Location-Sensitive Visual Recognition with Cross-IOU Loss

Kaiwen Duan[1] Lingxi Xie[2] Honggang Qi[1] Song Bai[3] Qingming Huang[1] Qi Tian[2]

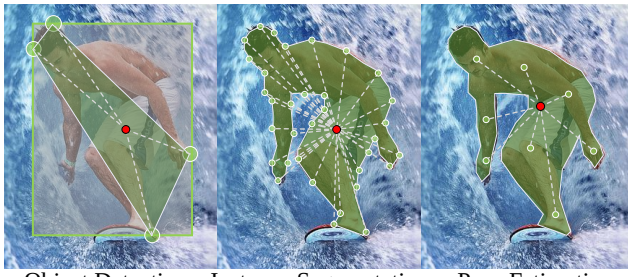
[1]University of Chinese Academy of Sciences [2]Huawei Inc
[3]Huazhong University of Science and Technology
kaiwenduan@outlook.com, 198808xc@gmail.com, {hgqi,qmhuang}@ucas.ac.cn
songbai.site@gmail.com, tian.qi1@huawei.com

Abstract

Object detection, instance segmentation, and pose estimation are popular visual recognition tasks which require localizing the object by internal or boundary landmarks. This paper summarizes these tasks as ***location-sensitive*** *visual recognition and proposes a unified solution named location-sensitive network (****LSNet****). Based on a deep neural network as the backbone, LSNet predicts an anchor point and a set of landmarks which together define the shape of the target object. The key to optimizing the LSNet lies in the ability of fitting various scales, for which we design a novel loss function named* ***cross-IOU loss*** *that computes the cross-IOU of each anchor-landmark pair to approximate the global IOU between the prediction and ground-truth. The flexibly located and accurately predicted landmarks also enable LSNet to incorporate richer contextual information for visual recognition. Evaluated on the MS-COCO dataset, LSNet set the new state-of-the-art accuracy for anchor-free object detection (a* 53.5% *box AP) and instance segmentation (a* 40.2% *mask AP), and shows promising performance in detecting multi-scale human poses. Code is available at* https://github.com/Duankaiwen/LSNet.

1. Introduction

Object recognition is a fundamental task in computer vision. Beyond image classification [16] that depicts an image using a single semantic label, there exist other recognition tasks that not only predict the class of the object but also localize it using fine-scaled information. In this paper, we consider three popular examples including object detection [19, 36], instance segmentation [19, 13], and human pose estimation [1, 36]. We notice that, despite the fact that the rapid progress of deep learning [32] has introduced powerful deep networks as the backbone [25, 60, 20], the designs of the *head* module for detection [38, 48, 49, 52, 65, 31, 30, 63, 6], segmentation [59, 24, 44, 7], and pose estimation [51, 57, 5, 40, 66] have fallen into individual subfields. This is mainly due to the difference in the prediction target, *i.e.*, a *bounding box* for detection, a pixel-level *mask* for segmentation, and a set of *keypoints* for pose estimation, respectively.



Object Detection Instance Segmentation Pose Estimation

Figure 1: The location-sensitive visual recognition tasks, including object detection, instance segmentation, and human pose estimation, can be formulated into localizing an anchor point (in red) and a set of landmarks (in green). Our work aims to offer a unified framework for these tasks.

Going one step further, we merge the aforementioned three tasks into one, named the **location-sensitive visual recognition (LSVR)**. On the basis of the definition, we propose a location-sensitive network (**LSNet**) as a unified formulation to deal with them all. The LSNet is built upon any network backbone, *e.g.*, those designed for image classification. The key is to relate an object to an anchor point and a set of landmarks that accurately localize the object. In particular, the landmarks should correspond to the four extreme points for object detection, sufficiently dense boundary pixels for instance segmentation, and the keypoints for human pose estimation. Note that the anchor point as well

as landmarks can be also used for extracting discriminative features of the object and thus assisting recognition. Figure 1 illustrates the overall idea.

The major difficulty of optimizing the LSNet lies in the requirement of fitting objects of different scales and different properties, which existing methods including the smooth-ℓ_1 loss and the IOU loss suffer cannot satisfy. This motivates us to design a novel loss function named the **cross-IOU loss**. It assumes that the landmarks are uniformly distributed around the anchor point and thus approximates the IOU (between the prediction and ground-truth) using the coordinate of the offset vectors. The cross-IOU loss is easily implemented in a few lines of codes. Compared to other loss functions, it achieves a better trade-off between the global and local properties and transplants easier to multi-scale feature maps without specific parameter tuning.

We perform all three tasks on the MS-COCO dataset [36]. LSNet, equipped with the cross-IOU loss, achieves competitive recognition accuracy. We further equip the LSNet with a pyramid of deformable convolution that extracts discriminative visual cues around the landmarks. As a result, LSNet reports a 53.5% box AP and a 40.2% mask AP, both of which surpass all existing anchor-free methods. For human pose estimation, LSNet reports competitive results without using the heatmaps, offering a new possibility to the community. Moreover, LSNet shows a promising ability in detecting human poses in various scales, some of which were not annotated in the dataset.

On top of these results, we claim two-fold contributions of this work. **First**, we present the formulation of location-sensitive visual recognition that inspires the community to consider the common property of these tasks. **Second**, we propose the LSNet as a unified framework in which the key technical contribution is the cross-IOU loss.

2. Related Work

Deep neural networks have been widely applied for visual recognition tasks. Among them, **image classification** [16] is the fundamental task that facilitates the design of powerful network backbones [25, 60, 20]. Beyond image-level description, there exist fine-scaled tasks, including object detection, instance segmentation, and pose estimation, which focus on depicting different aspects of the object. For example, the bounding boxes locate objects simply and efficiently but lack the details, while masks and keypoints reflect the shape and pose of the objects but usually need the bounding boxes to locate object firstly. According to the different properties of different tasks, many representative methods have been developed.

