Bottom-up Object Detection by Grouping Extreme and Center Points - Supplementary Material

Xingyi Zhou
UT Austin
zhouxy@cs.utexas.edu

Jiacheng Zhuo
UT Austin
jzhuo@cs.utexas.edu

Philipp Krähenbühl
UT Austin
philkr@cs.utexas.edu

Algorithm 1: Center Grouping

```
Input  : Center and Extremepoint heatmaps of an image for one
         category: Ŷ^(c), Ŷ^(t), Ŷ^(l), Ŷ^(b), Ŷ^(r) ∈ (0,1)^(H×W)
         Center and peak selection thresholds: τ_c and τ_p
Output: Bounding box with score
// Convert heatmaps into coordinates of keypoints.
// T, L, B, R are sets of points.
T ← ExtractPeak(Ŷ^(t), τ_p)
L ← ExtractPeak(Ŷ^(l), τ_p)
B ← ExtractPeak(Ŷ^(b), τ_p)
R ← ExtractPeak(Ŷ^(r), τ_p)
B_x ← A list of emtpy lists of size 2W
B_y ← A list of emtpy lists of size 2H
for t ∈ T, b ∈ B do
    if t_y ≤ b_y then
        Add (t, b) to B_y[t + b]
    end
end
for l ∈ L, r ∈ R do
    if l_x ≤ r_x then
        Add (l, r) to B_x[l + r]
    end
end
for i ∈ [0, 2W − 1], j ∈ [0, 2H − 1] do
    c_x = i/2, c_y = j/2
    if Ŷ^(c)_{c_x,c_y} ≥ τ_c then
        for (l, r) ∈ B_x[i], (t, b) ∈ B_y[j] do
            if t_y ≤ l_y, r_y ≤ b_y and l_x ≤ t_x, b_x ≤ r_x then
                Add Bounding box (l_x, t_y, r_x, b_y) with score
                (Ŷ^(t)_{t_x,t_y} + Ŷ^(l)_{l_x,l_y} + Ŷ^(b)_{b_x,b_y} + Ŷ^(r)_{r_x,r_y} +
                Ŷ^(c)_{c_x,c_y})/5.
            end
        end
    end
end
```

1. $O(n^2)$ grouping algorithm

Algorithm. 1 gives an $O(n^2)$ center grouping algorithm. The basic idea is to separate the enumeration between x dimension and y dimension and early reject invalid bounding boxes. However, the algorithm is hard to be accelerated by GPU. On the other hand, n is less than 40 in our experiments and the $O(n^4)$ grouping algorithm runs effectively on GPU. We find the $O(n^2)$ algorithm runs slower in practice.

	AP	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
CornerNet (SS)	40.5	35.3	54.3	59.1	37.4	61.9	76.9
CornerNet (MS)	42.1	36.4	55.7	60.0	38.5	62.7	77.4
ExtremeNet (SS)	39.8	31.6	49.9	53.0	30.0	56.5	69.5
ExtremeNet (MS)	43.2	34.7	55.6	60.0	37.0	63.3	78.1

