	52.0
Mask R-CNN [11]	ResNeXt-101	37.1	60.0	39.4	16.9	39.9	53.5
TensorMask [4]	ResNet-101	37.1	59.3	39.4	17.4	39.1	51.6
MaskLab+ [3]	ResNet-101	37.3	59.8	39.6	19.1	40.5	50.6
MS R-CNN [15]	ResNeXt-101-DCN	39.6	60.7	43.1	18.8	41.5	56.2
HTC [2]	ResNeXt-101	41.2	-	-	-	-	-
PANet(multi-scale) [24]	ResNeXt-101	42.0	65.1	45.7	22.4	44.7	58.1
CentripetalNet(single-scale)	Hourglass-104	38.8	60.4	41.7	19.8	41.3	51.3
CentripetalNet(multi-scale)	Hourglass-104	40.2	62.3	43.1	22.5	42.6	52.1

Table 3. Instance segmentation performance comparison on MS-COCO test-dev.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
associative emb.(1D)	37.3	53.1	39.0	17.8	39.4	50.8
associative emb.(2D)	37.5	53.1	39.7	17.7	39.4	51.2
center prediction	39.9	57.7	42.3	23.1	42.3	52.3
center regression	40.1	55.8	42.7	21.0	42.9	**55.6**
centripetal shift	**40.7**	**58.0**	**42.8**	**22.4**	**43.0**	55.4

Table 4. The effects of centripetal shift(without cross-star deformable convolution and mask head), compared with associative embedding, center regression and center heatmap prediction.



(a) Deformable Convolution (b) RoI Convolution (c) Cross-star Deformable Conv.

Figure 5. Different feature adaption methods. DConv means deformable convolution.

topic for anchor-free detectors. Deformable convolution is usually used for feature adaption, while the main difference between different feature adaption methods is how to obtain the offset field for deformable convolution. Guided anchoring [35] learns the offset field from the predicted anchor shapes to align the feature with different anchor shapes at different locations in the image. AlignDet [5] proposes a more precise feature adaption method, RoI convolution [5], which computes precise sampling locations for deformable convolution as shown in Figure 5(b). To compare RoI convolution with our feature adaption method, we regress the width and height of bounding boxes at the corners, and then we can apply RoI convolution on the feature map from corner pooling. As shown in the Table 5, our method performs better than both the original deformable convolution and RoI convolution. This suggests that our cross-star deformable convolution can refine the feature for better prediction of centripetal shift. AlignDet proves that precise RoI convolution is better than learning offset field from anchor shapes. However, for our model, learning the offset field from the guiding shift performs better than RoI convolution. There are two possible reasons. First, after corner pooling, a lot of information is gathered at the border of the box instead of the inside of the box. As shown in Figure 7, our cross-star deformable convolution tends to sample at the border of the bounding box. So it has better feature extraction ability. Second, the regression of the width and height of the bounding box is not accurate at the corner locations, so the computed sampling points of RoI convolution can not be well aligned with the ground truth.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
no feature adaption	40.7	58.0	42.8	22.4	43.0	55.4
deformable conv.	40.8	58.2	43.2	23.1	42.7	54.9
RoI conv.	41.1	58.5	43.4	22.9	43.4	55.5
cross-star deformable conv.	**41.5**	**58.7**	**44.4**	**23.3**	**44.1**	**55.7**

Table 5. Comparison of different feature adaption methods. Base model is CentripetalNet without feature adaption and mask head, then we add different feature adaption modules separately.

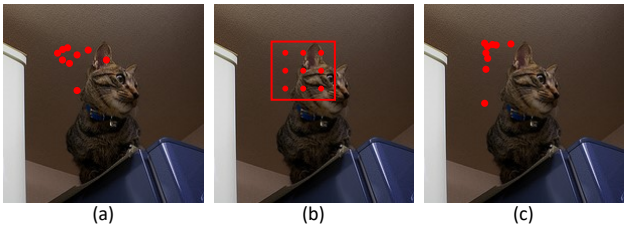


Figure 7. The sampling points of different feature adaption methods. (a) Standard deformable convolution. (b) RoI convolution. (c) Cross-star deformable convolution.

