Objects as Points

이지민

Overview

1. Introduction
2. Related Work
3. Preliminary
4. Objects as Points
5. Implementation Details
6. Experiments
7. Conclusion

1 Introduction

2-Stage vs. 1-Stage Object Detection

- 2-stage detector
  1. Region proposal → RoI
  2. Classification + bounding box regression
- 1-stage detector
  1. Bounding boxes + object class를 동시에 predict함

→ Region proposal step을 생략

CenterNet

- 1-stage detector
  - Center point = object
  - Object size, dimension, 3D extent, orientation, pose → directly regress
- Input image → fully convolutional network → heatmap
  - Heatmap peaks = object centers
- Image features → height, width prediction

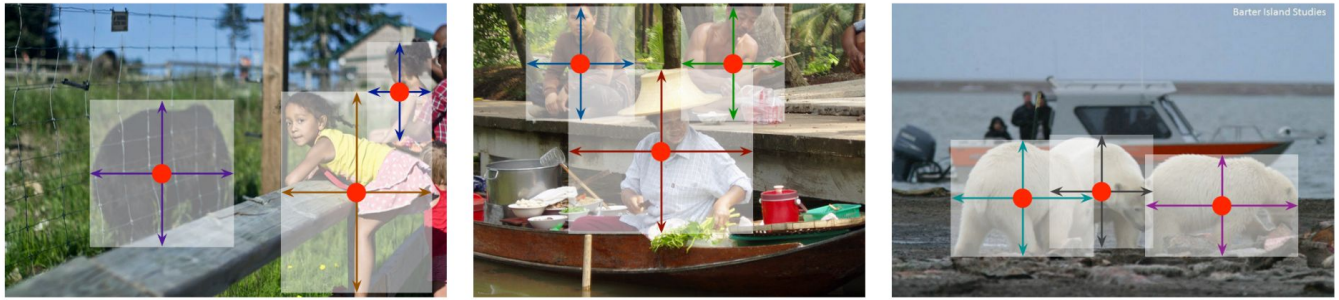


Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties are inferred from the keypoint feature at the center. Best viewed in color.

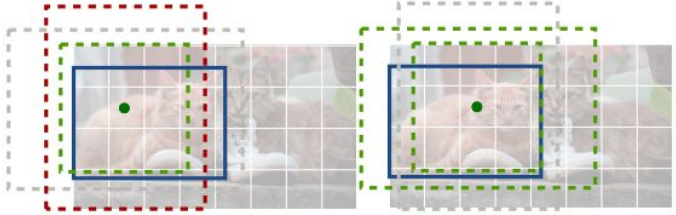
2 Related Work

Object Detection

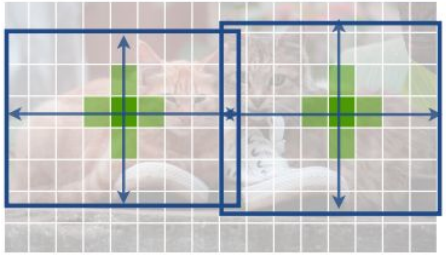
- Object detection by region classification
  - R-CNN: region candidates → crop proposals → object classification
  - Fast R-CNN: crop image features
- Object detection with implicit anchors
  - Faster R-CNN: region proposal → anchor boxes classification → object classification

Anchor-Based 1-Stage Approach vs. CenterNet

- Center point = single shape-agnostic anchor
- Based on location, not box overlap
- ONE positive “anchor” per object → Non-Maximum Suppression (NMS) 불필요
- Higher output resolution



(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.



(b) Center point based detection. The center pixel is assigned to the object. Nearby points have a reduced negative loss. Object size is regressed.

Figure 3: Different between anchor-based detectors (a) and our center point detector (b). Best viewed on screen.

Object Detection by Keypoint Estimation

- CornerNet: 2 bounding box corners = keypoints
- ExtremeNet: top-, left-, bottom-, rightmost + center points
- Combinatorial grouping stage → 느림
- 반면 CenterNet은 single center point만 뽑음 → grouping / post-processing 불필요

Monocular 3D Object Detection

- Deep3Dbox: R-CNN 2D detection + 3D estimation network
- 3D RCNN: Faster-RCNN + head + 3D projection
- CenterNet: 위와 같은 method들의 1-stage version

3 Preliminary

Image

$$I \in R^{W \times H \times 3}$$

Hourglass network
ResNet
Deep layer aggregation (DLA)

