应用型硕士学位论文

一种基于深度学习的路面弯道标志检测方法

Method for Traffic Sign Detection of the Curved Roads Based on Deep Learning

作　　者：刘向恒
导　　师：鲍　宇 副教授

中国矿业大学
二〇一八年五月

学位论文使用授权声明

本人完全了解中国矿业大学有关保留、使用学位论文的规定，同意本人所撰写的学位论文的使用授权按照学校的管理规定处理：

作为申请学位的条件之一，学位论文著作权拥有者须授权所在学校拥有学位论文的部分使用权，即：①学校档案馆和图书馆有权保留学位论文的纸质版和电子版，可以使用影印、缩印或扫描等复制手段保存和汇编学位论文；②为教学和科研目的，学校档案馆和图书馆可以将公开的学位论文作为资料在档案馆、图书馆等场所或在校园网上供校内师生阅读、浏览。另外，根据有关法规，同意中国国家图书馆保存研究生学位论文。

（保密的学位论文在解密后适用本授权书）。

作者签名：　　导师签名：

年　月　日　　年　月　日

中图分类号 TP393　　学校代码 10290

UDC 004　　密　　级 公开

中国矿业大学

应用型硕士学位论文

一种基于深度学习的路面弯道标志检测方法

Method for Traffic Sign Detection of the Curved Roads Based on Deep Learning

作　　者 刘向恒　　导　　师 鲍 宇

申请学位 工程硕士　　培养单位 计算机科学与技术学院

学科专业 计算机技术　　研究方向 图像检测

答辩委员会主席 姜淑娟　　评 阅 人 盲 评

二〇一八年五月

致谢

时光荏苒，不知不觉已经在中国矿业大学度过了我的研究生生涯。虽然只有短短的两年，却是我有生以来难忘的两年。期间，我不仅收获了快乐和成长，还有和老师同学满满的情谊。在这里对他们表示深深的谢意。

我要感谢我的指导老师鲍宇老师对我的教导。从论文的选题、构思、撰写到最终的定稿，鲍宇老师都给了我悉心的指导和热情的帮忙，使我的毕业论文能够顺利的完成。鲍宇老师对工作的认真负责、对学术的钻研精神和严谨的学风，都是值得我终生学习的。再次向给予我鼓励和指引的鲍宇老师致以真挚的谢意和崇高的敬意！

感谢计算机院为我提供了良好的学习生活环境，感谢领导以及各位老师对我无微不至的关怀和指导，感谢 A5081 宿舍可爱的舍友们，你们的帮助和陪伴，让我可以每天都开心快乐度过我的研究生生活，谢谢你们。

感谢我的家人在此期间给予我的包容、关爱和鼓励，以及所有陪我一路走来的同学和朋友，正是由于他们的支持和照顾，我才能安心学习，并顺利完成我的学业。

最后，我要感谢参与我论文评审和答辩的各位老师，给了我一个审视两年来学习成果的机会，让我能够明确今后的发展方向，这些帮助是一笔无价的财富。再次感谢各位老师，祝各位老师一生幸福、安康！

摘　要

近年来，全球每年都有数百万人死于车祸，使得越来越多的机构和个人致力于驾驶辅助技术和自动驾驶技术的研究。交通标志的检测与识别技术对带摄像头的自动驾驶系统是非常重要的。车辆在弯路上的侧翻的概率是直道的两倍，因此对弯道标志的检测与识别显得尤为重要。

有多种目标检测方法应用于交通标志分类，例如支持向量机和稀疏表示等方法。在德国交通标志检测竞赛中，使用卷积神经网络的方法比这些简单的分类器的性能更加优秀。所以本课题主要研究使用卷积神经网络来实现弯道标志检测。

本文的主要目的是设计并实现一个能自动检测弯道标志的程序，主要工作如下：

首先，建立了一个数据集。国内有关交通标志的研究较少，导致我国交通标志数据的缺乏。为实现对 5 类弯道交通标志的检测，通过互联网搜索和街景截图收集弯道标志图片，并使用图像标注工具标注。数据集包含连续弯道、反向弯道-先右转、反向弯道-先左转、左急转弯和右急转弯 5 类弯道交通标志，共 400 多张图片。

其次，将 YOLO v2 目标检测算法（You Only Look Once）和 Faster R-CNN 目标检测算法(Faster Regions with CNN features)应用于弯道标志检测，并比较两种算法的性能。为实现交通标志检测，需要考虑准确性和实时性要求。因此，使用弯曲道路交通标志训练集对基于 YOLO v2 算法的模型和基于 Faster R-CNN 算法的模型进行了训练，并比较两种算法的计算效率和分类精度，选择了表现较好的算法。实验结果表明，YOLO v2 算法的交通标志检测召回率为 100％，在摄像头上检测性能达到了实时性。

最后，提升了 YOLO v2 算法的平均精度均值。为提高算法的精度，提出一种能提高 YOLO v2 精度的新方法。修改 YOLO v2 的代码然后训练目标检测模型，最终平均精度均值为 0.1867。与 YOLO v2 相比，提高了检测精度。得到的结果并不理想，有较大的改善空间，可通过进一步改进模型取得更好的结果。

论文有图 48 幅，表 11 个，参考文献 77 篇。

关键词：交通标志；目标检测；卷积神经网络；弯道

Abstract

In recent years, millions of people have been killed each year by car accidents over the world. This has aroused a growing number of institutions and individuals to devote to the research of Advanced Driver Assistance System and self-driving. Traffic road sign detection and recognition is important to self-driving system with camera while driving on the road. The odds of a rollover on curved roads increase by two times as compared to the straight. Therefore, traffic sign detection of the curved roads is even more important.

Various object detection methods were adapted for traffic-sign classification, e.g. based on SVMs and sparse representations. But, convolutional neural network approaches have been shown to outperform such simple classifiers when tested on the GTSRB benchmark. So this research focuses on an implementation of traffic sign detection of the Curved Roads using a convolution neural network.

The main objective of this paper is to design and construct a computer based program which can automatically detect the road sign on the curved roads. The main contributions of this paper are as follows:

One data set has been built. There is little related research on Chinese traffic signs due to the lack of Chinese traffic sign datasets previously available. In order to achieve the detection of all 5 categories of the traffic sign on the curved roads, we have collected data from the internet and some photographs of street scenes, and use image annotation tool to label them. The data sets contain across 5 different categories of traffic sign on the curved roads, and more than 400 pictures. They are dangerous corner ahead sign, series of sharp corners - right warning sign, series of sharp corners - left sign, left turn sign, right turn sign.

The YOLO v2 object detection algorithm (You Only Look Once) and the Faster R-CNN (Faster Regions with CNN features) object detection algorithm are applied to traffic sign detection of the curved roads, and discusses the comparison of the efficiency of two algorithms. To implement traffic sign detection, both accuracy and real-time requirements need to be considered. So we trained and compared two models (one based on YOLO v2, the other based on Faster R-CNN) on the training set for traffic sign of the curved roads. The computational efficiency and classification accuracy of the two algorithms are compared, and been chosen the better one. The experimental results show that the traffic sign detection recall of YOLO v2 algorithm is 100% and

the detection time is towards real-time in camera.

Finally, to improve the mAP of YOLO v2 algorithm, and to raise the precision of the algorithm, a new method is proposed to aim the precision of YOLO v2. We modify open source code of YOLO v2 to train object detection model and achieve a mAP value that is 0.1867. Compared with YOLO v2, the proposed method improves in detection precision. The result we get is not ideal and leaves great space to improve, we can still improve the model to get a better result in our future work.

This paper has 48 charts, 11 tables, 77 references.

Keywords: Traffic sign; object detection; Convolutional Neural Network; Curved road

目 录

Contents

图清单

表清单

1 绪论

1 Introduction

1.1 研究背景和意义（Research Background and Significance）

随着中国城镇化速度的加快，以及中国经济持续快速发展，机动车保有量保持快速增长态势。截至 2017 年底，全国机动车保有量达 3.10 亿辆，仅 2017 年在公安交通管理部门新注册登记的机动车 3352 万辆[1]，均创历史新高。有文献表明，由于在弯道处存在驾驶员的视距受限、离心力、操作复杂等不利因素，相比直线路段而言更容易发生侧滑、急刹车、倾翻等事件，有报告称[2]平均每公里弯道发生事故次数比直线路段多 34%。

一般通过在弯道处合适位置设置警告标志、车道画标线和限速标志等方法以提高车辆在弯道处的安全性[3]。如果车辆在弯道处行驶时，有设备及时检测到弯道标志并根据驾驶员反应时间提醒驾驶员前方弯道种类，便可以及时做出正确地操作，降低弯道处发生事故的可能性。

驾驶辅助技术和自动驾驶技术可以减少交通事故数量[4]。历史上道路交通标志的检测和识别（Traffic Sign Recognition，TSR）[5-6]是高级驾驶辅助系统（Advanced Driver Assistance Systems，ADAS）中首先研究并应用的领域之一，近年来一些国内外的汽车厂商和互联网公司大力发展汽车自动驾驶技术[7]。目前虽然已有先进的驾驶员辅助系统[8]，但驾驶员仍然是导致道路交通事故的主要因素[9]，因此可通过使用驾驶员辅助系统降低因人为导致的交通事故数量以提高道路安全性。由于驾驶辅助技术和自动驾驶技术都需要根据路面交通标志，来了解实时路况，因此实现交通标志检测对驾驶辅助技术和自动驾驶技术必要的，弯道交通标志是交通标志的重要组成部分。自动驾驶汽车以雷达、GPS 和摄像头等设备感知测量周边环境，通过这些设备获得的信息做出合理的判断，使用摄像头检测周围的交通标志。

因此，研究并实现基于计算机视觉的道路弯道标志检测，能够有效地提升驾驶的安全性，并且具有一定的理论意义和实用价值。

1.2 国内外研究现状（Research Overview）

德国、美国和中国都曾举办过智能车竞赛，竞赛中参赛者研发了智能车系统，实现了对交通标志的检测与识别，并应用到实际环境中。

传统的交通标志检测和识别任务主要解决如何定位、如何提取特征、如何分类等问题[10]。

对道路交通标志提取特征的典型方法有基于颜色信息提取和基于形状提取。基于颜色信息的标志检测主要研究如何选择最合适的色彩空间，以及实现对色彩的鲁棒性；基于形状的标志检测一般使用霍夫变换[11]、Canny 边缘[12]或凸包[13]等方法。

简单的模板匹配（例如互相关相似性）和复杂的机器学习技术（例如 SVM[14]，boosting[15]，随机森林[16]等）等分类技术已足够解决交通标志的识别问题。

实现图像中的道路标志检测关键在于找到并分割出感兴趣的区域[17]（regions of interest，ROI），图像分割方法分为两大类：按颜色分割或按形状分割。在一些研究中，通过使用阈值以及 RGB，HSV[18]或 HSI[19]等色彩空间从图像中提取标志。然后使用例如模板匹配[20]或其他类型的分类器（例如 SVM[21]或神经网络[22-23]）。行人检测获得的研究成果[24]也被用于交通标志。

二十世纪八十年代末，卷积神经网络（Convolutional Neural Network,CNN）已应用于计算机视觉。然而，该方面的应用并不广泛，直到十年前由于大型公共图像库（如拥有上百万已标注图像的 ImageNet 训练集）的出现，以及计算机硬件飞速发展提供的强大计算能力，使卷积神经网络得到了快速发展。深度神经网络（Deep Neural Network，DNN）的出现，使得计算机视觉和语音识别在内的多个应用领域方面取得了巨大的成功[25]。计算机视觉领域中，被称为卷积神经网络的深度神经网络，在物体识别[26-27]和目标检测[28]中是其中最先进的技术，目标检测技术应用在交通标志上取得了较好的效果[29]。使用卷积神经网络（深度卷积网络）的深度学习[30]通过端到端学习彻底改变了计算机视觉，端到端学习也就是从原始数据中开始学习。

德国交通标志检测竞赛(GermanTraffic Sign Recognition Benchmark, GTSRB)[31]的举办，极大地促进交通标志检测技术的发展，通过该竞赛使欧洲的交通标志检测技术达到世界先进水平。本次比赛共有 18 支参赛队伍，VISICS 队[32]、Litsi 队[33]以及 wgy@HIT501 队[34]是表现最出色的队伍。VISICS 队使用 Dollár 等人[35]提出的用于行人检测的积分通道特征（Integral Channel Feature,ICF），并进一步改进了该方法[36]。Litsi 团队使用颜色分类和形状匹配来建立感兴趣的区域，然后使用 HOG 和带有 SVM 的颜色直方图来检测符号。最后，wgy@HIT501 队使用方向梯度直方图（Histogram of oriented gradient，HOG），通过潜在狄利克雷分布（LDA，Latent Dirichlet Allocation）查找候选区域，并使用 HOG 与 IK-SVM 执行更进一步的检测。这三种方法在特征提取时非常相似。

1.3 研究内容与论文结构（Research Contents and Main Work）

1.3.1 研究内容

本文的研究目的是设计并实现一个能自动检测弯道标志的程序。主要贡献如下：

（1）收集 5 类弯道标志数据集。包含 400 多张弯道标志图片，并使用标注工具制作用于图像检测的数据集。

（2）将 Faster R-CNN 目标检测算法和 YOLO v2 目标检测算法应用于弯道标志检测，然后对两种算法进行比较。

（3）在原 YOLO v2 算法框架上适当加深了网络结构，结果显示，改进后 YOLO v2 算法比原来的算法具有更高的检测精度。

1.3.2 论文结构

本文组织结构如下：

第 1 章：绪论部分。主要论述了弯道交通标志识别相关的研究背景与意义，总结了国内外的一些研究现状，并对论文的组织结构进行了简单的描述。

第 2 章：神经网络技术是本课题实现目标检测的主要技术，本章介绍了卷积神经网络的基础理论，如神经元、多层感知器、卷积神经网络的结构，以及神经网络的训练方法，通过这些理论的了解将为下一步系统设计提供理论支持。

第 3 章：本章建立弯道交通标志的目标检测数据集，介绍了 YOLO 系列目标检测算法和 Faster R-CNN 目标检测算法，将 YOLO v2 算法和 Faster R-CNN 算法应用于弯道标志检测，然后实现并比较测试结果。接下来选择表现优秀的 YOLO 算法进行下一步研究。

第 4 章：对原 YOLO v2 算法进行了改动，首先介绍了改进的内容和理由，然后使用图片和摄像头进行测试，计算改进后检测器的 mAP 值，与原算法进行比较，精度值有一定进步，证明改动是有效的。

第 5 章：总结和展望，对论文所做的主要工作进行总结与概述，然后对接下来希望进行的研究工作进行相关介绍。

2 卷积神经网络概述

2 An Overview of Convolution Neural Networks

2.1 人工神经网络基本结构（The Basic Structure of Artificial Neural Network）

2.1.1 感知机

一个神经元有多个用来接受传入信息的树突，只有一条轴突，神经元通过轴突尾端的轴突末梢与其他数个神经元的树突相连接来传递信息，在生物学上该连接位置称为“突触”。图 2-1 是生物的神经元形状的示意图。

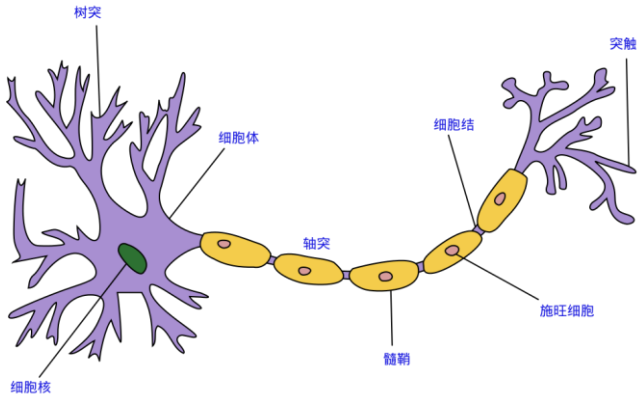


图 2-1 神经元

Figure 2-1 Nerve cell

十九世纪四十年代，McCulloch 和 Pitts 根据生物神经元的结构提出神经元模型。神经元模型是一个可以有多个输入，但只能有一个输出的单元。图 2-2 表示一个包含 3 个输入的神经元模型。

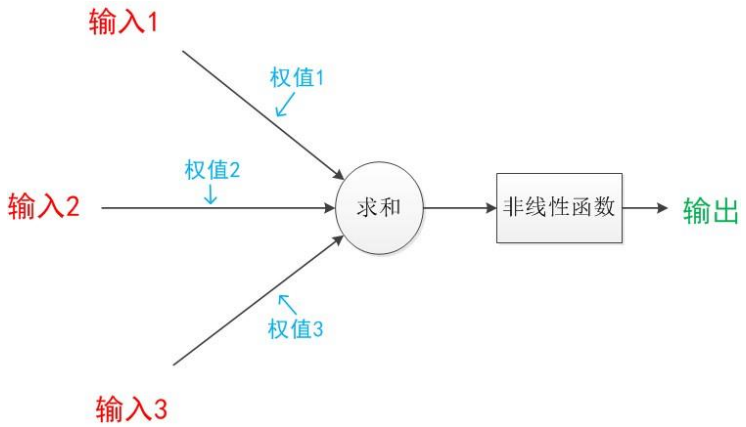


图 2-2 神经元模型

Figure 2-2 Neuron model

将神经元模型图中的变量用符号表示，如图 2-3 所示。

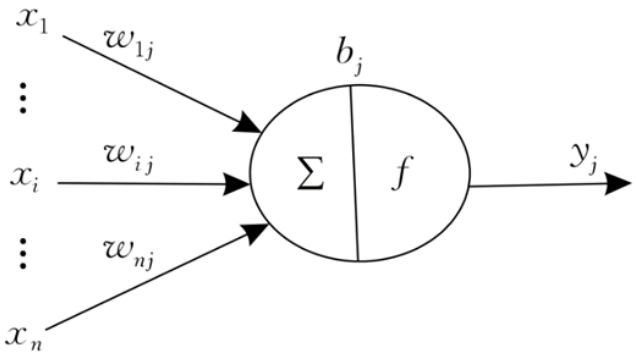


图 2-3 神经元模型的符号表示

Figure 2-3 Symbolic representation of neuron models

用公式表示：

$$y_j = \sum_{i=1}^{n} w_{ij}x_i - \theta_j \qquad (2-1)$$

神经元模型和生物神经元的对应关系如表 2-1 所示：

表 2-1 生物神经元和 MP 神经元模型

Table 2-1 Biological neurons and MP neurons model

生物神经元	MP 神经元模型
神经元	j
输入信号	x_i
权值	w_{ij}
输出信号	y_j
总和	$\sum$
膜电位	$\sum_{i=1}^{n} w_{ij}x_i$
阈值	θ_j

