RepPoints: Point Set Representation for Object Detection

Ze Yang*, Shaohui Liu*, Han Hu, Liwei Wang, Stephen Lin

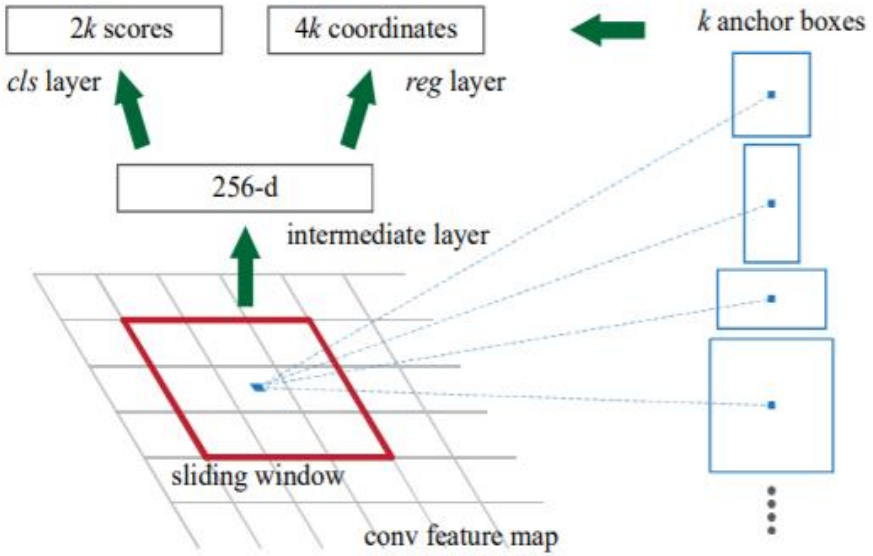
May 7, 2019

Microsoft Research Asia

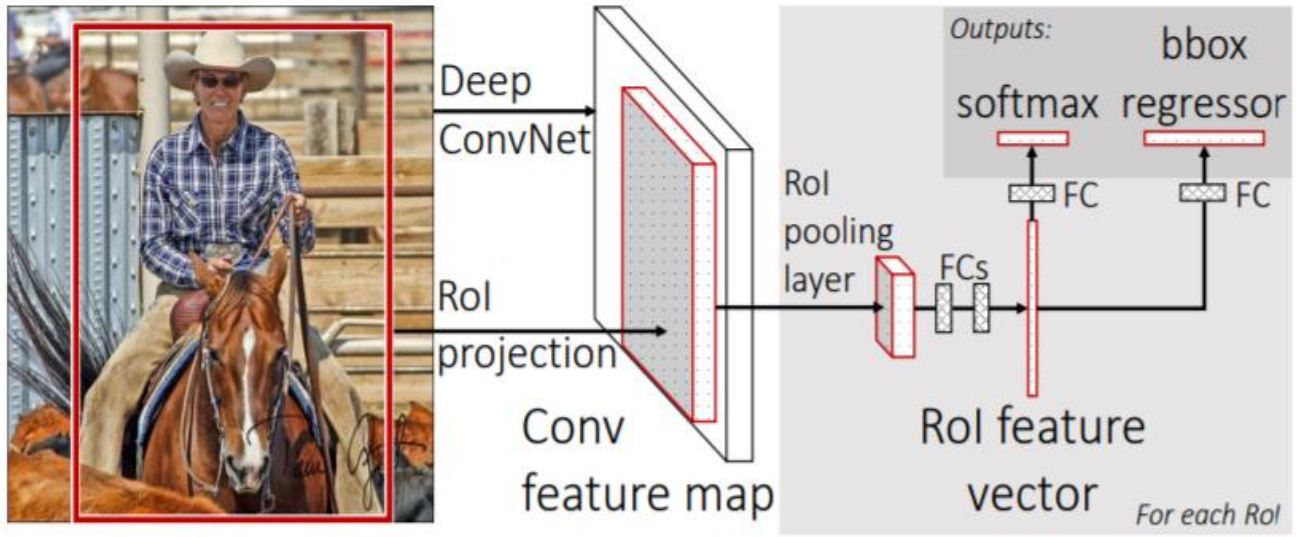
Overview

- Review of modern object detection pipelines
- RepPoints: bounding box -> point set representation
- RPDet: an anchor-free object detector based on RepPoints
- More discussion
  - interpretable deformation modeling
  - extending RepPoints: denser (seg) and finer target (correspondence)
  - regression vs. discrimination