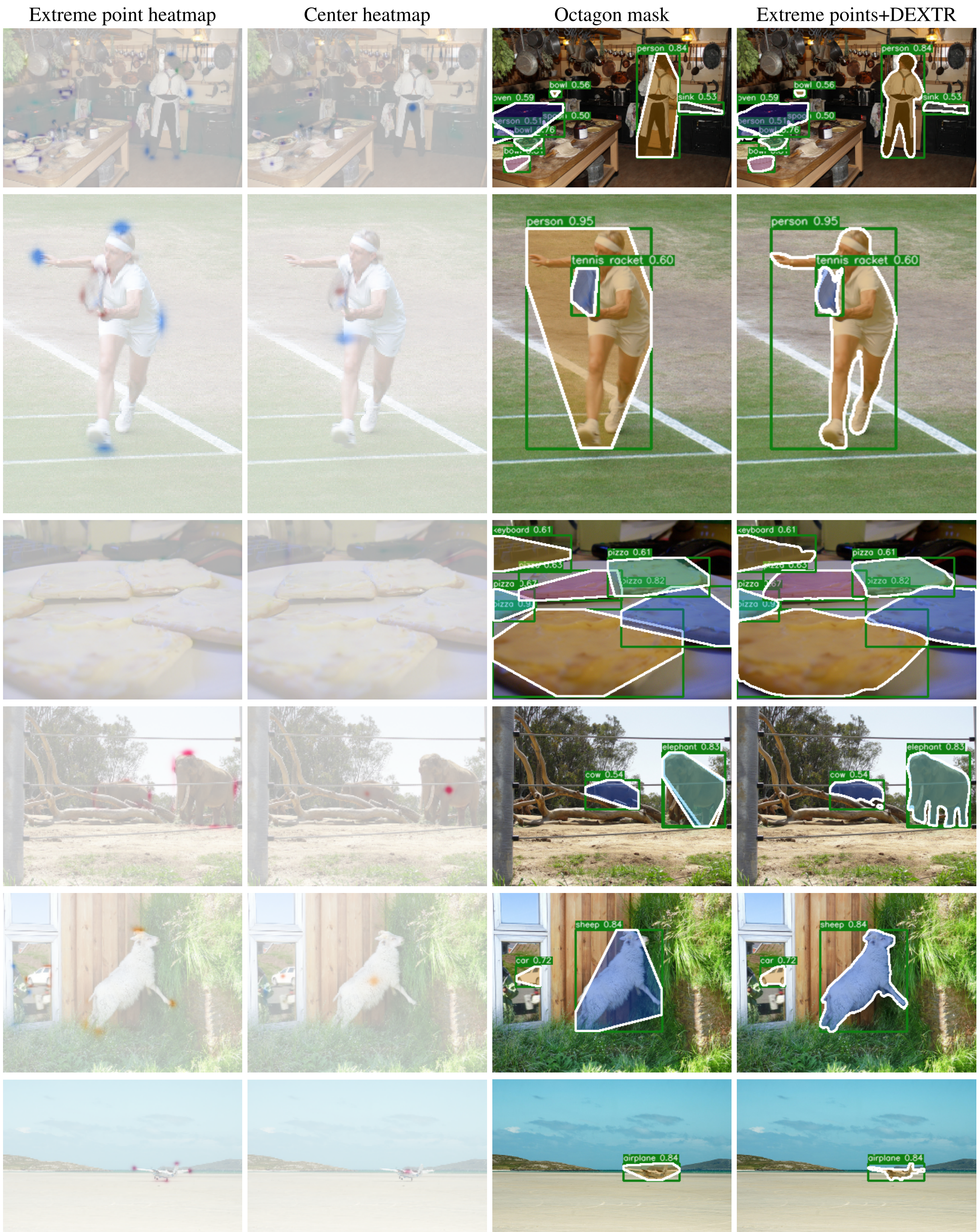
Table 1: Average recall evaluation on COCO test-dev comparing to CornerNet [1] . SS/ MS are short for single-scale/ multi-scale tesing, respectively. AR_k is the bounding box recall of the top k predictions. AR_S, AR_M, AR_L are the recall of the top 100 predictions for small, median, large objects, respectively.