Heatmap

$$\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$$

Keypoint Prediction

$$\hat{Y}_{x,y,c} = 1$$

Detected keypoint

$$\hat{Y}_{x,y,c} = 0$$

Background

Training Method

1. Ground truth keypoint $p \in \mathcal{R}^2$ → low-resolution equivalent $\tilde{p} = \lfloor \frac{p}{R} \rfloor$
2. Gaussian kernel → heatmap $Y \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$
3. Predicted heatmap vs. GT heatmap → loss

Focal Loss

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

Offset L1 Loss

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

4 Objects as Points

Size L1 Loss

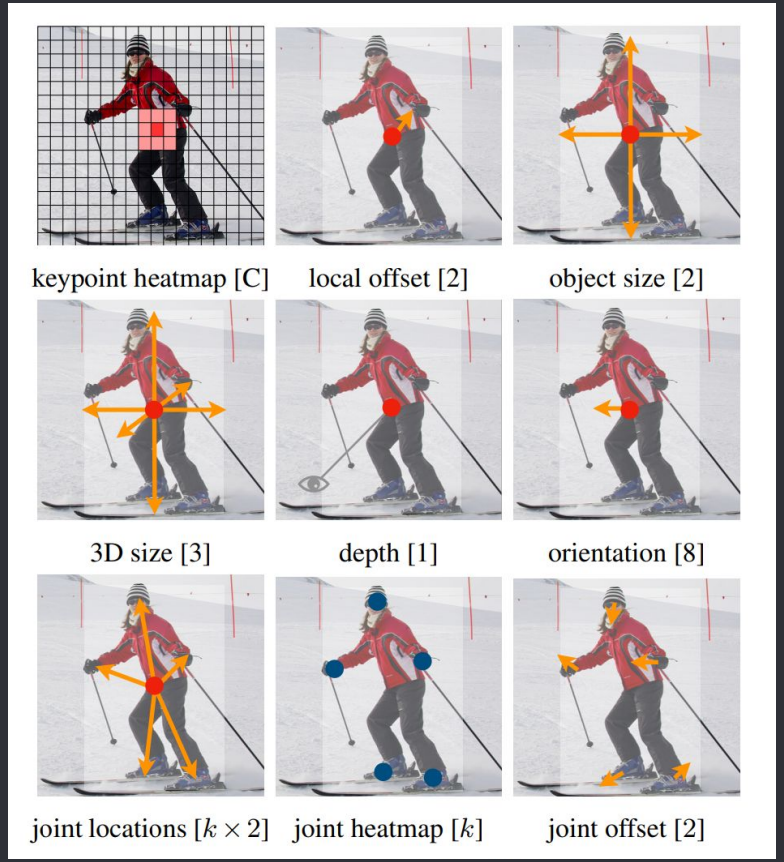
$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{p_k} - s_k \right|$$

Training Objective

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

Network

- Heatmap, offset, size
- Fully-convolutional backbone network


keypoint heatmap [C]
local offset [2]
object size [2]
3D size [3]
depth [1]
orientation [8]
joint locations [$k \times 2$]
joint heatmap [k]
joint offset [2]

From Points to Bounding Boxes

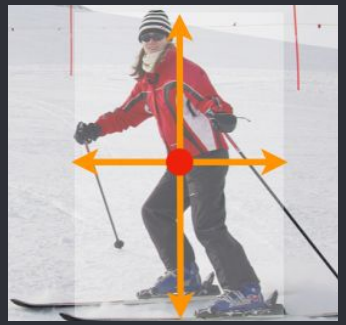
- Extract top 100 peaks
- $\hat{\mathcal{P}}_c$: set of *n* detected keypoints of class *c*

$$\hat{\mathcal{P}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n} \quad (\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i, \hat{y}_i} \quad (\hat{w}_i, \hat{h}_i) = \hat{S}_{\hat{x}_i, \hat{y}_i}$$

- Bounding box

$$\begin{aligned}(&\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2, \\ &\hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2).\end{aligned}$$

- IoU-based NMS 불필요



3D Detection

- Depth, 3D dimension, orientation → heads
- Depth를 directly regress하긴 어려움 → sigmoidal output transformation (Eigen *et al.*) 사용 → additional output channel로 계산
- Dimensions는 head사용해서 directly regress
- Orientation도 directly regress하긴 어려움 → 기존 방법 사용 (Mousavian *et al.*)

5 Implementation Details

4 Architectures

- Hourglass-104
- ResNet-18 (mod)
- ResNet-101 (mod)
- DLA-34 (mod)

