

CenterNet: Objects as Points

一. 概要

目前大多数主流的目标检测器都是 anchor-based 的，但基于 Anchor 的目标检测器事实上存在着一些严重的问题，比如：（1）Anchor 的定义在一定程度上会影响或限制检测算法的性能；（2）NMS 等后处理操作会降低整个检测算法的速度。随着基于 Anchor 的目标检测性能逐渐达到极限，Anchor-free 的目标检测算法成为了研究热点，代表性工作包括 CornerNet、FOCS、ExtremeNet 等。

本文提出了一种新的 anchor-free 方法，叫做 CenterNet，将目标作为单个点（其对应 bounding box 的中心点）进行建模。具体来说，CenterNet 利用关键点估计来寻找中心点，并回归其他目标属性，例如尺寸，3D 位置，朝向，甚至姿态。与基于 anchor 的检测器相比，CenterNet 端到端，更简单，更快，更精确，在速度和精度上实现了最好的权衡。

二. 创新

1. 去除了低效复杂的 Anchor 操作，通过目标的中心点来表示目标，然后利用中心点位置的图像特征回归出目标的其他属性，例如：size, dimension, 3D extent, orientation, pose。这样，目标检测问题就变成了一个标准的关键点估计问题。仅将图像输入全卷积网络，生成一个 heatmap 热力图，热力图峰值点即目标中心点，每个峰值点处的图像特征即可预测目标 bounding box 的宽和高。

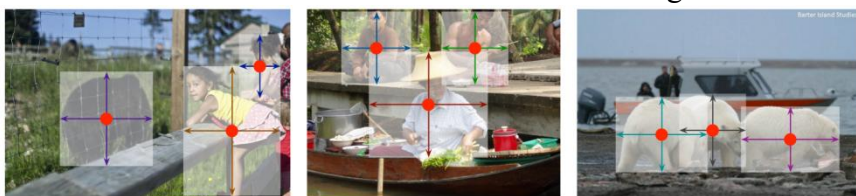
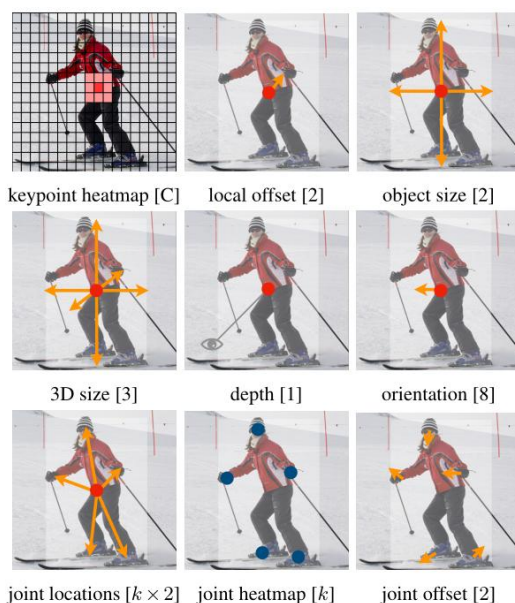


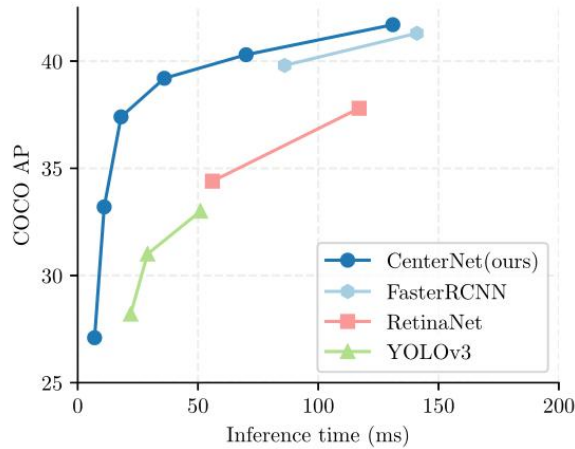
Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties are inferred from the keypoint feature at the center. Best viewed in color.

