Centernet: Object as Points

Arithmer Dynamics Div. Christian Saravia

2020/08/28

Christian Martin Saravia Hernandez

- Graduate School
  - Universidad Peruana de Ciencias Aplicadas
  - Bs. Software Engineering
- Former Job
  - Ficha Inc
    - Algorithm Development Engineer
      - Research & application of deep learning in computer vision problems
- Current Job
  - Application of machine learning / deep learning to computer vision problems
    - Object detection
    - Object classification

Purpose of this material

- Understand an anchor free approach object detection algorithm

Agenda

- Current object detection approaches
- Centernet approach
- Object as Points
- Training
    - Keypoint heatmap
    - Local offset
    - Size prediction
    - Loss function
- Network Architecture
    - DLA
    - Modified DLA
- Inference
- Results

Current approaches

- Object detections model (such as Yolo, SSD, etc.) rely on the usage of anchor boxes

- Anchor boxes are not completely optimal:
  - Wasteful: SSD300 does 8732 detections per class, and yolo448 does 98 detections per class, which means that most of the box are discarded

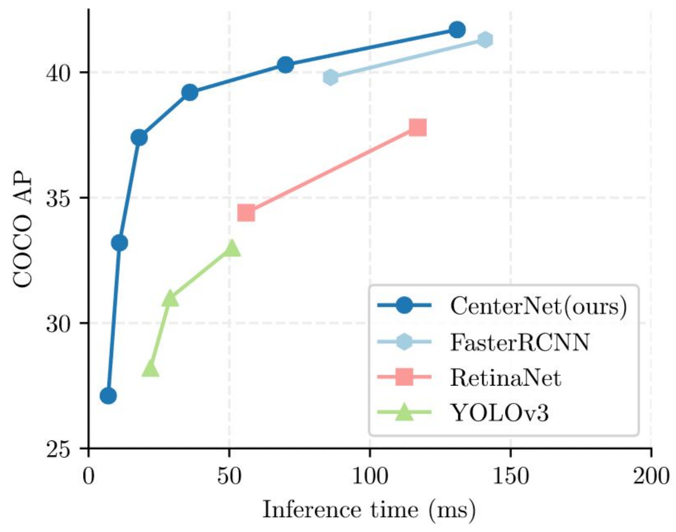
  - Inefficient: We have to process all the boxes (even we will discard them later), which comes with more processing time

  - Require post processing: like non-max suppression algorithm

  - Fixed: SSD requires fixed scale and steps of boxes, while yolov3 fixes the size of the anchors per detection level

Centernet approach

- End-to-end differentiable solution
- Relies on keypoint estimation to find the center points and regress all other object properties(such as size)
- As a result, the model is simpler, faster and more accurate than bounding-box based detectors



Speed-accuracy trade-off on COCO dataset

Object as Points

- Anchor box structure:

$$[b_{x1}, b_{y_1}, b_{x2}, b_{y_2}]$$

Where:

b_{x1}, b_{y1}: correspond to the x , y coordinates of the top-left corner.

b_{x2}, b_{y2}: correspond to the x , y coordinates of the bottom-right corner.

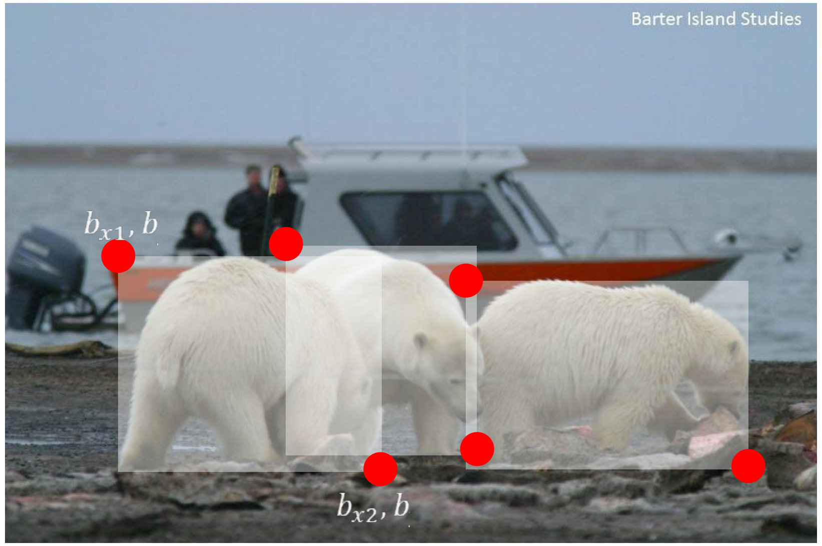
- Centernet proposal:

