2018云从人头技术-ZJUAI-PyramidBoxDetector

叶伟才 浙江大学 2019年2月13日

Wechat & phone: 13032893650

Email: [2755289083@qq.com](mailto:2755289083@qq.com) or [yeweicai@zju.edu.cn](mailto:yeweicai@zju.edu.cn)

2018 云从人头计数竞赛在Biendata上举办，截止时间是2018-09-15~2019-01-29。我们团队ZJUAI从11月下旬开始比赛，经过一个多月的鏖战（中间赶课程作业、项目、科研放水了IMG_256），在比赛最后两个星期，本人集齐实验室12块Titan GPU，全力打比赛（数据平衡、数据增强、网络选择、debug、网络训练测试trick等）。最后在初赛A榜截止时间1月18日前夺得冠军（0.60651）。更换数据集后，在测试集上取得冠军（0.75584），比第二名高约23个百分点，获得奖金人民币3万元。在额外的数据集上，取得第二名（0.477438），获得奖金人民币1500元。我们的模型在人头计数比赛上显示出较好的泛化性能。

以下我将介绍一下我们的团队ZJUAI，和我们的参赛方案（参赛历程、比赛代码、模型等）。

**一、团队介绍：ZJUAI**

ZJUAI 主要由浙江大学学生人工智能协会竞赛部的几个成员组成。队长：叶伟才（参赛名：maike， 邮箱：2755289083@qq.com），主要负责赛题调研、方案选择、数据平衡增强、网络训练测试trick、分工协作，指导队员打比赛、赛后总结等，涵盖比赛的整个过程，尤其是最后全力打比赛，带领团队夺得桂冠。由于队员先前没有打比赛的经验，只能本人全程carry（自己先前打过天池，现在打kaggle比赛），很庆幸最后能带领ZJUAI拿下冠军。

简单介绍自己：

叶伟才，男，浙大计算机学院18级硕士生研一，CAD&CG实验室，师从章国锋教授，研究方向：计算机视觉、SLAM等。 目前课题：文本到图像生成。

领英：<https://www.linkedin.com/in/weicai-ye-b9b36b129/>

Github: <https://github.com/ywcmaike>

**二、冠军方案介绍：基于PyramidBox的改进方案**

1. **任务介绍**

在拥挤的情况下，我们很自然地会根据身体的可见部分来计算人数，而可见部分主要是头部区域。这驱使我们去检测头部而不是整个人体为基础的人群计数问题。本次大赛邀请参赛者设计算法，主要针对复杂场景下人头的检测，返回每个人头的具体位置(x,y,w,h)，其中x表示人脸矩形框的横坐标，y表示人脸矩形框的纵坐标，w表示人脸矩形框的宽，h表示人脸矩形框的长。

比赛的目标就是检测出复杂场景的人头，并进行计算。其中复杂场景主要有：遮挡、光亮度变化、密集场景、不同场景区别较大等，另外比较大的问题是给定的数据集标注并不统一（标注人脸、标注头、标注包含肩膀），模型很难训练。

1. **评价指标**

比赛使用检测任务中常见的衡量指标mAP，这里就不赘述。

1. **赛题调研和模型选择**

本次的人头计数比赛跟先前的人脸检测比较相近，人脸检测主要是算法检测出人脸的区域，而人头计数也差不多，我们发现数据集中部分的人头其实也就是人脸的标注。有个小插曲：之前参加阿里天池瑕疵检测识别比赛，调研过当前较佳的检测算法，其中也包括人脸检测方面的算法，且之前在商汤实习，了解过mmlab的工作widerface数据集，且该比赛有leaderboard[1]。所以这个比赛就比较直接了，直接根据我先前的调研结果，ssh[2]和pyramidbox[3] 是比较好的检测网络，且能找到开源的代码作为参考。 我们试验了ssh和pyramidbox 在同样的数据和训练方式下，pyramidbox的mAP要明显高于ssh，主要在数据增强上要比ssh做的多，而且相对来说引入了一些context先验信息。想要了解更多的详情请看论文。由于只是工业界的小比赛，为了省时间，较快，较好的取得成果，我后续就只在pyramidbox进行试验。这里还有个小插曲，我们刚开始使用的百度开源的pyramidbox（官方版paddlepaddle），试验过程中发现太鸡肋了，并行训练不起来，还贼慢，魔改了一波代码，发现费力不讨好，虽然曾取得第二（0.5085）的成绩。还是pytorch比较方便，修改起来比较直接。最后果断弃坑，转向pytorch版本的pyramidbox[4]。虽然训练过程也是出现了一大堆问题，但自己比较熟悉，魔改了一天，终于跑通了IMG_256。 就差自己重新写一遍完整的网络框架了。

