Objects as Points

Xingyi Zhou, Dequan Wang, Philipp Krähenbühl

Processing Image in Topics Selected

Final Project

Under the supervision of Prof. Yitzhak Yitzhaky

Students:

Hod Elmakayes, Lior Yaacov

Agenda

Introduction	Project introduction and problem statement	1
Challenges	Project challenges and solutions	2
Related Work	Current methods for object detection	3
Method	CenterNet method Dive-in	4
Examples	Real-life examples	5
Conclusions	CenterNet novelty and limitations	6

Introduction

Motivation

Object detection is used for vision tasks such as segmentation, pose estimation, tracking and action recognition.
It has down-stream applications in surveillance, autonomous driving, and visual question answering.

Problem Statement

Some of current methods are wasteful as they process the data multiple times. In addition, they require post processing.

Solution

Representing objects by a single point at their bounding box center enables simplicity and efficiently.

Challenges

Challenges

1. **Simplicity**
   Sliding window based object detectors are wasteful, as they need to enumerate all possible object locations and dimensions

2. **Inference Time**
   Separating region proposal and feature extraction steps leads to increased computational complexity and longer inference times

3. **Applications**
   Most traditional methods are limited to object detection tasks only

Solutions

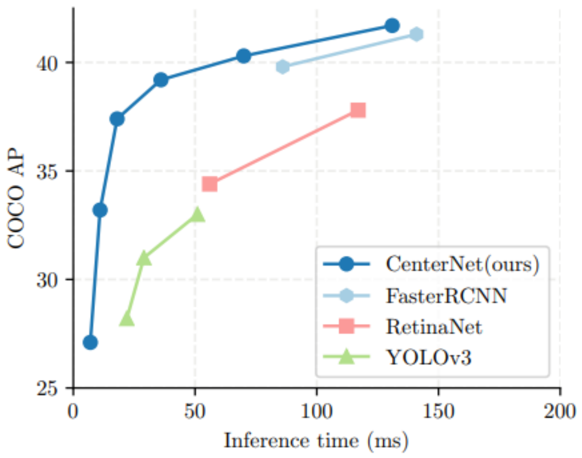
1. **Simplicity**
   Representing objects by a single point at their bounding box center, where other properties are regressed directly from image features at the center location

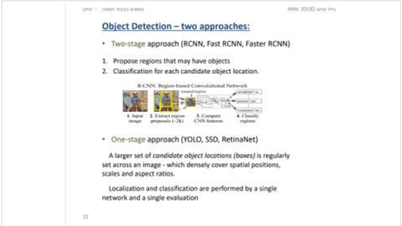
2. **Inference Time**
   CenterNet eliminates the need for separate region proposal and feature extraction steps, significantly reducing computational overhead and resulting in faster inference times.

3. **Applications**
   CenterNet method can be expanded to tasks like 3D detection and human pose estimation

Related Work

- One Stage Detectors:
  - ➢ YOLO
  - ➢ SSD
  - ➢ RetinaNet
- Two Stage Detectors:
  - ➢ RCNN
  - ➢ Fast RCNN
  - ➢ Faster RCNN





Object Detection – two approaches:

- Two-stage approach (RCNN, Fast RCNN, Faster RCNN)

1. Propose regions that may have objects
2. Classification for each candidate object location.

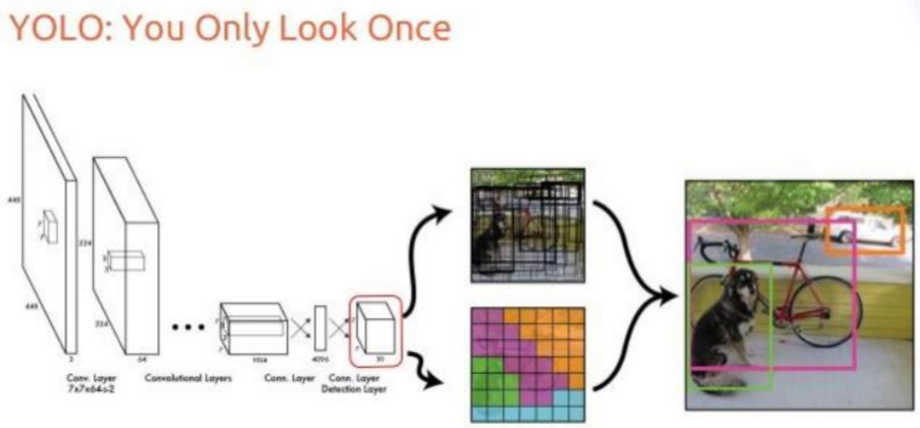
R-CNN: Region-based Convolutional Network