2. 直接在 heatmap 图上面执行了过滤操作，去除了耗时的 NMS 后处理操作，推理阶段仅前向传播，进一步提升了整个算法的运行速度；

3. 此模型具有通用性，可拓展进其他任务。对于 3D BBox 估计，直接回归得到目标的深度信息、3D 检测框的尺寸和朝向；对于人体姿态估计，将关节点（2D joint）位置作为中心点的偏移量，直接在中心点位置回归出偏移量的值。



CenterNet 模型去除了耗时的 Anchors 与 NMS 后处理操作，因此运行速度较快，精度也很高，适合部署在一些低性能的嵌入式设备中。



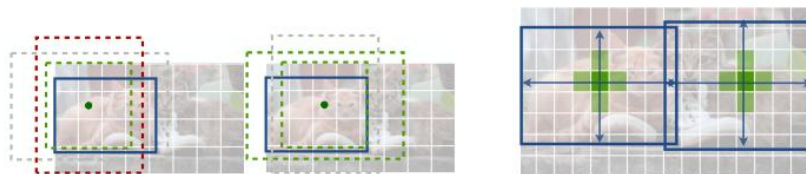
三. 与之前工作的对比

本文的方法与 anchor-based 单阶段方法紧密相关，一个中心点就可以被视为一个形状未知(shape-agnostic)的 anchor，但存在几个重要的不同：

第一，CenterNet 仅根据位置来分配 anchor，而不是框之间的 overlap。没有手动设计的用于前景背景分类的阈值。

第二，每个目标仅有一个 positive anchor，因此不需要进行 NMS。仅需在关键点 heatmap 上提取局部峰值点 (local peaks)；

第三，与传统目标检测器相比（下采样 16 倍），CenterNet 使用更大分辨率的输出特征图（下采样 4 倍），因此无需设置多尺度 anchor；



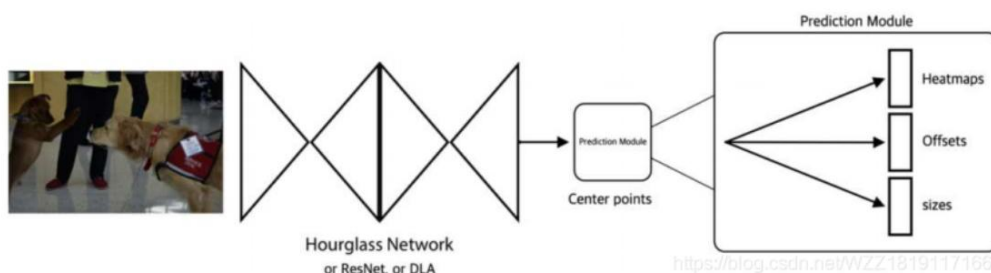
(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.

(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.

另一方面，本文并非第一个通过关键点估计来做目标检测的。CornerNet 将 bounding box 的两个角点作为关键点；ExtremeNet 检测目标的最上，最下，最左，最右极值点以及中心点。这些方法和 CenterNet 一样都建立在鲁棒的关键点估计网络上，然而，它们都需要经过一个关键点 grouping 阶段，这会降低算法整体的速度，而 CenterNet 仅提取每个目标的中心点，无需对关键点进行 grouping 或后处理。

四. CenterNet 整体架构



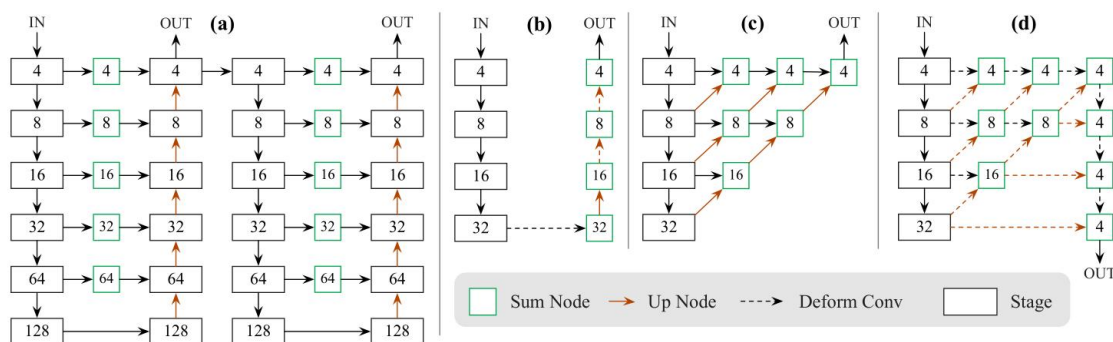
1. 最左边表示输入图片。输入图片需要裁减到 512*512 大小，即长边缩放到 512，短边补 0。

