

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



课程：数字图像处理

学生姓名： 汪旭鸿

学生学号： 017032910027

专 业： 控制科学与工程

年 级： 直博一年级

基于 SSD 神经网络的交通标志牌检测

摘要:

目标: 交通标志牌的目标检测是一种在场景图像中指出交通标志牌目标的研究问题。近些年, 在道路场景监控系统, 无人驾驶系统中有着广泛的应用。所以, 优化交通标志牌目标检测问题有着重要的意义。为了建立实时检测上海市区的交通标志牌的模型, 我使用了课程提供的数据集, 利用 SSD 目标检测神经网络进行训练。

方法: 为了训练 SSD 网络, ①我进行了数据集分割和预处理, 使用 opencv 库进行了图像增强和增广, 以便增加模型的泛化能力和鲁棒性。②不同于 ECCV2016 论文中提出的 SSD300 原型网络, 为了降低模型复杂度, 我使用了简化的 7 层 SSD 神经网络, 它可以在一般的显卡上跑出可以接受的效果。③我使用了分类损失和定位损失的双重结合来对模型进行评估, 不但让模型定位准确也可以让模型的识别更准确。④我使用了动态的、有停止条件的 Adam 优化算法训练模型, 是模型参数优化地更好。

结果: 在 GTX1060 上经过大概 12 个小时左右的训练, 在测试集和数据集上都达到了 5%左右的误差率。除了一些光线条件太差或者标志牌太小的数据样本, 本模型在绝大多数图片上都取得了较好的结果, 甚至还可以在该数据集中将一些人眼漏标的交通标志检测出来。

结论: 我使用的 7 层 SSD 目标检测神经网络确实可以在交通标志检测的任务中取得较好的效果, 并且该模型对电脑配置要求低, 训练简单, 预测实时性好。我相信做出一些更改后可以应用在嵌入式处理器上, 但该算法也有一些改进的空间, 比如我发现该算法对小目标的检测明显较差, 这是一个仍需思考、解决的问题。

关键词: 实时目标检测 卷积神经网络 交通标志 数据增广

目录

0 引言.....	3
1 相关工作.....	4
2 交通标志牌检测 SSD 过程	6
2.1 任务分析及数据准备.....	7
2.2 图像数据增强.....	8
2.3 SSD 神经网络原理	9
2.4 7 层 SSD 网络的实现	12
2.5 模型评价指标.....	14
3 模型训练和结果分析.....	15
3.1 优化方法.....	15
3.2 训练环境.....	15
3.3 训练过程分析.....	16
3.4 结果分析与可视化.....	17
参考文献.....	22
代码说明.....	23

0 引言

交通标志牌的目标检测是一种在场景图像中指出交通标志牌目标的研究问题。它是人工智能应用领域中一个很重要的组成模块。近些年，在道路场景监控系统，无人驾驶系统中有着广泛的应用。所以，优化交通标志牌目标检测问题有着重要的意义。最近，有很多关于在静态场景中进行交通标志牌目标检测的研究工作。他们用不同的方式试图更好地解决问题。通常，会提取手工特征[1]，组合特征，分类。或者将交通标志牌拆分成局部再进一步提取特征，以便得到更精确的检测结果。但是，手工设计特征进行提取，依赖于研究人员的经验，缺乏对问题的泛化能力，存储这些手工特征也需要一定的存储空间，再者，对分类器的选择也十分影响最后的检测效果。当问题变换检测目标，或者延伸到复杂场景中，传统方式将面临更加严峻的挑战。为了达到更好的检测效果，整个系统框架将变得更加复杂。

任何的场景目标发现问题，都可以看做是一个和该目标相关的视觉任务。视觉任务定义为：在某一场景中对一类或几类目标进行目标检测的任务描述。视觉任务是由和检测目标强相关的场景样本图像构成。当希望检测某一场景中的某种类型目标时，可以通过收集一系列该场景下的目标图像样本，组成视觉任务。

本文视觉任务是：检测在城市道路背景下的交通标志牌。通过对视觉任务学习得到一种稳定的，与任务强相关的目标检测模型。将新的样本图像输入目标检测模型便可以得到检测结果。我使用了一种不依赖手工特征，在静态场景中进行交通标志牌检测的方法，我将利用深度学习思想解决视觉任务的训练过程。通过卷积神经网络对样本进行计算得到卷积特征，利用 SSD 网络解决目标检测定位的问题。对交通标志牌检测的问题提供了更泛化和简洁的解决思路。

1 相关工作

在传统模型中，基于手工设计特征[2]的检测模型被广泛应用于交通标志检测问题。Lopez 等人[3]利用新型的颜色变换模型对颜色特征计算，快速找到重要的交通标志拍颜色，定位可能的候选区域。Maldonado 等人[4]在做视频流中的交通标志牌检测时，将临时的底层几何特征的时间信息与标志牌的动态信息结合，并利用支持向量机模型从背景中区分出目标车辆。

可以看出，在传统检测模型中，提取图像的底层特征是至关重要的一步。

传统模型有以下的缺点：

- 1) 需要针对不通的检测问题进行特征设计；
- 2) 需要对特征进行组合，选定分类器进行训练，其过程较为复杂；
- 3) 需要存储空间对特征进行存储。

近些年，深度卷积神经网络被成功地应用于物体识别与目标检测，其中以 AlexNet 为代表[5]。2012 年，Krizhevsky 等人在 ImageNet 大规模视觉识别的挑战赛中展现了 CNN 显著提高的图像分类精确度的能力。受到 AlexNet 在图像处理领域的显著效果的启发，Ross Girshick[6]等人提出一种深度学习模型，命名为 R-CNN，将 CNN 运用于目标检测问题，该模型首先使用选择性搜索算法来计算图像的候选区域，然后将所有的候选区域输入到 R-CNN 模型中提取特征，最后在 SVM 中完成分类。该模型设计了包围盒回归算法回归计算候选区域坐标，在 PASCAL VOC 的目标检测集上进行测试，平均正确率比非神经网络的算法高出大约 20%。该模型提出了一种模型预处理的思路，首先在 ImageNet 小数据集上初始化网络的权重，然后在 PASCAL VOC 数据集上微调网络，训练效果会更好。虽然 R-CNN 的正确率有了很大提升，但也存在着缺点，即由于每张图像有大约 2000 个候选区域，导致计算量很大。

微软亚洲研究院工作者将 SPPnet[7]的思想运用与目标检测中，它首先做了一种映射，计算出候选区域映射到最高卷积层的特征图上的位置，然后用基于 SPP 算法的池化层进行降维操作，最终会获取一个特定大小的特征层，其过程和