Hourglass Network

- 5 layers down- and up-convolutional network
- Detect small features
- Best keypoint estimation performance

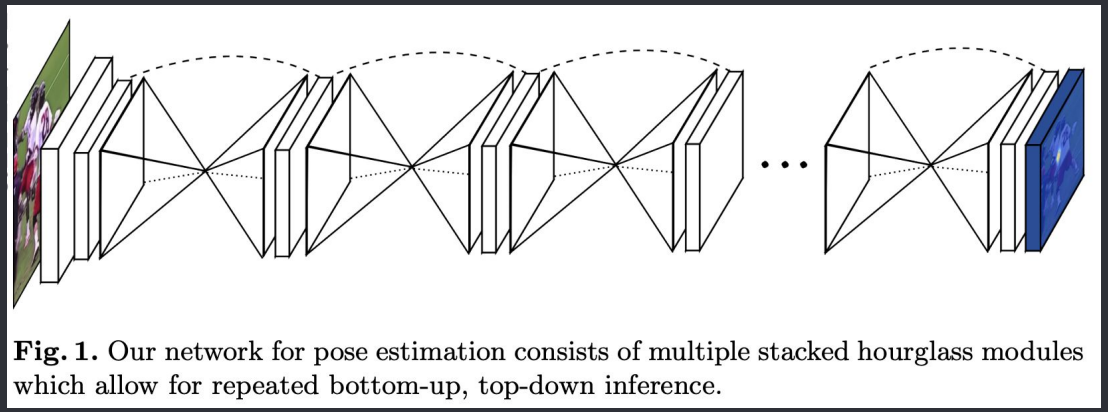


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

https://arxiv.org/abs/1603.06937

ResNet

- Output size ↓
- Deconvolution + deformable ConvNet

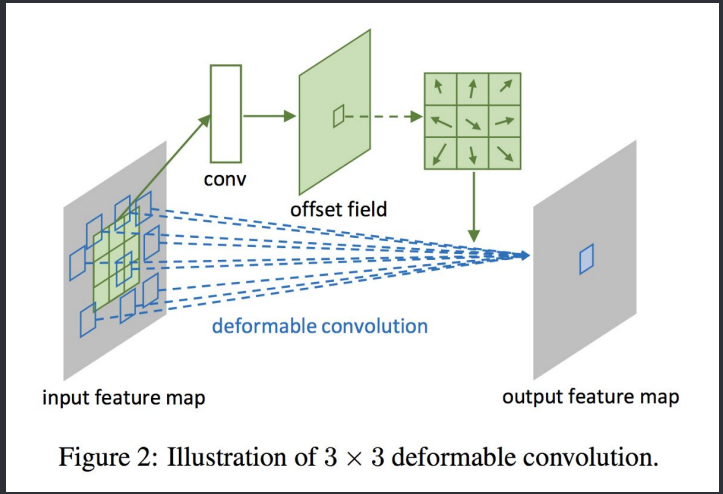


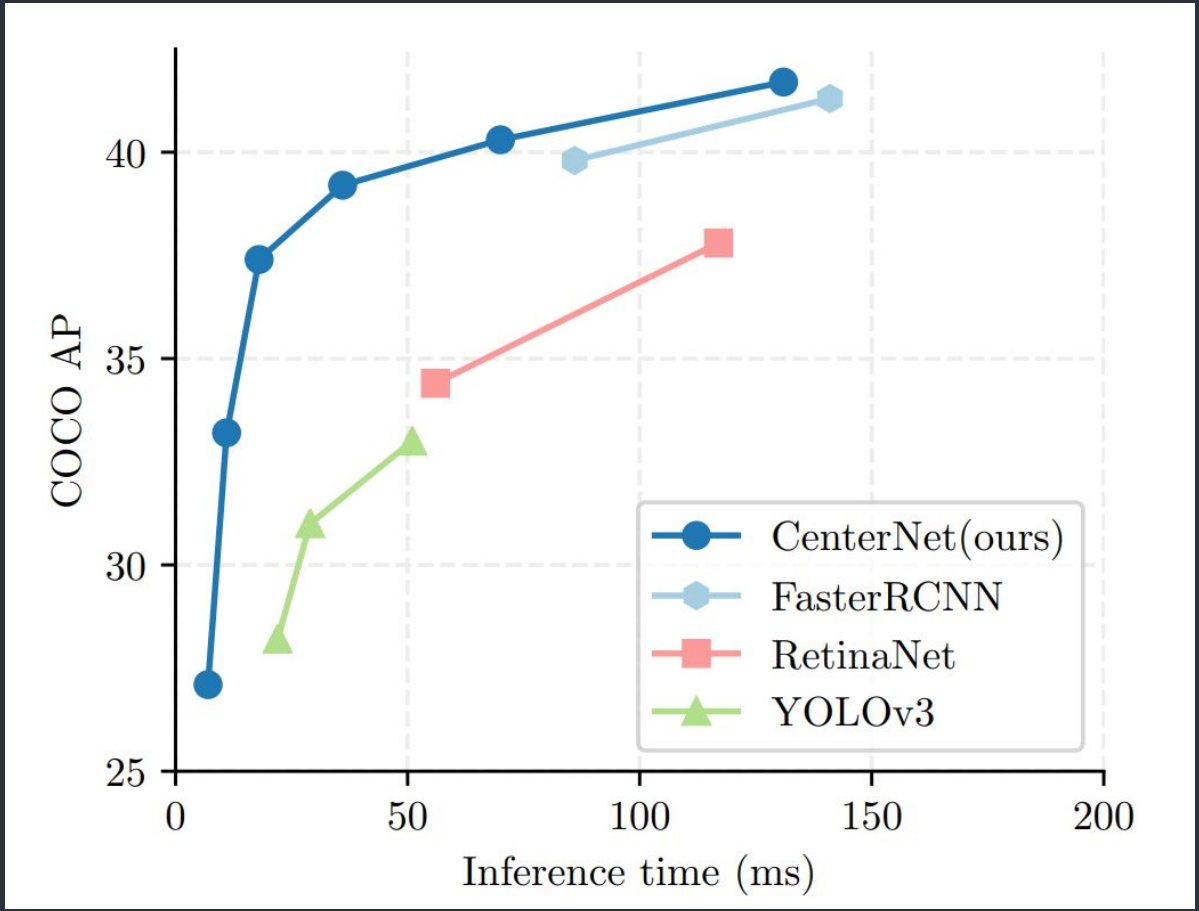
Figure 2: Illustration of 3×3 deformable convolution.

https://arxiv.org/abs/1703.06211

Deep Layer Aggregation (DLA)

- Network with skip connections
- Deformable ConvNet으로 augment
- Best speed-accuracy trade-off

6 Experiments


COCO AP
40
35
30
25
0
50
100
150
200
Inference time (ms)
CenterNet(ours)
FasterRCNN
RetinaNet
YOLOv3

State-of-the-Art Comparison

- CenterNet with Hourglass-104 outperforms all 1-stage detectors
- Sophisticated 2-stage detectors are more accurate but slower

3D Detection

- 3D bounding box estimation experiments on KITTI dataset
- DLA-34 backbone

	AP			AOS			BEV AP		
	Easy	Mode	Hard	Easy	Mode	Hard	Easy	Mode	Hard
Deep3DBox [38]	98.8	97.2	81.2	98.6	96.7	80.5	30.0	23.7	18.8
