Point-Set Anchors for Object Detection, Instance Segmentation and Pose Estimation

Fangyun Wei[1⋆], Xiao Sun[1⋆], Hongyang Li[2], Jingdong Wang[1], and Stephen Lin[1]

[1] Microsoft Research Asia
{fawe, xias, jingdw, stevelin}@microsoft.com
[2] Peking University
lhy_ustb@pku.edu.cn

Abstract. A recent approach for object detection and human pose estimation is to regress bounding boxes or human keypoints from a central point on the object or person. While this center-point regression is simple and efficient, we argue that the image features extracted at a central point contain limited information for predicting distant keypoints or bounding box boundaries, due to object deformation and scale/orientation variation. To facilitate inference, we propose to instead perform regression from a set of points placed at more advantageous positions. This point set is arranged to reflect a good initialization for the given task, such as modes in the training data for pose estimation, which lie closer to the ground truth than the central point and provide more informative features for regression. As the utility of a point set depends on how well its scale, aspect ratio and rotation matches the target, we adopt the anchor box technique of sampling these transformations to generate additional point-set candidates. We apply this proposed framework, called Point-Set Anchors, to object detection, instance segmentation, and human pose estimation. Our results show that this general-purpose approach can achieve performance competitive with state-of-the-art methods for each of these tasks.

Keywords: Object detection, instance segmentation, human pose estimation, anchor box, point-based representation

1 Introduction

A basic yet effective approach for object localization is to estimate keypoints. This has been performed for object detection by detecting points that can define a bounding box, e.g., corner points [20], and then grouping them together. An even simpler version of this approach that does not require grouping is to extract the center point of an object and regress the bounding box size from it. This method, called CenterNet [43], can be easily applied to human pose estimation as well, by regressing the offsets of keypoints instead.

[⋆] Equal contribution.

While CenterNet is highly practical and potentially has broad application, its regression of keypoints from features at the center point can be considered an important drawback. Since keypoints might not lie in proximity of the center point, the features extracted at the center may provide little information for inferring their positions. This problem is exacerbated by the geometric variations an object can exhibit, including scale, orientation, and deformations, which make keypoint prediction even more challenging.

In this paper, we propose to address this issue by acquiring more informative features for keypoint regression. Rather than extract them at the center point, our approach is to obtain features at a set of points that are likely to lie closer to the regression targets. This point set is determined according to task. For instance segmentation, the points are placed along the edges of an implicit bounding box. For pose estimation, the arrangement of points follows modes in the pose distribution of the training data, such as that in Fig. 1 (b). As a good task-specific initialization, the point set can yield features that better facilitate keypoint localization.

It can be noted that a point set best serves its purpose when it is aligned in scale and aspect ratio with the target. To accomplish this, we adapt the anchor box scheme commonly used in object detection by expressing point sets as *point-set anchors*. Like their anchor box counterparts, point-set anchors are sampled at multiple scales, aspect ratios, and image positions. In addition, different point-set configurations may be enumerated, such as different modes in a pose estimation training set. With the generated point-set candidates, keypoint regression is conducted to find solutions for the given task.

The main contributions of this work can be summarized as:

- A new object representation named *Point-Set Anchors*, which can be seen as a generalization and extension of classical box anchors. Point-set anchors can further provide informative features and better task-specific initializations for shape regression.
- A network based on point-set anchors called PointSetNet, which is a modification of RetinaNet [23] that simply replaces the anchor boxes with the proposed point-set anchors and also attaches a parallel regression branch. Variants of this network are applied to object detection, human pose estimation, and also instance segmentation, for which the problem of defining specific regression targets is addressed.

It is shown that the proposed general-purpose approach achieves performance competitive with state-of-the-art methods on object detection, instance segmentation and pose estimation.

2 Related Work

Object representations. In object detection, rectangular anchors [34,23,22] are the most common representation used in locating objects. These anchors serve as initial bounding boxes, and an encoding is learned to refine the object

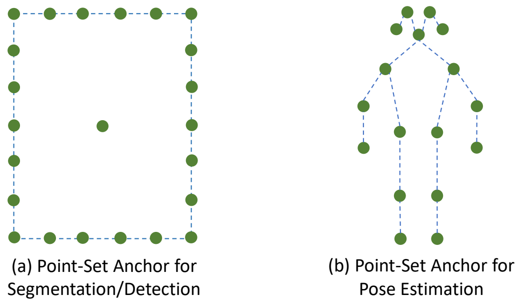
(a) Point-Set Anchor for Segmentation/Detection
(b) Point-Set Anchor for Pose Estimation

Fig. 1. Illustration of our point-set anchors for instance segmentation, object detection and pose estimation. Instance mask point-set anchors contain an implicit bounding box and n anchor points are uniformly sampled from the corresponding bounding box. Pose point-set anchors are initialized as the most frequent poses in the training set.