The **object detection** methods can be roughly categorized into anchor-based and anchor-free. The anchor-based methods detect objects by placing a pre-defined set of anchor boxes, predicting the class and score for each anchor, and finally regressing the preserved boxes tightly around the objects. The representative methods include Fast R-CNN [21], Faster RCNN [49], R-FCN [14], SSD [38], RetinaNet [35], Cascade R-CNN [4], *etc*. On another line, the anchor-free methods usually represent an object as a combination of geometry. Among them, CornerNet [30] and DeNet [55] generated a bounding box by predicting a pair of corner keypoints, and CenterNet (keypoint triplets) [17] and CPNDet [18] applied semantic information within the objects to filter out the incorrect corner pairs. FCOS [52], RepPoints [61], FoveaBox [29], SAPD [68], CenterNet (objects as points) [66], YOLO [47], *etc.*, defined a bounding box by placing a single point (called anchor point) within the object and predicting its distances to the object boundary.

For **instance segmentation**, there are mainly two kinds of methods, namely, the pixel-based and the contour-based methods. The pixel-based methods consider the segmentation problem as predicting the class of each single pixel. One of the representative work is Mask R-CNN [24], which first predicted the bounding box to help locating the objects, and then used pixel-wise classification to determine the object mask. The contour-based methods instead represent an object by the contour. They often start with a set points that are roughly located around the object boundary, and the points gradually get closer to the object boundary under iteration. The early representative methods are the snake series [27, 11, 23, 12] and the recent efforts include using deep neural network [22] and the idea of anchor-free to improve the features, such as DeepSnake [44] and PolarMask [59].

There are two mainstreams for **human pose estimation**, namely, the bottom-up methods [3, 10, 26, 39, 5] and top-down methods [66, 54, 58, 37]. The bottom-up methods first detect the human parts and then locates the keypoints in each object, while the top-down first locates all the keypoints in the human body and then composes the individual parts into a person. The keypoints are often sparsely distributed in an image and thus are difficult to be accurately located. A practical solution is to detect the keypoints in a high-resolution feature map, called heatmap [40, 10, 10]. However, applying the heatmap makes the optimization hard and introduces a complex post-processing operation. CenterNet [66] proposes a neat and simple method, which only predicts a center heatmap and the keypoints are obtained by regressing the vector from the center within objects to the keypoints.

This paper particularly focuses on the **anchor-free methods** for visual recognition. These methods originated from object detection and have drawn a lot of attention recently. They do not rely on the pre-defined anchor boxes to locate objects but by points and distance. Therefor, the

anchor-free methods enjoy the ability to extend into any directions. This offers the researchers a possibility to unify the visual recognition tasks. Recent trends have spent effort in extending the anchor-free methods in object detection into other tasks, *e.g.*, PolarMask [59] tries to extend the anchor-free into instance segmentation, while CenterNet [66] applies it into the pose estimation. Compared with our framework, both of them have limitations, which we will give a detailed discussion in section 3.2.

3. Our Approach

3.1. Location-Sensitive Visual Recognition

Visual recognition tasks start with an image, $\mathbf{X}$. Image classification aims to assign a class label c for the entire image, yet there are more challenging tasks for fine-scaled recognition. These tasks often focus on the instances (*i.e.*, individual objects) in the image and depict the object properties from different aspects. Typical examples include object detection that uses a rectangular box that tightly cover the object, instance segmentation that finds out each pixel that belongs to the object, and human pose estimation that localizes the landmarks of the object (*i.e.*, human keypoints). We use $\mathbf{b} \in \mathbb{R}^4$, $\mathbf{s} \in [0,1]^{W\times H}$, and $\mathbf{k} \in \mathbb{R}^{K\times 2}$ to indicate the target of these tasks, where W and H are the image width and height, and K is the number of keypoints.

An important motivation of our work is that, although these tasks differ from each other in the form of description, they share the common requirement that the model should be sensitive to the location of the anchor and/or landmarks. Throughout the remaining part of this paper, we refer to these tasks as **location-sensitive** visual recognition and design a unified framework for them.

3.2. LSNet: A Unified Framework

The proposed location-sensitive network (**LSNet**) starts with a backbone (*e.g.*, the ResNet [25], ResNeXt [60], *etc.*) that extracts features from the input image. We denote the process using $\tilde{\mathbf{x}} = f(\mathbf{X})$. Next, an anchor point and a few landmarks are predicted on top of $\tilde{\mathbf{x}}$, denoted as $\mathbf{p} = g(\tilde{\mathbf{x}})$. Here we define $\mathbf{p} \in \mathbb{R}^{(N+1)\times 2}$ where N is the number of landmarks, $\mathbf{p}_0 \in \mathbb{R}^2$ is the anchor point, and $\mathbf{p}_n \in \mathbb{R}^2$ is a landmark for $n = 1, 2, \ldots, N$.

As a unified framework, the key is to relate the prediction targets (*i.e.*, $\mathbf{b} \in \mathbb{R}^4$, $\mathbf{s} \in [0,1]^{W\times H}$, and $\mathbf{k} \in \mathbb{R}^{K\times 2}$ as aforementioned) to $\mathbf{p}$. For **object detection**, this is done by finding an extreme point (a pixel that belongs to the object and is tangent to the bounding box) on each edge of the bounding box[1], *i.e.*, $N = 4$. For **instance segmentation**, we locate a fixed number (*e.g.*, 36 in the experiments, $N = 36$) of landmarks along the contour and thus use the formed polygon to approximate the shape of the object[2]. For **human pose estimation**, we follow the definition of the dataset to learn a fixed number of keypoints, *e.g.*, in the MS-COCO dataset, $N = 17$.

Figure 2 shows the pipeline of LSNet. It belongs to the category of anchor-free methods, *i.e.*, there is no need to pre-define a set of anchor boxes for localizing the object. LSNet is partitioned into two stages, where the first stage predicts an anchor point from the FPN head and relates it with a set of landmarks, and the second stage composes the landmarks into an object with the desired geometry (*e.g.*, a bounding box). To facilitate accurate localization, we use the ATSS assigner [64] to assign more anchor points for each object and extract features with deformable convolution (DCN) upon the predicted landmarks. The entire model is an end-to-end trainable learnable function.

LSNet receives two sources of supervision for localization and classification, elaborated in Sections 3.3 and 3.4, respectively. The localization loss is added to both stages, where the major contribution is a unified loss that fits the properties of different tasks, and the classification loss is added to the second stage upon the DCN features.