R-CNN 一样，在 PASCAL VOC 测试，准确率和 R-CNN 相近，但运行时间大大减少。

2015 年，Ross Girshick 进一步将 SPPnet 的思想和 R-CNN 结合提出卷积神经网络模型 Faster R-CNN[8]，再用 softmax[9]回归替代 SVM 分类器降低空间和时间的开销。整个训练过程不需要分级进行，检测过程更加高效，准确度更高。经过在 GPU 上的训练与测试，实验结果表明提取候选区域的时间明显缩短，检测的时间也缩短为 1/10，分类精度也有所增加。

2016 年，Liu Wei[10]等人将 YOLO 网络[11]的结构与 Girshick 的 Faster R-CNN[12]相结合，提出了 SSD(Single shot multibox detector)目标检测算法，SSD 网络它的速度比 Faster R-CNN 要快很多，但其工作方式却和 Faster R-CNN 存在显著不同。Faster R-CNN 采用区域推测来生成候选区域，然后在各个候选区域内使用分类算法生成目标框；而 SSD 算法直接在整幅图像上生成各种大大小小的目标边界框，然后使用非极大值抑制技术将高度重叠的边界框整合成一个。将处理候选区域的问题转化为线性回归的问题，利用线性回归思想找出与目标最为接近的预测框，该算法提高了计算速度，准确率也有一定的保证。采用这种方式的优点就是极大地提升了运算速度，适合用于计算力较低的硬件环境。

2 交通标志牌检测 SSD 过程

本文方法过程分为两个阶段，训练阶段和测试阶段。

训练阶段，用课程给予的 5000 张附有标签（目标检测框）数据的图片对 SSD 卷积神经网络进行训练，得到目标检测模型；

测试阶段，将新样本输入目标检测模型得到检测结果，即在图像上画出预测框并与原有标签做对比。

如图 1 所示，整个计算流程包含以下几个步骤：

- 1) 分析交通标志牌检测任务，得到结论，该项目的目的是利用已有图像以及标签数据，将不带有标签的
- 2) 对样本图像和标签数据进行预处理，以达到符合神经网络输入条件。具体的操作是将课程提供的标签数据（4225 个 XML 文件）使用 python 进行读取并合并成一个长列表，并图像的编号一一对应，方便模型读取。
- 3) 分辨率调整和图像数据增强，使用 opencv 的 python 接口，先将图片分辨率由原图的 1200×1600 调整为 300×400 ，降低分辨率可以在硬件条件不足的情况下完成既定的训练任务；再使用图像增强的方法，加强在视角转动的情况下目标检测的鲁棒性。
- 4) 选定神经网络结构。将训练集及其对应的标定列表作为输入，通过 SSD 网络对模型进行反复迭代训练，得到最终的交通标志检测模型；
- 5) 利用测试集新样本对交通标志检测模型进行测试，得到新样本的检测结果。

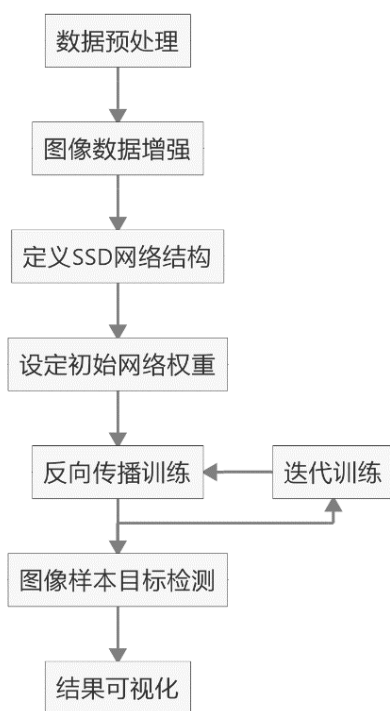


图 1 基于 SSD 网络的交通标志牌检测流程

2.1 任务分析及数据准备

本次课程作业的视觉任务是在城市道路场景中检测正面交通标志。视觉任务确定之后需要采集示例图像构建训练集，采集被测图像构建测试集。被检测的场景图像和目标是与视觉任务强相关的，所以在采集示例图像和被测图像应在相同环境下以相同的方式进行采集。在本课程作业的图像数据集中，图像采集者驾车在不同时间不同光照条件下，在上海市内行驶使用行车记录仪将行驶的图像存储。图像数据样本的典型场景如所示。



图 2 数据样本

再拿到数据样本之后，我们应该对样本进行标定，标定交通标志牌的左上和右下角的坐标值，形成一个矩形框，确定目标在图像上的位置

在本次课程作业中，所有图像数据已经准备好，对应的标签数据 xml 文件也已准备好，我们需要做得是人为地划定测试集和训练集，并将分散的 4225 个 xml 标签文件整合到一起。如表 1 所示，表格的第一列 image_name 是图像的名称；第二列到第四列代表在目标检测框的坐标；class_id 的值设为 1,1 代表着标志牌类，如果值设为 0，则代表背景类。最终，我采取了将图像编号为 4 的倍数的图像作为测试集，其他的图像作为训练集的方案。这样很具有代表意义。

表 1 整合后的数据标签示例

image_name	xmin	ymin	xmax	ymax	class_id
00000.jpg	472	450	504	546	1
00000.jpg	519	345	695	473	1
00000.jpg	695	363	737	475	1
00000.jpg	787	537	832	573	1
00001.jpg	502	326	678	472	1
00001.jpg	681	339	729	472	1
00001.jpg	449	442	485	537	1
00001.jpg	780	533	826	572	1
00002.jpg	481	314	665	455	1
00002.jpg	670	334	721	463	1
00002.jpg	422	431	462	535	1
00002.jpg	774	530	820	569	1
00003.jpg	463	298	658	448	1

2.2 图像数据增强

在图像的深度学习中，为了丰富图像训练集，更好的提取图像特征，泛化模型（防止模型过拟合），一般都会对数据图像进行数据增强，

数据增强，常用的方式就是随机裁剪(random crop),随机镜像(random mirror),随机尺度变换(random resize), 颜色抖动(color augmentation), 随机旋转(random rotation), 视角变换(aspect ratio augmentation), 扭曲图像特征，增强图像噪音等。

本文同时对训练数据做了数据增强和数据增广[15]。每一张训练图像，随机的进行如下一种或几种变换：

- 使用原始的图像
- 采样一个候选区域(patch)，与区域之间最小的重叠为：0.1，0.3，0.5，0.7 与 0.9
- 随机的采样一个候选区域(patch)，并进行视角转换