localization or to provide intermediate proposals for top-down solutions [15,8]. However, the anchor box is a coarse representation that is insufficient for finer degrees of localization required in tasks such as instance segmentation and pose estimation. An alternative is to represent objects in terms of specific points, including center points [38,43], corner points [20,12], extreme points [44], octagon points [32], point sets [41,42], and radial points [40]. These point representations are designed to solve one or two tasks among object detection, instance segmentation and pose estimation. Polygon point based methods, such as corner points [20,12] and extreme points [44], are hard to apply to instance segmentation and pose estimation due to their restricted shape. While center point representations [38,43] are more flexible, as offsets from the object center to the corresponding bounding box corners or human joints can be directly predicted, we argue that features extracted from center locations are not as informative as from our task-specific point sets, illustrated in Fig. 1. In addition, how to define regression targets for instance segmentation is unclear for these representations. Another way to perform instance segmentation from a center point is by regressing mask boundary points in radial directions [40]; however, radial regressions at equal angular intervals are unsuitable for pose estimation. Other representations such as octagon points or point sets [41,42] are specifically designed for one or two recognition tasks. The proposed point-set anchors combine benefits from anchor boxes and point representations, and its flexibility makes them applicable to object detection, instance segmentation and pose estimation.

Instance segmentation. Two-stage methods [15,21,17] formulate instance segmentation in a 'Detect and Segment' paradigm, which detects bounding boxes and then performs instance segmentation inside the boxes. Recently, there is much research focused on single-stage instance segmentation since two-stage methods are often slow in practice. PolarMask [40] uses a polar representation and n rays at equal angular intervals are emitted from the polar center for dense distance regression. YOLACT [1] generates a set of prototype masks and predicts per-instance mask coefficients for linearly combining the prototype masks.

ExtremeNet [44] detects four extreme points and a center point using a standard keypoint estimation network, which has the disadvantage of a long training time, and then grouping is applied to generate coarse octagonal masks. The recent Deep Snake [32] proposes a two-stage pipeline based on initial contours and contour deformation. Our method is different from Deep Snake in three ways. First, our method for instance segmentation operates in a single stage, without needing proposals generated by detectors. Second, Point-Set Anchors perform mask shape regression directly, in contrast to the iterative deformation in Deep Snake. Finally, our method is evaluated on the challenging MS COCO dataset [24] and is compared to state-of-the-art methods in object detection, instance segmentation and pose estimation.

Pose estimation. In pose estimation algorithms, most previous works follow the paradigm of estimating a heat map for each joint [2,3,5,6,7,14,15,33,18,28,39]. The heat map represents the confidence of a joint existing at each position in the image. Despite its good performance, a heat map representation has a few drawbacks such as no end-to-end training, a need for high resolution [37], and separate steps for joint localization and association. The joint association problem is typically addressed by an early stage of human bounding box detection [39,15] or a post-processing step that groups the detected joints together with additionally learned joint relations [3,27]. Recently, a different approach has been explored of directly regressing the offsets of joints from a center or root point [43]. In contrast to the heat map based approaches, the joint association problem is naturally addressed by conducting a holistic regression of all joints from a single point. However, the holistic shape regression is generally more difficult than part based heat map learning due to optimization on a large, high-dimensional vector space. Nie et al. [29] address this problem by factorizing the long-range displacement with respect to the center position into accumulative shorter ones based on the articulated kinematics of human poses. They argue that modeling short-range displacements can alleviate the learning difficulty of mapping from an image representation to the vector domain. We follow the regression based paradigm and propose to address the long-range displacement problem by regressing from a set of points placed at more advantageous positions.

3 Our Method

In this section, we first formulate the proposed task-specific point-set anchors. Next, we show how to make use of these point-set anchors for regression-based instance segmentation and pose estimation. Finally, we introduce our PointSet-Net, which is an extension of RetinaNet with a parallel branch attached for keypoint regression.

3.1 Point-Set Anchors

The point-set anchor $\mathbf{T}$ contains a number of ordered points that are defined according to the specific task. We describe the definitions for the tasks of human pose estimation and instance segmentation that we will handle.

Pose point-set anchor. We naturally use the human keypoints to form the pose point-set anchor. For example, in the COCO [24] dataset, there are 17 keypoints, and the pose point-set anchor is represented by a 34-dimensional vector. At each image position, we use several point-set anchors. We initialize the point-set anchors as the most frequent poses in the training set. We use a standard k-means clustering algorithm [26] to partition all the training poses into k clusters, and the mean pose of each cluster is used to form a point-set anchor. Fig. 1(b) illustrates one of the point-set anchors for pose estimation.

Instance mask point-set anchor. This anchor has two parts: one center point and n ordered anchor points which are uniformly sampled from an implicit bounding box as illustrated in Fig. 1(a). n is a hyper-parameter to control sampling density. Corner points in mask point-set anchor can be also served as a reference for object detection. At each image position, we form 9 point-set anchors by varying the scale and aspect ratios of the implicit bounding box.