Review of modern object detection pipelines



RPN design in Faster R-CNN



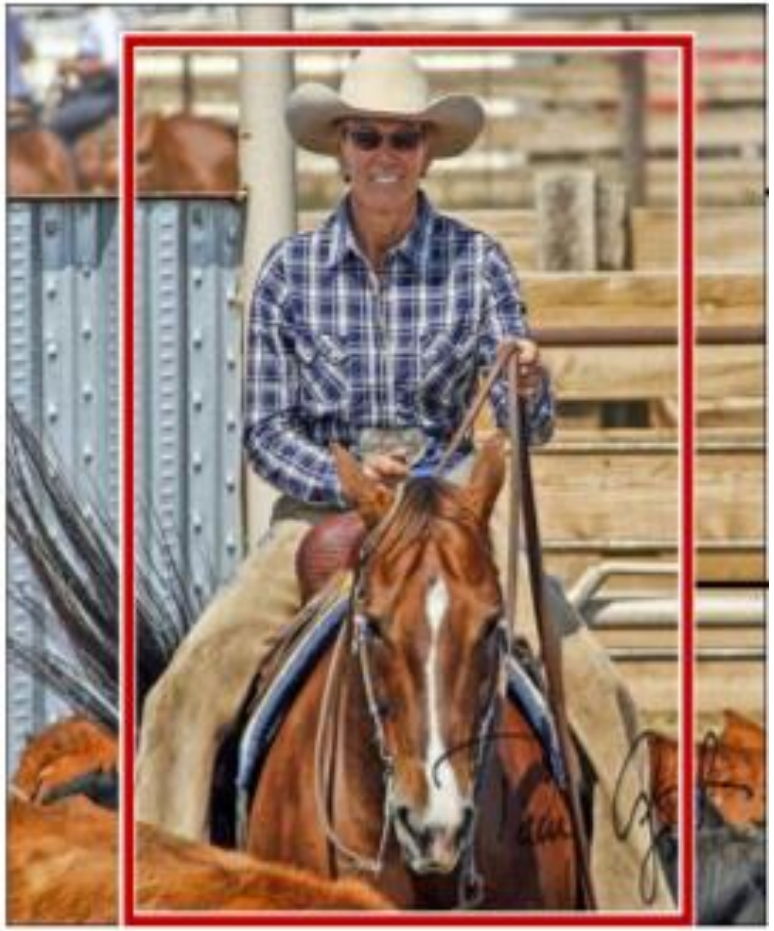
RoI feature extraction in Fast R-CNN

Bounding boxes are used as anchors, proposals and final predictions.

Bounding boxes are used as anchors, proposals and final predictions.

Bounding box has several advantages:

- Easy to be annotated
- Friendly for feature extraction
- Consistent with common metrics (bbox IoU)



Bounding box also has limitations:

- Insensitive to object shape and pose (coarse localization lack of geometric information)

