Computer Vision

第十一周 目标检测 下

庞彦
yanpang@gzhu.edu.cn

01

Stacked Hourglass Networks

堆叠沙漏网络

Stacked Hourglass Networks

Stacked Hourglass Networks for Human Pose Estimation

Alejandro Newell, Kaiyu Yang, and Jia Deng

University of Michigan, Ann Arbor
{alnewell,yangky,jiadeng}@umich.edu

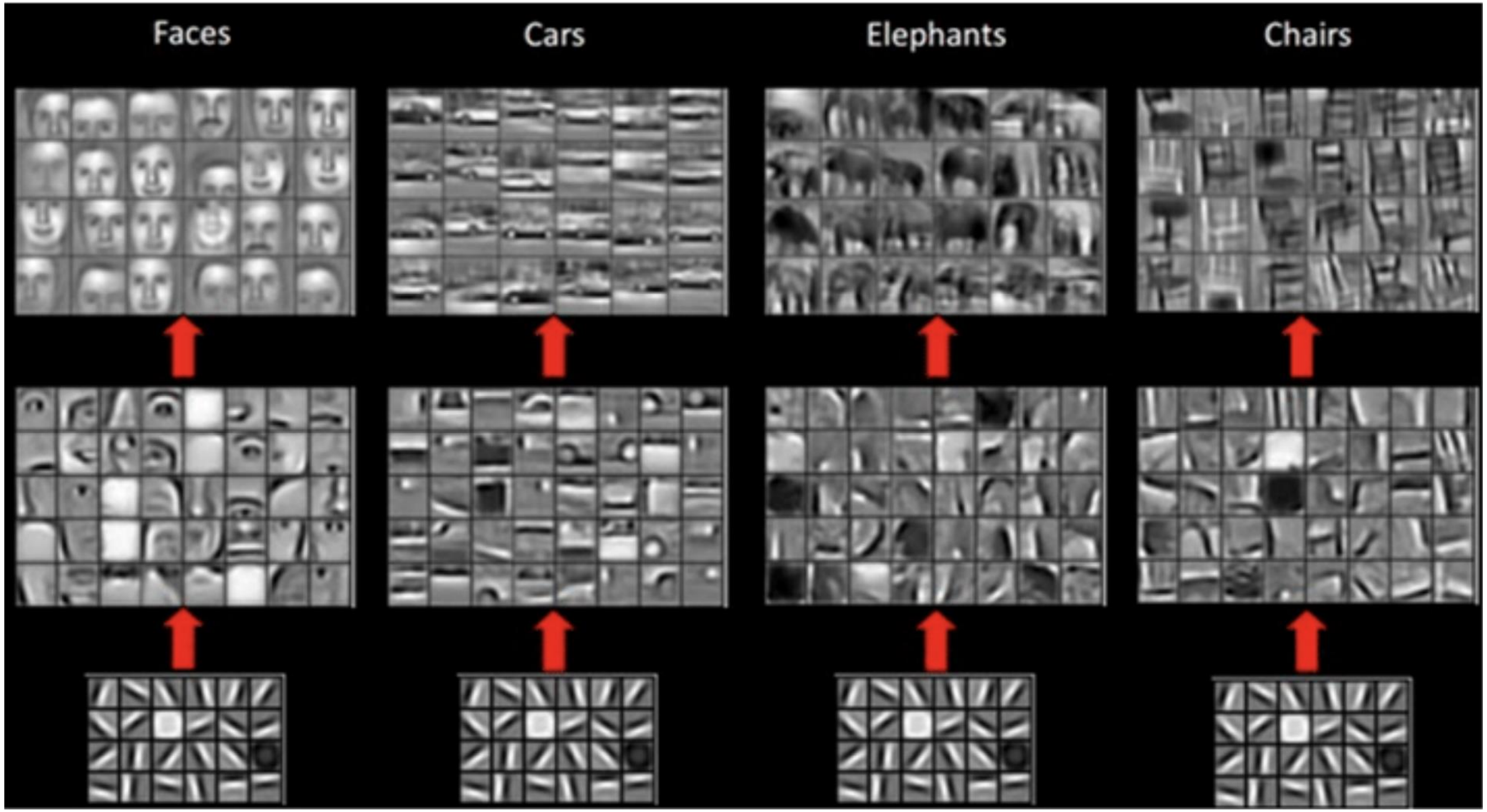
https://github.com/princeton-vl/pytorch_stacked_hourglass