3.2 Shape Regression

We treat instance segmentation, object detection and pose estimation as a shape regression problem. We represent the object by a shape $\mathbf{S}$, i.e., a set of n_s ordered points $\mathbf{S} = \{S_i\}_{i=1}^{n_s}$, where S_i represents the i-th *polygon vertex* for instance mask, i-th *corner vertex* for bounding box, or the i-th *keypoint* for pose estimation. Instead of regressing the shape point locations from the object center, we employ $\mathbf{T}$ as a reference for shape regression. Our goal is to regress the offsets $\Delta\mathbf{T}$ from the point-set anchor $\mathbf{T}$ to the shape $\mathbf{S}$.

Offsets for pose estimation. Each keypoint in the human pose estimation task represents a joint with semantic meaning, e.g., head, right elbow or left knee. We use 17 keypoints as the shape $\mathbf{S}$ and the offsets are simply the difference: $\Delta\mathbf{T} = \mathbf{S} - \mathbf{T}$.

Offsets for instance segmentation. The shape $\mathbf{S}$ of an instance mask also contains a set of ordered points, and the number of points might be different for different object instances. The point-set anchor $\mathbf{T}$ is defined for all instances and contains a fixed number of points. To compute the offsets $\Delta\mathbf{T}$, we find the matching points $\mathbf{T}^*$, each of which has a one-to-one correspondence to each point in $\mathbf{T}$, from the shape $\mathbf{S}$, and then $\Delta\mathbf{T} = \mathbf{T}^* - \mathbf{T}$. The matching strategies, illustrated in Fig. 2, are described as follows.

- **Nearest point.** The matching target of each point in the point-set anchor $\mathbf{T}$ is defined as the nearest polygon point in $\mathbf{S}$ based on L_1 distance. Thus, a single polygon point may be assigned to several anchor points, one anchor point, or none of them.
- **Nearest line.** We treat the mask contour $\mathbf{S}$ as a sequence of n_s line segments instead of n_s discrete polygon vertices. Each of the anchor points is projected to all the polygon edges, and the closest projection point is assigned to the corresponding anchor point.

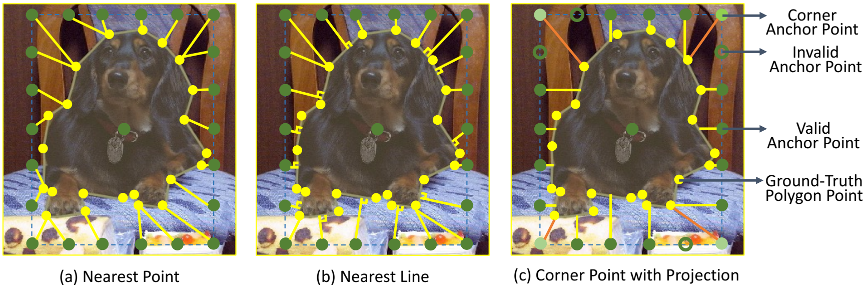

Fig. 2. Illustration of three matching strategies between point-set anchor and the ground-truth mask contour for instance segmentation. Yellow solid dots represent polygon points of the ground-truth mask. Green solid dots and green hollow dots denote valid and invalid anchor points, respectively. Orange and yellow lines indicate correspondences for corner and non-corner anchor points, respectively. Only valid anchor points are considered for training and inference.

- **Corner point with projection.** We first find the targets of the four corner anchor points by the Nearest Point strategy, and then the targets of these corner points are used to subdivide the mask contour into four parts, i.e., top, right, bottom and left parts, that are to be matched with anchor points on the corresponding side. For each of the four parts, the target of each anchor point is the nearest intersection point between the horizontal (for left and right parts) or vertical (for top and bottom parts) projection line and the line segments of the mask contour. If a matched point lies outside of the corresponding contour segment delimited by the matched corner points, we mark it as invalid and it is ignored in training and testing. The remaining anchor points and their matches are marked as valid for mask regression learning.

Offsets for object detection. The bounding box shape **S** in object detection can be denoted as two keypoints, i.e., the top-left and bottom-right point. The offsets $\varDelta\mathbf{T}$ are the distance from the target points to the corresponding corner points in mask point-set anchors.

Positive and negative samples. In assigning positive or negative labels to point-set anchors, we directly employ IoU for object detection and instance segmentation, and Object Keypoint Similarity (OKS) for pose estimation. Formally, in instance segmentation and object detection, a point-set anchor is assigned a positive label if has the highest IoU for a given ground-truth box or an IoU over 0.6 with any ground-truth box, and a negative label if it has IoU lower than 0.4 for all ground-truth boxes. In practice, we use the IoU between the implicit bounding box of the mask point-set anchors and ground-truth bounding boxes instead of masks in instance segmentation, to reduce computation cost. For human pose estimation, a point-set anchor is assigned a positive label if it has the highest OKS for a given ground-truth pose or an OKS over 0.5 with any ground-