Instance Segmentation Module Many works [11, 8] have proved that the instance segmentation task can improve the performance of anchor-based detectors. Hence we add a mask prediction module as described in section 3.3. As Table 6 shows, multi-task learning improves our model's AP_{bbox} by 0.3%, when training 110 epochs. If we train CentripetalNet with 210 epochs, the improvement becomes 0.4%. We find that mask head does not improve the performance of CornerNet at all. This result shows that this multi-task learning has almost little influence on the corner prediction and associative embedding prediction, but bene-

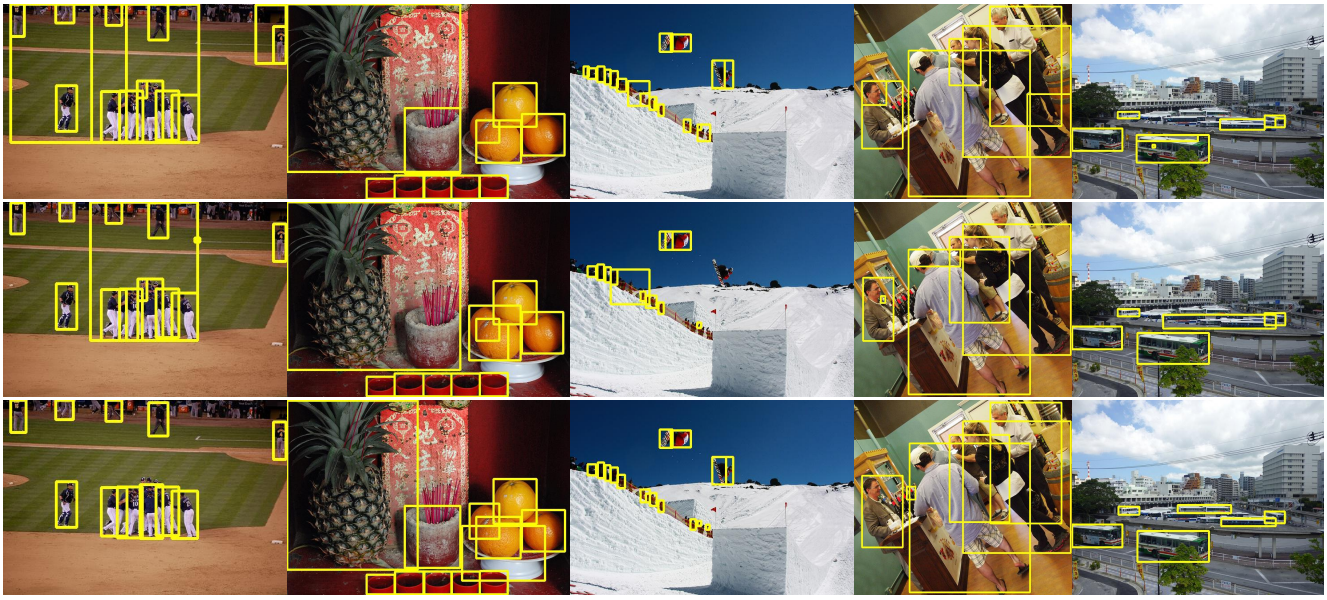


Figure 6. Above three rows show the results of CornerNet, CenterNet and CentripetalNet respectively. CornerNet and CenterNet do not perform well when the similar objects of the same category are highly concentrated. However, CentripetalNet can handle this situation.

	epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CornerNet	110	37.3	53.1	39.0	17.8	39.4	50.8
CornerNet w/mask	110	37.3	53.0	39.5	18.3	39.2	50.7
CentripetalNet w.o/mask	110	41.5	58.7	44.4	23.3	44.1	55.7
CentripetalNet	110	41.8	58.9	44.5	23.0	44.1	56.7
CentripetalNet w.o/mask	210	41.7	**59.0**	44.4	23.3	44.4	56.1
CentripetalNet	210	**42.1**	58.7	**44.9**	**23.7**	**44.5**	**56.8**

Table 6. The effect of mask prediction module on CornerNet and CentripetalNet, both with Hourglass-52 as backbone.

