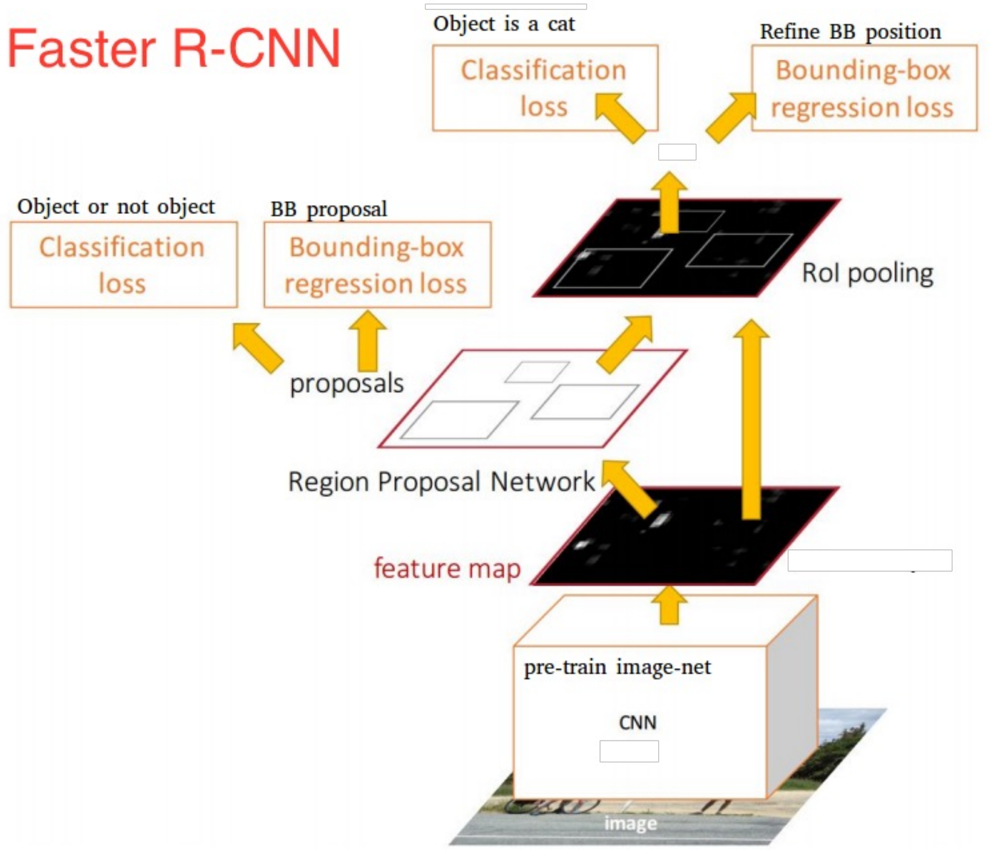
Point Set Representation for Object Detection and Beyond

Presented by Ze Yang

Object detection

- The pipeline of multi-stage object detection



bbox anchors $\xrightarrow{\text{bbox reg.}}$ bbox proposals (S1)

$\xrightarrow{\text{bbox reg.}}$ bbox proposals (S2)

…

$\xrightarrow{\text{bbox reg.}}$ bbox object targets

Bounding box regression

Why bounding box?

- Bounding box is convenient to annotate with little ambiguity.

- Almost all image feature extractors, both before and in the deep learning era, are based on an input patch in the grid form. Thus, it is convenient to use the bounding box representation to facilitate feature extraction.

Bounding box regression

Limitations

- Coarse object feature extraction.
- Unable to tackle irregular object (like road)
- It would perform badly when we need to regress object localization with large distance to the initial representation (need dense anchors)
- Scale difference between Δx, Δy and Δw, Δh, where usually different loss weights on them are required to be tuned for optimal performance.