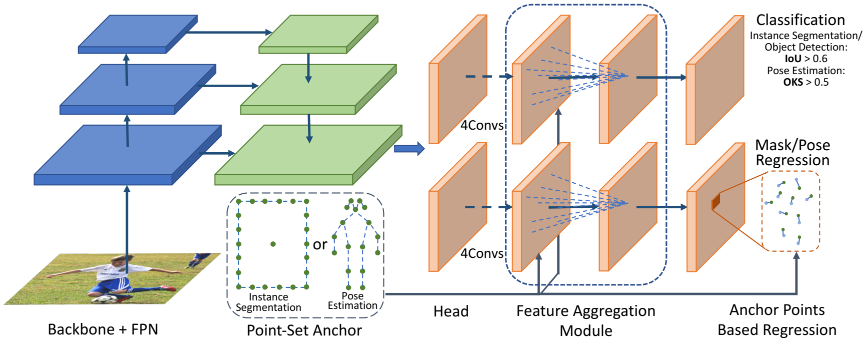

Fig. 3. Network architecture. The left part represents the backbone and feature pyramid network to extract features from different levels. The right part shows the shared heads for classification and mask/pose estimation with point-set anchors. We omit the bounding box regression branch for clearer illustration.

truth pose, and a negative label if it has OKS lower than 0.4 for all ground-truth poses.

Mask construction. In instance segmentation, a mask is constructed from the regressed anchor points as a post-processing step during inference. For the matching methods Nearest Point and Nearest Line, we choose an arbitrary point as the origin and connect adjacent points sequentially. For Corner Point with Projection, a mask is similarly constructed from only the valid points.

3.3 PointSetNet

Architecture. PointSetNet is an intuitive and natural extension of RetinaNet [23]. Conceptually, it simply replaces the classical rectangular anchor with the proposed point-set anchor, and attaches a parallel regression branch for instance segmentation or pose estimation in the head. Fig. 3 illustrates its network structure. Following RetinaNet, we use a multi-scale feature pyramid for detecting objects at different scales. Specifically, we make use of five levels of feature maps, denoted as $\{P_3, P_4, P_5, P_6, P_7\}$. P_3, P_4 and P_5 are generated by backbone feature maps C_3, C_4 and C_5 followed by a 1×1 convolutional layer and lateral connections as in FPN [22]. P_6 and P_7 are generated by a 3×3 stride-2 convolutional layer on C_5 and P_6, respectively. As a result, the stride of $\{P_3, P_4, P_5, P_6, P_7\}$ is $\{8, 16, 32, 64, 128, 256\}$. The head contains several subnetworks for classification, mask or pose regression, and bounding box regression. Each subnetwork contains four 3×3 stride-1 convolutional layers, a feature aggregation layer which is used only for the pose estimation task, and an output layer. Table 1 lists the output dimensions from the three subnetworks for instance segmentation and pose estimation. Following [22,23], we also share the head among $P3 - P7$.

Point-set anchor density. One of the most important design factors in anchor based detection frameworks [34,22,23,25] is how densely the space of possible anchors is sampled. For instance segmentation, following [23], we simply replace

Table 1. Output dimensions from different subnetworks for instance segmentation and pose estimation. K, n and C denote the number of point-set anchors, points number in each of the point-set anchors, and class number for the training/testing dataset, respectively.

Task	Classification	Shape Regression	Bounding Box Regression
Instance Segmentation	$K \times C$	$K \times (n \times 2)$	$K \times 4$
Pose Estimation	$K \times 2$	$K \times (17 \times 2)$	-

classical rectangular anchors by our mask point-set anchors, and use 3 scales and 3 aspect ratios per location on each of the feature maps. Specifically, we make use of the implicit bounding box in point-set anchors, where each bounding box has three octave scales $2^{k/3}(k \leq 3)$ and three aspect ratios $[0.5, 1, 2]$, and the base scale for feature maps $P3 - P7$ is $\{32, 64, 128, 256, 512\}$. The combination of octave scales, aspect ratios and base scales will generate 9 bounding boxes per location on each of the feature maps. Anchor points are uniformly sampled on the four sides of the generated bounding boxes. For pose estimation, we use 3 mean poses generated by the k-means clustering algorithm. Then we translate them to each position of the feature maps as the point-set anchors. We further use 3 scales and 3 rotations for each anchor, yielding 27 anchors per location. The other feature map settings are the same as in instance segmentation.

Loss function. We define our training loss function as follows:

$$L = \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) + \frac{\lambda}{N_{pos}} \sum_{x,y} \mathbb{1}_{\{c^*_{x,y}\}>0} L_{reg}(t_{x,y}, t^*_{x,y}), \qquad (1)$$

where L_{cls} is the Focal loss in [23] and L_{reg} is the L1 loss for shape regression. $c^*_{x,y}$ and $t^*_{x,y}$ represent classification and regression targets, respectively. N_{pos} denotes the number of positive samples and λ is the balance weight, which is set to 0.1 and 10.0 for instance segmentation and pose estimation, respectively. $\mathbb{1}_{\{c^*_{x,y}\}>0}$ denotes the indicator function, being 1 if $\{c^*_{x,y}\} > 0$ and 0 otherwise. The loss is calculated over all locations and all feature maps.

