

教程 | 如何使用TensorFlow API构建视频物体识别系统

2017-07-16 机器之心

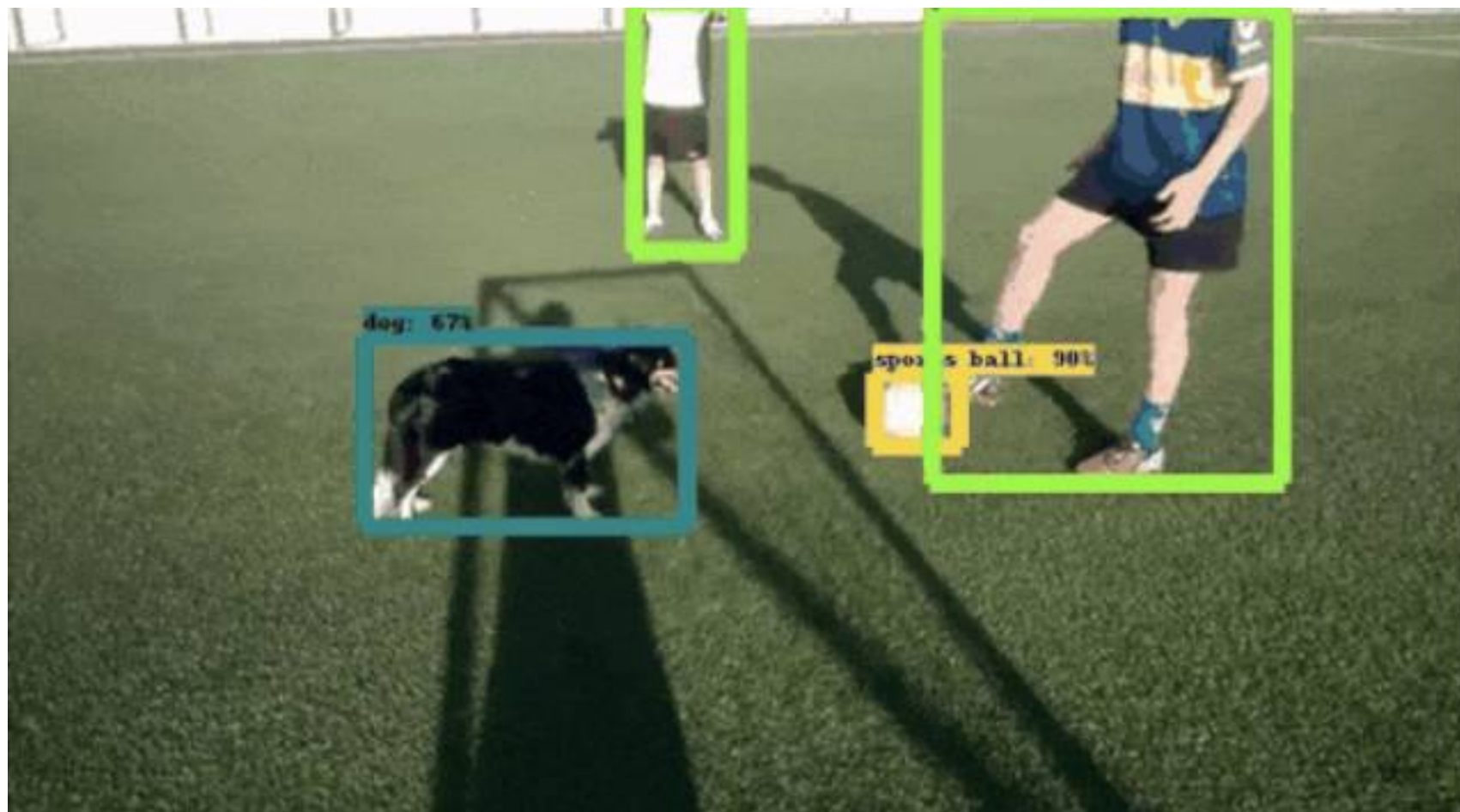
选自Medium

机器之心编译

参与：李泽南

在谷歌 TensorFlow API 推出后，构建属于自己的图像识别系统似乎变成了一件轻松的任务。本文作者利用谷歌开源的 API 中 MobileNet 的组件很快开发出了识别图像和视频内物体的机器学习系统，让我们看看她是怎么做到的。

市面上已有很多种不同的方法来进行图像识别，谷歌最近开源的 TensorFlow Object Detection API 是其中非常引人注目的一个，任何来自谷歌的产品都是功能强大的。所以，让我们来看看它能够做到什么吧，先看结果：



来自 TensorFlow API 的视频物体检测

你可以在 GitHub 上找到这个小项目的全部代码：<https://github.com/priya-dwivedi/Deep->

训练的过程有多复杂？首先让我们来看看 API 本身。

TensorFlow Object Detection API 的代码库是一个建立在 TensorFlow 之上的开源框架，旨在为人们构建、训练和部署目标检测模型提供帮助。