2. 中间表示基准网络，论文中尝试了 Hourglass、ResNet 与 DLA 这 3 种网络架构，各个网络架构的精度及帧率为：

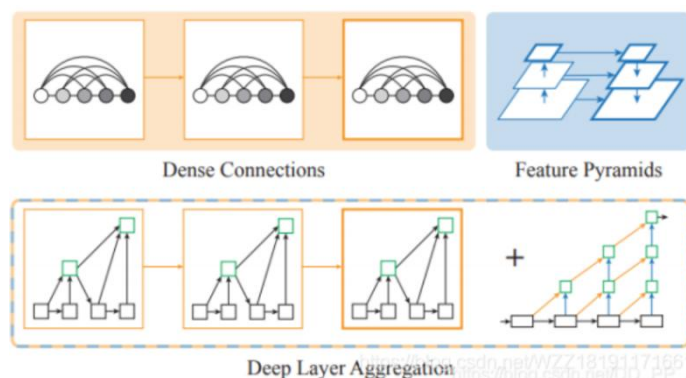
Hourglass-104: 40.3% COCOAP and 14 FPS

Resnet-18 with up-convolutional layers: 28.1% coco and 142 FPS

DLA-34: 37.4% COCOAP and 52 FPS



上图展示了 3 中不同的网络架构。图(a)表示 Hourglass 网络，该网络是在 ECCV2016 中的 Stacked hourglass networks for human pose estimation 论文中提出的一种网络，用来解决人体姿态估计问题，其思路主要通过将多个漏斗形状的网络堆叠起来，从而获得多尺度信息。图(b)表示带有反卷积的 ResNet 网络，作者在每一个上采样层之前增加了一个 3*3 的可变形卷积，即先使用可变形卷积来改变通道数，然后使用反卷积来对特征图执行上采样操作。图(c)表示用于语义分割的 DLA34 网络。图(d)表示修正的 DLA34 网络，该网络在原始的 DLA34 网络的基础上增加了更多的残差连接，该网络将 Dense Connection 与 FPN 的思路融合起来，前者源于 DenseNet，可以用来聚合语义信息，能够提升模型推断是“what”的能力；后者用来聚合空间信息，能够提升模型推断在“where”的能力，具体的细节如下图所示。

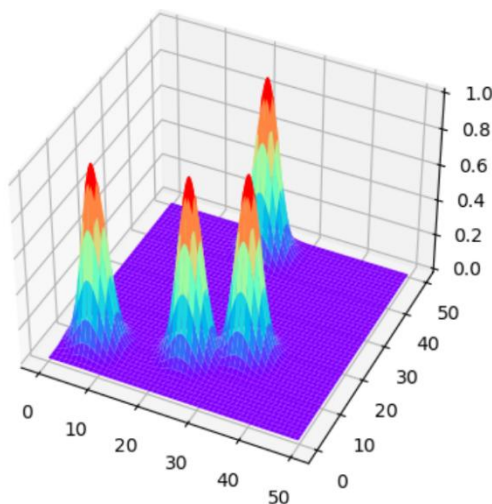


3. 最右边表示预测模块，该模块包含 3 个分支，具体包括中心点 heatmap 图分支、中心点 offset 分支、目标尺寸分支。heatmap 图分支包含 C 个通道，每一个通道包含一个类别，heatmap 中白色的亮区域表示目标的中心点位置；中心点 offset 分支用来弥补将池化后的低 heatmap 上的点映射到原图中所带来的像素误差；目标尺寸分支用来预测目标框的 w 与 h 偏差值。结构均为 $3*3conv+ReLU+1*1conv$ ，三个分支独立。