Elements specific to pose estimation. Besides target normalization and the embedding of prior knowledge in the anchor shapes, we further show how feature aggregation with point-set anchors achieves a certain feature transformation invariance and how point-set anchors can be extended to multi-stage learning.

– **Deep shape indexed features.** Learning of shape/transformation invariant features has been a fundamental problem in computer vision [36,11], as they provide consistent and robust image information that is independent of geometric configuration. A point-set anchor acts as a shape hypothesis of the object to be localized. Though it reflects a coarse estimate of the target object shape, it still achieves a certain feature invariance to object shape, as it extracts the feature in accordance with the ordered point-set. The blue dashed rectangle in Fig. 3 depicts the feature aggregation module. In principle, the closer the anchor points are to the object shape, the better shape invariance of the feature. The deep

Table 2. Comparison of different matching strategies between anchor points and mask contours on the instance segmentation task.

Matching strategy	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Nearest Point	21.9	42.1	20.6	11.4	24.6	29.8
Nearest Line	23.2	46.5	21.0	12.5	26.2	32.0
Corner Point with Projection	**27.0**	**49.1**	**26.6**	**13.8**	**30.6**	**36.7**

shape indexed feature is implemented by DCN [9]. Specifically, we replace the learnable offset in DCN with the location of points in a point-set anchor.

– **Multi-stage refinement.** Holistic shape regression is generally more difficult than part based heat map learning [37]. This is on one hand because of the large and continuous solution space of poses. On the other hand, it is due to the extremely unbalanced transformation variance between different keypoints. To address this, a classic paradigm is to estimate the pose progressively via a sequence of weak regressors where each weak regressor uses features that depend on the estimated pose from the previous stage [11,36].

To this end, we use an additional refinement stage for pose estimation. While the k mean poses in the training set are used as the initial anchors for the first stage, we can directly use the pose predictions of the first stage as the point-set anchors for the second stage. Since the joint positions in the point-set anchors are well-initialized in the first stage, the point-set anchors for the second stage are much closer to the ground truth shapes. This facilitates learning since the distance of the regression target becomes much smaller and better shape-invariant features can be extracted by using the more accurate anchor shapes.

Conceptually, this head network can be stacked for multi-stage refinement, but we find the use of a single refinement to be most effective. Hence, we use one-step refinement for simplicity and efficiency.

4 Experiments

4.1 Instance Segmentation Settings

Dataset. We present experimental results for instance segmentation and object detection on the MS-COCO [24] benchmark. We use COCO `trainval35k` (115k images) for training and the `minival` split (5k images) for ablations. Comparisons to the state-of-the-art are reported on the `test-dev` split (20k images).

Training details. All our ablation experiments, except when specifically noted, are conducted on the `minival` split with ResNet-50 [16] and FPN. Our network is trained with synchronized stochastic gradient descent (SGD) over 4 GPUs with a mini-batch of 16 images (4 images per GPU). We adopt the 1× training setting, with 12 epochs in total and the learning rate initialized to 0.01 and then divided by 10 at epochs 8 and 11. The weight decay and momentum parameters are set to 10^{-4} and 0.9, respectively. We initialize our backbone network with the weights pre-trained on ImageNet [10]. For the newly added layers, we keep

Table 3. Comparison on different numbers of anchor points.

n	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
24	26.7	48.8	26.1	13.6	30.2	36.4
36	27.0	49.1	26.6	13.8	30.6	36.7
48	27.2	49.2	26.8	13.9	30.7	36.7
60	28.0	**49.8**	27.9	13.9	31.4	38.6
72	**28.0**	49.6	**27.9**	**14.6**	**31.5**	**38.8**

Table 4. Comparison of two regression origins.

Origin	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Center Point	26.0	48.4	24.8	13.6	29.3	35.4
Anchor Points	**27.0**	**49.1**	**26.6**	**13.8**	**30.6**	**36.7**

the same initialization as in [23]. Unless specified, the input images are resized to have a shorter side of 800 and a longer side less than or equal to 1333. Samples with IoU higher than 0.6 and lower than 0.4 are defined as positive samples and negative samples, respectively. We use the Corner Point with Projection matching strategy and set the number of anchor points to 36 by default.

Inference details. We forward the input image through the network and obtain the classification scores and the corresponding predicted classes. According to the classification scores, the top-1k anchors from each level of the feature maps are sent for mask construction. Then the top predictions from all levels are merged and non-maximum suppression (NMS)[3] with a threshold of 0.5 is employed as post-processing.

4.2 Experiments on Instance Segmentation

Mask matching strategies. First, we compare the results from the three matching strategies between point-set anchor and the corresponding mask contour as shown in Table 2. Nearest Point has the worst performance, perhaps because each polygon point may be assigned to multiple anchor points and this inconsistency may misguide training. Both Nearest Line and Corner Point with Projection treat the ground-truth mask as a whole contour instead of discrete polygon points in the mask matching step. However, there still exist ambiguous anchor points for the Nearest Line method as shown in Fig. 2(b). Corner Point with Projection eliminates inconsistency, as the subdivision of the mask contour into segments leads to better-defined matches, and it achieves the best performance among the three methods.