上个世界五十年代，科学家 Frank Rosenblatt 提出感知机（perceptron）[37]，感知机是一个单层神经网络[38]。

以神经元节点为输入单元，添加到神经元模型的“输入”位置，就变为感知机模型[39]。感知机可以视为生物神经细胞的简单抽象。单个神经细胞可以被看做一种只有“是”和“否”两种状态的机器——激活时为“是”，未激活时为“否”。当神经细胞的输入信号量总和超过了某个阈值时，就会激活产生电脉冲，并将电脉冲传递到其它神经元。于是提出了模拟神经细胞的感知机概念，其中权重对应突触、偏置对应阈值及激活函数对应细胞体。感知器可视为前馈神经网络，即输入层向输出层单向传播，是一种二元线性分类器。

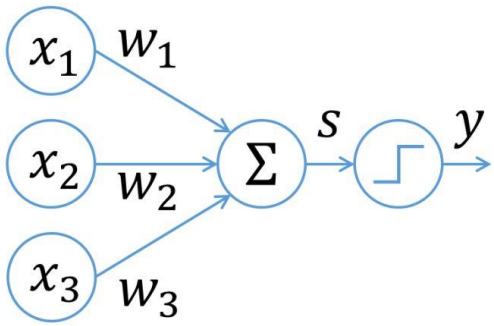


图 2-4 感知机

Figure 2-4 Perceptron

如图 2-4 所示是一个简单的感知机模型，x_1、x_2、x_3 是二进制数，表示感知器的输入，用w_1、w_2、w_3 等实数作为权重来决定x_1、x_2、$x_3 \ldots x_m$ 对输出的重要程度，给每一个输入x_i 赋一个权值 w_i，当加权和$\sum w_i x_i$ 小于等于阈值（threshold value）时则输出值是 0，否则为 1。如以下公式所示：

$$\text{output} = \sigma(\sum w_i x_i) = \begin{cases} 0 & \sum w_i x_i \leq \text{threshold} \\ 1 & \sum w_i x_i > \text{threshold} \end{cases} \qquad (2-2)$$

改变权重和阈值的数值，就可以得到新的感知机，表示新的决策模型。

为了简化对感知机的描述，可以使用向量点乘$w \cdot x$来表示 $\sum w_j x_j$，w=[w_1, w_2,…, w_i], x=[x_1,x_2,…,x_i]，w表示感知机的权重向量，x表示感知机输入向量，令 $b = -\text{threshold}$，即将阈值（threshold）替换为偏置（bias），然后后可得感知机规则：

$$\text{output} = \sigma(w \cdot x + b) = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \qquad (2-3)$$

感知机的激活函数是阶跃函数σ，即：

$$\sigma(x) = \begin{cases} 0, x \leq 0 \\ 1, x > 0 \end{cases} \qquad (2-4)$$

激活函数有很多不同的形式，阶跃函数是其中最简单的一种。当多个感知机叠加能产生了比单个感知机更为强大的表达能力。

2.1.2 人工神经网络

多层感知机是一个人工神经网络。神经元被排列成神经元网络，一排神经元称为层，一个网络可以有多个层。网络中神经元的结构通常称为网络拓扑。多层感知机由多个的神经元层组成，相邻层之间都是全连接的[40]，即每一层的每一个神经元与相邻层的所有神经元都有连接。

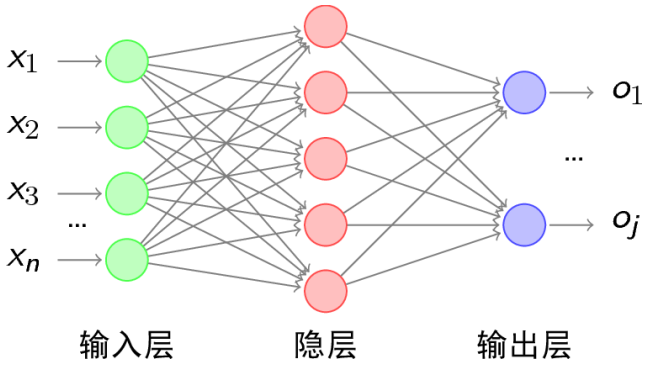


图 2-5 多层感知机

Figure 2-5 Multilayer perceptron

图 2-5 是一个多层感知机模型。其中接受输入的层称为可见层，因为它在网络中接受外界输入。

输入层后面的层被称为隐层，因为不直接接收输入。最简单的网络结构是在隐藏层中有一个直接输出值的单个神经元。由于计算能力和效率的增加，可以构建非常深的神经网络。深层学习是指在你的神经网络中有许多隐藏层，深度神经网络在以前计算能力弱的时候训练比较慢，但现在的技术和硬件可能需要几秒或几分钟就训练完成。

隐藏层中最后一层为输出层，它负责输出值或符合格式的值向量。

一种被称为反向传播算法的监督学习方法常被用来训练多层感知机[41]。多层感知机由感知机推广而来，相对感知机来说有能识别线性不可分的数据的优点[42]。

2.1.3 基于梯度的优化方法

人工神经网络训练的目的，就是使得参数尽可能的与真实的模型逼近[43]。大多数深度学习算法需要进行某种优化[44]。优化指通过调整自变量 x 的值使得某个函数$f(x)$的值最小或者最大，函数$f(x)$称为目标函数，如果是最小化，那么该函数称为损失函数。

假设有一个函数$y = f(x)$，其中 x 和 y 是实数。这个函数的导数（derivative）记为$f'(x)$或 dy/dx。导数$f'(x)$代表$f(x)$在点 x 处的斜率，表明如何缩放输入的变化才能在输出获得相应的变化：$f(x+\epsilon) \approx f(x) + \epsilon f'(x)$。

因此导数对于最小化一个函数很有用，因为通过它可得知如何更改 x 来略微地改善 y。例如，对于足够小的 ϵ 来说，$f(x - \epsilon \mathrm{sign}(f'(x))) \approx f(x) + \epsilon f'(x)$是比$f(x)$小的。因此可以将 x 往导数的反方向移动一小步来减小$f(x)$。这种方法叫作梯度下降[45]（gradient descent）。

当$f'(x)=0$ 时，无法根据导数得到往哪个方向移动$f(x)$如何变化的信息。$f'(x)=0$ 的点称为临界点（critical point）或驻点（stationary point）。一个局部极

小点（local minimum）意味着这个点的$f(x)$小于所有邻近点，因此不可能通过移动无穷小的步长来减小$f(x)$。一个局部极大点（local maximum）意味着这个点的$f(x)$大于所有邻近点，因此不可能通过移动无穷小的步长来增大$f(x)$。有些临界点既不是最小点也不是最大点。这些点被称为鞍点（saddle point）。图 2-6 为各种临界点的示例。

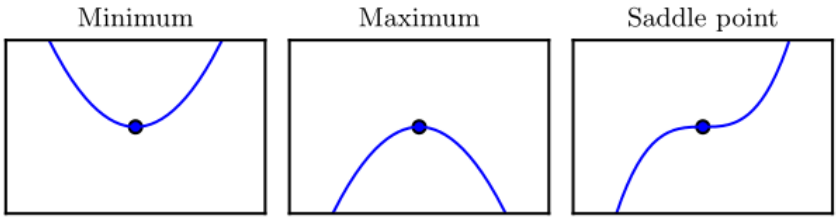


图 2-6 临界点

Figure 2-6 Critical point

使$f(x)$取得绝对的最小值（相对所有其他值）的点是全局最小点（global minimum）。函数可能只有一个全局最小点或存在多个全局最小点，还可能存在不是全局最优的局部极小点。在深度学习中，要优化的函数可能含有许多不是最优的局部极小点，或者还有很多处于非常平坦的区域内的鞍点。尤其是当输入是多维的时候，所有这些都将使优化变得困难。因此，通常寻找使f非常小的点，但这在任何形式意义下并不一定是最小。

最小化经常具有多维输入的函数：f:R_n→R。为了使“最小化”的概念有意义，输出必须是一维的(标量)。

对于具有多维输入的函数，需要用到偏导数（partial derivative）的概念。偏导数$\frac{\partial}{\partial x_i} f(x)$是衡量点 x 处只有$x_i$增加时$f(x)$如何变化。梯度（gradient）是相对一个向量求导的导数: f的导数是包含所有偏导数的向量，记为 $\nabla x f(x)$。梯度的第 i 个元素是f关于x_i的偏导数。在多维情况下，临界点是梯度中所有元素都为零的点。

在 u（单位向量）方向的方向导数（directional derivative）是函数 f 在 u 方向的斜率，既方向导数是函数 f(x+ α u)关于 α 的导数（在 α =0 时取得）。使用链式法则，可得当 α =0 时，$\frac{\partial}{\partial \alpha} f(x+\alpha u) = u^{\top}\nabla x f(x)$。

为找到$f(x)$最小值，需要找到使$f(x)$降得最快的方向，可通过下式计算方向导数找到该方向：

$$\min_{u,u^{\top}u=1} u^{\top}\nabla_x f(x) = \min_{u,u^{\top}u=1} \| u \|_2 \| \nabla_x f(x) \|_2 \cos\theta \qquad (2-5)$$

其中 θ 是 u 与梯度的夹角。将$||u||_2$=1 代入，忽略与 u 无关的项，就能简化得

到 mincosθ。这在 u 与梯度方向相反时取得最小。即梯度向量指向上坡，负梯度向量指向下坡。在负梯度方向上移动可以减小f。这被称为最速下降法(method of steepest descent)或梯度下降（gradient descent）。

最速下降建议新的点为$x' = x - \epsilon \nabla x f(x)$。其中 ϵ 为学习率（learning rate），是一个确定步长大小的正标量。可以通过几种不同的方式选择 ϵ。普遍的方式是选择一个小常数。有时通过计算，选择使方向导数消失的步长。还有一种方法是根据几个 ϵ 计算$f(x - \epsilon \nabla x\ f(x))$，并选择其中能产生最小目标函数值的 ϵ。这种策略被称为线搜索。

最速下降在梯度的每一个元素为零时收敛（或在实践中，很接近零时）。在某些情况下，也许能够避免运行该迭代算法，并通过解方程$\nabla x f(x) = 0$直接跳到临界点。

虽然梯度下降被限制在连续空间中的优化问题，但不断向更好的情况移动一小步（即近似最佳的小移动）的一般概念可以推广到离散空间。递增带有离散参数的目标函数被称为爬山（hill climbing）算法。

2.1.4 反向传播算法

反向传播的算法过程[46]为：正向计算得到误差函数，反向求导梯度下降。反向传播算法经常被误解为用于多层神经网络的整个学习算法，实际上反向传播仅指用于计算梯度的方法，而另一种算法，例如随机梯度下降[47]，使用该梯度来进行学习。

当使用前馈神经网络接收输入 x 并产生输出 $\hat{y}$ 时，信息通过网络向前流动。输入 x 提供初始信息，然后传播到每一层的隐藏单元，最终产生输出$\hat{y}$。这称之为前向传播（forward propagation）。在训练过程中，前向传播可以持续向前直到它产生一个标量代价函数$J(\theta)$。反向传播（back propagation）算法简称为 backprop，允许来自代价函数的信息通过网络向后流动，以便计算梯度。

数值化地求解这样的表达式在计算上的代价可能很大。反向传播算法使用简单和廉价的程序来实现这个目标。微积分中的链式法则（为了不与概率中的链式法则相混淆）用于计算复合函数的导数。反向传播是一种计算链式法则的算法，使用高效的特定运算顺序。

设 x 是实数，f和 g 是从实数映射到实数的函数。假设 y=g(x)并且 z =f(g(x))= f(y)。那么链式法则表示为：

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} \tag{2-6}$$

使用链式规则，可以直接写出某个标量关于计算图中任何产生该标量的节点的梯度的代数表达式。然而，实际在计算机中计算该表达式时会引入一些额外的

考虑。

具体来说，许多子表达式可能在梯度的整个表达式中重复若干次。任何计算梯度的程序都需要选择是存储这些子表达式还是重新计算它们几次。在某些情况下，计算两次相同的子表达式纯粹是浪费。在复杂图中，可能存在指数多的这种计算上的浪费，使得简单的链式法则不可实现。在其他情况下，计算两次相同的子表达式可能是以较高的运行时间为代价来减少内存开销的有效手段。

给出确定的反向传播算法，它指明了梯度的直接计算方式。考虑描述如何计算单个标量 u(n)（例如训练样本上的损失函数）的计算图。想要计算这个标量对 n_i 个输入节点 u(1)到 u(n_i)的梯度。换句话说，希望对所有的 i∈{1,2,...,n_i}计算∂u(n)∂u(i)。在使用反向传播计算梯度来实现参数的梯度下降时，u(n)将对应单个或者小批量实例的代价函数，而 u(1)到 u(ni)则对应于模型的参数。

假设图的节点已经以一种特殊的方式被排序，使得可以一个接一个地计算他们的输出，从 u(ni+1)开始，一直上升到 u(n)。每个节点 u(i)与操作 f(i)相关联，并且通过对以下函数求值来得到 u(i)=f(A(i)),其中 A(i)是 u(i)所有父节点的集合。

该算法详细说明了前向传播的计算[48]，可以将其放入图 G 中。为了执行反向传播，可以构造一个依赖于 G 并添加额外一组节点的计算图。这形成了一个子图 B，它的每个节点都是 G 的节点。B 中的计算和 G 中的计算顺序完全相反，而且 B 中的每个节点计算导数∂u(n)∂u(i)与前向图中的节点 u(i)相关联。这通过对标量输出 u(n)使用链式法则来完成，如下式所示：

$$\frac{\partial u^{(n)}}{\partial u^{(j)}} = \sum_{i:j\in Pa(u^{(i)})} \frac{\partial u^{(n)}}{\partial u^{(i)}} \frac{\partial u^{(i)}}{\partial u^{(j)}} \qquad (2-7)$$

反向传播算法被设计为减少公共子表达式的数量而不考虑存储的开销。具体来说，它大约对图中的每个节点执行一个 Jacobian 乘积。反向传播算法访问了图中的节点 u(j)到节点 u(i)的每条边一次，以获得相关的偏导数∂u(i)∂u(j)。反向传播因此避免了重复子表达式的指数爆炸。然而，其他算法可能通过对计算图进行简化来避免更多的子表达式，或者也可能通过重新计算而不是存储这些子表达式来节省内存。

2.2 卷积神经基本网络结构（The Basic Structure of Convolution Neural Network）

卷积神经网络（convolutional neural network,CNN）也叫做卷积网络（convolutional network），适合处理具有像网格分布特征的数据[49]，比如声音数据（可看作按时间先后顺序采样形成的一维网格）和图像数据（可看作空间上像素排列成的二维网格）。“卷积神经网络”中的“卷积”表示使用了卷积（convolution）

运算。神经网络中只要有卷积层就可以看作卷积神经网络。

卷积神经网络是一个深度监督学习结构[50]，可以被视为由一个自动特征提取器和一个自动特征提取器构成的体系结构 Icarnable 分类器。从中提取特征的自动特征提取器原始图像通过两个步骤执行：卷积滤波和下采样。用权重集实现的卷积滤波被认为是利用卷积层的内核来提取输入图像中的某些局部特征。在卷积图层中，卷积滤波后计算出的每个神经元中的值仅与输入图像或图层输出的小子集相关。此外，卷积层中的神经元共享相同的连接权重，以控制整个输入中的权重数量。卷积滤波后，每个滤波器的结果是通过非线性激活函数来拟合任何函数以提高卷积神经网络的能力。

卷积神经网络是一种前馈式神经网络[51]。信息流只往从输入到输出一个方向流动。与人工神经网络一样，卷积神经网络也是由受到生物启发而来。1958 年，DavidHubel 和 Torsten Wiesel 通过研究猫瞳孔区域与大脑皮层神经元的对应关系，发现当出现在眼前的物体边缘指向某个方向，就会导致某种神经元细胞活跃[52]。在 20 世纪 80 年代，Fukushima 在前人工作的指引下，提出了神经认知机这一模型，这是卷积神经网络的早期实现。在大脑的视觉皮层中，它由复杂细胞组成的层简单交替组合对视觉信息进行处理。

卷积神经网络体系结构有多个变种[53]。一般来说，它们由卷积层和池化层（或者叫下采样层）组成，卷积层和池化层组合成模块，堆叠在一起形成一个深层模型。图 2-7 展示了用于图像分类任务的经典卷积神经网络架构。卷积神经网络以图像为输入，然后经过多个由卷积层和下采样层组成的模块；然后，将这些操作的结果作为一个或多个全连接层的输入；最后，最末尾的一个全连接层输出各个类标签的置信度，得出分类结果。这是最流行的基础架构，但近年来已经出现了一些变种，这些变种能提高图像分类准确性或降低计算成本。本节简单地介绍标准的卷积神经网络体系结构。