-> lower localization capability

- Distractive background content and informative foreground content included

-> degraded feature and lower recognition capability

RepPoints: Point Set Representation

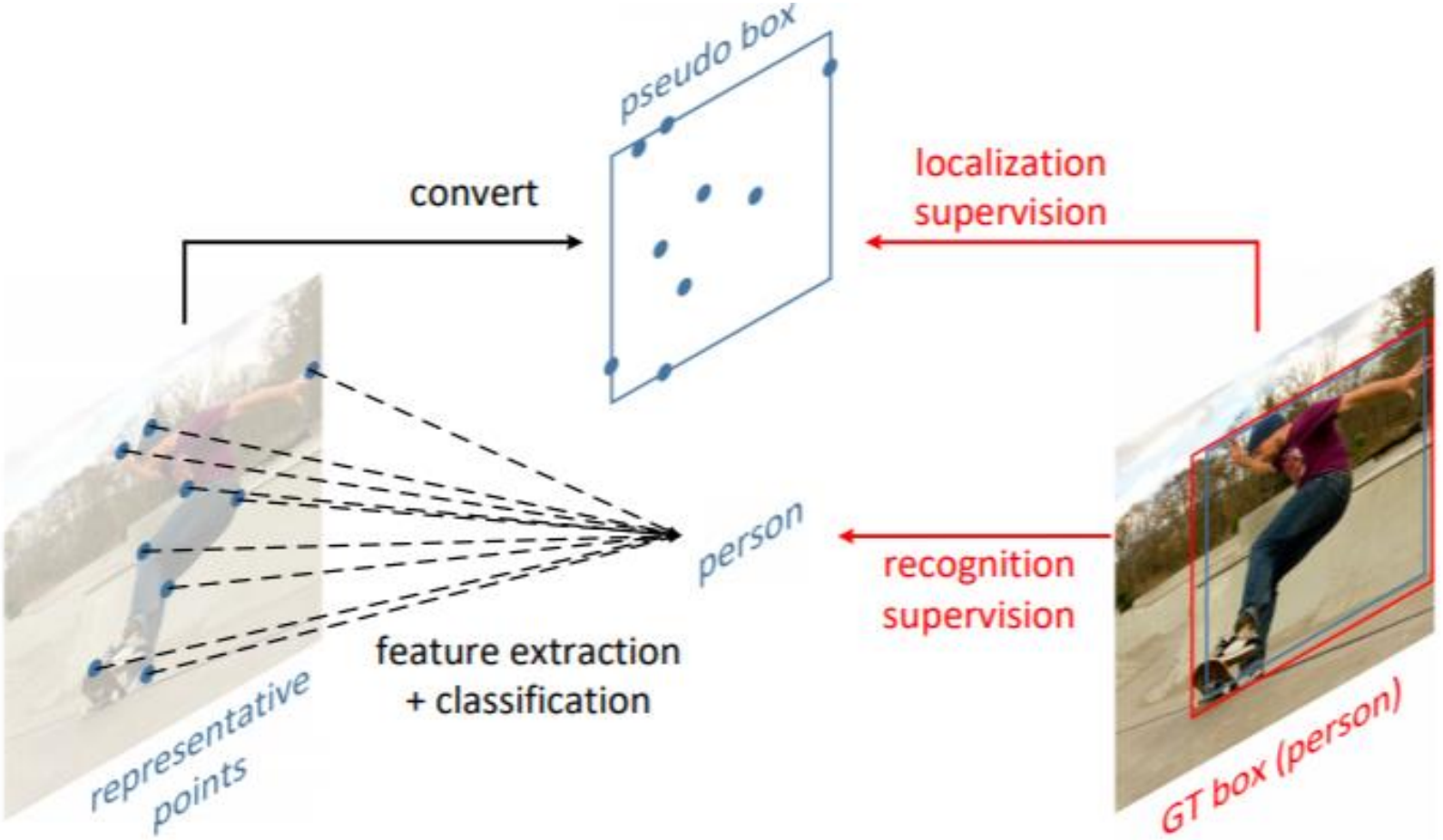
Bounding box vs. RepPoints

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

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

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

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



Learning Representative Points (RepPoints)