Features on CNN

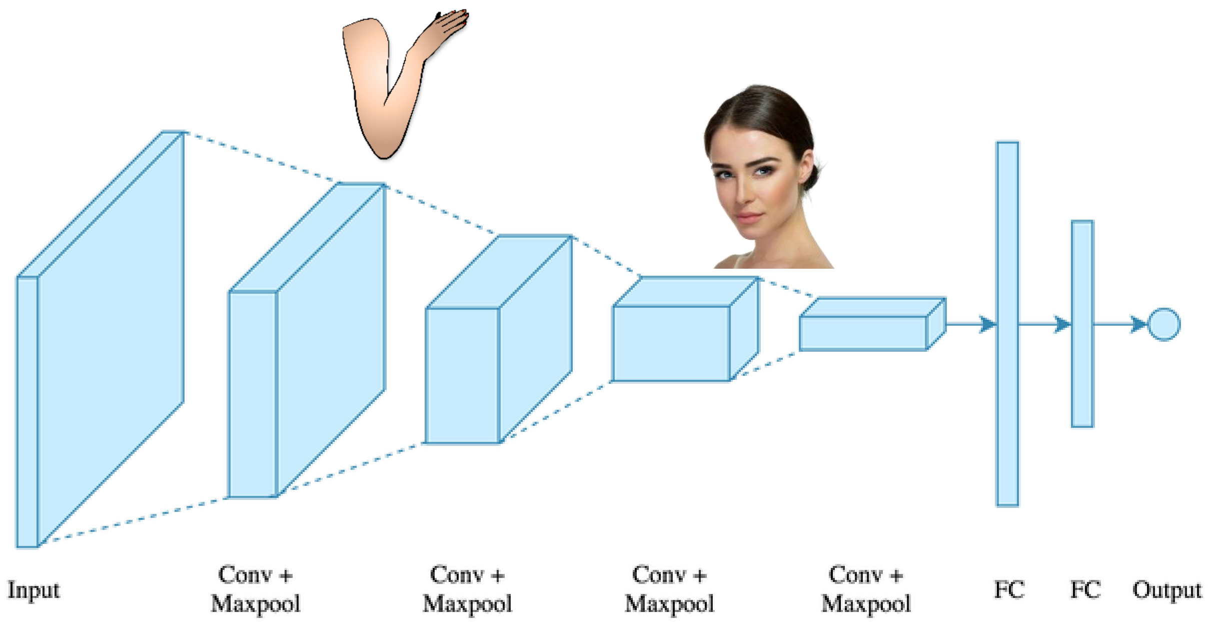
Layer 5 →

Layer 3 →

Layer 1 →



Features on CNN



Stacked Hourglass Networks

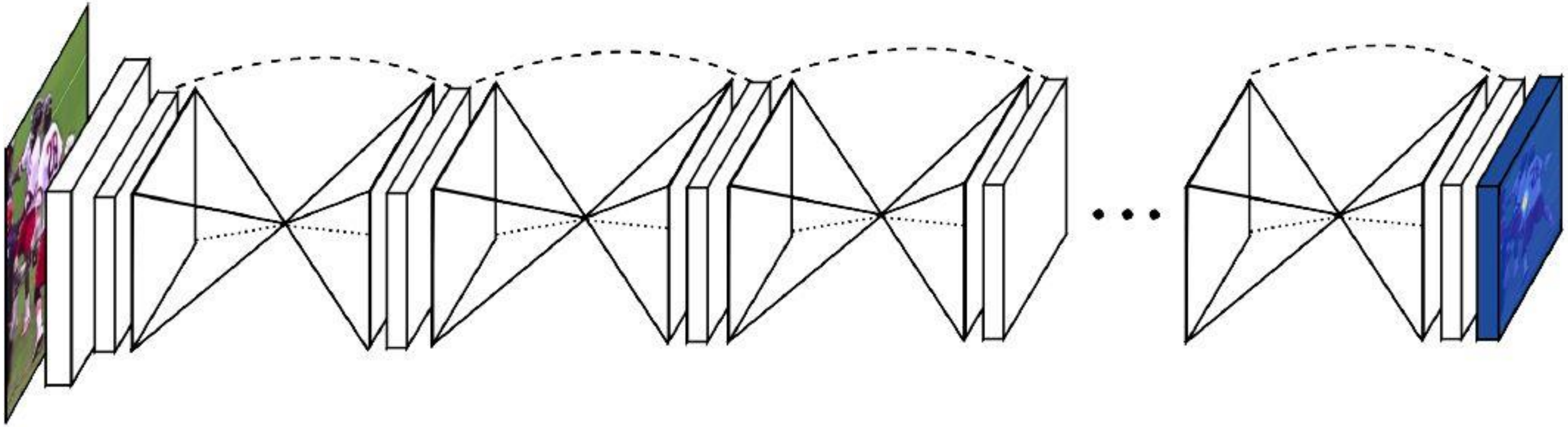
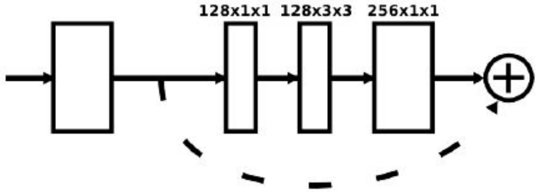
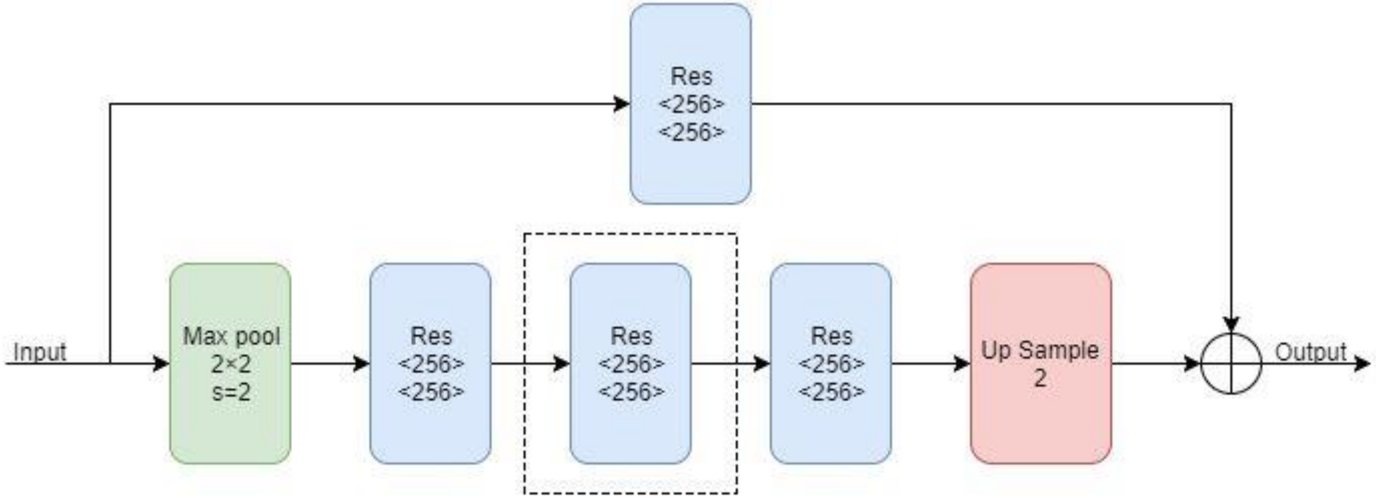


Fig. 1. Our network for pose estimation consists of multiple stacked hourglass modules which allow for repeated bottom-up, top-down inference.

