CentripetalNet: Pursuing High-quality Keypoint Pairs for Object Detection

Zhiwei Dong\textsuperscript{1,2}  Guoxuan Li\textsuperscript{3}  Yue Liao\textsuperscript{4}  Fei Wang\textsuperscript{2}  Pengju Ren\textsuperscript{1}  Chen Qian\textsuperscript{2}  
\textsuperscript{1}Institute of Artificial Intelligence and Robotics, Xian Jiaotong University  
\textsuperscript{2}SenseTime Research  
\textsuperscript{3}University of Chinese Academy of Sciences  
\textsuperscript{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 will be available at [https://github.com/KiveeDong/CentripetalNet](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 \cite{10, 9, 29, 26, 23, 19, 11, 20, 1, 17}. The anchor-based methods \cite{9, 29, 23} 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 \cite{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 \cite{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 concentrate on the second step.

The conventional methods \cite{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 a similar embedding, 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 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 a novel CentripetalNet using a corner matching method based on centripetal shift, 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 first 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 the proposed CentripetalNet, we evaluate the method on the challenging MS-COCO benchmark [21]. 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 on MS-COCO test-dev.

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. Generally, anchor-based methods can be divided into two types, namely two-stage methods and single-stage methods.

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 [29] 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 [23] utilizes features maps from multiple different convolution layers to classify and regress anchor boxes with different strides. Compared with YOLO [20], YOLOv2 [27] uses preset anchors. However, the above methods are bothered by the imbalance between negative and positive samples. RetinaNet [20] uses focal loss to mitigate classification imbalance problem. RefineDet [37] refines the FPN structure by introducing the anchor refinement module to filter and eliminate negative samples.

Other works cooperating with anchor-based detectors are proposed to deal with different issues, such as improving anchor selection procedure [33], refining feature learning process [39,18], optimizing location prediction method [24], and improving the loss function [30,16].

Anchor-free Approach: For anchor-free 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 [26] 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 [36] 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 em-
We first generate corner candidates based on the CornerNet cross-star deformable convolution, and instance mask head. 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.

### 3.1. Centripetal Shift Module

**Centripetal Shift.** For \( \text{bbox}_i = (tx^i, ty^i, bx^i, by^i) \), its geometric center is \( (ctx^i, cty^i) = \left( \frac{tx^i + bx^i}{2}, \frac{ty^i + by^i}{2} \right) \). We define the centripetal shifts for its top-left corner and bottom-right corner separately as

\[
\begin{align*}
    cs_{tl}^i &= \left( \log(\frac{ctx^i - tlx^i}{s}), \log(\frac{cty^i - tly^i}{s}) \right) \\
    cs_{br}^i &= \left( \log(\frac{brx^i - ctx^i}{s}), \log(\frac{bry^i - cty^i}{s}) \right)
\end{align*}
\]

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. 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.
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

$$L_{cs} = \frac{1}{N} \sum_{k=1}^{N} \left[ L_1(c_{tl}^k, c_{br}^k) + L_1(c_{tl}^k, c_{br}^k) \right]$$

where $L_1$ 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 $tlx < brx$ and $tly < bry$ to construct predicted bounding boxes. For each bounding box $bbox$, 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.

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

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

$$\begin{align*}
\text{ctlx} &= \frac{tlx + brx}{2} - \frac{brx - tlx}{\mu} \\
\text{ctly} &= \frac{tly + bry}{2} - \frac{bry - tly}{\mu} \\
\text{cbrx} &= \frac{tlx + brx}{2} + \frac{brx - tlx}{\mu} \\
\text{cbry} &= \frac{tly + bry}{2} + \frac{bry - tly}{\mu}
\end{align*}$$

where $0 < \mu \leq 1$ indicates that width and height of central region are $\mu$ times of the bounding box’s width and height. With the centripetal shift, we can decode the center $(\mu tlx, \mu tly)$ and $(brx, bry)$ for top-left corner and bottom-right corner separately.

Then we calculate the score weight $w^j$ for each predicted bounding box that satisfies $(\mu tlx, \mu tly) \in R^j_{central}$ and $(brx, bry) \in R^j_{central}$ as follows

$$w^j = e^{-\frac{|brx - cbrx| + |bry - cbry|}{(cbrx - ctlx)^2 + (cbry - cly)^2}}$$

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 star’, not only a large receptive field is required, but also the geometric structure of ‘cross star’ should be learned. Following the above intuition, we proposed the cross-star deformable convolution, a novel convolution operation to enhance the visual features at corners.