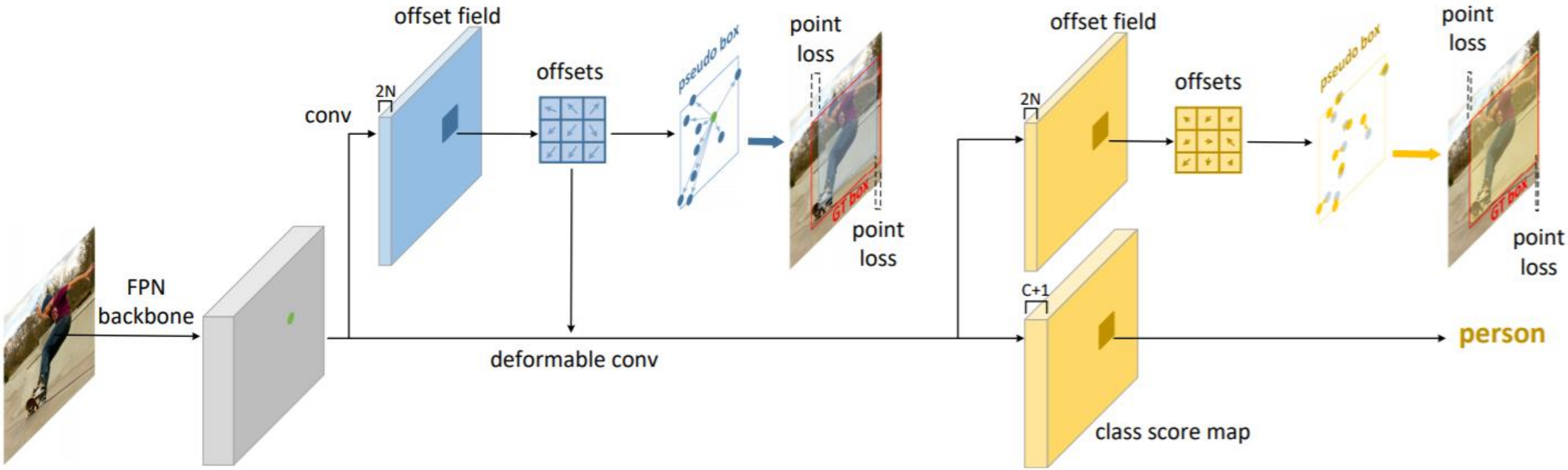
RepPoints: Point Set Representation

$$\mathcal{B}_p = (x_p, y_p, w_p, h_p) \qquad \mathcal{R} = \{(x_k, y_k)\}_{k=1}^{n},$$

$$\mathcal{B}_r = (x_p + w_p\Delta x_p, y_p + h_p\Delta y_p, w_p e^{\Delta w_p}, h_p e^{\Delta h_p}). \qquad \mathcal{R}_r = \{(x_k + \Delta x_k, y_k + \Delta y_k)\}_{k=1}^{n},$$

bbox anchors $\xrightarrow{\text{bbox reg.}}$ bbox proposals (S1) $\xrightarrow{\text{bbox reg.}}$ bbox proposals (S2) ... $\xrightarrow{\text{bbox reg.}}$ bbox object targets

object centers $\xrightarrow{\text{RP refine}}$ RepPoints proposals (S1) $\xrightarrow{\text{RP refine}}$ RepPoints proposals (S2) ... $\xrightarrow{\text{RP refine}}$ RepPoints object targets



RPDet: an anchor-free object detector based on RepPoints

Bounding box vs. RepPoints

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.

Studies on assigner, supervision and anchors for RepPoints

Method	AP	AP_{50}	AP_{75}
Single anchor	36.9	58.2	39.7
Center point	**38.3**	**60.0**	**41.1**

Representation	Supervision		AP	AP_{50}	AP_{75}
	loc.	rec.			
bounding box	✓		36.2	57.3	39.8
	✓	✓	36.2	57.5	39.8
RepPoints		✓	33.8	54.3	35.8
	✓		37.6	59.4	40.4
	✓	✓	**38.3**	**60.0**	**41.1**

Table 2. Ablation of the supervision sources, for both bounding box and RepPoints based object detection. "loc." indicates the object localization loss. "rec." indicates the object recognition loss from the next detection stage.

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.