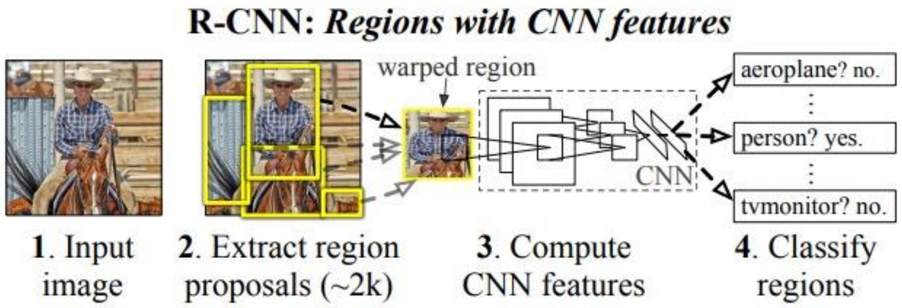
- One-stage approach (YOLO, SSD, RetinaNet)

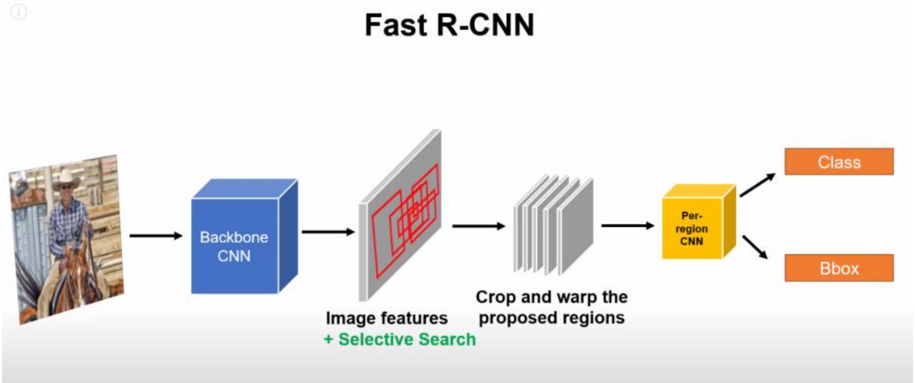
A larger set of *candidate object locations (boxes)* is regularly set across an image - which densely cover spatial positions, scales and aspect ratios.

Localization and classification are performed by a single network and a single evaluation