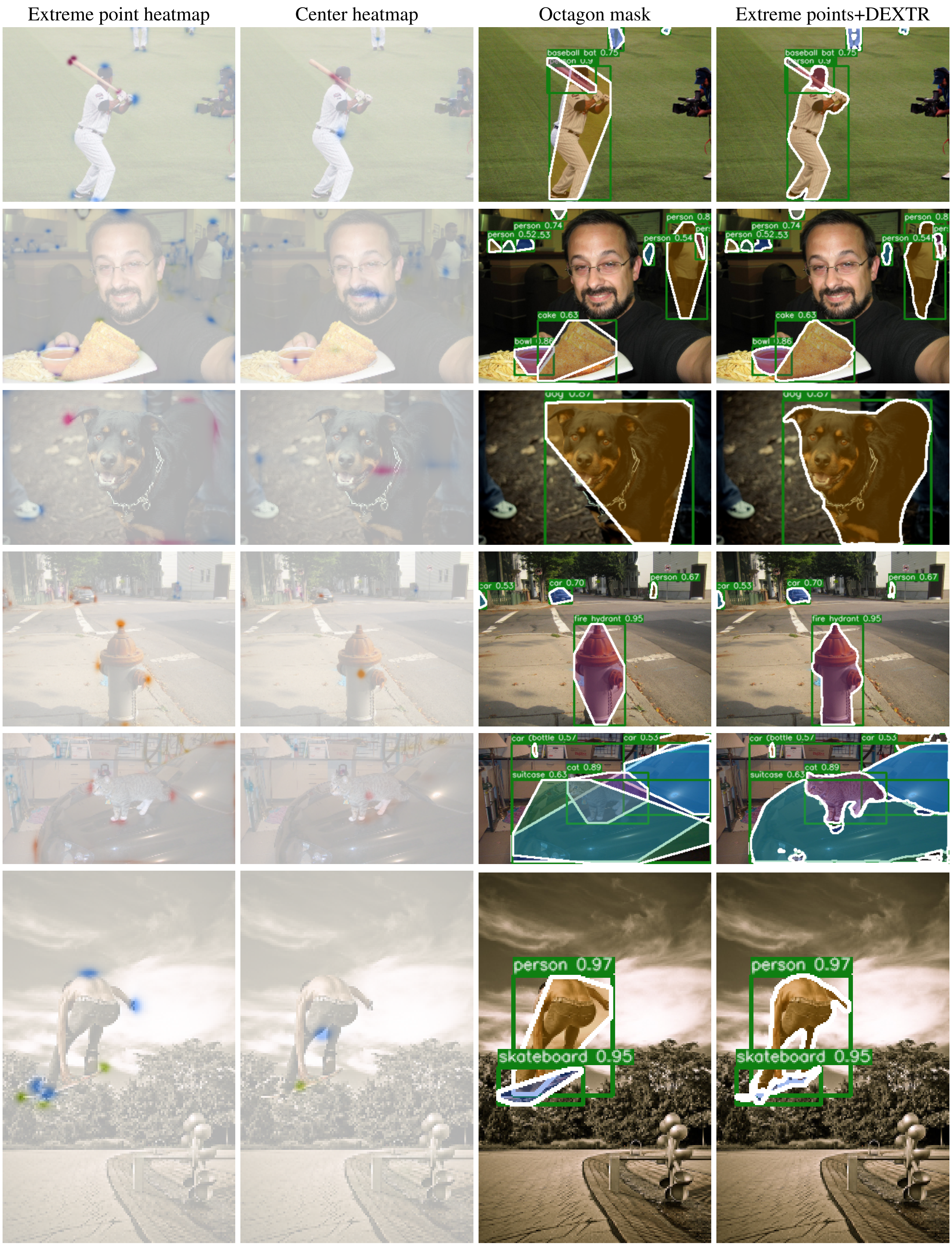
2. Average recall evaluation

Table. 1 shows the average recall on COCO test-dev set. We observe that our single-scale model yields considerably lower recall than single scale CornerNet, although their overall AP are close. While in the multi-scale setting, the AR of ExtremeNet and CornerNet are comparable. This shows multi-scale testing benefits ExtremeNet by increasing its recall. It can be understood that our center grouping requires strict pixel-level accuracy for the matching between the computed geometric center and the center heatmap. Multi-scale testing implicitly gives an error tolerance and gives more detections at a good precision.