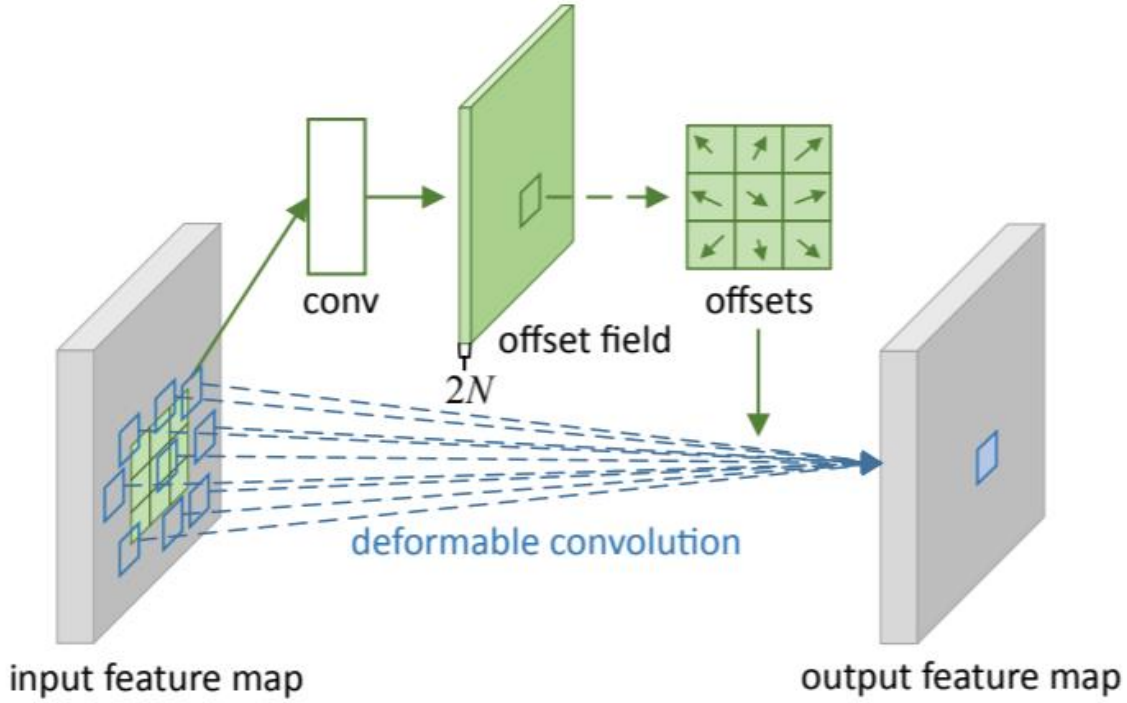
System level comparison

	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**

	Bounding box	RepPoints
Definition	$\mathcal{B} = (x, y, w, h)$	$\mathcal{R} = \{(x_k, y_k)\}_{k=1}^{n}$
Regression	$\mathcal{B}_r = (x_p + w_p \Delta x_p, y_p + h_p \Delta y_p, w_p e^{\Delta w_p}, h_p e^{\Delta h_p})$	$\mathcal{R}_r = \{(x_k + \Delta x_k, y_k + \Delta y_k)\}_{k=1}^{n}$
Feature Extraction	RoIAlign	Deformable convolution
Transformation	N/A	Via pseudo bbox
Supervision	Localization	Localization and recognition
Initial State	Anchors	Center points
Intermediate States	bbox proposals	RepPoints proposals
Final State	bbox target	RepPoints target

Discussion: some thoughts on RepPoints

Discussion A: Interpretable Deformation Modeling



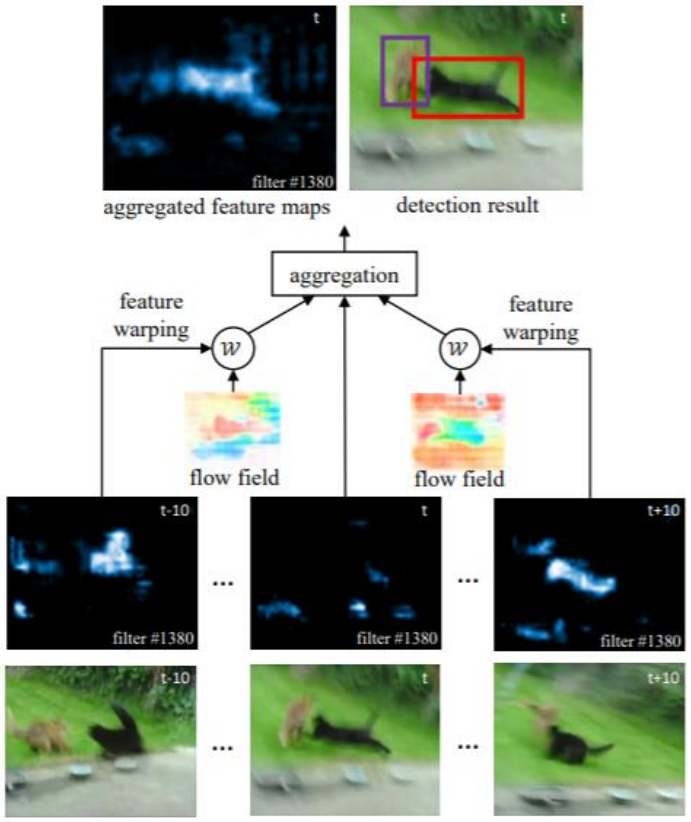
Deformable Convolutional Networks [2]