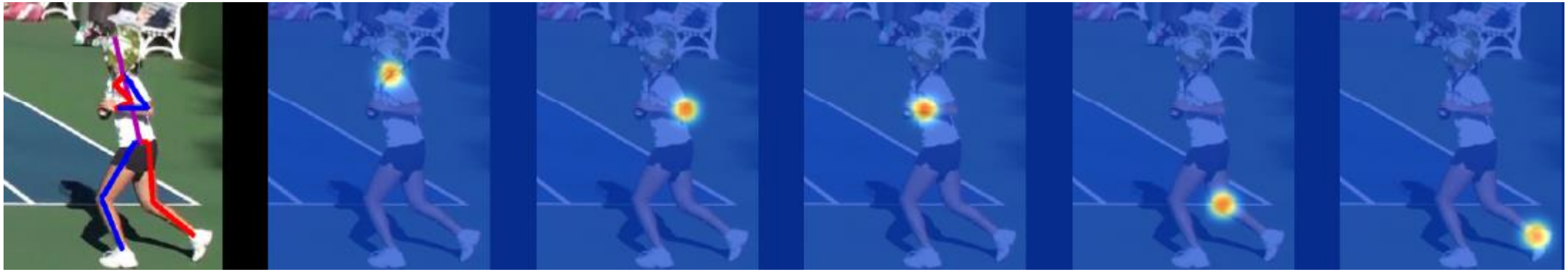
Hourglass Module



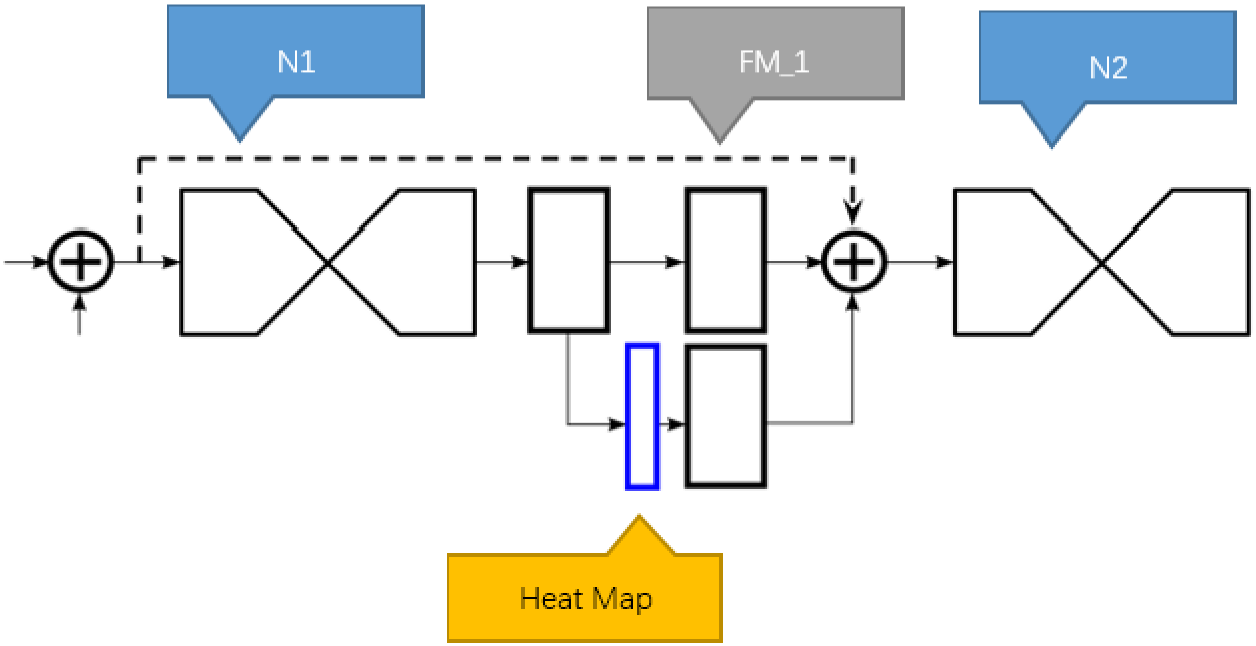


HeatMap

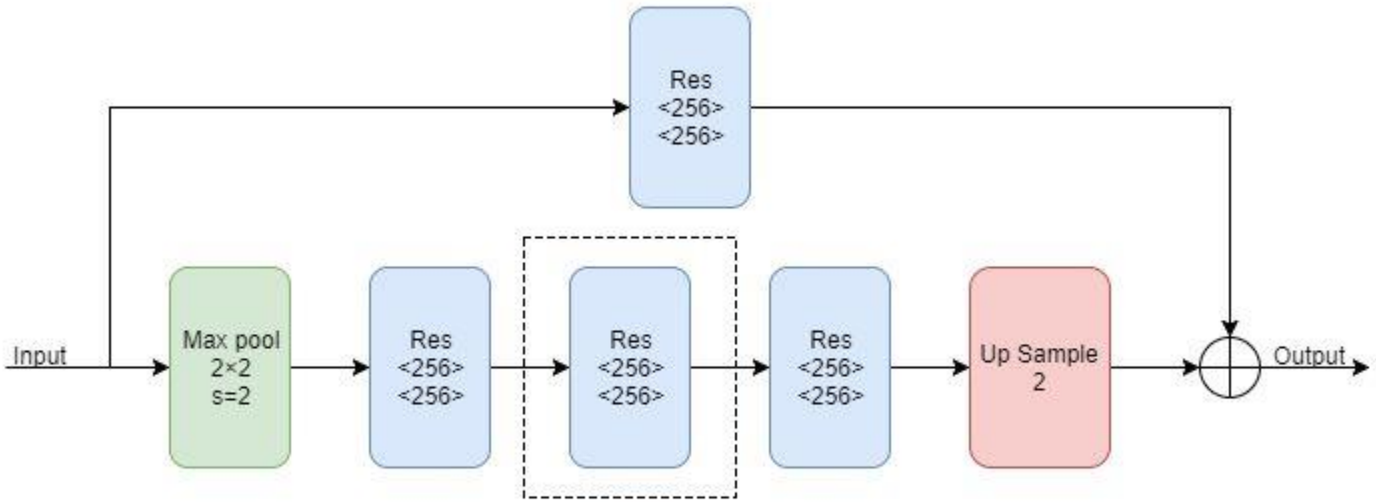
Probability of each key point



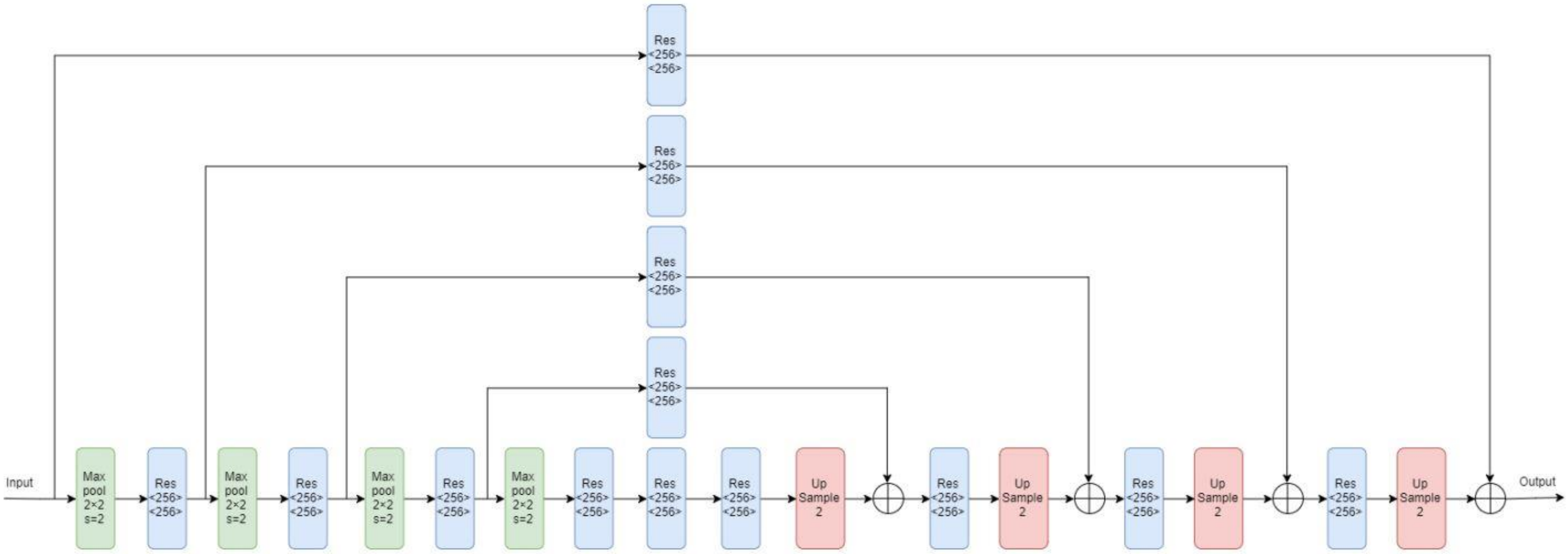
Stacked Hourglass Networks



Hourglass Module



Stacked Hourglass Networks



Stacked Hourglass Networks

Fig. 5. Example output on MPII's test set.

02

CornerNet Series

CornerNet 系列

CornerNet

CornerNet: Detecting Objects as Paired Keypoints

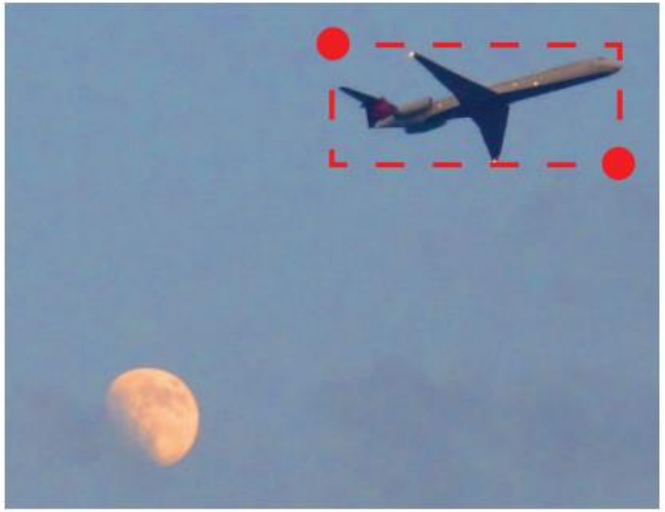
Hei Law[0000−0003−1009−164X], Jia Deng[0000−0001−9594−4554]

University of Michigan, Ann Arbor
{heilaw,jiadeng}@umich.edu

Abstract. We propose CornerNet, a new approach to object detection where we detect an object bounding box as a pair of keypoints, the top-left corner and the bottom-right corner, using a single convolution neural network. By detecting objects as paired keypoints, we eliminate the need for designing a set of anchor boxes commonly used in prior single-stage detectors. In addition to our novel formulation, we introduce corner pooling, a new type of pooling layer that helps the network better localize corners. Experiments show that CornerNet achieves a 42.1% AP on MS COCO, outperforming all existing one-stage detectors.

