CornerNet：目标检测中的一种新方法

汤浩霖
21821298
浙江大学计算机学院

摘　要

目标检测是计算机视觉领域一直以来的研究热点。本文首先讨论了目标检测算法近年来的发展脉络，包括传统的检测算法与基于深度学习的检测算法。然后在此基础上，介绍了一种区别于目前主流方法的具有特色的目标检测新方法*CornerNet*，它将目标边界框的检测变为对边界框左上角与右下角这一成对顶点的检测。在*MS COCO* 数据集上的实验结果表明，其优于当前所有的一阶段目标检测方法。

1. 引言

目标检测是计算机视觉领域中的经典问题，一直以来都是计算机视觉中具有挑战性的研究热点，受到许多研究者的关注。目标检测技术当前也已经在许多领域中取得应用，例如人脸的检测与识别、自动驾驶、智能安防监控和计算机辅助医疗诊断等。

目前，基于深度学习的目标检测主要分为两类：两阶段的目标检测算法和一阶段的目标检测算法。前者是先产生一系列的候选框作为样本，再通过卷积神经网络进行样本分类；后者则不用产生候选框，直接将目标边框定位的问题转化为回归问题处理。正是由于这两种方法的差异，其在性能上也有不同，前者在检测准确率和定位精度上占优，后者在速度上占优。当前，已有的目标检测算法已经能够取得较好的结果，它们大多都采用了anchor boxes，尤其是一阶段的目标检测算法。

使用anchor boxes有两个主要的缺点。一是，通常需要数量非常庞大的anchor boxes，大量的anchor boxes中其实只有少部分是和真实边界框相重叠的，正样本和负样本极其不平衡，影响网络的训练效果。二是，anchor boxes的使用会引入大量的超参数和额外的设计上的选择，例如anchor boxes的大小、比例和数量等都需要人为的设计，比较复杂。

Hei Law等人[1]舍弃了传统的采用anchor boxes的思路，提出了一种区别于主流目标检测方法的具有特色的新模型CornerNet，通过目标的左上角和右下角这一对顶点来预测目标的边界框，其大致流程如图1所示。该篇论文发表于ECCV 2018。

2. 目标检测技术的发展

2.1. 传统目标检测算法

传统的目标检测算法的主要流程是，首先使用滑动窗口对图像进行遍历，由于目标的尺寸和长宽比多种多样，故需要采用不同大小、不同比例的滑动窗口对图像进行截取。有了截取的窗口图像后，要对其进行预处理操作，包括尺寸变换、去均值、消除无关特征、减少噪音等操作。然后对图像使用由人工设计的特征算子进行特征提取，特征提取是传统目标检测方法中的关键，特征的质量在很大程度上决定了最终结果的好坏，常用的特征有HOG特征、SIFT特征和LBP特征等，不同的任务常常对特征有着不同的要求，也因此不存在一种适应所有情况的通用的特征。在经过特征提取之后，使用机器学习中常用的分类器，进行分类和回归，并最终生成检测结果。

但是，HOG、SIFT和LBP等手工设计的特征，在简单的场景中尚能满足需求，但无法适用于复杂的应用场景。而在具有大规模训练样本的条件下，基于深度学习的目标检测算法的识别准确度能够达到传统目标检测算法难以达到的高度。

