

# CornerNet: Detecting Objects as Paired Keypoints

Hei Law, Jia Deng

University of Michigan, Ann Arbor

ECCV 2018

## 摘要

我们提出了 CornerNet, 这是一种新的物体检测方法, 我们使用单个卷积神经网络为边界框检测为一对关键点, 即左上角和右下角。通过将物体检测为成对的关键点, 我们无需设计一组在先前单阶段检测器中常用的锚框。除了我们的新提法之外, 我们还引入了角池化, 这是一种新型的池化层, 可帮助网络更好地定位角点。实验表明, CornerNet 在 MS COCO 上实现了 42.1% 的 AP, 优于所有现有的单阶段检测器。

## 1 简介

基于卷积神经网络 (ConvNets) [20, 36, 15] 的对象检测器在各种具有挑战性的基准测试 [24, 8, 9] 上取得了最先进的结果。最先进方法的一个常见组成部分是锚框 [32, 25], 它们是各种大小和纵横比的框, 用作检测候选者。锚框广泛用于一级检测器 [25, 10, 31, 23], 可以实现与二级检测器 [32, 12, 11, 13] 向媲美的结果, 同时更高效。一级检测器将锚框密集地放置在图像上, 并通过对锚框进行评分并通过回归细化其坐标来生成最终的框预测。

但是使用锚框有两个缺点。首先, 我们通常需要一组非常大的锚框, 例如在 DSSD [10] 中超过 40k, 在 RetinaNet [23] 中超过 100k。这是因为检测器被训练来分类每个 anchor box 是否与一个 ground truth box 充分重叠, 并且需要大量的 anchor box 来确保与大多数 ground truth box 有足够的重叠。结果, 只有一小部分锚框会与地面实况重叠; 这在正负锚框之间造成了巨大的不平衡, 并减慢了训练速度 [23]。

其次，锚框的使用引入了许多超参数和设计选择。这些包括多少个框、什么尺寸和什么纵横比。这种选择主要是通过临时启发式进行的，当与多尺度体系结构结合时会变得更加复杂，其中单个网络在多个分辨率下进行单独预测，每个尺度使用不同的特征和它自己的一组锚框 [25, 10, 23]。

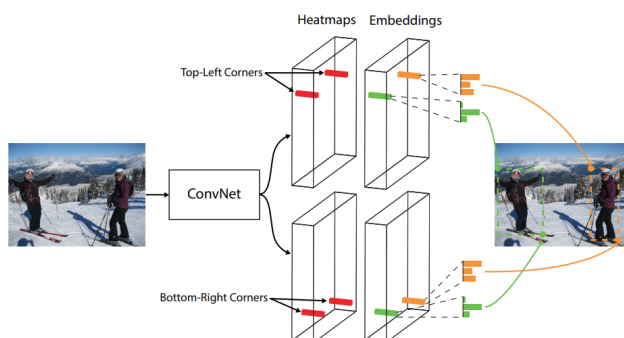


图 1: 我们将一个对象检测为一组组合在一起的边界框角。卷积网络输出所有左上角的热图、所有右下角的热图以及每个检测到的角的嵌入向量。该网络经过训练，可以预测属于同一对象的角点的相似嵌入。

在本文中，我们介绍了 CornerNet，这是一种新的单阶段目标检测方法，无需锚框。我们将一个对象检测为一对关键点——边界框的左上角和右下角。我们使用单个卷积网络来预测同一对象类别的所有实例的左上角的热图、所有右下角的热图以及每个检测到的角的嵌入向量。嵌入用于对属于同一对象的一对角进行分组——网络被训练为预测它们的相似嵌入。我们的方法极大地简化了网络的输出并消除了设计锚框的需要。我们的方法受到 Newell 等人??提出的关联嵌入方法的启发，他们在多人人体姿势估计的背景下检测和分组关键点。图1说明了我们方法的整体流程。



图 2: 通常没有本地证据来确定边界框角的位置。我们通过提出一种新型的池化层来解决这个问题。

CornerNet 的另一个新颖组件是角池化，这是一种新型的池化层，可帮助卷积网络更好地定位边界框的角点。边界框的角通常在对象之外——考虑圆形的情况以及图2中的示例。在这种情况下，不能基于局部证据来定位角。相反，要确定某个像素位置是否有左上角，我们需要水平向右看对象的最上边界，垂直向下看最左边界。这激发了我们的角池层：它接受两个特征图；在每个像素位置，它最大池化第一个特征图右侧的所有特征向量，最大池化第二个特征图正下方的所有特征向量，然后将两个合并的结果加在一起。一个例子如图3所示。

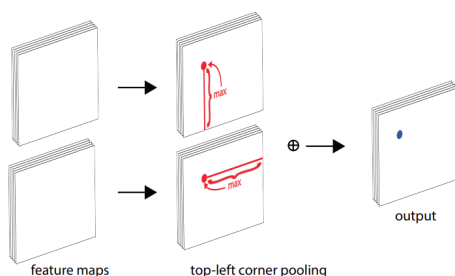


图 3: 角池化：对于每个通道，我们取两个方向（红线）上的最大值（红点），每个方向都来自一个单独的特征图，并将两个最大值相加（蓝点）

我们假设检测角点比边界框或候选的中心更有效的两个原因。首先，一个盒子的中心可能更难定位，因为它取决于对象的所有 4 个边，而定位一个角取决于 2 个边，因此更容易，对于角池化更是如此，它编码了一些关于角定义的显式先验知识。其次，角提供了一种更有效的方法来密集离散框的空间：我们只需要  $O(wh)$  个角来表示  $O(w^2h^2)$  个可能的锚框。

我们证明了 CornerNet 在 MS COCO [24] 上的有效性。CornerNet 实现了 42.1% 的 AP, 优于所有现有的单级检测器。此外, 通过消融研究, 我们表明角池化对于 CornerNet 的卓越性能至关重要。代码可在 <https://github.com/umichvl/CornerNet> 获得。

## 2 CornerNet