五. CenterNet 网络细节

1. 训练阶段 Heatmap 生成

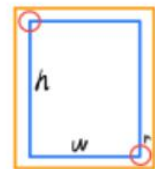
CenterNet 将目标检测问题转换成中心点预测问题，即用目标的中心点来表示该目标，并通过预测目标中心点的偏移量与宽高来获取目标的矩形框。Heatmap 通道为 C，对应 C 个类别（每个类别将会产生一张单独的 Heatmap 图）。在训练中，我们需要生成 heatmap 标签，对于每张 Heatmap 图而言，当某个坐标处包含目标的中心点时，则会在该目标处产生一个关键点，利用高斯圆来表示整个关键点，每个通道大概如下图所示，每个峰值所对应的位置代表一个目标的中心：



生成 GT heatmap 的具体步骤如下所示：

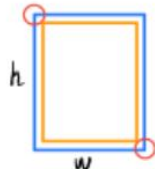
- **步骤 1**-将输入的图片缩放成 $512*512$ 大小，对该图像执行 $R=4$ 的下采样操作之后，获得一个 $128*128$ 大小的 Heatmap 图；
- **步骤 2**-将输入图片中的 Box 缩放到 $128*128$ 大小的 Heatmap 图上面，计算该 Box 的中心点坐标，并执行向下取整操作，将其定义为 keypoint；
- **步骤 3**-根据目标 Box 大小来计算高斯圆的半径 r；

关于高斯圆的半径确定，主要还是依赖于目标 box 的宽高，实际情况通常会取 $IOU=0.7$ ，即下图中的 $overlap=0.7$ 作为临界值，然后分别计算出三种情况的半径（情况 1 为预测框和 GT box 两个角点以 r 为半径的圆外切，情况 2 为预测框和 GT box 两个角点以 r 为半径的圆内切，情况 3 为预测框和 GT box 两个角点以 r 为半径的圆一个边内切，一个边外切），取三者的最小值作为高斯核的半径 r。将每种情况的 IoU 公式分别展开之后，均成为二元一次方程的求解问题。具体细节如下图所示：



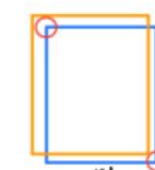
$$IoU = \frac{h \times w}{(h+2r) \times (w+2r)}$$

$$4 \times IoU \times r^2 + 2 \times IoU \times (h+w) \times r + (IoU-1) \times (h \times w) = 0$$

$$\begin{aligned} a &= 4IoU \\ b &= 2IoU \times (h+w) \\ c &= (IoU-1) \times (h \times w) \end{aligned} \quad \begin{aligned} \Delta &= b^2 - 4ac \\ r &= \frac{-b + \sqrt{\Delta}}{2a} \end{aligned}$$


$$IoU = \frac{(h-2r) \times (w-2r)}{h \times w}$$

$$4r^2 - 2(h+w)r + (1-IoU)hw = 0$$

$$\begin{aligned} a &= 4 \\ b &= -2(h+w) \\ c &= (1-IoU)hw \end{aligned} \quad \begin{aligned} \Delta &= b^2 - 4ac \\ r &= \frac{-b + \sqrt{\Delta}}{2a} \end{aligned}$$


$$IoU = \frac{(h-r) \times (w-r)}{2hw - (h-r)(w-r)}$$

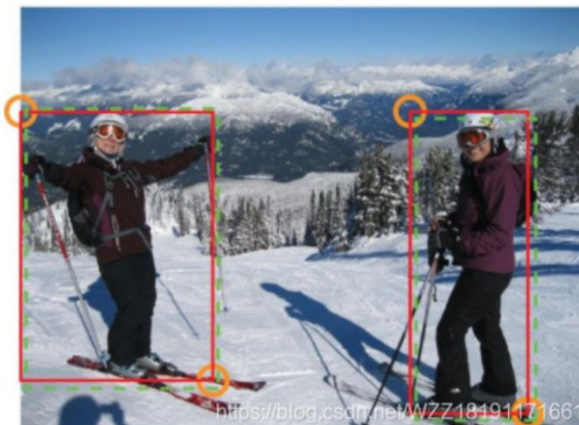
$$r^2 - (h+w)r + \frac{(1-IoU)wh}{1+IoU} = 0$$

