

分类号： TP391.4

单位代码： 10636

密 级： 公开

学 号： 20180991013

# 四川师范大学

## 专业学位硕士学位论文



中文论文题目： 基于深度学习的车辆检测与识别  
研究

英文论文题目： Research on Vehicle Detection and  
Recognition Based on Deep Learning

论文作者： 杨帆

指导教师： 廖磊

专业学位类别： 工程硕士

专业领域： 电子与通信工程

论文形式： 专题研究

所在学院： 物理与电子工程学院


论文提交日期： 2021 年 5 月 13 日

论文答辩日期： 2021 年 5 月 20 日

## 四川师范大学学位论文独创性声明

本人声明：所呈交学位论文 基于深度学习的车辆检测与识别研究，是本人在导师 廖磊 指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

本人承诺：已提交的学位论文电子版与论文纸本的内容一致。如因不符而引起的学术声誉上的损失由本人自负。

学位论文作者： 

签字日期：2021 年 06 月 20 日


## 四川师范大学学位论文版权使用授权书


本人同意所撰写学位论文的使用授权遵照学校的管理规定：

学校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大学拥有学位论文的部分使用权，即：1) 已获学位的研究生必须按学校规定提交印刷版和电子版学位论文，可以将学位论文的全部或部分内容编入有关数据库供检索；2) 为教学、科研和学术交流目的，学校可以将公开的学位论文或解密后的学位论文作为资料在图书馆、资料室等场所或在有关网络上供阅读、浏览。

本人授权万方数据电子出版社将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。同意按相关规定享受相关权益。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名： 

导师签名： 

# 基于深度学习的车辆检测与识别研究

电子与通信工程专业

研究生 杨帆      指导教师 廖磊

**摘要** 随着 AI 时代的到来，国家对 AI 产业的支持力度大大提高，国内众多企业纷纷投入 AI 产品的研发。智能交通作为其中的一环，成为工业界和学术界研究的热点之一。车辆的检测与识别是建设智能交通的核心技术之一，能够实现自动驾驶、车流量统计、摄像头追踪定位等，具有非常大的应用前景和研究价值。然而，在实际应用中，由于自然场景复杂，车辆检测将面临各种各样的挑战，例如天气、视角、遮挡等。本文结合实际工程应用场景，研究了深度学习在车辆检测与识别中的应用。本文的具体研究内容如下：

首先，对深度学习的历史进行了回顾，阐述深度学习中的卷积神经网络的理论基础、实现原理和优缺点分析。主要分析了卷积神经网络的卷积层、几种常用的激活函数、池化层以及以 VGG, ResNet 为代表的基础网络。然后，对以 R-CNN 为代表的两阶段检测算法和以 YOLO 系列和 SSD 为代表的一阶段检测算法进行了深入的研究。

其次，针对传统的目标检测不能及时和准确响应多变交通环境的问题，本文选择一阶段检测算法来实现车辆检测，主要选择目前性能最为优良的 YOLOv4 框架为基准，结合实际项目应用，以压缩模型提高速度为目标，构建快速高效的车辆检测模型。通过从算法框架和数据扩增策略等方面进行改进，最终在检测精度与 YOLOv4 相差不大的情况下，使车辆检测模型的推理速度提高了 43%。

最后，对于车辆识别同样以保障精度为前提的情况下，提升分类网络的速度。通过对 PeleeNet 引入 CSPNet、SE 注意力机制、Swish 激活函数等策略，最终 BFLOPs 减少了 13%，TOP-5 精度提高了 1 个百分点。

论文研究成果应用于“智慧停车”项目和“AI 值守”项目。其中“智慧停车”已完成上线测试，将很快投入生产应用。“AI 值守”已完成算法的设计，以及硬件平台的算法移植，业务逻辑还在开发中。

**关键词：**车辆检测    深度学习    卷积神经网络    YOLOv4    PeleeNet

# Research on Vehicle Detection and Recognition Based on Deep Learning

**Major:** Electronics and communication engineering

**Graduate Student:** Yang Fan      **Supervisor:** Liao Lei

**Abstract** With the development of Artificial Intelligence(AI) technology, and the AI industry has greatly increased under the support of the government, many companies and institutions have invested in the research and development of AI products. For one, Intelligent Transportation System(ITS) has become one of the hotspots of research in industry and academia. Vehicle detection and recognition is an important technology for the construction of ITS. The main aspects include automatic driving, traffic-flow statistics, object tracking, and positioning, etc. It has great application prospects and research value. However, in the practical implementation, as the complexity of natural scenes, vehicle detection will encounter various challenges, such as weather, viewing angle, occlusion, and so on. In this paper, we combined with actual engineering application scenarios, we studied vehicle detection and recognition based on the deep learning method. The main research content of this article is as follows:

First, briefly reviewed the history of deep learning, and the theoretical basis, implementation principles, and described the advantages and disadvantages of convolutional neural networks(CNNs). Mainly analyze the convolutional layer of CNN, several commonly used activation functions, pooling layer, and the basic network including with VGG and ResNet. Then, we have deeply studied the object detection models including the two-stage detection algorithms represented by R-CNN and the one-stage detection algorithms represented by the YOLO series and SSD.

Secondly, as traditional object detection cannot quickly and accurately respond to the changing traffic environment, this paper chooses a one-stage detection algorithm to achieve vehicle detection, selecting the current best-performing YOLOv4 framework as the benchmark, combined with actual project applications, and compressing the model the goal is to increase speed and build a fast and efficient vehicle detection model.

By improving the algorithm framework and data augmented strategy, the inference speed of the vehicle detection model is increased by 43% while the detection accuracy is closed to the YOLOv4.

Finally, in the case, that vehicle recognition is also based on ensuring accuracy, improved the speed of the classification network. By introducing CSPNet, SE attention mechanism, Swish activation function, and other strategies to PeleeNet, the final BFLOPs were reduced by 13%, and the accuracy of TOP-5 was improved by 1%.

The research results of the thesis are applied to the "Smart Parking" project and the "AI Guard" project. Among them, "Smart Parking" has completed the online test and will soon be put into production and application. "AI Guard" has completed the algorithm design and the algorithm migration of the hardware platform, and the business logic is still under development.

**Keywords:** Vehicle Detection   Deep Learning   CNN   YOLOv4   PeleeNet

## 插图与附表清单

图 2.1 卷积操作 .....	6
图 2.2 最大值池化和平均值池化 .....	7
图 2.3 LeNet-5 网络结构 .....	7
图 2.4 VGG-16 .....	8
图 2.5 残差模块 .....	9
图 2.6 RCNN 算法流程 .....	9
图 2.7 SPP 结构 .....	10
图 2.8 Fast RCNN 网络结构 .....	11
图 2.9 Faster RCNN 网络结构 .....	12
图 2.10 RPN 网络结构 .....	13
图 2.11 YOLOv1 网络结构 .....	14
图 2.12 SSD 结构框图 .....	14
图 2.13 Darknet19 .....	16
图 2.14 YOLOv3 网络结构 .....	17
图 3.1 改进 YOLOv4 的框架 .....	18
图 3.2 普通卷积 .....	21
图 3.3 Depthwise 卷积 .....	21
图 3.4 Pointwise .....	22
图 3.5 Relu 和 Leaky Relu 激活函数 .....	22
图 3.6 数据集制作流程图 .....	23
图 3.7 图像样本 .....	24
图 3.8 XML 标签信息 .....	24
图 3.9 训练样本 .....	25
图 3.10 异常样本 .....	25
图 3.11 数据分析 .....	26
图 3.12 YOLOv4 与改进的 YOLOv4 实验对比 .....	28

图 3.13 路边智慧停车系统 .....	29
图 3.14 业务逻辑.....	30
图 3.15 实测场景.....	30
图 3.16 准确率统计 .....	31
图 4.1 SE 块结构图 .....	33
图 4.2 Squeeze 和 Excitation 操作 .....	33
图 4.3 Stem Block.....	34
图 4.4 Two-Way Dense Layer.....	35
图 4.5 CSPPeleeNet-SE 结构图 .....	36
图 4.6 图片样本.....	37
表 3.1 主干网络 MobileNetv3 网络结构 .....	20
表 3.2 两种模型的计算量 .....	27
表 3.3 本章算法与现阶段算法对比结果 .....	28
表 4.1 PeleeNet .....	36
表 4.2 对比实验.....	38

# 目次

摘要 .....	I
Abstract.....	II
插图与附表清单 .....	IV
1 绪论 .....	1
1.1 研究背景与意义 .....	1
1.2 国内外研究现状 .....	2
1.3 本文主要工作 .....	3
1.4 论文章节安排 .....	4
2 相关理论研究 .....	5
2.1 卷积神经网络 .....	5
2.2 两阶段检测算法 .....	8
2.3 一阶段检测算法 .....	12
2.4 本章小结 .....	17
3 基于改进 YOLOv4 的车辆检测方法 .....	18
3.1 算法流程 .....	18
3.2 数据集制作 .....	22
3.3 实验结果分析 .....	25
3.4 工程实测 .....	28
3.5 本章小结 .....	30
4 基于改进 CNN 的车辆识别方法 .....	31
4.1 分类网络设计 .....	31
4.2 实验结果分析 .....	35
4.3 工程应用 .....	37
4.4 本章小结 .....	37
5 总结与展望 .....	38
5.1 工作总结 .....	38
5.2 研究展望 .....	38



参考文献 .....	40
致谢 .....	44

# 1 绪论

## 1.1 研究背景与意义

随着当今社会的快速发展,机动车数量与日剧增,从而导致的交通拥堵现象日益严重<sup>[1]</sup>。据相关部门统计,2020 年全国的机动车保有量达 3.72 亿辆,其中汽车 2.81 亿辆;机动车驾驶人达 4.56 亿人,其中汽车驾驶人 4.18 亿人<sup>[2]</sup>。据世界卫生组织《道路安全全球现状报告 2018》统计,2016 年道路交通死亡人数达到 135 万人<sup>[3]</sup>,因此,交通环境恶化和车辆引发的交通事故引起了公众的关注。随着智慧城市的建设,监控设备遍布在城市的各个地方,尤其是位于交通道路的监控设备可以采集到大量的车辆视频信息<sup>[4]</sup>。仅仅通过人工分析视频中的车流量和车牌信息,很难满足现代交通的需求。只有通过快速准确地获取道路上车辆的位置和属性,才能提前预警或及时响应潜在的交通事故。

因此,人们提出了智能交通的概念。智能交通系统涵盖了物联网、大数据、人工智能和自动控制理论等技术领域,有效地将交通设施、车辆、驾驶员等要素关联起来,实现高效的管理<sup>[5]</sup>。通过构建智能交通系统,来准确分析当前的交通环境,能够有效提高交通部门的运营效率。

车辆检测和识别是构建智能交通系统最重要的方面之一。传统的车辆检测和识别技术大多使用手工设计的特征和经典分类器<sup>[6]</sup>。车辆在空间的位置是通过对图像的背景建模来获得,然后根据手工设计的特征训练一个分类器对车辆进行分类<sup>[7]</sup>。传统技术的局限性在于对场景的依赖性很强,一旦场景发生了变化了,那么之前训练的分类器的效果下降得就很明显。而自然场景下的环境是复杂多变的,不同的天气、光照、监控设备的视角对车辆检测与识别的影响很大。所以传统的车辆检测和识别技术无法满足智能交通对响应的强实时性和高准确率的要求。自 2006 年以来,深度学习(Deep Learning)作为机器学习(Machine Learning)一个新研究领域,已成为图像、音频或文本等数据的高效建模方法。同时,GPU 芯片的研发以及硬件平台的快速发展,使计算机的计算能力稳步跃升,为深度学习技术承载了重要的开发环境。卷积神经网络(CNN)作为深度学习技术中的重要组成部分,近年来在图像检测和识别任务中取得了令人瞩目的成就。为了高效处理大量的图像数据,基于卷积神经网络的计算框架不断涌现出来。因此,采用深度学习的方法进行车辆检测与识别是非常有研究意义和应用前景的。本文在此背景下,结合实习单位的项目,对现有的深度学习方法进行研究,提出一个高效的车辆检测与识别方法。

## 1.2 国内外研究现状

### 1.2.1 图像分类现状

车辆识别属于图像分类应用范畴。随着国家兴起打造智慧城市，智慧交通系统的不断发展，车辆识别一直受到学术界的广泛关注<sup>[8]</sup>。

目前，图像分类的方法主要分为两大类，一类是基于传统图像处理的方法，通过提取分类目标的纹理、形状等特征进行分类，另一类是基于深度学习的方法。

传统图像分类算法一般包括预处理、特征提取、分类器设计三个阶段。图像预处理一般分为图像去噪和图像增强。常用的图像去噪方法包括均值滤波、中值滤波、稀疏编码去噪和小波去噪等。图像增强的目的是为了突出分类目标的信息从而提高分类的准确率，一般根据实际应用情况选择，常见的图像增强方法包括图像锐化、直方图均衡化、gamma 校正、显著性检测等。特征提取是计算机视觉任务的核心，其目的是提取出有利于分类识别的特征，一般根据目标的纹理、颜色、梯度、形状等去计算，常用的特征提取方法有 SIFT<sup>[9]</sup>、HOG<sup>[10]</sup>、SURF<sup>[11]</sup>、Haar-like<sup>[12]</sup>等。经过特征提取后，得到一个目标的特征向量，然后在通过分类器对特征向量进行分类，常用的分类器有 SVM<sup>[13]</sup>（支持向量机）、DPM<sup>[14]</sup>、贝叶斯<sup>[15]</sup>等。基于传统图像处理的方法依靠人工方式设计特征，需要很强的先验知识，而且很难提取数据中隐含的判别特征，一个模型的好坏往往依赖于设计人员的经验。

而基于深度学习的方法就避免了在学习过程中的人为干预，它依靠网络本身，通过学习大规模的数据来提取特征，不需要人工设计特征，这就使得训练出来的模型更具鲁棒性。Yann Lecun 在 1998 年提出了 LeNet-5<sup>[16]</sup>，包含了卷积层、池化层、激活函数层、全连接层，是现代神经网络的雏形，手写数字的识别准确率达到了 99%。2012 年 Alex Krizhevsky 团队提出的 AlexNet<sup>[17]</sup>获得了当年的 ImageNet 图像分类挑战赛的冠军，Top-5 的错误率为 15.3%，远远低于采用传统方法的 26.2%，体现了卷积神经网络强大的优势。随后，各种更准确、更高效的网络相继出现，在 AlexNet 中使用了 11x11 的大卷积核来提取特征，这就导致卷积核的参数过多，图像快速缩小而丢失细节特征。针对 AlexNet 的弊端，Simonyan 等人提出了 VGGNet<sup>[18]</sup>，它使用较小的 3x3 卷积核来代替大卷积核，这样可以增加特征图像的层数，以及不同尺度的感受野。随后何凯明等人提出了 ResNet<sup>[19]</sup>，通过将低层特征与高层特征结合，解决了因网络层数过多而导致的梯度弥散问题，它的网络多达 152 层，获得了 2015 年 ImageNet 大赛冠军。但是，随着网络的深度增大，计算量也随之增大，如何给网络“瘦身”，在保证精度的情况下降低计算量成为计算机视觉领域研究热点之一。近几年研究者们提出了很多策略在降低计算量的同时还能提高网络的精度，比如谷歌在 2017 年提出的

MobileNet<sup>[20]</sup>，通过使用深度可分离卷积来降低卷积操作的计算量。同年，旷视科技提出的 ShuffleNet<sup>[21]</sup>通过采用分组卷积的方式来降低计算量。诸如此类降低网络计算量的策略还有很多，使得一些实时性要求高的应用得以落地。

### 1.2.2 目标检测现状

车辆检测属于目标检测应用范畴。目标检测任务是指对图像中的目标物体分类的同时，并返回该目标物体在图像中的具体位置。目标检测一般包括生成候选区域、提取区域特征、对区域进行分类、后处理等几个步骤<sup>[22]</sup>。分为传统的目标检测方法和基于深度学习的目标检测方法。