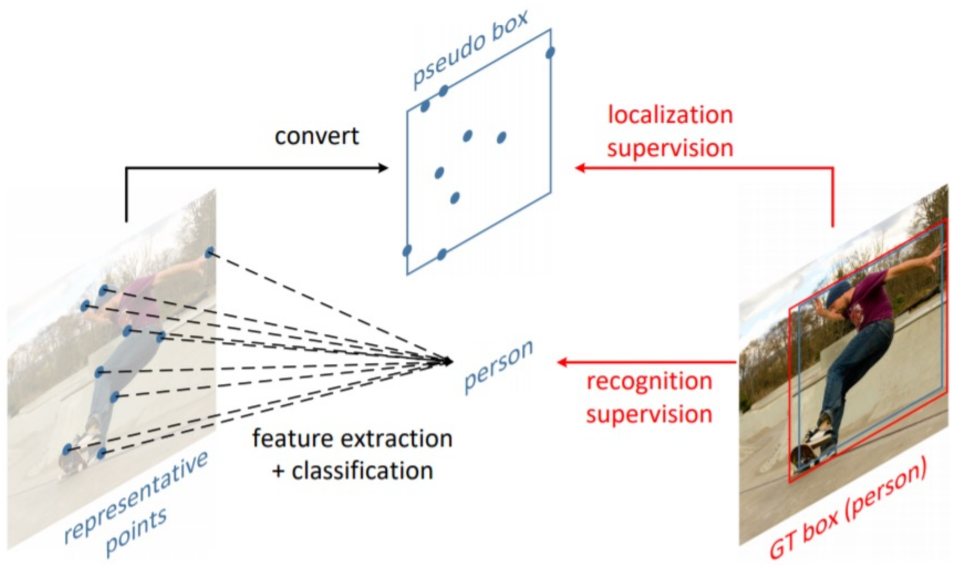
$$\mathcal{B}_r = (x_p + w_p \Delta x_p, y_p + h_p \Delta y_p, w_p e^{\Delta w_p}, h_p e^{\Delta h_p})$$

RepPoints: Point Set Representation for Object Detection

Representative points

- A new representation for object.

A **RepPoints** is defined as **a set of adaptive sample points**. The adaptive nature makes this new object representation more flexible than the bounding box representation in encoding the semantics-related object information.



Representative points

- Convert reppoints to bounding box

For a **RepPoints**, we can perform pre-defined function to transform **RepPoints** into **pseudo-box** so that the bounding box supervision can be imposed.

1. *Min-max function*: Min-max operation over both axes are performed to acquire rectangular box
2. *Moment-based function*. The mean value and the second-order moment of the deformable box is used to estimate the center points and the scale of rectangular box, where the scale is multiplied by globally shared learnable multipliers.

Representative points

- RepPoints refinement

$$\mathcal{D} = \{(x_k, y_k)\}_{k=1}^{n},$$

$$\mathcal{D}_r = \{(x_k + \Delta x_k, y_k + \Delta y_k)\}_{k=1}^{n},$$

$$\begin{aligned} \text{object centers} \quad &\xrightarrow{\text{RP refine}} \quad \text{RepPoints proposals (S1)} \\ &\xrightarrow{\text{RP refine}} \quad \text{RepPoints proposals (S2)} \\ &\ldots \\ &\xrightarrow{\text{RP refine}} \quad \text{RepPoints object targets} \end{aligned}$$

- Bounding box refinement

$$\mathcal{B}_p = (x_p, y_p, w_p, h_p)$$

$$\mathcal{B}_r = (x_p + w_p \Delta x_p, y_p + h_p \Delta y_p, w_p e^{\Delta w_p}, h_p e^{\Delta h_p}).$$

$$\begin{aligned} \text{bbox anchors} \quad &\xrightarrow{\text{bbox reg.}} \quad \text{bbox proposals (S1)} \\ &\xrightarrow{\text{bbox reg.}} \quad \text{bbox proposals (S2)} \\ &\ldots \\ &\xrightarrow{\text{bbox reg.}} \quad \text{bbox object targets} \end{aligned}$$

Representative points

- RepPoints Detector (RPDet)