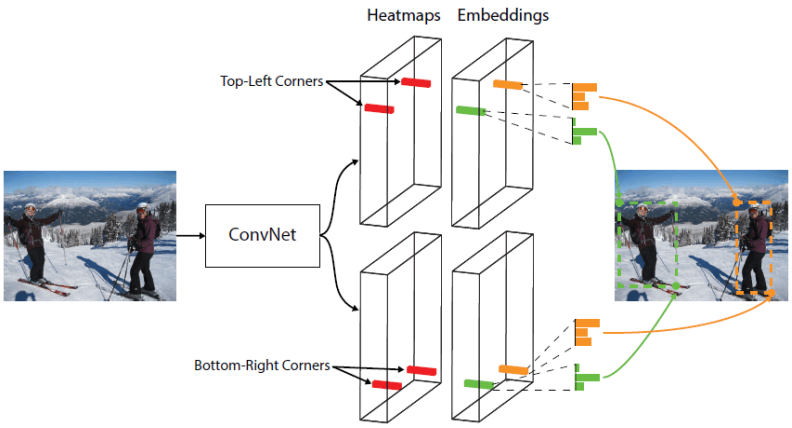

图 1. CornerNet的整体流程

2.2. 基于深度学习的目标检测算法

2.2.1 两阶段的目标检测算法

R-CNN[2]是卷积神经网络应用于目标检测问题的里程碑之作，R-CNN首先使用选择性搜索算法(Selective Search) 根据图像的低级特征产生可能包含目标的候选区域，接着将这些区域进行预处理，变换为统一的尺寸，然后分别送入一个已经训练好的卷积神经网络，生成每幅图的特征，利用这些特征进行分类，并经过回归器得到目标的边界框位置。相比较于传统方法，R-CNN的目标检测准确度有明显的提高。

随后，Kaiming He等人[3]对R-CNN进行了改进，提出了SPP-Net，引入了空间金字塔池化（Spatial Pyramid Pooling），使得R-CNN在不牺牲检测质量的情况下运行速度得到显著提升。在此基础上，Ross Girshick等人[4]结合R-CNN 与SPP-Net的主要思想，提出了Fast R-CNN。再之后，Shaoqing Ren等人[5]更进一步提出了Faster R-CNN，利用RPN网络通过一定规则设置不同尺度的anchor boees进行候选框选择，代替了选择性搜索等传统的候选框生成方法，进一步提高了目标检测的速度。

此后，Jifeng Dai等人[6]提出了一种消除了隐含全连接层的全卷积网络目标检测框架R-FCN，能够以更快的速度得到与Faster R-CNN准确度相媲美的结果。Kaiming He 等人[7]还提出了Mask R-CNN，实现了像素级别的目标实例分割。

2.2.2 一阶段的目标检测算法

以R-CNN算法为代表的两阶段的方法由于RPN结构的存在，虽然检测精度越来越高，但是其速度却遇到瓶颈，比较难于满足部分场景实时性的需求。因此出现一种基于回归方法的一阶段的目标检测算法，不同于两阶段的方法的分步训练共享检测结果，一阶段的方法能实现完整单次训练共享特征，且在保证一定准确率的前提下，速度得到极大提升。

具有代表性的一阶段目标检测方法有YOLO[8]和SSD[9]。YOLO的检测速度能达到每秒45帧，能够满足实时检测的要求。YOLO算法是基于图像的全局信息进行预测的端到端算法，整体结构简单，检测精度不高但检测速度快。SSD在YOLO的基础上借鉴了RPN的思路，在保证高精度检测的同时，兼顾检测速度。其后，YOLO经过不断的改进发展出了多个版本，包括YOLO V2[10]，YOLO V3[11]等，在检测准确度与速度上不断提升。

3. CornerNet

3.1. 方法概述

CornerNet把目标的边界框看作左上角和右下角这一对顶点，这样目标检测就变成了一对顶点的检测。卷积网络生成两个heatmap分别对应目标边界框左上角和右下角的顶点，同时对每个检测出的顶点生成一个embedding vector，使得同一个对象的embedding vector距离很小，可以用embedding vector来把属于同一类别的点组合起来构成目标的边界框。为了提升

