

问题分析

问题描述

在 3024 张已经标记好的图像上进行训练，可以额外增加数据进行训练，最终在测试集上进行测试。

实验过程

编程环境

本实验所选择的编程环境为 Windows 10, Python 3.5.x, Keras 2.1.x, Tensorflow 1.6.0

训练预处理

需要将数据集转换成 `image_file_path box1 box2 ... boxN` 的格式，其中每一个框的格式为 `x_min,y_min,x_max,y_max,class_id`。

我们的识别为 5 类，根据论文 [1] 中给定的 $filters = (classes + 5) * 3$ 计算公式，得到 `filters` 为 30。

结果测试

选取了清华大学的公开数据集¹进行了测试，对于测试图像图 1(a)，得到的结果如图 1(b) 所示。

可以看到在图 1(a) 中至少有 7 个不同位置的交通标志，但在实际检测中只有 3 个交通标志被检测出，推测最有可能的原因是图像过大，导致检测效果不好，故按照表 1 所示对图像进行了裁剪。

在分割过程中也遇到了人工切割可以识别，但是通过代码切割未能识别的情况，查找原因是在绘制识别标记的时候将原图进行了篡改，导致交通标志被遮挡等情况。修改代码后，分割后的测试结果如图 2 所示。

¹<https://cg.cs.tsinghua.edu.cn/traffic-sign/>



(a) 测试图像

(b) 测试结果

图 1: 测试图像及其结果

表 1: 图像裁剪划分

图像裁剪 box	说明
(0, 0, 1, 1)	原始图像
(0, 0, 1/2, 1/2)	左上
(1/2, 0, 1, 1/2)	右上
(0, 1/2, 1/2, 1)	左下
(1/2, 1/2, 1, 1)	右下
(1/4, 1/4, 3/4, 3/4)	中心

可以看到与未分割的图像相比，分割后的图像更有利于识别，提高检测精度。但对于图2来说，仍然存在着两个问题：第一个问题是对于倾斜角度较大的交通标志（左上）未能识别，第二个问题是多次切割后对于重复区域的检测存在重复，在评判标准中会将其算为误检，故应想办法进行“去重”操作。对于第一个问题暂时未能想到好的解决办法，更需要依赖于对数据集方面的加强。第二个问题相对而言比较好解决，通过检测评价函数 **intersection-over-union(IOU)**[2] 即可对重复的检测区域进行判断，而这个函数也是对于物品检测准确度的一种考量方式。最终的识别结果如图3所示。

通过对重复标注的数据进行 IoU 检测并通过置信度进行筛选，成功地在提升了检测率



图 2: 图1分割后测试结果



图 3: 图2 “去重” 后结果

的同时降低了误检率，对于图1(a)来说，成功地将 13 个重复的检测结果精确为 7 个检测结果。

总结分析

首先要感谢陈嘉俊同学²和邓兆利同学³花费时间对数据集进行了重新的核对，对未标注的交通标志或者是错误标注的数据进行了重标和补标。因为对于一些曾经不在 77 类但却在 5 大类的的数据，在最开始标注的时候可能会存在遗漏，这对于本来训练数据就稀缺的情况来说，可能会对模型的最终效果造成影响。

对于交通标志检测与识别这个问题来说，仍然有一些问题可以继续深入地研究下去。

在解决输入图像过大的问题上，我之前所采用的方法是对原始图像进行分割，而分割的方法也较为粗略，只是将图像分为四块并为了弥补中间可能出现的将交通标志分割成两块而额外增加的一块中间位置的分割。我认为可以改进的方法是对图像先进行一次预检测，之后按照预检测的结果对图像进行分割，从而极大地保证不会出现将交通标志分割到两张子图的情况。

还有一类问题如图4所示，一些停车场等场景下的一些指示灯和一些角度下的车灯与交通灯标志相同，可能会产生错误检测的情况，这类问题的解决方法之一是：在对数据进行标注的时候，刻意地对交通标志周围的固定环境也一起标注进去（比如交通灯一般都是三个灯相连的），不过还需要再一次对检测结果进行一次细检测，对于模型的复杂性和时间上都有一定的影响。

²邮箱: 657739830@qq.com

³邮箱: 304628356@qq.com



(a) 混淆图像 1



(b) 混淆图像 2

图 4: 混淆图像

参考文献

- [1] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [2] Intersection over union (iou) for object detection. <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. 访问日期: 2019-01-18.