采样的 patch 是原始图像大小比例是 $[0.1, 1]$ ，视角转换比例(aspect ratio)在 $\frac{1}{2}$ 与 2 之间。

当目标检测框的中心(center)在采样的候选区域(patch)中时，保留重叠部分。在这些采样步骤之后，每一个采样的候选区域(patch)被拉伸到固定的大小，并且以 0.5 的概率随机的水平翻转(horizontally flipped)。

在本课程项目中，以上的图像操作使用了 Opencv 中的 python 接口 CV2 集成库。在某些角度上相当于我将原来的 5000 张图片数量变为几万张，增强了数据的广泛性和普遍性，使模型更加具有鲁棒性和泛化能力。

2.3 SSD 神经网络原理

SSD 模型来源于发表在 ECCV 2016 的一篇文章，是 UNC Chapel Hill（北卡罗来纳大学教堂山分校）的 Wei Liu 教授的新作。也是最近在目标检测领域炙手可热的方法之一，SSD 模型的特点就在既保证了目标检测的速度，又保证了精度。过去老旧的目标检测模型通常是将候选区域寻找和在区域内进行分类检测分为两个较为独立的网络结构，这种网络训练较为困难，且实时监测性不好；与传统网络结构不同，SSD 模型将整个检测过程整合成了一个相对简单的深度神经网络。

现今流行的效果较出色的检测系统大致都是如下步骤，先生成一些假设的边界框，然后在这些边界框中提取特征，之后再经过一个分类器，来判断里面是不是物体，是什么物体。

从 2013 年 Uijlings 等人的 Selective Search for Object Recognition[13]开始，2015 年发表的基于 Faster R-CNN 的 ResNet，这类方法的特点就是检测过程按部就班，导致检测精度很高，其中 ResNet 在 PASCAL VOC、MS COCO、ILSVRC 数

据集上一直取得领先。但对于嵌入式系统和移动端，这类方法所需要的计算时间太久了，不足以实时的进行检测。当然也有很多工作是朝着实时检测迈进，但目前为止，都是牺牲检测精度来换取实时性。

SSD 模型这种实时检测方法，消除了中间的生成边界框和像素采样的过程，取而代之使用了一种统一的网络架构达到目标检测的目的。虽然 SSD 不是第一个使用这种方法的模型（YOLO[14]在 SSD 之前发明），但是 YOLO 的检测精度仍然不够高，SSD 模型做了一些提升性的工作，既保证了速度，也保证了检测精度。该模型的最后效果是：比原先最快的 YOLO 方法，还要快，还要精确。其次在保证速度的同时，其结果的 mAP 指标可与使用 Faster R-CNN 相媲美。

SSD(Single Shot MultiBox Detector，单次检测)在 YOLO 回归得到检测框的思想及 anchor 机制基础上，解决了回归边框不精确的问题。

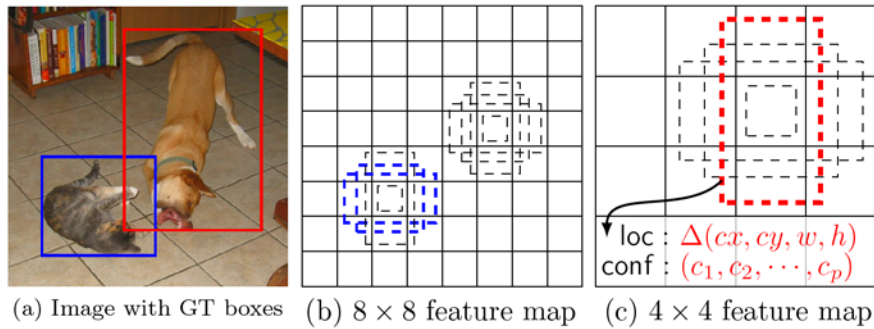


图 3 SSD 的检测过程示意图

该模型的结构如图 3 所示，也是使用回归得到目标位置和类别，不同之处在于 YOLO 使用所有的特征预测目标位置，而 SSD 使用这个位置周围的特征预测某个位置，并且使用更快的 R-CNN 的 anchor 机制建立某个位置和特征的对对应关系。在 8×8 大小的特征图中，选择 3×3 大小的滑动窗口提取特征，然后回归计算目标位置和类别信息，该模型的 anchor 作用在不同的特征图上，每个特征图对 3×3 的滑动窗口的感受范围不同，充分利用不用层的特征图，达到多尺度检测的目的。

SSD 利用了回归思想来获取边框，借助多尺度 anchor 机制回归计算出每个位置的多尺度特征，这样既缩短了目标检测的时间，也提高了预测窗的精确性。该模型在 VOC2007 测试，得到的 mAP 得分高达 72.1%，利用 GPU 可以

每秒处理 58 帧图像。

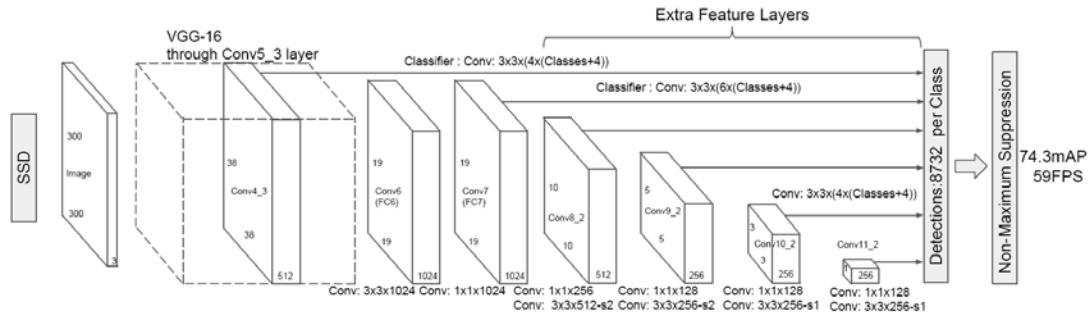


图 4 SSD 网络结构

具体的网络结构如图 4，SSD 的网络结构是一个单层深度神经网络，它将预测框输出空间离散化为一组默认框，并在所有特征图上设置了多种宽高比和尺寸。SSD 对所有的默认框计算对象的类别概率，并纠正边框向对象轮廓靠拢，对于不同尺寸的对象，预测的是不同分辨率下的多个特征图。由于消除了边界框的备选计算和之后的像素或特征重采样阶段，而用多个宽度比不同的滤波器来计算预测框中的对象类别与偏移，并在不同的特征图上使用滤波器作用于网络的后期，以便执行多尺度检测，进而实现了高精度检测，SSD 的速度有了显著的提升。