LSNet extends the border of anchor-free methods for location-sensitive visual recognition. We briefly review two counterparts. **(i)** CenterNet predicted horizontal or vertical offsets beyond the anchor point for object detection. This limits its ability in finding the extreme points and extracting discriminative features, yet it cannot perform instance segmentation. **(ii)** PolarMask used a polar coordinate system for instance segmentation, making it difficult to process the situation that a ray intersects the object multiple times at some direction. In comparison, LSNet easily handles the challenging scenarios and report superior performance (see the experimental part, section 4).

3.3. Cross IOU Loss

The unified framework raises new challenges to the supervision of localization, because the function needs to consider both the global and local properties of the object. To clarify, we notice that the evaluation of object detection and instance segmentation judges if an object is correctly recognized by *the global IOU* between the prediction and the ground-truth, while pose estimation measures the accuracy by each *individual keypoint*.

To this end, we design the **cross-IOU loss** as the unified supervision. The loss is defined upon the predicted and ground-truth objects, and we use $\mathbf{p}^\star$, where $\mathbf{p}^\star = (\mathbf{p}^\star_x, \mathbf{p}^\star_y)$, to denote the corresponding ground-truth of the anchor point and landmarks.

[1] As a disclaimer, there may exist multiple or even continuous extreme points on each edge. We assume the method to find any one of them, by which it confirms the prediction of the bounding box.

[2] In case that the mask is not *simply-connected* in topology, we follow PolarMask [59] to deal with each part separately and ignore the holes.

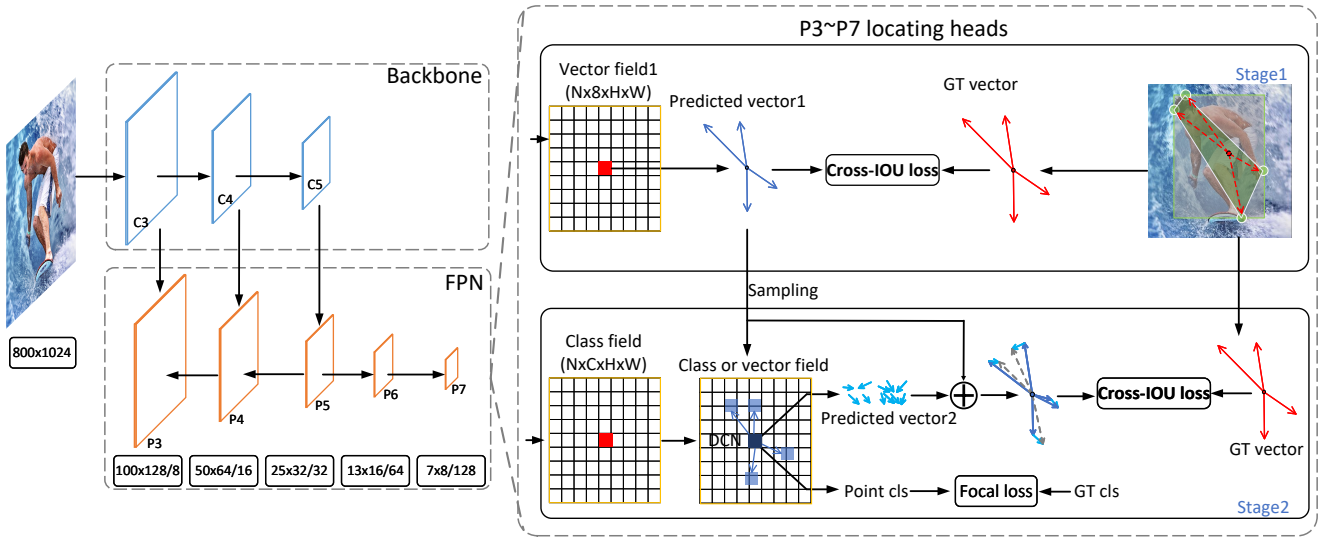


Figure 2: An illustration of the overall framework of LSNet, using object detection as an example, *i.e.*, $N = 4$. In the left part, C3–C5 denote the feature maps in the backbone that are connected to the feature pyramid [34], and P3–P7 denote the FPN layers used for final prediction. LSNet is partitioned into two stages. In the first stage, we predict a set of offset vectors that relate the anchor point to the extreme points under the supervision of the cross-IOU loss. In the second stage, the predicted vectors are used as the offsets of the deformable convolution (DCN) [15] to extract complementary visual features around the extreme points. These features are used for refining the localization and predicting the object class.

We compute the offset from the anchor point to the landmarks in a **cross-coordinate system**, *i.e.*, $\mathbf{q}_n = \left[(p_{n,x}-p_{0,x})^-, (p_{n,x}-p_{0,x})^+, (p_{n,y}-p_{0,y})^-, (p_{n,y}-p_{0,y})^+\right]$ for $n = 1, 2, \ldots, N$, where $(a)^-$ and $(a)^+$ denotes $\max\{-a, 0\}$ and $\max\{a, 0\}$, respectively. Finally, we write the cross-IOU loss as:

$$\mathrm{cIOU}(\mathbf{q}_n, \mathbf{q}_n^\star) = \frac{|\min\{\mathbf{q}_n - \mathbf{q}_n^\star\}|_1}{|\max\{\mathbf{q}_n - \mathbf{q}_n^\star\}|_1}, \tag{1}$$

where $|\cdot|_1$ indicates the ℓ_1-norm. In other words, the cross-IOU function rewards the components ($\mathbf{q}_n$ and $\mathbf{q}_n^\star$) of similar length (in which case the prediction and the ground-truth maximally overlap) and penalizes the components on different directions. Based on the Eqn (1), we define the cross-IOU loss as $\mathcal{L}_{\mathrm{cIOU}} = 1 - \frac{1}{n}\sum_{n=1}^{N} \mathrm{cIOU}(\mathbf{q}_n, \mathbf{q}_n^\star)$. Obviously, when $\mathbf{p}_n = \mathbf{p}_n^\star$ for all n, we have $\mathcal{L}_{\mathrm{cIOU}} = 0$ as expected[3].