https://arxiv.org/pdf/1808.01244.pdf

CornerNet



Anchor Free

CornerNet

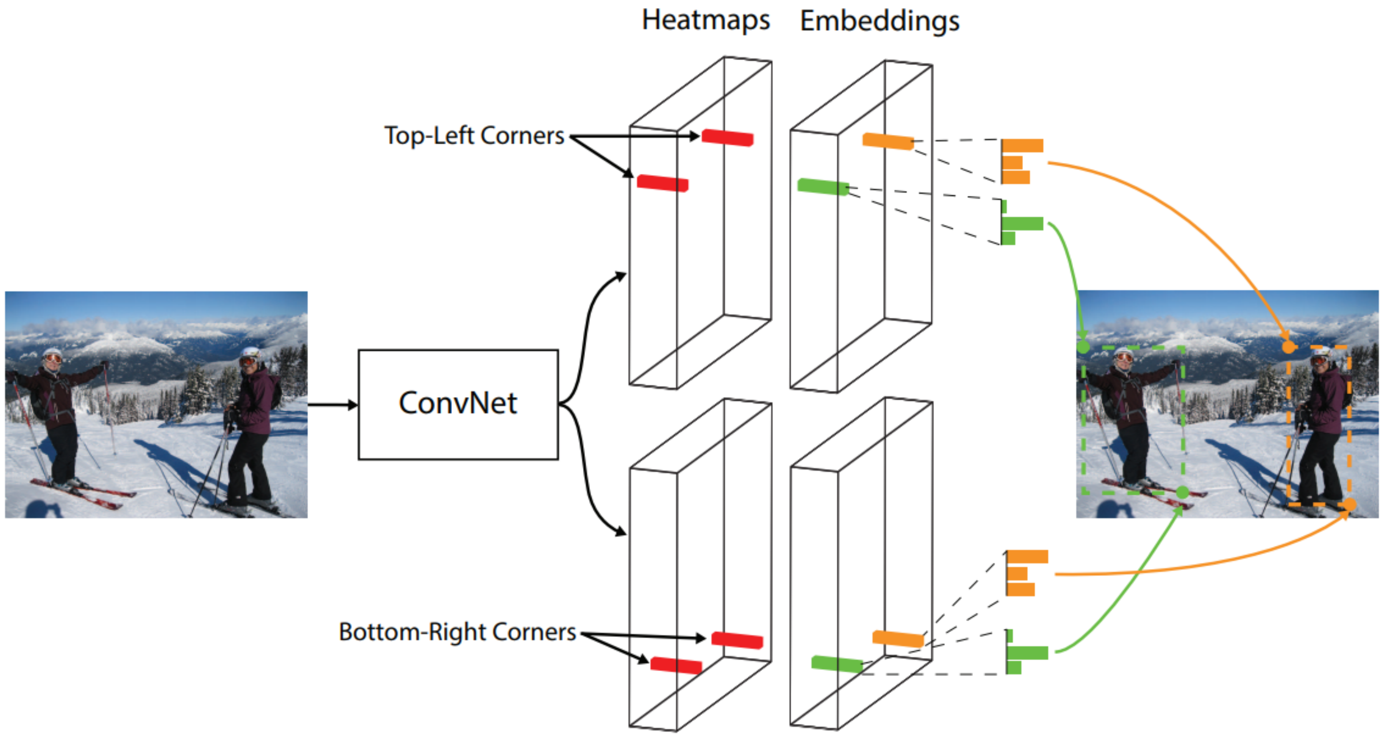


Fig. 1. We detect an object as a pair of bounding box corners grouped together. A convolutional network outputs a heatmap for all top-left corners, a heatmap for all bottom-right corners, and an embedding vector for each detected corner. The network is trained to predict similar embeddings for corners that belong to the same object.

Anchor Free

CornerNet

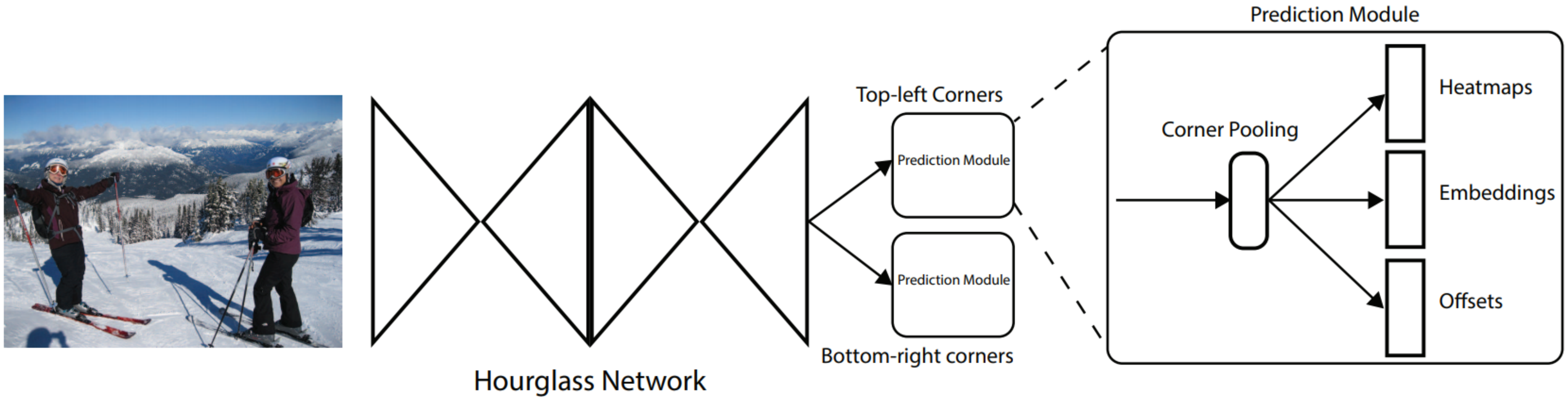
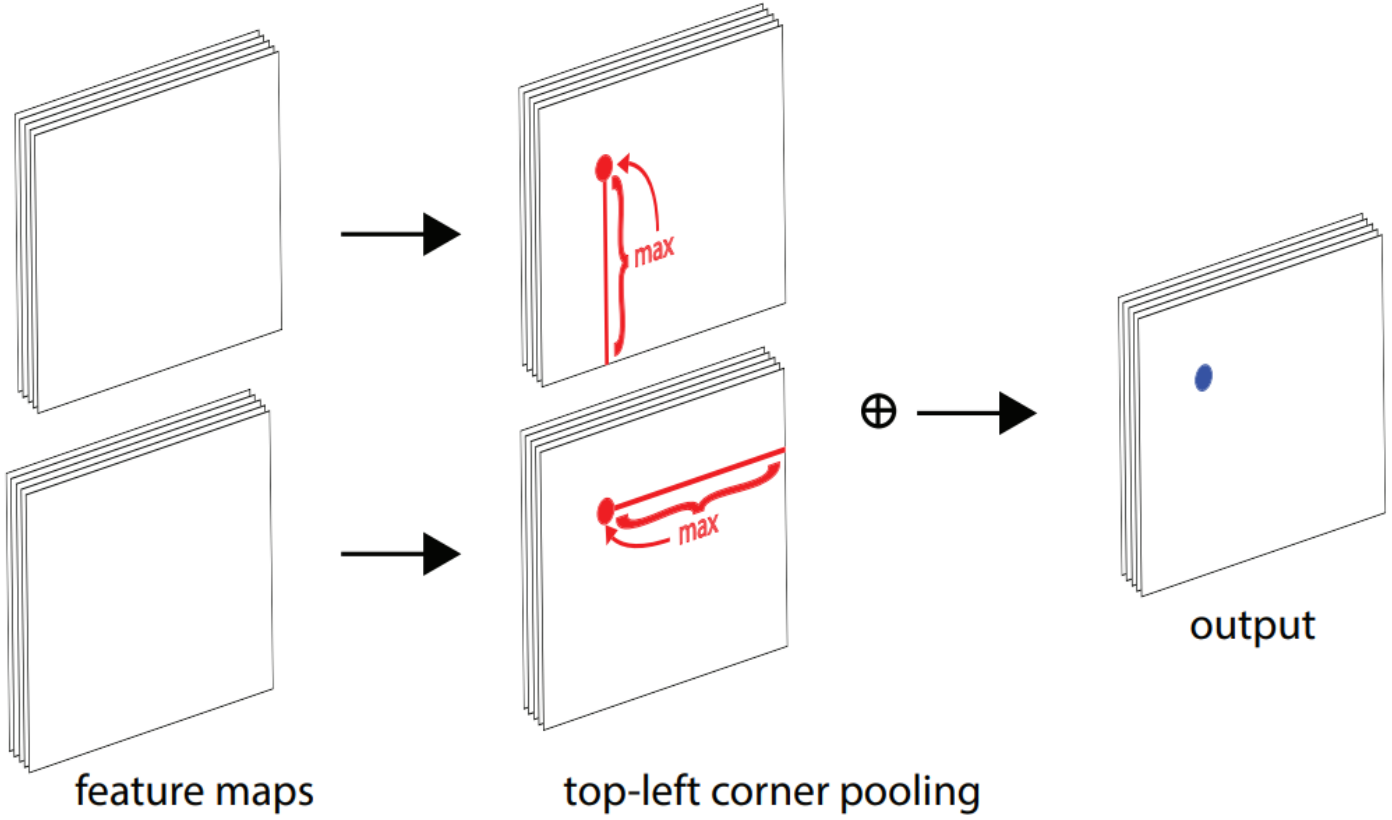


Fig. 4. Overview of CornerNet. The backbone network is followed by two prediction modules, one for the top-left corners and the other for the bottom-right corners. Using the predictions from both modules, we locate and group the corners.