SSD 是属于前馈型的网络，其回归计算出预测框集合和预测框内目标类别的分数，最后采用非最大化抑制过程获取检测结果。基础网络是一个用于图像分类的前置网络，它可在分类层之前进行截断以提取特征，能实现这种功效的类似网络有 VGG16、GoogleNet、AlexNet、ResNet 等。SSD 添加了辅助网络结构，具有三个主要特征：

①多尺度特征图检测：在基础网络截断后面连接了多个特征提取层，用于预测出多种尺度与宽高比检测框的置信度和偏移量，这些卷积层提取不同的特征，输出的尺寸逐层变小，最终获取不同尺度下的分值。

②用于目标检测的卷积预测器：在整个网络结构中，共有六层网络，分为两类，即前置网络的截断层和添加的卷积层，使用卷积滤波器与特征层进行卷积运算，可以获取一个固定的预测集合，如模型中有 8732 种预测。如某个特征层由 P 个通道巧 $M*N$ 大小的像素组成，卷积核为 $3*3*P$ 大小，那么就会计算得到默认框内目标的类别分数及坐标偏移，在每个应用卷积核运算的 $M*N$ 大小位置处，

对于 $M \times N$ 的位置上，在卷积运算后得到输出值，坐标偏移值是相对于目标默认框的，目标默认框的位置是相对于整个特征图的。

③默认框与宽高比：在每个特征图上关联了一组多种尺度的默认框，并使用这些默认框进行卷积运算，那么这些默认框与对于的单元格的位置是固定不变的。预测的结果是每个默认框的位置偏移和目标的类别分数。如在某个位置的 K 的默认框，每个默认框会计算出 C 种类别分数与 4 个相对位置偏移量，于是每个位置的滤波器共有 $(C + 4) \times K$ 个，最终在 $M \times N$ 大小的特征图上得到 $(C + 4) \times K$ 个输出值。所以在不同的特征图上采用多种尺度的默认框，可以最大限度的离散出可能预测框位置。

优点：在训练阶段，SSD 只需要每个目标的图像与真实类别及边框信息，真正实现了端到端的训练与测试。在识别与检测上，对大目标比小目标有更好的性能，类别判断与边界框检测有更高的精度，速度相对于其他网络有很大提升，可以用于实时性检测的系统。

缺点：网络层次较深，计算起来可能会比较复杂，会影响到实时性的速度。可以采用简化网络结构来减轻计算量。

2.4 7 层 SSD 网络的实现

受限于显卡算力不足，我采用了简化版的七层 SSD 网络，它可以在交通标志牌目标检测任务中，保证一定准确度的同时可以快速地进行训练，免去了许多无畏的等待时间。

我使用的网络的结构如图 5 所示，我的 SSD 网络结构是在 7 层 CNN 卷积网络的基础上构建的。

首先先介绍网络的基础 CNN 卷积架构。其中，每个卷积层不单单是传统意义上的卷积网络，按顺序地集合了卷积层、规范层、激活函数以及池化层。

卷积层可以通过图像卷积来提出图像中的特征，在图 5 中，以卷积层 1 为例， $(5 \times 5) \times 32$ 代表着使用了 32 个 5×5 的卷积核，在 7 个卷积层中，卷积步长都为 1，每次卷积之前都会在图像外沿行补零操作。

规范层可以将前一层的激活值重新规范化，便于下一层进行运算，也即使其输出数据的均值接近 0，其标准差接近 1。我们在这里采用的是 3 个特征轴的规范，动态系数为 0.99。

激活函数可以为网络引入非线性性质，以更好地逼近结果，使得神经网络可以更好地解决较为复杂的问题。本文中，我采用了 ELU 激活函数，它的特点是计算较简单切非线性性质比较强，不容易产生梯度爆炸和梯度消失的现象。

池化层相当于对图像卷积后的结果进行了采样，有的时候我们不需要那么精确的结果。加入池化操作之后，可以降低卷积层输出的特征向量的维度。同时可以进一步减少过拟合现象。过拟合现象的存在是因为结果对于输入的某些误差过于敏感通过池化可以减少噪声。本文中，我采用的是 2×2 的极大值池化（Max-Pooling）。

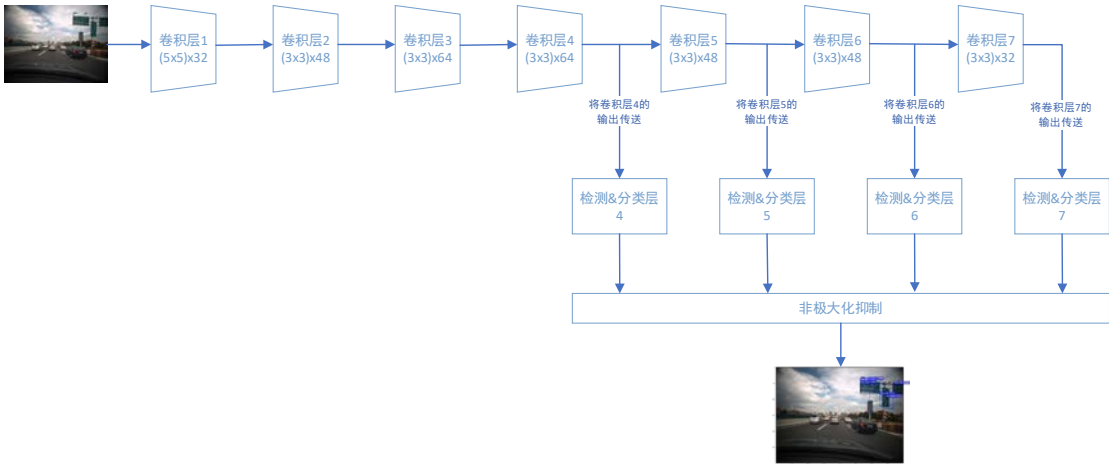


图 5 本文使用的简化版 7 层 SSD 网络结构

其次，我将介绍本课程作业的重点，检测&分类器。如图 5 所示，我将卷积层 4 到卷积层 7 的输出，分别传送到四个不同的检测&分类层 4 到 7（之所以命名为 4 到 7 是为了跟卷积层的名称对应）。在每一个检测&分类层中，我都进行了本文所需的检测+分类的任务，每一层都会根据该层输入图像的不同，进行不同的目标检测和分类（即绘制目标检测框并判定所属类别）。通过对这四个检测&分类层的检测结果进行非极大化抑制，最终 SSD7 模型将综合地生成一个最优的结果。