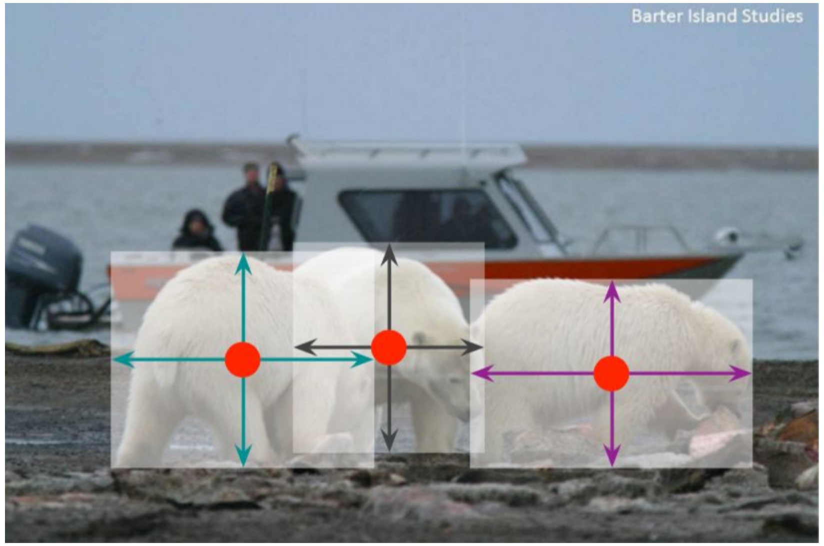
$$[c_x, c_y]$$

Where:

$$c_x = \left(\frac{b_{x1} + b_{x2}}{2}\right)$$

$$c_y = \left(\frac{b_{y1} + b_{y2}}{2}\right)$$





Training

- Let $I \in R^{W \times H \times 3}$ be an input image of width W and height H.
- The objective is to produce:
  - Keypoint heatmap $\hat{Y}$
  - Local offset $\hat{O}$
  - Size prediction $\hat{S}$

Keypoint Heatmap

- A keypoint heatmap $\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ is generated, where:
  C: number of keypoint types (classes)
  R: output stride
- The output stride downsamples the output prediction by a factor R. ($R = 4$ is default value)

$$\hat{Y}_{x,y,c} \begin{cases} 1, detected\ keypoint \\ 0, background \end{cases}$$

Arithmer

Keypoint Heatmap

- How to produce the ground truth for $\hat{Y}$?
  - Law and Deng: For each ground truth keypoint $p \in R^2$ of class c, we compute a low-resolution equivalent $\tilde{p} = \left\lfloor \frac{p}{R} \right\rfloor$. We then splat all ground truth keypoint onto a heatmap $Y \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ using a unnormalized gaussian kernel.

$$Y_{xyc} = \exp\left(-\frac{(x - \tilde{p}_x)^2 + \left(y - \tilde{p}_y\right)^2}{2\sigma_p^2}\right)$$

where:

σ_p^2: object size-adaptive standard deviation

* If two gaussians of the same class overlap, take the element-wise maximum

Loss calculation

- Pixel-wise logistic regression using focal loss:

$$L_k = -\frac{1}{N}\sum_{xyc}\begin{cases}\left(1-\hat{Y}_{xyc}\right)^{\alpha}\log\left(\hat{Y}_{xyc}\right) & if\ Y_{xyc}=1\\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha}\log\left(1-\hat{Y}_{xyc}\right) & otherwise\end{cases}$$

where:

α, β: hyperparameters of focal loss.($\alpha = 2, \beta = 4$)

N: number of keypoints

Motivation

- To recover the discretization error caused by the output stride (R), we predict a local offset $\hat{O} \in R^{\frac{W}{R} \times \frac{H}{R} \times 2}$ for each center point.
- All classes share the same offset prediction.

Loss calculation

- The offset is trained with L1 loss.

$$L_{off} = \frac{1}{N}\sum_{\rho} \left| \hat{O}_{\rho} - \left(\frac{p}{R} - \tilde{p} \right) \right|$$

where:

N: number of keypoints

* Remember $\tilde{p} = \left\lfloor \frac{p}{R} \right\rfloor$

Motivation

- Regress to the object size $S_k = \left(x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)}\right)$ for each object k.
- To avoid the computational burden, use a single size prediction $\hat{s} \in R^{\frac{W}{R} \times \frac{H}{R} \times 2}$for all object categories

Loss calculation

- The size prediction is trained with L1 loss.