22



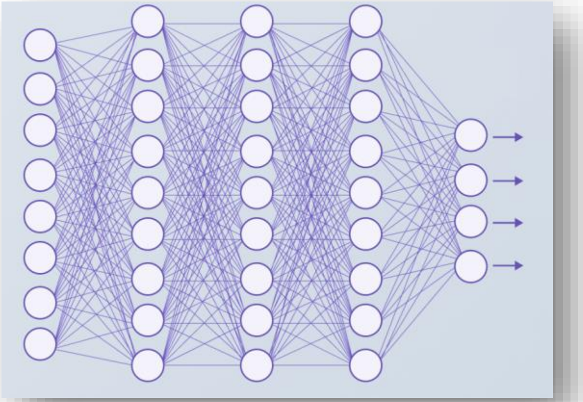




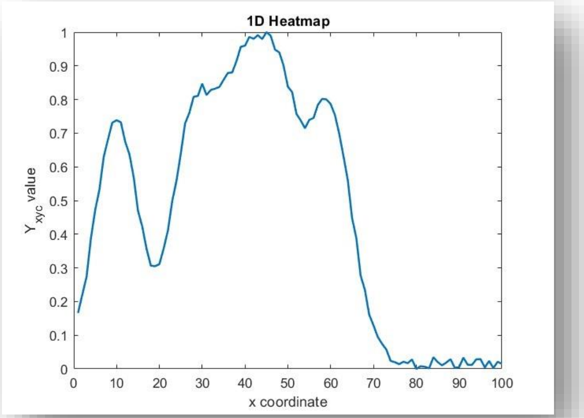
Method – Object Detection

RGB Image

CNN Network



Heatmaps



Coordinates

$$\{(x_i, y_i)\}_{i=1}^{n}$$

Offset

$$O_{x_i,y_i} = (\delta x_i, \delta y_i)$$

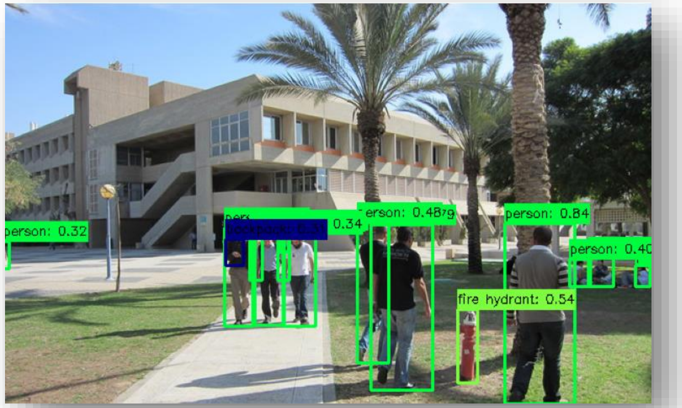
Dimension

$$S_{x_i,y_i} = (w_i, h_i)$$

Bounding Box Location

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

Output



Loss Functions – Object Detection

Heatmap Loss

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

*

Dimension Loss

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

Offset Loss

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

Total Loss

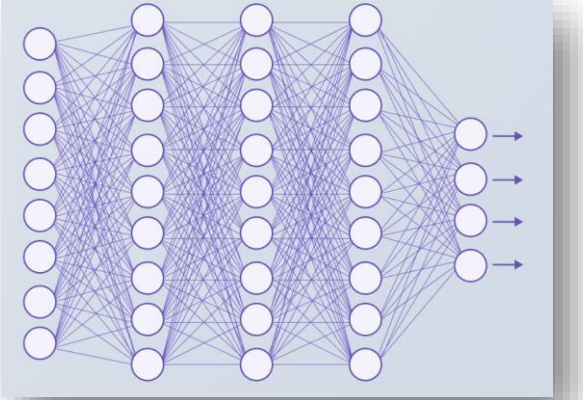
$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off}$$

*H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In ECCV, 2018.