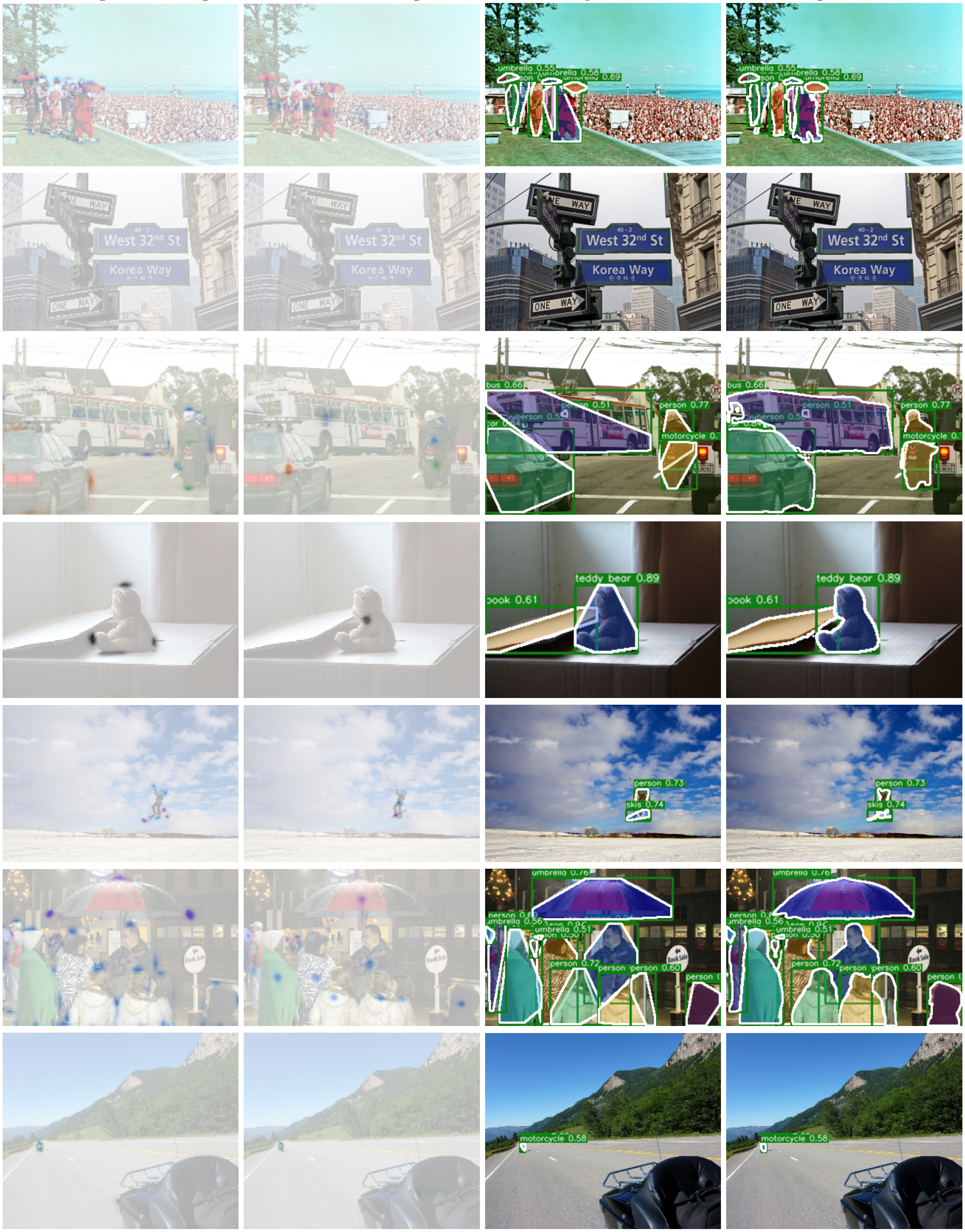
3. More qualitative results

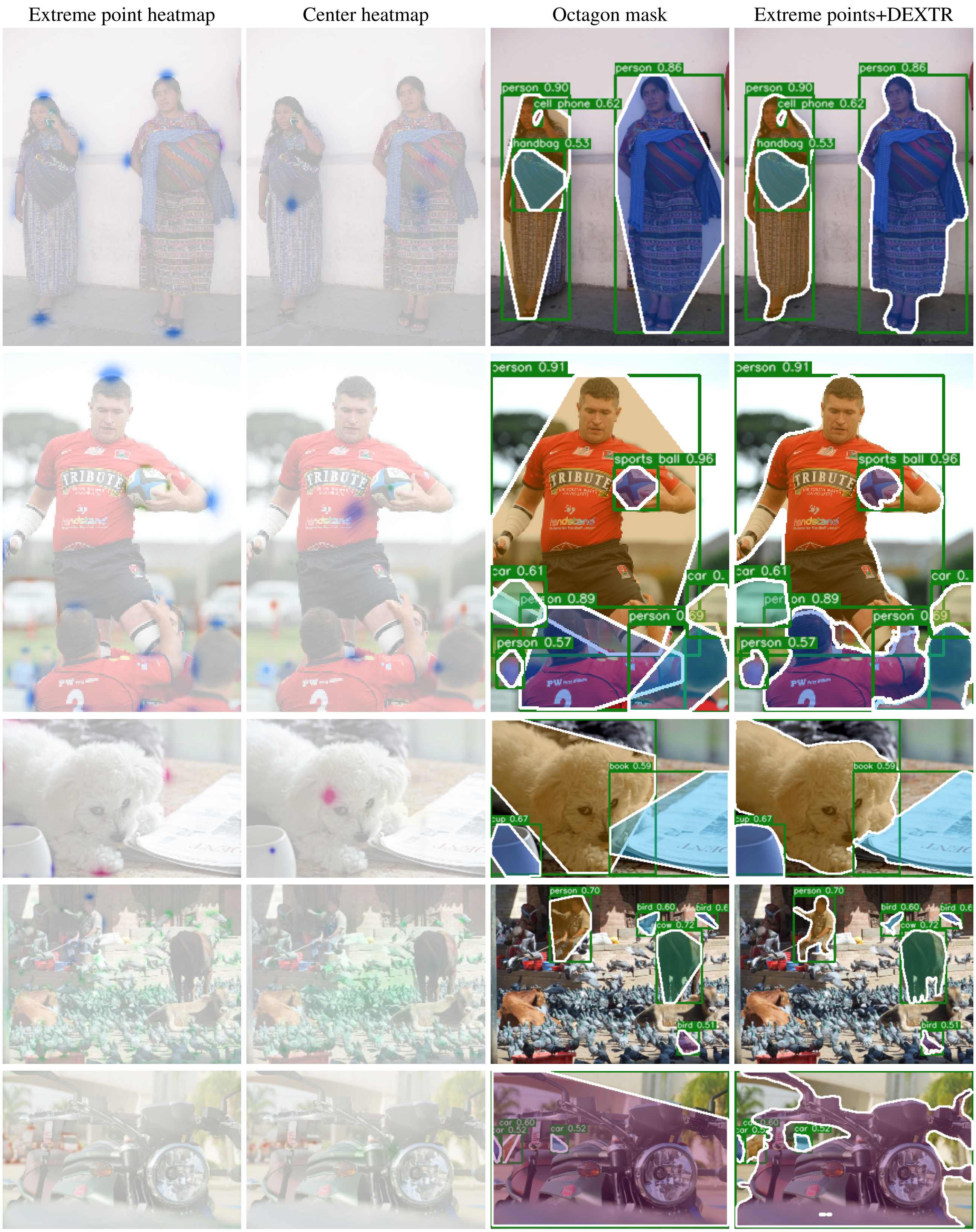
More qualitative results. We *uniformly* sample 50 images from the 5000 images of COCO val2017. We show detection results with confidence ≥ 0.5. First and second column: our predicted extreme point heatmap and center heatmap. Third column: our predicted bounding box and the octagon mask formed by extreme points. Fourth column: resulting masks of feeding our extreme point predictions to DeepExtremeCut [2].