传统的目标检测方法采用滑动窗口的方式生成候选区域，使用不同大小的窗口在图像上滑动，这种类似于穷举的方法能生成大量的候选区域<sup>[23]</sup>。但是这种方式没有针对性，会生成很多的冗余窗口，计算量非常大。特征提取和区域分类就和传统的图像分类方法一样，使用人工设计的特征，利用 HOG、SIFT 等方法进行特征提取，然后用一个分类器对区域分类。后处理用 NMS（非极大值抑制算法）<sup>[24]</sup>剔除多余的框。

基于深度学习的目标检测方法早期根据是否需要生成候选区域分为两阶段(Two-Stage)检测算法和一阶段(One-Stage)检测算法<sup>[25]</sup>，后期又根据是否使用 Anchor 机制来分为 Anchor-Free 算法和 Anchor-Based 方法。2014 年，R.Girshick 等人提出的两阶段检测算法 RCNN(Regions with CNN Features)<sup>[26]</sup>，首次将卷积神经网络应用到目标检测任务中，利用 Selective Search（选择性搜索）算法<sup>[27]</sup>生成候选区域，为后续的两阶段检测算法研究提供了思路。2015 年，J.Redmon 等人提出了一阶段检测算法 YOLO(You Only Look Once)算法<sup>[28]</sup>，将目标检测问题转换为回归问题，是一阶段检测算法的开山之作。近年来，目标检测主要从生成候选区域、特征提取网络、损失函数等几个方向进行优化，涌现了大量的优秀算法。两阶段检测算法经典的有 RCNN 系列、SPP-Net<sup>[29]</sup>、FPN<sup>[30]</sup>等。一阶段检测算法经典的有 YOLO 系列、SSD<sup>[31]</sup>、RetinaNet<sup>[32]</sup>、EfficientDet<sup>[33]</sup>。Anchor-free 经典的有 CornerNet<sup>[34]</sup>、CenterNet<sup>[35]</sup>等。两阶段检测算法和一阶段检测算法各有优点，两阶段检测算法精度高，但实时性没有一阶段检测算法好。但随着时代的进步，应用的标准提高，如何做到高精度的同时兼顾实时性是未来目标检测的研究重点。

## 1.3 本文主要工作

随着深度学习的快速发展，基于深度学习的目标检测与识别方法渐渐取代了传统的目标检测与识别方法。但是，这些算法只是针对一般物体的检测与识别，没有针对特定领域进行优化<sup>[36]</sup>。比如车辆检测中出现的遮挡、光照、天气、摄像

头视角等问题，以及部署在边缘端的实时性等。因此，本文结合实际项目，研究基于深度学习的车辆检测与识别，提出了基于改进YOLOv4的车辆检测算法和基于改进CNN的车辆分类算法。主要工作如下：

（1）算法分析。对深度学习中的分类和检测算法的历史进行了回顾，对VGGNet、ResNet、RCNN系列、YOLO系列等代表性的算法进行原理分析与优缺点总结。

（2）数据集制作。采集车辆检测数据集。针对项目的实际应用场景，收集位于路灯上挂载的摄像头所拍摄的视频，筛选标注。

（3）检测算法设计。结合实际项目应用改进YOLOv4算法，更换轻量级的主干网络，同时修改激活函数，提升网络的检测速度和易于在边缘端部署。

（4）分类算法设计。测试不同分类网络的性能，组合自己的分类网络。

## 1.4 论文章节安排

本文研究内容以以下几个章节呈现：

第1章为绪论。首先简单阐述了本文研究课题的背景和意义。其次，介绍了基于传统图像处理的目标检测与分类方法到基于深度学习的目标检测与分类方法的发展历程。最后介绍了本文的研究内容。

第2章介绍相关理论。首先介绍了卷积神经网络，以其中具有代表性的网络模型为例进行说明。其次对两阶段目标检测算法和一阶段目标检测算法的原理进行了分析。

第3章提出了基于改进YOLOv4的车辆检测算法与车辆检测数据集制作。本章结合应用场景和边缘端硬件，更改YOLOv4的主干网络，将CSPDarknet换成更轻量级的MobileNetv3，详细介绍了算法的流程与原理，探讨了改进点的作用并通过实验评估验证了算法的有效性。

第4章提出了基于改进的CNN的车辆分类算法。本章通过分析对比目前主流CNN框架，组合一个适用于车辆分类的特征提取网络。

第5章为总结与展望。总结本文的研究内容及创新点，分析其中的不足之处和改进方案，以及后续的研究重点。

## 2 相关理论研究

### 2.1 卷积神经网络

卷积神经网络(Convolutional Neural Networks,CNN)是由 LeCun 在 1998 年提出的,这是一种前馈型神经网络<sup>[37]</sup>,广泛的应用于计算机视觉中。简单的卷积神经网络包含输入层、卷积层(Convolutional layer)、激活函数、池化层(Pooling layer)、全连接层(FC layer)等<sup>[38]</sup>。对于图像分类问题,在卷积神经网络后面会连接一个 softmax 函数用于分类任务。对于检测问题,卷积神经网络可以与区域候选网络进行结合,也可以与回归网络进行结合。

#### 2.1.1 卷积层

卷积层是卷积神经网络的核心部分,主要用于对输入图像进行特征提取。它要求输入是具有宽度、高度和深度的三维形状<sup>[39]</sup>。

实际上,在分析数学中,卷积运算是一种算子,并且在卷积神经网络中一般只是处理离散的情况<sup>[40]</sup>。卷积运算的过程如图 2.1 所示,使用一定大小的卷积核(又称“过滤器”)作用在输入图像局部,将对应位置相乘然后求和,并按照这种方式对图像进行扫描来得到特征图<sup>[41]</sup>。卷积层的超参数有很多,例如卷积核的个数、大小、步长等。

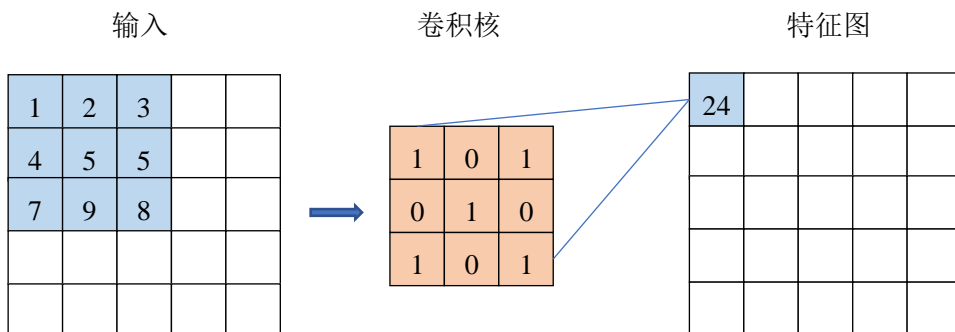


图 2.1 卷积操作

#### 2.1.2 激活函数

激活函数的作用去线性化,引入非线性操作。因为多个线性操作层的叠加依然只是线性映射,如果只通过线性变换,那么任意层的神经网络模型和单层神经网络模型的表达能力就没有什么区别<sup>[42]</sup>。因此需要引入非线性操作,以此来增加模型的表达能力,也就是使用激活函数层。否则,网络将无法模拟复杂的函数。常用的激活函数有很多,例如 Sigmoid 激活函数和 ReLU 激活函数(Rectified Linear Units)等,公式分别为:

$$\sigma = \frac{1}{1+e^{-z}} \quad (2-1)$$

$$R(z) = \max(0, z) \quad (2-2)$$

### 2.1.3 池化层

池化层也叫下采样层(Down-Sampling)，操作与卷积层相似，不过池化层没有参数。池化层可以降低特征图的尺寸，减少网络的参数量，进而可以加快训练的速度<sup>[43]</sup>。除此之外，池化层还能够防止网络出现过拟合的情况。池化一般包括平均值池化(Mean-Pooling)和最大值池化(Max-Pooling)等。如图 2.2 所示。

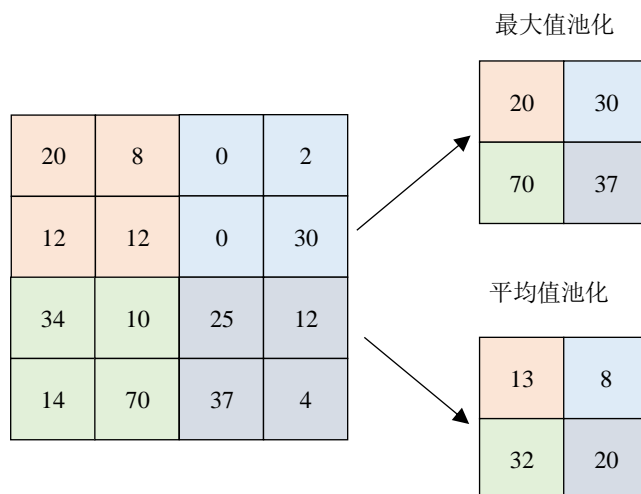


图 2.2 最大值池化和平均值池化

### 2.1.4 LeNet-5

Yann Lecun 在 1998 年提出了 LeNet-5。它包含了输入层、卷积层、池化层、激活函数层、全连接层，是现代神经网络的雏形。LeNet-5 的网络结构如图 2.3 所示。

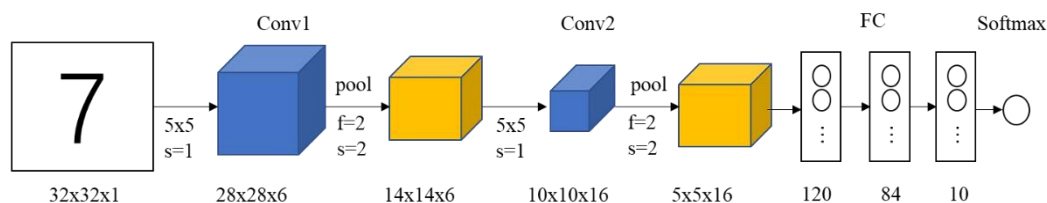


图 2.3 LeNet-5 网络结构

LeNet-5 主要是用来做手写数字识别的，网络一共只有 7 层，包括 2 个卷积层、2 个池化层和 3 个全连接层。2 个卷积层都是采用步长为 1、大小为 5x5 的卷积，池化层使用 2x2 的平均值池化，最后由 SoftMax 输出数字 0 到 9 的概率。LeNet-5 的结构虽然简单，但在 MNIST 数据集上，手写数字的识别准确率达到了 99%。

### 2.1.5 VGGNet

VGGNet 是由 Simonyan 等人在 2015 年提出的深度神经网络模型。它的网络

深度为 16 到 19 层，即 VGG-16 和 VGG-19 两种结构。VGGNet 整个网络重复使用大小为  $3 \times 3$  和  $1 \times 1$  的卷积，以及  $2 \times 2$  的池化操作。 $3$  个  $3 \times 3$  卷积和  $1$  个  $7 \times 7$  的卷积效果相当，而参数量只有  $7 \times 7$  卷积的 55%。引入  $1 \times 1$  卷积的好处是在保持特征图大小不变的情况下，引入非线性变换，增加网络的深度，同时能对特征图的通道数进行一个升维和降维。这就意味着使用多个小卷积核能够使得网络层数加深的同时使得网络的总参数量减少。

如图 2.4 所示，VGG-16 共用 16 层，包含 13 个卷积层和 3 个全连接层。在 VGG-16 中，每 2 到 3 个卷积层堆叠组成卷积序列，利用 Same Padding 卷积方式保持输入输出的特征图大小不变，步长为 1；池化层使用  $2 \times 2$  的池化操作，步长为 2；全连接层包含 3 个全连接，通道数依次为 4096、4096、1000；最后在使用 SoftMax 进行分类输出。

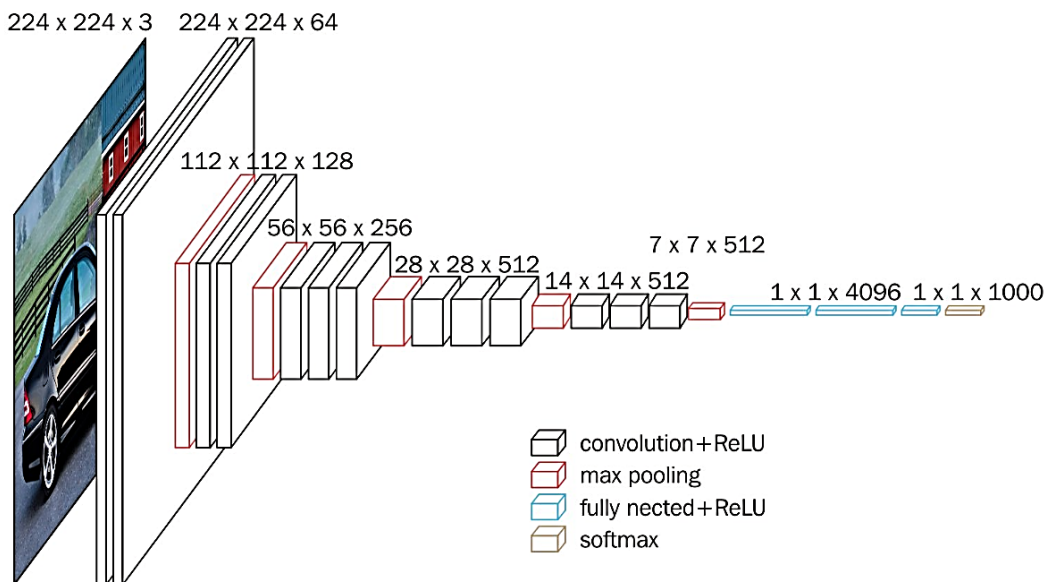


图 2.4 VGG-16

### 2.1.6 ResNet

神经网络的发展从 LeNet 到 AlexNet，再到 VGGNet，网络的层数在不断增加，这意味着网络的深度对网络性能有很大的影响<sup>[44]</sup>，层数深的网络可以提取图片的低层、中层、高层等特征。但是，当网络达到一定深度后，仅仅在后面继续堆叠更多的层会出现很多问题。第一个问题就是梯度爆炸，由于网络层数过深，在反向传播的时候，不能有效的把梯度传到前面的网络层，从而导致前面的网络层参数无法更新<sup>[45]</sup>。第二个问题就是退化，网络层数过深会使得优化更困难，导致准确率下降。

在 2015 年，何凯明提出了 ResNet(残差网络)，通过引入残差结构来解决退化问题，如图 2.5 所示。



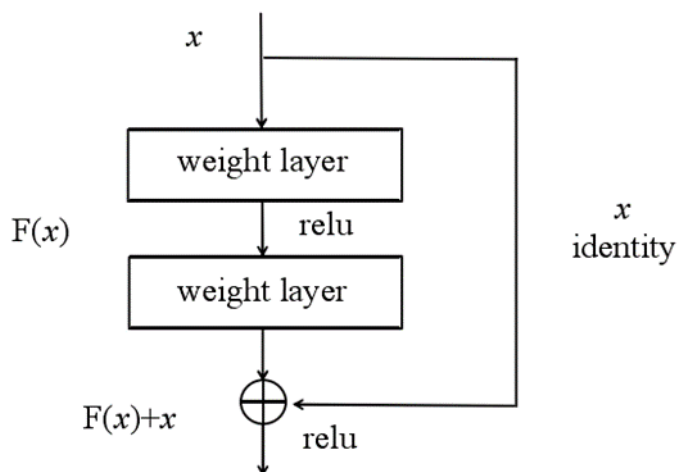


图 2.5 残差模块

其中  $x$  是输入,  $F(x)$  是残差结构拟合的函数,  $H(x)$  是期望的潜在映射, 与其让  $F(x)$  直接学习  $H(x)$ , 不如直接让  $F(x)$  学习残差  $H(x)-x$ , 即  $F(x)=H(x)-x$ , 这样原本的前向路径就变为  $F(x)+x$ , 用  $F(x)+x$  来拟合  $H(x)$ , 那么最终的映射就为  $H(x)=F(x)+x$ 。这种残差映射的方式比原始映射更容易调优。此外, 通过重复利用中间特征层的方式, 有效的解决了因网络层数过深而导致的网络性能退化问题。

## 2.2 两阶段检测算法

两阶段检测算法是一种基于区域的方法, 首先要生成可能包含目标的候选区域, 然后再对候选区域进行分类<sup>[46]</sup>。最具代表性的有 R-CNN、SPP-Net、Fast RCNN<sup>[47]</sup>、Faster RCNN<sup>[48]</sup>等。