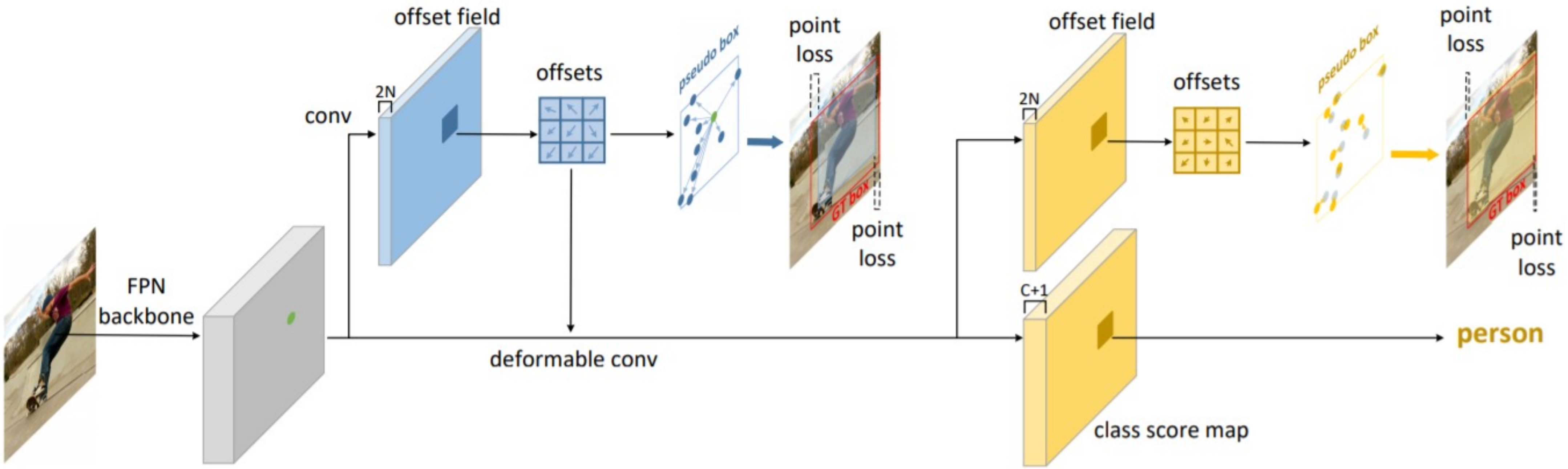


Figure 2. Overview of the proposed RPDet (RepPoints detector). While feature pyramidal networks (FPN) [27] are adopted as the backbone, we only draw the afterwards pipeline of one scale of FPN feature maps for clear illustration. Note all scales of FPN feature maps share the same afterwards network architecture and the same model weights.

Representative points

- RepPoints Detector (RPDet)

Center point initialization: center point as the initial representation of objects, leading to our anchor free object detector.

The use of RepPoints: the learning **RepPoints** is driven by: 1) the corner distance loss between the induced pseudo box and the ground-truth bounding box; 2) the object recognition loss of the subsequent stage.

Unified design across stages: without the need of RPN, NMS, ROI-Pooling…

Representative points

- Ablation on objects representation

Representation	Backbone	AP	AP_{50}	AP_{75}
Bounding box	ResNet-50	36.2	57.3	39.8
RepPoints (ours)	ResNet-50	**38.3**	**60.0**	**41.1**
Bounding box	ResNet-101	38.4	59.9	42.4
RepPoints (ours)	ResNet-101	**40.4**	**62.0**	**43.6**

Table 1. Comparison of the RepPoints and bounding box representations in object detection. The network structures are the same except for processing the given object representation.

Representative points

- Ablation on anchor free design

method	backbone	# anchors per scale	AP
RetinaNet [28]	ResNet-50	3×3	35.7
FPN-RoIAlign [27]	ResNet-50	3×1	36.7
YOLO-like	ResNet-50	-	33.9
RPDet (ours)	ResNet-50	-	**38.3**
RetinaNet [28]	ResNet-101	3×3	37.8
FPN-RoIAlign [27]	ResNet-101	3×1	39.4
YOLO-like	ResNet-101	-	36.3
RPDet (ours)	ResNet-101	-	**40.4**

Table 4. Comparison of the proposed method (RPDet) with an anchor-based method (RetinaNet, FPN-RoIAlign) and an anchor-free method (YOLO-like). The YOLO-like method is adapted from the YOLOv1 method [35] by additionally introducing FPN [27], GN [48] and focal loss [28] into the method for better accuracy.

Representative points

- Ablation on transform functions.

pseudo box converting function	AP	AP_{50}	AP_{75}
$\mathcal{T} = \mathcal{T}_1$: min-max	38.2	59.7	40.7
$\mathcal{T} = \mathcal{T}_2$: partial min-max	38.1	59.6	40.5
$\mathcal{T} = \mathcal{T}_3$: moment-based	38.3	60.0	41.1

Table 5. Comparison of different transformation functions from RepPoints to pseudo box, $\mathcal{T}$.