$$\begin{aligned} a &= 1 \\ b &= -(h+w) \\ c &= \frac{(1-IoU)wh}{1+IoU} \end{aligned} \quad \begin{aligned} \Delta &= b^2 - 4ac \\ r &= \frac{-b + \sqrt{\Delta}}{2a} \end{aligned}$$

<https://blog.csdn.net/WZZ18191171661>

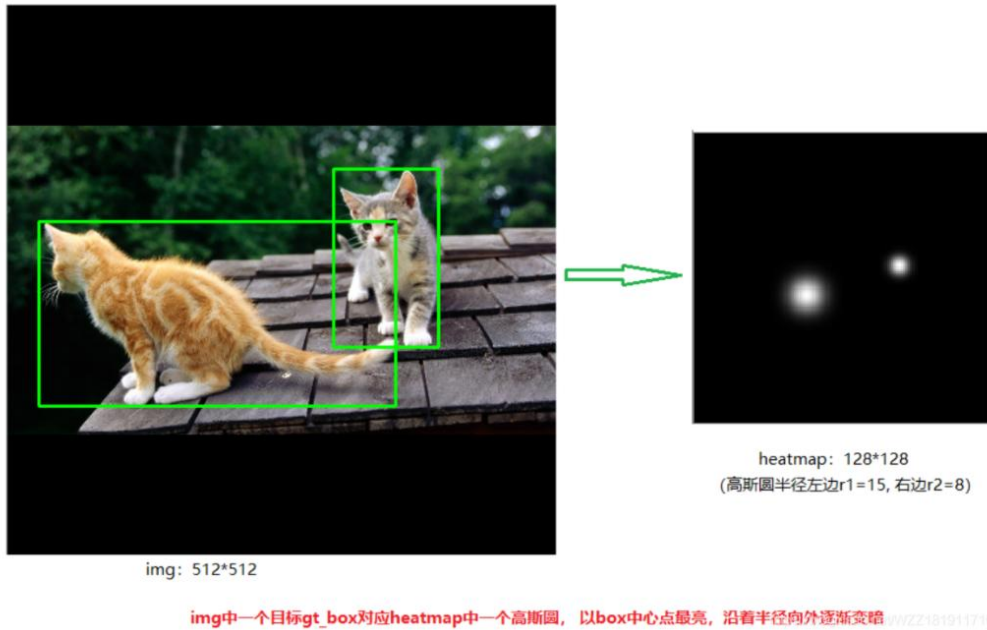
- **步骤 4-**在 128*128 大小的 Heatmap 图上面，以 keypoint 为中心点，半径为 r 填充高斯值，keypoint 点处数值最大，周围的标签呈 $e^{-\frac{x^2+y^2}{2\sigma^2}}$ 规律递减，其中标准差也是根据半径 r 自适应求取的。

注：Heatmap 上的关键点之所以采用二维高斯核来表示，是由于对于在目标中心点附近的一些点，其预测出来的框与真实框的 IOU 可能会大于 0.7，也可以很好地包围该目标，不能直接对这些预测值进行惩罚（即不能直接将对应位置标签置为 0），所以采用高斯核来进行缓冲。



通俗来讲，只要预测的中心点在真实中心点的某一个半径 r 内，而且该矩形框与真实框之间的 IoU 大于阈值 0.7 时，就可以将这些点处的值设置为一个高斯

分布的数值，而不是直接置为 0。



上图展示了一个样例，左边表示经过裁剪之后的 512*512 大小的输入图片，右边表示经过高斯操作之后生成的 128*128 大小的 Heatmap 图。由于图中包含两只猫，这两只猫属于一个类别，因此在同一个 Heatmap 图上面生成了两个高斯圆，高斯圆的大小与矩形框的大小有关。