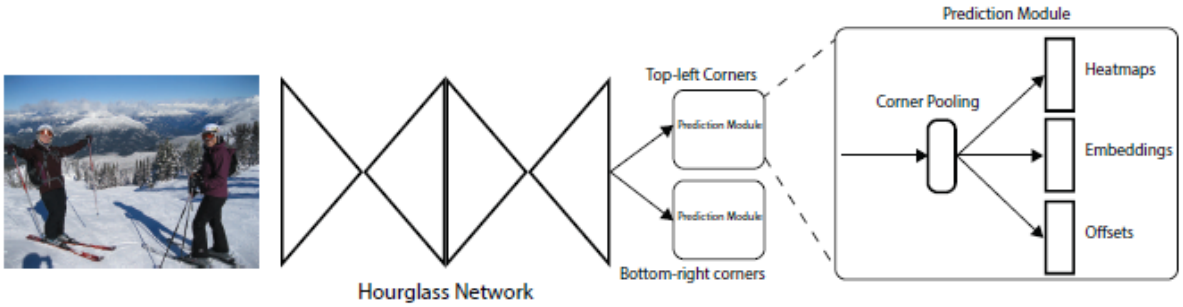

图 2. CornerNet的整体结构

边界框的预测精度，网络同时需要预测偏移量来对角的位置进行微调。在获得了heatmap, embedding vector和offset之后，运用一个简单的处理算法来获得目标最终的边界框。

CornerNet的整体结构如图2所示。作者使用沙漏网络[12]来作为CornerNet的主干网络。在沙漏网络之后连接两个预测模块，分别用于预测左上角的顶点和右下角的顶点。每个预测模块都有自己的corner pooling来对沙漏网络输出的特征进行池化，之后再分别预测heatmap, embedding vector和offset。与其他的目标检测算法不同的是，该方法不使用不同尺度的特征来检测不同大小的物体。

3.2. 顶点检测

网络分别预测左上角和右下角的两组heatmap，每组heatmap大小为$H \times W$，有C个通道，其中C是类别的数量（不包括背景），所以整个heatmap输出的大小为$H \times W \times C$，每个通道都使用二进制掩码来表示该类顶点的位置。

对于每一个顶点，只有一个该位置的正样本，其他的都是负样本，但是作者特意减少了在正样本周围r半径内的负样本的惩罚，原因在于这些点组成的边界框仍然可以足够地覆盖真实值。通过确保半径r内的每一对点产生的边界框与真实值的IoU都大于某个阈值（实验中为0.7）来确定物体的大小，从而确定半径r。对于给定的半径r，惩罚值的大小是以正样本顶点位置为中心的非标准化的二维高斯分布$e^{-\frac{x^2+y^2}{2\sigma^2}}$，其中$\sigma = \frac{1}{3}r$

设p_{cij}表示heatmap中预测为c类的位置的得分，y_{cij}为相应的用非标准化高斯增强的真实值在heatmap中的值。作者设计了一个focal损失函数[13]的变体来作为损失函数，如下所示：

$$L_{det} = \frac{-1}{N} \sum_{c=1}^{C} \sum_{i=1}^{H} \sum_{j=1}^{W} \begin{cases} (1-p_{cij})^{\alpha} \log(p_{cij}) & \text{if } y_{cij}=1 \\ (1-y_{cij})^{\beta} (p_{cij})^{\alpha} \log(1-p_{cij}) & \text{otherwise} \end{cases} \tag{1}$$

其中，N是图片中目标的数量，α, β是超参数用来控制每个点的分布，$(1-y_{cij})$这项减少了真实值周围点的惩罚。

由于采用降采样使得网络的输出相比原图像小了很多，图像中(x, y)位置的像素会被映射到heatmap中的$(\lfloor \frac{x}{n} \rfloor, \lfloor \frac{y}{n} \rfloor)$，其中$n$是降采样因子，这样就会发生错位。为了解决这个问题，作者通过预测位置偏移量来微调顶点的位置。设(x_k, y_k)为顶点的位置坐标，偏移offset为o_k：

$$o_k = \left(\frac{x_k}{n} - \left\lfloor \frac{x_k}{n} \right\rfloor, \frac{y_k}{n} - \left\lfloor \frac{y_k}{n} \right\rfloor \right) \tag{2}$$

模型预测两个offset集，分别由左上角顶点和右下角顶点共享，训练的时候在真实值的顶点位置使用平滑L1损失[4]：

$$L_{off} = \frac{1}{N} \sum_{k=1}^{N} \text{SmoothL1loss}\left(\boldsymbol{o}_k, \hat{\boldsymbol{o}}_k\right) \tag{3}$$

为了判断一对顶点（左上角和右下角）是不是来自同一个目标的边界框，网络为每个顶点预测一个embedding vector，使得属于同一个边界框的两个角的顶点的embedding vector距离会很接近，故可以通过embedding vector之间的距离来组合目标边界框的左上角顶点和右下角顶点。