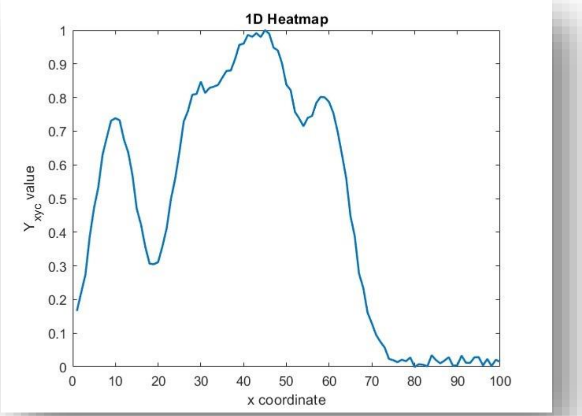
Method – 3D Bounding Box

RGB Image

CNN Network



Heatmaps



Depths

$\{d_i\}_{i=1}^{n}$

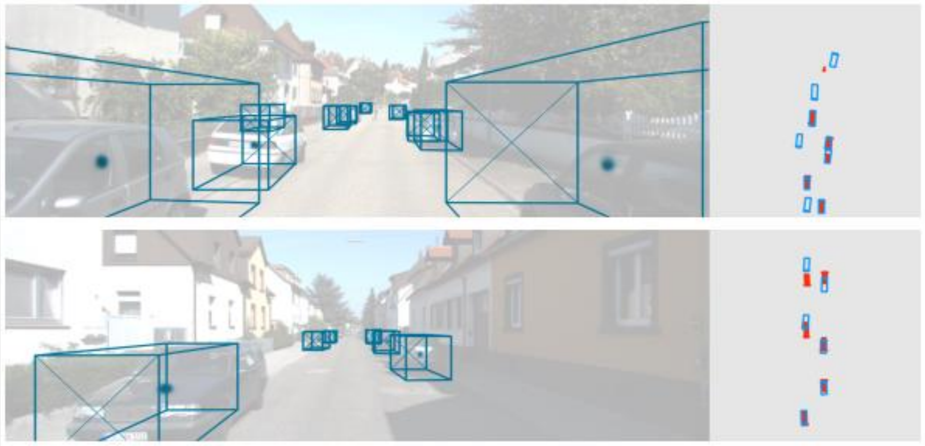
3D Dimension

$\{(w_i, h_i, l_i)\}_{i=1}^{n}$

Orientation

$\{\alpha_i\}_{i=1}^{n}$

3D Bounding Box Location



Loss Functions – 3D Bounding Box

Depth Loss

$$L_{dep} = \frac{1}{N}\sum_{k=1}^{N} \left|\frac{1}{\sigma(\hat{d}_k)} - 1 - d_k\right|$$

Dimension Loss

$$L_{dim} = \frac{1}{N}\sum_{k=1}^{N} |\hat{\gamma}_k - \gamma_k|$$

Orientation Loss

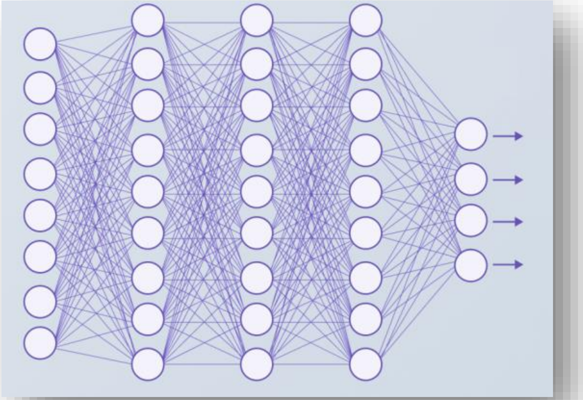
$$L_{ori} = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{2} (softmax(\hat{b}_i, c_i) + c_i|\hat{a}_i - a_i|)$$

where $c_i = \mathbb{1}(\theta \in B_i)$, $a_i = (\sin(\theta - m_i), \cos(\theta - m_i))$ $\hat{\theta} = arctan2(\hat{a}_{j1}, \hat{a}_{j2}) + m_j$