通过上述步骤，我们就得到了 heatmap 标签。实际训练过程中，CenterNet 会在输出 heatmap 后找出所有局部峰值(若某个值大于等于其周围 8 个值则视为峰值)，再从中选择峰值大于一定阈值的做为正样本，其峰值所对应的位置则为目标的大致中心位置(此过程类似简化版的 NMS)。因为输出的 heatmap 相比原图像的尺寸缩小了步长 R 倍，这会带来中心点定位的偏差。为了解决这个问题，作者又设置了两个通道的输出来回归每个位置的偏差 offset，从而完成中心点的定位任务。

2. 损失函数

整个 CenterNet 的损失函数包含 3 个部分， L_k 表示 heatmap 中心点损失， L_{off} 表示目标中心点偏移损失， L_{size} 表示目标长宽损失函数。

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \\ (\lambda_{size} = 0.1, \lambda_{off} = 1)$$

(1) Heatmap 损失函数

$$L_K = \frac{-1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}), & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^\beta (\hat{Y}_{xyc})^\alpha \log(1 - \hat{Y}_{xyc}), & \text{otherwise} \end{cases} \\ (\text{其中超参数: } \alpha = 2, \beta = 4)$$

此函数是在 Focal Loss 的基础上进行了改进，其中的 α 与 β 是两个超参数，分别用来均衡难易样本不平衡和正负样本不平衡问题。 Y_{xyc} 表示 GT 值， \hat{Y}_{xyc} 表示预测值，N 表示关键点的个数。

- 当 $Y_{xyz}=1$ 时，易分类样本的预测值接近为 1，此时 $(1-\hat{Y}_{xyz})^\alpha$ 就表示一个很小的数值，其损失被大大衰减，该样本权重相对被降低。
- 当 $Y_{xyz}=1$ 时，难分类样本的预测值接近为 0，此时 $(1-\hat{Y}_{xyz})^\alpha$ 就表示一个较大的数值，可近似认为其损失与原来相比变化不大，该样本权重相对被增加。
- 当 $Y_{xyz} \neq 1$ 时，利用 $(\hat{Y}_{xyz})^\alpha$ 来充当惩罚项，原理与上面正好相反。
- 而 $(1-Y_{xyz})^\beta$ 这个参数，距离中心点越近，其数值越小，相当于弱化了对真实中心点周围负样本的惩罚。

$(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}, y_2^{(k)})$ 表示类别 c 中目标 k 的 bounding box 坐标，则其中心点坐标可表示为 $(\frac{x_1^{(k)} + x_2^{(k)}}{2}, \frac{y_1^{(k)} + y_2^{(k)}}{2})$ 。

(2) 中心点偏移损失函数

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

其中 $\hat{O}_{\tilde{p}}$ 表示网络预测的偏移量数值， p 表示图像中心点坐标， R 表示 Heatmap 的缩放因子， \tilde{p} 表示缩放后中心点的近似整数坐标，整个过程利用 L1 Loss 计算正样本块的偏移损失。由于 backbone 输出的 feature map 的空间分辨率是原始输入图像的四分之一，即输出 feature map 上的每一个像素点对应到原始图像的一个 4x4 区域，这会带来较大的量化误差，因此引入了偏置损失。

假设目标中心点 p 为(125, 63)，由于输入图片大小为 512*512，缩放尺度 $R=4$ ，因此缩放后的 128*128 尺寸下中心点坐标为(31.25, 15.75)，相对于整数坐标(31, 15)的偏移值即为(0.25, 0.75)。

(3) 目标长宽损失函数

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

其中 N 表示关键点的个数， s_k 表示目标的真实尺寸， \hat{S}_{pk} 表示预测的尺寸，整个过程利用 L1 Loss 来计算正样本的长宽损失。 $s_k = (x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)})$