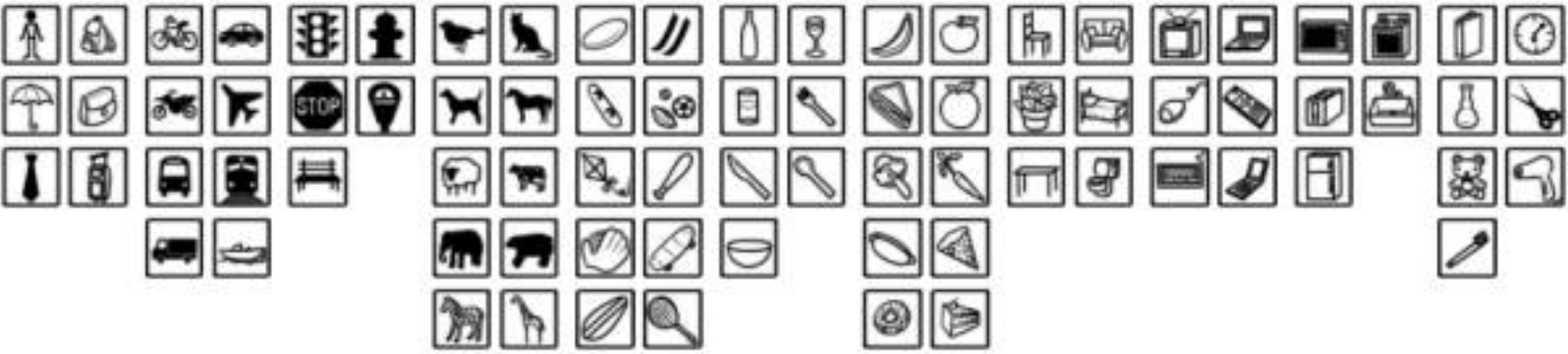
该 API 的第一个版本包含：

- 一个可训练性检测模型的集合，包括：
- 带有 MobileNets 的 SSD (Single Shot Multibox Detector)
- 带有 Inception V2 的 SSD
- 带有 Resnet 101 的 R-FCN (Region-Based Fully Convolutional Networks)
- 带有 Resnet 101 的 Faster RCNN
- 带有 Inception Resnet v2 的 Faster RCNN
- 上述每一个模型的冻结权重（在 COCO 数据集上训练）可被用于开箱即用推理。
- 一个 Jupyter notebook 可通过我们的模型之一执行开箱即用的推理
- 借助谷歌云实现便捷的本地训练脚本以及分布式训练和评估管道

SSD 模型使用了轻量化的 MobileNet，这意味着它们可以轻而易举地在移动设备中实时使用。在赢得 2016 年 COCO 挑战的研究中，谷歌使用了 Fast RCNN 模型，它需要更多计算资源，但结果更为准确。

如需了解更多细节，请参阅谷歌发表在 CVPR 2017 上的论文：<https://arxiv.org/abs/1611.10012>。

在 TensorFlow API 的 GitHub 中，已经有经过 COCO 数据集训练过的可用模型了。COCO 数据集包含 30 万张图片，90 中常见事物类别。其中的类别包括：



COCO 数据集的部分类别

- TensorFlow Object Detection API 的 GitHub : https://github.com/tensorflow/models/tree/master/object_detection
- COCO 数据集: <http://mscoco.org/>

如上所述，在 API 中，谷歌提供了 5 种不同的模型，从耗费计算性能最少的 MobileNet 到准确性最高的带有 Inception Resnet v2 的 Faster RCNN：

Model name	Speed	COCO mAP	Outputs
ssd_mobilenet_v1_coco	fast	21	Boxes
ssd_inception_v2_coco	fast	24	Boxes
rfcn_resnet101_coco	medium	30	Boxes
faster_rcnn_resnet101_coco	medium	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	slow	37	Boxes

在这里 mAP（平均准确率）是精度和检测边界盒的乘积，它是测量网络对目标物体敏感度的一种优秀标准。mAP 值越高就说明神经网络的识别精确度越高，但代价是速度变慢。

想要了解这些模型更多的信息，请访问：
https://github.com/tensorflow/models/blob/477ed41e7e4e8a8443bc633846eb01e2182dc68a/object_detection/g3doc/detection_model_zoo.md

使用 API

首先，我尝试使用了其中最轻量级的模型（ssd_mobilenet）。主要步骤如下：

- 下载封装好的模型（.pb - protobuf），将其载入内存，链接：<https://developers.google.com/protocol-buffers/>
- 使用内建帮助代码来载入标签、分类、可视化工具等内容
- 打开一个新的会话并在一个图像上运行模型

总体而言，这个过程非常简单。API 文件还提供了一个 Jupyter 笔记本来帮助记录主要步骤：
https://github.com/tensorflow/models/blob/master/object_detection/object_detection_tutorial.ipynb

这个模型在示例图片中的表现非常不错（如下图）：



在视频中运行

随后我开始尝试让这个 API 来识别视频中的事物。为了这样做，我使用了 Python 中的 moviepy 库（链接：

<http://zulko.github.io/moviepy/>)。主要步骤如下：

- 使用 VideoFileClip 函数从视频中抓取图片。
- fl_image 函数非常好用，可以用来将原图片替换为修改后的图片，我把它用于传递物体识别的每张抓取图片。
- 最后，所有修改的剪辑图像被组合成为一个新的视频。

这段代码需要一段时间来运行，3 到 4 秒的剪辑需要约 1 分钟的处理，但鉴于我们使用的是预制模型内固定的加载内存空间，所有这些都可以在一台普通电脑上完成，甚至无需 GPU 的帮助。这太棒了！只需要几行代码，你就可以检测并框住视频中多种不同的事物了，而且准确率很高。

当然，它还有一些可以提高的空间，如下图所示，它几乎没有识别出鸭子的存在。



原文链接：<https://medium.com/towards-data-science/is-google-tensorflow-object-detection-api-the-easiest-way-to-implement-image-recognition-a8bd1f500ea0>

本文为机器之心编译，转载请联系本公众号获得授权。



加入机器之心（全职记者/实习生）：hr@jiqizhixin.com

投稿或寻求报道：editor@jiqizhixin.com

广告&商务合作：bd@jiqizhixin.com

[点击阅读原文](#)，查看机器之心官网↓↓↓

[阅读原文](#)