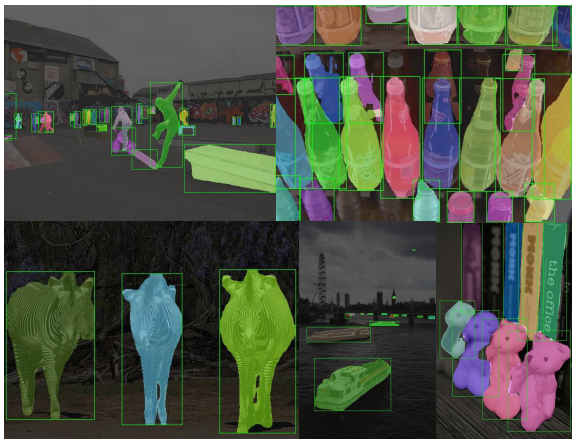


Figure 8. CentripetalNet instance segmentation results.

fits the prediction of our centripetal shift. As shown in Figure 8, CentripetalNet can generate fine segmentation masks.

4.4. Qualitative analysis

As Figure 6 shows, CentripetalNet successfully removes the wrong corner pairs in CornerNet. Compared to CenterNet, CentripetalNet has two advantages. Firstly, CentripetalNet does not rely on center detections, so it can keep the correct predicted bounding boxes, which are incorrectly deleted in CenterNet due to the missed detection of centers. Secondly, CenterNet cannot handle the situations in which the center of an object is in the central region of a box composed of the corners of another two objects. This situation usually occurs in a dense situation, such as the crowd. However, as shown in the last picture in Figure 6, with extreme aspect ratios, the prediction of centripetal offset becomes difficult, so our model may not perform well.

5. Conclusion

In this work, we introduce simple yet effective centripetal shift to solve the corner matching problem in recent anchor-free detectors. Our method establishes the relationship between corners through positional and geometric information and overcomes the ambiguity of associative embedding caused by similar appearance. Besides, we equip our detector with an instance segmentation module and firstly conduct end-to-end instance segmentation using the anchor-free detector. Finally, the state-of-the-art performance on MS-COCO proves the strength of our method.

Acknowledgment This work was supported in part by Sensetime Ltd. Group, the National Science and Technology Major Project of China No. 2018ZX01028-101-001, National Natural Science Foundation of China No.61773307 and No.61621003, the National Key Research and Development Program of China under grant 2016YFB1000902.

References

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

[2] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[3] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *CVPR*, 2018.

[4] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. *arXiv preprint arXiv:1903.12174*, 2019.

[5] Yuntao Chen, Chenxia Han, Naiyan Wang, and Zhaoxiang Zhang. Revisiting feature alignment for one-stage object detection. *arXiv: Computer Vision and Pattern Recognition*, 2019.

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

[7] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. *arXiv preprint arXiv:1904.08189*, 2019.

[8] Chengyang Fu, Mykhailo Shvets, and Alexander C Berg. Retinamask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv: Computer Vision and Pattern Recognition*, 2019.

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

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

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

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

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

[15] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019.

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

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

[18] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. *arXiv preprint arXiv:1901.01892*, 2019.

[19] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. *arXiv preprint arXiv:1909.07072*, 2019.

[20] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. *arXiv preprint arXiv:1912.12898*, 2019.

[21] Tsung Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[22] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):2999–3007, 2017.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[25] 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.

[26] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[28] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017.

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

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

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019.

[32] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[33] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[34] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

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

[36] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[37] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. *arXiv preprint arXiv:1909.13226*, 2019.

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

[39] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017.

[40] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM-MM*, 2016.

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

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