### 2.2.1 RCNN

RCNN(Regions with CNN Features)是在 2014 年由 Ross Girshick 等人提出来的目标检测方法, 它开创性的将深度学习引入目标检测任务, 其算法流程如图 2.6 所示。

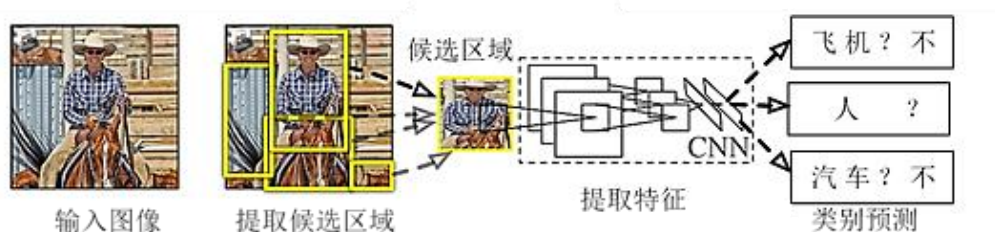


图 2.6 RCNN 算法流程

RNN 先利用 Selective Search (选择性搜索)<sup>[49]</sup>算法生成候选区域<sup>[50]</sup>。用 EGBIS (Efficient Graph-based Image Segmentation) 图像分割算法将图像分割为多个区域, 再计算相邻区域的相似度, 根据相似度进行合并, 直到整张图合并为一个

个区域<sup>[51]</sup>。对于所有在合并过程中产生的候选区域都会给出相应的矩形框，得到用于检测的候选区域。

其次利用卷积神经网络提取区域特征。由于全连接层只接收固定大小的输入，因此先将候选区域缩放到相同大小(227x227)，然后将所有候选区域送入到 AlexNet 中提取特征，以最后一个全连接层的输出来作为候选区域的特征表示。

最后在对区域进行分类和边框校准。分类采用的是 SVM 分类器，需要针对每个类别训练一个分类器。而边框校准使用的是线性回归模型，让检测框的位置更加准确，同时边框也更紧凑。

RCNN 相比于传统目标检测算法，准确率有了很大的提升<sup>[52]</sup>。但缺点是训练过慢，对于每一个候选区域都要单独提取特征。由于全连接层的输入要求固定大小的向量，对生成候选区域后的图片做缩放操作会影响图片的质量和-content。而且对于每一类都要单独训练一个 SVM 分类器。

### 2.2.2 SPP-Net

针对 RCNN 的缺点，何凯明等人提出了 SPP-Net，改进点主要有两个，一个是卷积共享，在特征提取的时候不是输入基于候选框的图片，而是将整张图片送入神经网络做特征提取，在根据生成的候选区域在特征图上的映射，找到其对应的特征。另一个改进点是引入空间金字塔池化(Spatial Pyramid Pooling, SPP),它可以接受不同尺寸的特征图输入，然后输出固定维度的向量到全连接层。SPP 的结构如图 2.7 所示。

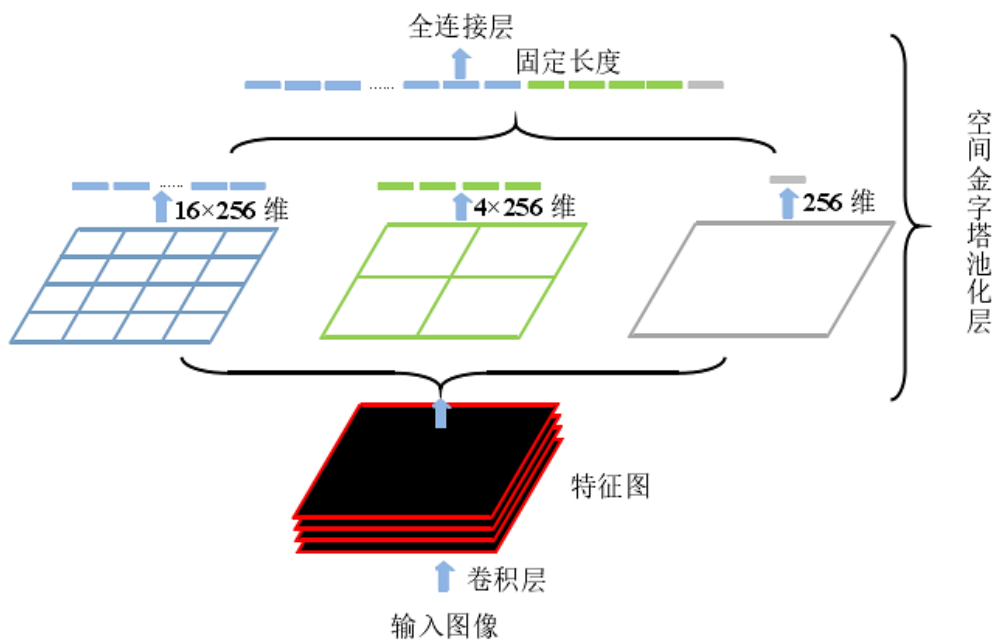


图 2.7 SPP 结构

SPP 对输入的特征图进行 3 次池化操作，分别将特征图划分为 1 个块、4 个

块和 16 个块，对每个块做最大池化操作，最后输出 SPP 输出的向量维度是  $256+4 \times 256+16 \times 256=21 \times 256$ 。

虽然 SPP-Net 比 RCNN 快了不少，但依然会为每个类别单独训练一个 SVM 分类器，不能实现端到端的训练。

### 2.2.3 Fast RCNN

RCNN 的作者提出了 RCNN 的改进版 Fast RCNN，改进点主要由两个，一个是结合 SPP-Net 的空间金字塔池化思想提出了 ROI Pooling。另一个是将分类和边框回归合并在一起训练。Faster R-CNN 网络结构如图 2.8 所示。

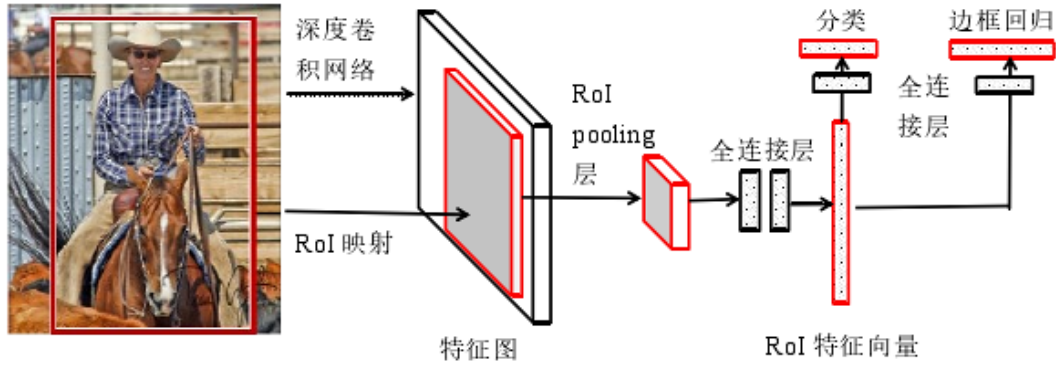


图 2.8 Fast RCNN 网络结构

ROI Pooling 具体操作是将生成的 ROI(Region Of Interest, ROI)映射到对应的特征图上，将映射后的区域划分为大小相等的子区域，然后只进行一次池化操作。

其次，Fast RCNN 采用了多任务损失函数，将分类和边框回归边框放在网络里一起训练。同时将 SVM 分类器用 SoftMax 代替，SoftMax 输出的是每个类别的概率。Fast RCNN 的损失函数如下：

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \quad (2-3)$$

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} smooth_{L_1}(t_i^u - v_i) \quad (2-4)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (2-5)$$

其中， $L_{cls}(p, u)$  是分类的对数损失函数， $u$  是真实类别值， $p$  是预测类别的概率值， $L_{loc}(t^u, v)$  是边框回归的损失函数， $v$  是真实物体的边框位置， $t^u$  是预测的边框位置， $\lambda$  是一个超参数，用来衡量分类和边框回归损失函数的权重。

Fast RCNN 虽然检测准确率提升了不少，但由于需要使用 Selective Search 算法来生成候选区域，这种类似于穷举的方法无法高效的筛选出候选区。

## 2.2.4 Faster RCNN

针对 Fast RCNN 的缺点，该作者对 fast RCNN 的结构做出了调整，提出了性能更好的 Faster RCNN。

Faster R-CNN 主要改进了生成候选区域的方法，采用了区域建议网络 (Region Proposal Net-work, RPN)来生成候选区域。Faster RCNN 的网络结构如图 2.9 所示。同样是将整张图片输入卷积神经网络提取特征，得到特征图，然后将特征图送入 RPN 生成候选框，最后在同时进行分类与边框回归。

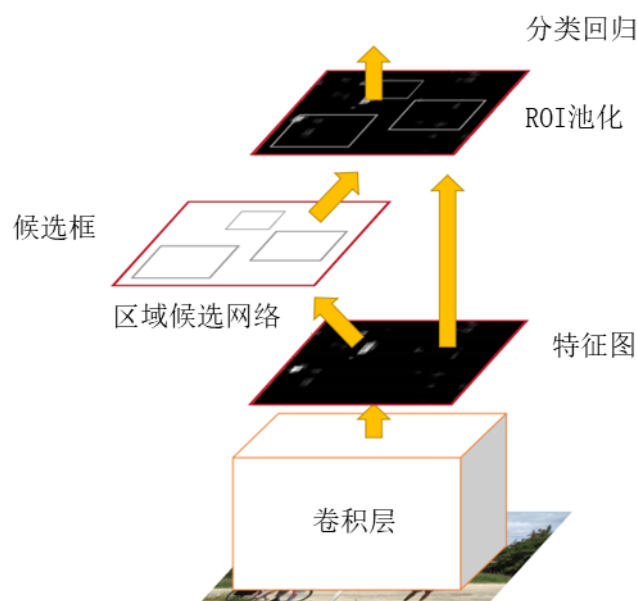


图 2.9 Faster RCNN 网络结构

RPN 采用了 Anchor（锚）机制来生成候选框，它先对卷积层输出的特征图做一次填充操作，然后用  $3 \times 3$  的滑动窗口扫描特征图上的每一个像素点，为每个像素点按不同的长宽比生成  $k$  个 Anchor Box (通常  $k$  的值为 9)。然后通过 Anchor box 与真实框的 IoU（交并比）来判断是否为正样本（是否存在待检目标）。如果 IoU 值大于 0.7，就标记 Anchor Box 为正样本；如果 IoU 小于 0.3，就标记 Anchor Box 为负样本；其它的不参与训练。最后能剩下 2000 个左右的 Anchors。其次，RPN 还会对 Anchor Box 的位置进行修正，通过计算 Anchor Box 与真实框的偏移量，学习 Anchor Box 与真实框的差异。RPN 结构如图 2.10 所示。

RPN 会输出 2000 个类别得分，和 4000 个物体位置信息。在输入 ROI Pooling 层之前，还会使用 NMS（非极大值抑制算法）剔除堆叠较高的 Anchor Box，最终剩下 300 个左右的 Anchor Box。

所以，Faster RCNN 通过使用 RPN 来生成候选区域，使得训练过程更为简便。

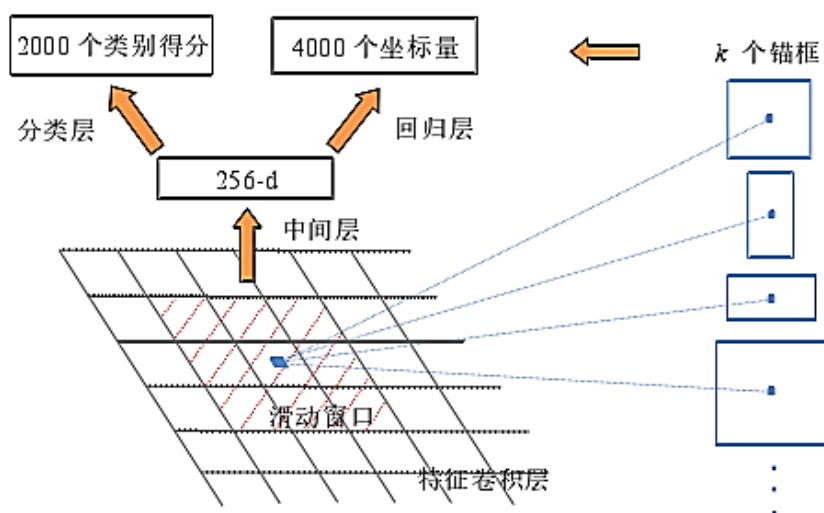


图 2.10 RPN 结构图

## 2.3 一阶段检测算法

一阶段检测算法与两阶段检测算法的区别在于是否需要生成候选区域。一阶段算法速度具有显著的优势，且随着相关研究逐渐对图像特征更深层次的理解，以及优秀的锚框生成规则的设计，许多优秀的目标检测算法相继提出，其中经典的检测框架有 SSD 系列和 YOLO 系列，其主要的思路是使用先验框对图像的不同位置以及多个维度进行抽样，然后使用 CNN 提取特征后再对多个特征块直接进行分类与回归，此类算法具有实时高效的特点。

### 2.3.1 YOLOv1

YOLOv1(You Only Look Once)是在 2016 年由 Joseph Redmon 等人提出来的目标检测方法。YOLOv1 把目标检测定义为一个回归问题，整个过程只经过一个神经网络，同时预测 Bounding Box(边界框)和类别的概率，能够端到端的对网络进行优化。

YOLOv1 生成候选框的方式与 RNN 不同。首先将输入图片划分为  $7 \times 7$  的网格，当标注目标的中心坐标落在哪个网格，那么这个网格就要负责预测这个目标<sup>[53]</sup>。每个网格负责预测 2 个 Bounding Box 的坐标和置信度(confidence)。每个 Bounding Box 有 5 个输出，分别是  $(x, y, w, h, c)$ 。其中， $x$  和  $y$  表示预测的 Bounding Box 的中心坐标与其所在网格左上角的偏移值； $w$  和  $h$  表示预测的 Bounding Box 的宽高占整个图片宽高的比例；置信度( $c$ )表示网格包含目标的概率。假如网格不包含目标的中心点，那么置信度为 0。反之，置信度就等于 IOU 的值。置信度的计算公式如下：

$$c = P_r(\text{Object}) * IOU_{pred}^{truth} \quad (2-6)$$

然后设置一个阈值过滤掉置信度较低的 Bounding Box。同时采用 NMS 剔除重叠度较高的 Bounding Box。此外，每个网格需要预测 C 个类别的概率（在论文中  $C=20$ ）。最终输出的向量维度为  $7 \times 7 \times ((4+1) \times 2 + C)$ 。YOLOv1 的网络结构如图 2.11 所示。

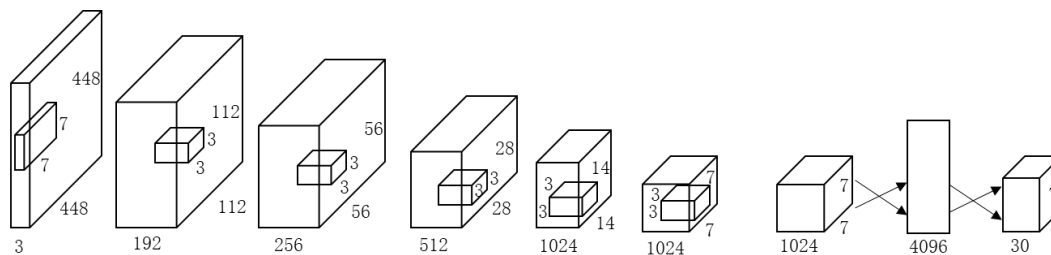


图 2.11 YOLOv1 的网络结构

YOLOv1 的优点在于将目标检测任务定义成回归问题，检测速度相比较于 Fast RCNN，有了很大的提高。但是，由于 YOLOv1 的每个网格只预测置信度大的那个类别，对于小目标的预测效果不是很好。其次，YOLOv1 只支持固定大小的输入图片，其它大小的图片需要经过缩放才能进行训练或者预测。另外，YOLOv1 采用网格划分的方式生成 Bounding Box，这种方式导致 Bounding Box 的回归定位不是很准确。