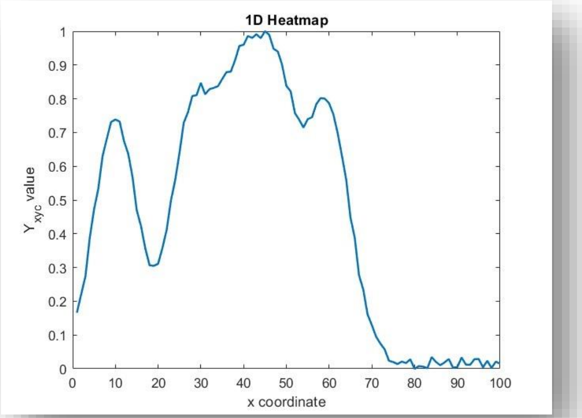
Method – Human Pose Estimation

RGB Image

CNN Network



Heatmaps



Human Joint Heatmap

$\{\phi_i\}_{i=1}^{n}$

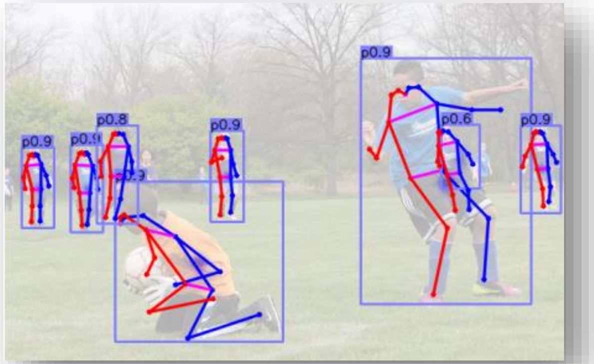
Joint Locations

$\{J\}_{i=1}^{n}$

Joint Offset

$O_{x_i,y_i,k} = (\delta x_i, \delta y_i)$

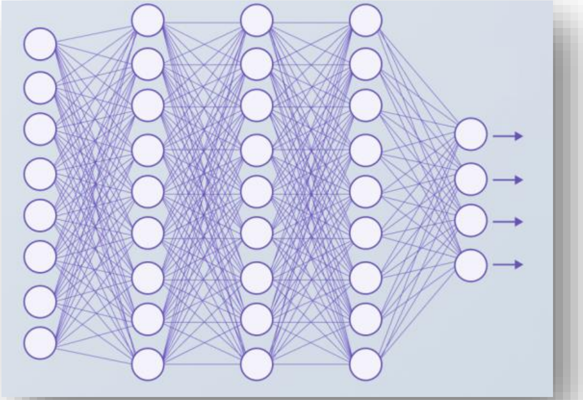
Human Pose Estimation



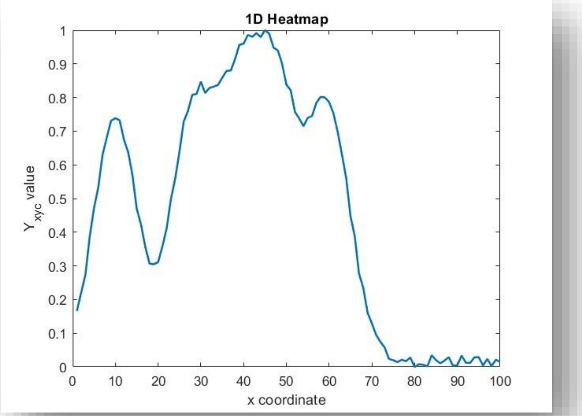
Method – Human Pose Estimation

RGB Image

CNN Network



Heatmaps



Human Joint Heatmap

$\{\phi_i\}_{i=1}^{n}$

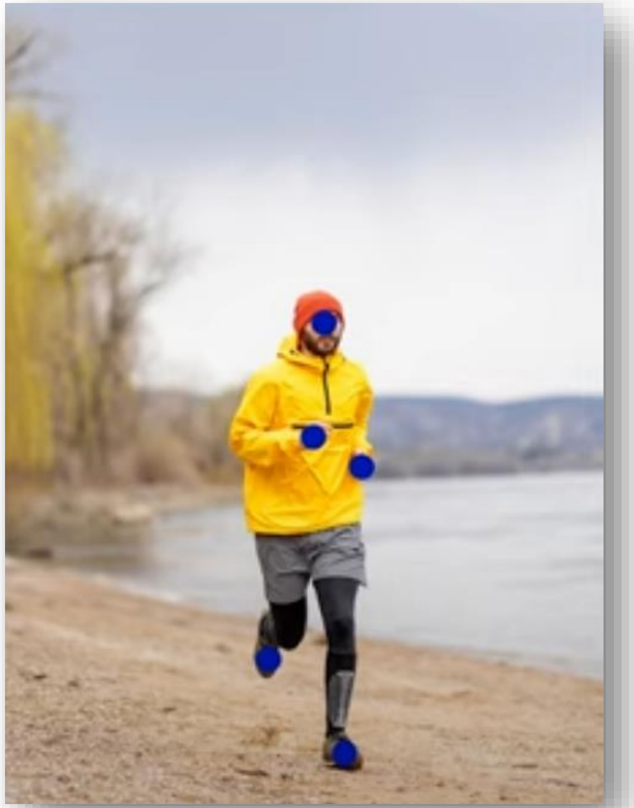
Joint Locations

$\{J\}_{i=1}^{n}$

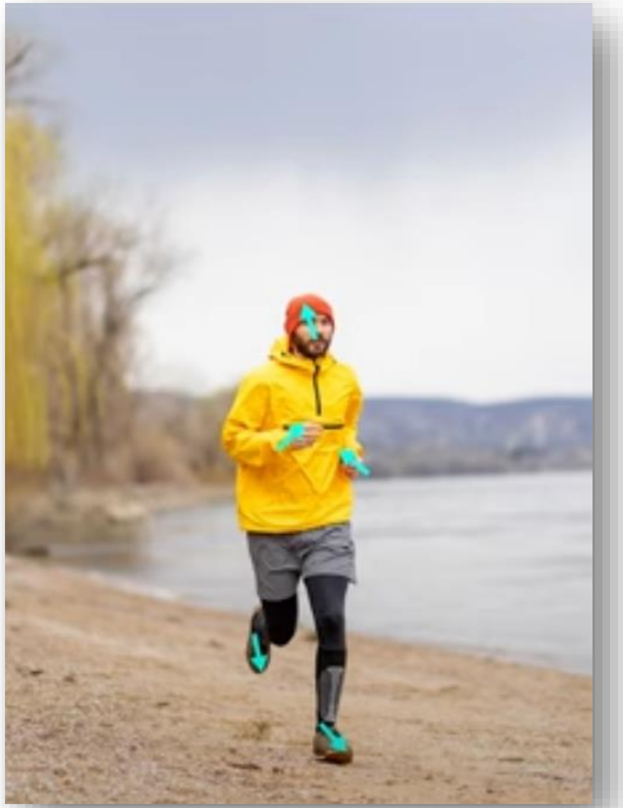
Joint Offset

$O_{x_i,y_i,k} = (\delta x_i, \delta y_i)$