The cross-IOU loss brings a direct benefit that it fits different scales of features without the need of specific parameter tuning. This alleviates the difficulty of integrating multi-scale information, *e.g.*, using the feature pyramid [34]. In comparison, the smooth-ℓ_1 loss [21] is sensitive to the scale of vector (*e.g.*, the loss value tends to be large when the feature resolution is large) and neglects the relationship between the components that are from the same vector. Moreover, by approximating the IOU using individual components, the cross-IOU loss is flexibly transplanted to instance segmentation and human pose estimation, unlike the original IOU loss [63] that is difficult to compute on polygons (for segmentation) and undefined for discrete keypoints (for pose estimation).

[3]The current form of $\mathcal{L}_{\mathrm{cIOU}}$ can cause the gradients over $p_{n,x}$ and $p_{n,y}$ to be 0 when the corresponding dimensions of prediction and ground-truth are of different signs. We design a softened prediction mechanism to solve the issue. Please refer to the Appendix A for details.

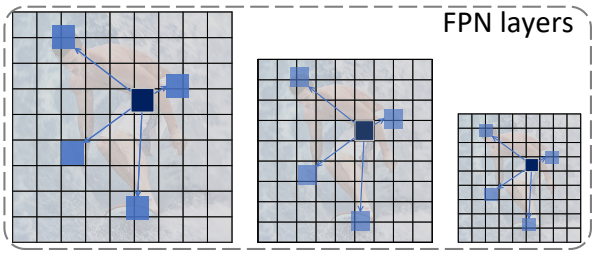


Figure 3: An illustration of the Pyramid DCN structure. The DCN kernel is allowed to be rescaled to the adjacent FPN layers to extract richer features. The blue boxes and arrows denote the positions of convolution determined by the predicted offset vectors (the blue arrows).

3.4. Pyramid DCN

To enhance discriminative information for recognition, we use deformable convolution (DCN) [15, 70] to extract features from the landmarks. The standard DCN has 9 off-

sets, while the number of offsets is 4, 36, and 17 for detection, segmentation, and pose estimation, respectively. In the latter two cases, to avoid redundant features extracted from close areas, we sample 9 landmarks uniformly from the candidates. We further build the feature extraction module upon the feature pyramid [34]. The offsets are adjusted to different stages by accordingly rescaling the vectors.

We name the proposed method Pyramid-DCN, and illustrate it in Figure 3. As shown in experiments, both feature extraction from the landmarks and using the pyramid structure improve recognition accuracy.

4. Experiments

4.1. Dataset and Evaluation Metrics

We evaluate our framework on the MS-COCO 2017 dataset [36], which is a popular, large-scale object detection, segmentation, and human pose dataset. For object detection and segmentation, it contains over 118K training images, 5K validation images and 20K *test-dev* images covering 80 object categories. For human pose, the person instance is labeled with 17 keypoints, containing over 57K training images with 150K person instances, 5K validation images and 20K *test-dev* images.

The average precision (AP) metric is applied to characterize the performance of our method as well as other competitors. There are subtle differences in the definition of AP for different tasks. For object detection, AP is calculated the average precision under different bounding box IOU thresholds(from 0.5 to 0.95), while the bounding box IOU is replaced with the mask IOU in instance segmentation. In the human pose task, AP is calculated based on the object keypoint similarity (OKS), which reflects the distance between the predicted keypoints and the annotations.

4.2. Implementation Details

We use ResNet [25], ResNeXt [60] and Res2Net [20] with the weights pre-trained on ImageNet [16] as our backbones, respectively. The feature pyramid network (FPN) [34] is applied to deal with objects with different scales. For object detection, we set four vectors for each object to learn to find the four extreme points (top, left, bottom, right). We refer to ExtremeNet [67] to obtain extreme point annotations from the object mask[4]. For instance segmentation and human pose estimation, we set 36 vectors for each instance to regress the location of contour points and 17 vectors to regress the 17 keypoints.

[4]As a side comment, when annotating the bounding boxes on a dataset, we recommend annotating an object by clicking the four extreme points (top-most, left-most, bottom-most, right-most) of the object. According to [42], this way is roughly four times faster than directly annotating the bounding boxes. In addition, the extreme point itself contains the object information as well.

Training and Inference. We train our framework on eight NVIDIA Tesla-V100 GPUs with two images on each GPU. The initial learning rate is set as 0.01, the weight decay as 0.0001 and momentum as 0.9. In the ablation study, we use a ResNet-50 [25] pre-trained on ImageNet [16] as the backbone, and fine-tune the model for 12 epochs using a single-scale of $[800, 1333]$ and augment the training images with random horizontal flipping. The learning rate decays by a factor of 10 at after the 8th and 11th epochs, respectively. We also use stronger backbones and longer training epochs (24 epochs for object detection, 30 epochs for instance segmentation and 60 epochs for human pose with the learning rate decayed by a factor of 10 after the 16th, 22nd, and 50th epochs, respectively) and multi-scale input images (from $[400, 1333]$ to $[960, 1333]$) to further improve the recognition accuracy. In the first stage, we only select the anchor point closest to the center of the object as a positive sample. In the second stage, we use the ATSS [64] assigner to assign the anchor points for each object. The overall loss function is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{vector}_1} + \gamma \cdot \mathcal{L}_{\text{vector}_2}, \tag{2}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{vector}}$ denote the Focal loss [35] and our cross-IOU loss, respectively. We set the balancing coefficients, β and γ, to be 1.0 and 2.0 in the experiments. During the inference, both the single-scale testing and multi-scale testing strategy are applied. We use the scale of $[800, 1333]$ for single-scale testing. For the multi-scale testing, we refer to ATSS [64] to set the image scales. We also use the non-maximum suppression (NMS) strategy with a threshold of 0.6 to remove the redundant results.