### 2.3.2 SSD

SSD 模型主要由用于特征提取的基础网络和用于目标检测的若干个多尺度特征块级联而成。前置的基础网络块由深度卷积神经网络堆叠而成，其功能是从原始输入图片中进行特征提取。位于后端的级联多尺度特征检测块组成的网络，接收前端网络输出特征在多个卷积层中进行预测。SSD 算法巧妙的设计了在不同尺度上对特征图进行检测的特点，为后来特征金字塔网络的设计提供了一定的思路和借鉴作用。多尺度检测在目标检测任务中显著提升了检测质量。具有大尺度的特征图保留了较多的空间位置信息，在该尺度对小目标具有一定表征能力。小尺度特征图有着更为丰富的语义信息，更利于对大尺寸的目标进行回归与分类，SSD 的整体框架结构如图 2.12 所示。

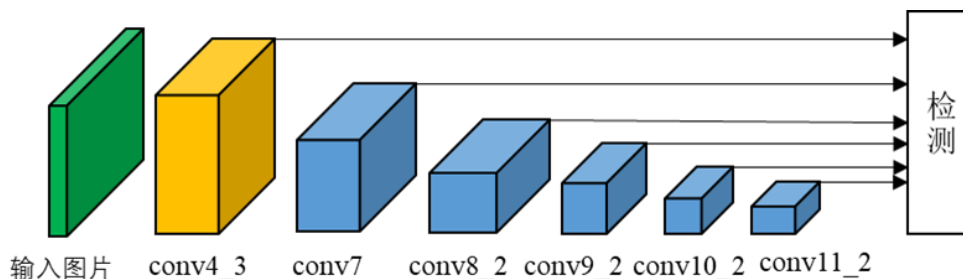


图 2.12 SSD 结构框图



SSD 算法是一种端到端训练的单次检测的深度学习神经网络，结合了 YOLO 的回归策略和 Faster R-CNN 的 anchor 机制，采用的回归策略可以在一定程度上简化神经网络的计算复杂度，提高检测的实时性<sup>[54]</sup>。SSD 通常使用 VGG-16 作为主干网络从原始输入图片中提取特征，并且将 VGG-16 最后的两个全连接层替换成卷积核大小为  $3 \times 3$  与  $1 \times 1$  的卷积层，同时采用空洞卷积操作提高网络的感受野，通过去除分类层并增加多尺度的卷积层提高检测质量。

SSD 直接产生多个固定大小的先验框，以及对框偏移量和分类准确率计算置信度。通过非极大抑制操作（NMS）消除多余的边框，保留目标预测最大得分输出不同尺寸大小目标的检测结果，在不同尺度生成数量不等和大小不同的先验框。SSD 采用多组级联特征图用于目标的定位与回归，其尺寸分别为  $38 \times 38$ 、 $19 \times 19$ 、 $10 \times 10$ 、 $5 \times 5$ 、 $3 \times 3$  与  $1 \times 1$ ，假设模型检测时采用  $m$  层特征图，则第  $k$  个特征图的默认框比例计算如下：

$$s_k = s_{\min} + (s_{\max} - s_{\min})(k - 1) / (m - 1), k \in \{1, 2, \dots, m\} \quad (2-7)$$

其中， $s_{\min}$ 、 $s_{\max}$  分别表示最低层与最高层的默认框占输入图像大小的比例，在算法实施中，设置  $s_{\min} = 0.2$ ， $s_{\max} = 0.95$ ，SSD 采用 anchor 生成策略，在生成设计中，预设不同纵横比例的系数，主要用于增强多个默认框对不同尺寸大小物体的检测。同时，默认框的高宽比预设为 1、2、1/2、3、1/3，每个默认框的高和宽的计算公式如下：

$$w_k^n = s_k \sqrt{r_n}, n \in \{1, 2, 3, 4, 5\} \quad (2-8)$$

$$h_k^n = s_k / \sqrt{r_n}, n \in \{1, 2, 3, 4, 5\} \quad (2-9)$$

当  $r=1$  时， $s_k = \sqrt{s_k s_{k+1}}$ ，有： $w_k^6 = h_k^6 = \sqrt{s_k s_{k+1}}$ ，设定默认框的中心坐标为： $((a+0.5)/|f_k|, (b+0.5)/|f_k|)$ ，其中  $|f_k|$  是第  $k$  个特征图的尺寸大小， $a, b \in \{0, 1, 2, \dots, |f_k| - 1\}$ ，并截取默认框的坐标使其在  $[0, 1]$  内<sup>[55]</sup>。特征图上默认框坐标与原始图像坐标的映射关系如下：

$$x_{\min} = (c_x + w_b / 2)w_{\text{img}} / w_{\text{feature}} = ((a + 0.5) / |f_k| - w_k / 2)w_{\text{img}} \quad (2-10)$$

$$y_{\min} = (c_y + h_b / 2)h_{\text{img}} / h_{\text{feature}} = ((b + 0.5) / |f_k| - h_k / 2)h_{\text{img}} \quad (2-11)$$

$$x_{\max} = (c_x + w_b / 2)w_{\text{img}} / w_{\text{feature}} = ((a + 0.5) / |f_k| + w_k / 2)w_{\text{img}} \quad (2-12)$$

$$y_{\max} = (c_y + h_b / 2)h_{\text{img}} / h_{\text{feature}} = ((b + 0.5) / |f_k| + h_k / 2)h_{\text{img}} \quad (2-13)$$

式中， $(c_x, c_y)$  为特征层上默认框中心的坐标， $w_b$ 、 $h_b$  为默认框的宽和高， $w_{\text{feature}}$ 、 $h_{\text{feature}}$  为特征层的宽和高， $w_{\text{img}}$ ， $h_{\text{img}}$  为原始图像的宽和高，

$(x_{\min}, y_{\min}, x_{\max}, y_{\max})$  为第  $k$  层特征图上中心为  $((a+0.5)/|f_k|, (b+0.5)/|f_k|)$  大小为  $w_k, h_k$  的默认框映射到原始图像的物体框坐标。

### 2.3.3 YOLOv2

YOLOv2<sup>[56]</sup>针对 YOLOv1 的缺点, 进行了一系列的改进, 大大提高了检测精度。YOLOv2 引入了 Faster RCNN 中的 Anchor 机制, 同时使用 K-means 算法在训练集上进行聚类分析, 得到更好的 Anchor 模板。YOLOv2 还使用新的特征提取网络 Darknet19, 如图 2.13 所示。

Type	Filters	Size/Stride	Output
Convolutional	32	$3 \times 3$	$224 \times 224$
Maxpool		$2 \times 2/2$	$112 \times 112$
Convolutional	64	$3 \times 3$	$112 \times 112$
Maxpool		$2 \times 2/2$	$56 \times 56$
Convolutional	128	$3 \times 3$	$56 \times 56$
Convolutional	64	$1 \times 1$	$56 \times 56$
Convolutional	128	$3 \times 3$	$56 \times 56$
Maxpool		$2 \times 2/2$	$28 \times 28$
Convolutional	256	$3 \times 3$	$28 \times 28$
Convolutional	128	$1 \times 1$	$28 \times 28$
Convolutional	256	$3 \times 3$	$28 \times 28$
Maxpool		$2 \times 2/2$	$14 \times 14$
Convolutional	512	$3 \times 3$	$14 \times 14$
Convolutional	256	$1 \times 1$	$14 \times 14$
Convolutional	512	$3 \times 3$	$14 \times 14$
Convolutional	256	$1 \times 1$	$14 \times 14$
Convolutional	512	$3 \times 3$	$14 \times 14$
Maxpool		$2 \times 2/2$	$7 \times 7$
Convolutional	1024	$3 \times 3$	$7 \times 7$
Convolutional	512	$1 \times 1$	$7 \times 7$
Convolutional	1024	$3 \times 3$	$7 \times 7$
Convolutional	512	$1 \times 1$	$7 \times 7$
Convolutional	1024	$3 \times 3$	$7 \times 7$

图 2.13 Darknet19

相比 YOLOv1 的特征提取网络, YOLOv2 去掉了 Dropout, 在每个卷积层中加入了 Batch Normalization (批量标准化), 对网络每一层的输入做归一化操作, 防止过拟合和加速损失函数的收敛。同时使用  $3 \times 3$  卷积替代 YOLOv1 中的  $7 \times 7$  卷积, 在减少参数数量的同时增加了网络的深度。-

YOLOv2 在训练上使用迁移学习的方法, 先在分类数据集 ImageNet 上以  $224 \times 224$  的输入大小预训练 Darknet19。然后调整输入为  $448 \times 448$  再次训练, 让网络适应高分辨率的输入。最后再在检测数据集上训练。

### 2.3.4 YOLOv3

YOLOv3<sup>[57]</sup>对 YOLOv2 的改进点主要有以下几点, 改进后的 YOLOv3 结构如图 2.14 所示。



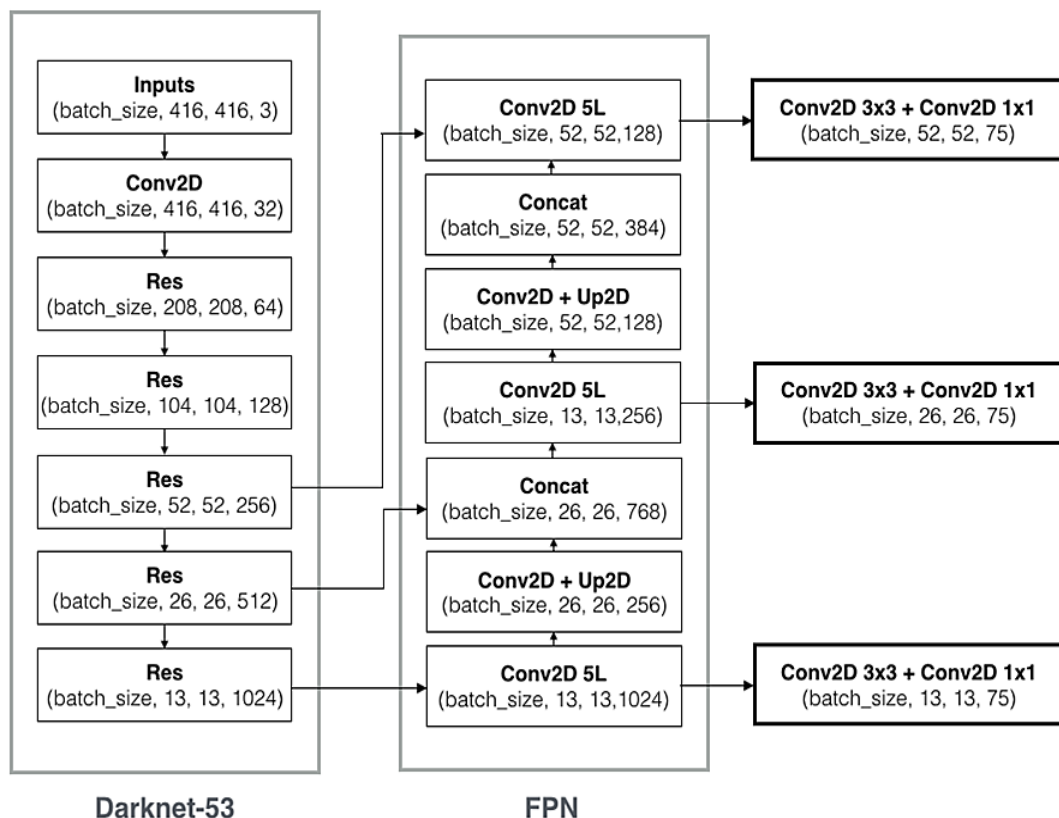


图 2.14 YOLOv3 网络结构

首先，YOLOv3 改进了 YOLOv2 的 Darknet19，加深了网络深度，同时引入 ResNet 的残差结构，来避免因网络过深而引起的梯度爆炸，提升网络的学习能力。并且使用卷积操作来代替池化操作进行下采样。

其次，YOLOv3 借鉴了 FPN 的上采样和特征融合思想。高层特征具有丰富的语义信息，但位置信息丢失严重。而低层特征则反之。FPN 的思想就是把高层特征和低层特征融合起来，这样的特征层就同时具备丰富的语义信息和位置信息。YOLOv3 融合了三个尺度的特征层，输出的特征融合层大小分别是 13x13、26x26 和 52x52，然后分别在这三个特征融合层上进行检测。这种方式提升了模型对不同尺度目标检测的鲁棒性。

### 2.3.5 YOLOv4

YOLOv4<sup>[58]</sup>是 AlexeyAB 等人在 YOLOv3 的基础上提出的，YOLOv4 的初衷加快模型的运行速度，在并行计算上对神经网络进行优化，使得模型可以在常规 GPU 进行训练和检测。YOLOv4 指出对目标检测模型的改进方向分为 Bag of freebies (BoF) 和 Bag of Specials (BoS)。BoF 指只改变训练策略或只增加训练成本的方法，目标检测中常用的符合 BoF 定义的方法有数据增强、标签平滑、改进损失函数等。BoS 指只会增加少量网络的推理成本但能显著提高算法精度的

模块和后处理方法，目标检测中常用的符合 BoS 定义的方法有扩大感受野（如 SPP）、引入注意力机制、增强特征之间的交流（如 FPN）等。YOLOv4 采用 CSPDarknet53 作为特征提取主干网络，同时还在网络中引入 SPP 和 PAN<sup>[59]</sup>来实现特征的跨层融合，此外还增加了一系列的 BoF 和 BoS 操作。

YOLOv4 巧妙结合了多种优化技巧，其在 MS COCO 数据集上的 AP 达到了 43.5%，在 Tesla V100 上的检测速度达到了 65FPS。YOLOv4 在精度和速度方面都达到了很高的水准，是现阶段目标检测算法中性能最好的检测算法之一。

## 2.4 本章小结

本章首先研究了卷积神经网络的基础内容。然后分析了 VGGNet、ResNet、GooleNet 等经典的卷积神经网络。然后研究了两阶段检测算法和一阶段检测算法，对算法的原理以及优缺点进行了分析总结。本章所介绍的基础理论为本文的后续内容做铺垫。因此，在结合实际应用场景的情况下，经过对比分析一阶段检测算法和两阶段检测算法在精度和速度等方面的优劣，本文决定使用在精度和速度上都非常优秀的 YOLOv4 用于车辆检测。

### 3 基于改进Yolov4的车辆检测方法

在车辆检测的应用中一般对实时性要求很高,因而一阶段的检测算法备受青睐。YOLOv4 是目前性能最优的检测算法之一。YOLOv4 引入了大量先进的特性以增强网络的识别能力,汲取了 CSPNet<sup>[60]</sup>、SPP、PAN 等网络结构中的一些核心思想,以及一些数据增强、训练技巧等等。本章在此基础上进行改进,提出改进的 YOLOv4 算法。原始的 YOLOv4 采用了 CSPdarkNet-53 作为主干网络来进行特征提取,考虑到实际应用是部署在边缘端,对实时性要求很高,因此采用更轻量级 MobileNetv3<sup>[61]</sup>网络来做特征提取。此外,还结合边缘端硬件对 YOLOv4 的一些操作进行了替换,易于在边缘端部署。训练时,针对数据集的不均衡,结合实际场景对数据集做了增强。

#### 3.1 算法流程

由于 Yolov4 在目标检测方面的突出效果,本文针对实际应用场景对 Yolov4 进行了改进,改进后的整体框架如图 3.1 所示。整个网络结构可以分为三个部分,分别是:

- 主干特征提取网络,对应图中的 MobileNetv3。
- 增强特征提取网络,对应图中的 SPP、PANet。
- 预测网络,对应图中的三个 YoloHead。