Effect of point-set anchors. The number of anchor points can greatly affect instance segmentation. From Table 3, it can be seen that the more accurate representation from more anchor points leads to better performance. Also, the

[3] IoU is calculated by predicted bounding boxes and ground-truth rectangles due to the high computational cost of mask IoU. If there is no bounding box branch in the network, we use the smallest rectangle that encompasses the predicted mask instead.

Table 5. Results on the MS COCO `test-dev` compared to state-of-the-art **instance segmentation** and **object detection** methods. '$*$' denotes the multi-scale testing.

Method	Backbone	Regression Based	Segmentation			Detection		
			AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Mask RCNN [15]	ResNeXt-101	✗	37.1	60.0	39.4	39.8	62.3	43.4
TensorMask [4]	ResNet-101	✗	37.1	59.3	39.4	-	-	-
FCIS [21]	ResNet-101	✗	29.2	49.5	-	-	-	-
YOLACT [1]	ResNet-101	✗	31.2	50.6	32.8	33.7	54.3	35.9
ExtremeNet*[44]	Hourglass-104	✗	18.9	44.5	13.7	43.7	60.5	47.0
CornerNet* [20]	Hourglass-104	✗	-	-	-	42.2	57.8	45.2
RetinaNet [23]	ResNext-101	✓	-	-	-	40.8	61.1	44.1
FCOS [38]	ResNext-101	✓	-	-	-	42.1	62.1	45.2
CenterNet*[43]	Hourglass-104	✓	-	-	-	45.1	63.9	49.3
RepPoints*[41]	ResNeXt-101-DCN	✓	-	-	-	46.5	67.4	50.9
PolarMask [40]	ResNeXt-101-DCN	✓	36.2	59.4	37.7	-	-	-
Dense RepPoints [42]	ResNeXt-101-DCN	✓	33.7	59.9	34.3	45.8	66.7	49.6
Ours	ResNeXt-101-DCN	✓	34.6	60.1	34.9	45.1	66.1	48.9
Ours*	ResNeXt-101-DCN	✓	36.0	61.5	36.6	46.2	67.0	50.5

performance is seen to saturate beyond a certain number of anchor points, e.g., 72 points. We also demonstrate the benefits of using point-set anchors as the regression origin as shown in Table 4. It can be seen that with the same regression targets, using point-set anchors as the regression origin outperforms center point based regression by 1.0 AP.

Comparison with state-of-the-art methods. We evaluate PointSetNet on the COCO `test-dev` split and compare with other state-of-the-art object detection and instance segmentation methods. We use 60 anchor points for contour representation and ResNext-101 with DCN [9] as backbone. For data augmentation, we randomly scale the shorter side of images in the range of 480 to 960 during training and increase the number of training epochs to 24 (2× training setting). An image pyramid with a shorter side of $\{400, 600, 800, 100, 1200\}$ is applied during the inference. Table 5 reports the results. On both object detection and instance segmentation, PointSetNet achieves performance competitive with the state-of-the-art algorithms. The gap between TensorMask and PointSetNet arises from the tensor bipyramid head which brings a +5.1 AP improvement. We do not plug this into our framework due to its heavy time and memory cost. Although the AP is 0.2 lower than PolarMask on instance segmentation, our proposed point-set anchor has the benefit of being applicable to human pose estimation.

4.3 Pose Estimation Settings

Dataset. We conduct comprehensive comparison experiments on the MS COCO Keypoint dataset [24] to evaluate the effectiveness of point-set anchors on multi-person human pose estimation. The COCO train, validation, and test sets contain more than 200k images and 250k person instances labeled with keypoints. 150k of the instances are publicly available for training and validation. Our models are trained only on the COCO `train2017` dataset (which includes 57K

Table 6. Comparing different point-set anchors. Better results are achieved when anchors *efficiently* cover more ground truth shapes with relatively large positive/negative sample ratio.

Anchor Type	Matched GT(%)	Pos/Neg(‱)	AP	AP_{50}	AP_{75}
Center Point	0	4.5	16.9	48.7	6.0
Rectangle	4.0	7.7	12.0	33.9	5.9
Mean Pose	18.0	6.6	**40.9**	**69.4**	**42.0**
K-means_3	27.7	6.4	43.3	70.7	45.5
K-means_5	32.1	6.1	**43.8**	**71.3**	**46.6**
K-means_7	34.6	6.0	43.8	71.2	46.5
Scale (0.8:0.2:1.2)	25.2	6.5	42.6	69.8	44.9
Scale (0.6:0.2:1.4)	27.6	6.2	**42.6**	**69.7**	**44.5**
Scale (0.4:0.2:1.6)	29.6	5.9	39.4	65.4	41.2
Rotation (-10:10:10)	27.2	9.0	42.6	70.6	44.2
Rotation (-20:10:20)	36.0	6.6	**43.5**	**71.6**	**45.9**
Rotation (-30:10:30)	40.5	6.2	42.6	70.6	44.4

images and 150K person instances) with no extra data. Ablations are conducted on the `val2017` set, and final results are reported on the `test-dev2017` set for a fair comparison to published state-of-the-art results [3,13,15,27,30,31,43,29].