4.3. Object Detection

Comparisons to SOTA. We evaluate the detection accuracy of LSNet on the MS-COCO *test-dev* set, the results are shown in Table 1. As Table 1 shows, our method is an anchor-free detector, with a backbone of ResNet-50, LSNet achieves a box AP of 44.8% with 12.7 FPS, which has been competitive with other detectors that equipped with deeper backbones. When equipped with stronger backbones, LSNet performs even better. This benefits from our proposed cross-IOU loss. It helps the LSNet to locate the landmarks with high accuracy, the rich global information contained in the landmarks further promote the cross-IOU loss to regress the landmarks more accurately. With the additional corner point verification (CPV) [9] and multi-scale testing [64], LSNet achieves a box AP of 53.5%, which outperforms all the anchor-free detectors as we know.

Cross-IOU Loss for Vector Regression. To evaluate the performance of cross-IOU loss, we design four contrast experiments on the MS COCO [36] validation set, which are (i) the GIOU loss [50] (a variant of the IOU loss) for rectangle bounding box regression, (ii) the smooth-ℓ_1 loss for rect-

Method	Backbone	Epoch	MS_{train}	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Anchor-based:										
Libra R-CNN [41]	X-101-64x4d	12		8.5	43.0	64.0	47.0	25.3	45.6	54.6
AB+FSAF [69] †	X-101-64x4d	18	✓	-	44.6	65.2	48.6	29.7	47.1	54.6
FreeAnchor [65] †	X-101-32x8d	24	✓	-	47.3	66.3	51.5	30.6	50.4	59.0
GFLV1 [65]	X-101-32x8d	24	✓	10.7	48.2	67.4	52.6	29.2	51.7	60.2
ATSS [64] †	X-101-64x4d-DCN	24	✓	-	50.7	68.9	56.3	33.2	52.9	62.4
PAA [28] †	X-101-64x4d-DCN	24	✓	-	51.4	69.7	57.0	34.0	53.8	64.0
GFLV2 [33]†	R2-101-DCN	24	✓	-	53.3	70.9	59.2	35.7	56.1	65.6
YOLOv4-P7 [56]†	CSP-P7	450	✓	-	**56.0**	**73.3**	**61.2**	**38.9**	**60.0**	**68.6**
Anchor-free:										
ExtremeNet [67] †	HG-104	200	✓	-	43.2	59.8	46.4	24.1	46.0	57.1
RepPointsV1 [61] †	R-101-DCN	24	✓	-	46.5	67.4	50.9	30.3	49.7	57.1
SAPD [68]	X-101-64x4d-DCN	24	✓	4.5	47.4	67.4	51.1	28.1	50.3	61.5
CornerNet [30] †	HG-104	200	✓	-	42.1	57.8	45.3	20.8	44.8	56.7
DETR [6]	R-101	500	✓	10	44.9	64.7	47.7	23.7	49.5	62.3
CenterNet [17] †	HG-104	190	✓	-	47.0	64.5	50.7	28.9	49.9	58.9
CPNDet [18] †	HG-104	100	✓	-	49.2	67.4	53.7	31.0	51.9	62.4
BorderDet [46] †	X-101-64x4d-DCN	24	✓	-	50.3	68.9	55.2	32.8	52.8	62.3
FCOS-BiFPN [53]	X-101-32x8-DCN	24	✓	n/a	50.4	68.9	55.0	33.2	53.0	62.7
RepPointsV2 [9] †	X-101-64x4d-DCN	24	✓	-	52.1	70.1	57.5	34.5	54.6	63.6
LSNet	R-50	24	✓	12.7	44.8	64.1	48.8	26.6	47.7	55.7
LSNet	X-101-64x4d	24	✓	7.2	48.2	67.6	52.6	29.6	51.3	60.5
LSNet	X-101-64x4d-DCN	24	✓	5.9	49.6	69.0	54.1	30.3	52.8	62.8
LSNet-CPV	X-101-64x4d-DCN	24	✓	5.1	50.4	69.4	54.5	31.0	53.3	64.0
LSNet-CPV	R2-101-DCN	24	✓	6.3	51.1	70.3	55.2	31.2	54.3	65.0
LSNet-CPV †	R2-101-DCN	24	✓	-	**53.5**	**71.1**	**59.2**	**35.2**	**56.4**	**65.8**

Table 1: A comparison between LSNet and the sate-of-the-art methods in object detection on the MS-COCO *test-dev* set. LSNet surpasses all competitors in the anchor-free group. The abbreviations are: 'R' – ResNet [25], 'X' – ResNeXt [60], 'HG' – Hourglass network [40], 'R2' – Res2Net [20], 'CPV' – corner point verification [9], 'MS_{train}' – multi-scale training, '†' – multi-scale testing [64].

Loss	Box Style	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
GIOU	rectangle	34.6	54.7	36.6	19.3	38.2	44.4
Smooth-ℓ_1	rectangle	33.8	54.5	36.0	18.6	37.1	43.8
Smooth-ℓ_1	extreme	31.5	50.1	34.0	16.8	34.3	41.4
Cross-IOU	extreme	34.3	54.9	36.5	19.6	37.6	44.7

Table 2: The bounding box AP (%) with different experimental settings. The cross-IOU loss achieves a high score even when regressing the extreme bounding box which is more difficult. Note that the GIOU loss cannot regress a non-rectangle bounding box.

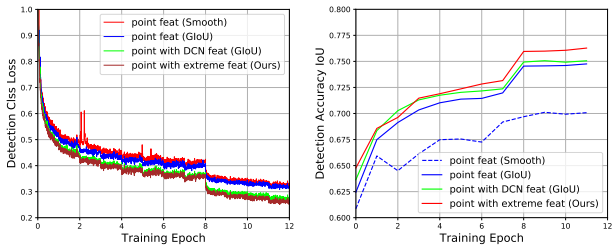
Loss	PA	PE	PP	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Cross-IOU	✓			34.3	54.9	36.5	19.6	37.6	44.7
	✓	✓		35.5	54.8	38.3	20.0	39.2	45.3
	✓		✓	**36.2**	**55.4**	**38.9**	**19.8**	**39.8**	**46.3**

Table 3: The detection accuracy (%) of using different features. PA, PD, and PP denote using the anchor point features along, anchor point features with the single-scale extreme point features, and with the pyramid extreme point features, respectively.