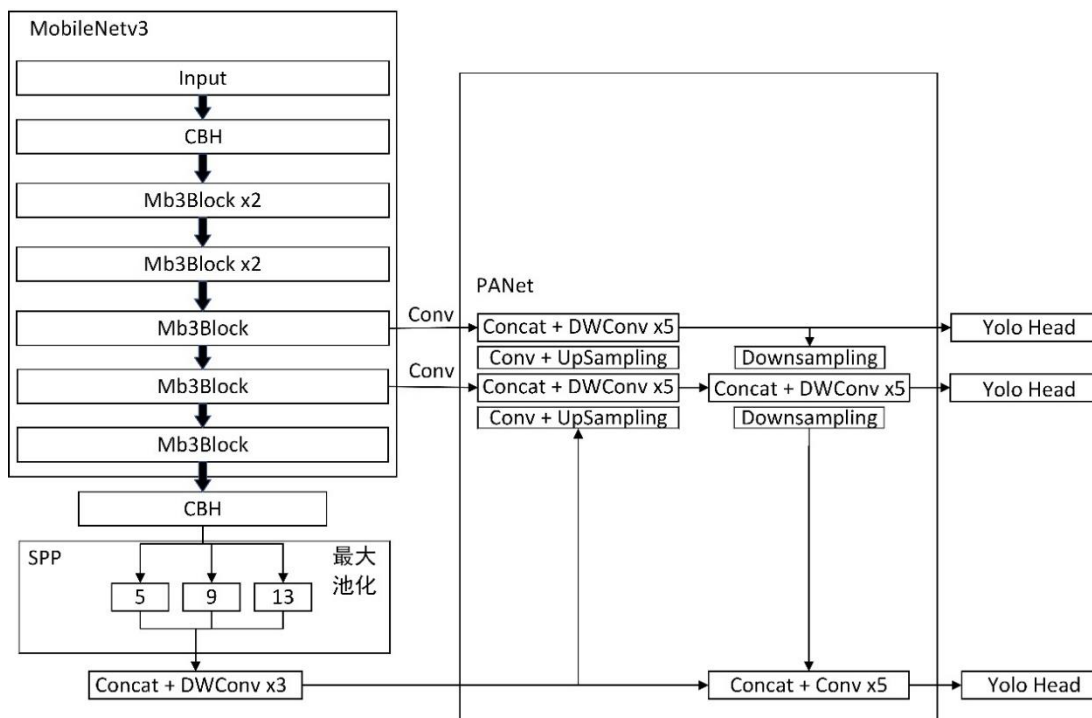


图 3.1 改进的 YOLOv4 框架

其中 a 部分的作用是对输入的图片进行初步的特征提取,这样可以得到三个初步的有效特征层。b 部分的作用是加强特征提取,通过对 a 得到的三个特征层进行特征融合操作,从而增强模型对不同缩放尺度目标的检测<sup>[62]</sup>。c 部分的作用是利用 b 得到的特征层来获得预测结果。

### 3.1.1 主干网络的改进

在原始的 YOLOv4 模型中使用 CSPDarknet53 作为主干网络来进行特征提取,但是 CSPDarknet53 的参数量太大,边缘设备的算力有限,在同时检测多路视频的时候无法达到实时检测,所以把主干网络替换成轻量级的网络,其结构如表 3.1 所示。

表 3.1 主干网络 MobileNetv3 网络结构

Operator	Exp size	#out	SE	NL	s
CBH	-	16	-	H	2
mb3Block	16	16	√	R	2
mb3Block	72	24	-	R	2
mb3Block	88	24	-	R	1
mb3Block	96	40	√	H	2
mb3Block	240	40	√	H	1
mb3Block	240	40	√	H	1
mb3Block	120	48	√	H	1
mb3Block	144	48	√	H	1
mb3Block	288	96	√	H	1
mb3Block	576	96	√	H	1
mb3Block	576	96	√	H	1
CBH	-	576	-	H	1

本章使用的主干网络修改自 MobileNetv3。与原始 MobileNetv3 不同的是,由 15 个 Block 变为 11 个 mb3Block,同时修改了每个 Block 的通道数来匹配 YOLOv4 的输入尺寸。在表 3.1 中,第一列表示每层特征经历的 Block 结构, CBH 表示一个由 Conv2d、BatchNormalization、Hard-swish 组成的 Block, mb3Block 表示一个由倒置残差模块、SE 注意力机制、膨胀卷积和深度可分离卷积组成的 Block,第二列表示膨胀系数,第三列表示输入 Block 时特征层的通道数,第四列表示是否使用 SE 注意力机制,第五列表示激活函数的种类, H 表示 Hard-swish 激活函数, R 表示 Relu 激活函数,第六列表示步长。

### 3.1.2 卷积改进

原始的 YOLOv4 采用的是普通的 2D 卷积,为了进一步的减少参数量,这里

借鉴 MobileNetv1 中的深度可分离卷积(Depthwise Separable Convolution), 将 YOLOv4 中的普通卷积替换成深度可分离卷积。

常规的卷积操作如图 3.2 所示, 输入是一个  $6 \times 6 \times 3$  (通道数为 3) 的矩阵, 经过卷积核大小为  $3 \times 3$  的卷积层, 假设输出的通道数为 4, 采用 valid padding 的方式, 步长为 1, 则最终会输出 4 个 feature map, 每个 feature map 的大小为  $4 \times 4$ 。因此卷积层的参数量为  $P_{\text{conv}} = 4 \times 3 \times 3 \times 3 = 108$ 。

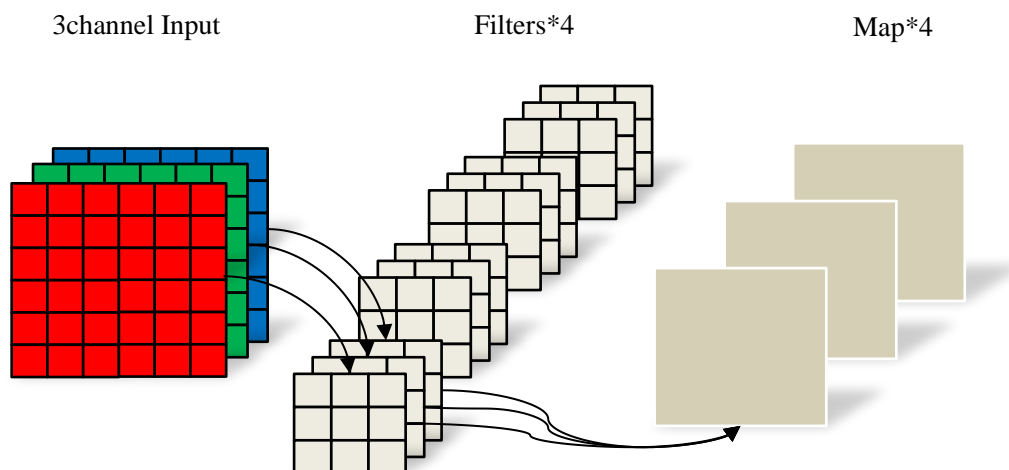


图 3.2 普通卷积

深度可分离卷积是一个可分解卷积的操作, 包含 Depthwise 卷积和 Pointwise 卷积两部分, 即将一个完整的卷积分为两个步骤进行。与普通卷积不同的是, Depthwise 卷积的每个卷积核只负责一个输入的通道, 而普通卷积的每个卷积核同时操作输入的每个通道。同样对于一个  $6 \times 6 \times 3$  的输入, 输入的每一个通道只和一个  $3 \times 3$  的卷积核进行卷积操作, 最后生成 3 个  $4 \times 4$  的 Feature map。如图 3.3 所示, 其中一个 Filter 仅包含一个  $3 \times 3$  的卷积核, 所以卷积部分的参数量为  $P_{\text{depthwise}} = 3 \times 3 \times 3 = 27$ 。

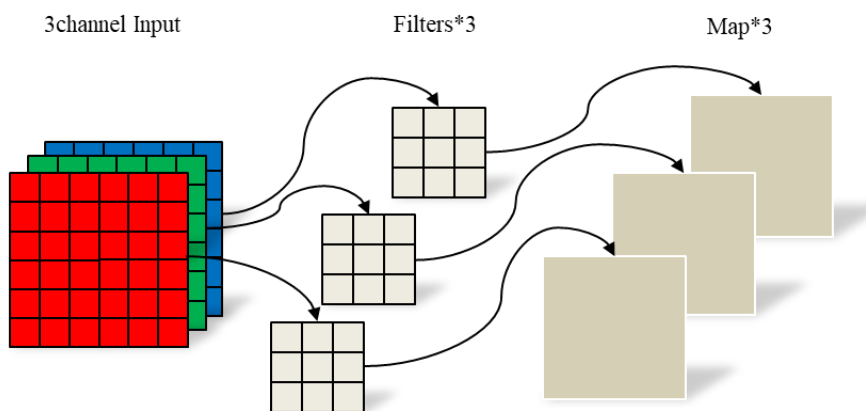


图 3.3 Depthwise 卷积

Pointwise 卷积运算与常规卷积相似，主要作用是将上一步 Depthwise 卷积得到的 Feature map 在深度方向进行加权组合。它使用  $1 \times 1 \times d$  ( $d$  为上一层的通道数) 的卷积核，生成的 Feature map 个数与卷积核的个数相等。如图 3.4，这里一个 Filter 包含一个  $1 \times 1 \times 3$  的卷积核，所以 Pointwise 卷积的参数量为  $P_{\text{pointwise}} = 1 \times 1 \times 3 \times 4 = 27$ 。

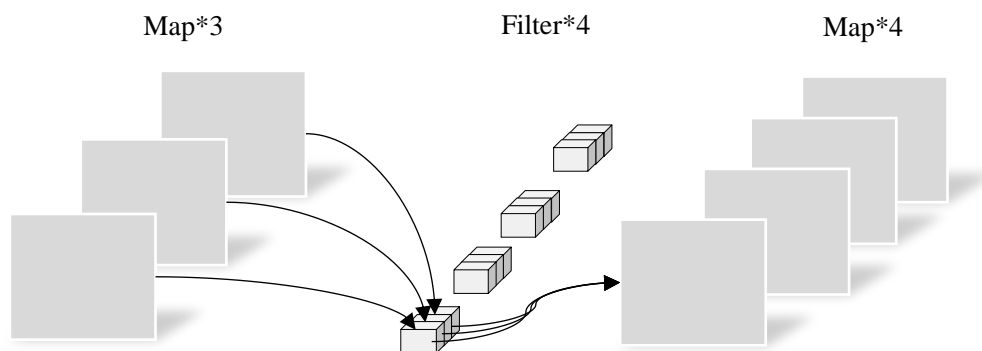


图 3.4 Pointwise 卷积

所以，深度可分离卷积的参数量为  $P_{\text{separable}} = 27 + 12 = 39$ ，相比普通卷积的参数量，前者的参数量仅为后者的三分之一。由此可见，将原始 YOLOv4 的普通卷积替换为深度可分离卷积，能大大减少计算量，有效的提高模型在边缘端的推理速度。

### 3.1.3 激活函数

在原始的 YOLOv4 中使用了 Relu、Leaky Relu 和 Mish 等激活函数。使用激活函数的目的是为了给神经元引入非线性因素，从而使神经网络可以逼近任意的非线性函数。然而在一些边缘设备中，有些激活函数并不被支持，比如在 FPGA 中，由于只能做加、乘运算，就无法使用指数型激活函数。所以需要将原始 YOLOv4 中的 Mish 激活函数替换为 Leaky Relu 激活函数。本章使用的 ReLU 激活函数和 Leaky Relu 激活函数如图 3.5。

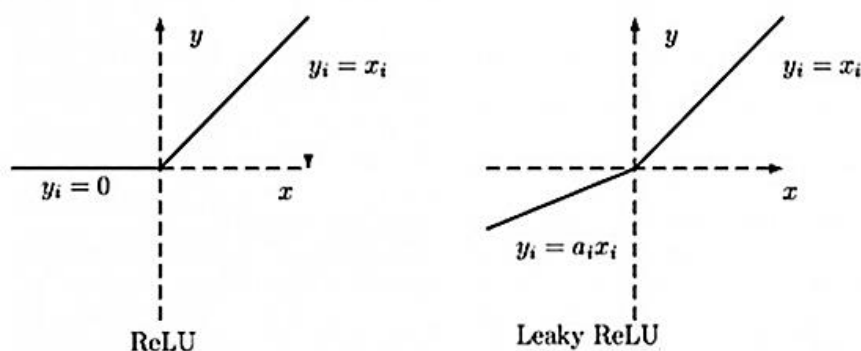


图 3.5 Relu 和 Leak Relu 激活函数

## 3.2 数据集制作

对于深度学习检测任务而言，最关键的往往不是算法本身，而在于有一个高质量的数据集。数据集的质量在很大程度上影响了模型的最终效果，所以在工作中，算法工程师大部分时间都在对数据集进行清洗。

通过对车辆检测开源数据集的调研，发现并不适合当前的项目场景，所以本文制作了自己的数据集，满足本文实际需求。图 3.6 所示是数据集的制作过程。

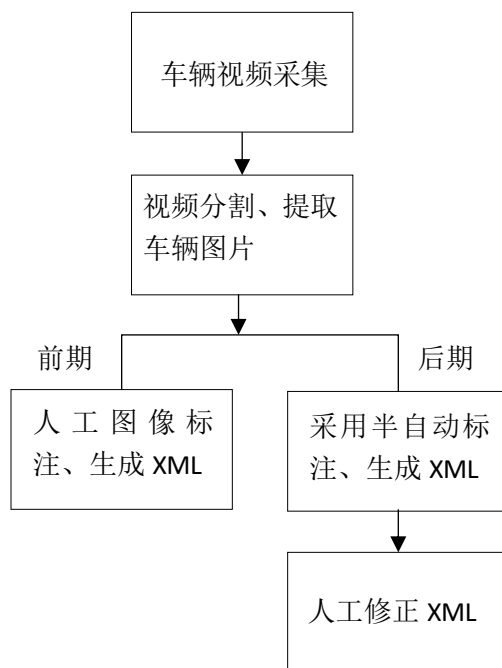


图 3.6 数据集制作流程图

### 3.2.1 数据采集

本文使用的数据集来自于实习单位的相关深度学习项目，通过道路旁灯杆上挂载的工业摄像头采集视频，然后在分割成图片进行标注。本文自制的数据集主要包含了晴天、阴天、夜晚和目标遮挡等几种自然场景的图片数据。一共截取了 10000 张图片，格式为 jpg,大小为 1920x1080,图像样本如图 3.7。

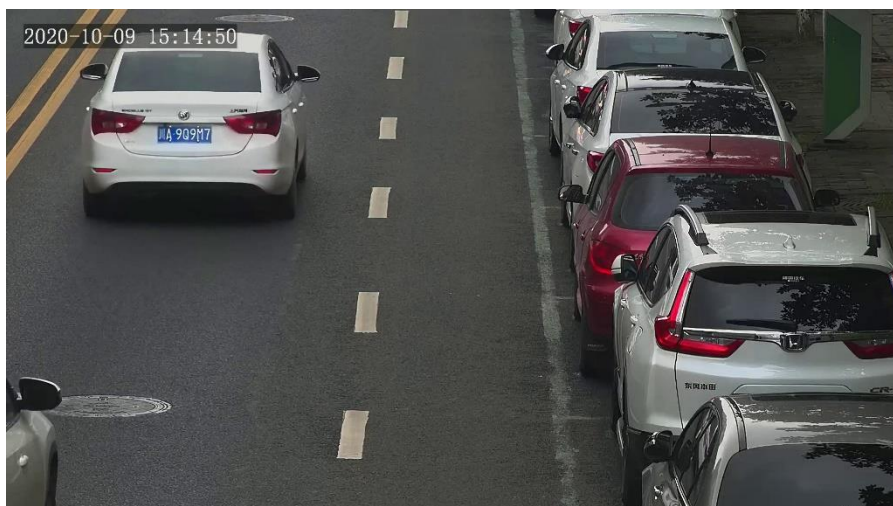


图 3.7 图像样本

### 3.2.2 数据标注

目标检测需要图片中目标的位置信息和类别，因此需要对采集的图片进行标注。在前期主要采用人工标注的方式，在后期分析测试异常视频后，在针对性的使用半自动标注的方式添加相应场景的数据。标注采用的工具是 LabelImg，在做目标检测的任务时，常常会使用它来进行标注数据。本文的数据集采用 VOC 格式标注，在标注完成后，每张图片都会生成一个对应的 XML 文件。一个 XML 文件包含了图片的名字、格式和目标物体的类别和位置。生成的 XML 文件如图 3.8。

```
<annotation>
  <folder>nonuva_carv2.0</folder>
  <filename>14800.jpg</filename>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>car</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>1157</xmin>
      <ymin>10</ymin>
      <xmax>1624</xmax>
      <ymax>380</ymax>
    </bndbox>
  </object>
</annotation>
```