Method – Human Pose Estimation



Human Joint Heatmap



Joint Offset

Human Joint Heatmap

$\{\phi_i\}_{i=1}^{n}$

Joint Locations

$\{J\}_{i=1}^{n}$

Joint Offset

$O_{x_i,y_i,k} = (\delta x_i, \delta y_i)$

Method – Human Pose Estimation

Human Joint Heatmap

Joint Locations

Human Joint Heatmap

$\{\phi_i\}_{i=1}^{n}$

Joint Locations

$\{J\}_{i=1}^{n}$

Joint Offset

$O_{x_i,y_i,k} = (\delta x_i, \delta y_i)$

Loss Functions – Human Pose Estimation

Joint Heatmap Loss

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

Joint Location Loss

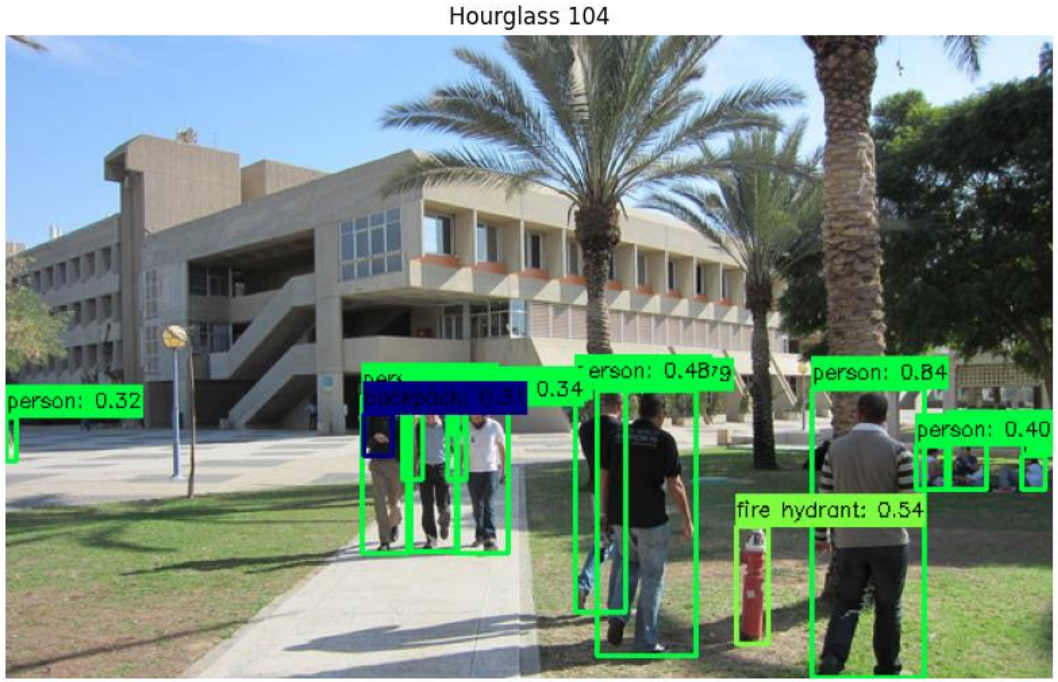
$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i|$$

