

分类号: [\[单击此处键入分类号\]](#)

密 级： 公开/秘密

UDC: _____

单位代码: 10142

沈阳工业大学
 硕士学位论文

多种道路场景下的轻量化道路车辆细粒度检测技术研究

—— [单击此处继续键入副题名或删除此提示]



学 号: 202120698

作者：赵乃贤

学位类别: [\[单击此处键入学位类别\]](#)

领域: [\[工程硕士单击此处键入领域\]](#)

论文类型: [\[单击此处键入论文类型\]](#)

20××年××月××日

[单击此处键入论文中文题目]

单击此处键入作者姓名

沈阳工业大学

沈阳工业大学硕士学位论文

多种道路场景下的轻量化道路车辆细粒度检测技术研究

Research on fine-grained detection technology of lightweight road vehicles in various road scenarios

作 者： 赵乃贤 单位： 沈阳工业大学

指 导 教 师： 彭曦霆 副教授 单位： 沈阳工业大学

协助指导教师： [导师姓名] [导师职称] 单位： [导师所在单位]

[导师姓名] [导师职称] 单位： [导师所在单位]

论文答辩日期： 20××年××月××日

学位授予单位： 沈 阳 工 业 大 学

独 创 性 说 明

本人郑重声明：所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得沈阳工业大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

签名：_____ 日期：_____

关于学位论文使用授权的说明

本学位论文作者和指导教师完全了解沈阳工业大学有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权沈阳工业大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

（保密的论文在解密后应遵循此规定）

签名：_____ 导师签名：_____ 日期：_____

摘要

随着交通环境日益复杂和智能交通系统的快速发展，智能交通系统受到越来越多的关注。通过捕获道路上的车流量和车辆信息，用于自动驾驶、公路路况预测和路线规划等任务。车辆检测是获取道路上车流量和车辆信息的前提。目前主流算法都是基于深度网络的，需要大量的训练参数、资源和样本。它们无法满足车联网任务的快速响应，难以部署在计算资源有限的边缘设备上。本文提出了一种多种道路场景下的轻量化道路车辆细粒度检测技术。其主要研究内容如下：

第一：本文通过采集车辆的行车记录仪的视频，经过预处理、数据标注等过程构建了车辆检测数据集。具体来说，首先采集来自不同城市的近十台车辆的行车记录仪视频，然后通过提取关键帧将视频中的画面转化成图像，由于不同行车记录仪之间的差异性导致图像的分辨率有所不同，本文统一将提取关键帧后的图像重构成相同分辨率，随后对数据进行数据增强，确保模型的泛化性和鲁棒性。在本文的数据集中一共收集到 20814 张图像，其中的车辆类型共包含四大类。

第二：车辆分类是车辆检测的重要组成部分，想实现车辆检测的轻量化首先要实现车辆分类的轻量化，在这一部分本文使用更加轻量化的分类方法对车辆进行分类，并且将训练后得到的模型转化成开放神经网络交换（Open Neural Network Exchange, ONNX）格式，在边缘设备上完成部署。在该部分本文使用宽度学习系统对车辆进行分类，宽度学习系统网络结构更加简单，需要计算的参数量更少，同时通过使用其增量学习方法对模型进行快速迭代更新，算法模型收敛速度更快。在云端完成模型的训练后，将保存好的模型转换成 ONNX 格式模型，将其部署在用树莓派模拟的边缘端设备上。

第三：在该部分中本文将更轻量化的车辆分类网络替换到车辆检测网络中，同时整合数据预处理部分，将整个车辆检测系统在云端进行训练，并且将训练好的模型转进行保存，由于保存的模型不能直接在边缘端部署，因此将其换成 ONNX 格式在部署在边缘端设备上。具体来说，整个车辆检测系统主要流程首先是对数据进行预处理，然后将处理后的数据发送到云端训练模型，最后将训练好的模型转成 ONNX 格式部署到边缘端设备上。云端训练是在 RTX 3080ti 上来进行，使用增量学习算法实现整个模型的快速更新，而不必重新训练整个系统，这样极大程度上提升了车辆检测的效率。并且将训练好的模型转换成 ONNX 格式并且部署到 HUAWEI Atlas 500 上，这样可以避免因将任务上传到云端而造成的延迟和带宽拥塞。实验结