Training details. For pose estimation, we use the Adam [19] optimizer for training. The base learning rate is 1e-4. It drops to 1e-5 at 80 epochs and 1e-6 at 90. There are 100 epochs in total. Samples with OKS higher than 0.5 and lower than 0.4 are defined as positive samples and negative samples, respectively. The other training details are the same as for instance segmentation including the backbone network, network initialization, batch size and image resolution.

4.4 Experiments on Pose Estimation

Effect of point-set anchors. We first compare the proposed point-set anchors with strong prior knowledge of pose shapes to other point-based anchors like the center point and points on a rectangle. We denote as *mean pose* the use of the average pose in the training set as the canonical shape. Then we translate this *mean pose* to every position in the image as the point-set anchors for pose estimation. No other transformation is used to augment the anchor distribution for a fair comparison to the *center point* and *rectangle* anchors. A ground truth pose is assigned to anchors with OKS higher than 0.5. If no anchor is found higher than OKS 0.5, the ground truth is assigned to the closest anchor.

In Table 6, it is shown that the mean pose anchor outperforms the *center point* and *rectangle* anchors by a large margin with more ground truth poses assigned with OKS greater than 0.5. Specifically, it surpasses *center point* anchors by 24 AP and *rectangle* anchors by 28.9 AP. This indicates that an anchor that better approximates the target shape is more effective for shape regression.

We obtain further improvements by using additional canonical pose shapes generated by the K-Means clustering algorithm or by augmenting the *mean pose*

Table 7. Comparing the deep shape indexed feature with other feature extraction methods. Deep-SIF-n denotes deep shape indexed feature with n points used for feature extraction.

Feature Types	Loss_cls	Loss_reg	AP	AP_{50}	AP_{75}
Center Feature	0.31	5.92	40.9	69.4	42.0
Box Corner Feature	0.31	6.22	42.6	68.2	45.6
Box Region Feature	0.30	5.99	42.8	69.0	46.0
Deep-SIF-9 for Cls	0.30	6.29	41.6	68.2	44.0
Deep-SIF-9 for Reg	0.32	5.93	45.5	69.6	49.1
Deep-SIF-9 for Cls & Reg	0.29	5.92	46.0	71.7	49.1
Deep-SIF-25 for Cls & Reg	**0.29**	**5.79**	**47.5**	**72.0**	**51.6**

Table 8. Effect of multi-stage refinement.

Stage	OKS	Matched GT(%)	Pos/Neg(‱)	Loss_cls	Loss_reg	AP	AP_{50}	AP_{75}
Stage-1	0.5	45.7	12.7	0.25	5.52	48.3	74.3	52.5
Stage-2	0.99	81.5	11.5	0.23	4.28	58.0	80.8	62.4

shape with additional sampling of rotation and scaling transformations. However, more anchor shapes and transformation augmentations also introduce more negative anchors that are not assigned to any of the ground truth poses, which makes learning less efficient. Better performance can be attained with a better trade-off between covering more ground truth shapes and incurring fewer negative samples. Empirically, we achieve the best performance by using 5 canonical pose shapes (+2.9AP), 5 scale transformations (+1.7AP) and 5 rotation transformations (+2.6AP).

Effect of deep shape indexed feature. Table 7 compares the deep shape indexed feature with other feature extraction methods. First, three regular feature extraction methods, namely from the center point, 4 corner points of the bounding box and 9 grid points in the bounding box region are compared. With more feature extraction points, slightly better performance is obtained.

Then, we use the deep shape indexed feature which uses the point set in the anchors for feature extraction. A point set based on pose shape priors extracts more informative features that greatly improve learning and performance. Specifically, with the same number of points (part of the 17 joints) as in the box region feature (i.e., 9), the deep shape indexed feature improves the AP from 42.8 to 46.0 (+3.2 AP, relative 7.5% improvement). Further improvement can be obtained by using more joint points for feature extraction. Note that if the shape indexed feature is used only for the person classification sub-network, the improvement is not as great as using it only in the pose regression sub-network. This indicates that it mainly enhances pose regression learning.

Effect of multi-stage refinement. Table 8 shows the result of using a second stage for refinement. Since the anchors for the second stage are much closer to the ground truth, we use a much higher OKS threshold (0.99) for positive sample selection. Even though the second stage uses a much higher OKS threshold,

Table 9. Effect of backbone network and multi-scale testing.

