Objects As Points - CenterNet

논문 : https://arxiv.org/pdf/1904.07850v2.pdf
official code : https://github.com/xingyizhou/CenterNet

기존 object detector - 많은 양의 bounding box(anchor) 예측 후, 이에 대한 Non-Maximum Suppression(NMS)를 Post-Processing으로 적용하여 Object Detection 수행

Centernet - 하나의 Object의 중심점을 하나의 Keypoint로 바라보는 Keypoint Estimation 문제로 치환한 CenterNet 구조 제안 -> NMS 적용 X / 속도 향상

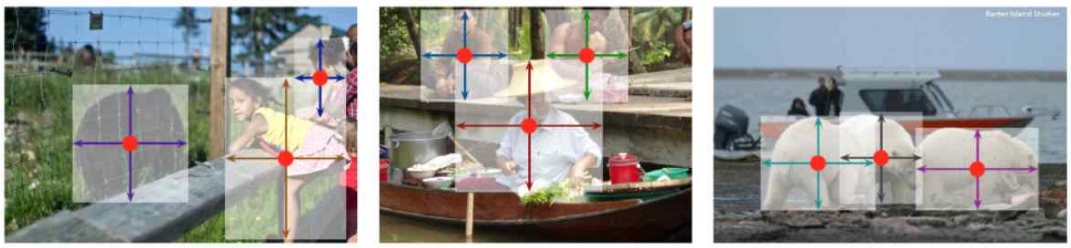
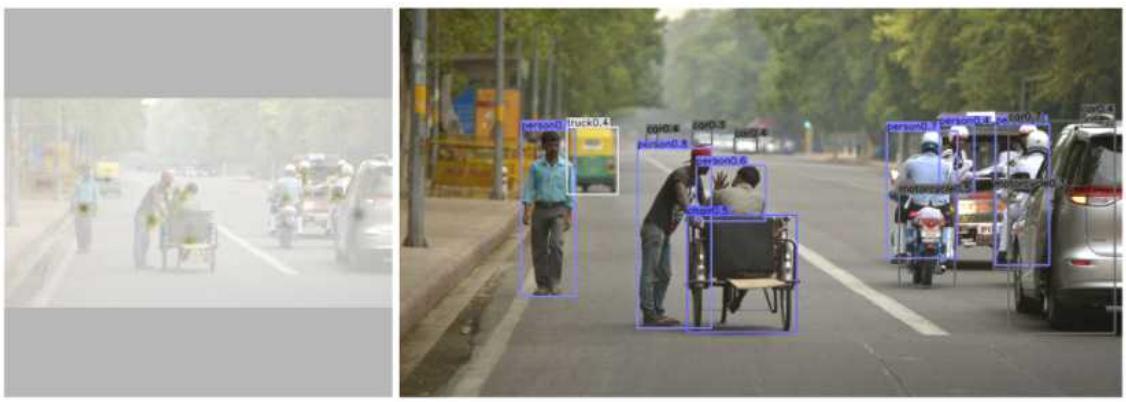


Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties are inferred from the keypoint feature at the center. Best viewed in color.

기존 Object Detection 모델과의 다른 점

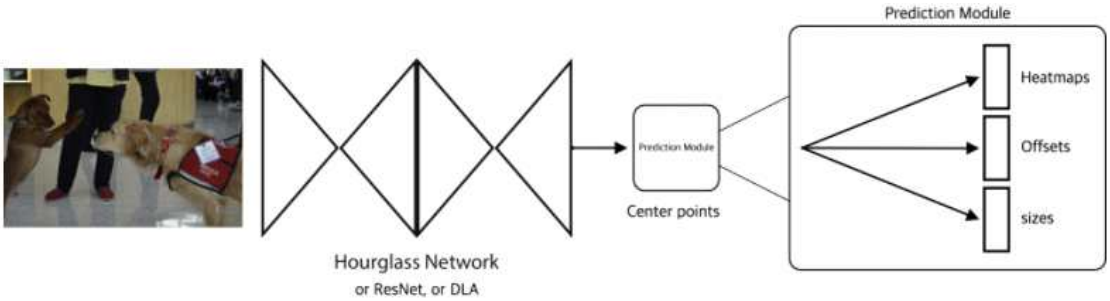
1) box overlap이 아닌 오직 위치만 가지고 "Anchor"를 할당합니다.
2) 하나의 "Anchor"만을 사용합니다.
3) 더 큰 output resolution (output stride of 4) 을 가집니다.