angle bounding box regression, (iii) the smooth-ℓ_1 loss for extreme bounding box regression, and (iv) the cross-IOU loss for extreme bounding box regression, respectively. All the experiments are done in the first stage in our framework (shown in Figure 2) with ResNet-50 [25] as the backbone, and we train the model for each experiment for 12 epochs. Table 2 summarizes the results. We can see that the smooth-ℓ_1 loss reports an AP of 33.8% and 31.5% when regressing the rectangle bounding boxes and the the extreme bounding boxes, respectively. This reveals that it is more difficult to regress an angled vector than a straight vector. By contrast, the cross-IOU loss performs much better than the smooth-ℓ_1 loss and even produces competitive results with the IOU loss for the rectangle bounding box. Although the GIOU loss at present still performs better than the cross-IOU loss, the cross-IOU loss allows the framework regressing the location of the landmarks, thus we could extract the discriminative information around the landmarks to enhance recognition. We will show in the next section that the combina-

Method	Backbone	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Pixel-based:								
YOLACT [2]	R-101	48	31.2	50.6	32.8	12.1	33.3	47.1
TensorMask [8]	R-101	72	37.1	59.3	39.4	17.1	39.1	51.6
Mask R-CNN [24]	X-101-32x4d	12	37.1	60.0	39.4	16.9	39.9	53.5
HTC [7]	X-101-64x4d	20	41.2	63.9	44.7	22.8	43.9	54.6
DetectoRS [45] †	X-101-64x4d	40	48.5	72.0	53.3	31.6	50.9	61.5
Contour-based:								
ExtremeNet [67]	HG-104	100	18.9	44.5	13.7	10.4	20.4	28.3
DeepSnake [44]	DLA-34 [62]	120	30.3	-	-	-	-	-
PolarMask [59]	X-101-64x4d-DCN	24	36.2	59.4	37.7	17.8	37.7	51.5
LSNet	X-101-64x4d-DCN	30	37.6	64.0	38.3	22.1	39.9	49.1
LSNet	R2-101-DCN	30	38.0	64.6	39.0	22.4	40.6	49.2
LSNet †	X-101-64x4d-DCN	30	39.7	65.5	41.3	25.5	41.3	50.4
LSNet †	R2-101-DCN	30	**40.2**	**66.2**	**42.1**	**25.8**	**42.2**	**51.0**

Table 4: Comparison of LSNet to the sate-of-the-art methods in instance segmentation task on the COCO *test-dev* set. Our LSNet achieves the state-of-the-art accuracy for contour-based instance segmentation. 'R': ResNet [25], 'X': ResNeXt [60], 'HG': Hourglass [40], 'R2':Res2Net [20], '†': multi-scale testing [64]



(a) (b)

Figure 4: The detection classification loss and average IOU with respect to the number of elapsed epochs. 'DCN feat', 'point feat' and 'extreme feat' denote the features adaptively learned by the DCN kernel, the features at the anchor points and extreme points, respectively. 'Smooth', 'GIOU' and 'Ours' denote the smooth-ℓ_1 loss, the GIOU loss [50], and the cross-IOU loss, respectively.

tion of the cross-IOU loss and landmark feature extraction significantly boosts recognition accuracy.

Landmark Features Improve Precision. The landmarks (in particular, the extreme points in the detection task) are often related to discriminative appearance features, which may benefit visual recognition. To confirm this, we investigate different settings by using the anchor point features alone and integrating the anchor point features with either DCN or extreme point features, respectively. We still report the detection accuracy with all other settings remaining the same as in the previous experiments (studying the cross IOU loss). The results are shown in Figure 4. For the smooth-ℓ_1 loss and the GIOU loss, we regress the rectangle bounding boxes, and the DCN features are extracted by the adaptively learned DCN kernel; for the cross-IOU loss, we use two sets of vectors both of which regress the extreme bounding boxes – the first set is trained to predict the extreme points and extract the extreme features, and we use the extreme features along with the anchor point features to train the second set from scratch. As shown in Figure 4, both the extreme and DCN features boost the classification accuracy. Recall that the prior experiments suggested the usefulness of the extreme features are useful for localization, combining the current results, we verify that the features around the landmarks are discriminative and thus benefit visual recognition.

Pyramid DCN Improves Precision. We further equip the LSNet with the pyramid DCN to extract the multi-scale features around the landmarks. Table 3 shows our method achieves an AP of 36.2% with the features extracted by the Pyramid DCN, which outperforms the AP with single-scale features by a margin of 0.7%.

4.4. Instance Segmentation

Comparisons to SOTA. We show the instance segmentation inference results evaluated on the MS-COCO *test-dev* set [36] on Table 4. LSNet achieves a mask AP of 38.0% and 40.2% using the single-scale and multi-scale testing protocols, respectively, surpassing all published contour-based methods to the best of our knowledge, and the accuracy is even competitive among the pixel-based approaches.

Comparisons with PolarMask [59]. It is interesting to further compare our method with PolarMask, the previous best contour-based approach for instance segmentation. The major difference is that PolarMask assumed the entire object boundary to be seen by the anchor point, but this may not be the case especially for some complicated objects. Once the ray along some direction intersects with the border more than once, the method considered only one and thus in-

Method	Backbone	Epoch	AP	AP_{50}	AP_{75}	AP_M	AP_L
Heatmap-based:							
CenterNet-jd [66]	DLA-34	320	57.9	84.7	63.1	52.5	67.4
OpenPose [5]	VGG-19	-	61.8	84.9	67.5	58.0	70.4
Pose-AE [39]	HG	300	62.8	84.6	69.2	57.5	70.6
CenterNet-jd [66]	HG104	150	63.0	86.8	69.6	58.9	70.4
Mask R-CNN [24]	R-50	28	63.1	87.3	68.7	57.8	71.4
PersonLab [43]	R-152	>1000	66.5	85.5	71.3	62.3	70.0
HRNet [51]	HRNet-W32	210	74.9	92.5	82.8	71.3	80.9
Regression-based:							
CenterNet-reg [66]	DLA-34	320	51.7	81.4	55.2	44.6	63.0
	HG-104	150	55.0	83.5	59.7	49.4	64.0
LSNet w/ obj-box	X-101-64x4d-DCN	60	55.7	81.3	61.0	52.9	60.5
LSNet w/ kps-box	X-101-64x4d-DCN	20	59.0	83.6	65.2	53.3	67.9