Backbone	Multi-Test	AP	AP_{50}	AP_{75}
ResNet-50	✗	58.0	80.8	62.4
ResNeXt-101-DCN	✗	62.5	83.1	68.3
ResNeXt-101-DCN	✓	65.7	85.4	71.8
HRNet	✓	69.8	88.8	76.3

Table 10. Results on the MS COCO `test-dev2017` compared to state-of-the-art **pose estimation** methods.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_M	AP_L
Heat Map Based						
CMU-Pose [3]	3CM-3PAF (102)	61.8	84.9	67.5	57.1	68.2
RMPE [13]	Hourglass-4 stacked	61.8	83.7	69.8	58.6	67.6
Mask-RCNN [15]	ResNet50	63.1	87.3	68.7	57.8	71.4
G-RMI [31]	ResNet-101+ResNet-50	64.9	85.5	71.3	62.3	70.0
AE [27]	Hourglass-4 stacked	65.5	86.8	72.3	60.6	72.6
PersonLab [30]	ResNet-152	68.7	89.0	75.4	64.1	75.5
Regression Based						
CenterNet [43]	Hourglass-2 stacked (104)	63.0	86.8	69.6	58.9	70.4
SPM [29]	Hourglass-8 stacked	66.9	88.5	72.9	62.6	73.1
Ours	HRNet	68.7	89.9	76.3	64.8	75.3

it is found that many more ground truth poses are covered. Both the person classification and the pose regression losses are decreased. We thus obtain significant improvement from the second stage, specifically, 9.7 AP (relative 20.1% improvement) over the first stage.

Effect of stronger backbone network and multi-scale testing. Table 9 shows the result of using stronger backbones and multi-scale testing. Specifically, we obtain 4.5 AP improvement from using the ResNeXt-101-DCN backbone network. A 3.2 AP improvement is found from using multi-scale testing. We obtain the further improvement by using HRNet [35] as backbone (+4.1 AP).

Comparison with state-of-the-art methods. Finally, we test our model (with HRNet backbone and multi-scale testing) on the MSCOCO `test-dev2017` dataset and compare the result to other state-of-the-art methods in Table 10. PointSetNet outperforms CenterNet [43] by 5.7 AP and achieves results comparable to the state-of-the-art [30].

5 Conclusion

In this paper, we propose Point-Set Anchors which can be seen as a generalization and extension of classical anchors for high-level recognition tasks such as instance segmentation and pose estimation. Point-set anchors provide informative features and good task-specific initializations which are beneficial for keypoint regression. Moreover, we propose PointSetNet by simply replacing the anchor boxes with the proposed point-set anchors in RetinaNet and attaching a parallel branch for keypoint regression. Competitive experimental results on object detection, instance segmentation and human pose estimation show the generality of our point-set anchors.

References

1. Bolya, D., Zhou, C., Xiao, F., Lee, Y.J.: Yolact: real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9157–9166 (2019)
2. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: European Conference on Computer Vision. pp. 717–732. Springer (2016)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
4. Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: A foundation for dense object segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2061–2069 (2019)
5. Chen, Y., Shen, C., Wei, X.S., Liu, L., Yang, J.: Adversarial posenet: A structure-aware convolutional network for human pose estimation. arXiv preprint arXiv:1705.00389 (2017)
6. Chou, C.J., Chien, J.T., Chen, H.T.: Self adversarial training for human pose estimation. arXiv preprint arXiv:1707.02439 (2017)
7. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. arXiv preprint arXiv:1702.07432 (2017)
8. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3150–3158 (2016)
9. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1078–1085. IEEE (2010)
12. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6569–6578 (2019)
13. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2334–2343 (2017)
14. Gkioxari, G., Toshev, A., Jaitly, N.: Chained predictions using convolutional neural networks. In: European Conference on Computer Vision. pp. 728–743. Springer (2016)
15. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: International Conference on Computer Vision (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6409–6418 (2019)

18. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In: European Conference on Computer Vision. pp. 34–50. Springer (2016)
19. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018)
21. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2359–2367 (2017)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
26. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)
27. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems. pp. 2277–2287 (2017)
28. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision. pp. 483–499. Springer (2016)
29. Nie, X., Feng, J., Zhang, J., Yan, S.: Single-stage multi-person pose machines. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6951–6960 (2019)
30. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 269–286 (2018)
31. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.: Towards accurate multi-person pose estimation in the wild. arXiv preprint arXiv:1701.01779 (2017)
32. Peng, S., Jiang, W., Pi, H., Bao, H., Zhou, X.: Deep snake for real-time instance segmentation. arXiv preprint arXiv:2001.01629 (2020)
33. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4929–4937 (2016)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)

35. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5693–5703 (2019)
36. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 824–832 (2015)
37. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 529–545 (2018)
38. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9627–9636 (2019)
39. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. arXiv preprint arXiv:1804.06208 (2018)
40. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. arXiv preprint arXiv:1909.13226 (2019)
41. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9657–9666 (2019)
42. Yang, Z., Xu, Y., Xue, H., Zhang, Z., Urtasun, R., Wang, L., Lin, S., Hu, H.: Dense reppoints: Representing visual objects with dense point sets. arXiv preprint arXiv:1912.11473 (2019)
43. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
44. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 850–859 (2019)