Corner Polling



Corner Polling

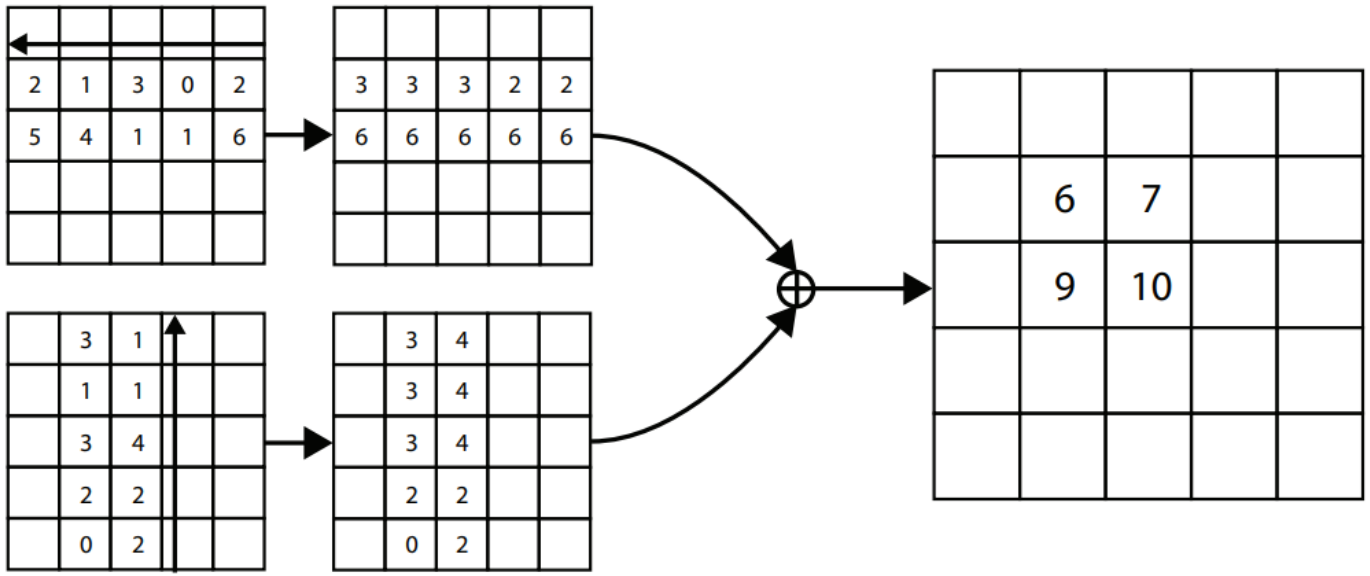


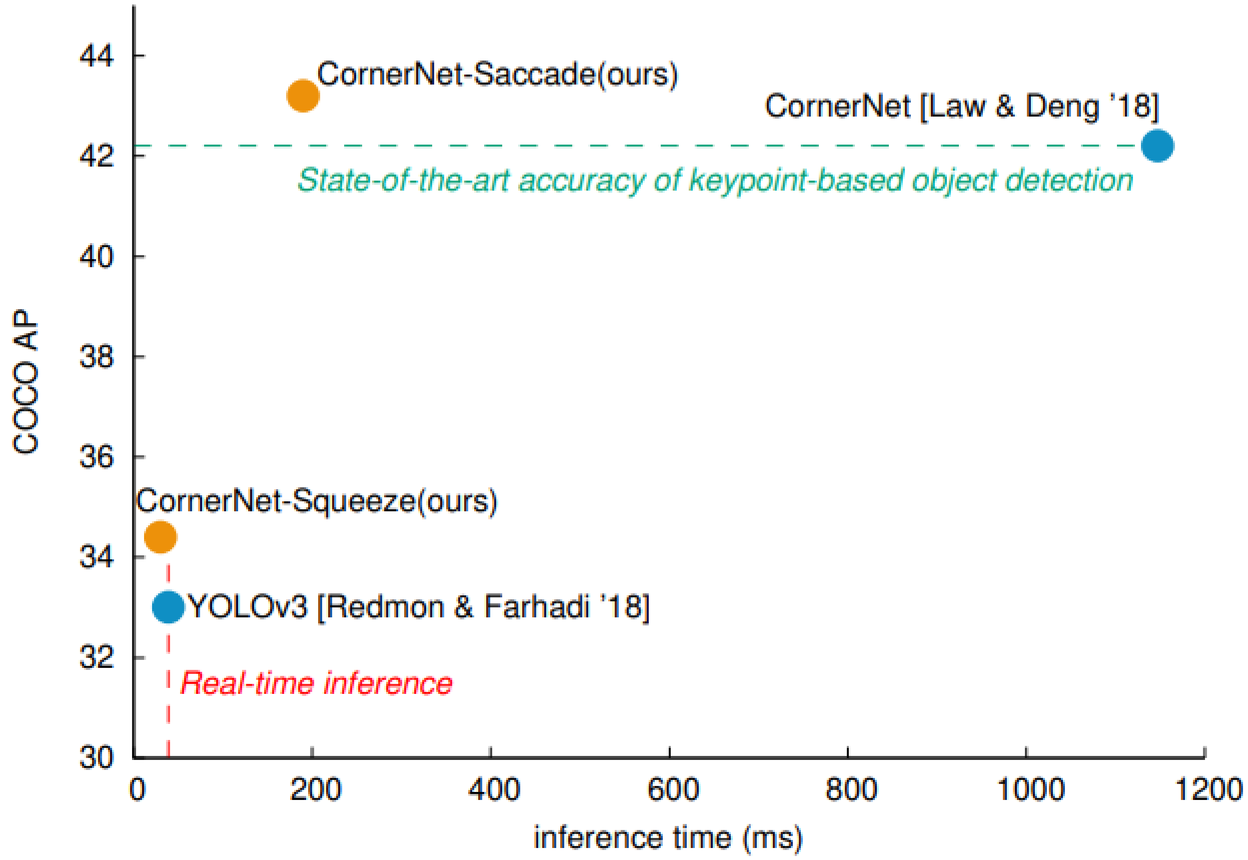
Fig. 6. The top-left corner pooling layer can be implemented very efficiently. We scan from left to right for the horizontal max-pooling and from bottom to top for the vertical max-pooling. We then add two max-pooled feature maps.