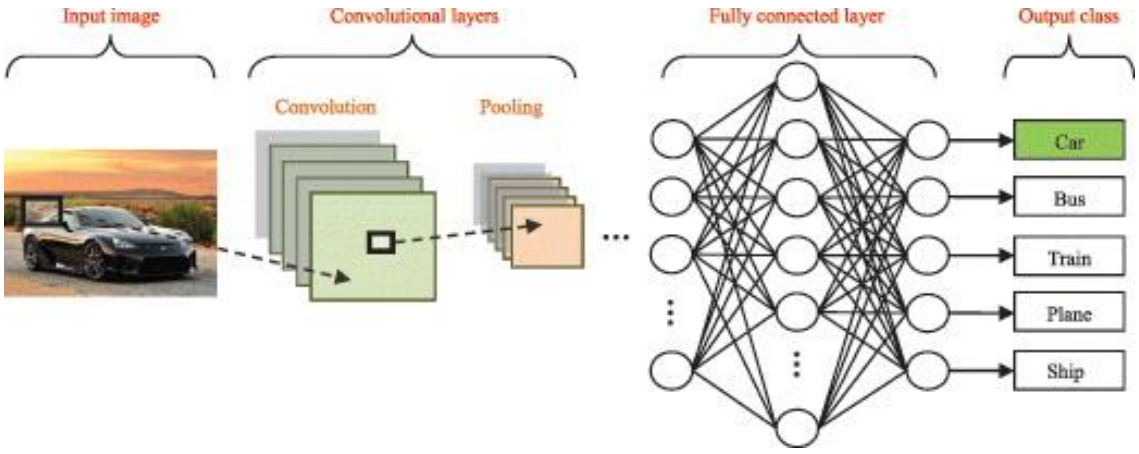


图 2-7 用于图像分类任务的卷积神经网络

Figure 2-7 CNN image classification

2.2.1 卷积层

卷积层可以学习输入图像的特征，所以卷积层一般作为特征提取器使用[54]。卷积图层中的神经元被排列成特征图。特征图中的每个神经元都有一个局部感受野[55]，它通过一组可训练的权重（也称为滤波器或者卷积核）连接到上一层某个神经元的邻域。局部感受野与已知权重进行卷积以计算新的特征映射，并通过非线性激活函数得到卷积结果。同一个特征图中的所有神经元都具有相同的权重[56]，同一卷积层内的不同特征图具有不同的权重。第 k 个输出特征图 Y_k 计算公式为：

$$Y_k = f(w_k * x) \tag{2-8}$$

x 表示输入图；w_k表示第 k 个特征图的卷积核；乘法符号表示 2 维卷积运算，用于计算输入图每个位置与卷积核的内积；f表示非线性激活函数，以前一般使用 sigmoid 函数和双曲正切函数作为激活函数，现在 ReLU 激活函数用得较多。

假设一种卷积核，可以提取一种图像的特征，比如卷积核可以提取图像的边缘特征，抑或是图像的颜色信息，当需要提取多种特征时，可以使用多种卷积核，同时对图像进行的操作，就是多核卷积[57]，对于提取多种图像特征具有很重要的作用。

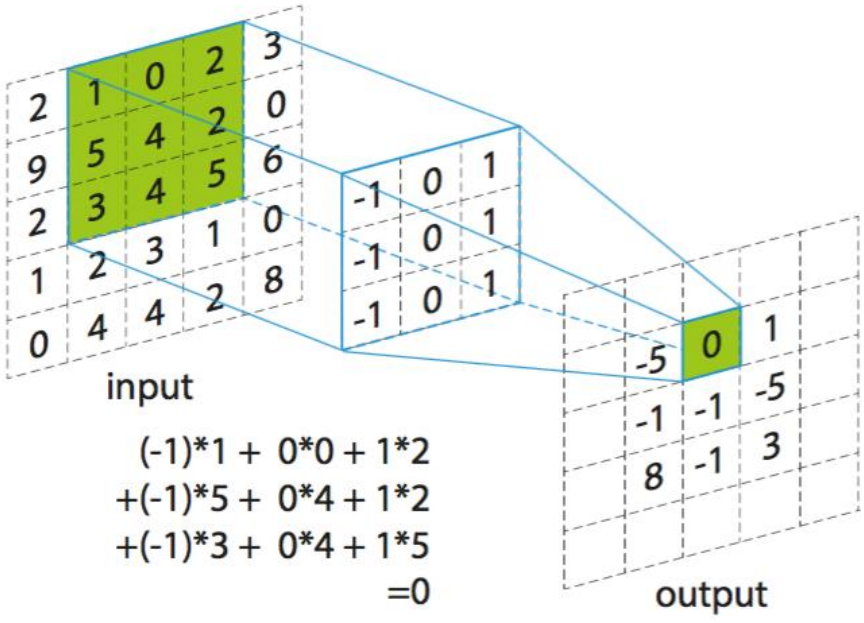


图 2-8 卷积操作示意图

Figure 2-8 Convolution operation diagram

卷积操作如图 2-8 所示，一个可学习的卷积核与输入层局部感受野进行卷积操作来提取局部特征，生成特征图，上一层可以是原始图像或者前面生成的特征图[58]。卷积操作公式为：

$$x_j^i = f\left(\sum_{i\epsilon M_j} X_i^{l-1} * \mathrm{k}_{\mathrm{ij}}^{\mathrm{l}} + \mathrm{b}_{\mathrm{j}}^{\mathrm{l}}\right) \qquad (2-9)$$

其中，l 表示上一层通道数，$\mathrm{k}_{\mathrm{ij}}^{\mathrm{l}}$表示卷积核，$M_j$表示输入层的感受野，$\mathrm{b}_{\mathrm{j}}^{\mathrm{l}}$表示偏置。偏置初始值一般为 0。

2.2.2 池化层

池化层也叫下采样层，目的是为了减少特征图的分辨率。池化操作具有一定程度上的平移和失真不变性，并有一定的防止过拟合的作用。

常见的池化操作有平均值池化、最大值池化[59]。平均值池化是将池化域输入值的平均值作为下一层的输入，能更好的保留背景；最大值池化是将池化域输入值的最大值作为下一层的输入，能更好的提取纹理。形式上，最大值池化选择每个池化域内的最大元素，如下所示：

$$Y_{kij} = \max_{(p,q)\epsilon z_{ij}} Y_{kpq} \qquad (2-10)$$

该公式表示第 k 个特征图的池化操作的输出，Y_{kpq}表示位于（p,q）处进行池化操作后得到Y_{kij}，z_{ij}表示位于（i,j）的感受野。

图 2-9 展示了平均值池化与最大值池化的操作和不同，给出一个分辨率为4×4的输入图像，选择2×2的池化操作，步幅为 2，则最大池化输出图像中每个2×2区域中的最大值，平均池化输出图像中每个2×2区域中的平均值（为四舍五入后的整数），

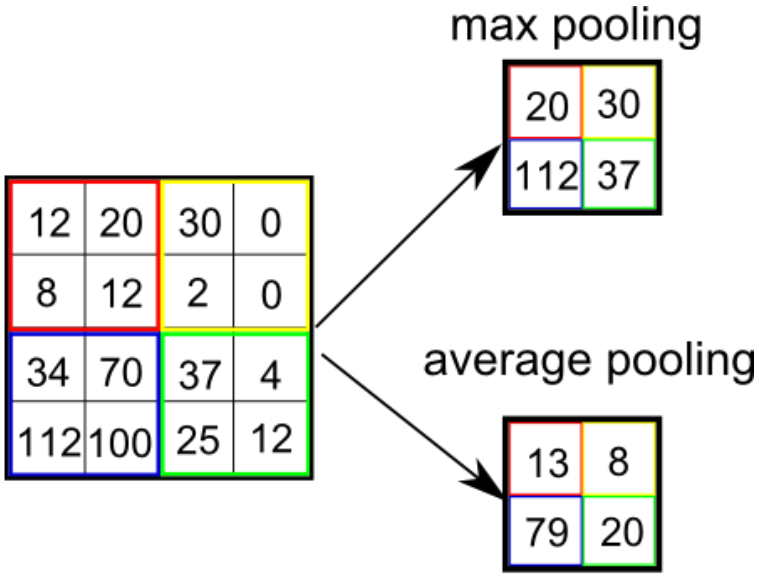


图 2-9 平均值池化与最大值池化对比

Figure 2-9 Average versus max pooling

2.2.3 激活函数

激活函数的选择会影响网络训练时间，因此激活函数的选择对大型数据集上的大型深度卷积神经网络的性能有重要影响。Krizhevsky 等人[60]对深度卷积神经网络使用 ReLU 作为激活函数，尽管 Glorot 等人[61]已经表明，他们在完全监督的网络中训练时间更快，而不需要无监督的预训练。在 CIFAR-10 数据集上训练的使用 ReLUs 激活函数的深度卷积神经网络比使用双曲正切激活的等效网络训练的速度快 6 倍[62]。

传统的激活函数，如 Sigmoid 函数和双曲正切函数如式$(2-11)$和式$(2-12)$所示：

$$\mathrm{f}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{2-11}$$

$$\mathrm{f}(x)=\tanh(x) \tag{2-12}$$

f(x)作为以 x 为输入的神经元输出的激活函数。

ReLU 是一个分段线性函数，如下式所示：

$$\mathrm{f}(\mathrm{x})=\ \max(0,\mathrm{x}) \tag{2-13}$$

ReLU 只保留激活的正面部分，将负面部分减少到零，而综合最大值算子促进更快的计算。RELU 已经在最先进的图像分类系统上使用。

从早期的人工神经网络开始，相互连接的神经节点之间通过激活函数建立起从输入到输出的映射关系。激活层主要在卷积网络中设置激活函数[63]，其本质是一种函数映射，对输入数据进行映射变换，提供网络的非线性建模能力。在运算过程中，逐元素计算，不改变原始数据的尺寸，即输入和输出的数据尺寸是相等的。

用卷积神经网络来处理图像，相当于给每一个像素点分配了一个权值，这是一种线性操作。要处理的问题不一定是线性可分的，有可能是极其复杂的非线性问题，为了解决这个问题，要引入非线性单元，这就是激活函数。

通过激活函数来为网络加入非线性单元，来解决线性模型表达力不够的问题。常用的激活函数有 Sigmoid、tanh、ReLU 等，因为神经网络的数学基础是处处可微的，因此要选取能保证数据输入与输出均可微的激活函数，学习过程是不断地进行循环的计算，所以在神经元的权重与偏置值也是一直在变化的。

激活函数（Activation Function）能够把输入的特征保留并映射下来。常见激活函数如图 2-10 所示。

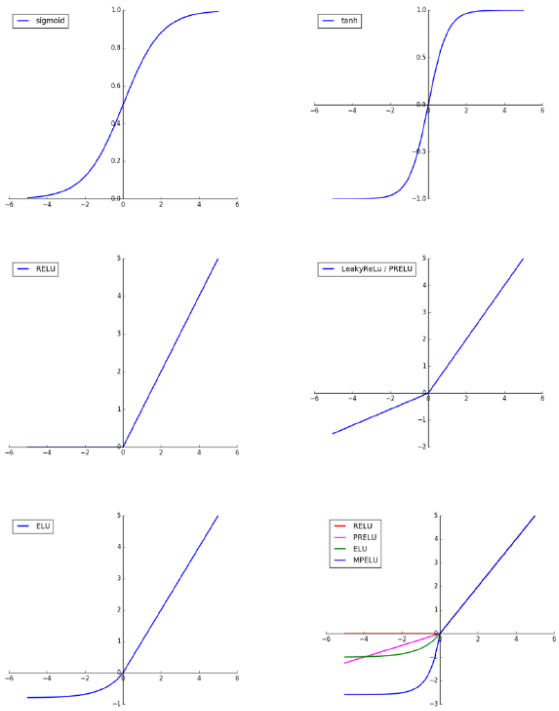


图 2-10 激活函数

Figure 2-10 Activation function

（1）sigmoid 函数 sigmoid 函数的函数是一个连续、光滑、严格单调的阈值函数。其数学公式如下：

$$f(z) = \frac{1}{1 + \exp(-z)} \qquad (2-14)$$

Sigmoid 函数有两个主要的缺点：

Sigmoid 过饱和和丢失梯度问题[64]。Sigmoid 激活函数的最大缺点就是容易饱和，当激活值在接近于 0 和 1 时，在这些地方，函数的梯度接近于 0。根据反向传播算法，梯度过小，会导致经过神经元的信号非常小。比如，当神经元初始化的是很好，初始权值过大的话，神经元会进入饱和状态，梯度过小以至于学习速率过小，甚至不能学习。

Sigmoid 的输出不是零中心的。这个特性导致梯度要么全正，要么就是全负的，假设输出全是负数，那么下一层输入就全是负数，在使用梯度下降法进行网络训练时，会呈现出 Z 字下降，因此，要使用批量梯度下降来解决这个问题。

（2）tanh 激活函数

tanh 激活函数，类似于 sigmoid 激活函数，但是进行了 0 标准化，将输入值压缩至-1 到 1 之间。但是它还是具有和 sigmoid 一样的缺点，在输入趋近于负无穷或 者正无穷大的时候，函数的激活数值接近于 0，但是它的输出是 0 中心化的。实际上，tanh 可表示为 sigmoid 的变形：

$$\tanh(x) = 2\mathrm{Sigmoid}(2x) - 1 \qquad (2-15)$$

（3）ReLU 激活函数

ReLU 是修正线性单元(The Rectified Linear Unit)的简称，ReLU 激活函数是非线性的，具有非饱和的形式，在计算梯度的时候相比于 Sigmoid 函数和 tanh 函数设计复杂的计算，ReLU 可以通过简单的阈值化的激活来实现参数稀疏。其数学表达式为：

$$f(x) = \max(0, x) \qquad (2-16)$$

使用 ReLU 后梯度下降算法的收敛速度会比 sigmoid 和 tanh 快很多，相比于 sigmoid 和 tanh，ReLU 只需要一个阈值就可以得到激活值，而不用去算一大堆复杂的运算。因此目前 ReLU 已经成为使用最多的激活函数。

ReLU 激活函数也有一定的缺点，在一个很大的梯度经过 ReLU 的时候，根据在这个梯度进行参数更新以后可能会导致这个神经元再也无法被激活，当学习率设置为比较高的值的时候，这种现象更加明显。

2.2.4 全连接层

卷积神经网络的由多个卷积层和池化层提取特征后，连接一个或者多个全连接卷积神经网络的由多个卷积层和池化层提取特征后，连接一个或者多个全连接层[65]，全连接层中的某一层的每个神经元都与相邻层的所有神经元相连接，全连接层与多层感知器结构一样。全连接层中的神经元一般使用 ReLU 函数[66]作为激活函数，ReLU 函数计算简单，可以提高全连接层的性能。使用 softmax 逻辑回归（softmax regression）对全连接层最后一层的输出结果进行分类，可以将含有 softmax 的分类层称为 softmax 层。有研究发现用支持向量机代替 softmax 算法作为分类器，可以提高分类准确度[67]。

当较深的神经网络的训练数据集时，容易出现过拟合[68]，为避免该问题，可使用正则化方法如 Dropout 技术[69]，即使隐层的神经元以一定的概率使输出为零，通过该技术使一些神经元失活，这些失活的神经元不参加前向传播也不参加反向传播[70]。这种技术能有效防止神经网络过拟合，使得神经网络学习到的特征更加具有鲁棒性。卷积神经网络的相关研究大多使用了 RelU 激活函数和 Dropout 正则化技术[71]，最后取得的效果非常好[72]。

2.3 本章小结（Summary）

神经网络技术是本课题实现目标检测的主要技术，本章介绍了卷积神经网络的基础理论，如神经元、多层感知器、卷积神经网络的结构，以及神经网络的训练方法，这些是下一步系统设计的理论基础。

3 弯道标志数据集与目标检测算法比较
3 Dataset for Traffic Sign of the Curved Roads and Comparison of Object Detection Algorithms

使用目标检测算法目的是为了识别图像中目标物体的位置和类别。在ImageNet 挑战中，基于卷积神经网络的图像识别算法就已经超过人类的水准。

目标检测算法可分为 3 类：

（1）传统图像检测算法，有基于 Boosting 框架的，如 haar、Adaboost、ACF feature 等，有基于 svmd,如 HOG+SVM、DPM 等算法。

（2）候选区域（Region Proposal）与深度学习结合的目标检测算法，即先获取候选区域，然后对候选区域使用深度学习方法分类，如：R-CNN、Fast-R-CNN、Faster-R-CNN、SPP-net、R-FCN 等算法。

（3）基于深度学习的回归算法，有：YOLO、SSD、DenseBox、RRC detection、Deformable CNN 等方法。

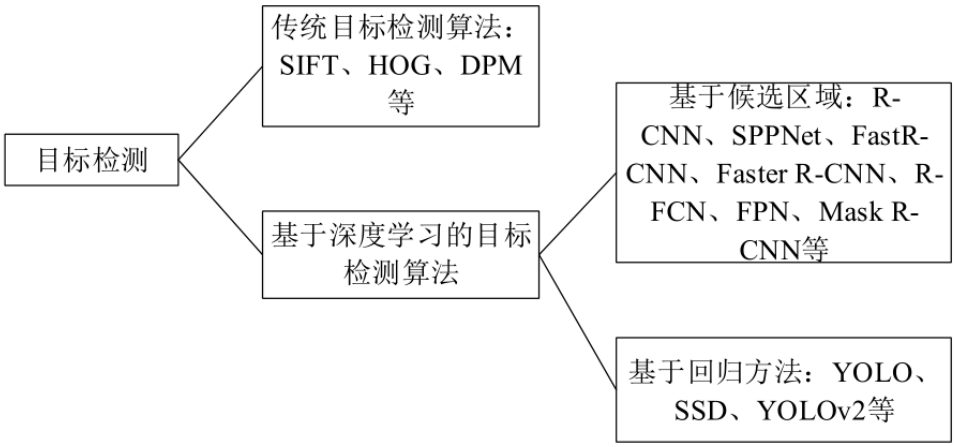


图 3-1 目标检测算法的类别
Figure 3-1 Category of object detection algorithm

深度学习技术在图像检测的目标分类方面有着广泛的运用[73]，大幅提高了目标分类的准确率[74]。Faster R-CNN 是基于候选区域（Region Proposal）的目标检测算法类别中精度较高的算法，YOLO v2 是一种检测速度快、精度高的目标检测算法，因此在本章将这两种算法应用于弯道标志目标检测并作比较。