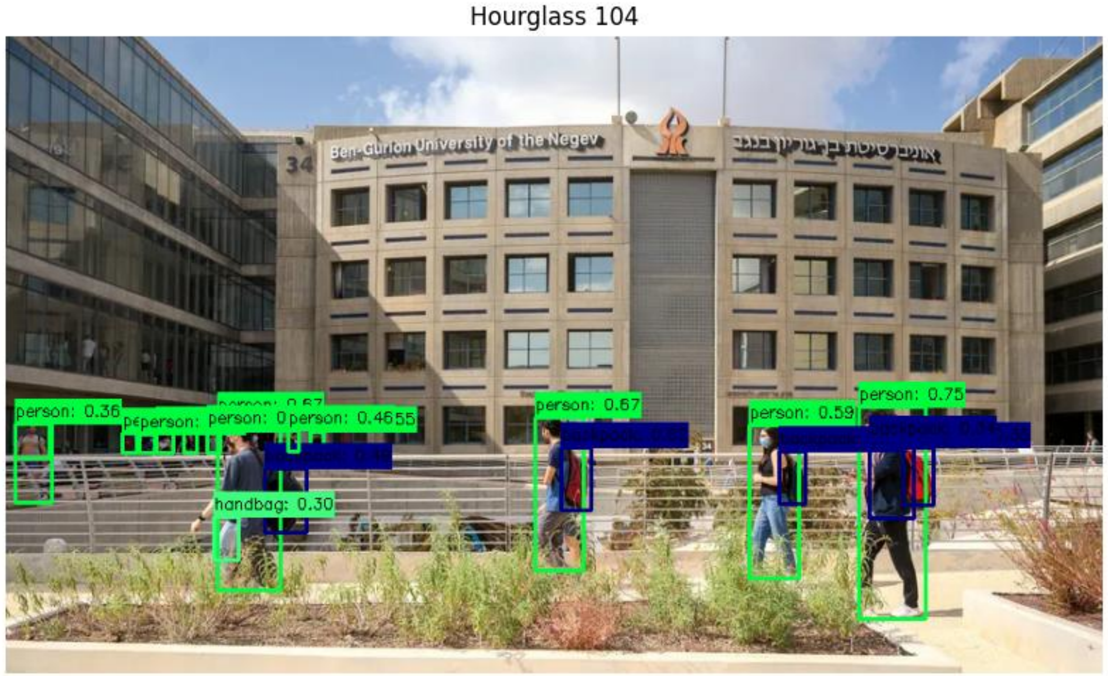
Joint Offset Loss

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

Examples

- Real life examples from Ben-Gurion university





Conclusions

- The paper introduces CenterNet, a method that models objects as single points and uses keypoint estimation to find the center points of bounding boxes
- CenterNet achieves better speed-accuracy trade-off compared to bounding box based detectors, with 28.1% AP at 142 FPS, 37.4% AP at 52 FPS, and 45.1% AP with multi-scale testing at 1.4 FPS on the MS COCO dataset
- The method is also applied to estimate 3D bounding boxes in the KITTI benchmark and human pose on the COCO keypoint dataset, performing competitively with multi-stage methods and running in real-time
- This method is limited to only 100 objects per image
- Overall, the paper concludes that CenterNet, with its center point based approach and keypoint estimation, offers a simpler, faster, and more accurate alternative to traditional bounding box based object detectors

End

Thanks for listening!