CornerNet-Lite

速度慢？

✓ 减少处理的像素数量；

✓ 减少每个像素的处理量。



https://arxiv.org/pdf/1904.08900.pdf

CornerNet-Lite

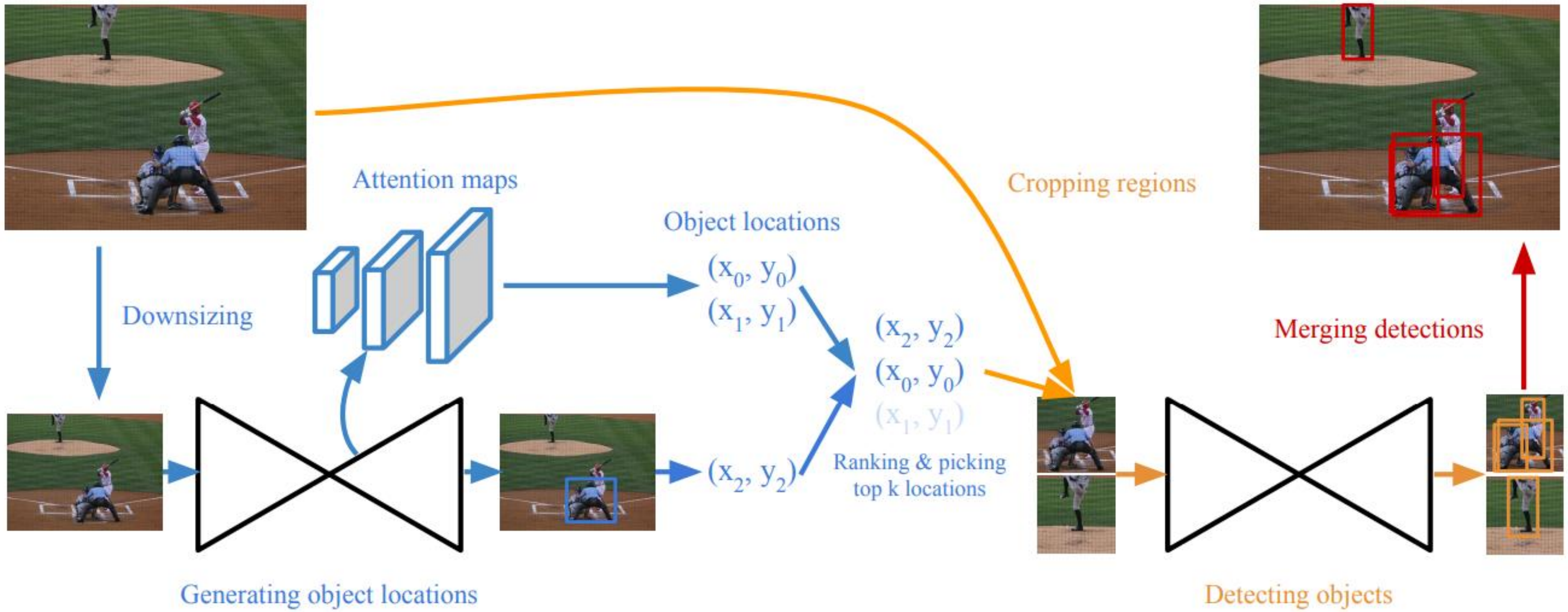


Figure 2: Overview of CornerNet-Saccade. We predict a set of possible object locations from the attention maps and bounding boxes generated on a downsized full image. We zoom into each location and crop a small region around that location. Then we detect objects in each region. We control the efficiency by ranking the object locations and choosing top k locations to process. Finally, we merge the detections by NMS.

CornerNet-Lite

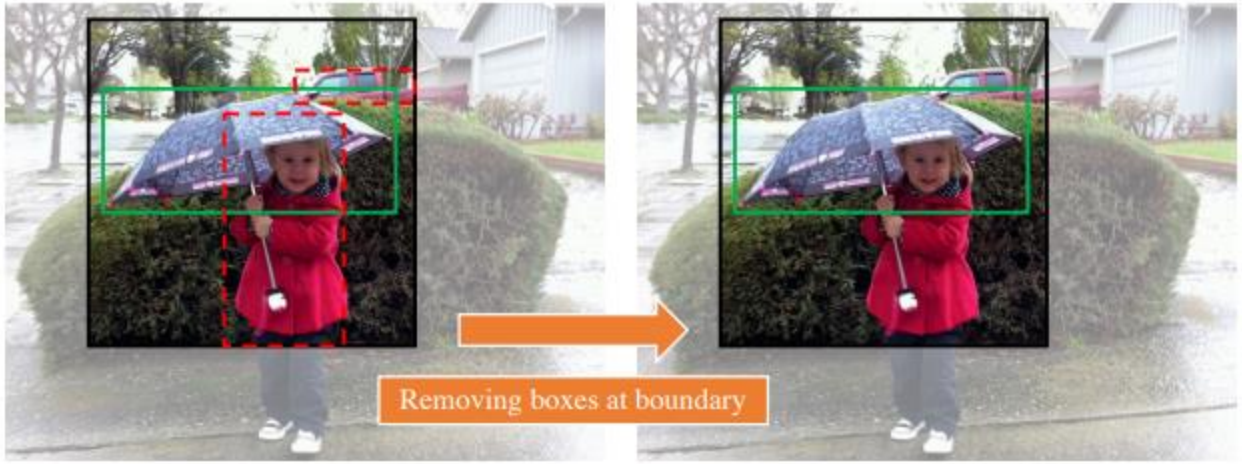


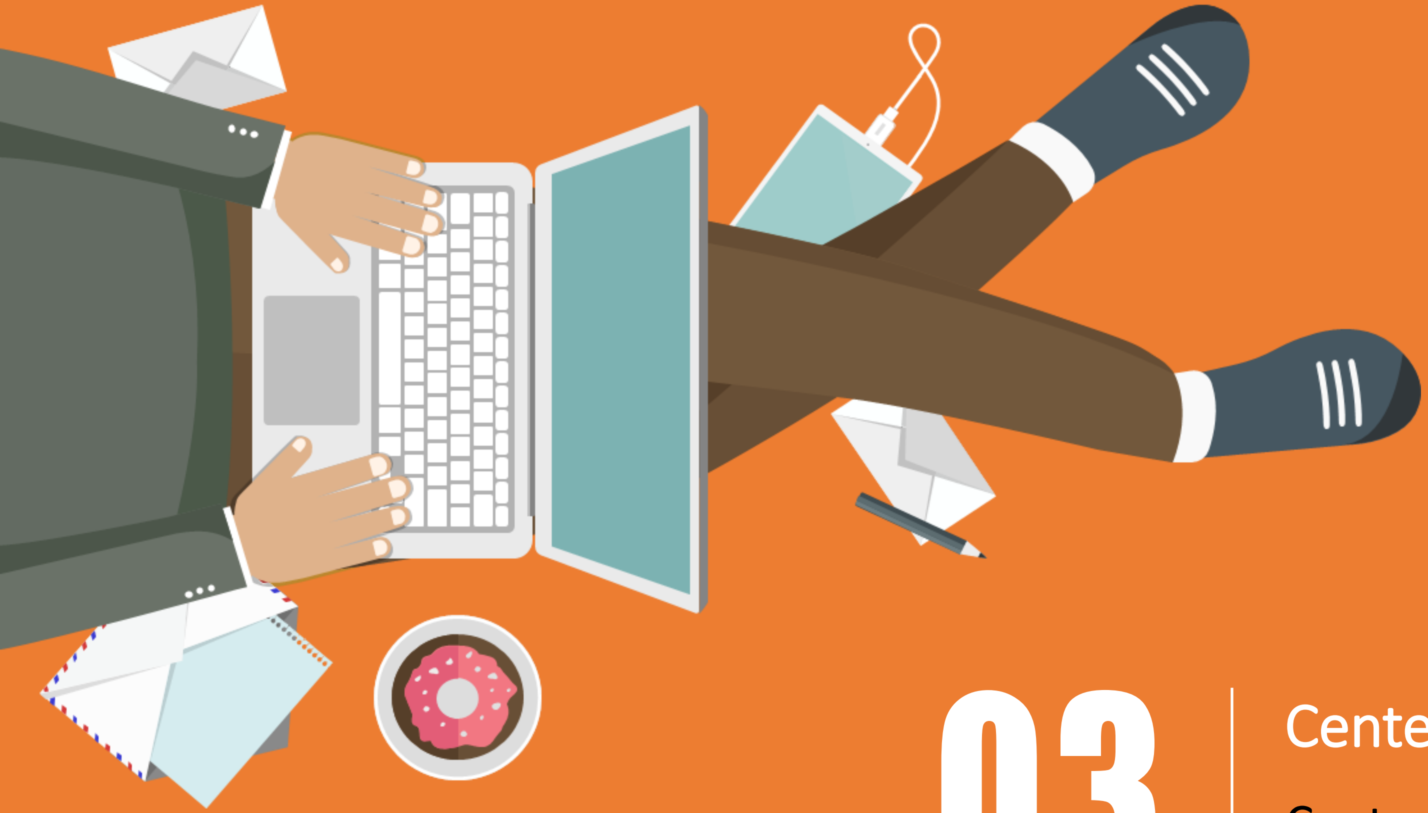
Figure 3: Some objects may not be fully covered by a region. The detector may still generate bounding boxes (red dashed line) for those objects. We remove the bounding boxes which touch the boundaries to avoid such bounding boxes.

CornerNet-Lite

Figure 4: When the objects are close to each other, we may generate regions that highly overlap with each other. Processing either one of them is likely to detect objects in all highly overlapping regions. We suppress redundant regions to improve efficiency.

CornerNet-Lite

Figure 5: Qualitative examples on COCO validation set.