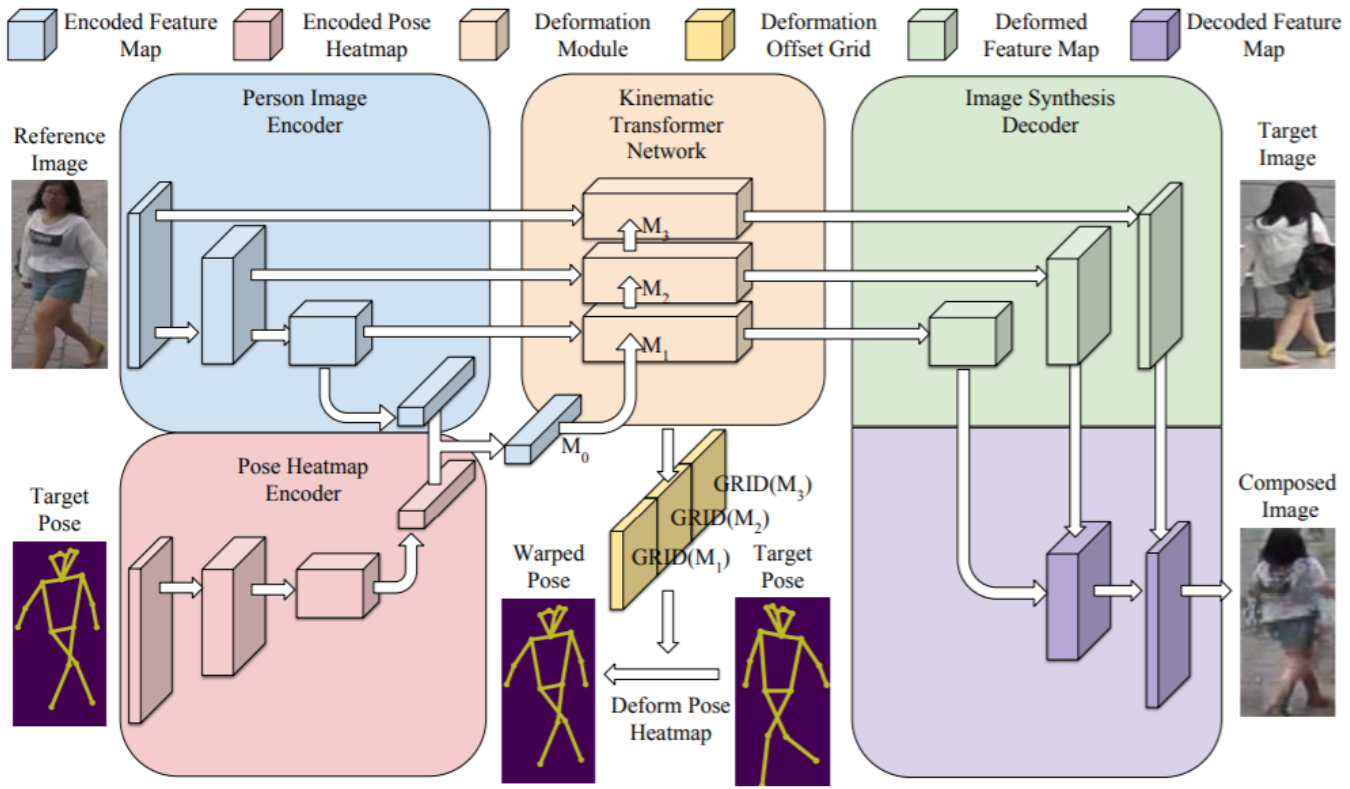
Only using recognition feedback in an implicit manner & Lacking geometric interpretation on the learned offset.