2.5 模型评价指标

一般来说，要评估某个目标检测模型的性能，主要分为一下两个方面进行，一是**定位损失**，二是**分类损失**。定位损失主要是指预测的目标检测框与标签给定目标检测框的差异；分类损失是指算法是否有正确地将物体识别出来，在交通标志牌检测的问题中，可以认为是一个交通标志牌和背景之间的二分类问题。最后，我要将定位损失和分类损失通过一定的方式有机结合，以较好地评估交通标志牌检测模型的性能。

①对于**定位损失**，我采用了 Smooth_L1_Loss[12]损失函数来评估：

在对某张图片进行目标定位的过程中，我们通常使用 $(Xmin, Xmax, Ymin, Ymax)$ 的坐标形式确定一个目标检测框。设该图片标签的目标检测框坐标为 (X_1, X_2, X_3, X_4) ，模型预测的目标检测框坐标为 (Y_1, Y_2, Y_3, Y_4) ，则 Smooth_L1_Loss 损失可以用以下方法估计：

$$Loss_{smooth} = \begin{cases} |X_i - Y_i| & otherwise, \\ 0.5 \times (X_i - Y_i)^2 & if |X_i - Y_i| < 1 \end{cases}$$

②对于**分类损失**，我采用了在分类问题中被普遍采用的对数极大似然估计：

$$Loss_{log} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M X(i|j) \log Y(i|j)$$

N：测试集样本数

M：类别数,这是是二分类问题，M=2

$X(i|j)$ ：第 i 个样本属于分类 j 时为 1，否则为 0

$Y(i|j)$ ：第 i 个样本被预测为第 j 类为 1，否则为 0

在本课程作业中，我定义模型的总损失函数为定位损失+分类损失。特殊的是，对于定位损失，我只计算了分类正确的样本的定位损失，分类错误的样本没有被我计算在内。综上，总的损失函数为：

$$Loss_{all} = Loss_{smooth}(positive\ sample) + Loss_{log}$$

3 模型训练和结果分析

3.1 优化方法

在本模型中，我采用了动态的 Adam 优化方法，具体地说，就是在 Adam 优化中，加入了反馈型的动态步长，当损失函数的变化率变小时，相应地减小学习步长，这样可以最大限度地避免求最优解时不断震荡，无法收敛的问题。

Adam 优化方法利用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率。Adam 的优点主要在于经过偏置校正后，每一次迭代学习率都有个确定范围，使得参数比较平稳。

除了具有动态学习率的优点以外，Adam 优化方法对内存需求较小，也适用于大多非凸优化，对大数据集和高维空间的数据表现良好。

3.2 训练环境

本程序的运行环境如表 2 所示，如果不在相应的版本下运行可能会有意想不到的兼容性问题。

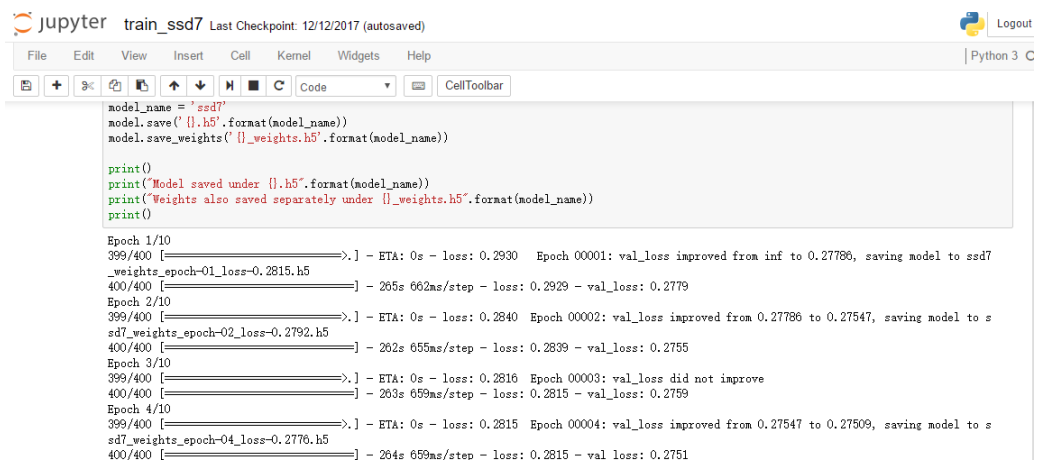
表 2 运行环境

框架	版本	硬件	型号
Python	3.60	CPU	Intel I7-6820
Anaconda	4.3.0 (64-bit)	显卡	NVIDIA GTX-1060
Numpy	1.13.3	Windows10	1703
TensorFlow	1.4.0		
Keras	2.1.1		
OpenCV	3.3.1		
Jupyter Notebook	4.3.1		

3.3 训练过程分析

根据一般的目标检测训练经验，SSD 的训练需要进行 100~200 个 epoch。为防止长时间训练的电脑断电、死机等情况，本程序的一次训练分为 40 个 epoch（数据流），每进行 40 个 epoch 的训练后对模型进行暂存，然后接着训练，直到误差值不再下降为止。