03

CenterNet

CenterNet

CenterNet

CenterNet: Keypoint Triplets for Object Detection

Kaiwen Duan[1*] Song Bai[2] Lingxi Xie[3] Honggang Qi[1] Qingming Huang[1] Qi Tian[3]

[1]University of Chinese Academy of Sciences [2]University of Oxford [3]Huawei Noah's Ark Lab

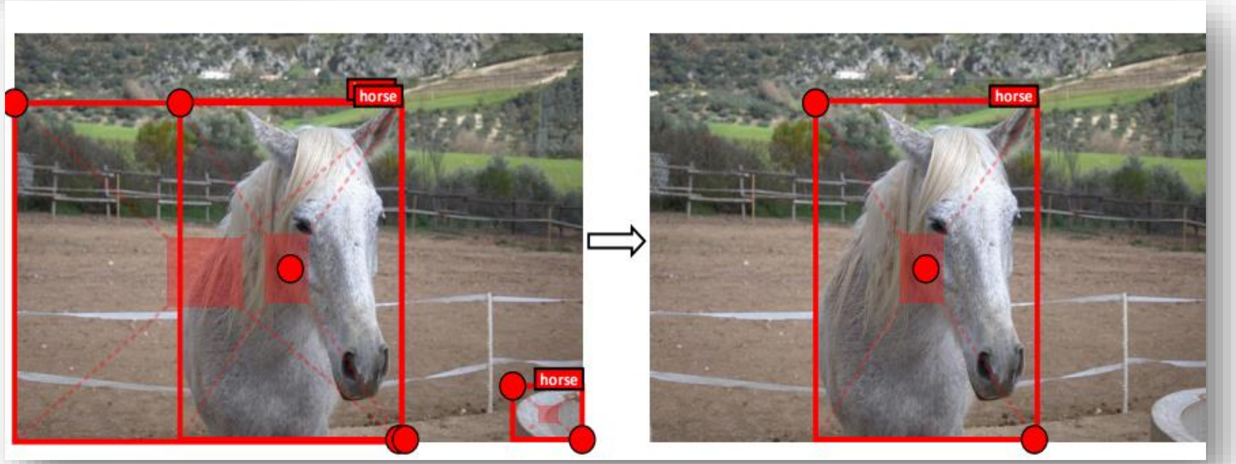
kaiwen.duan@vipl.ict.ac.cn songbai.site@gmail.com 198808xc@gmail.com hgqi@ucas.ac.cn

qmhuang@ucas.ac.cn tian.qi1@huawei.com

https://arxiv.org/pdf/1904.08189.pdf

CenterNet

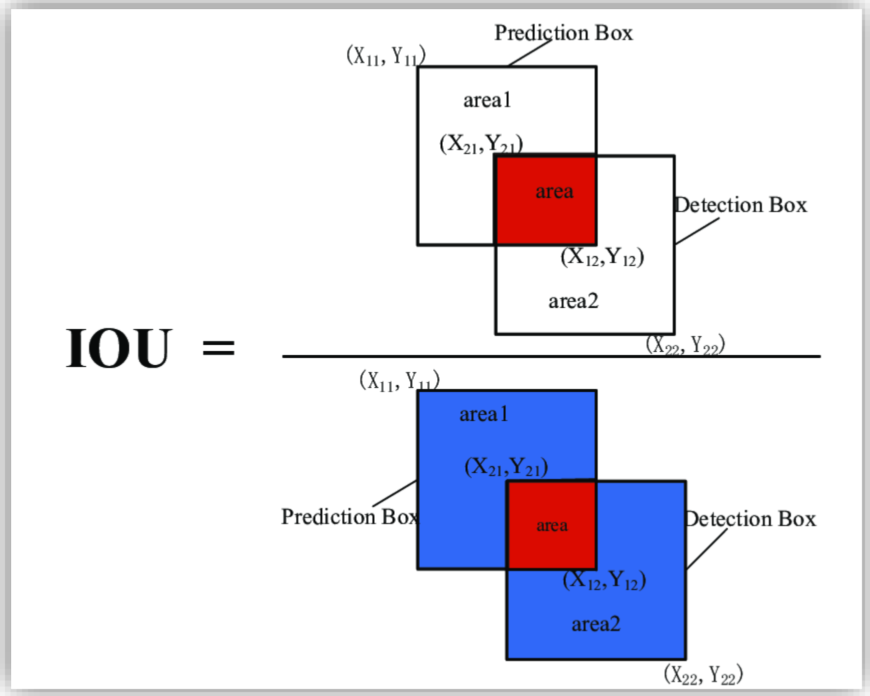

potted plant
person
vase
carrot
broccoli


horse
horse
horse

CenterNet

Method	FD	FD_5	FD_{25}	FD_{50}	FD_S	FD_M	FD_L
CornerNet	37.8	32.7	36.8	43.8	60.3	33.2	25.1

Table 1: False discovery rates (%) of CornerNet. The false discovery rate reflects the distribution of incorrect bounding boxes. The results suggest the incorrect bounding boxes account for a large proportion.



https://arxiv.org/pdf/1904.08189.pdf

CenterNet

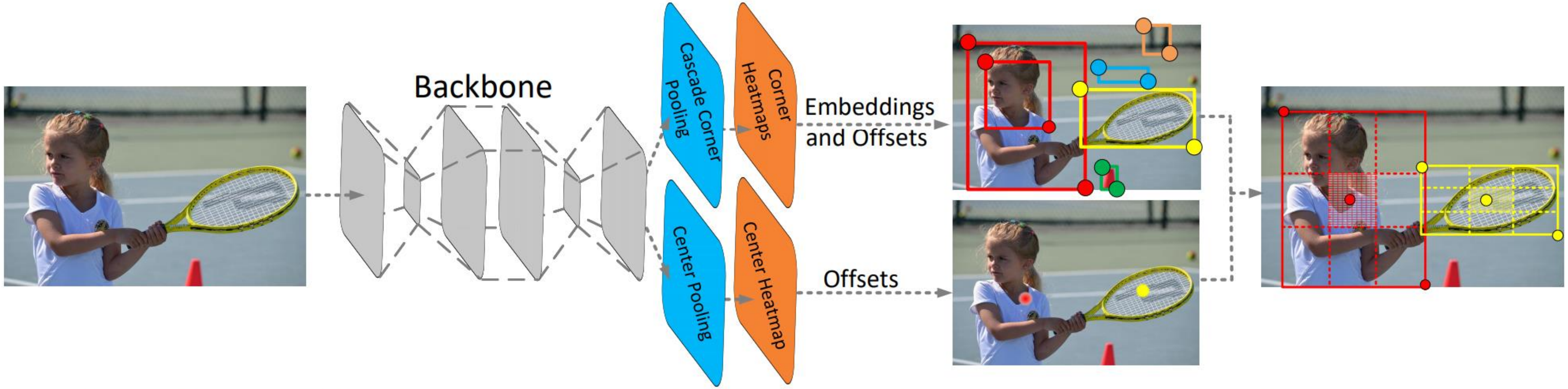
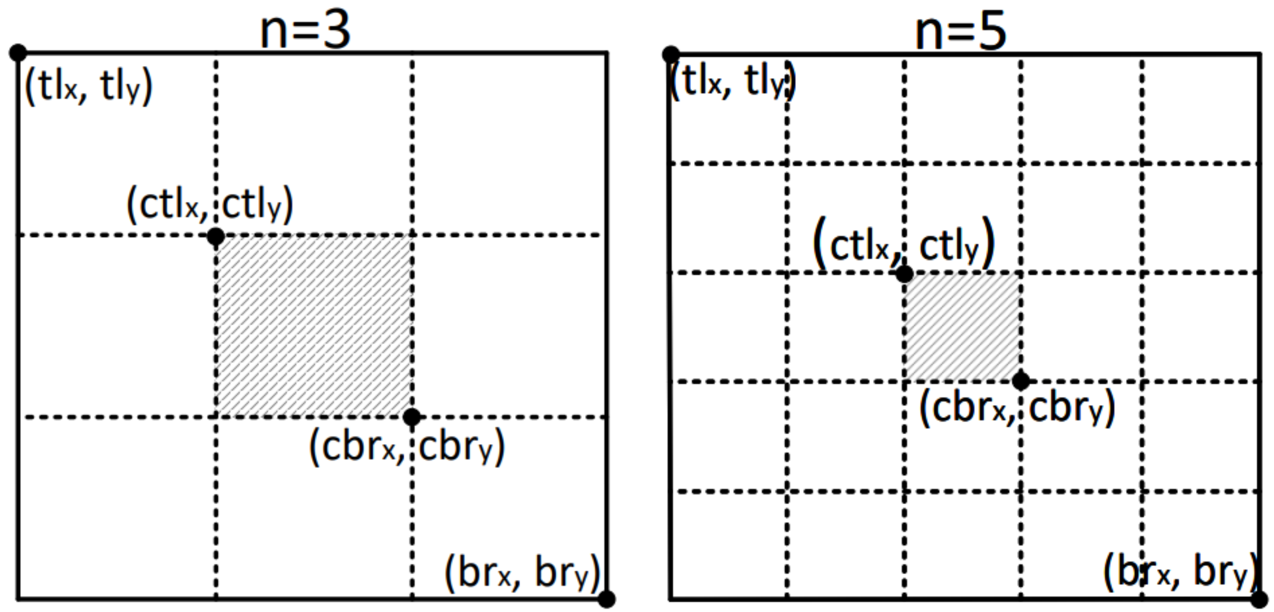


