使用卷积神经网络进行目标检测

1. fine-tune mobilenet-ssd

Learning rate=0.004，训练过程中损失变化如图1所示。最终loss在3左右。

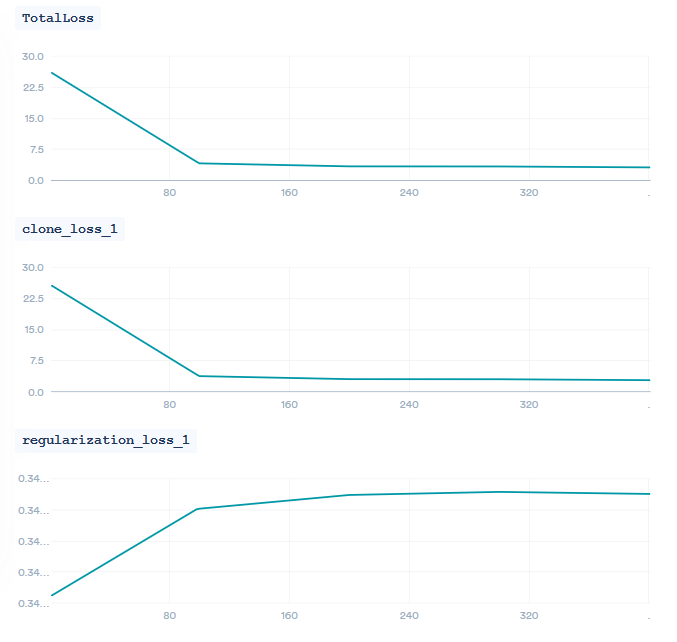
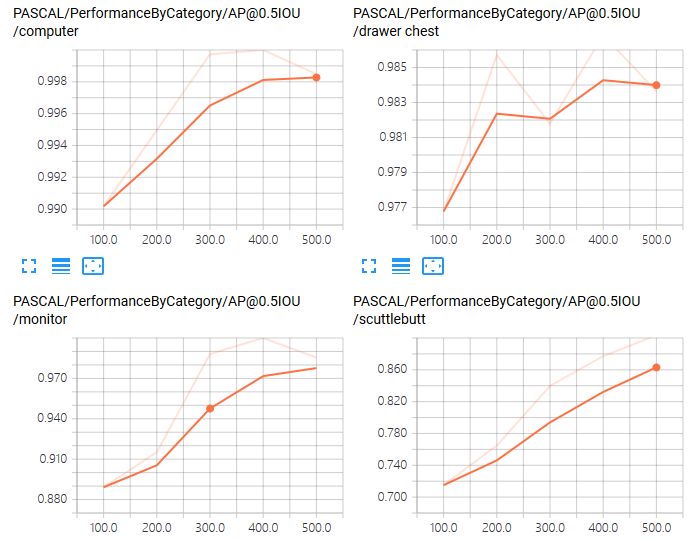
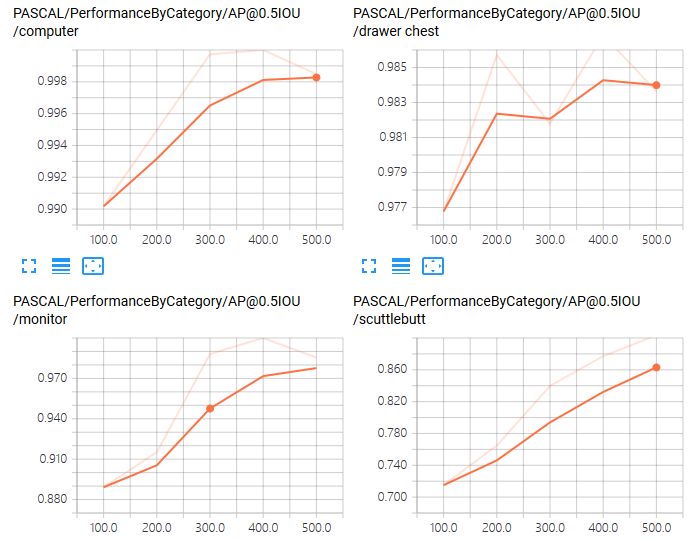