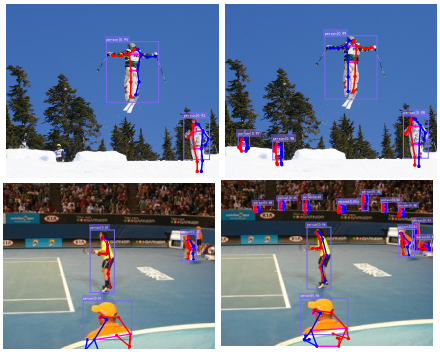


Table 5: Left: Comparison of LSNet to the sate-of-the-art methods in pose estimation task on the COCO *test-dev* set. LSNet predict the keypoints by regression. 'obj-box' and 'kps-box' denote the object bounding boxes and the keypoint-boxes, respectively. For LSNet w/ kps-box, we fine-tune the model from the LSNet w/ kps-box for another 20 epochs. Right: We compared with the CenterNet [66] to show that our LSNet w/ 'obj-box' tends to predict more human pose of small scales, which are not annotated on the dataset. Only pose results with scores higher than 0.3 are shown for both methods.

Figure 5: Some location-sensitive visual recognition results on the MS-COCO validation set. As discussed in Section 4.4, the contour of the 'motorcycles' in the figure cannot be well represented by the polar coordinate system in PolarMask.

curred accuracy loss (a typical example is the 'motorcycle' contour in Figure 5). In our approach, this issue is solved by ranking the landmarks more flexibly, being compatible to complicated shapes.

The Number of Landmarks. LSNet represents each instance using a polygon. Using a larger number of landmarks improves the upper-bound of accuracy, but can also incur heavy computational costs and cause the landmark prediction module difficult to be optimized. To choose a proper number of landmarks, we refer to the ground-truth masks of the MS-COCO validation set and quantize each mask into a polygon that best describes it. We find that using 18, 36, and 72 landmarks achieves APs of 89.0%, 97.4%, and 99.2%, respectively, and we consider $N = 36$ to be a nice tradeoff.

4.5. Human Pose Estimation

Comparisons to SOTA. Unlike most of the human pose estimation methods that predict the keypoints using the heatmaps, LSNet predicts the keypoints using regression only. In the experiment, we use the object bounding boxes ('obj-box') and keypoint-boxes ('kps-box') to assign training samples, respectively. We will give a detailed discussion of the difference between the two methods in the Appendix B. On the MS-COCO *test-dev* set, LSNet reports an AP of 55.7% w/ obj-box and 59.0% w/ kps-box, respectively, which outperform CenterNet-reg [66] with the Hourglass-104 backbone. However, LSNet does not perform as well as the heatmap-based methods, and we analyze the reason as follows.

Error Analysis. We can observe that the LSNet struggles particularly in the high OKS regimes, *e.g.*, compared to Pose-AE [39], the deficit of AP_{50} (for LSNet w/ obj-box) is 3.3% while that of AP_{75} grows to 9.0%. Note that using keypoint regression is not as accurate as using the heatmaps for refinement, and thus LSNet is less sensitive in the pixel-level prediction. However, the AP metric of pose estimation is largely impacted by this factor. To show this, we artificially add an average deviation of 1, 2, and 3 pixels to the

prediction results of CenterNet-jd [66] (with a backbone of Hourglass-104). The AP on the MS-COCO validation set is significantly reduced from 64.0% (corresponding to the test AP of 63.0% in Table 5) to 61.1%, 53.4%, and 44.0%, respectively.

On the other hand, we use the heatmaps produced by CenterNet-jd (Hourglass-104) to refine the prediction of LSNet w/ obj-box. As a result, the AP on the MS-COCO validation set is improved from 56.5% (corresponding to the test AP of 55.7% in Table 5) to 60.7%. This suggests that LSNet still needs further manipulation of high-resolution features towards higher pixel-level accuracy.

The Benefit of LSNet. Despite the relatively weak pixel-level localization, LSNet (w/ obj-box) enjoys the ability of perceiving multi-scale human instances, many of which are not annotated in the dataset. Some examples are shown in the right side of the Table 5. Since the ground-truth is not available to evaluate the impact, we refer to the heatmaps of CenterNet-jd (Hourglass-104) to deliberately remove these 'false positives'. Consequently, AP is further improved from 60.7% to 63.0%, comparable with the heatmap-based methods, though the improvement seems less meaningful.

4.6. Qualitative results for LSNet

We show some visualized results of LSNet in Figure 5, inlcuding object detection, instance segmentation and human pose estimation. Please refer to the appendix for more qualitative results.

5. Conclusions

This paper unifies three location-sensitive visual recognition tasks (object detection, instance segmentation, and human pose estimation) using the location-sensitive network (LSNet). The key module that supports the framework is a novel cross-IOU loss that is friendly to receiving supervision from multiple scales. Equipped with a pyramid DCN, LSNet achieves the state-of-the-art performance on anchor-free detection and segmentation. This work suggests that using keypoints to define and localize objects is a promising direction, and we hope to extend our approach to achieve a stronger ability of generalization.

APPENDIX

A. The Softened Prediction Mechanism for Cross-IOU Loss

As mentioned in Section 3.3, the form of $\mathcal{L}_{\text{cIOU}}$ (Equation 1) can cause the gradients over $p_{n,x}$ and $p_{n,y}$ to be 0 when the corresponding dimensions of prediction and ground-truth are of different signs. To solve this problem, we predict four components for each offset vector, as shown in Figure 6. Then $\mathbf{q}_n$ can be rewritten as: $\mathbf{q}_n = [q_{n,t}, q_{n,l}, q_{n,b}, q_{n,r}]$, where $q_{n,l}, q_{n,r}, q_{n,t}, q_{n,b}$ are all greater than 0. On the other hand, when transforming $\mathbf{q}_n^\star$ into the cross-coordinate system, as shown in Figure 6, we assign the minimum sides ($q^\star_{n,l}$ and $q^\star_{n,b}$) the non-zero value, which are α times the corresponding maximum sides ($q^\star_{n,t}$ and $q^\star_{n,r}$), where $0 < \alpha < 1$. In all our experiments, we set $\alpha = 0.2$.