3.1 创建弯道标志的目标检测数据集（Create Dataset for Traffic Sign Detection of the Curved Roads）

目前网络上只有几个研究机构提供交通标志数据和注释的下载，如德国交通标志识别数据集(GTSRB)、比利时交通数据集(BelgiumTS Dataset)，虽然有很多研究人员进行了中国交通标志的检测和识别，但到目前为止还没有中国交通标志数据集供研究人员下载。

反向弯道，先左弯
标签：cornerleft

反向弯道，先右弯
标签：cornerright

连续弯道
标签：continuous

向左急转弯
标签：left turn

向右急转弯
标签：right turn

图 3-2 弯道警告标志
Figure 3-2 Traffic sign on the curved roads

需收集如图 3-2 所示弯道的 5 类交通标志图片。包含连续弯道、反向弯道-先右转、反向弯道-先左转、左急转弯和右急转弯 5 类弯道交通标志。主要通过街景截图收集数据，另外通过互联网进行搜索和外出拍照等方式进行补充。因为需将图像加载到显存训练卷积神经网络，所以截图时交通标志所占获取图片的比例不能过小。共计收集 400 多张弯道标志图片。收集后，对图像进行预处理，使图片分辨率大小的一致性进行处理，训练时输入的图片分辨率长宽满足要求。收集的部分数据如图 3-3 所示。

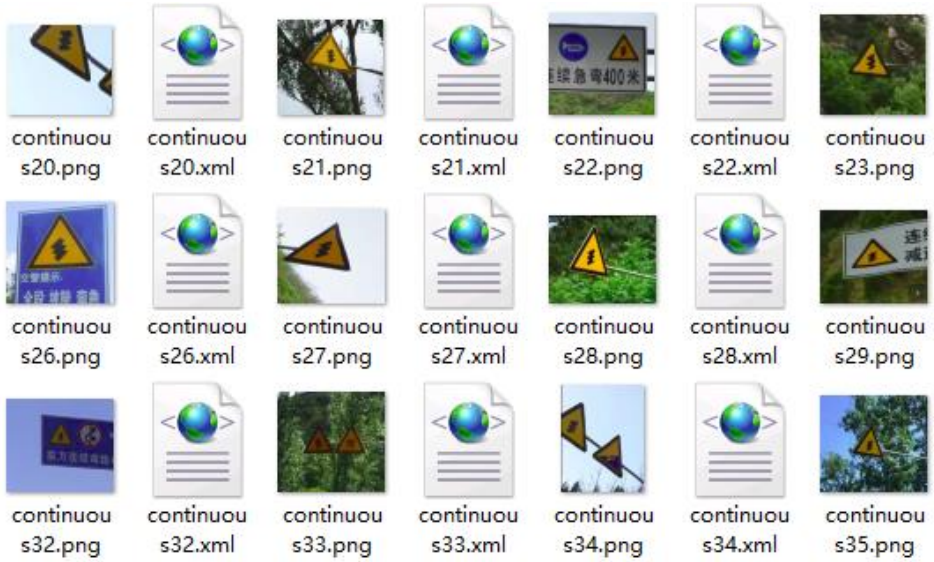


图 3-3 弯道标志数据集
Figure 3-3 Dataset for traffic sign detection of the curved roads

由于 cornersleft 类标志较少，cornersright 类和 cornersleft 类两种反向弯道标志可通过横向翻转相互转换，因此可将 cornersright 类标志横向翻转后得到 cornersleft 类标志，如图 3-4 所示。

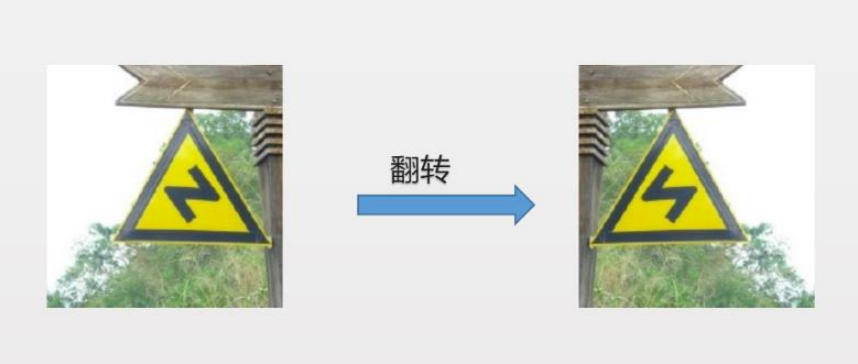


图 3-4 横向翻转图像
Figure 3-4 Flop an image horizontally

每类标志数量如表 3-1 所示：

表 3-1 弯道标志数量
Table 3-1 Number of curve signs

标签	来源	数量	含义
continuous detour	互联网搜索、街景截图、拍照	119	连续弯道
cornersleft	互联网搜索、街景截图、cornersright 类横向翻转	66	反向弯道，先左弯
cornersright	互联网搜索、街景截图	66	反向弯道，先右弯
Left turn	互联网搜索、街景截图、Right turn 类横向翻转	83	向左急转弯
Right turn	互联网搜索、街景截图	74	向右急转弯

在使用深度学习做图像识别时，采用监督学习的方式，用手工标注图像中识别目标的位置与名称等信息。使用 LabelImg 标注工具标注图片中所要识别目标的位置与类别。标注时，如图 3-5 所示，选择能完全包含目标物体的最小方框作为标注框，这样标注框与物体边缘会有部分接触。

图 3-5 图片标注工具
Figure 3-5 Image annotation tool

图片中的目标信息存储为 XML 格式文件，与 PASCAL VOC 所用格式一样，数据集里每一张图片都对应一个 XML 文件。XML 文件中存储了图片和标注信息，如图像大小、路径、名字、通道数、检测目标类别、和标注边框对角的坐标信息等。标签格式如下：

```
<annotation>
    <folder>Continuous detour</folder>
    <filename>Continuous07.png</filename>    //文件名
    <path>/home/z/桌面/data/Continuous detour/Continuous07.png</path> //文件绝对路径
    <source>
        <database>Unknown</database>
    </source>
    <size>          //图像尺寸（长宽以及通道数）
        <width>220</width>
        <height>213</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>      //是否用于分割
    <object>                  //检测到的物体
        <name>continuous detour</name>  //物体类别
        <pose>Unspecified</pose>    //拍摄角度
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>       //bounding-box,包含左下角和右上角 xy 坐标
            <xmin>40</xmin>
            <ymin>35</ymin>
            <xmax>192</xmax>
            <ymax>170</ymax>
        </bndbox>
    </object>
</annotation>
```

总共标记数据集大约 400 个图片，将每个类数据集的 80%作为训练集，该类剩下的 20%作为验证集，然后分别将训练集和验证集的图片路径和文件名分别使用 Python 脚本导入到 txt 文本文件。数据集中的图像光照条件充足、目标尺度变化大、背景多样但无遮挡。

3.2 目标检测算法评价指标(Performance Measures for Object Detection Evaluation)

根据综述的文献,一般使用平均精度均值(mean Average Precision，mAP)和交并比(Intersection Over Union，IOU)作为目标检测算法的评价指标。

分类目标分为两类，即正例（positive）和负例（negtive），根据表 3-2 得以下名词定义：

（1）True positives(TP):事实上为正例且被分类器划分为正例的实例数；

（2）False positives(FP):事实上为负例但被分类器划分为正例的实例数；

（3）False negatives(FN):事实上为正例但被分类器划分为负例的实例数；

（4）True negatives(TN):事实上为负例且被分类器划分为负例的实例数。

表 3-2 混淆矩阵

Table 3-2 Confusion matrix

	预测类别			
实际类别		正例	负例	总计
	正例	TP	FN	P(实际为正例)
	负例	FP	TN	N(实际为负例)
	总计	p′（被分为正例）	N′(被分为负例)	P+N

召回率（recall）是正例样本中预测正确的个数与实际的正例样本个数的比值，又叫做查全率，公式为：

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-1}$$

精确率（Precision）是正确被分为正例的个数和所有被分为正例的样本的比值，又叫做查准率，公式为：

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3-2}$$

将召回率和精确率合并得出一个度量F_β：

$$\mathrm{F}_\beta=\frac{(1+\beta^2)\mathrm{Recall}*\mathrm{Precision}}{\beta^2*\mathrm{Precision}+\mathrm{Recall}} \tag{3-3}$$

选取不同的β就有不同的召回率和精确率，根据这些值可以画出一条曲线，该曲线与横坐标以及纵坐标围成的面积叫平均精度(Average Precision)，分类器的平均精度越高越好。平均精度均值(mean Average Precision，mAP)就是所有类别的平均精度的平均值。平均精度均值和平均精度的计算[75]公式为：

$$\mathrm{MAP(Q)}=\frac{1}{|\mathrm{Q}|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}P\left(R_{jk}\right) \tag{3-4}$$

$$P\left(R_{jk}\right)=\frac{\mathrm{k}}{\mathrm{rank(k)}} \tag{3-5}$$

$$\mathrm{AP(j)}=\frac{1}{m_j}\sum_{k=1}^{m_j}P\left(R_{jk}\right) \tag{3-6}$$

AP(j)是第 j 类别的平均精度,Q 是所有要分类图像的集合，|Q|是所有要分类图像的个数，m_j为检索第 j 类别的个数，rank(k)检索结果中第 k 类别排名。

交并比用来测量目标检测算法的定位准确率，计算预测的边框内和真实标注的边框内交集面积和并集面积的比值就是交并比。如图 3-6 所示。

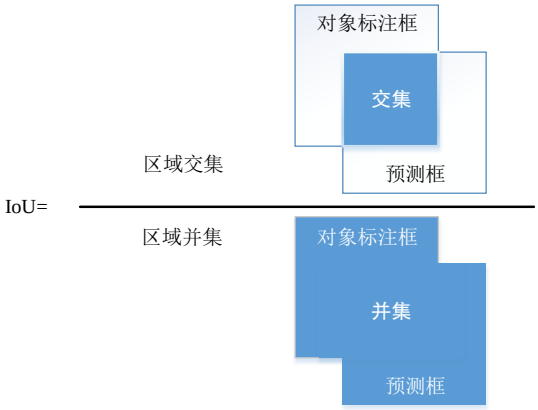


图 3-6 交并比图示
Figure 3-6 A graphical explanation of the IOU metric

3.3 基于 YOLO 的弯道标志检测(Traffic Sign Detection of the Curved Roads Based on YOLO)

3.3.1 YOLO v2 目标检测算法

YOLO 是 You Only Look Once 的英文缩写，是在 CVPR2016 提出的一种目标检测算法。该算法直接将目标检测当作一个回归问题来解决，将图像作为输入，经过一次运算便能同时得到图像中要识别物体的检测框和其所属类别。检测流程如图 3-7 所示。

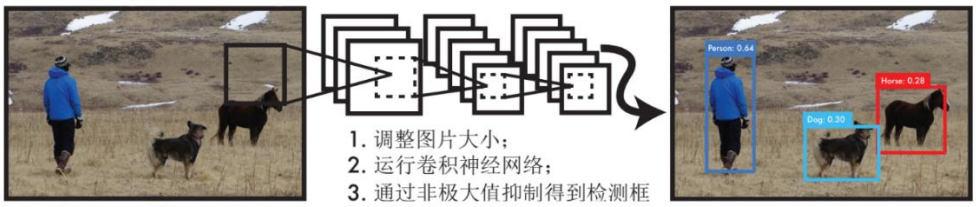


图 3-7 YOLO 检测系统流程
Figure 3-7 The processing of YOLO detection system

YOLO 将输入图像划分成 S×S 的网格，输入图像的分辨率是 448*448，是 7 的倍数，所以划分成 7×7 的网格。如果某个网格位于需要检测的目标中心，那么该网格单元负责检测需检测的目标，如图 3-8 所示。

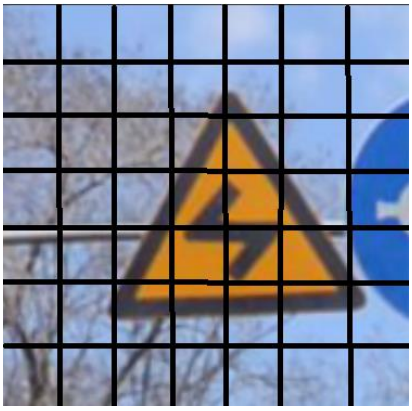


图 3-8 将图片分成 S×S 的网格
Figure 3-8 S×S grid on image

其中每个网格单元预测 B 个包含物体的预测边框和其置信度，以及预测 C 个条件概率$Pr(class_i|object)$，有 C 个类别就计算 C 次概率。

物体一般位于图片中心，因此 YOLO v2 将输入图像的分辨率从 YOLO v1 的 448*448 改成 416*416，使得特征图的输出变为奇数，从而有一个网格单元位于图像正中心预测中心的物体。416 不是 7 的倍数，是 13 的倍数，所以将输入图像划分为 13×13 的网格。每个网格只预测一类物体，图片上划分的网格数量变多，网格大小变小，提升了小物体检测效果。

YOLO v2 网络只有卷积层和池化层，使得在训练时可以动态地改变输入图像的分辨率，YOLO v2 每 10 次训练都会随机选择输入图片的分辨率，由于 downsamples 倍率为 32，输入图片的分辨率范围为{320*320，352*352，384*384，…，608*608}，该集合分辨率边长为 32 的倍数，训练时动态改变输入图片分辨率使得 YOLOv2 对不同尺寸图片具有鲁棒性。YOLOv2 可以通过改变输入图片分辨率运行，以便在速度和准确性之间进行简单折衷。

3.3.2 基于 YOLO 的弯道标志检测算法

YOLO v2 使用 Darknet-19 作为预训练分类网络。使用 Image-Net 数据集预训练，训练图片调整为 224*224 训练 160 批次，然后上调为 448*448 训练 10 批次，得到 darknet-19 里面卷积层的卷积核权值，作为预训练模型。

将分类网络改成检测网络，去掉 Darknet-19 最后一层，在每层卷积层加一个卷积核为 1×1 的卷积层，然后再插入 3 层卷积核为 3×3 的卷积层。

载入预训练模型对检测网络进行初始化，然后使用 fine tuning 技术用收集到弯道交通标志数据集进行对检测网络模型再次训练，训练框架如图 3-9 所示。

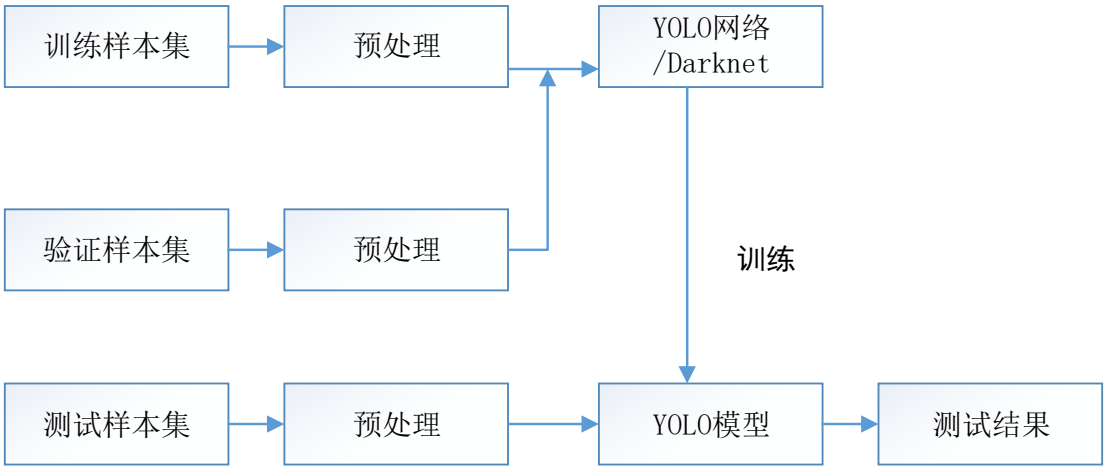


图 3-9 YOLO 网络训练框架图

Figure 3-9 YOLO network train

最后一层卷积层中卷积核的个数是 5×（类别数+5），共有 5 类交通标志，因此将最后一层卷积核有 50 个，输入图片划分为 13×13 的网格，因此最后一层特征图为 13×13×50。