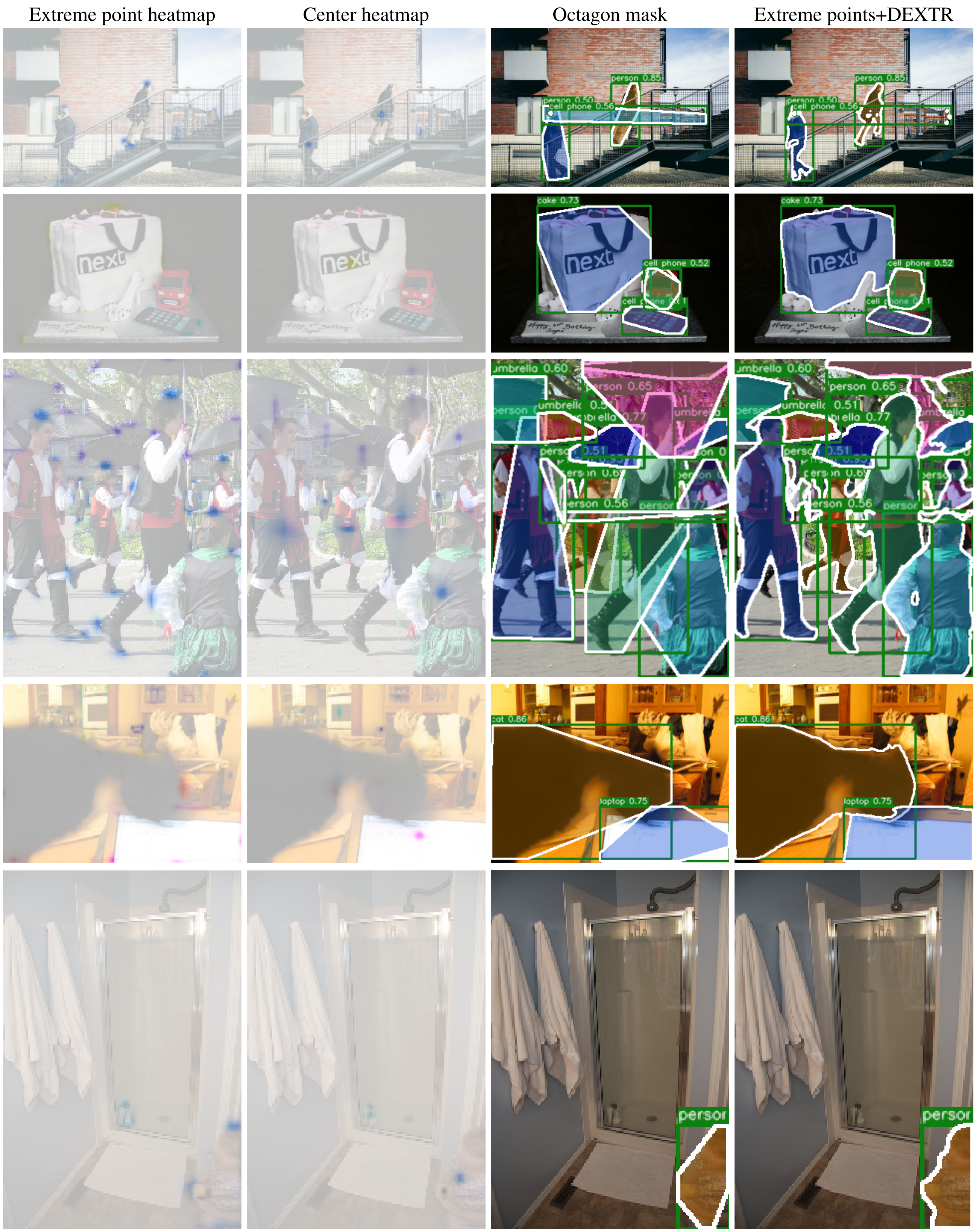

Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR
person 0.84
bowl 0.56
sink 0.53
bowl 0.50
person 0.95
tennis racket 0.60
keyboard 0.61
pizza 0.61
pizza 0.82
elephant 0.83
cow 0.54
sheep 0.84
car 0.72
airplane 0.84


Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR
baseball bat 0.75
cake 0.63
fire hydrant 0.95
cat 0.89
person 0.97
skateboard 0.95

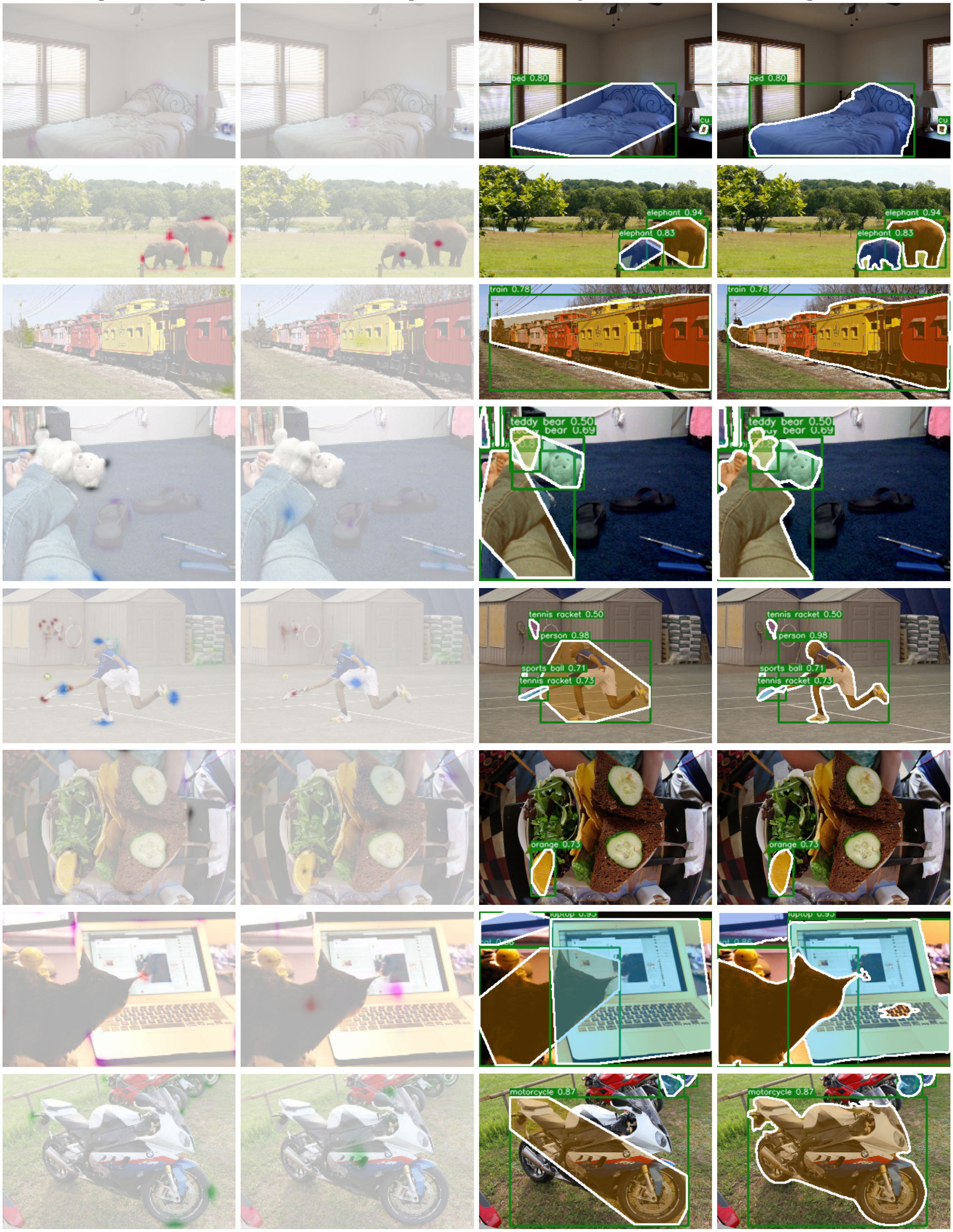
Extreme point heatmap | Center heatmap | Octagon mask | Extreme points+DEXTR

