# CornerNet：基于关键点的目标识别

## abstract:

CornerNet这种新型目标检测方法是使用单个卷积神经网络将目标边界框检测为一对关键点即(左上角和右下角)。

相比现有单级检测方法：

1、消除了anchor box（先验框，锚盒）。

2、引入了corner pooling（角点汇集），一种汇集层有助于网络更好地定位角点。

## 1 Introduction

One-stage检测将anchor box密集地放置在图像上，通过对anchor box评估和回归提炼其坐标来生成最终的预测。

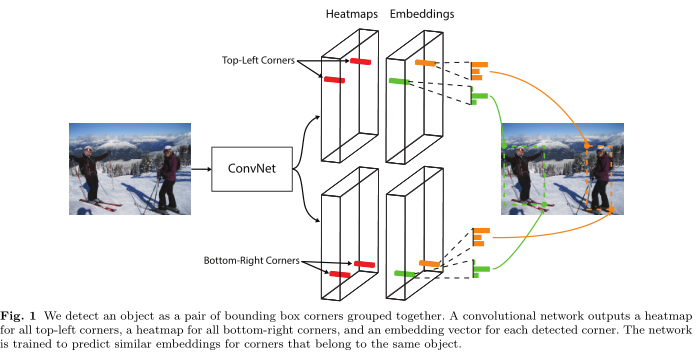
作者提出使用Anchor box两个缺点：

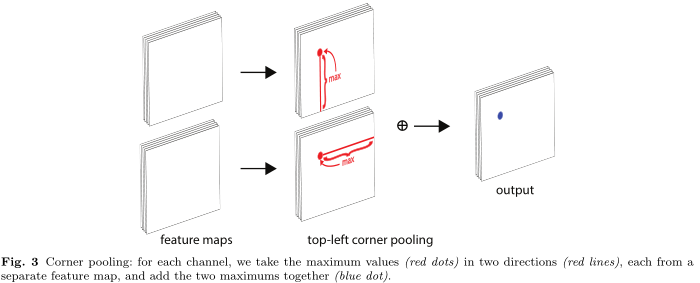
1. 通常需要一套很多的anchor boxes。检测器被训练来分类这些anchor box是否与truth boxes（即真实目标）重合，并需要大量的anchor box与 truth box重合，结果只有小部分anchor box与truth box重合。这样会造成正负锚盒数量的巨大不平衡，并减缓了训练的速度。
2. Anchor box的使用引入了许多的超参数和设计选择（包括盒子数量、尺寸比例）。这种选择通常通过ad-hoc heuris-tics（自组织试探法）做出来的，并与多尺度体系结构结合时变得更加复杂。在多尺度体系结构中，单个网络以多个分辨率做出单独的预测，每个尺度使用不同的特征和自己的锚盒集合。

本篇论文介绍的CornerNet，是一种用于目标检测新型的One-stage方法。”新型”体现在其取消了anchor box，将目标检测为一对关键点——边界框的左上角和右下角。

CornerNet使用单个卷积神经网络来预测同一目标类别的所有实例的左上角、右下角的heatmap（热图），以及每个检测到的角的嵌入向量。嵌入向量的作用是对同一个目标的一对角进行分组，训练网络来预测它们的相似嵌入。

图Fig1，即为CornerNet整体流水线。



CornerNet另一个特点是引入了corner pooling（角池），一种新型的池层，来帮助卷积神经网络更好的定位边界框的角。角点池化层：在每个像素位置，最大化第一个特征图右侧的所有特征向量，最大化第二个特征图正下方的所有特征向量，然后将两者结果合并在一起。如图Fig3

为什么检测corner比bounding box更加有效呢？作者假设了两点原因。1、盒子中心更难定位，中心依赖目标所有的4个边界，而一个角依赖2个边，因此角池更容易。2、角点提供了一种更高效的密集离散化盒子空间的方法：只需O(wh)个角点来表示O(w2h2)个可能的anchor boxes。

Corner pooling对CornerNet至关重要。

## 2 Related wrok

### 2.1 Two-stage object detectors

Two-stage方法首先由R-CNN引入，两阶段检测器通过生成一组稀疏的interest区域，并通过网络对每一个interest区域进行分类。R-CNN使用低水平视觉算法生成ROIs，然后从图像中提取每一个区域，并由ConvNet独立处理，这产生了大量冗余计算。