基于 YOLO v2 的交通标志检测网络结构图如图 3-10 所示：

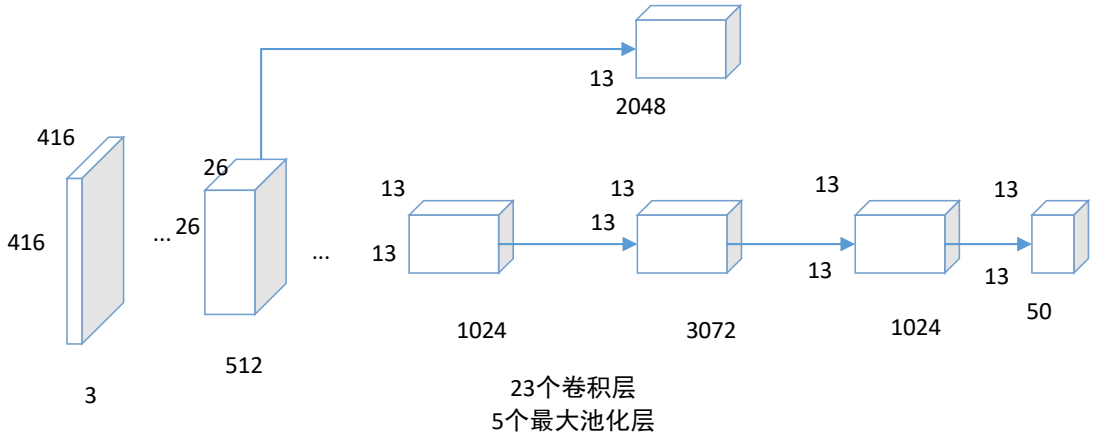


图 3-10 YOLO v2 结构图
Figure 3-10 The Architecture of YOLO v2

根据 YOLO 和 YOLO v2 算法原理，得到基于 YOLO 的弯道标志检测步骤如下：

步骤 1：将图片调整为 416×416。若图片长宽不相等，则将图片填充至长宽相等，然后缩小或放大调整为 416×416，调整结果如图 3-11 所示；

图 3-11 调整图片
Figure 3-11 Resize image

步骤 2：将图片划分为 13×13 的网格，判断物体中心是否落在哪些网格，并对包含物体中心的网格进行检测该物体；

步骤 3：输出预测张量，使用概率阈值（thresh）对检测框进行筛选，生成检测框。

基于 YOLO 的弯道标志检测流程如图 3-12 所示。

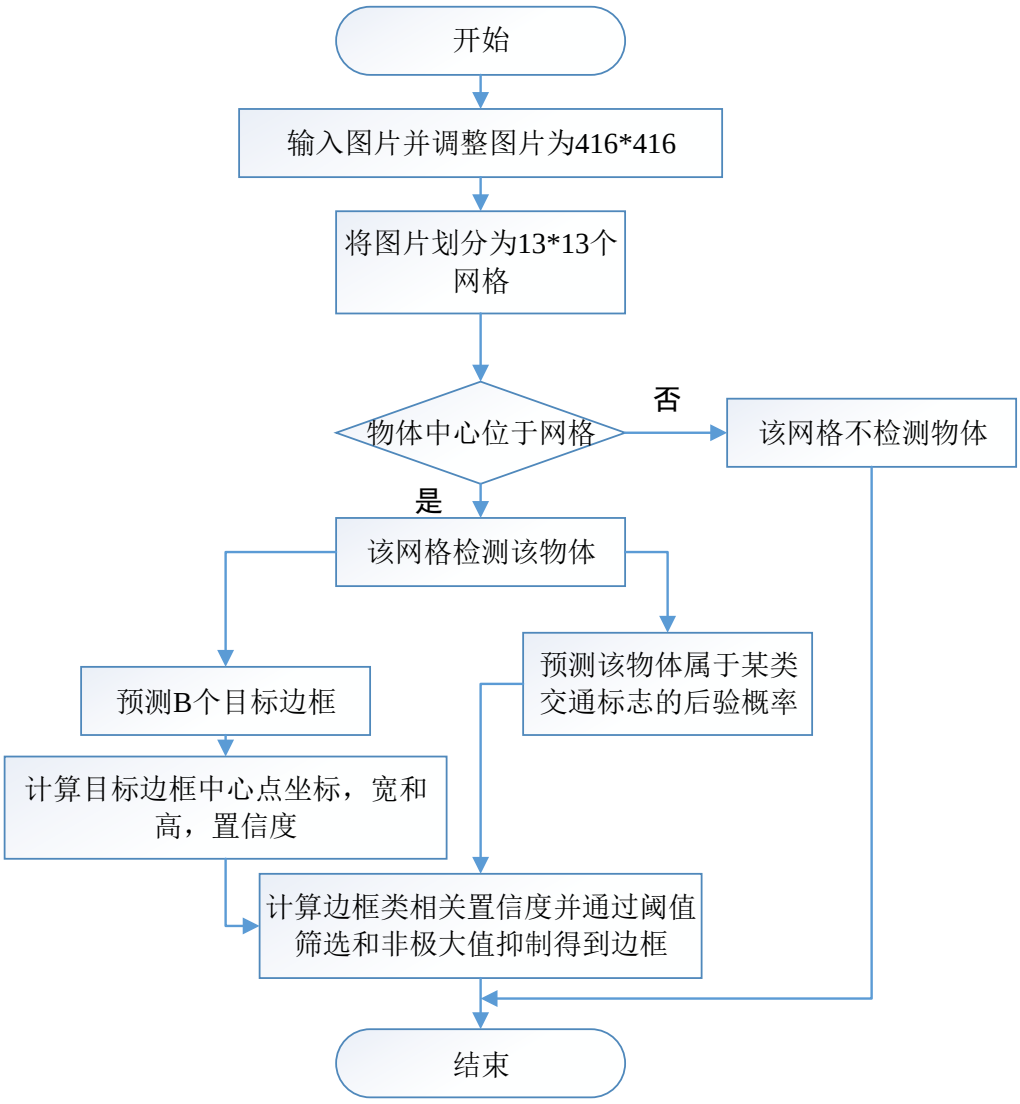


图 3-12 检测流程图

Figure 3-12 Flowchart of detection

输入图像分辨率为416×416，划分为SxS网格，416是13的倍数，所以S=13。每个预测边框包含5个预测值，总共预测C个类别，共有5类交通标志，因此C=5，网络输出为S×S×(B×5+C)的预测张量，即13×13×(B×5+5)。

3.3.3 实验步骤与结果分析

本节按上节提出弯道标志检测算法，进行检测实验，实验环境如下：

硬件配置：NVIDIA GTX960m显卡，Intel Pentium G4400处理器，8GB内存。

软件环境：Ubuntu16.04系统，OpenCV 2.4.9，cuda 8.0，Python 3.5，GCC，Darknet深度学习框架。

将弯道数据集每个类中的80%的图片划分为训练集，剩下的图片划为验证集，使用Python脚本将训练集和验证集的图片的绝对路径导入到文本文档中，如图3-13和图3-14所示。

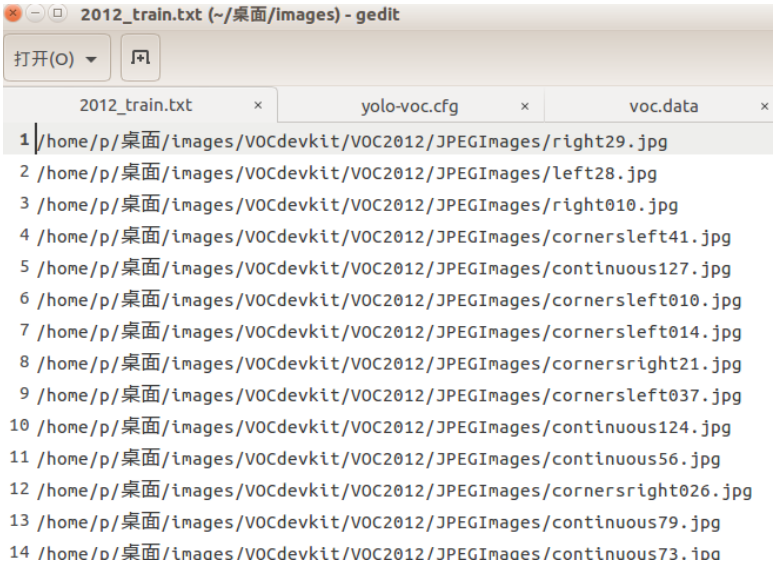


图 3-13 训练集

Figure 3-13 Training dataset

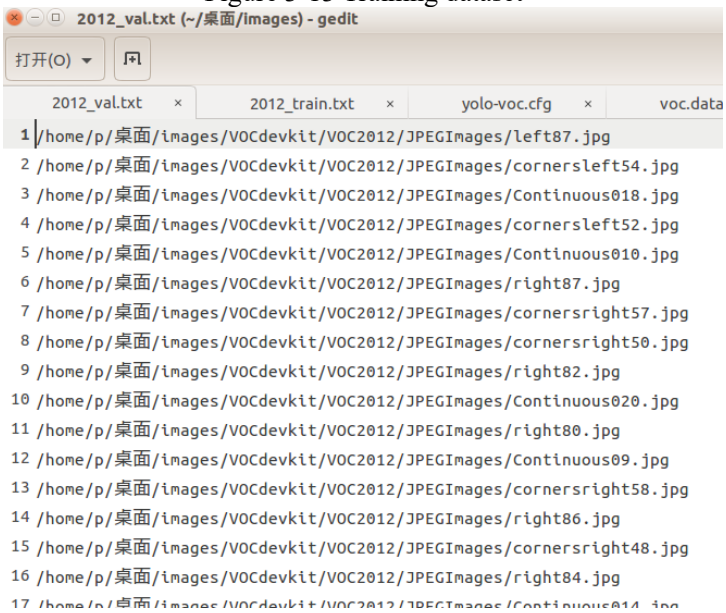


图 3-14 验证集

Figure 3-14 Validation dataset

将 5 类弯道标志的标签每行一个存储到 obj.names 文件中，放入 darknet/cfg 目录下，如图 3-15 所示。

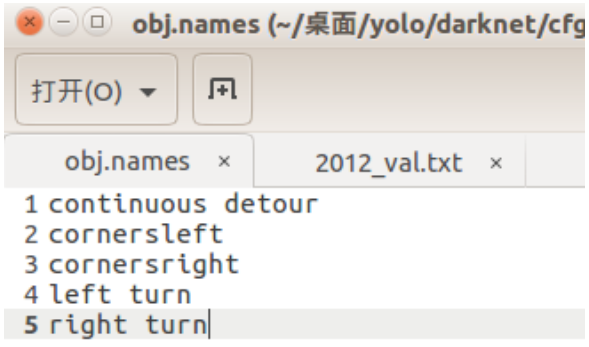


图 3-15 obj.names 文件

Figure 3-15 obj.names file

将包含类别信息文件的路径、包含验证集路径文件的绝对路径、类别数等信息存储到 voc.data 文件，并放入 darknet/cfg 目录下，voc.data 存储信息如图 3-16 所示。

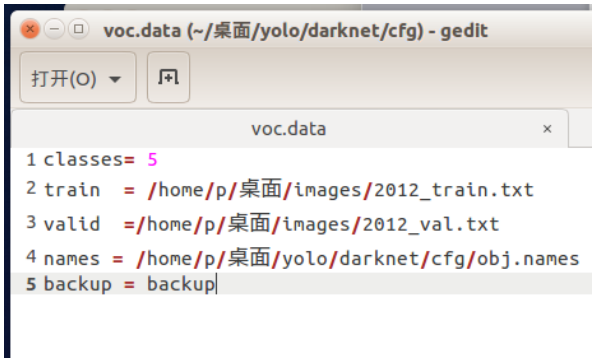


图 3-16 voc.data 文件
Figure 3-16 voc.data file

使用 Python 脚本将训练集图片的标签转换成 Darknet 需要的格式，文件存储了是类的索引、图片中目标类的位置信息，格式示例如图 3-17 所示。

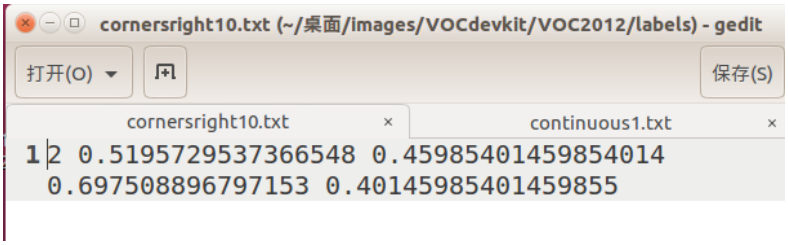


图 3-17 Darknet 标签文件
Figure 3-17 Darknet label file

因为显存只有 4GB，所以将每次迭代训练的图片数量 batch 设为 16，最大迭代次数 Max_batches 设为 80200。最后一层的滤波器的数量为（类别+5）*5，因此将最后一层的 filters 参数设置为 50。

使用 Imagenet 训练网络得到预训练模型，载入该预训练模型并对检测网络模型使用 fine turning 技术进行训练直到最大迭代次数，得到最终的权值文件。

重新收集 20 张不属于数据集的弯道交通标志图片，对基于 YOLO 的弯道标志检测系统进行测试，正确检测出分类的图片共有 17 张，1 张未检出，2 张错误分类。测试结果示例如图 3-18 所示。

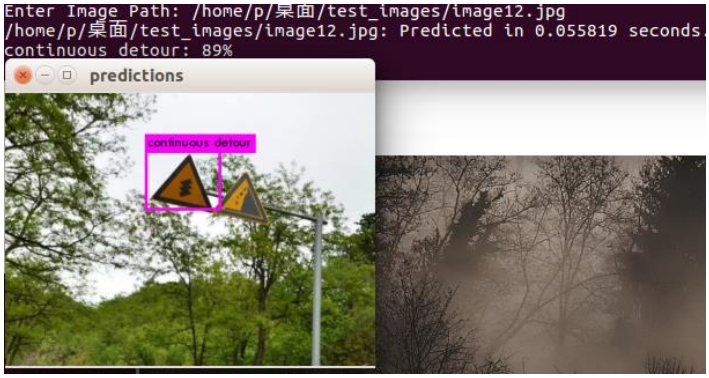


图 3-18 测试结果
Figure 3-18 Test Results

在控制台计算验证集中每个图片的召回率和交并比，所得结果如图 3-19 和表 3-3 所示。经统计得所以交并比平均值为 87.48%，交并比在 50%就可以接受，因此结果比较良好。召回率基本上为 100%,说明对弯道标志的分类结果较准确。

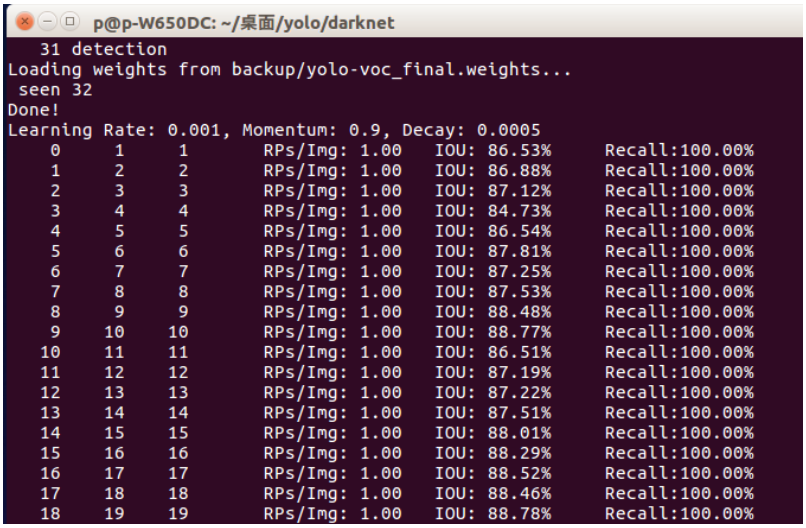


图 3-19 召回率和交并比

Figure 3-19 Recall and IOU

表 3-3 统计结果

Table 3-3 Statistical result

召回率平均值	100%
交并比平均值	87.48%
交并比方差	0.9699

计算每个类的精度均值，所得计算结果如图 3-20 和表 3-4 所示。从表中可以看出连续弯道警告标志（continuous detour）类精度均值最高，向左急转弯警告标志（left turn）和向右急转弯警告标志（right turn）其次，先左弯反向弯道（cornersleft）和先右弯反向弯道（cornersright）最低。

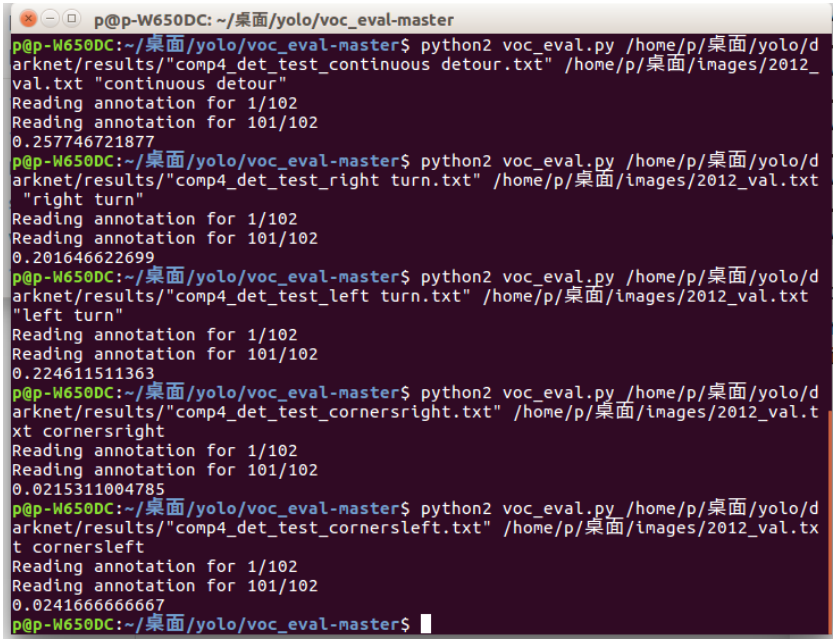


图 3-20 每个类的精度均值

Figure 3-20 AP of each class

表 3-4 精度均值
Table 3-4 Average Precision

类别	AP 值
continuous detour	0.2577
cornersleft	0.0215
cornersright	0.0241
left turn	0.2246
right turn	0.2016
mAP	0.1464