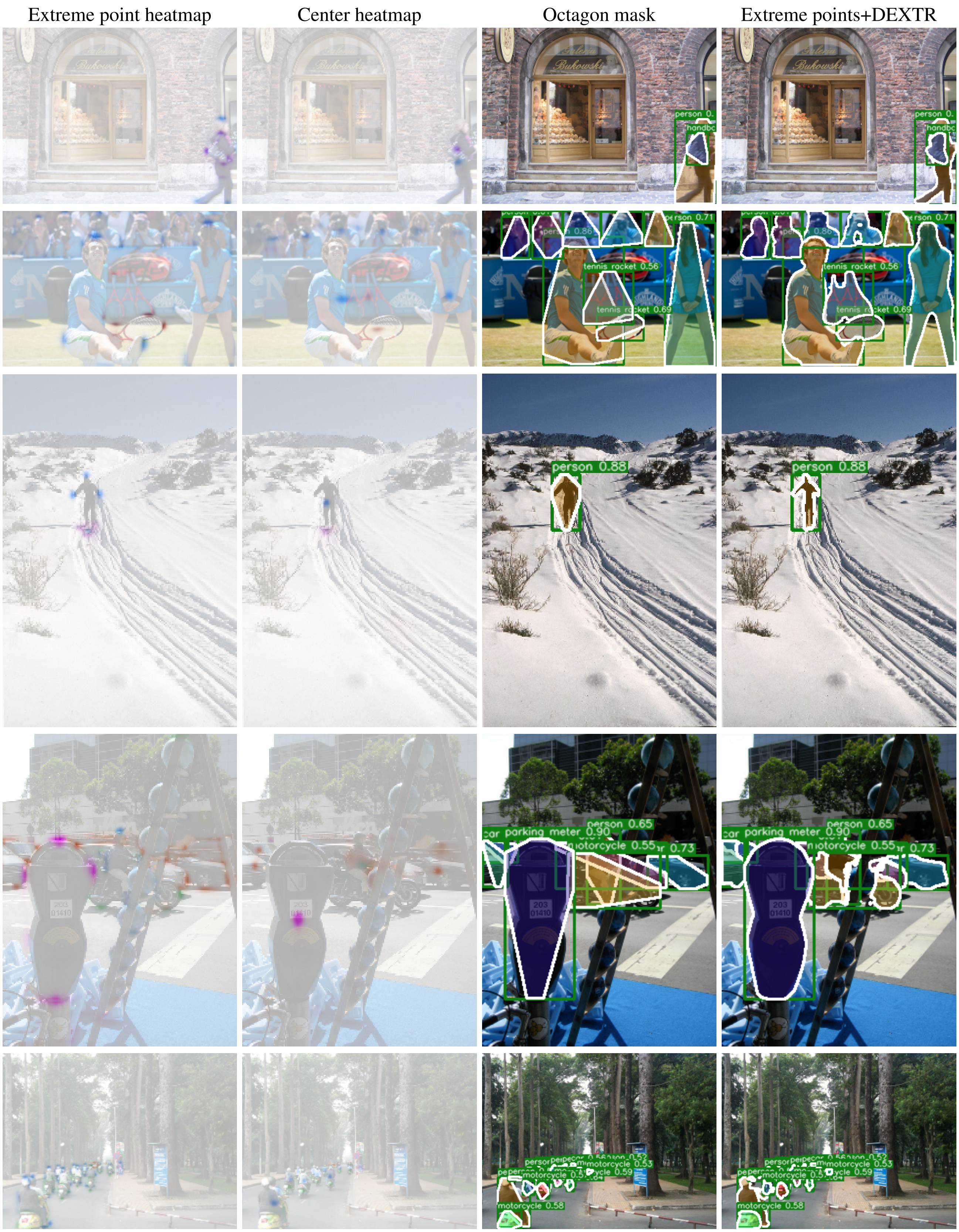


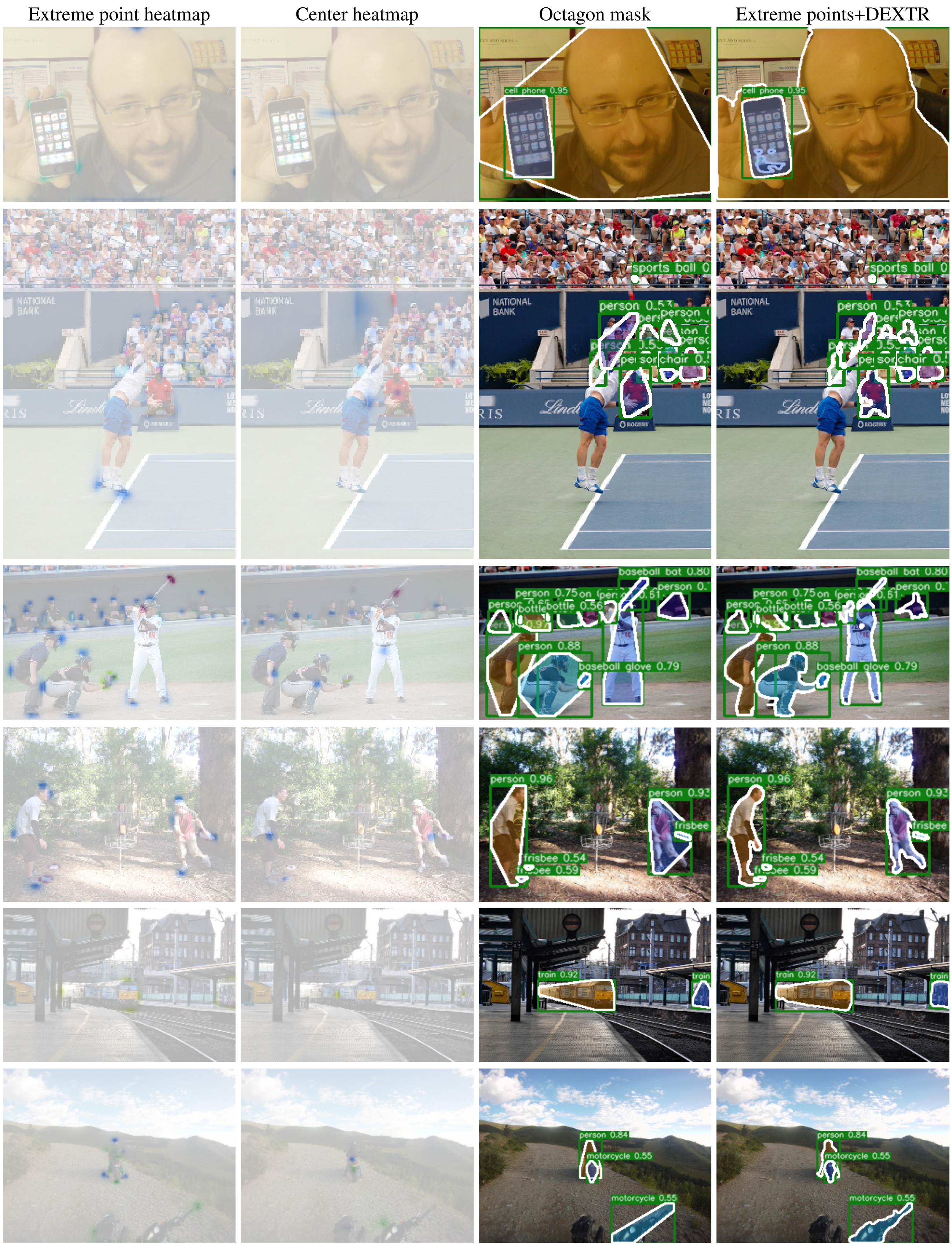

Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR


Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR
person 0.85
person 0.50
cell phone 0.56
cake 0.73
cell phone 0.52
umbrella 0.60
person 0.65
umbrella 0.77
person 0.56
cat 0.86
laptop 0.75

Extreme point heatmap	Center heatmap	Octagon mask	Extreme points+DEXTR




Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR
tennis racket 0.56
tennis racket 0.69
person 0.71
person 0.88
parking meter 0.90
person 0.65
motorcycle 0.55
motorcycle 0.58


Extreme point heatmap
Center heatmap
Octagon mask
Extreme points+DEXTR
cell phone 0.95
person 0.96
person 0.93
frisbee 0.54
frisbee 0.59
train 0.92
person 0.84
motorcycle 0.55
person 0.88
baseball glove 0.79
baseball bat 0.80

Extreme point heatmap | Center heatmap | Octagon mask | Extreme points+DEXTR

References

[1] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 1

[2] K.K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018. 1