图1

验证集上AP如图2所示。经过5个epoch部分物体，如电脑的检测精度达0.998，而饮水机只有0.86。最终的mAP=0.95





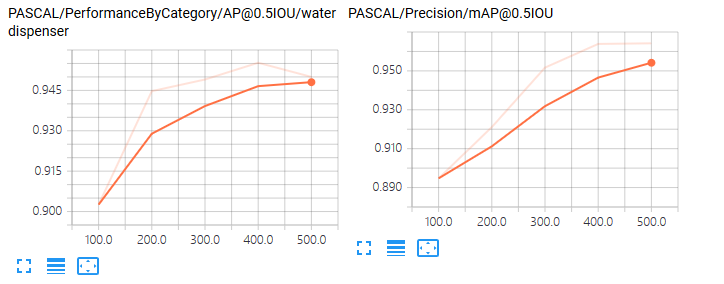


图2

最终图片的测试结果如图3所示。

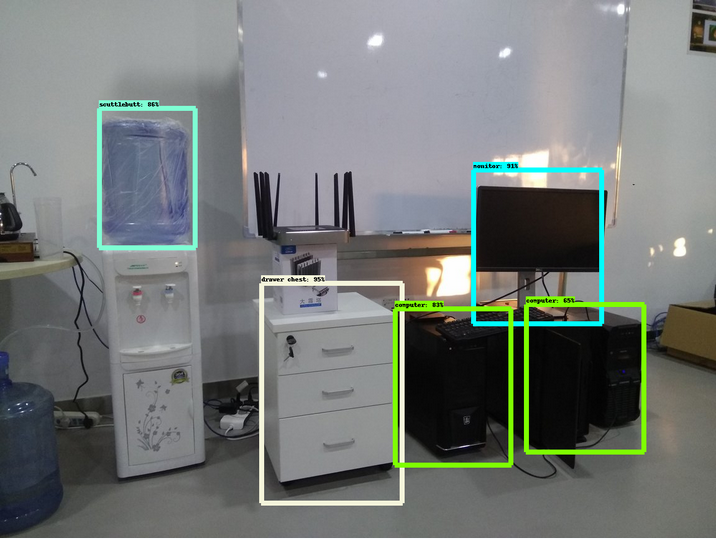


图3

2.yolo实现

通过对object\_detection框架下mobilenet-ssd的相关代码进行修改，粗略实现了yolo。主要修改了特征抽提部分网络的结构，如图4所示。除此之外在pipeline config文件中将anchor\_generator改为grid\_anchor\_generator，width\_stride和height\_stride均为16，scales=[0.25,0.5,1.0,2.0]，aspect\_ratio=[0.5,1.0,2.0]。