During inference, we transform the predicted offset vectors from the cross-coordinate system into the rectangular coordinate system by taking the maximum value in the horizontal and vertical direction, respectively, *i.e.*, $\mathbf{q}_n = [\max\{q_{n,t}, q_{n,b}\}, \max\{q_{n,l}, q_{n,r}\}]$.

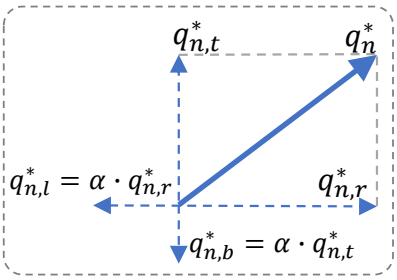


Figure 6: We use four components to represent a vector, and all the components are greater than 0.

B. Assign Samples by Keypoint-boxes to Improve Human Pose Estimation

In Section 4.5, we mainly discuss the characteristics of using the object bounding boxes ('obj-box') to assign training samples, which lets LSNet enjoy the ability of perceiving multi-scale human instances especially for small human instances, many of which are not annotated in the dataset. In this section, we mainly discuss the characteristics of using the keypoint-boxes ('kps-box', bounding box generated by the topmost, leftmost, bottommost and rightmost keypoints of an objects) to assign training samples. Compared with the former, the later will no longer treat the human instances that only have the object bounding box annotations but lack of pose annotations as positive samples. This makes the network pay more attention to learn the human instances that have the pose annotations, which helps to improve the AP score. As shown in Table 5, LSNet using keypoint-boxes reports an AP of 59.0%, an improvement of 3.3% over 55.7%, achieved by LSNet using object bounding boxes.

However, we find that, with the 'improved' AP score, the ability of the algorithm at perceiving multi-scale human instances is weakened. As shown in Figure 7, the modified algorithm mostly fails to detect the small person instances. This proves that the annotations of the dataset is biased.

Figure 7: Left: LSNet uses the object bounding boxes to assign training samples. Right: LSNet uses the keypoint-boxes to assign training samples. Although LSNet with keypoint-boxes enjoys a higher AP score, its ability of perceiving multi-scale human instances is weakened.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 1

[2] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019. 7

[3] Y. Cai, Z. Wang, Z. Luo, B. Yin, A. Du, H. Wang, X. Zhang, X. Zhou, E. Zhou, and J. Sun. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*, pages 455–472. Springer, 2020. 2

[4] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019. 1, 2, 8

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 6

[7] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 1, 7

[8] X. Chen, R. Girshick, K. He, and P. Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2061–2069, 2019. 7

[9] Y. Chen, Z. Zhang, Y. Cao, L. Wang, S. Lin, and H. Hu. Reppoints v2: Verification meets regression for object detection. *Advances in Neural Information Processing Systems*, 33, 2020. 5, 6

[10] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognitio*, pages 5386–5395, 2020. 2

[11] L. D. Cohen. On active contour models and balloons. *CVGIP: Image understanding*, 53(2):211–218, 1991. 2

[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 2

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1

[14] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 2

[15] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2, 5

[17] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578, 2019. 2, 6

[18] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian. Corner proposal network for anchor-free, two-stage object detection. *European Conference on Computer Vision*, pages 399–416, 2020. 2, 6

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[20] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 2, 5, 6, 7

[21] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 4

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[23] S. R. Gunn and M. S. Nixon. A robust snake implementation; a dual active contour. *IEEE Transactions on pattern analysis and machine intelligence*, 19(1):63–68, 1997. 2

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 7, 8

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 3, 5, 6, 7

[26] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. 2

[27] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 2

[28] K. Kim and H. S. Lee. Probabilistic anchor assignment with iou prediction for object detection. *European Conference on Computer Vision*, 2020. 6

[29] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 2020. 2

[30] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *European conference on computer vision*, pages 734–750, 2018. 1, 2, 6

[31] H. Law, Y. Teng, O. Russakovsky, and J. Deng. Cornernet-lite: Efficient keypoint based object detection. *arXiv preprint arXiv:1904.08900*, 2019. 1

[32] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[33] X. Li, W. Wang, X. Hu, J. Li, J. Tang, and J. Yang. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021. 6

[34] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4, 5

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 5

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, 2014. 1, 2, 5, 7

[37] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng. Improving convolutional networks with self-calibrated convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10096–10105, 2020. 2

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, 2016. 1, 2

[39] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, pages 2277–2287, 2017. 2, 8

[40] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1, 2, 6, 7

[41] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019. 6

[42] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939, 2017. 5

[43] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy. 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)*, pages 269–286, 2018. 8

[44] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8533–8542, 2020. 1, 2, 7

[45] S. Qiao, L.-C. Chen, and A. Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020. 7

[46] H. Qiu, Y. Ma, Z. Li, S. Liu, and J. Sun. Borderdet: Border feature for dense object detection. In *European Conference on Computer Vision*, pages 549–564. Springer, 2020. 6

[47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[48] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[49] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[50] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 5, 7

[51] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1, 8

[52] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019. 1, 2

[53] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 6

[54] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 2

[55] L. Tychsen-Smith and L. Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *Proceedings of the IEEE international conference on computer vision*, pages 428–436, 2017. 2

[56] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*, 2020. 6

[57] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[58] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 2

[59] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo. Polarmask: Single shot instance segmentation

with polar representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12193–12202, 2020. 1, 2, 3, 7

[60] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1, 2, 3, 5, 6, 7

[61] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019. 2, 6

[62] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018. 7

[63] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016. 1, 4

[64] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 3, 5, 6, 7

[65] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye. Freeanchor: Learning to match anchors for visual object detection. In *Advances in Neural Information Processing Systems*, pages 147–155, 2019. 1, 6

[66] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 3, 8, 9

[67] X. Zhou, J. Zhuo, and P. Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 850–859, 2019. 5, 6, 7

[68] C. Zhu, F. Chen, Z. Shen, and M. Savvides. Soft anchor-point object detection. *arXiv preprint arXiv:1911.12448*, 2019. 2, 6

[69] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019. 6

[70] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 4