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{s}_{P_k} - S_k\right|$$

where:

N: number of keypoints

- At calculation, the scale is not normalized and directly used raw pixel coordinates.

Loss Function

$$L_{\mathrm{det}} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off}$$

where:

λ_{size}: size loss's scale constant($\lambda_{size} = 0.1$).

λ_{off}: offset loss's scale constant($\lambda_{off} = 1$).

Network Architecture

- Authors experiment with different backbone architectures, obtaining different results:

	AP			AP_{50}			AP_{75}			Time (ms)			FPS		
	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS
Hourglass-104	**40.3**	**42.2**	**45.1**	**59.1**	**61.1**	**63.5**	**44.0**	**46.0**	**49.3**	71	129	672	14	7.8	1.4
DLA-34	37.4	39.2	41.7	55.1	57.0	60.1	40.8	42.7	44.9	19	36	248	52	28	4
ResNet-101	34.6	36.2	39.3	53.0	54.8	58.5	36.9	38.7	42.0	22	40	259	45	25	4
ResNet-18	28.1	30.0	33.2	44.9	47.5	51.5	29.6	31.6	35.1	**7**	**14**	**81**	**142**	**71**	**12**

Results without test augmentation(N.A.), flip testing(F), and multi-scale augmentation(MS). HW: Intel Core i7-8086k CPU, Titan Xp GPU

- The backbone that produces best speed/accuracy tradeoff is DLA-34 (modified by authors)

Network Architecture

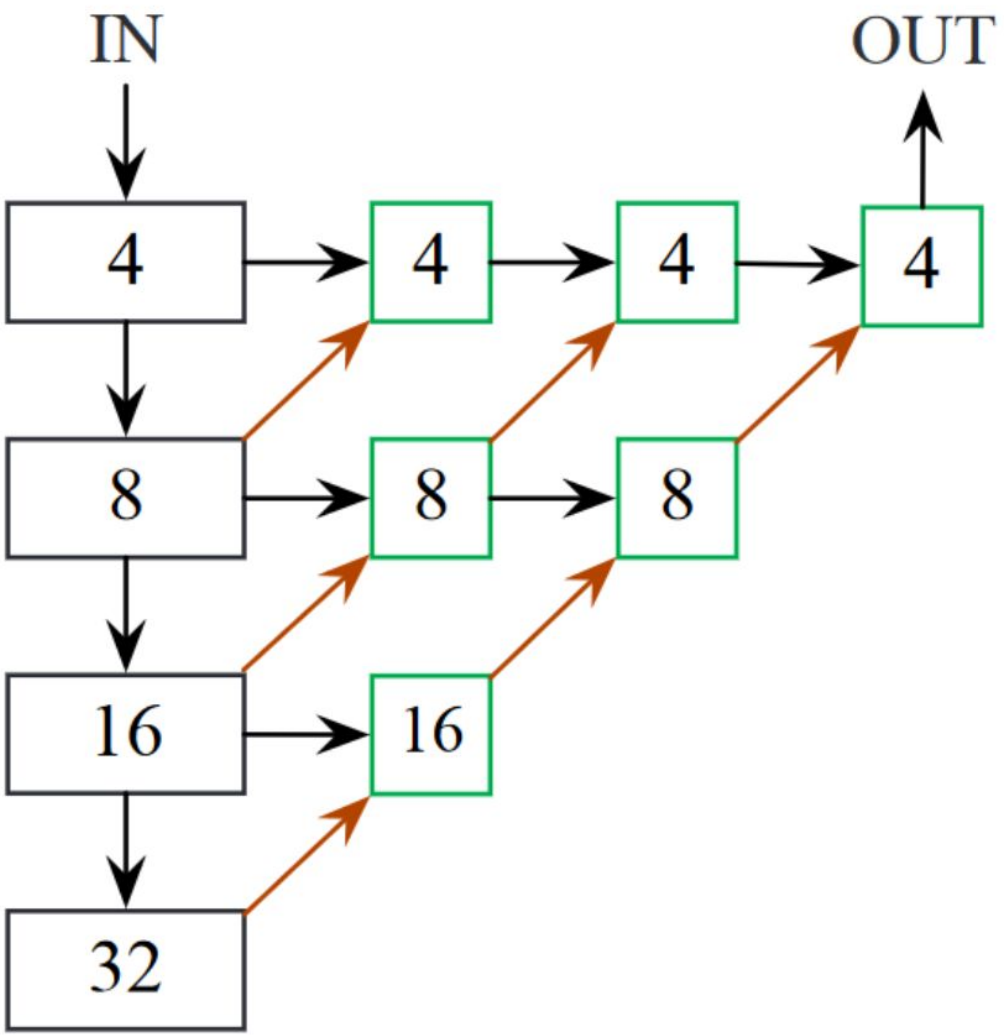
- Deep Layer Aggregation:

Iterative deep aggregation

Upsample 2x

Aggregation node:
Conv2d(1x1)
Batch Normalization
Relu

Stage node:
Conv2d(3x3)
Batch Normalization
Relu
Conv2d(3x3)
Batch Normalization
Relu



Network Architecture

- Modified Deep Layer Aggregation:

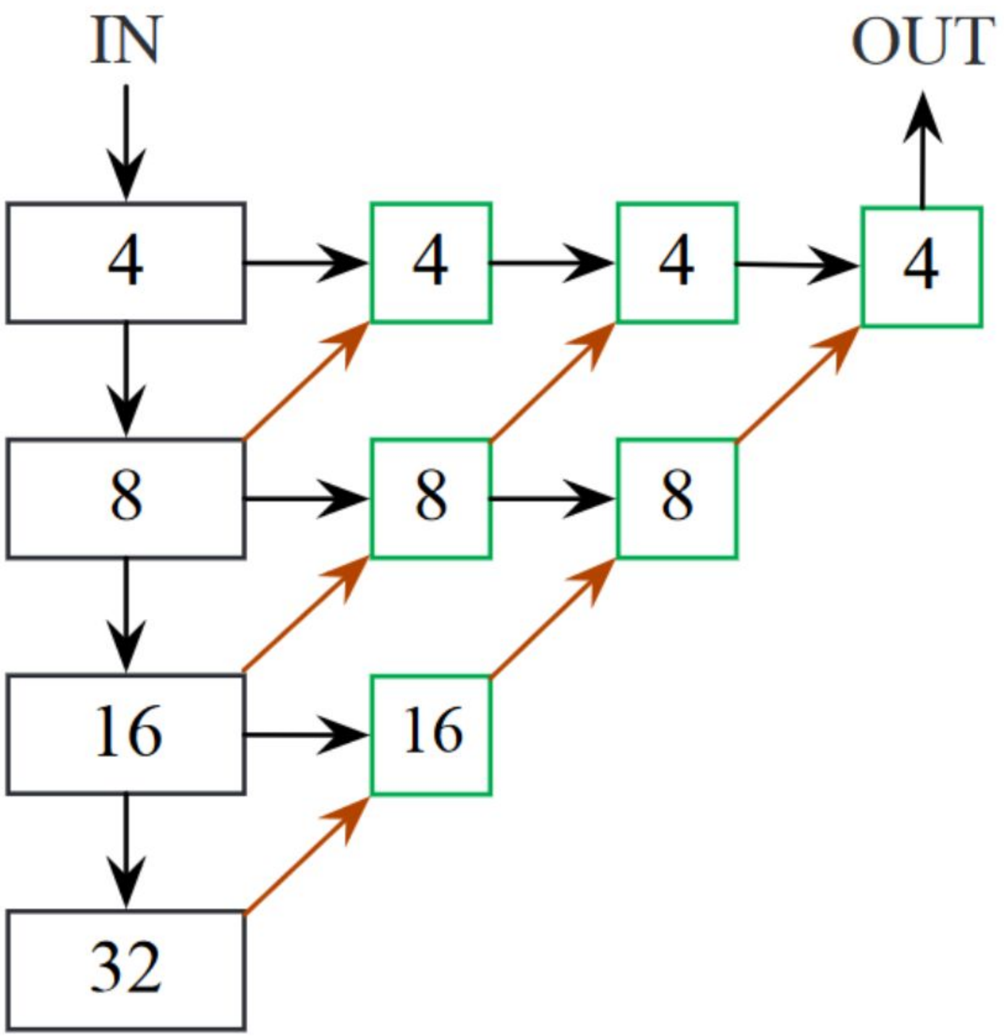
Iterative deep aggregation

Upsample 2x

Deformable convolution

Aggregation node:
Conv2d(1x1)
Batch Normalization
Relu

Stage node:
Conv2d(3x3)
Batch Normalization
Relu
Conv2d(3x3)
Batch Normalization
Relu



Points to boxes

- Extract peaks in the heatmap for each category independently.
- Detect all responses whose value is greater or equal to its 8-connected neighbors.
- Keep top 100 peaks
- Let $\hat{P} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{\eta}$ be the set of n detected center points of class c. Each keypoint top-left location is given by coordinates (x_i, y_i).