图 3.8 XML 标签信息

### 3.2.3 数据分析

由于制作的数据集白天、阴天、夜晚和遮挡等样本不均衡，因此采用了离线



的方式对数据集做了增强。分别采用了图像平移、对比度调整、添加噪声等方式，增强的数据集大概 20000 张左右。增强后的样本图 3.9。



图 3.9 训练样本

当用这个数据集训练出来的模型上线测试 RTSP 视频流的时候，发现对于车辆侧身的检测效果不是很好，如图 3.10 这种样本。



图 3.10 异常样本

通过标注文件 XML 中的车辆宽、高等信息可以算出侧身车辆的样本在数据集中占的比例。计算方式为：首先剔除图像边缘的不完整车辆，由于侧身车辆的宽肯定是大于高的。因此采用计算车辆宽高比的方法来统计含有侧身车辆的图片数量，车辆宽高比的阈值设为 1.25。统计信息如图 3.11。从图中可以看出，侧身车辆在

数据集中占的比例非常小。

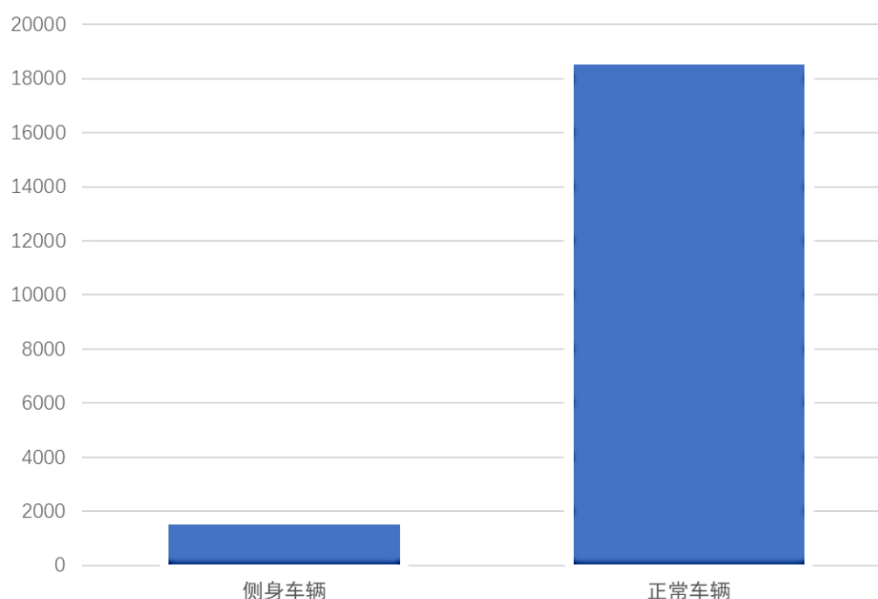


图 3.11 数据分析

对于这种情况采取了两种方式来增加侧身车辆的图片数量。第一种就是采用半自动标注的方式。首先使用一个精度好的大模型，对线上的 RTSP 流进行跳帧检测。跳帧的值如果设置得太小，那么连续的两张图片差异不大，就没有保存下来的意义。经过测试，间隔 2 秒检测一次时，保存的数据效果最好。设置一个筛选保存策略，即和前面提到的方式一样，当检测到的车辆宽高比大于 1.25 时，就把这帧图片保存下来，同时把对应的位置信息写入 XML 中。最后再人工对数据进行修正。

第二种方式是采用数据增强的方法。先搜集一部分含有侧身车辆的图片，把侧身车辆从图中抠出来，随机的选择 2 到 3 个侧身车辆，以不同尺度粘贴到不含侧身车辆的图片的空白处，同时更新位置信息到 XML 中。

这样的数据集分布就更加均衡了。最后依然用了 10000 张作为原始数据集，然后进行数据增强，尽量使得各个场景的数据分布均匀。

### 3.3 实验结果分析

#### 3.3.1 实验环境

在本节，将通过测试集的 AP、测试视频的效果和推理时间来验证本文提出的改进算法的性能。本文实验的服务器硬件配置为 3 张 NVIDIA GeForce GTX 2080 Ti 独立显卡，系统环境为 Ubuntu 18.04。算法采用 PyTorch 框架实现，使用 PyTorch1.6 版本，CUDA（Compute Unified Device Architecture）版本为 10.2。在本章的所有实验，都采用同一个数据集，将数据集按 7:2:1 的比例随机划分训练集

、验证集和测试集。

### 3.3.2 评价指标

在车辆检测任务中,需要同时做到车辆定位与识别,因此评价指标要素复杂。由于本章只对一类物体检测,所以采用 AP (average precision) 来评估实验的结果。这里采用 IoU 的方式来度量预测框的准确度,一般情况下,当  $\text{IoU} > 0.5$ , 就认为检测正确,对于精度要求比较高的情况,阈值也可以设为 0.75、0.9 等。接下来介绍 AP 计算涉及到的 TP、FP、FN、Precision、Recall 和 P-R 曲线等概念。

由于涉及的目标只有车辆,那么分类目标只有两类,是车(正样本, Positive)和非车(负样本, Negative),对其做如下定义:

TP (True Positives): 当  $\text{IoU} > 0.5$  时,即正确的将车识别为车。

FP (False Positives): 当  $\text{IoU} < 0.5$  时,即错误的将非车识别为车。

FN (False Negatives): 表示漏检的车辆。

P-R (Precision-Recall) 曲线的横纵轴分别为 Precision (准确率) 和 Recall (召回率), 计算公式如下:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3-1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3-2)$$

AP 就是 P-R 曲线所围成的面积。

### 3.3.3 实验对比与分析

#### (一) 主观对照

由于本章算法是基于 YOLOv4 做出的改进,因此将原始的 YOLOv4 与本章提出的改进 YOLOv4 的进行实验对比。两个模型的参数量如图 3.10 所示,测试集的部分检测样本结果如图 3.12 所示,左边是原始的 YOLOv4 的测试结果,右边是改进后的 YOLOv4 结果。

从表 3-2 可以看出,改进后 YOLOv4 在参数量、BFLOPs 和模型尺寸上都远远小于原始的 YOLOv4。从图 3.12 的测试图片来看,两者的效果也相差不大。改进后的 YOLOv4 在黑夜、遮挡等情况下依然能达到很好的效果。

表 3-2 两种模型的计算量

模型	参数总量	BFLOPs	模型尺寸
YOLOv4	52515500	60.1	246.19
改进 YOLOv4	5576548	8.2	10.9

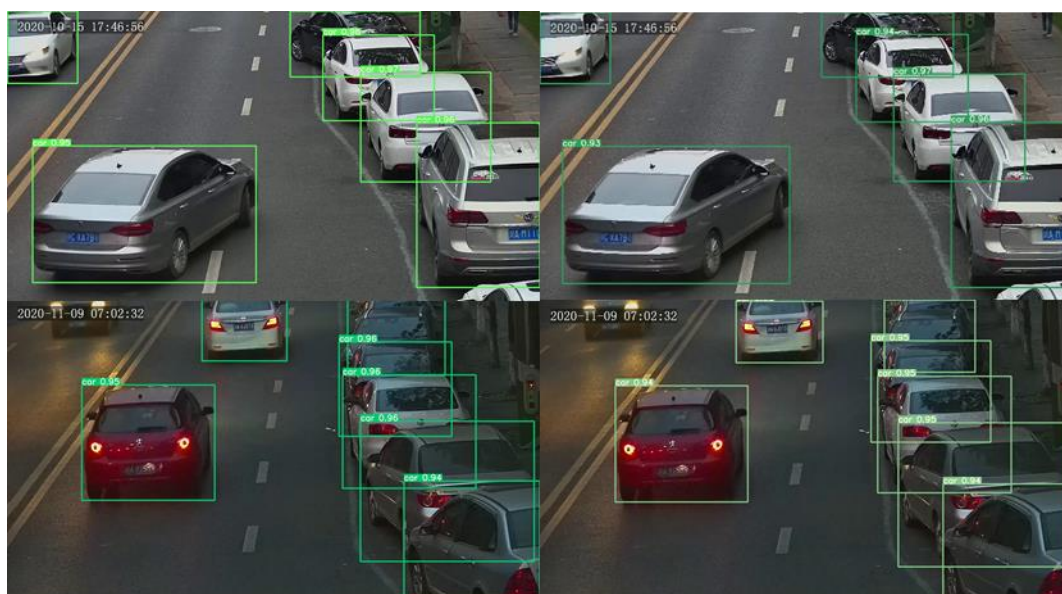


图 3.12 YOLOv4 与改进的 YOLOv4 实验对比

## (二) 客观对照

为了评估本章提出的改进的 YOLOv4 算法，将该算法与其它的检测算法进行对比。由于本章提出的算法是基于 YOLOv4 改进的，故与 YOLOv4 进行对比。同时，与目前最新的算法也进行了对比，为了保证实验的公平性，上述所有算法均使用本章标注的数据集，同时为了验证数据增强的效果，分别对每个算法先后在原始数据集和增强后的数据集上训练，并且有着相同的实验配置环境。其对比的实验结果如表 3.3。

表 3.3 本章算法与现阶段算法对比结果

算法	Backbone	Data Augmentation	AP	Inference
DetectNetv2	Resnet18	False	81.3	0.002
		True	87.1	
YOLOv3	Darknet53	False	84.3	0.013
		True	92.4	
YOLOv4	CSPDarknet53	False	87.2	0.016
		True	97.3	
NanoDet	ShuffleNetV2	False	77.3	0.007
		True	87.6	
RepPointsv2	Resnet50	False	83.4	0.088
		True	93.7	
EfficientDet-D0	EfficientNet	False	81.6	0.010
		True	90.3	
YOLOv4	MobileNetv3	False	85.2	0.009
		True	94.2	

从表 3.2 中可以看出，改进后的 YOLOv4 在 AP 上比原始 YOLOv4 减小了



3%，但是在单张图片的推理速度上提升了 0.007 秒。相比于轻量级网络 DetectNetv2、NanoDet，改进的 YOLOv4 在 AP 上非常有优势，但是在速度上低于 DetectNetv2，高于 NanoDet。同时，训练数据的均衡对于 AP 值影响很大，证明一个分布均衡的数据集对模型是非常重要的。

综合上表中各个算法的精度和推理速度，在 AP 相差不大的情况下，改进后的 YOLOv4 推理速度明显的占据优势，这说明本章所做的改进是有效的。

### 3.4 工程实测

#### 3.4.1 智慧停车系统

本章提出的车辆检测模型应用于“智慧停车”项目，整个系统如图 3.13。

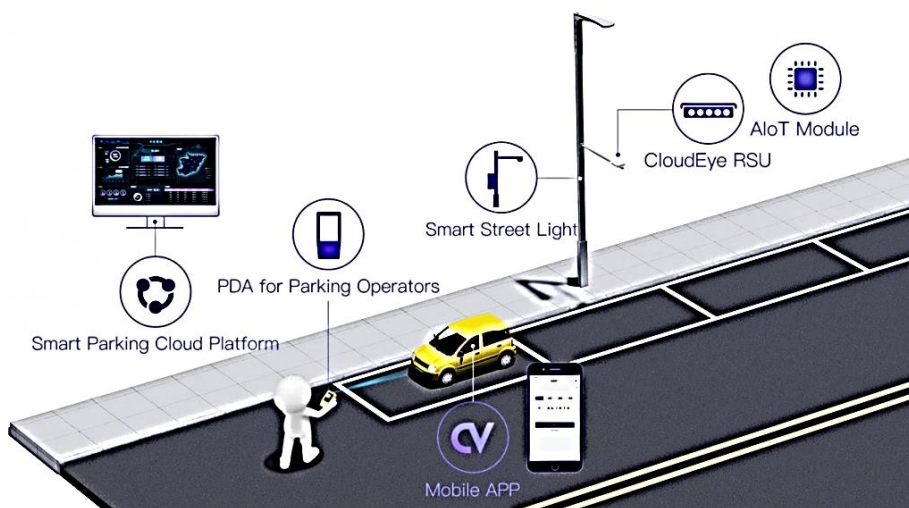


图 3.13 路边智慧停车系统

位于灯杆上的网络摄像头对准路边停车位，然后将采集到 RTSP 视频流通过局域网传输到位于灯杆底部机箱的边缘设备中。边缘设备上部署了两部分，一部分是深度学习模型，包括车辆检测、车牌检测、车牌识别等模型。另一部分是业务逻辑部分，主要是根据模型检测的结果来对业务进行判定，包括进出场行为、停车时长、违规停车等等。然后，边缘设备会将得到结果传到服务器端，完成停车费用的计算。最后，服务器端向用户手机上的 APP 客户端发送停车时长、停车费用等信息，由用户通过手机支付停车费用。

#### 3.4.2 业务逻辑

整个业务逻辑如图 3.14。首先由车辆检测模型对从摄像头获取的 RTSP 视频流进行实时检测。其次将检测到车辆从原图中截取出来，送入车牌检测模型进行车牌检测。最后将检测到的车牌图截取出来，送入车牌识别模型进行车牌识别。当车辆的 Bounding Box 与车位框的 IoU 大于一个阈值（在一定时间范围内），

就判定为进场（进入车位），同时会给车辆一个数字 ID，并记录上传车牌号。当记录了数字 ID 和车牌号的车辆的 BoundingBox 与车位框的 IoU 小于一个阈值，就判定为离场，然后上传离场时间，同时清理数字 ID。服务器会根据上传的进场时间和离场时间向用户发送收费信息。这样就完成了整个停车、收费的行为，且全程无人工干预，真正做到停车智能化、便捷化。

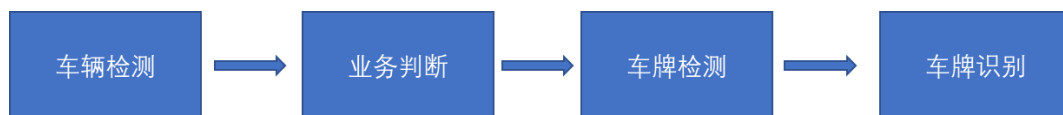


图 3.14 业务逻辑

### 3.4.3 上线实测

将车辆检测、车牌检测、车牌识别等模型经过剪枝、量化等压缩操作，部署到边缘设备上运行。通过不间断的运行 48 小时来评估各个模型的准确率以及停车订单的准确率。测试的场景如图 3.15。

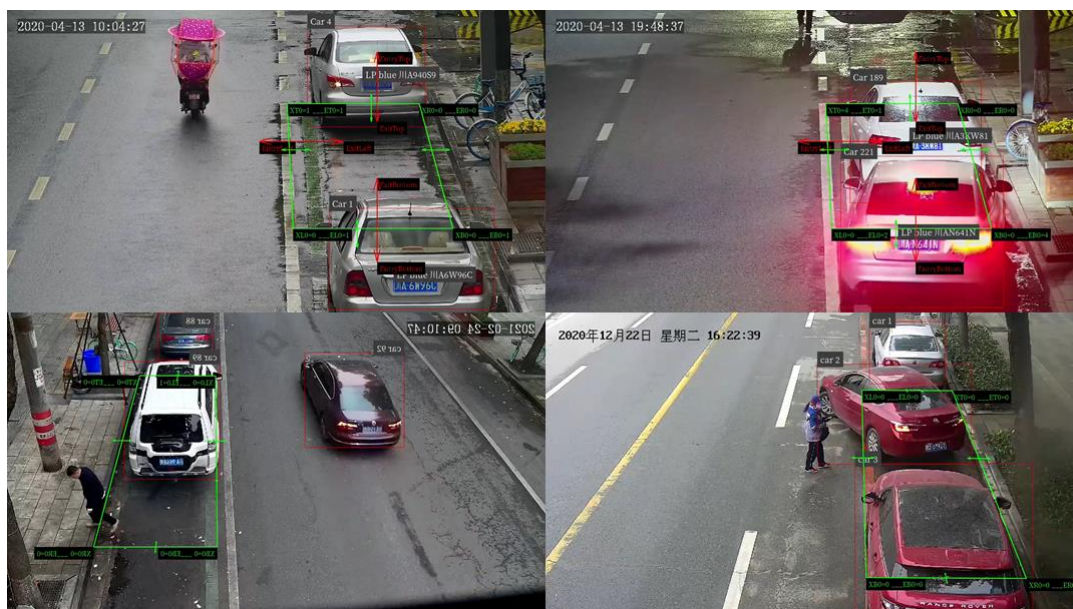


图 3.15 实测场景

由于收集的训练数据不可能覆盖所有场景，为了测试算法的鲁棒性，测试场景的选择避开了收集数据所用的几路摄像头。经过连续运行 48 小时，共向服务器端上传了 148 个订单信息。然后根据服务器上的订单信息，对比运行日志、原始视频和结果视频来统计准确率。准确率如图 3.16。其中车辆的检测的准确率依然达到了 92%，可以看出算法的鲁棒性非常好，符合实际应用的要求。