![Figure 4](image-url)
in Figure 3(b), which contains both shape and direction information, to the offset field branch. 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:

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

where $\delta$ means the guiding shift and is defined as

$$\delta_{tl} = (\frac{ctx^i}{s} - [\frac{tx^i}{s}], \frac{cty^i}{s} - [\frac{ty^i}{s}])$$  \hspace{1cm} (7)

The second convolution layer maps the above feature into the offset field, which contains the context and geometric information explicitly. 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

To get the instance segmentation mask, 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 $14 \times 14$ and predict a mask of $28 \times 28$.

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

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

$$L = L_{\text{det}} + L_{\text{off}} + \alpha L_{\delta} + L_{\text{cs}} + L_{\text{mask}}$$  \hspace{1cm} (9)

where $L_{\text{det}}$ and $L_{\text{off}}$ are defined as CornerNet. We set $\alpha$ to 0.05, as we find that large $\alpha$ 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 $L_{\text{mask}}$.

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 $511 \times 511$, 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 $3 \times 3$ 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 $\mu = \frac{1}{21}$ for those bounding boxes with an area larger than $3500$, and $\mu = \frac{1}{24}$ 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 [13].

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} | AR_{S} | AR_{M} | AR_{L} |
|-------------------------------|---------------------|-----|---------|---------|--------|--------|--------|
| **Two-stage:**                |                     |     |         |         |        |        |        |
| Faster R-CNN w/FPN [19]      | ResNet-101          | 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) [22]     | 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 [23]                  | ResNet-101          | 31.2| 50.4    | 33.3    | 10.2   | 34.5   | 49.8   |
| YOLOv3 [28]                  | DarkNet-53          | 33.0| 57.9    | 34.4    | 18.3   | 35.4   | 41.9   |
| RetinaNet800 [20]            | ResNet-101          | 39.1| 59.1    | 42.3    | 21.8   | 42.7   | 50.2   |
| **Single-stage anchor-free:**|                     |     |         |         |        |        |        |
| ExtremeNet (single-scale) [38]| 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) [38]| 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) [35]    | ResNet-101-DCN      | 45.0| 66.1    | 49.0    | 26.6   | 48.6   | 57.5   |
| RPDet (multi-scale) [35]     | 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   |

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_{S} | AR_{M} | AR_{L} | AR_{T} | AR_{100} | AR_{200} | AR_{500} |
|-------------------------------|--------|--------|--------|--------|----------|----------|----------|
| 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**| **84.8**|          |

Table 2. Comparison of AR metric of multi-scale test on MS-COCO test-dev 2017.

**Instance segmentation** We also report CentripetalNet’s instance segmentation performance on MS-COCO test-dev 2017 for the comparison with state-of-the-art methods. 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 de-
Table 3. Instance segmentation performance comparison on MS-COCO test-dev.

| Method                        | Backbone       | AP  | AP50 | AP75 | APs | APm | APl |
|-------------------------------|----------------|-----|------|------|-----|-----|-----|
| PolarMask [34]                | ResNeXt-101    | 32.9| 55.4 | 33.8 | 15.5| 35.1| 46.3|
| ExtremeNet [38]+DEXTR [25]   | 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 [13]                | ResNeXt-101-DCN| 39.6| 60.7 | 43.1 | 18.8| 41.5| 56.2|
| HTC [2]                      | ResNeXt-101    | 41.2| -    | -    | -   | -   | -   |
| PANet(multi-scale) [22]      | ResNeXt-101    | 42.0| 65.1 | 45.7 | 22.4| 44.7| 58.1|

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.

|                  | AP  | AP50 | AP75 | APs | APm | APl |
|------------------|-----|------|------|-----|-----|-----|
| 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 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.

|                  | AP  | AP50 | AP75 | APs | APm | APl |
|------------------|-----|------|------|-----|-----|-----|
| 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|

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 APbbox by 0.3%, when training 110 epochs. If we train CentripetalNet with 210 epochs, the improvement becomes more significant.
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.

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 benefits 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.

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.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In CVPR, 2018.
[38] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In CVPR, 2019.

[39] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. arXiv preprint arXiv:1903.00621, 2019.