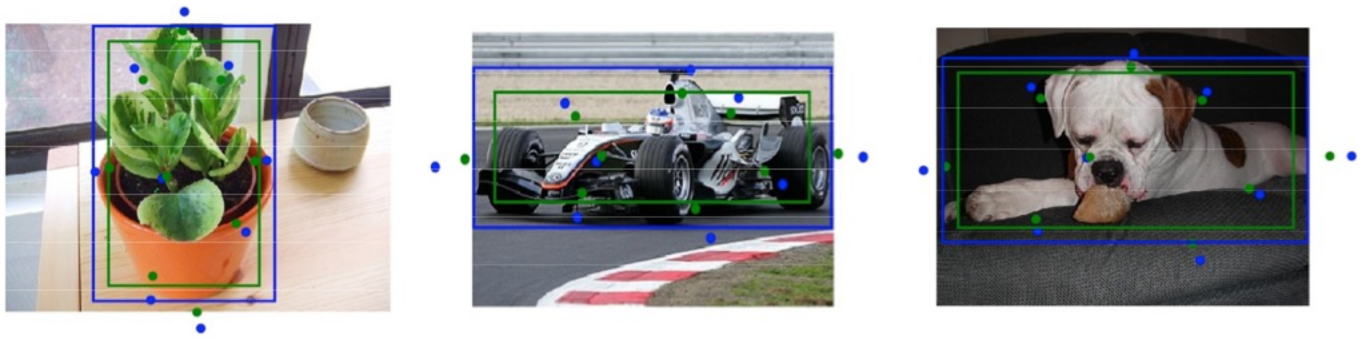
Representative points

- Comparison with Deformable RoI Pooling

representation method	w. dpool	AP	AP_{50}	AP_{75}
bounding box		36.2	57.3	39.8
	✓	36.9	58.0	41.0
RepPoints		38.3	60.0	41.1
	✓	**39.1**	**60.6**	**42.4**

Table 6. The effect of applying the deformable RoIpooling layer [4] on the proposals of the first stages (see Eq. (1) and Eq. (6)). The deformable RoIpooling layer can boost both the methods using bounding boxes and RepPoints, respectively.

The **RepPoints** target at both representing the fine-grained localization of objects as well as extracting semantic aligned object features, deformable RoI pooling is mainly driven by the recognition target. Actually, deformable RoI pooling cannot learn the accurate localization of objects.



Representative points

- State-of-the-art Comparison

	Backbone	Anchor-Free	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv2 [36]	DarkNet-19		21.6	44.0	19.2	5.0	22.4	35.5
SSD [31]	ResNet-101		31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [37]	DarkNet-53		33.0	57.9	34.4	18.3	35.4	41.9
DSSD [10]	ResNet-101		33.2	53.3	35.2	13.0	35.4	51.1
Faster R-CNN w. FPN [27]	ResNet-101		36.2	59.1	39.0	18.2	39.0	48.2
RefineDet [52]	ResNet-101		36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [28]	ResNet-101		39.1	59.1	42.3	21.8	42.7	50.2
Deep Regionlets [49]	ResNet-101		39.3	59.8	-	21.7	43.7	50.9
Mask R-CNN [14]	ResNeXt-101		39.8	62.3	43.4	22.1	43.2	51.2
FSAF [56]	ResNet-101		40.9	61.5	44.0	24.0	44.2	51.3
LH R-CNN [26]	ResNet-101		41.5	-	-	25.2	45.3	53.1
Cascade R-CNN [2]	ResNet-101		42.8	62.1	46.3	23.7	45.5	55.2
CornerNet [24]	Hourglass-104	✓	40.5	56.5	43.1	19.4	42.7	53.9
ExtremeNet [54]	Hourglass-104	✓	40.1	55.3	43.2	20.3	43.2	53.1
RPDet	ResNet-101	✓	41.0	62.9	44.3	23.6	44.1	51.7
RPDet	ResNet-101-DCN	✓	**42.8**	**65.0**	**46.3**	**24.9**	**46.2**	**54.7**

Table 7. Comparison the proposed RPDet to the state-of-the-art detectors on COCO [29] `test-dev`. Without multi-scale training and testing, our proposed framework achieves 42.8 AP with ResNet-101-DCN backbone [16, 4], which is on-par with 4-stage anchor-based Cascade R-CNN [2] method and outperforms all existing anchor free detectors. Moreover, RPDet obtains an AP_{50} of 65.0, surpassing all baselines by a significant margin.

Representative points

- Visualization

Figure 3. Visualization of the learned RepPoints and the corresponding detection results on several examples from the COCO [29] minival set (using pseudo box converting function of $\mathcal{T}_1$). In general, the learned RepPoints are located on extreme or semantic keypoints of the objects.

Conclusion

- In this paper, we propose a new object representation: **representative points**. Our work takes a new step towards learning the natural object representation. Exploiting dense point sets as the **RepPoints** and extending this representation beyond detection remain to be interesting future directions.

Future direction

- **Box-free** objection recognition tasks: multi-person pose estimation, instance segmentation …
- **Correspondence** from video: use flow or image augmentation to learn dense correspondence.
- **Better representation**: combine the merits from masks (finer / denser representation) and key-points (the points are semantic meaningful)
- **End-to-end Tracking.**
- **…**

Thanks for your attending!