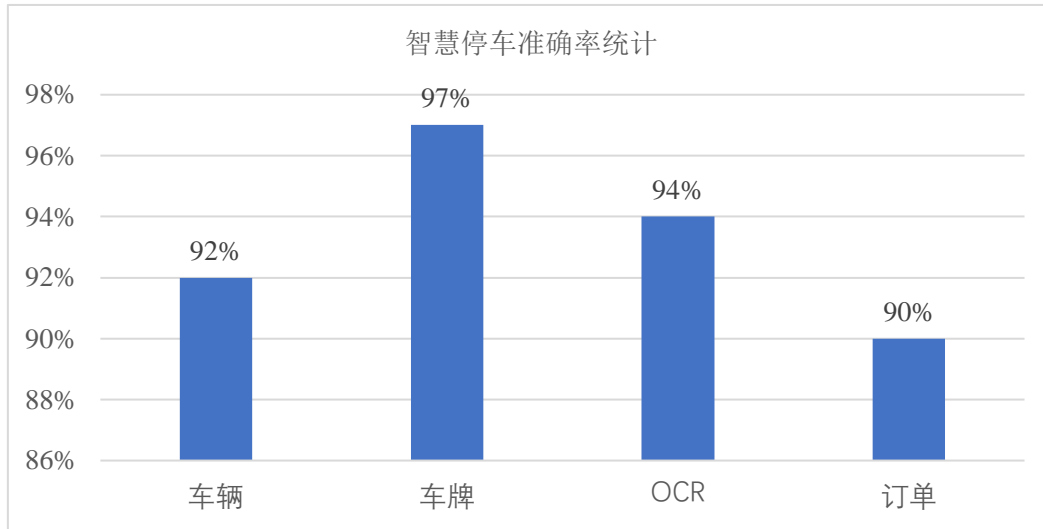


图 3.16 准确率统计

### 3.5 本章小结

本章结合实际应用场景，在 YOLOv4 的基础上做了一些改进。在不影响检测精度的前提下，通过更换轻量级的网络做特征提取，同时将普通卷积替换为深度可分离卷积，提升检测速度。其次，考虑一些边缘设备目前还不支持 Mish 激活函数，将 Mish 激活函数替换为 Leaky Relu 激活函数。然后还搜集制作了用于车辆检测的数据集，通过一些数据增强的方法使数据集的分布更均衡。经过实验证明，本章改进的 YOLOv4 与原始 YOLOv4 相比，在参数量、计算量、检测速度上都优于对方。还通过对比其它的一些网络，来验证本章提出的改进 YOLOv4 算法。

其次，将车辆检测模型部署到智慧停车项目中，经过实际场景进行测试，证明本章提出的基于 YOLOv4 的车辆检测模型具有实际的工程应用价值。

## 4 基于改进CNN的车辆识别方法

车辆的识别属于图像分类任务，即把车辆按照不同品牌、车系、车型、颜色进行分类。

本章基于深度学习的方法，在考虑边缘端算力有限的情况下，对轻量级网络 PeleeNet 进行了改进，在降低参数量的情况下提升分类的准确度，实验表明，改进的 PeleeNet 比原始的 PeleeNet 性能更为优良。

### 4.1 分类网络设计

车辆识别的基本思想是：对于第一阶段 YOLOv4 从复杂的背景下把车辆检测出来，作为车辆识别模型的输入。车辆识别模型简单来讲就是一个分类任务，输入图片经过主干网络进行特征提取，然后在通过一个 Softmax 进行分类。

本文主要参考轻量化网络 PeleeNet，在此基础上引入 Squeeze-and-Excitation (SE)模块和 Cross Stage Partial Network(CSPNet)。

#### 4.1.1 CSPNet

CSPNet 设计的目的是为了降低网络的计算量，提升准确率和推理速度。CSPNet 论文中提到推理计算量过高是因为在网络优化过程中，存在大量重复的梯度信息。于是通过将基础层的特征图划分为 2 个部分，把梯度流分开，使得梯度流在不同的网络路径传播，然后再通过跨层连接将两个部分融合起来，在减少计算量的情况下又能提升准确率。CSPNet 是一种优化思想，能够轻易的与 ResNet、DenseNet、PeleeNet 和 DarkNet 等特征提取网络相结合。

#### 4.1.2 SE模块

SENet (Squeeze-and-Excitation Networks) 获得了 2017 年的 ImageNet 分类比赛冠军，其提出的 SE 模块简单，易于实现，能够轻松的插入到现有的网络模型中。SE 其实是一种通道注意力机制，通过跨通道的连接将各通道中重要的特征进行强化，即通过学习各通道之间的相关性，来获得每个特征通道的重要程度，根据这个重要程度来提升对任务有用的特征，并抑制对任务作用不大的特征<sup>[63]</sup>。如图 4.1 为 SE 块的结构图，主要分为 Squeeze (压缩) 过程和 Excitation (激励) 过程。其中， $F_{tr}$  为卷积结构， $X$  是  $F_{tr}$  的输入， $U$  是  $F_{tr}$  的输出。



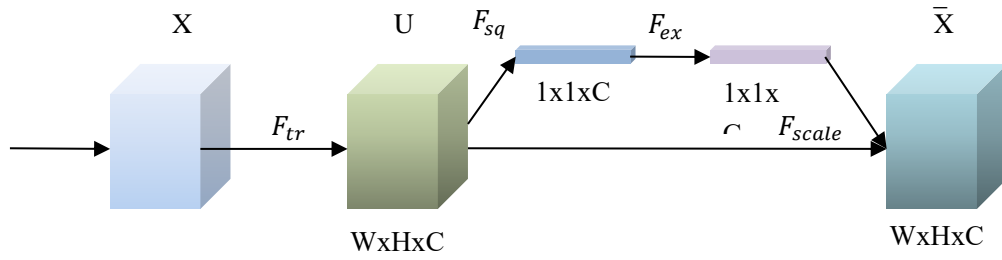


图 4.1 SE 块结构图

现在的特征提取网络一般是由一个一个 Block 构成，SE 模块可以加在任意一个 Block 结束的位置，进行一个信息的提炼。首先是 Squeeze 操作，仅包含一个全局平均池化（Global Average Pooling），如图 4.2（左）。输入特征图大小为  $W \times H \times C$ ，经过 Squeeze 操作后，特征图被压缩为  $1 \times 1 \times C$ ，这样可以利用基于全局信息的各个通道间的相关性来计算通道权重。

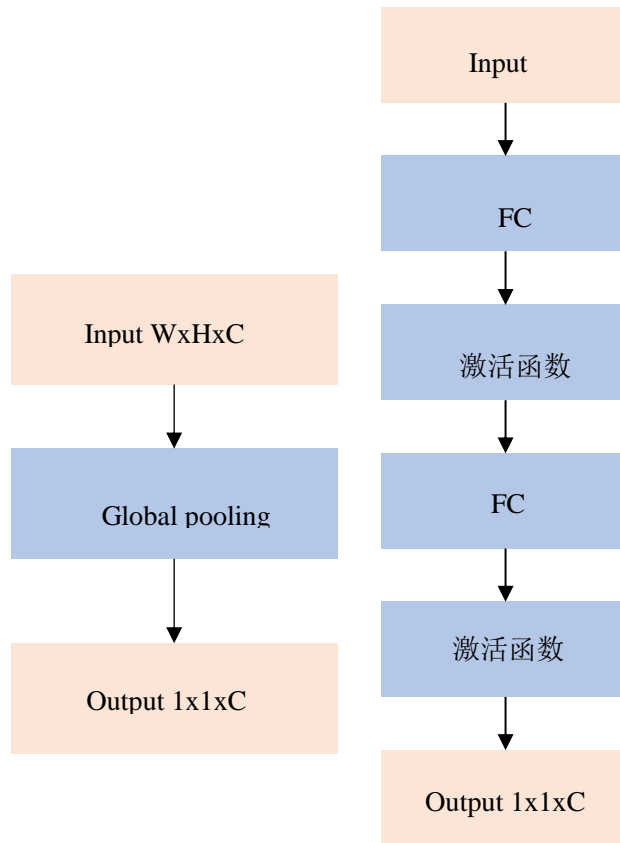


图 4.2 Squeeze 和 Excitation 操作

通道权重计算公式为：

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^W \sum_{j=1}^H u_c(i, j) \quad (4-1)$$

其中， $u_c$  表示  $U$  中第  $c$  个通道的二维矩阵。

之后就是 Excitation 过程，如图 4.2（右），包含两个全连接层和两个激

活函数。第一个全连接层是为了减少通道数来降低计算量，输入到 Excitation 的通道数为  $C$ ，经过第一个全连接层时会将通道数压缩为  $C/r$  ( $r$  为压缩比例)。其次经过一个 ReLU 激活函数。第二个全连接层将通道数由  $C/r$  变为  $C$ ，即恢复通道数以保持和输入到 Excitation 的通道数一致。最后再通过 Sigmoid 函数来输出每个特征图的权重。计算公式为：

$$s = F_{cs}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \quad (4-2)$$

其中， $s$  是 Excitation 操作的输出， $z$  是前面 Squeeze 操作的输出， $W_1$  和  $W_2$  分别时两个全连接层的权重。

最后进行 scale 操作，将 SE 模块输出的通道权重分别与原特征图对应通道的二维矩阵相乘。计算公式为：

$$x_c = F_{scale}(u_c, s_c) = u_c s_c \quad (4-3)$$

由此可以理解 SE 模块的实现过程：对于一个特征图，SE 模块通过学习一个与特征图通道数一致的一维向量来作为各个通道的权重值，并将这个权重值作用到特征图对应的通道上。

#### 4.1.3 PeleeNet

PeleeNet 是一种基于 DenseNet 的轻量化网络变体，由 Stem Block 和 Two-Way Dense Layer 组成。

**Stem Block:** 对输入图像进行一次降采样 ( $\text{stride}=2$ ) 和增加通道数，结构如图 4.3 所示。

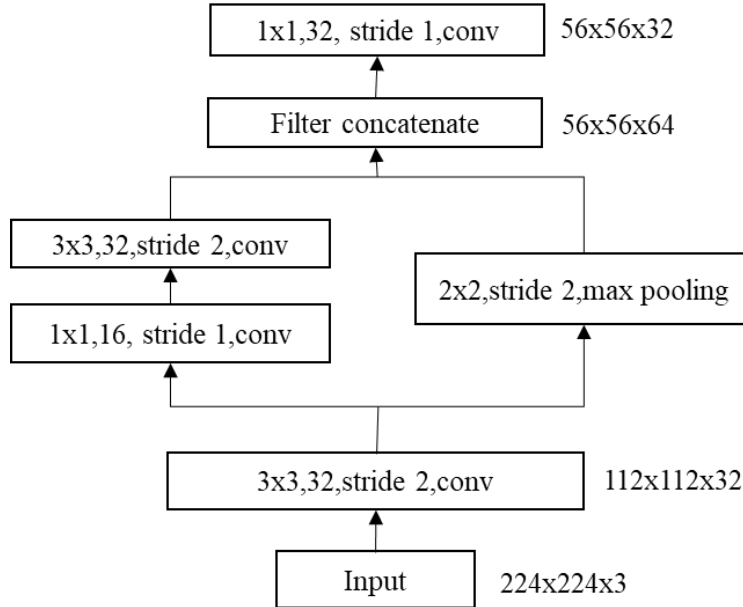


图 4.3 Stem Block

首先利用  $3 \times 3$  卷积对输入图像做一个简单的提取特征操作。然后分为两条支路进行不同的操作，一条路进行  $3 \times 3$  卷积操作，另一条路进行最大值池化操作。

最后在将两条支路的输出拼接起来。

**Two-Way Dense Layer:** 受 Inception 结构的启发, 由两路分别捕捉不同尺度感受野信息的网络分支构成。第一路经过一层  $1 \times 1$  卷积完成 bottleneck 之后, 再经过一层  $3 \times 3$  卷积; 第二路则在 bottleneck 之后, 再经过两层  $3 \times 3$  卷积, 结构如图 4.4。

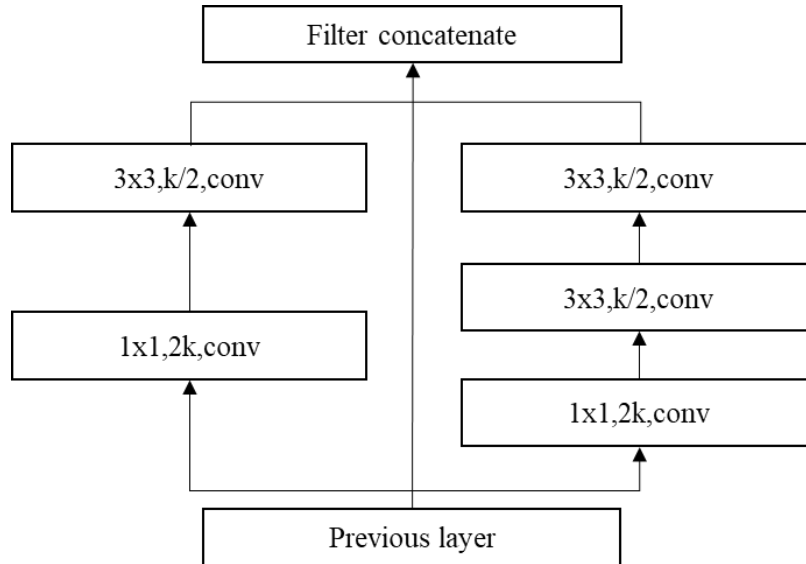


图 4.4 Two-Way Dense Layer

**Transition Layer without Compression:** 过渡层 (transition layer) 的输入输出通道数保持一致, 即为 dense group 中最后一个 dense block 的输出通道数 ( $\text{in\_ch} + n * \text{growth\_rate}$ )。

**Composite Function:** 采用 post-activation 结构, 替换 DenseNet 中的 pre-activation 结构。因而在 inference 阶段, BN 层和卷积层可以融合在一起, 以提升推理速度。

PeleeNet 分类网络的总体结构如下表 4.1。包含一个 StemBlock 和四个特征提取器。每个特征提取器都包含一个 Dense Block 和过渡层。除了最后一个过渡层, 每个过渡层采用  $1 \times 1$  卷积进行跨通道信息整合, 采用  $2 \times 2$  的平均值池化操作对特征图进行下采样。在分类层采用了  $7 \times 7$  的全局平均池化, 来解决全连接层的固定输入问题, 同时也能减少参数量。

表 4.1 PeleeNet

Stage		Layer	Output Shape
Input			224 x 224 x 3
Stage 0	Stem Block		56 x 56 x 32
Stage 1	Dense Block	DenseLayer <b>x 3</b>	28 x 28 x 128
	Transition Layer	1 x 1 conv, stride 1 2 x 2 average pool, stride 2	
Stage 2	Dense Block	DenseLayer <b>x 4</b>	14 x 14 x 256
	Transition Layer	1 x 1 conv, stride 1 2 x 2 average pool, stride 2	
Stage 3	Dense Block	DenseLayer <b>x 8</b>	7 x 7 x 512
	Transition Layer	1 x 1 conv, stride 1 2 x 2 average pool, stride 2	
Stage 4	Dense Block	DenseLayer <b>x 6</b>	7 x 7 x 704
	Transition Layer	1 x 1 conv, stride 1	
Classification Layer		7 x 7 global average pool	1 x 1 x 704
		1000D fully-connecte,softmax	