后来SPP和Fast-RCNN通过设计一个特殊的池化层来改进R-CNN，该层将每个区域从特征图中池化。但两者依然依赖separate proposal algorith-ms 并且不能被end-to-end（不能被反复训练）地训练。

Faster-RCNN引入了RPN（region proposal network）来消除low level proposal algorithms，该网络从一组预定的候选框生成proposals。这不仅使检测器更有效，而且允许检测器被从头到尾训练。

R-FCN通过用全卷积子检测网络代替全连通子检测网络，进一步提高了Faster-RCNN的效率。

之后其他的研究侧重于纳入子类别信息，利用更多的上下文信息在多个尺度上生成目标proposals。

### 2.2 One-stage object detectors

One-stage方法的代表是YOLO和SSD，One-stage取消了Rol池化步骤，且在单个网络中检测对象。

One-stage方法通常在保持性能的同时比Two-stage方法更高效。

SSD将anchor box密集地放置在多个比例的特征图上，直接对每个锚盒进行分类和细化。

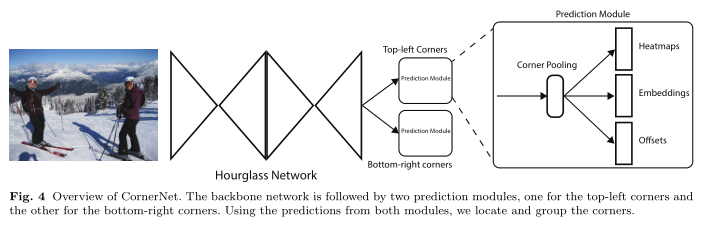
YOLO直接从图像中预测边界框坐标，后来在YOLO9000中通过切换到锚框进行了改进。

DSSD和RON采用了类似沙漏网络的网络，使他们能够通过跳跃连接将低级和高级特征结合起来，从而更准确地预测边界框。

RetinaNet作者认为密集的锚盒在训练期间造成了正负锚盒之间的巨大不平衡。这种不平衡导致训练效率低下，因此性能不佳。故提出了一种新的损失，焦点损失，来动态调整每个锚盒的权重。

## 3 CornerNet

### 3.1 Overview



CornerNet以Hourglass Network作为主干网络，后面是两个预测模块，一个用于左上角，另一个用于右下角。其中，每个模块都有自己的corner pooling module用于汇集沙漏网络的特征（在预测热图、嵌入和偏移之前）。

不同于其他目标检测，CornerNet不使用不同尺度的特征来检测不同尺寸的物体，只对沙漏网络的输出应用上述的两个预测模块。

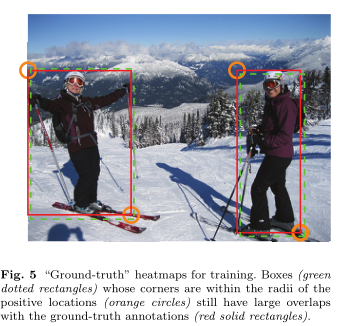
### 3.2 Detecting Corners

在训练过程中，如图5所示，由于在正位置附近的负位置产生的边界框依然会和真实目标有高度的重合，所以在正位置为圆心的某个半径的圆中的负位置的罚减量应适度减小。

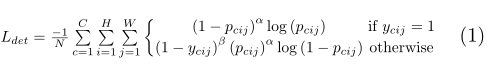
在给定半径的情况下，对负位置的罚减量是由一个非正规2D高斯函数



给出，其中圆中心在正位置，σ为半径的1/3。



令pcij 为在预测热图中c类在（i，j）的得分值，ycij 为被非正高斯扩充真实目标的热图。



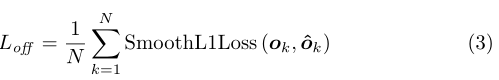
其中，N是图像中物体的数量，α和β是控制每个点的贡献的超参数(在所有实验中，我们将α设置为2，将β设置为4)。由于高斯凸起编码在ycij中，(1 -ycij)项减少了真实目标位置周围的损失。

众多网络涉及对层进行下采样，以收集全局信息并减少内存使用。当以完全卷积的方式应用于图像时，输出的大小通常小于图像，一个图像的位置（x，y）被映射到热图的（n为下采样因子）。当我们重新将热图映射到输入图片就可能会损失一些精度，就可能会影响带有真实目标的边界框。为了解决这个问题，我们预测位置偏移，以便在将角位置重新映射到输入分辨率之前稍微调整角位置。