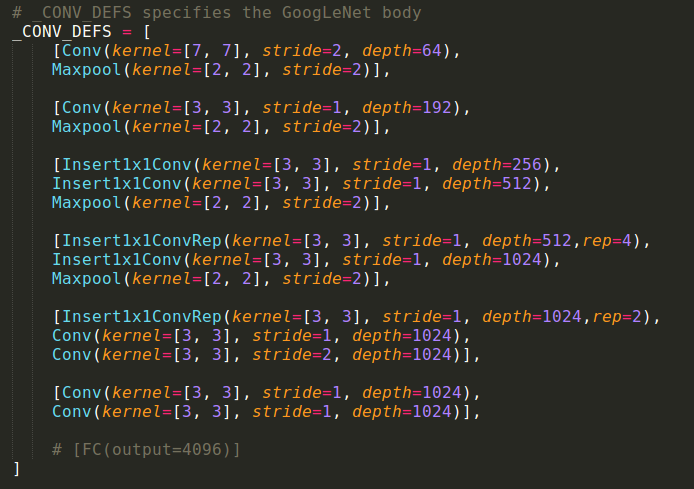
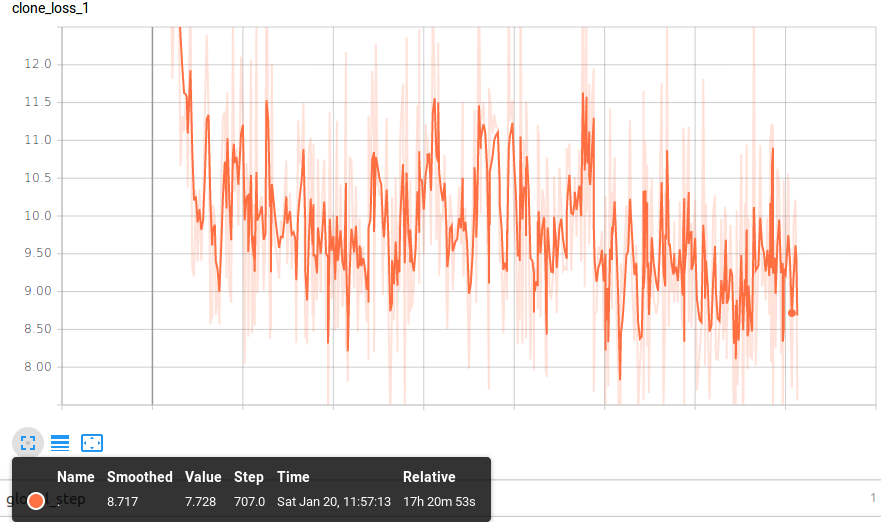


图4

训练损失如图5所示。





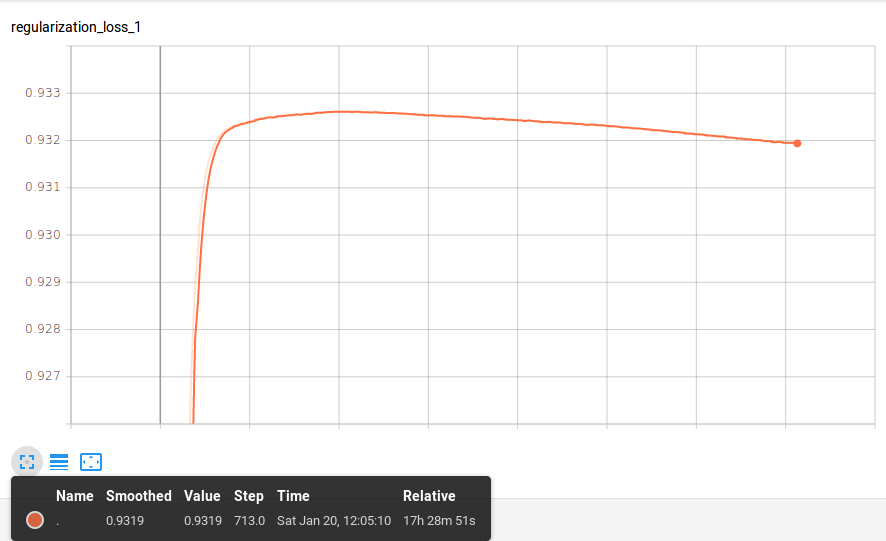


图5

在第717个step处进行eval，只有少数几类的mAP在0.01~0.03。

3.目标检测综述

R-CNN 在2013年由Ross Girshick提出，在ILSVRC2013上mAP=31.4%， VOC 2012上mAP=53.3%。同时期Overfeat在ILSVRC2013上mAP=24.3%，使用滑动窗口法+CNN来完成目标检测；UVA在VOC 2012上mAP=35.1%，使用selective search+SIFT+SVM来完成目标检测。算法首先通过selective search划出候选区域，然后缩放到227\*227，使用5个卷积层，2个全连接层的CNN提取4096维特征，最后使用SVM进行分类。通过使用非最大抑制，即如果有个区域跟它的IoU大于某一阈值，但是通过SVM给的分更高，丢掉它；以及外接矩形回归，即在pool5后接全连接层拟合真正的（x,y,w,h），给出最终结果。Selective search首先使用Graph-Based Image Segmentation，将图片划分为许多颜色变化剧烈、缓慢和基本不变的区域，然后将这些区域依照相似度组织成一棵树。相似度由颜色相似度、纹理相似度、大小相似度和位置相似度加权得到。最后找定位密集的区域进行目标检测，因为依照颜色认为这里有东西，依照纹理认为这里也有东西，那么这里很有可能真的有东西。