Ours	90.2±1.2	80.4±1.4	71.1±1.6	85.3±1.7	75.0±1.6	66.2±1.8	31.4±3.7	26.5±1.6	23.8±2.9
Mono3D [9]	95.8	90.0	80.6	93.7	87.6	78.0	30.5	22.4	19.1
Ours	97.1±0.3	87.9±0.1	79.3±0.1	93.4±0.7	83.9±0.5	75.3±0.4	31.5±2.0	29.7±0.7	28.1±4.6

Table 4: KITTI evaluation. We show 2D bounding box AP, average orientation score (AOS), and bird eye view (BEV) AP on different validation splits. Higher is better.

7 Conclusion

Summary

- Objects = points
- Keypoint estimation networks
- Directly regress to bounding box from center point
- Fast, simple algorithm
  - NMS, post-processing 불필요

Works Cited

[Paper]

Zhou, Xingyi, et al. “Objects as Points.” ArXiv.org, Cornell University, 25 Apr. 2019, https://arxiv.org/abs/1904.07850.

[Lecture]

https://www.youtube.com/watch?v=mDdpwe2xsT4

[Image]

Newell, Alejandro, et al. “Stacked Hourglass Networks for Human Pose Estimation.” ArXiv.org, Cornell University, 26 July 2016, https://arxiv.org/abs/1603.06937.

Dai, Jifeng, et al. “Deformable Convolutional Networks.” ArXiv.org, Cornell University, 5 June 2017, https://arxiv.org/abs/1703.06211.