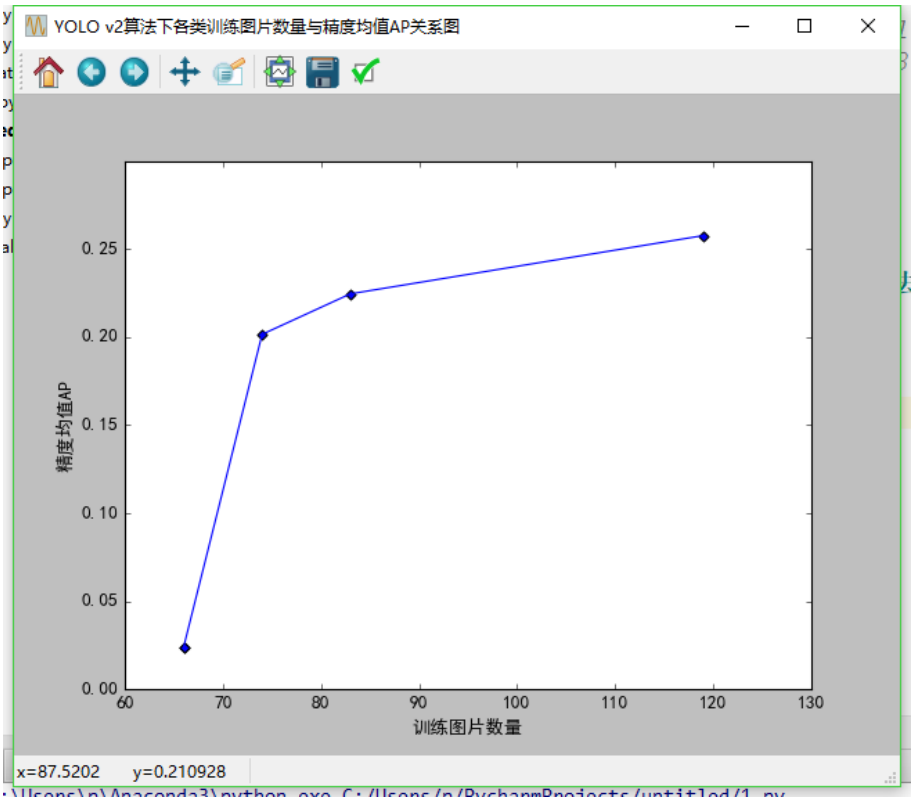


图 3-21 训练集数量与精度均值关系
Figure 3-21 The relationship between the number of training sets and AP

使用折线图表示各个弯道标志类的数量与该类的精度均值关系，如图 3-21 所示。由图 3-21 可得某类包含的训练图片越多，则该类精度均值越高，因此可以通过增加训练集来提高弯道标志检测模型精度。

3.4 基于 Faster R-CNN 的弯道标志检测(Traffic Sign Detection of the Curved Roads Based on Faster R-CNN)

3.4.1 Faster R-CNN 目标检测算法

2015 年，微软的研究团队（Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun）提出一种 Faster R-CNN 的方法，使得 region proposal 非常高效。

Faster R-CNN 使用卷积神经网络一次性计算整幅图像的特征，重用这些相同的卷积神经网络特征进行提出建议区域（region proposals），从而取代单独的

selective search 算法，如图 3-22 所示。

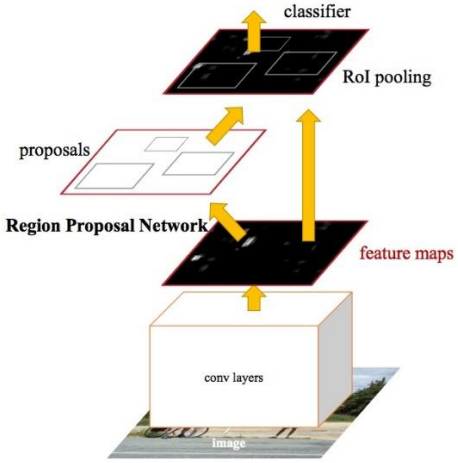


图 3-22 Faster R-CNN 基本结构

Figure 3-22 The structure of Faster R-CNN

用于检测区域的卷积特征图（类似于 Fast R-CNN）同样用来生成 region proposals。

模型的输入和输出如下：

输入：图像（不需要 region proposals）

输出：图像中物体的类别及其对应的 bounding box 坐标

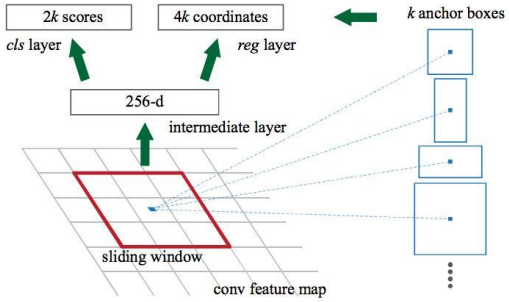


图 3-23 建议区域网络

Figure 3-23 Region Proposal Network

Region Proposal Network 在卷积神经网络的特征图上进行滑动窗口操作，如图 3-23 所示，对于每个窗口的位置，网络会对每个候选区域（anchor）输出分数和 bounding box（共 4*k 个 box 坐标，k 是 anchor 的个数）

直觉上认为图像中的物体应该符合一定的大小和长宽比，比如行人的形状更可能是长方形的 box，但一般也不会关注那些非常窄的 boxes。据此，构建了 k 个这样的 common aspect ratios，并称作 anchor boxes。每个 anchor box 输出 bounding box 和对应的位置图像的分数。

Region Proposal Network 的输入和输出如下：

输入：CNN 特征图

输出：每个 anchor 对应的 bounding box 以及分数（表示 bounding box 中对应图像部分是物体的可能性）

将 Region Proposal Network 中输出的目标可能 bounding box 输入到 Fast R-CNN，并生成类别和 tightened bounding box。如果设计具有实用价值的弯道标志检测器，除了考虑识别目标的准确性，还需要考虑到检测器运行的实时性。

3.4.2 基于 Faster R-CNN 的弯道标志检测算法

训练的 Faster R-CNN 网络结构如图 3-24 所示，可以将其看作 RPN 网络和 Fast R-CNN 网络的组合。

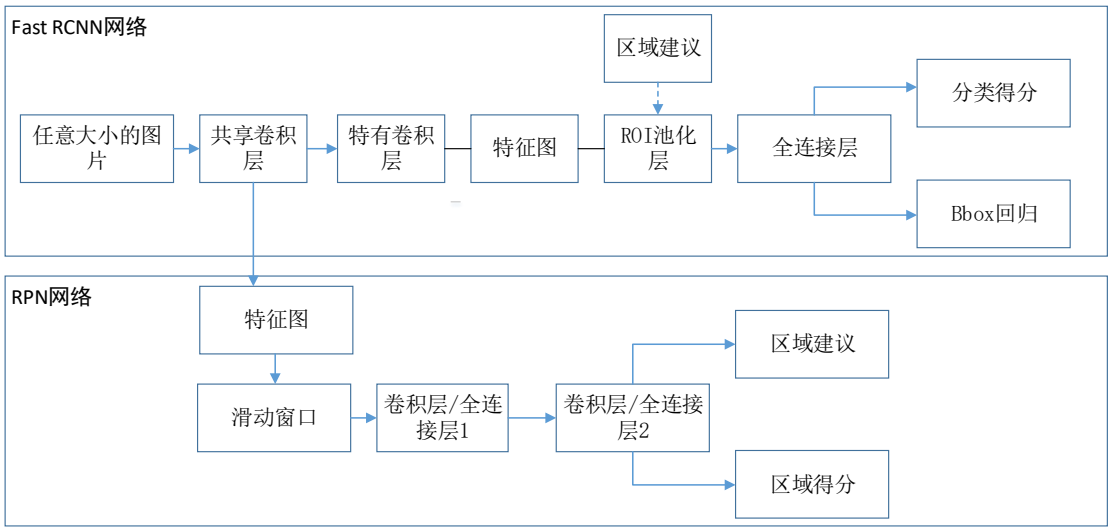


图 3-24 Faster R-CNN 网络结构

Figure 3-24 Network structure of Faster R-CNN

算法具体步骤如下：

步骤 1：将图片作为卷积神经网络的输入；

步骤 2：经过卷积神经网络前向传播到共享卷积层，得到的特征图作为 RPN 网络的输入，然后继续前向传播至特有卷积层，产生更高维特征图；

步骤 3：共享的卷积层输出的特征图经过 RPN 网络得到区域建议和区域得分，使用非极大值抑制筛选出区域得分较高的区域建议；

步骤 4：将高维特征图和 RPN 网络输出的区域建议作为 RoI 池化层的输入，提取区域建议的特征，通过全连接层后，输出该区域的分类得分以及回归后的 bounding-box。

3.4.3 实验步骤与结果分析

本节按上节提出弯道标志检测算法，进行检测实验，实验环境如下：

硬件配置：NVIDIA GTX960m 显卡，4GB 显存，Intel Pentium G4400 处理器，8GB 内存。

软件环境：Ubuntu16.04 系统，Windows 10 系统，OpenCV 2.4.9，cuda 8.0，Python 3.5，TensorFlow 1.3.0 深度学习框架。

在实验中，将弯道数据集每个类中 80%的图片划分为训练集，剩下的 80%图

片作为验证集。接下来使用 Python 脚本将数据集包含的图片对应的 XML 文件的部分信息导出到 2 个 csv 格式的文件，一个 csv 文件包含训练集，另一个包含测试集。csv 文件储存的信息包含图片名、图片长和宽、所需识别目标的标注框位置，如图 3-25 所示。

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	continuous	174	170	continuous	95	71	132	108
3	cornersleft	258	232	cornersleft	40	40	239	160
4	continuous	189	156	continuous	35	18	107	112
5	right37.pn	270	238	right turn	55	32	157	114
6	right65.pn	184	168	right turn	66	23	183	107
7	continuous	191	192	continuous	58	15	175	129
8	Continuou	213	213	continuous	27	23	181	170

图 3-25 csv 文件

Figure 3-25 csv format file

根据 csv 文件里的信息将图片数据和标签转换成 2 个 TFRecord 格式文件，一个 TFRecord 文件包含训练集，另一个包含测试集。将训练类别信息存储到 object-detect.pbtxt 文件里，如图 3-26 所示。

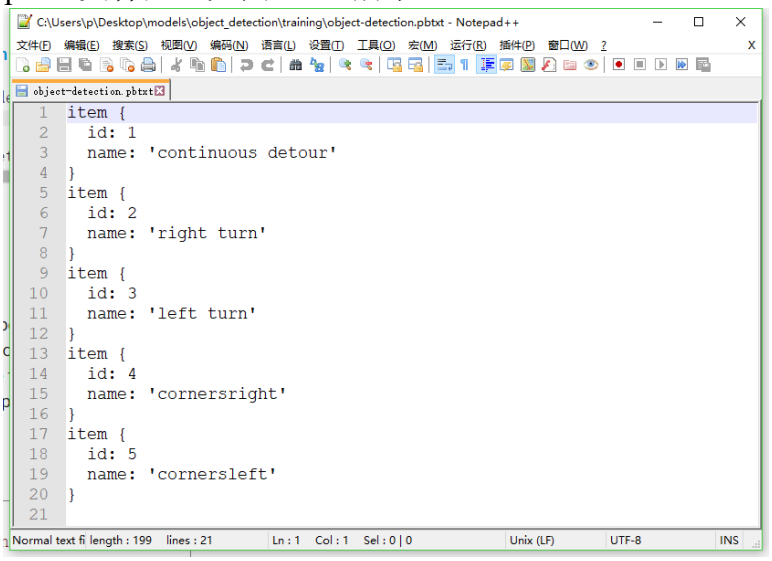


图 3-26 object-detect.pbtxt 文件

Figure 3-26 object-detect.pbtxt file

将分类参数 num_classes 设为 5，总训练步数 num_steps 设为 20000，重要网络参数设置如表 3-4 所示。

表 3-5 参数配置

Table 3-5 Parameter configuration

参数	值
batch_size	1
classes	5
num_steps	20000
initial_learning_rate	0.0003
9000 step learning_rate	0.00003

载入 COCO 数据集和 Kitti 数据集训练好的预训练模型对检测网络进行初始化，然后使用 fine tuning 技术用收集到弯道交通标志数据集对模型再次训练，得

到最终的权值文件。

使用 3.3.3 节收集的 20 张不属于数据集的弯道交通标志图片，对基于 Faster R-CNN 的弯道标志检测系统进行测试，检测结果如表 3-6 所示。

表 3-6 测试结果

Table 3-6 Test Results

正确分类张数	未检出张数	错误分类张数
7 张	7 张	6 张

部分测试结果如图 3-27 所示。

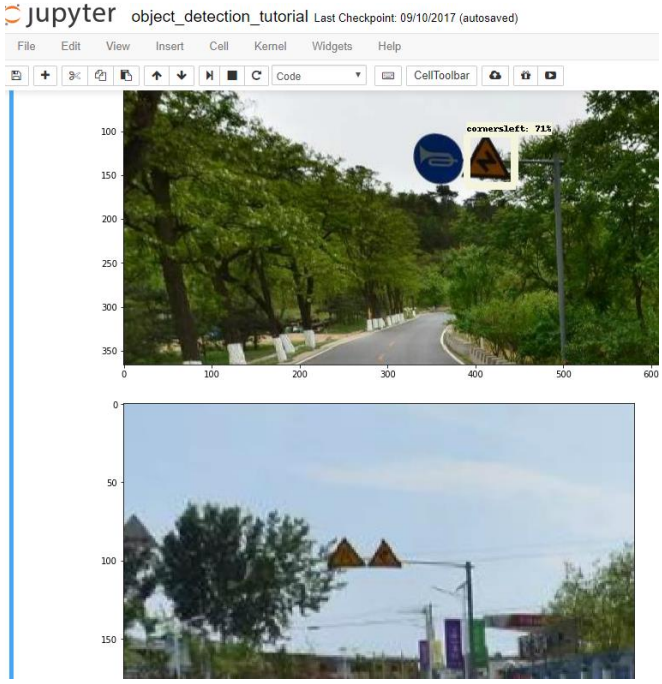


图 3-27 Faster R-CNN 算法的部分检测结果

Figure 3-27 Some detection results of Faster R-CNN algorithm

部分错误分类结果如图 3-28 所示。

图 3-28 Faster R-CNN 算法的部分分类错误结果

Figure 3-28 Partial error test results of Faster R-CNN algorithm

3.5 YOLO v2 与 Faster R-CNN 比较(Comparision of YOLO v2 and Faser R-CNN)

3.2.3 节收集的 20 张不属于数据集的弯道交通标志图片中，有 8 张图片中的弯道交通标志占面积较小，人眼看起来较模糊。使用这些图片对 YOLO v2 算法和 Faster R-CNN 进行测试。

表 3-7 不同的交通标志占图片面积比的检测结果

Table 3-7 Detection results under different area ratio

	8 张弯道标志占比小的图片		12 张弯道标志占比正常测试图	
	正确分类张数	召回率	正确分类张数	召回率
YOLO v2	6 张	75%	11 张	91%
Faster R-CNN	0 张	0%	7 张	58%

由表 3-7 可得，使用 YOLO v2 算法和 Faster R-CNN 算法检测图片，图片中弯道标志占比大的召回率高，YOLO v2 算法比 Faster R-CNN 算法在标志占比大的图片和标志占比小的图片识别率高，是因为 TensorFlow 实现的 Faster R-CNN 算法写得不好，速度和效果没有原作者使用 C 语言实现的 YOLO v2 算法高。

20 张弯道交通标志图片中有 7 张图片中的标志周围以天空为背景，背景干净；其他图片的标志后面是树木或者山丘，背景复杂。

表 3-8 不同背景下的检测结果

Table 3-8 Detection results under different background

	对背景干净图片分类正确数	对背景干净图片分类召回率	对背景复杂图片分类正确数	对背景复杂图片分类召回率
YOLO v2	6 张	85.7%	11 张	84.6%
Faster R-CNN	2 张	28.5%	5 张	38.4%

2 类图片分类正确数量和召回率如表 3-8 所示，各种算法中 2 类图片的召回率相差不大，因此背景对分类正确率的影响不大。训练集中包含弯道标志的背景复杂的图片和背景是天空的图片大概是 3:2，2 种背景的训练样本数量差不多，所以背景对分类正确率的影响不大。

使用 TensorFlow Object Detection API 实现的检测器在摄像头上每秒大概检测 2 到 5 帧图像。基于 YOLO v2 算法的弯道标志检测器在摄像头上每秒大概检测 10 到 15 帧图像。两种算法检测检测速度的比较如表 3-9 所示。

表 3-9 目标检测速度比较

Table 3-9 Comparison of object detection speed

	摄像头上每秒检测帧数	检测 1 张图片所需时间	检测 20 张图片所需时间
YOLO v2	10-15 帧	0.056 秒	1.09 秒
Faster R-CNN	2-5 帧	0.181 秒	3.53 秒

由表 3-9 可得，YOLO v2 算法比 Faster R-CNN 算法每秒检测帧数多，检测相同图片所花时间少。TensorFlow 实现的 Faster R-CNN 算法使用的是 Python 语言，YOLO v2 算法是用 C 语言实现的，Python 语言是解释型语言，执行时才编