图 3. 目标边界框左上角顶点和右下角顶点的预测结果示例

作者参考[14]中的方法，使用1维的embedding，设e_{tk}为目标左上角点的embedding，e_{bk}为目标右下角点的embedding，模型训练L_{pull}损失函数使同一目标的顶点进行分组，L_{push}损失函数用于分离不同目标的顶点。损失函数如下所示，其中e_k为e_{tk}和e_{bk}的平均值。

$$L_{pull} = \frac{1}{N}\sum_{k=1}^{N}\left[(e_{t_k} - e_k)^2 + (e_{b_k} - e_k)^2\right] \quad (4)$$

$$L_{\text{push}} = \frac{1}{N(N-1)}\sum_{k=1}^{N}\sum_{\substack{j=1\\ j\neq k}}^{N}\max(0, \Delta - |e_k - e_j|) \quad (5)$$

3.3. Corner Pooling

因为模型预测的是目标边界框左上角和右下角的两个顶点，但是通常这对顶点周围局部并没有目标的特征信息。考虑到目标边界框左上角顶点的在水平方向的右侧有目标顶端的特征信息，在顶点垂直方向上的下侧有目标左端的特征信息，因此如果左上角顶点经过池化操作后能有这两个信息，那么就有利于顶点的预测。

作者提出了一种针对边界框顶点预测的新的池化方式corner pooling，可以有效地从目标区域整体捕捉到顶点边界框顶点的位置信息。以左上角顶点为例，对每个feature map，在水平方向上从右到左做max pooling，在垂直方向上从下到上做max pooling，然后求和，其过程如图4 所示。

4. 实验

4.1. 实验方法

作者采用PyTorch实现CornerNet,使用MS COCO数据集[15]对CornerNet进行实验评估。MS COCO数据集

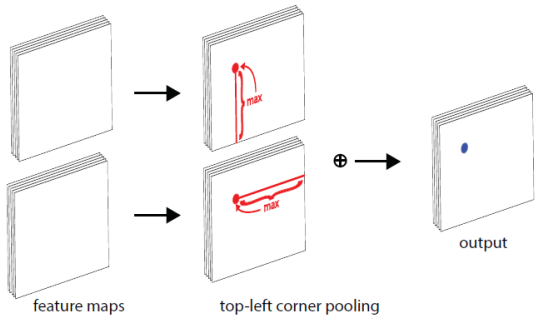

图 4. Corner pooling示例

包含80k 图像用于训练，40k图像用于验证，20k图像用于测试。作者使用训练集中的所有图像和验证集中的35k图像用于训练，将验证集中剩余的5k图像用于超参数搜索和模型简化测试。

4.2. 实验结果

经过实验，CornerNet在MS COCO数据集上取得了42.1%的AP，优于目前所有的一阶段目标检测方法，其检测结果可以与两阶段的目标检测方法相媲美。通过模型简化测试，作者证明了所提出的Corner pooling对于CornerNet的性能有着至关重要的作用。图3展示了两组模型对目标边界框左上角顶点和右下角顶点的预测结果。

5. 结语

本文在讨论了目标检测方法近年来的发展状况的基础上，介绍了一种与当前目标检测方法有较大不同的具有特色的新方法CornerNet，区别于主流的目标检测方法，它将目标边界框的检测变为成对的顶点的检测。也正是因为该方法的创新性，本文才选择该方法进行介绍。该方法在MS COCO数据集上的实验结果表明，其优于当前所有的一阶段目标检测算法。

参考文献

[1] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *european conference on computer vision*, pages 765–781, 2018.

[2] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361, 2014.

[4] Ross B. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[5] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *neural information processing systems*, pages 379–387, 2016.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[8] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *european conference on computer vision*, pages 21–37, 2016.

[10] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[12] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *european conference on computer vision*, pages 483–499, 2016.

[13] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[14] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *neural information processing systems*, pages 2277–2287, 2017.

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *european conference on computer vision*, pages 740–755, 2014.