3. CenterNet 推理阶段

CenterNet 网络的推理阶段的实现步骤如下所述：

- 步骤 1-将输入图片缩到 512*512 大小；
- 步骤 2-对输入图片执行下采样，并对下采样后的图像执行预测，即在 128*128 大小的 Heatmap 上执行关键点预测；
- 步骤 3-在 128*128 大小的 Heatmap 上面采用 3*3 大小的最大池化操作来获取 Heatmap 中满足条件的关键点（类似于 anchor-based 检测中 NMS 的效果），并选取 100 个关键点；
- 步骤 4-将关键点预测值作为分类的 confidence，对应位置的 bounding box 为

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

网络预测效果如下图所示：



4. 在 3D 目标检测任务上的应用

对于 3D 检测任务中的 object，确定中心点的同时，还需要确定中心点的三个附加属性：depth、3D dimension 和 orientation，这 3 个附加属性通过 3 个不同的 head 进行预测。对于第 k 个 object，从 3 个输出特征图上提取输出结果表示为 $\hat{d}_k \in R, \hat{\gamma}_k \in R^3, \hat{\alpha}_k \in R^8$ 。

(1) depth 回归

网络预测 depth 输出为 $\hat{D} \in R^{\frac{W}{R} \times \frac{H}{R}}$ 。depth 很难直接去回归，作者令网络的输出经过变换 $d = 1/\sigma(\hat{d}) - 1$ 之后得到网络的 depth，其中， \hat{d} 为网络输出的预测值，d 为变换后的 depth， σ 为 sigmoid 函数。在 backbone 之后接卷积+ReLU+卷积进行 depth 的预测，输出经 sigmoid 取倒数后-1 变换到真实的 depth 尺度上，利用 L1 距离计算损失，损失表示为：

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

其中， d_k 为真实的 depth。

(2) 3D dimension 回归

网络预测的 3d 维度输出为 $\hat{\Gamma} \in R^{\frac{W}{R} \times \frac{H}{R} \times 3}$ 。预测 3D dimension，直接回归，L1 损失表示为：

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

(3) orientation 回归

网络预测 orientation 输出为 $\hat{A} \in R^{\frac{W}{R} \times \frac{H}{R} \times 8}$ 。原本 orientation θ 是 1 个标量，作者这里使用 Mousavian et al. 的想法，使用 8 个标量去编码方向，从而能够更容易训练。

这 8 个标量被分成两组，每组记为一个 bin，一组的角度范围是 $B_1 = [-\frac{7\pi}{6}, \frac{\pi}{6}]$ ，另一组的角度范围是 $B_2 = [-\frac{\pi}{6}, \frac{7\pi}{6}]$ ，每个 bin 用 4 个标量表示，其中两个标量 $b_i \in R^2$ 用于 softmax 分类，剩下两个标量 $a_i \in R^2$ 表示 θ 到 bin 中心 m_i 的 offset 的 sin 和 cos 值。因此一个 bin 的预测结果为 $\hat{\alpha} = [\hat{b}_1, \hat{a}_1, \hat{b}_2, \hat{a}_2]$ ，其损失函数表示为：

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

其中， $c_i = \mathbb{1}(\theta \in B_i)$ ， $a_i = (\sin(\theta - m_i), \cos(\theta - m_i))$ ， c_i 表示示性函数。最终的预测角度由 8 个标量编码，表示为：