1. **数据分析处理**

我们观察了数据集，主要有5个场景组成：our/mall/ucsd/para/partb。我们也对数据集进行统计分析，发现不同场景的图片数量各异（5302/1500/2000/300/400）、每张图片的bbox数量各异、bbox的尺寸也不尽相同，标注方式也不一样（含肩膀/小头，不含肩膀/含肩膀，几乎有半身/人脸/有点随意，不含肩膀）,具体可看附件我们的训练测试数据统计。Our数据集是官方自己的数据。由于数据图像数量差异较大，我们主要对数据量少的数据进行增强到跟our的数据集图片数目基本一致（主要有加噪点，模糊，修改色度、光亮、旋转平移等，使用图像增强库[5],更多的细节可以参考我们的data\_gen.py），来进行平衡数据，使得网络训练的模型更具有通用性，我训练出来的网络best\_mod\_Res50\_pyramid.pth 即为通用的模型，在a榜数据集上能够达到0.59~0.60的mAP，算是单模型的最高精度。

1. **模型框架**

模型框架基本还是采取百度pyramidbox的论文框架，框架上没有进行太大的改动。原模型的backbone使用vgg16，修改为res50，能一定程度上提高mAP。模型采用类似图像金字塔的方式来融合更多层次的特征，使用半监督的方法来学习较小的、模糊的和部分遮挡的人脸的语境特征，如肩膀占比多大之类的，也使用新的采样来重点关注较小的人脸，这对于单图多人头的检测的提升起到一定的帮助。更多的信息请参考pyramidbox原文。

1. **模型训练trick**
2. 我们将数据增强平衡过的训练数据集，分为train、val数据集，这里得着重提一下，val选取应尽量跟测试集的分布保持一致，我们按照约9:1的比例，进行顺序划分数据集，使用我们的模型进行validate，计算mAP的结果跟线上的结果基本一致，这有利于我们选择模型结果提交。
3. 训练过程中我采用预训练模型来加速我们的训练，我们使用4块titan GPU进行训练，batch\_size=32， lr设置为1e-3, 采用warm up的方式，前1600个iteration按照每400个iter进行线性增加，直到lr=1e-3，接着设置8000， 12000， 18000进行lr decay， 使用sgd进行优化，momentum=0.99。这是通用的模型的基本设置方案。
4. 通过测试训练的单模型能达到0.59~0.60的mAP，我实验了很多的参数都基本无法进一步提升。
5. 接着我采取并行的按照数据集的类别进行多个模型的训练，我们按照batch\_size=16，每两块titan训练一个模型，总共训练了5个单一的模型（还好手头有12块titan，并行训练一天出现结果），按照数据集类别分别进行测试，模型能得到1个多点的提升。
6. **模型测试trick**
7. 使用多尺度测试、图像金字塔测试、翻转测试和原图测试能提高测试的精度，提高mAP。
8. 放大图像为原来的2倍，也能提高mAP，原因是能够检测到较小的人头。
9. 检测的BBox的个数提升，能一定程度的提高mAP，对于partA、partB的图像的bbox可能较多，设置为BBox=1000。 再增大也没有提升。
10. 在测试过程中发现可以使用多模型求得结果，再进行nms，也能一定程度上提升mAP。在数据集A榜中，能达到0.606512675015的mAP。 换数据集后，发现数据集跟原有的our数据集的bbox的数目比较接近，我们将训练好的通用模型、和our模型在原有的our数据集上进行测试，能分别达到0.60和0.80的结果，将两个模型的结果nms后能达到0.86的结果。感觉也只能这样了，毕竟换后的数据集跟原有的数据集差距太大了。很庆幸，我们的模型策略在新来的数据集上表现出较好的泛化能力。
11. **模型方案使用方式**

运行环境：Ubuntu16.04 pytorch0.3.1 torchvision numpy cuda cudnn imgaug等

1）将数据置于yuncong\_test\_set文件夹中，

2）直接运行test\_yuncong\_\*.py 可以得到对应的生成文件，更多的参数设置请看文件。其中test\_yuncong\_mod\_test2.py 生成通用模型的结果，test\_yuncong\_our\_test2.py 生成our模型的结果。 test\_yuncong\_nms\_test2.py生成nms后的结果。

1. **Reference**
2. <http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/WiderFace_Results.html>
3. <https://github.com/mahyarnajibi/SSH>
4. <https://github.com/PaddlePaddle/models/tree/develop/fluid/PaddleCV/face_detection> 百度paddlepaddle源码
5. <https://github.com/Goingqs/PyramidBox> 民间开源pytorch版本，直接跑不起来，需要使用我修改的代码及模型
6. **<https://github.com/aleju/imgaug>**
7. **Q&A**

由于本人现在家里，还没到实验室，而且实验室寒假断网断电，连不上服务器，所以比赛过程中的数据增强、平衡，网络训练代码，暂时无法提供。只能提供之前比赛提交模型代码要求的相关文件，即给定训练好的模型和测试代码。如有需要，欢迎致电，会过段时间提交。

联系方式：

Wechat & phone: 13032893650

Email: [2755289083@qq.com](mailto:2755289083@qq.com) or [yeweicai@zju.edu.cn](mailto:yeweicai@zju.edu.cn)