Figure 2: Architecture of CenterNet. A convolutional backbone network applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Similar to CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes.

https://arxiv.org/pdf/1904.08189.pdf

CenterNet



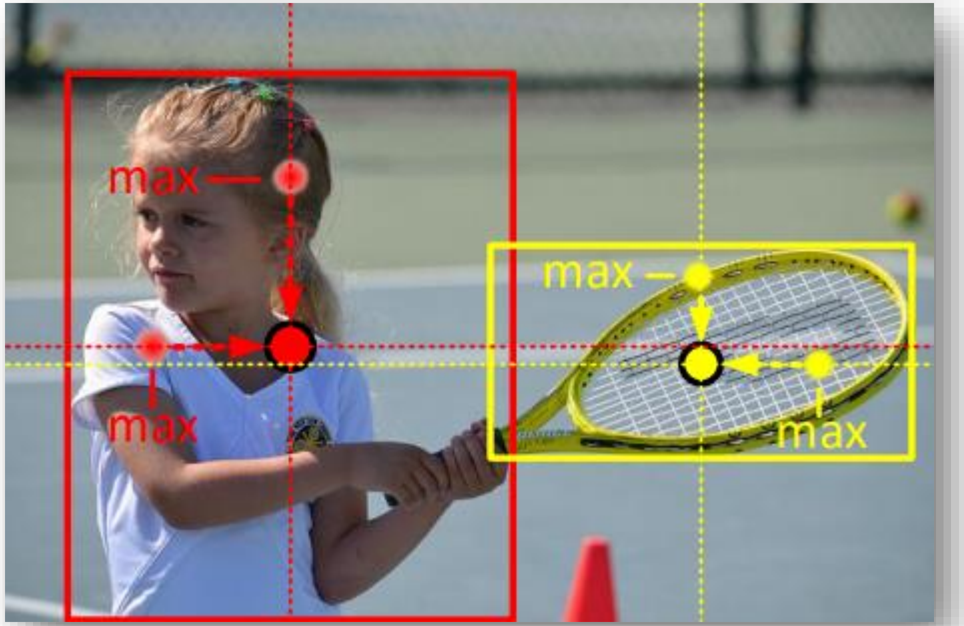
(a) (b)

Figure 3: (a) The central region when $n = 3$. (b) The central region when $n = 5$. The solid rectangles denote the predicted bounding boxes and the shaded regions denote the central regions.

$$
\begin{cases}
\mathrm{ctl_x} = \dfrac{(n+1)\mathrm{tl_x} + (n-1)\mathrm{br_x}}{2n} \\
\mathrm{ctl_y} = \dfrac{(n+1)\mathrm{tl_y} + (n-1)\mathrm{br_y}}{2n} \\
\mathrm{cbr_x} = \dfrac{(n-1)\mathrm{tl_x} + (n+1)\mathrm{br_x}}{2n} \\
\mathrm{cbr_y} = \dfrac{(n-1)\mathrm{tl_y} + (n+1)\mathrm{br_y}}{2n}
\end{cases}
$$

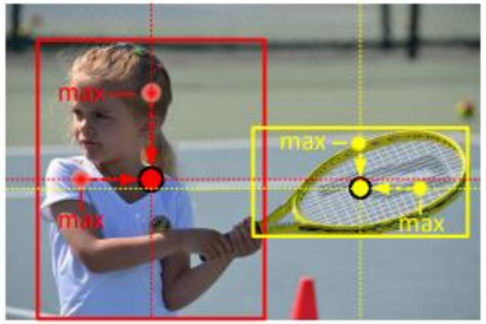
https://arxiv.org/pdf/1904.08189.pdf

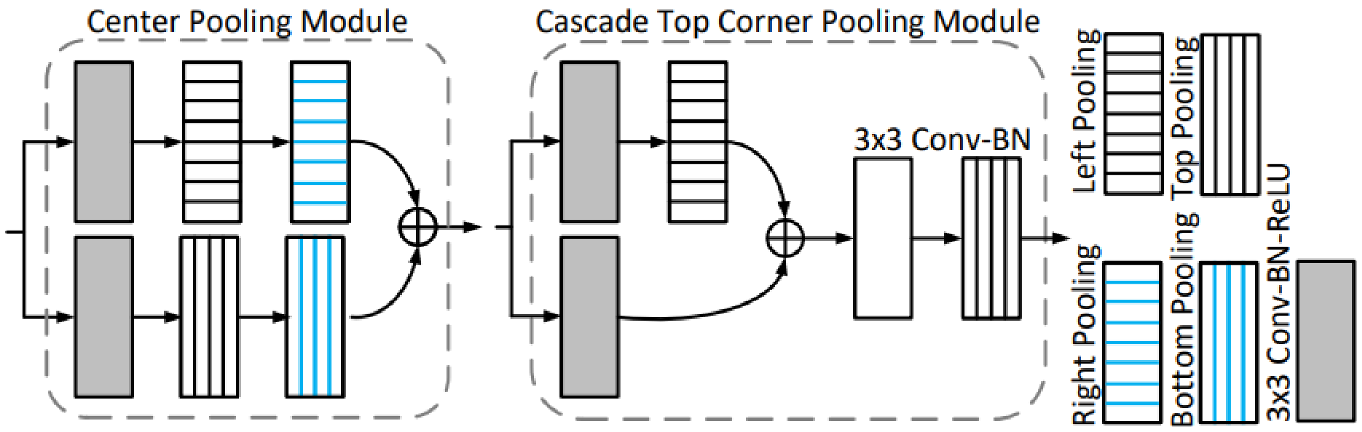
Center Pooling



https://arxiv.org/pdf/1904.08189.pdf

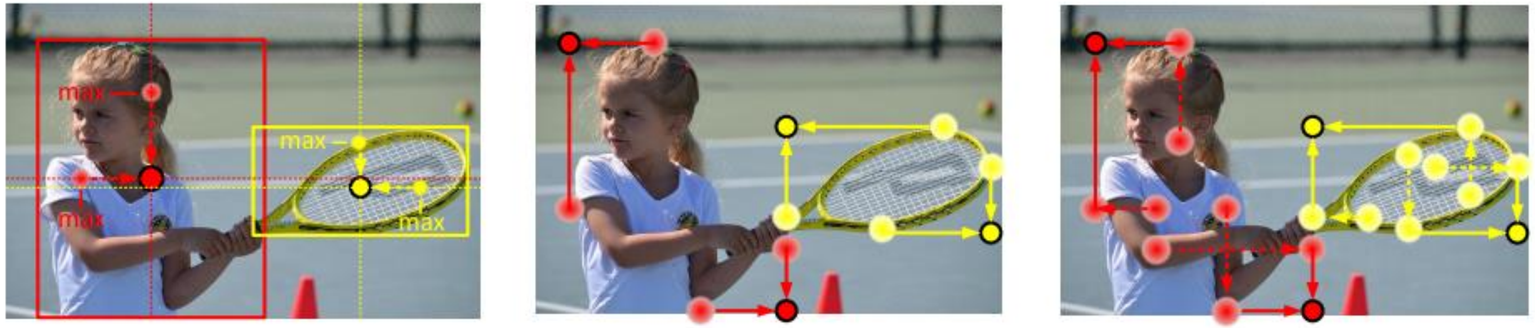
CenterNet





https://arxiv.org/pdf/1904.08189.pdf

CenterNet



(a) (b) (c)

Figure 4: (a) Center pooling takes the maximum values in both horizontal and vertical directions. (b) Corner pooling only takes the maximum values in boundary directions. (c) Cascade corner pooling takes the maximum values in both boundary directions and internal directions of objects.

https://arxiv.org/pdf/1904.08189.pdf

Q&A