(왼쪽) Keypoint Estimation 으로 얻어진 Heatmap. (오른쪽) Object Detection 결과

Centernet에서는 KeyPoint Estimation을 사용하는데,
Centernet에서의 keypoint는 Object의 중심점이고, Centernet의 목적은 network를 통해 keypoint heatmap을 얻어내는데 있음

Centernet 구조



CenterNet은 **keypoints, offset, object size**를 predict 하기 위해서 하나의 네트워크를 사용합니다.

1) Keypoint

Loss Function : Focal Loss

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

$\hat{Y}_{x,y,c} = 1$ corresponds to a detected keypoint

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

N is the number of keypoints in image I

$\alpha = 2$ and $\beta = 4$

Keypoint 학습에는 Hard Positives (keypoint) << Easy Negatives (background)에
적합한 RetinaNet의 **Focal Loss**를 사용하였습니다.

2) Offsets

Loss Function : L1 Loss

$$L_{off} = \frac{1}{N} \sum_{p} \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right| . \quad (2)$$

$$\boldsymbol{o}_k = \left(\frac{x_k}{n} - \left\lfloor \frac{x_k}{n} \right\rfloor , \frac{y_k}{n} - \left\lfloor \frac{y_k}{n} \right\rfloor \right) \text{ from CornerNet}$$

이미지가 Network를 통과하면 output의 사이즈가 보통 이미지보다 줄기 때문에 조정해주는 변수인 **Offset** 사용
예측된 heatmap에서 keypoint들의 위치를 다시 input image로 remapping 할 때, output의 사이즈가 줄어들게 되면 정확성이 떨어질 가능성이 있음

Offset 학습의 loss function은 **L1 Loss** 사용했습니다.

3) Object Size

Loss Function : L1 Loss

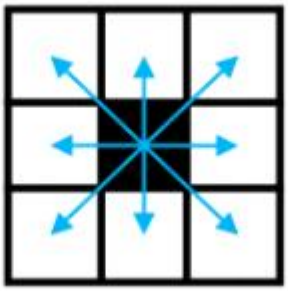
$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{p_k} - s_k \right| . \quad (3)$$

$(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}, y_2^{(k)})$ Object k의 bounding box 좌표

$s_k = (x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)})$

CenterNet은 측정한 keypoint로부터 추가적으로 object size를 regress합니다.
Object size 학습에는 **L1 Loss**를 사용하였습니다.

- Bounding Box



Centernet은 heatmap으로부터 각 category 마다 peaks를 뽑아냄
-> heatmap에서 주변 8개 pixel보다 값이 크거나 같은 중간값들은 모두 저장하고 값이 큰 100개의 peak들은 남겨둠

$$\hat{\mathcal{P}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n} \text{ of class } c.$$

뽑아낸 peaks(keypoint)의 위치는 정수 형태인 (x,y)로 나타내고, 이를 통해 bounding box의 좌표를 아래 식과 같이 나타냄

$$(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2,$$
$$\hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2, \quad \hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2)$$

Offset Prediction $(\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i, \hat{y}_i}$

Size Prediction $(\hat{w}_i, \hat{h}_i) = \hat{S}_{\hat{x}_i, \hat{y}_i}$

bounding box의 좌표

모든 output들이 single keypoint estimation으로부터 나왔습니다.