#### 4.1.4 CSPPeleeNet-SE

在 PeleeNet 的基础上，引入 CSPNet 结构和 SE 注意力机制。CSPPeleeNet-SE 具体的结构如图 4.5 所示。

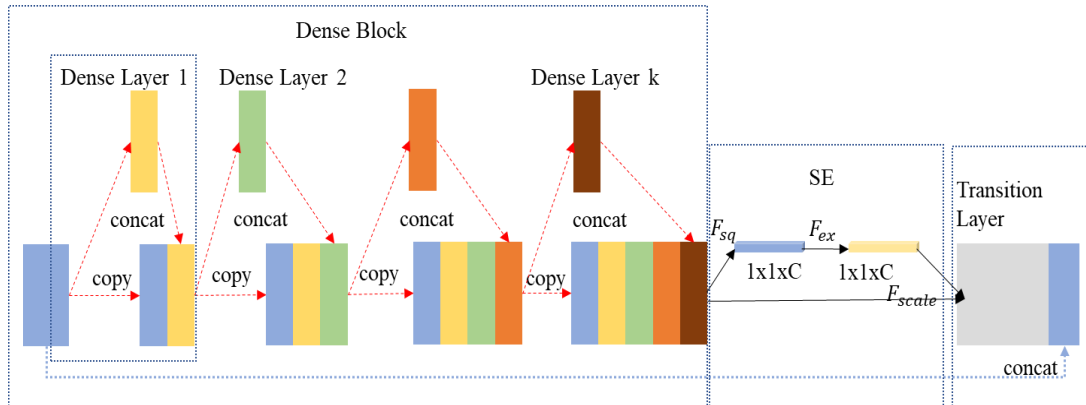


图 4.5 CSPPeleeNet-SE 结构图

在将特征图输入到 DenseBlock 之前，将特征图分为两个部分，一部分经过 Dense Block 进行特征提取，然后送入 SE 模块学习各通道间的相关性，最后送入过渡层。另一部分直接越过中间的 Dense Block、SE，直接与过渡层的输出进行拼接操作。

## 4.2 实验结果分析

### 4.2.1 数据集介绍

车辆分类的数据采集比较困难，因此使用的是开源的数据Stanford Cars[192]。Stanford Cars数据集主要用于细粒度分类任务。该数据集中一共包含196类汽车的

16185 张汽车图片，其中 8144 张为训练集，8041 张为测试集，其中每个类别大概 40 张左右。图片样本如图 4.6 所示。



图 4.6 图片样本

#### 4.2.2 实验环境

本文实验的服务器硬件配置为 NVIDIA GeForce GTX 2080 Ti 独立显卡。软件环境为 Ubuntu 18.04 LTS 64 位系统，算法基于 keras 框架，使用 keras2.2.4 版本，CUDA（Compute Unified Device Architecture）版本为 10.0。

#### 4.2.3 实验对比与分析

##### （一）训练

使用预先在 imagenet 数据集上训练的权重，并通过迁移学习来训练模型。所有层将进行微调，最后一个完全连接层将完全替换成 196 类。用于训练的技巧如下：

**Cyclic Learning Rate:** 即在让学习率在一个区间内周期性地增大和缩小。

**Data Augmentation:** 由于用于训练的每一类的车辆图片只有 40 张左右，因此通过随机裁剪，水平翻转，旋转，剪切，AddToHueAndSaturation, AddMultiply, GaussianBlur, ContrastNormalization, 锐化，浮雕等方法进行数据扩充。

**Cross-validation 5 folds:** 数据集分为 5 份，每份有 20% 的数据。相当于把训练分为 5 步，每一步都选取 1 份作为验证集，剩下的 4 份作为训练集，重复这个过程，直到每一份被用作验证集。

##### （二）实验对比

表 4.2 是改进的 CSPPeleeNet-SE 网络与目前主流的分类网络的对比，从表中可以看出，CSPPeleeNet-SE 相对于改进前的 PeleeNet，分类准确率涨了大概 1%，参数增加了 1%，但是在加入了 CSPNet，参数虽然增加了，但运算量 BFLOPS 减小了 13%。其次，验证了 Swish 激活函数比 ReLU 激活函数的效果更好。这说明本章对 PeleeNet 的改进是有效的。

表 4.2 对比实验

Model	Parameter	BFLOPs	Top-1	Top-5
PeleeNet	2.79M	1.017	71.7	91.0
PeleeNet-swish	2.79M	1.017	72.5	91.7
PeleeNet-swish-SE	2.81M	1.017	73.1	92.0
CSPPeleeNet	2.83M	0.888	71.9	91.2
CSPPeleeNet-swish	2.83M	0.888	72.7	91.8
CSPPeleeNet-swish-SE	2.85M	0.888	73.4	92.1
EfficientNet-B0	4.81M	0.915	72.3	91.4
MobileNet-v2	3.47M	0.858	69.3	88.7
CSPMobileNet-v2	2.51M	0.764	68.7	88.3
CSPDenseNet	3.48M	0.886	66.4	87.3
ResNet-10	5.24M	2.273	63.5	85
CSPResNet-10	2.73M	1.905	65.3	86.5
Darknet53	41.57M	18.57	73.2	93.6
ResNet-50	22.73M	9.74	76.8	92.4

### 4.3 工程应用

本章提出的车辆识别算法将用于“AI 值守”项目中。但由于车辆识别的数据收集比较困难费时，目前只能通过使用公共数据集来完成算法的验证。

车辆识别的流程是先使用车辆检测模型将车辆从复杂的背景图中截取出来，然后再送入车辆识别模型进行车辆的分类识别。目前已完成算法的设计、模型的压缩量化、以及业务逻辑的编写。

### 4.4 本章小结

本章在 PeleeNet 上的基础上做出了一些改进，目的是为了提高车辆分类准确度的同时减少模型的计算量，因此引入了 CSPNet 的思想、SE 通道注意力机制和 Swish 激活函数。由于数据量太少，又采用了旋转、平移、剪切、添加噪声等方式进行数据扩充。在训练的时候，采用 5 折交叉验证的方法进行训练。经实验证明，本章提出的 CSPPeleeNet-SE 虽然参数量比 PeleeNet 多一点，但在计算量、准确率等指标都优于 PeleeNet。

## 5 总结与展望

### 5.1 工作总结

随着智能交通的兴起，国内外众多的企业对此投入了大量的人力、物力来研发智能交通应用，比如百度、谷歌的自动驾驶，海康的智慧停车，等。而车辆检测与识别是构建智能交通系统的核心技术之一，具有非常大的应用前景和研究价值。

因此，本文结合实习单位的项目，对基于深度学习的车辆检测与识别进行研究。主要工作如下：

(1) 对深度学习的历史进行了回顾，阐述深度学习中的卷积神经网络的理论基础、实现原理和优缺点总结。主要包括卷积神经网络的卷积层、几种常用的激活函数、池化层以及以 VGG, ResNet 为代表的基础网络，对近年来快速发展的深度学习目标检测算法进行原理介绍和优缺点分析。包括以 R-CNN 为代表的两阶段检测算法和以 YOLO 系列和 SSD 为代表的一阶段检测算法。

(2) 针对传统的目标检测不能及时和准确响应多变的交通环境问题，本文选择一阶段检测算法来实现车辆检测，主要选择目前性能最为优良的 YOLOv4 框架为基准，结合实际项目应用，以压缩模型提高速度为目标，构建快速高效的车辆检测模型。通过从算法框架和数据扩增策略等方面进行改进，最终在检测精度与 YOLOv4 相差不大的情况下，使车辆检测模型的推理速度提高了 37%。

(4) 对于车辆识别同样以保障精度为前提的情况下，提升分类网络的速度。通过对 PeleeNet 引入 CSPNet、SE 注意力机制、Swish 激活函数等策略，最终 BFLOPs 减少了 13%，TOP-5 精度提高了 1 个百分点。

### 5.2 研究展望

在接下来的研究工作中，我会更加关注模型的压缩量化、部署等方面。模型部署一般部署在高性能服务器上和一些边缘设备，这个可以更具业务需求来区分。对于一些不需要实时响应，模型调用量不大场景，比如门禁的人脸识别、停车场的车牌识别等，部署在服务器端就能满足业务需求。但对于一些要求实时检测，模型需要 24 小时不断运行的场景，用边缘设备更为适合。但边缘设备算力有限，如果直接把训练好的模型部署在上面运行，无法做到实时检测。所以通常会对模型做加速。

在硬件上，有很多减少模型推理时间的框架，比如 Nvidia 推出的基于 GPU 加速的 TensorRT 计算框架，Inter 推出的基于 CPU 加速的 OpenVINO 计算架构，还有近几年又开始使用嵌入式设备来进行加速运算。

在模型上，主要采用剪枝和量化的方法。剪枝就是剔除掉模型中不重要的参数，主要有层剪枝和通道剪枝，剪枝完后的模型还需要重新训练一次，回调因剪枝而丢失的精度。模型量化是将训练好的模型权重从 Float32 转换为低精度的 Int8 存储，从而达到减少模型内存消耗、提升推理速度。



## 参考文献

- [1]张辉. 邢台居住区机动车停车位配建标准研究与分析[D].燕山大学,2016.
- [2]陈秀娟.全国汽车保有量达 2.81 亿辆[J].汽车观察,2021(01):7.
- [3]岳颖,程书波.中国道路交通事故原因甄别与对策建议[J].科技与创新,2021(04):21-24.
- [4]李骏,王永军.中国智慧城市、智慧交通和智能汽车（SCSTSV）——发展战略、系统构架和应用（英文）[J].汽车文摘,2021(03):1-7.
- [5]杨建才, 交通安全管理 智能交通建设. 赵指真 主编,宾川年鉴,云南出版集团云南人民出版社,2020,247-248,年鉴.
- [6]杨恩泽. 基于深度学习的交通车辆检测与识别算法研究[D].北京交通大学,2019.
- [7]谭光兴,孙才茗,王俊辉.基于 HOG 特征与 SVM 的视频车辆检测系统设计[J].广西科技大学学报,2021,32(01):19-23+30.
- [8]江昆鹏. 基于深度学习的车型识别与车辆检索研究[D].河南工业大学,2020.
- [9]Lindeberg T. Scale invariant feature transform[J]. 2012.
- [10]Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, 1: 886-893.
- [11]Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008, 110(3): 346-359.
- [12]Papageorgiou C P, Oren M, Poggio T. A general framework for object detection[C]//Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271). IEEE, 1998: 555-562.
- [13]Optimization S M. A fast algorithm for training support vector machines[J]. CiteSeerX, 1998, 10(1.43): 4376.
- [14]P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, Sept. 2010, doi: 10.1109/TPAMI.2009.167.
- [15]Bayes F R S. An essay towards solving a problem in the doctrine of chances[J]. Biometrika, 1958, 45(3-4): 296-315.
- [16]LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [17]Krizhevsky A , Sutskever I , Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012.
- [18]Simonyan K,Zisserman A.Very deep convolutional networks for large-scale image

- recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [19]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.
- [20]Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.
- [21]Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848-6856.
- [22]张慧,王坤峰,王飞跃.深度学习在目标视觉检测中的应用进展与展望[J].自动化学报,2017,43(08):1289-1305.
- [23]张璧程. 基于区域卷积神经网络的目标检测与识别算法[D].电子科技大学,2020.
- [24]Neubeck, A. and L. Gool. “Efficient Non-Maximum Suppression.” 18th International Conference on Pattern Recognition (ICPR'06) 3 (2006): 850-855.
- [25]段仲静,李少波,胡建军,杨静,王铮.深度学习目标检测方法及其主流框架综述[J].激光与光电子学进展,2020,57(12):59-74.
- [26]Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 580–587.
- [27]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.
- [28]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.
- [29]He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.
- [30]Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [31]Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.
- [32]Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
- [33]Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of

- the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.
- [34]Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 734-750.
- [35]Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6569-6578.
- [36]姜康. 基于深度学习的车辆检测算法研究[D]. 吉林大学, 2020.
- [37]王涛. 视频内容理解研究与应用[D]. 兰州理工大学, 2019.
- [38]周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(06): 1229-1251.
- [39]张顺, 龚怡宏, 王进军. 深度卷积神经网络的发展及其在计算机视觉领域的应用[J]. 计算机学报, 2019, 42(03): 453-482.
- [40]姜萌萌. 鱼群密度光学检测系统的研究[D]. 燕山大学, 2018.
- [41]魏秀参. 解析卷积神经网络——深度学习实践手册[M]. 南京大学出版社: 江苏, 2017: 31.
- [42]姜万录, 刘庆平, 刘涛. 神经网络学习算法存在的问题及对策[J]. 机床与液压, 2003(05): 29-32.
- [43]林景栋, 吴欣怡, 柴毅, 尹宏鹏. 卷积神经网络结构优化综述[J]. 自动化学报, 2020, 46(01): 24-37.
- [44]葛道辉, 李洪升, 张亮, 刘如意, 沈沛意, 苗启广. 轻量级神经网络架构综述[J]. 软件学报, 2020, 31(09): 2627-2653.
- [45]田晓明. 基于深度学习的目标检测技术研究及其应用[D]. 湖南大学, 2018.
- [46]黄健, 张钢. 深度卷积神经网络的目标检测算法综述[J]. 计算机工程与应用, 2020, 56(17): 12-23.
- [47]Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [48]Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. arXiv preprint arXiv:1506.01497, 2015.
- [49]Buzcu I, Alatan A A. Fisher-selective search for object detection[C]//2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016: 3633-3637.
- [50]张索非, 冯烨, 吴晓富. 基于深度卷积神经网络的目标检测算法进展[J]. 南京邮电大学学报(自然科学版), 2019, 39(05): 72-80.
- [51]周伟伟. 基于道路交叉口的分辨率遥感影像道路提取[D]. 武汉大学, 2018.
- [52]陆思文. 交通场景中基于深度学习的视频图像处理算法及应用研究[D]. 东南大学, 2019.
- [53]徐凡. 基于计算机视觉的铁路场景识别和扣件定位方法研究[D]. 北京交通大学, 2018.
- [54]唐聪, 凌永顺, 郑科栋, 杨星, 郑超, 杨华, 金伟. 基于深度学习的多视窗 SSD 目标检测方法[J]. 红外与激光工程, 2018, 47(01): 302-310.
- [55]赵庆北. 改进的 SSD 的目标检测研究[D]. 广西大学, 2018.

- [56]Redmon J, Farhadi A.YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
- [57]Redmon J, Farhadi A.Yolov3: An incremental improvement[J].arXiv preprint arXiv:1804.02767, 2018.
- [58]Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.
- [59]Yang J, Fu X, Hu Y, et al. PanNet: A deep network architecture for pan-sharpening[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5449-5457.
- [60]Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391.
- [61]Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1314-1324.
- [62]陈建强,刘明宇,符秦沈,姚卓荣.基于深度学习的热轧钢带表面缺陷检测方法[J].自动化与信息工程,2019,40(04):11-16+19.
- [63]冯雨,易本顺,吴晨玥,章云港.基于三维卷积神经网络的肺结节识别研究[J].光学学报,2019,39(06):256-261.

## 致谢

时光匆匆，短暂又漫长的三年研究生生活接近了尾声。回顾我在四川师范大学生活学习的七年时光中，从懵懂的大一新生，到即将毕业踏入工作岗位的新人，期间收获的成长和经历是我人生当中宝贵的财富。在此我将最诚挚的谢意致以一路走来尤其是研究生期间给予我帮助和鼓励的老师、同学以及家人，你们的引导、陪伴和支持是我前进的动力。

首先要感谢的就是在研究生期间指导我研究工作的导师廖磊，廖磊老师在学术研究中为我指引方向，更是指导我为人处世的人生导师。

其次要感谢的是在 305 实验室中一起工作学习的同学们，感谢你们的陪伴和帮助。

最后要感谢一直以来默默付出，支持我、鼓励我的父母，为我提供了良好的学习和生活环境。我一定不负你们的期望，照顾好自己，独当一面，也衷心希望你们能够健康快乐地享受生活。

再次感谢所有陪伴我走过学生生涯的人们以及四川师范大学的老师和职工们，祝你们生活幸福，事业顺利。