Fast R-CNN在2015年4月由Ross Girshick提出，在VOC2012上mAP=66%(vs. 62% for R-CNN)。通过在同一张图片的不同region proposal之间共享特征信息，将速度（不包括region proposal的时间）提升到0.32s/ image (vs.47s/image for R-CNN)。相较于R-CNN，主要提出了RoI pooling layer。首先让图片经过CNN抽提特征，然后将region proposal划定的区域映射到特征图上h\*w的区域，将h\*w的区域划分为7\*7的网格，在每一个网格内做max pooling，即为RoI pooling。RoI pooling为每个region proposal提取了固定大小的特征。之后通过多个全连接层，最后softmax得到的分类结果，回归得到每一个类对应的bounding box。

Faster R-CNN在2015年6月由任少卿等提出，在VOC2007上mAP=59.9%(vs. 58.7% for Fast R-CNN)。使用Region Proposal Network (RPN)来代替selective search，然后使用fast R-CNN来检测。通过使RPN和fast R-CNN共享卷积层，提出了一个针对目标检测的unified network，将region proposal的时间从 2s(CPU实现)缩减到10ms，将proposal的数量从2000降到了300。包括全部步骤，处理速度提升到5fps。为了使RPN能够高效地预测不同大小和长宽比的region proposal，提出了“anchor boxes”的概念。在共享卷积层的最后一层，对feature map上每一个3\*3的区域，预测中心在滑动窗口的中心，scale=[0.5, 1.0, 2.0]，aspect\_ratio=[0.5, 1.0, 2.0]，共k=9个“anchor boxes”上，region proposal的类别（2\*9=18维）及位置（4\*9=36维）。在一张W\*H (typically ~2400)的特征图上, 有WHk 个anchors。

R-FCN(Region-based Fully Convolutional Networks)在2016年由代季峰等提出，在VOC2007上mAP=83.6%，处理速度170ms per image。使用全卷积结构代替了R-CNN中的全连接，提高了参数共享的程度。将ground true分成了3\*3的方格，region proposal也是得到3\*3的方格，然后对格子进行逐个比对。

YOLO在2015年由Joseph Redmon提出，在VOC2007上mAP=63.4%，在VOC 2012上mAP=57.9%，处理速度约45 FPS。当时Faster R-CNN的精度虽然达到了70%以上，但是处理时间约为0.5FPS，并且将背景误认为是物体的情况是YOLO的2倍以上。YOLO在网络结构上将448\*448\*3的输入图片通过卷积和池化映射为7\*7\*30的输出。其中7\*7指将图片划成7\*7的网格，30指的是每一个格子对应了2个由（x,y,w,h,confidence）表达的外接矩形和对应20个待检测类别的onehot向量。x和y表示矩形框的中心落在网格中的位置，除以网格长宽归一化到（0，1）；w和h指外接矩形的宽和高，除以图片大小归一化到（0,1）；Confidence被定义为预测的外接矩形与真实的外接矩形的IOU(intersection over union)。由于一个网格只能预测2个矩形框和1个类别，因此YOLO对小物体，尤其是一堆小物体的检测效果很差。

SSD在2016年由Wei Liu提出，59 FPS with mAP 74.3% on VOC2007 test, 而Faster R-CNN 7 FPS with mAP 73.2% or YOLO 45 FPS with mAP 63.4%。SSD将300\*300的图片通过卷积和池化映射到38\*38\*（4\*（class+4））、19\*19\*（6\*（class+4））、10\*10\*……等一系列特征图，其中38\*38是网格大小，4是默认定义在这个方格上不同大小和长宽比的矩形框，class+4分别代表类别的onehot向量和（x,y,w,h）。相较于YOLO，SSD旨在处理大小不同的物体，并且YOLO是通过两个全连接层得到（x,y,w,h，confidence），而SSD是通过一个卷积层。