$$\hat{\theta} = \arctan2(\hat{a}_{j1}, \hat{a}_{j2}) + m_j$$

其中j是具有较大分类 score 的那个 bin。

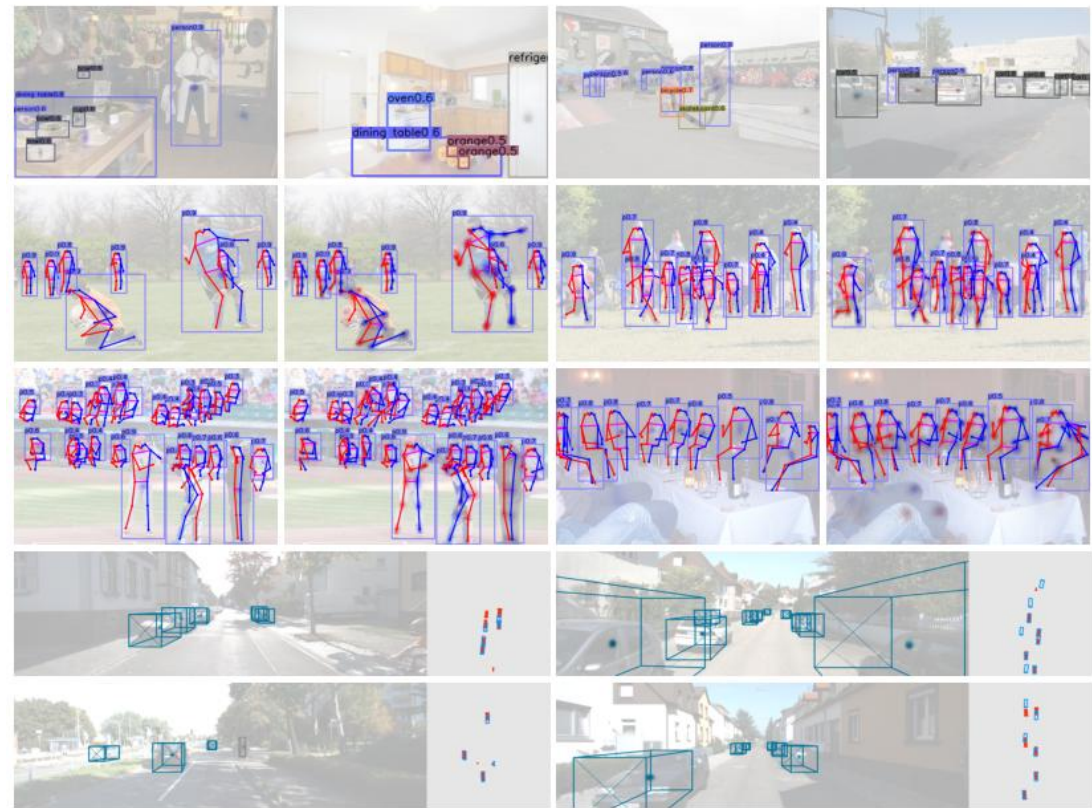
$$\text{atan2}(y,x)=\begin{cases} \arctan(\frac{y}{x}) & x>0 \\ \arctan(\frac{y}{x})+\pi & y\geq 0,x<0 \\ \arctan(\frac{y}{x})-\pi & y<0,x<0 \\ +\frac{\pi}{2} & y>0,x=0 \\ -\frac{\pi}{2} & y<0,x=0 \\ \text{undefined} & y=0,x=0 \end{cases}$$

六. CenterNet 效果展示与分析

	AP			AP ₅₀			AP ₇₅			Time (ms)			FPS		
	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS
Hourglass-104	40.3	42.2	45.1	59.1	61.1	63.5	44.0	46.0	49.3	71	129	672	14	7.8	1.4
DLA-34	37.4	39.2	41.7	55.1	57.0	60.1	40.8	42.7	44.9	19	36	248	52	28	4
ResNet-101	34.6	36.2	39.3	53.0	54.8	58.5	36.9	38.7	42.0	22	40	259	45	25	4
ResNet-18	28.1	30.0	33.2	44.9	47.5	51.5	29.6	31.6	35.1	7	14	81	142	71	12

Table 1: Speed / accuracy trade off for different networks on COCO validation set. We show results without test augmentation (N.A.), flip testing (F), and multi-scale augmentation (MS).

上表展示了 CenterNet 目标检测在 COCO 验证集上面的精度与速度。通过观察可以发现，基于 DLA-34 的基准网络能够在精度与速度之间达到一个折中。



上图展示了 CenterNet 目标检测算法、CenterNet 人体位姿估计算法、CenterNet 3D 目标检测算法在一些复杂的测试场景上面的测试效果。通过观察我们可以发现该算法在不同的复杂场景下均可以取得较好的效果。

（本文部分内容整理自 CSDN 博客）