Discussion A: Interpretable Deformation Modeling

RepPoints: deformation modeling with explicit geometric interpretation.

Discussion B. Extending RepPoints: Denser and Finer





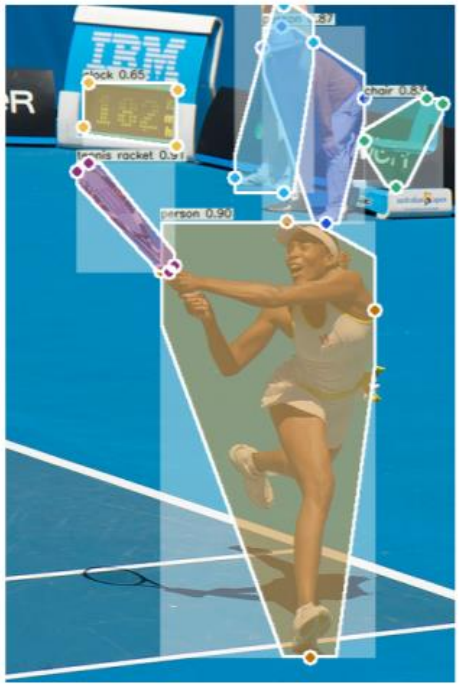
Zhu et al. Flow-guided feature aggregation.

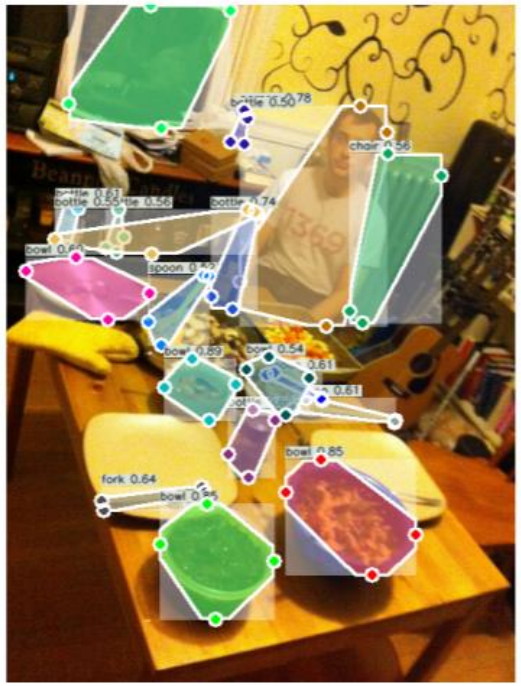
Zhang et al. Pose-guided image generation, project at Upenn.

Related Work: Deformation modeling for frame-to-frame correspondence in videos.

Discussion B. Extending RepPoints: Denser and Finer

- Possible direction for extension: dense object perception.





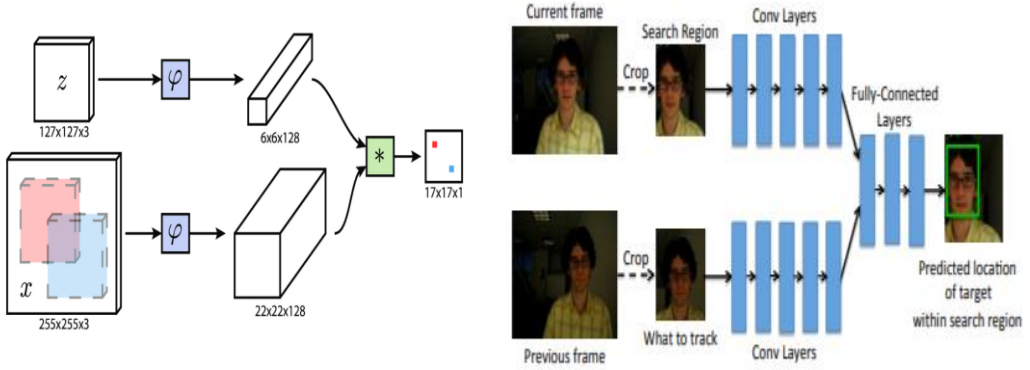
Segmentation (From Zhou et al. ExtremeNet)