如图 6 所示，在刚开始进行训练时，准确率一直不高，误差一直在 30% 左右，但是其误差值一直维持着一个下降的趋势。



```
model_name = 'ssd7'
model.save('{}.h5'.format(model_name))
model.save_weights('{}_weights.h5'.format(model_name))

print()
print("Model saved under {}.h5".format(model_name))
print("Weights also saved separately under {}_weights.h5".format(model_name))
print()

Epoch 1/10
399/400 [=====] - ETA: 0s - loss: 0.2930 Epoch 00001: val_loss improved from inf to 0.27786, saving model to ssd7_weights_epoch-01_loss-0.2815.h5
400/400 [=====] - 265s 662ms/step - loss: 0.2929 - val_loss: 0.2779

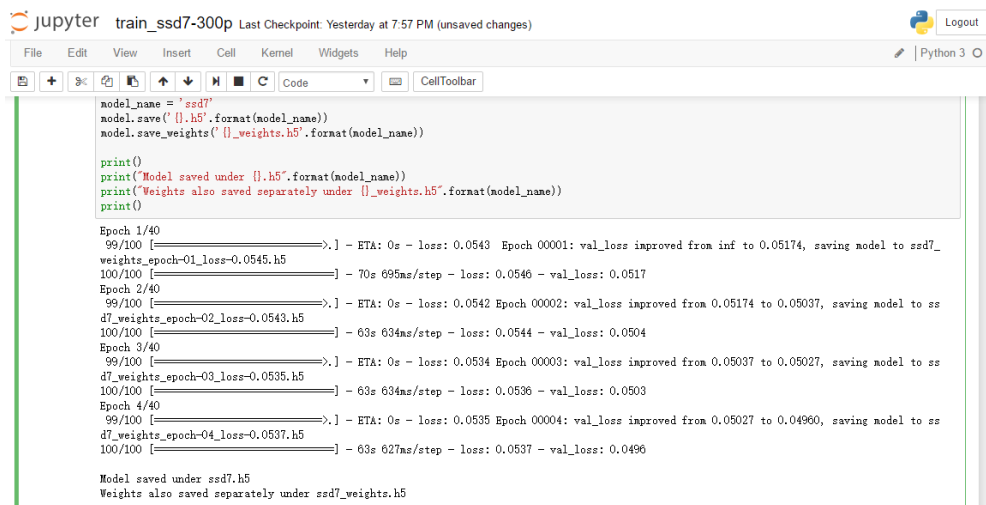
Epoch 2/10
399/400 [=====] - ETA: 0s - loss: 0.2840 Epoch 00002: val_loss improved from 0.27786 to 0.27547, saving model to ssd7_weights_epoch-02_loss-0.2792.h5
400/400 [=====] - 262s 655ms/step - loss: 0.2839 - val_loss: 0.2755

Epoch 3/10
399/400 [=====] - ETA: 0s - loss: 0.2816 Epoch 00003: val_loss did not improve
400/400 [=====] - 263s 659ms/step - loss: 0.2815 - val_loss: 0.2759

Epoch 4/10
399/400 [=====] - ETA: 0s - loss: 0.2815 Epoch 00004: val_loss improved from 0.27547 to 0.27509, saving model to ssd7_weights_epoch-04_loss-0.2776.h5
400/400 [=====] - 264s 659ms/step - loss: 0.2815 - val_loss: 0.2751
```

图 6 刚开始训练时的迭代结果

如图 7 所示，经过 84($40 \times 2 + 4$)个 epoch 训练，在我定义的评价指标下，最终的测试集和训练集的误差都达到了 5% 左右，并且误差值较稳定不再变动。由此可以证明，我建立的 SSD 网络在交通标志牌任务中的效果十分不错。



```
model_name = 'ssd7'
model.save('{}.h5'.format(model_name))
model.save_weights('{}_weights.h5'.format(model_name))

print()
print("Model saved under {}.h5".format(model_name))
print("Weights also saved separately under {}_weights.h5".format(model_name))
print()

Epoch 1/40
99/100 [=====] - ETA: 0s - loss: 0.0543 Epoch 00001: val_loss improved from inf to 0.05174, saving model to ssd7_weights_epoch-01_loss-0.0545.h5
100/100 [=====] - 70s 695ms/step - loss: 0.0546 - val_loss: 0.0517

Epoch 2/40
99/100 [=====] - ETA: 0s - loss: 0.0542 Epoch 00002: val_loss improved from 0.05174 to 0.05037, saving model to ssd7_weights_epoch-02_loss-0.0543.h5
100/100 [=====] - 63s 634ms/step - loss: 0.0544 - val_loss: 0.0504

Epoch 3/40
99/100 [=====] - ETA: 0s - loss: 0.0534 Epoch 00003: val_loss improved from 0.05037 to 0.05027, saving model to ssd7_weights_epoch-03_loss-0.0535.h5
100/100 [=====] - 63s 634ms/step - loss: 0.0536 - val_loss: 0.0503

Epoch 4/40
99/100 [=====] - ETA: 0s - loss: 0.0535 Epoch 00004: val_loss improved from 0.05027 to 0.04960, saving model to ssd7_weights_epoch-04_loss-0.0537.h5
100/100 [=====] - 63s 627ms/step - loss: 0.0537 - val_loss: 0.0496

Model saved under ssd7.h5
Weights also saved separately under ssd7_weights.h5
```

图 7 训练接近完成时迭代结果

3.4 结果分析与可视化

将训练好的模型对测试集进行测试，我选择了以下九张最具有代表性的结果进行分析。

如图 8，绿色方框代表标签（即实际交通标志牌存在的位置）；蓝色方框代表预测的目标检测框，蓝色方框上方的白字 `big_sign` 代表了该物体的类别，旁边的数字代表了该物体分类正确的概率。为了便于观察，其余的 8 张图中，不展示类别名和预测概率。



图 8 目标检测结果 a

在图 8 中，共有三个标志牌，它们成丁字状分布，左右下各一块标志牌。在课程提供的数据标签中，只标注了左和下的标志牌，即绿色框只存在于左边和下面。而在我建立的模型中，不但标签里有的左和下的标志牌被检测出来了，右边没有标签的标志牌也被我的模型检测出来了，说明我的模型鲁棒性和泛化能力很强。