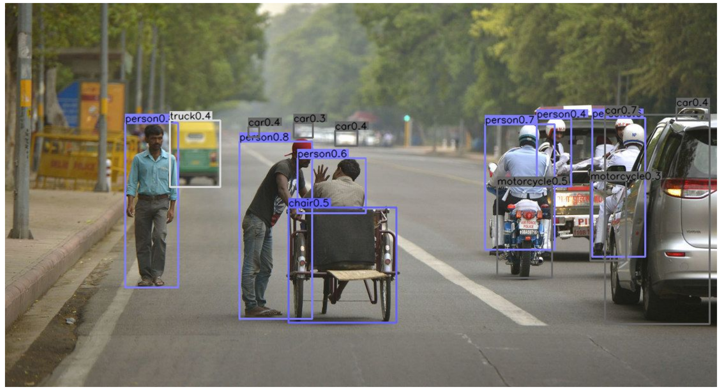
$$x_i = \hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2$$
$$y_i = \hat{y}_i + \delta\hat{y}_i - \hat{h}_\mathrm{i}/2$$

where:

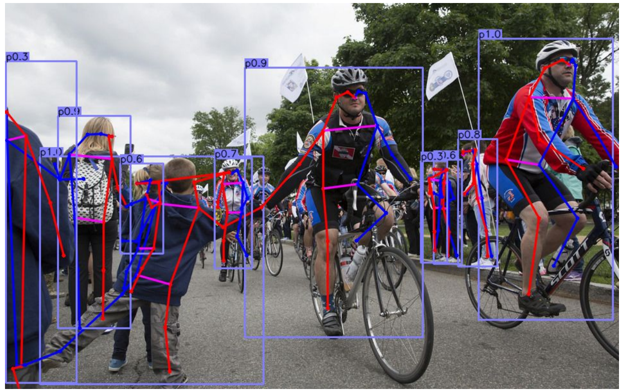
$$(\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i,\hat{y}_i}$$
$$(\hat{w}_\mathrm{i}, \hat{h}_i) = \hat{S}_{\hat{x}_i,\hat{y}_i}$$

Results

- Object detection



- Human pose estimation



Results

	Resolution	mAP@0.5	FPS
Faster RCNN [46]	600 × 1000	76.4	5
Faster RCNN* [8]	600 × 1000	79.8	5
R-FCN [11]	600 × 1000	80.5	9
Yolov2 [44]	544 × 544	78.6	40
SSD [16]	513 × 513	78.9	19
DSSD [16]	513 × 513	81.5	5.5
RefineDet [59]	512 × 512	81.8	24
CenterNet-Res18	384 × 384	72.6	142
CenterNet-Res18	512 × 512	75.7	100
CenterNet-Res101	384 × 384	77.6	45
CenterNet-Res101	512 × 512	78.7	30
CenterNet-DLA	384 × 384	79.3	50
CenterNet-DLA	512 × 512	80.7	33

Experimental results on Pascal VOC 2007 test. The results are shown in mAP@0.5. Flip test is used for Centernet. The FPSs for other methods are copied from the original publications

	Backbone	FPS	*AP*
MaskRCNN [21]	ResNeXt-101	**11**	39.8
Deform-v2 [63]	ResNet-101	-	46.0
SNIPER [48]	DPN-98	*2.5*	46.1
PANet [35]	ResNeXt-101	-	47.4
TridentNet [31]	ResNet-101-DCN	0.7	**48.4**
YOLOv3 [45]	DarkNet-53	20	33.0
RetinaNet [33]	ResNeXt-101-FPN	5.4	40.8
RefineDet [59]	ResNet-101	-	36.4 / 41.8
CornerNet [30]	Hourglass-104	4.1	40.5 / 42.1
ExtremeNet [61]	Hourglass-104	3.1	40.2 / 43.7
FSAF [62]	ResNeXt-101	*2.7*	**42.9** / 44.6
CenterNet-DLA	DLA-34	**28**	39.2 / 41.6
CenterNet-HG	Hourglass-104	7.8	42.1 / **45.1**

COCO test-dev. Frame-per-second (FPS) were measured on the same machine whenever possible. Italic FPS highlight the cases, where the performance measure was copied from the original publication

Arithmer 株式会社
〒106-6040
東京都港区六本木一丁目６番１号 泉ガーデンタワー 38/40F(受付)
03-5579-6683
https://arithmer.co.jp/