Ok是偏移量，xk，yk为角K的坐标。

特别地，我们预测所有类别的左上角共享一组偏移量，右下角共享另一组偏移量。在训练中，我们在真实目标角的位置应用the smooth L1 Loss。



### 3.3 Grouping Corners

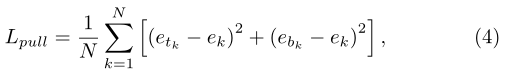
怎么将检测出来的多组左上角右下角进行匹配，认定来自同一个边界框？

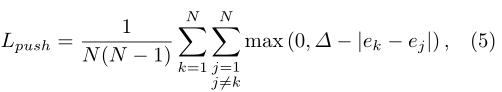
作者受the Associative Embed-ding method proposed by Newell et al. (2017) for thetask of multi-person pose estimation（多人姿态估计任务嵌入方法）的启发。（检测所有的人类关节，并为每个检测到的关节生成一个嵌入。他们根据嵌入物之间的距离对接缝进行分组。）

CornerNet为每个检测到的角预测一个嵌入向量，如果左上角和右下角属于同一个边界框，它们的嵌入之间的距离应该很小。然后，可以根据左上角和右下角嵌入点之间的距离对角进行分组。嵌入的实际值并不重要。只有嵌入物之间的距离用于对角进行分组。

CornerNet使用的一维嵌入。

令etk 为对象K左上角的嵌入向量，ebk 为右下角嵌入。如Newell and Deng (2017)，使用“pull”损失来训练网络以分组角点，使用“push”损失来分离角点：

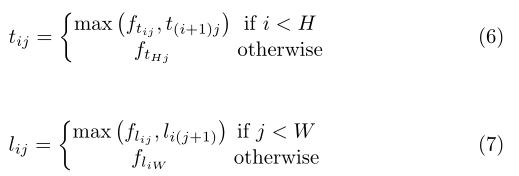




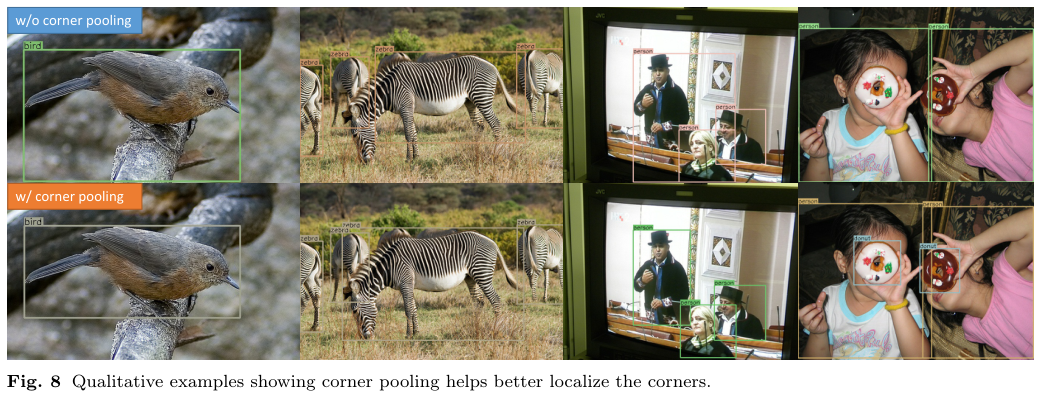
ek是etk和ebk的平均值，将所有实验的∆设置为1。与偏移损失相似，只在真实目标位置应用损失。