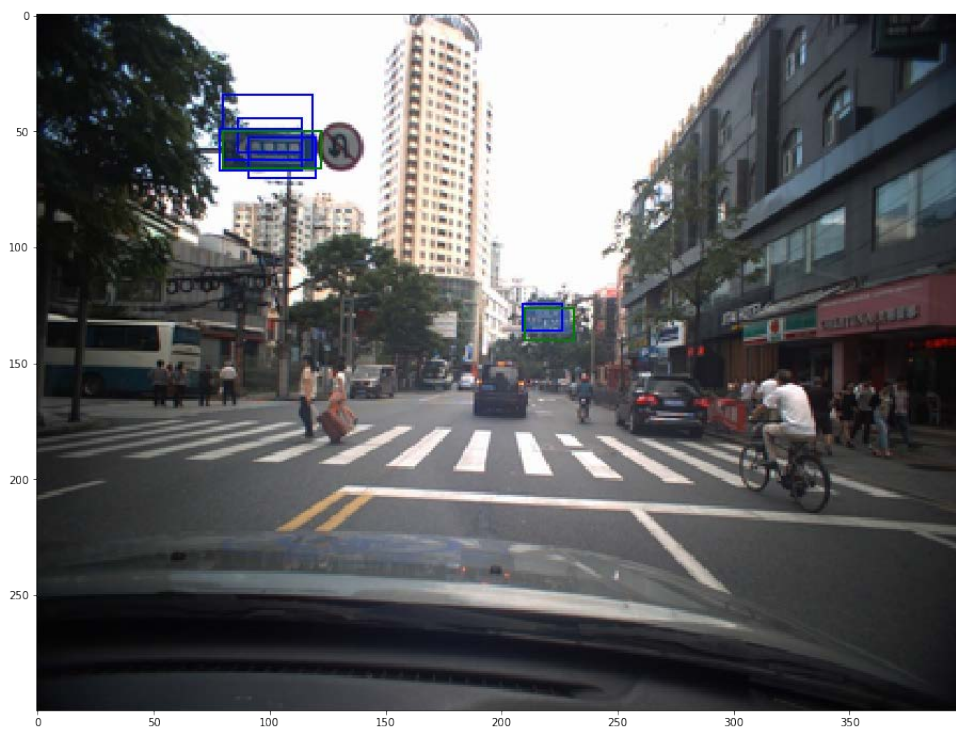


图 9 检测结果 b

如上图 9，在照明正常的条件下，两个标志牌都被检测出来，并且蓝色框和绿色框的重叠程度很高，属于较好的样本。



图 10 检测结果 c

如图 10，在黄昏时，照明条件不好，本模型还是能成功地检测。



图 11 检测结果 d

如图 11，在光线较暗的情况下，大目标的交通标志被检测出来，目标太小的标志牌无法被检测出来。

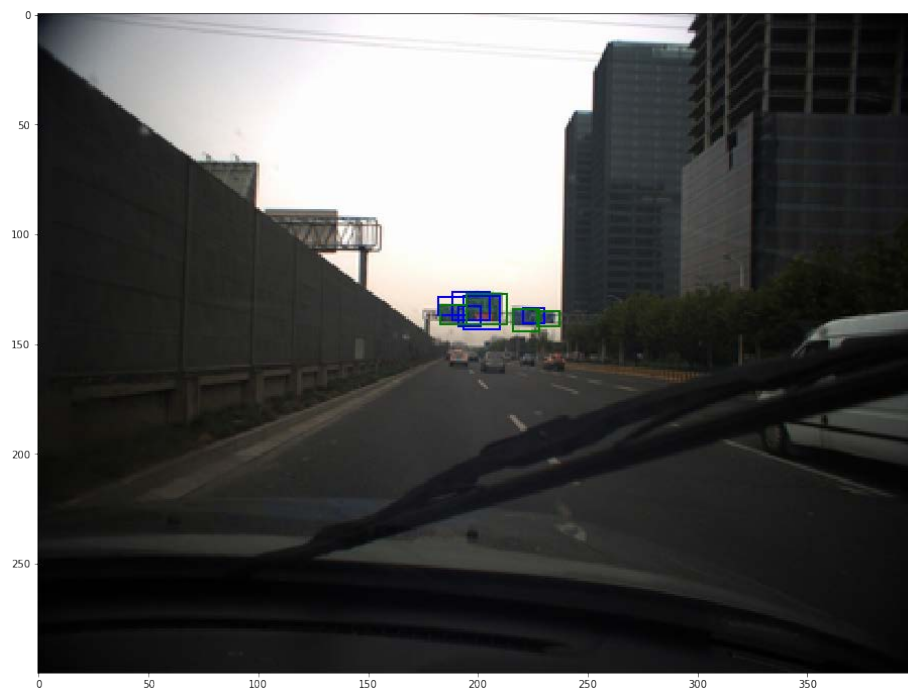


图 12 检测结果 e

如图 12，在逆光条件下，较小的标志牌还是可以检测出来，但目标框重叠情况一般。

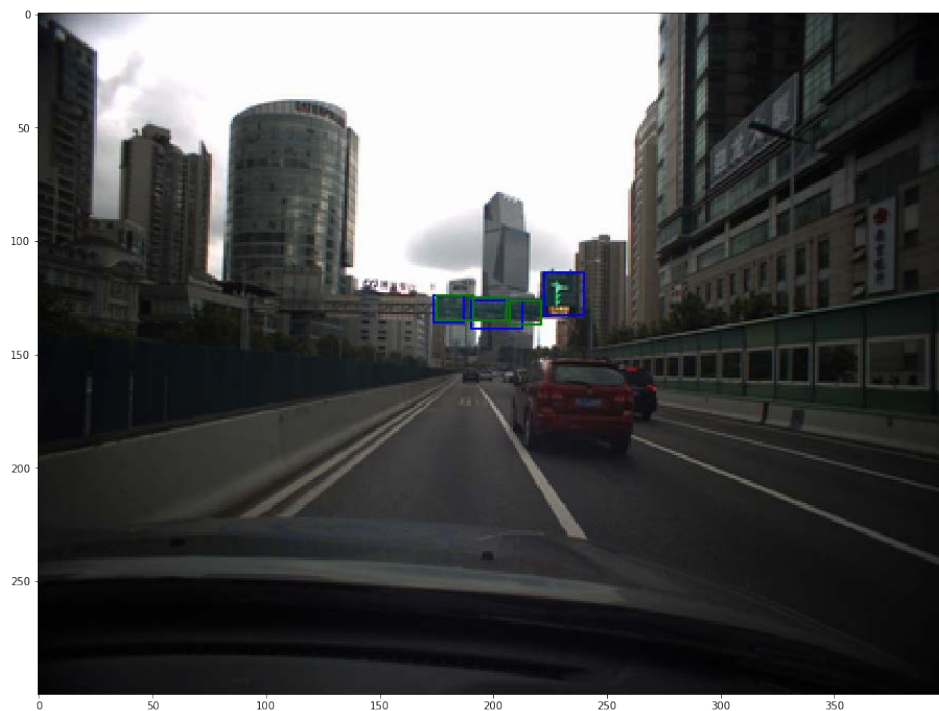


图 13 检测结果 f

如图 13 所示，三个交通标志牌只检测出了两个，相反地，旁边的路况信息牌被识别成了标志牌。

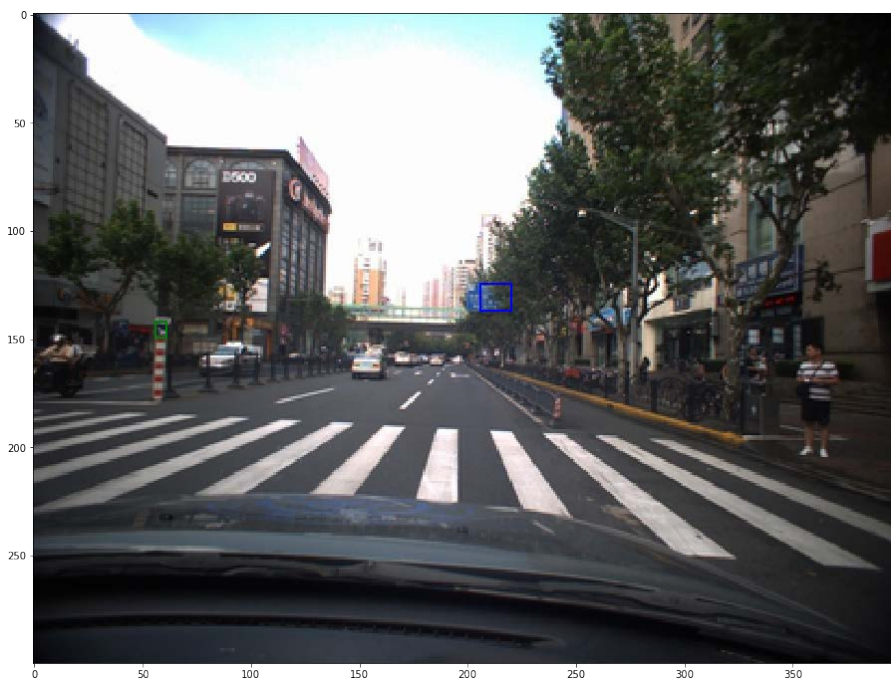


图 14 检测结果 g

如图 14 所示，在该图像中，本模型检测出了一个标签漏标的图像（只有蓝框没有绿色框）。



图 15 检测结果 h

如图 15 所示，在光线特别昏暗的条件下，交通标志牌无法被检测。即只有绿框无蓝框。

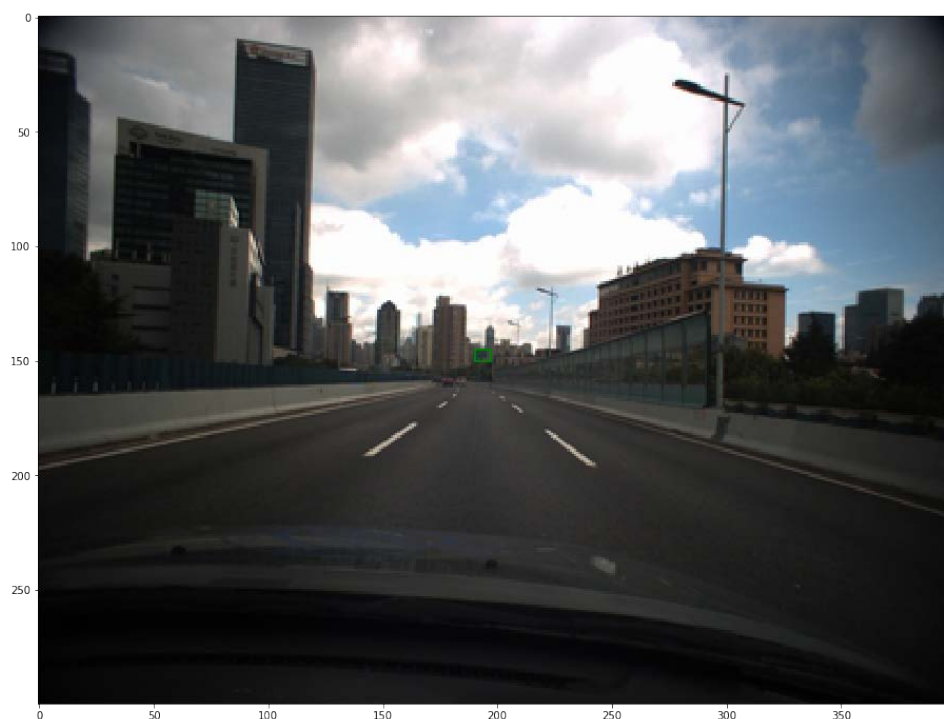


图 16 检测结果 i

如图 16 所示，在逆光的条件下，就及其小的交通标志牌无法被检测。

参考文献

- [1] De l E A, Moreno L E, Salichs M A, et al. Road traffic sign detection and classification[J]. IEEE Transactions on Industrial Electronics, 1997, 44(6):848-859.
- [2] Fang C Y, Chen S W, Fuh C S. Road-sign detection and tracking[J]. IEEE Transactions on Vehicular Technology, 2003, 52(5):1329-1341.
- [3] Lopez L D, Fuentes O. Color-based road sign detection and tracking[C]//International Conference Image Analysis and Recognition. Springer, Berlin, Heidelberg, 2007: 1138-1147.
- [4] Maldonado-Bascon S, Lafuente-Arroyo S, Gil-Jimenez P, et al. Road-sign detection and recognition based on support vector machines[J]. IEEE transactions on intelligent transportation systems, 2007, 8(2): 264-278.
- [5] LeCun Y, Kavukcuoglu K, Farabet C. Convolutional networks and applications in vision[C]//Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, 2010: 253-256.
- [6] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