Semantic Correspondence (From Novotny et al. AnchorNet)

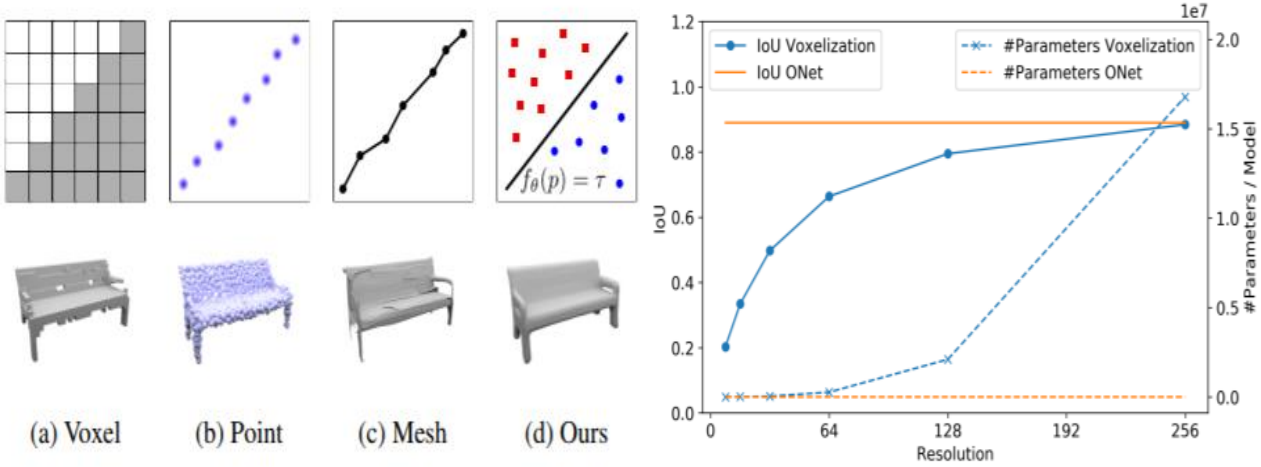
Bottleneck: to design effective and efficient guidance on RepPoints.

Discussion C. Regression vs. Classification

Another bottleneck: the localization ability of regression methods are lower than classification methods.



[6] discrimination vs. [7] regression



regression vs. discrimination : occupancy networks [8]

e.g. Object Tracking: reg is more efficient

e.g. 3D reconstruction: reg has higher resolution

Regression is relatively more efficient and does not need predefined proposals, while classifying each pixel is more suitable for accurate localization. Combining regression with classification can potentially reduce time complexity and number of proposals.

Thanks!

b1ueber2y@gmail.com

[1] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, Stephen Lin. RepPoints: Point Set Representation for Object Detection. arxiv preprint arxiv: 1904.11490.

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei. Deformable Convolutional Networks. In ICCV 2017.

[3] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, Yichen Wei. Flow-Guided Feature Aggregation for Video Object Detection. In ICCV 2017.

[4] Xingyi Zhou, Jiacheng Zhuo, Philipp Krähenbühl. Bottom-up Object Detection by Grouping Extreme and Center Points. In CVPR 2019.

[5] David Novotny, Diane Larlus, Andrea Vedaldi. AnchorNet: A Weakly Supervised Network to Learn Geometry-sensitive Features For Semantic Matching. In CVPR 2017.

[6] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, Philip H. S. Torr. Fully-Convolutional Siamese Networks for Object Tracking. In CVPR 2017.

[7] David Held, Sebastian Thrun, Silvio Savarese. Learning to Track at 100 FPS with Deep Regression Networks. In ECCV 2016.

[8] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In CVPR 2019.