译，比 C 语言这种运行前编译好的语言运行速度慢，因此 YOLO v2 算法检测速度快。

由上知 YOLO v2 算法比 Faster R-CNN 算法更准确快速，因此选择基于 YOLO v2 算法来改进并实现弯道标志检测系统。

3.6 本章小结（Summary）

本章建立了弯道交通标志的目标检测数据集，介绍 YOLO 系列目标检测算法和 Faster R-CNN 目标检测算法，并将 YOLO v2 算法和 Faster R-CNN 算法应用于弯道标志检测，然后实现并比较测试结果。TensorFlow 实现的 Faster R-CNN 算法在弯道交通标志数据集上的检测效果不理想，没有 YOLO 算法使用的 C 语言运行速度快，因此接下来选择表现优秀的 YOLO 算法进行下一步研究。

4 弯道标志检测方法的设计与实现
4 The Realization of object detection system

4.1 修改 YOLO v2 网络结构（Modify the Network Structure of YOLO v2）

卷积神经网络的网络层数越多，提取的特征越高度抽象，从而使得分类准确率更高，可尝试通过增加网络的层数来提高检测网络精度。但网络层数太多就会很难训练，网络层数越多，梯度消失的问题就愈加明显，可能会出现梯度消失和梯度爆炸问题，本课题主要目的是对图片上的弯道标志进行分类，可以适当增加网络层数来提高分类准确率。

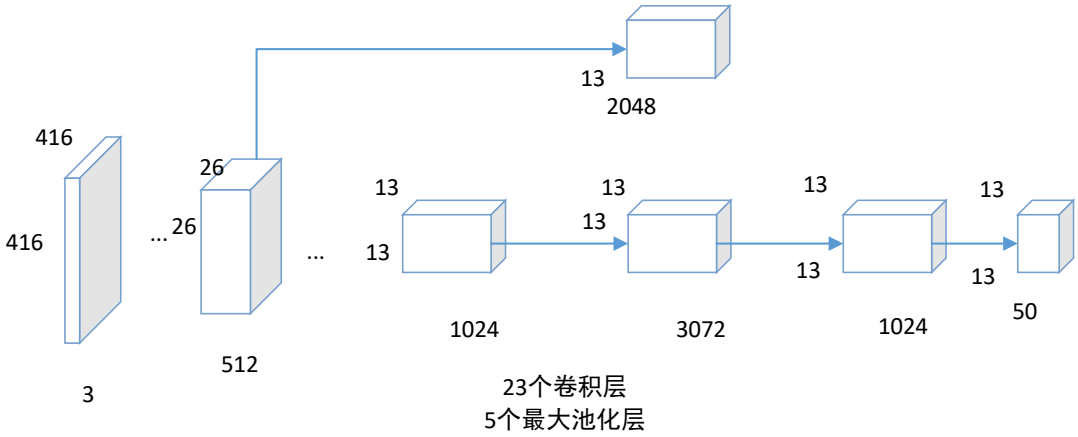


图 4-1 YOLO v2 网络模型
Figure 4-1 The Architecture of YOLO v2

原 YOLO v2 网络如图 4-1 所示，经多次实验决定往 YOLO v2 网络插入 2 个卷积核为1×1的卷积层、2 个卷积核为3×3的卷积层和 1 个最大值池化层得到新的网络模型，新网络模型如图 4-2 所示。

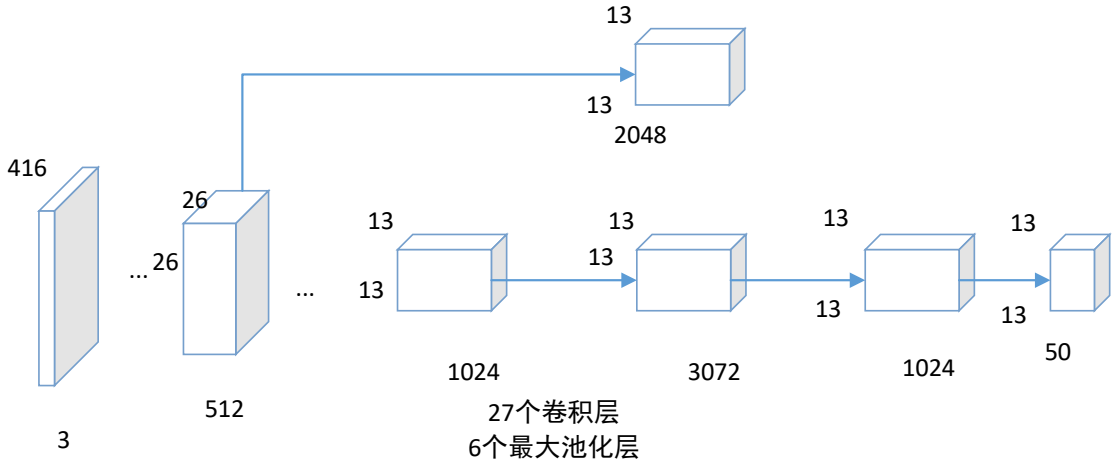


图 4-2 改进后的模型
Figure 4-2 An improved model

改进后的 YOLO v2 算法流程如图 4-3 所示。

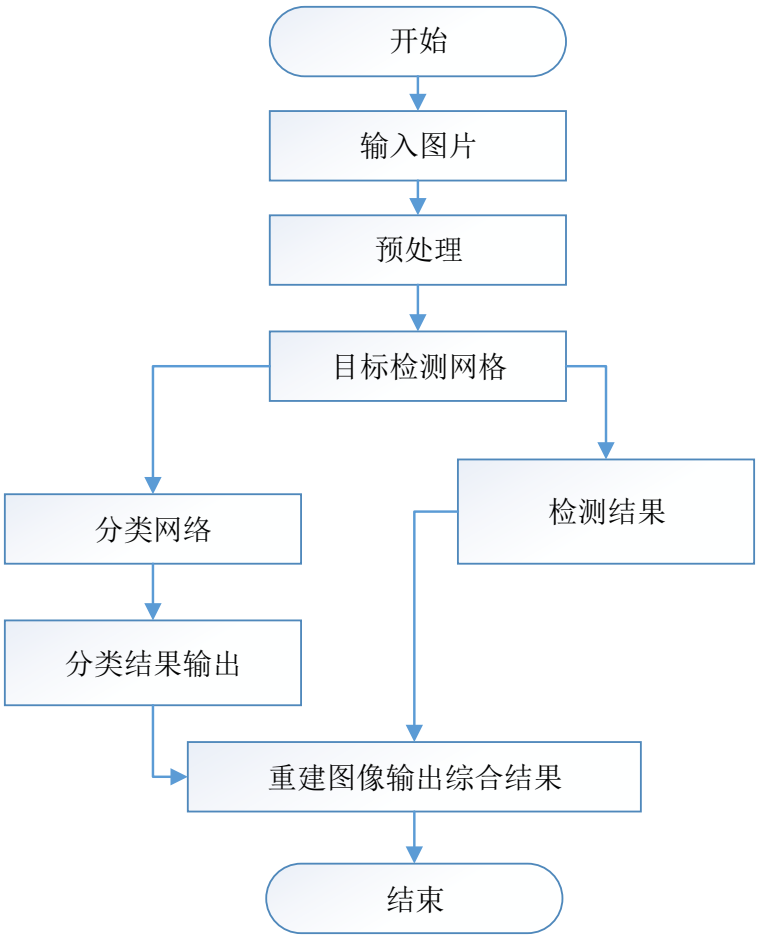


图 4-3 算法流程

Figure 4-3 Flow chart of the proposed algorithm

4.2 实验步骤与结果分析(Experimental Steps and Results Analysis)

硬件配置如下：

显卡：NVIDIA GTX960m，显存 4GB;

处理器：Intel Pentium G4400;

内存大小：8GB。

软件环境如下：

编译环境：Ubuntu16.04 系统，OpenCV 2.4.9，cuda，Python 3.5，GCC;

深度学习框架：Darknet。

使用 cuda 实现 GPU 训练网络，OpenCV 用于测试时能显示图片和使用摄像头测试。

因为显存不大，那么每次迭代训练的图片数量不能设太大，多次尝试后将 batch 设为 16，最大迭代次数 Max_batches 设为 80200。各类重要网络参数设置如图 4-4 和表 4-1 所示。

图 4-4 YOLO v2 参数设置

Figure 4-4 Parameter settings of YOLO v2

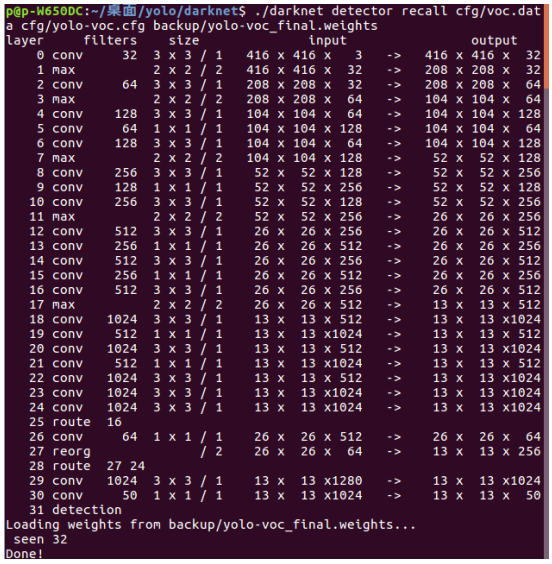
表 4-1 参数配置

Table 4-1 Parameter configuration

参数	值	设置原因
batch	16	显存较小
classes	5	类别数
Max_batches	80200	训练次数适当大一点
Learning rate	0.001	初始学习率可设置大一点
momentum	0.9	为起到加速收敛的作用，建议设置为 0.9
weight_decay	0.0005	防止过拟合

因为数据集只有四百多张，不算太多，如果从头开始训练检测网络，容易产生过拟合的现象，并且特征提取泛化能力不强，训练耗时也较长。因此首先使用 ImageNet 数据集进行模型预训练，接下来载入训练 Imagenet 得到预训练模型对检测网络进行初始化，然后使用 fine tuning 技术，用收集到弯道交通标志数据集，对检测网络模型再次训练。

原 YOLOv2 模型载入与改进后 YOLO v2 模型载入如图 4-5 所示。



a. YOLO v2

b. 改进后的 YOLO v2

图 4-5 YOLO v2 网络结构

Figure 4-5 The Architecture of YOLO v2

完成训练后得到最终权重，输入测试图片得到测试结果，如图 4-6 所示。

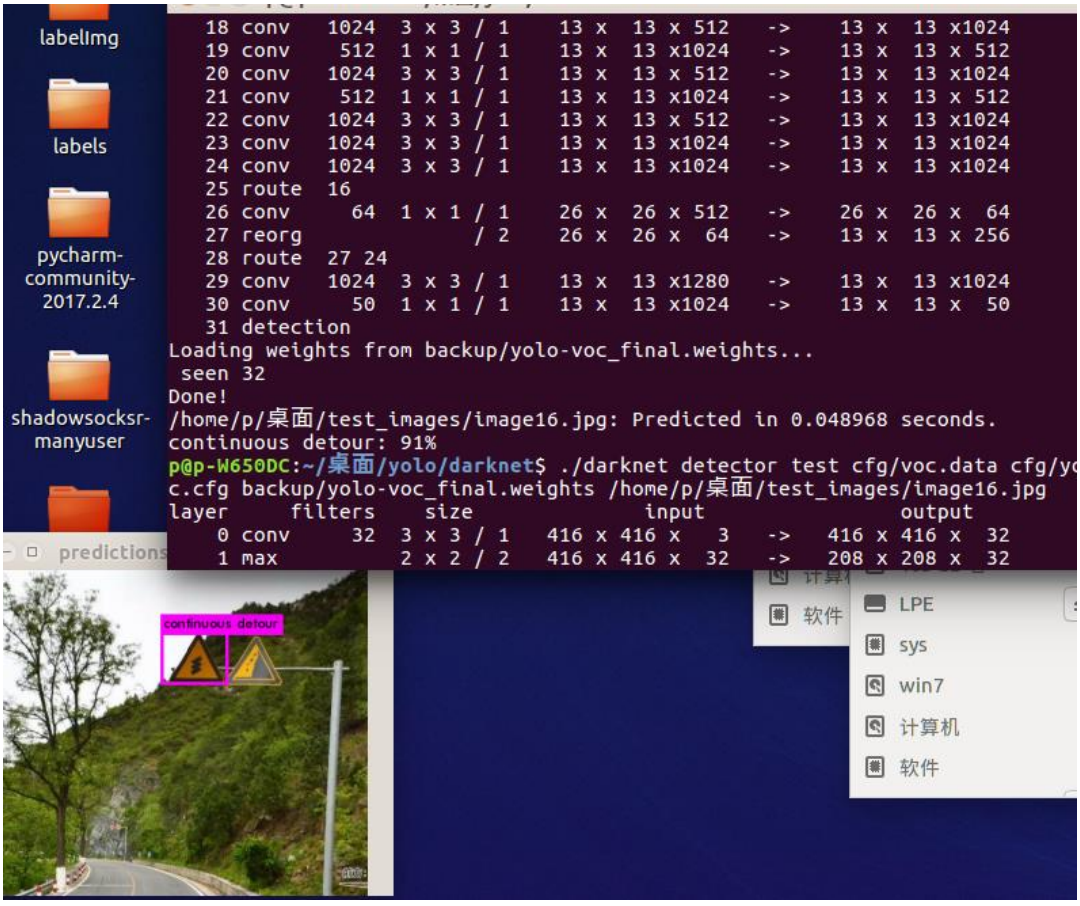


图 4-6 图片测试

Figure 4-6 Picture test results

接下来检测摄像头前的弯道标志，将弯道标志图片放在摄像头前，测试弯道标志检测系统性能，如图 4-7 所示：

a．单一背景

b．多个标志

图 4-7 使用摄像头识别

Figure 4-7 Camera test results

经统计后得出，每秒识别帧率为 10 到 15 帧，达到实时性要求。计算改进后 YOLO v2 算法的验证集中每个图片的召回率、IOU 和精准率，任何根据这些值计算改进后 YOLO v2 算法每个类 AP 与平均精度均值，结果如图 4-8 所示。

```
p@p-W650DC: ~/桌面/voc_eval-master
p@p-W650DC:~/桌面/voc_eval-master$ python2 voc_eval.py /home/p/桌面/yolo/darknet
/results/"comp4_det_test_continuous detour.txt" /home/p/桌面/images/2012_val.txt
 "continuous detour"
Reading annotation for 1/102
Reading annotation for 101/102
0.296340684139
p@p-W650DC:~/桌面/voc_eval-master$ python2 voc_eval.py /home/p/桌面/yolo/darknet
/results/comp4_det_test_cornersleft.txt /home/p/桌面/images/2012_val.txt corners
left
Reading annotation for 1/102
Reading annotation for 101/102
0.0244565217391
p@p-W650DC:~/桌面/voc_eval-master$ python2 voc_eval.py /home/p/桌面/yolo/darknet
/results/comp4_det_test_cornersright.txt /home/p/桌面/images/2012_val.txt corner
sright
Reading annotation for 1/102
Reading annotation for 101/102
0.0656565656566
p@p-W650DC:~/桌面/voc_eval-master$ python2 voc_eval.py /home/p/桌面/yolo/darknet
/results/"comp4_det_test_left turn.txt" /home/p/桌面/images/2012_val.txt "left t
urn"
Reading annotation for 1/102
Reading annotation for 101/102
0.147773279352
p@p-W650DC:~/桌面/voc_eval-master$ python2 voc_eval.py /home/p/桌面/yolo/darknet
/results/"comp4_det_test_right turn.txt" /home/p/桌面/images/2012_val.txt "right
 turn"
Reading annotation for 1/102
Reading annotation for 101/102
0.178694638695
p@p-W650DC:~/桌面/voc_eval-master$
```

图 4-8 改进后 YOLO v2 每个类的精度均值

Figure 4-8 Per class AP of Improved YOLO v2

根据计算得到的值与原 YOLO 算法比较，得到柱状图如图 4-9 所示。

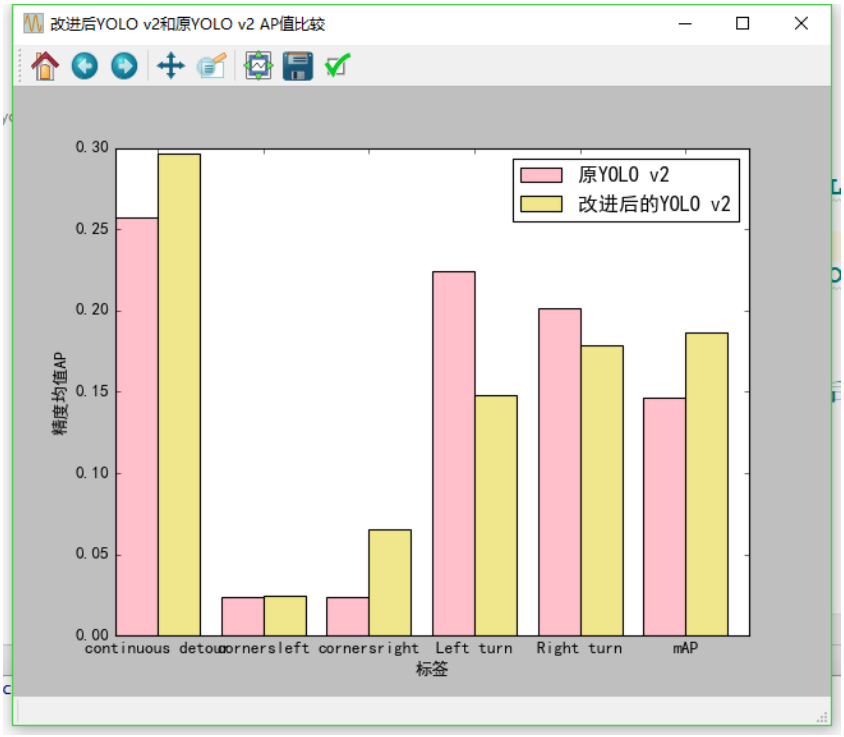


图 4-9 比较结果

Figure 4-9 Comparison of results

由图 4-9 可知，改进后的 YOLO v2 除了 left turn 类和 right turn 类外各个类的精度都提高了，总的平均精度均值 mAP 也提高了，因此加深卷积神经网络能提高 YOLO v2 算法的总体精度。left turn 类和 right turn 类精度没提高应该是训练迭代次数不够，应增加迭代次数。