- [7] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[C]//European Conference on Computer Vision. Springer, Cham, 2014: 346-361.
- [8] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [9] Bouchard G. Clustering and classification employing softmax function including efficient bounds: U.S. Patent 8,065,246[P]. 2011-11-22.
- [10] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

- [11] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.
- [12] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[C]//Advances in neural information processing systems. 2015: 91-99.
- [13] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition[J]. International journal of computer vision, 2013, 104(2): 154-171.
- [14] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.
- [15] Must Know Tips/Tricks in Deep Neural Networks
<http://210.28.132.67/weixs/project/CNNTricks/CNNTricks.html>

代码说明

本项目的 github: https://github.com/WangXuhongCN/classproject_DIP

名称	修改日期	类型	大小
.git	2017/12/22 星期五 10:22	文件夹	
.github	2017/12/22 星期五 10:21	文件夹	
.ipynb_checkpoints	2018/1/4 星期四 19:57	文件夹	
__pycache__	2017/12/22 星期五 10:22	文件夹	
.gitattributes	2017/12/22 星期五 10:21	GITATTRIBUTES ...	1 KB
.gitignore	2017/12/6 星期三 22:01	GITIGNORE 文件	2 KB
keras_layer_AnchorBoxes	2017/12/6 星期三 22:01	PY 文件	15 KB
keras_layer_L2Normalization	2017/12/6 星期三 22:01	PY 文件	3 KB
keras_ssd_loss	2017/12/6 星期三 22:01	PY 文件	13 KB
keras_ssd7	2017/12/6 星期三 22:01	PY 文件	21 KB
ssd_batch_generator	2017/12/6 星期三 22:01	PY 文件	70 KB
ssd_box_encode_decode_utils	2017/12/12 星期二 20:39	PY 文件	64 KB
ssd7.h5	2017/12/13 星期三 18:04	H5 文件	2,345 KB
ssd7_weights.h5	2017/12/13 星期三 18:04	H5 文件	832 KB
train_ssd7 我的程序	2018/1/5 星期五 23:27	IPYNB 文件	324 KB