### 3.4 Corner Pooling

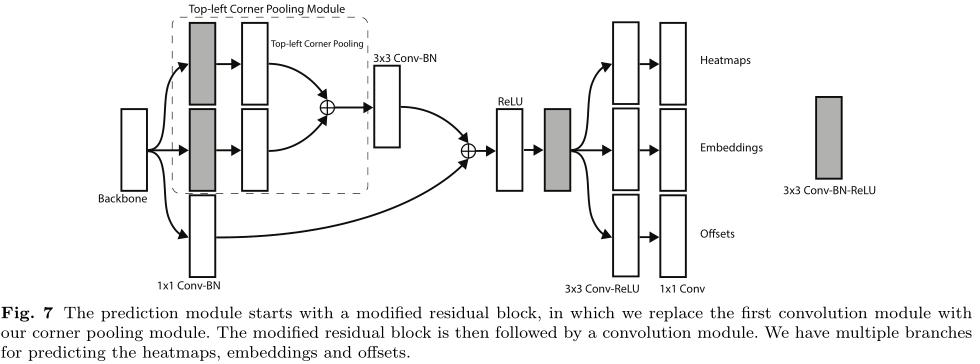
假设我们想要确定位置(i，j)处的像素是否是左上角。让ft 和 fl成为输入到左上角汇集层的特征映射，让ftij和 flij分别成为ft和 fl中位置(I，j)处的向量。对于H × W特征映射，角点汇集层首先将ft中(i，j)和(i，H)之间的所有特征向量最大汇集到特征向量tij，并将fl中(i，j)和(W，j)之间的所有特征向量最大汇集到特征向量lij。最后，它将tij和 lij合并在一起。该计算可由以下等式表示:



在这里我们应用元素方式的最大操作。如图8所示，tij和lij都可以通过动态编程来有效地计算。



同理，判断右下角的位置使用相同的办法。



预测模块从修改后的残差块开始，在残差块中，我们用角池模块替换第一个卷积模块。修改后的残差块之后是卷积模块。我们有多个分支来预测热图、嵌入和偏移。

### 3.5 Hourglass Network

沙漏模块首先通过一系列卷积和最大池层对输入要素进行下采样。然后，它通过一系列上采样和卷积层将要素上采样回原始分辨率。由于最大池图层中的细节会丢失，因此会添加跳过图层，以将细节带回上采样的要素。沙漏模块在一个统一的结构中捕获全局和局部特征。当多个沙漏模块堆叠在网络中时，沙漏模块可以重新处理特征以捕获更高级别的信息。这些属性使得沙漏网络也成为对象检测的理想选择。事实上，目前的许多探测器已经采用了类似沙漏网络的网络。

CornerNet 包括两个沙漏模块并在沙漏结构上做了一些改变。 使用步幅为2减小特征分辨率，不使用max pooling。特征分辨率减少5倍，增加特征通道数（256，384，384，384，512）。当对特征进行上采样时，我们使用2个residual module后接着最近neighbor upsampling。每一个skip connection都包括两个residual module。

沙漏模块中间有4个剩余模块，有512个通道。在沙漏模块之前，我们使用7 × 7卷积模块将图像分辨率降低了4倍，该卷积模块具有步幅2和128个通道，随后是具有步幅2和256个通道的剩余块。

在此之后，作者在训练中还增加了中间监督。但是没有增加中间预测，以免降低网络的性能。在第一个沙漏模块的输入输出都使用了一个1 x 1 Conv-BN模块。通过逐元素加法合并它们，接着是ReLU和具有256个通道的剩余块，然后将其用作第二个沙漏模块的输入。沙漏网络的深度是104。与许多其他最先进的探测器不同，我们只使用整个网络最后一层的特征来进行预测。

## 4 Experiments

### 4.1 Training Details

在PyTorch中实现了corner net。网络在PyTorch的默认设置下随机初始化，不需要对任何外部数据集进行预处理。当应用焦点损失时，遵循(Linet al., 2017)来设置预测拐角热图的卷积层中的偏差。在训练过程中，我们将网络的输入分辨率设置为511 × 511，这导致输出分辨率为128 × 128。为了减少过度拟合，我们采用了标准的数据增强技术，包括随机水平翻转、随机缩放、随机裁剪和随机颜色抖动，包括调整图像的亮度、饱和度和对比度。最后，我们将PCA (Krizhevsky等人，2012)应用于输入图像。

我们使用Adam (Kingma和Ba，2014)来优化完全训练损失:



其中，α、β和γ分别是拉、推和偏置损耗的权重。将α和β设置为0.1，将γ设置为1。我们发现1或更大的α和β值会导致较差的性能。我们使用49的批量大小，在10个Titan X(PASCAL)GPU上训练网络(主GPU上4个图像，其余GPU每个GPU 5个图像)。为了节省GPU资源，在我们的消融实验中，我们以2.5×104的学习速率训练网络进行250k次迭代。当我们将我们的结果与其他检测器进行比较时，我们对网络进行额外的250k次迭代训练，并在最后50k次迭代中将学习速率降低到2.5×105。