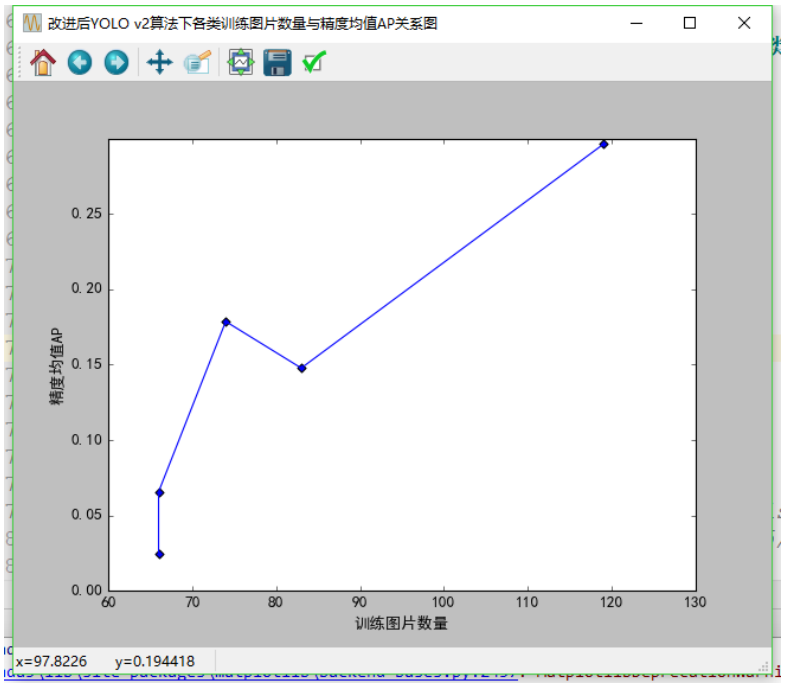


图 4-10 改进后 YOLO v2 训练图片数量与 AP 关系

Figure 4-10 The relationship between the number of training sets and AP

由图 4-10 可得，总体而言，某个类训练图片数量越多，精度就越大。因此

可通过增加数据集来提高精度。

4.3 本章小结（Summary）

本章对原 YOLO v2 算法进行了改动，首先介绍了改进的内容和理由，然后使用图片和摄像头进行测试，计算改进后检测器的 mAP 值，与原算法进行比较，精度值有一定提升，证明改动是有效的。

5 总结与展望
5 Summary and Future Work

5.1 总结（Summary）

自动驾驶的发展催生了对检测道路交通的强烈需求，弯道的特殊性显示出检测弯道标志的重要性。本文的研究目的是实现一种适用于自然场景的弯道标志目标检测方法。本文主要工作如下：

（1）收集 5 类弯道标志数据集。包含 400 多张弯道标志图片，并使用标注工具制作用于图像检测的数据集。

（2）将 Faster R-CNN 目标检测算法和 YOLO v2 目标检测算法应用于弯道标志检测，然后对两种算法进行比较。

（3）在原 YOLO v2 算法框架上适当加深了网络结构，结果显示，改进后 YOLO v2 算法比原来的算法具有更高的检测精度。

本论文完成了弯道标志检测的基本任务，如果需要实现更多功能，还需要做更多的工作。

5.2 展望（Future Work）

目前目标检测技术突飞猛进，深度学习技术在目标检测分类中取得了巨大的成功，目前的瓶颈就是计算的时间复杂度。基于深度学习的检测器虽然在检测准确率上达到了非常高的水平，但是却需要消耗大量的计算资源，以 YOLO 网络为例，在实验室 GTX960m 显卡的测试环境下，每秒钟只能处理 10-13 帧，勉强达到实时性要求，这就要求减少检测次数，如果使用跟踪算法进行跟踪，只在规定的检测帧进行目标检测，而非检测帧，就能减少检测次数，以达到实时处理的要求。因此将来可以增加一个目标跟踪器来提高帧数

本研究实现了使用摄像头对弯道标志进行检测，但还存在一些问题，需进一步研究改善：

（1）检测精度低，需要增加数据集，可通过添加训练图片，或者修改 YOLO 的参数如 angle、hue 等来使得每次训练基于角度、色调产生新的训练图片以扩充数据[76]，来改善数据集数据规模过小的缺陷。

（2）在 GTX960m 显卡的测试环境下，每秒钟能处理 10-15 帧图片，为加快处理速度，可通过更换性能更好的 GPU 提高帧数，或者使用跟踪算法进行跟踪[77]，只在规定的检测帧进行目标检测等方法来达到实时处理要求。

参考文献

[1] 乔永锋.汽车行业正迈入新时代[J].中国汽配市场,2017(06):10.

[2] MCDONALD N. Look and learn: capitalising on individual responsibility in speed management[C]// Proceedings of the 2004 Australian Institute of Traffic Planning and Management (AITPM) National Conference. Adelaide,Australia:[s.n.].2004: 71－83.

[3] 赵晓华, 关伟, 黄利华,等. 急弯处警告标志位置对驾驶行为的影响研究[J]. 公路交通科技, 2014, 31(9):101-107.

[4] Souani C, Faiedh H, Besbes K. Efficient algorithm for automatic road sign recognition and its hardware implementation[J]. Journal of Real-Time Image Processing, 2014, 9(1):79-93.

[5] Greenhalgh J, Mirmehdi M. Traffic sign recognition using MSER and Random Forests[C]// Signal Processing Conference. IEEE, 2012:1935-1939.

[6] Lim H, You K, Sung W. Design and Implementation of Speech Recognition on a Softcore Based Fpga[C]// IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. IEEE, 2006:III-III.

[7] Zhao J, Zhu S, Huang X. Real-time traffic sign detection using SURF features on FPGA[C]// High PERFORMANCE Extreme Computing Conference. IEEE, 2013:1-6.

[8] Farhat W, Faiedh H, Souani C, et al. Embedded system for road sign detection using MicroBlaze[C]// International Multi-Conference on Systems, Signals & Devices. IEEE, 2015:1-5.

[9] Greenhalgh J, Mirmehdi M. Real-Time Detection and Recognition of Road Traffic Signs[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4):1498-1506.

[10] Farhat W, Faiedh H, Souani C, et al. Real-time recognition of road traffic signs in video scenes[C]// International Conference on Advanced Technologies for Signal and Image Processing. IEEE, 2016:125-130.

[11] 田文利. 基于霍夫直线检测与二维透视变换的图像校正恢复算法[J]. 电子测量技术, 2017, 40(9):128-131.

[12] 世亮浦. Research on Road Sign Detection Method Based on Phase Symmetry and MSER[J]. 2017, 07(12):1206-1220.

[13] 鲍朝前. 针对圆形和三角形交通标志的检测与识别[D]. 北京工业大学, 2015.

[14] Lafuente-Arroyo S, Gil-Jimenez P, Maldonado-Bascon R, et al. Traffic sign shape classification evaluation I: SVM using distance to borders[C]// Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE, 2005:557-562.

[15] 刘慧琪. 视觉感知结合学习的道路交通标线检测与识别方法[D].长安大学,2016.

[16] 刘成云. 行车环境下多特征融合的交通标识检测与识别研究[D]. 山东大学, 2016.

[17] 秦飞. 交通标志实时检测与识别技术研究[D].重庆大学,2011.

[18] Farag A A, Abdel-Hakim A E. Detection, categorization and recognition of road signs for autonomous navigation[J]. Proc of Acivs, 2004:125--130.

[19] 蔡自兴, 谷明琴. Traffic sign recognition algorithm based on shape signature and dual-tree complex wavelet transform[J]. Journal of Central South University, 2013, 20(2):433-439.

[20] 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.

[21] Greenhalgh J, Mirmehdi M. Real-Time Detection and Recognition of Road Traffic Signs[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4):1498-1506.

[22] Bahlmann C, Zhu Y, Ramesh V, et al. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information[C]// Intelligent Vehicles Symposium, 2005. Proceedings. IEEE. IEEE, 2005:255-260.

[23] Meuter M, Nunn C, Gormer S M, et al. A Decision Fusion and Reasoning Module for a Traffic Sign Recognition System[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(4):1126-1134.

[24] 高琦煜, 方虎生. 多卷积特征融合的 HOG 行人检测算法[J]. 计算机科学, 2017, 44(b11):199-201.

[25] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[26] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[28] 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.

[29] Zhang J, Huang M, Jin X, et al. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2[J]. Algorithms, 2017, 10(4):127.

[30] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Cvpr, 2015.

[31] Houben S, Stallkamp J, Salmen J, et al. Detection of traffic signs in real-world images: The German traffic sign detection benchmark[C]// International Joint Conference on Neural Networks. IEEE, 2008:1-8.

[32] Mathias M, Timofte R, Benenson R, et al. Traffic sign recognition — How far are we from the solution?[C]// International Joint Conference on Neural Networks. IEEE, 2013:1-8.

[33] Liang M, Yuan M, Hu X, et al. Traffic sign detection by ROI extraction and histogram features-based recognition[C]// International Joint Conference on Neural Networks. IEEE, 2013:1-8.

[34] Wang G, Ren G, Wu Z, et al. A robust, coarse-to-fine traffic sign detection method[C]// International Joint Conference on Neural Networks. IEEE, 2013:1-5.

[35] Jian Yang. Real-time Discrimination of Frontal Face Using Integral Channel Features and Adaboost[A]. IEEE Beijing Section.Proceedings of 2014 IEEE 5th International Conference on Software Engineering and Service Science[C].IEEE Beijing Section:,2014:4.

[36] Gool L V. Pedestrian detection at 100 frames per second[C]// Computer Vision and Pattern Recognition. IEEE, 2012:2903-2910.

[37] Rosenblatt F. The perceptron, a perceiving and recognizing automaton Project Para[M]. Cornell Aeronautical Laboratory, 1957.

[38] 孙志军, 薛磊, 许阳明,等. 深度学习研究综述[J]. 计算机应用研究, 2012, 29(8):2806-2810.

[39] Near-infrared spectroscopy in food science and technology[M]. John Wiley & Sons, 2006.

[40] 田苗, 林岚, 张柏雯,等. 深度学习在神经影像中的应用研究[J]. 中国医疗设备, 2016, 31(12):4-9.

[41] White B W. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, by Frank Rosenblatt[M]. Spartan Books, 1962.

[42] Cybenko G. Approximation by superpositions of a sigmoidal function[J]. Mathematics of Control Signals & Systems, 1993, 2(3):17-28.

[43] 王珏, 李洪研. 基于局部二值模式与多层感知器的中文车牌字符识别高效算法[J]. 计算机应用, 2015(s1):283-285.

[44] 万维. 基于深度学习的目标检测算法研究及应用[D]. 电子科技大学, 2015.

[45] 吴健, 杨慧珍, 龚成龙. 基于分段随机扰动幅值的随机并行梯度下降算法研究[J]. 中国激光, 2014, 41(7):222-227.

[46] 顾乃杰, 赵增, 吕亚飞,等. 基于多 GPU 的深度神经网络训练算法[J]. 小型微型计算机系统, 2015, 36(5):1042-1046.

[47] 刘曙光, 郑崇勋, 刘明远. 前馈神经网络中的反向传播算法及其改进:进展与展望[J]. 计算机科学, 1996, 23(1):76-79.

[48] 蒋伊琳, 佟岐, 张荣兵,等. 自适应梯度下降观测矩阵优化算法[J]. 计算机应用研究, 2017, 34(7):1950-1952.

[49] Charles R Q, Su H, Kaichun M, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[C]//Computer Vision and Pattern Recognition (CVPR),

2017 IEEE Conference on. IEEE, 2017: 77-85.

[50] 刘冶, 潘炎, 夏榕楷,等. FP-CNNH:一种基于深度卷积神经网络的快速图像哈希算法[J]. 计算机科学, 2016, 43(9):39-46.

[51] 徐姗姗, 刘应安, 徐昇. 基于卷积神经网络的木材缺陷识别[J]. 山东大学学报 (工学版), 2013, 43(2): 23-28.

[52] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.[J]. Journal of Physiology, 1962, 160(1):106-154.

[53] 汤浩, 何楚. 全卷积网络结合改进的条件随机场-循环神经网络用于 SAR 图像场景分类[J]. 计算机应用, 2016, 36(12):3436-3441.

[54] 郜丽鹏, 郑辉. 基于 PReLUs-Softplus 非线性激励函数的卷积神经网络[J]. 沈阳工业大学学报, 2018(1):54-59.

[55] 郁松, 彭志文. 基于卷积神经网络的自然背景字符识别[J]. 计算机应用与软件, 2017(12):228-234.

[56] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1):1-17.

[57] 张文达, 许悦雷, 倪嘉成,等. 基于多尺度分块卷积神经网络的图像目标识别算法[J]. 计算机应用, 2016, 36(4):1033-1038.

[58] Sermanet P, LeCun Y. Traffic sign recognition with multi-scale convolutional networks[C]//Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011: 2809-2813.

[59] 刘兵, 张鸿. 基于卷积神经网络和流形排序的图像检索算法[J]. 计算机应用, 2016, 36(2):531-534.

[60] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[61] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]// International Conference on Artificial Intelligence and Statistics. 2012:315-323.

[62] 郜丽鹏, 郑辉. 基于 PReLUs-Softplus 非线性激励函数的卷积神经网络[J]. 沈阳工业大学学报, 2018(1):54-59.

[63] 余萍, 赵继生, 张洁. 基于非线性修正函数的卷积神经网络图像识别研究[J]. 科学技术与工程, 2015, 15(34):221-225.

[64] 王鑫, 侯志强, 余旺盛,等. 基于深度稀疏学习的鲁棒视觉跟踪[J]. 北京航空航天大学学报, 2017, 43(12):2554-2563.

[65] 李彦冬, 郝宗波, 雷航. 卷积神经网络研究综述[J]. 计算机应用, 2016, 36(9):2508-2515.

[66] Gu J, Wang Z, Kuen J, et al. Recent Advances in Convolutional Neural Networks[J]. Computer Science, 2016.

[67] LeCun Y, Jackel L D, Bottou L, et al. Learning algorithms for classification: A comparison on handwritten digit recognition[J]. Neural networks: the statistical mechanics perspective, 1995, 261: 276.

[68] Schmidhuber J. Deep learning in neural networks: An overview[J]. Neural networks, 2015, 61: 85-117.

[69] 吴宗胜, 傅卫平, 韩改宁. 基于深度卷积神经网络的道路场景理解[J]. 计算机工程与应用, 2017, 53(22):8-15.

[70] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.

[71] Sainath T N, Kingsbury B, Saon G, et al. Deep Convolutional Neural Networks for Large-scale Speech Tasks[J]. Neural Networks, 2015, 64:39-48.

[72] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1):1929-1958.

[73] 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.

[74] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3):211-252.

[75] 王瑞霞, 彭国华. 基于黎曼流形稀疏编码的图像检索算法[J]. 自动化学报, 2017, 43(5):778-788.

[76] 俞汝劼, 杨贞, 熊惠霖. 基于深度卷积神经网络的航空器检测与识别[J]. 计算机应用, 2017, 37(6):1702-1707.

[77] 鄢宇烈. 一种在线学习实时长期目标跟踪算法研究与实现[D].电子科技大学,2017.

作者简历

一、基本情况

姓名：刘向恒 性别：男 民族：汉 出生年月：1992-01-12 籍贯：湖南省永州市

2012-09—2016-06 湖南大学信息科学与工程学院学士；

2016-09—2018-06 中国矿业大学计算机科学与信息学院硕士

二、软件著作权

1. 具有无线通信功能的环境温度监控软件，登记号：2017SR458503，授权日期：2017 年 8 月 21 日

三、获奖情况

2016 年 9 月，获得研究生一等奖学金

学位论文原创性声明

本人郑重声明：所呈交的学位论文《一种基于深度学习的路面弯道标志检测方法》，是本人在导师指导下，在中国矿业大学攻读学位期间进行的研究工作所取得的成果。据我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

年　　月　　日

学位论文数据集

关键词*	密级*	中图分类号*	UDC	论文资助
交通标志；目标检测；卷积神经网络；弯道	公开	TP393		

学位授予单位名称*	学位授予单位代码*	学位类别*	学位级别*
中国矿业大学	10290	工学	硕士

论文题名*	并列题名*	论文语种*
一种基于深度学习的路面弯道标志检测方法	Method for Traffic Sign Detection of the Curved Roads Based on Deep Learning	中文

作者姓名*	刘向恒	学号*	TS16170047P2
培养单位名称*	培养单位代码*	培养单位地址	邮编
中国矿业大学	10290	江苏省徐州市	221116
学科专业*	研究方向*	学制*	学位授予年*
计算机技术	图像检测	两年	2018

论文提交日期*	2018 年 4 月

导师姓名*	鲍宇	职称*	副教授

评阅人	答辩委员会主席*	答辩委员会成员
盲 评	姜淑娟	

电子版论文提交格式　文本（ ✓ ） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ）
其他（ ）

推荐格式：application/msword; application/pdf

电子版论文出版（发布）者	电子版论文出版（发布）地	权限声明

论文总页数*	51

注：共 33 项，其中带*为必填数据，共 22 项。