果表明，与其他基于深度网络的车辆检测技术相比，本文在检测精度方面与其他方法持平，但是在训练速度上本文提出的方法提高了 2 倍。

关键词：车辆检测，宽度学习系统，边缘设备，深度网络

Abstract

With the increasingly complex traffic environment and the rapid development of Intelligent Transportation System (ITS), intelligent transportation systems have received more and more attention. By capturing traffic flow and vehicle information on the road, it can be used for tasks such as autonomous driving, highway condition prediction, and route planning. Vehicle detection is the premise of obtaining traffic flow and vehicle information on the road. At present, the mainstream algorithms are based on deep networks, which require a large number of training parameters, resources, and samples. They cannot meet the fast response of vehicle networking tasks and are difficult to deploy on edge devices with limited computing resources. This paper presents a fine-grained detection technology for lightweight road vehicles under various road scenarios. Its main research contents are as follows:

First: In this paper, the vehicle detection data set is constructed through the process of preprocessing and data labeling by collecting the video of the vehicle's driving recorder. Specifically, the traffic recorder video of ten vehicles from different cities is first collected, and then the frames in the video are converted into images by extracting keyframes. Due to the differences between different traffic recorders, the resolution of the images is different. Ensure the generalization and robustness of the model. A total of 20814 images are collected in the data set of this paper, and the vehicle types in the data set include four categories.

Secondly, vehicle classification is an important part of vehicle detection. To realize the lightweight of vehicle detection, the first step is to realize the lightweight of vehicle classification. In this part, we use a more lightweight classification method to classify vehicles and convert the obtained model after training into ONNX format for deployment on edge devices. In this part, the broad learning algorithm is used to classify vehicles. The network structure of the broad learning algorithm is simpler and the number of parameters that need to be calculated is less. At the same time, by using its incremental learning method to update the model quickly and iteratively, the convergence speed of the algorithm model is faster. After completing the model training in the cloud, the saved model was converted into an ONNX format model, and it was deployed on the edge device simulated

by Raspberry PI.

Third: In this part, this paper replaces the lighter vehicle classification network with the vehicle detection network and integrates the data preprocessing part. The whole vehicle detection system is trained in the cloud, and the trained model is converted into ONNX format and deployed on the edge device. Specifically, the main process of the whole vehicle detection system is to first preprocess the data, then send the processed data to the cloud to train the model, and finally convert the trained model into ONNX format and deploy it to the edge device. The training in the cloud is simulated using the RTX 3080ti, and the incremental learning algorithm is used to realize the rapid update of the whole model without having to retrain the whole system. The trained model is converted into ONNX format and deployed on HUAWEI Atlas 500, which can avoid the delay and bandwidth congestion caused by uploads to the cloud. Experimental results show that compared with other vehicle detection technologies based on deep networks, the system improves the training speed by 2 times with similar detection accuracy.

Key Words: Vehicle Detection; Broad Learning System; Edge Devices; Deep Network

目录

摘要	I
Abstract	III
第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 车辆检测算法研究	2
1.2.2 宽度学习系统研究	3
1.3 本论文的结构安排	4
1.5 本章小结	5
第 2 章 车辆检测相关技术	6
2.1 卷积神经网络	6
2.1.1 卷积层	6
2.1.2 激活层	7
2.1.3 池化层	9
2.1.4 全连接层	10
2.2 常用车辆检测算法	10
2.2.1 二阶段车辆检测算法	10
2.2.2 一阶段车辆检测算法	12
2.3 本章小结	15
第 3 章 真实道路车辆数据集的获取与介绍	16
3.1 真实道路车辆数据集	16
3.2 数据处理方式	17
3.2.1 数据预处理	17
3.2.2 数据标注	19
3.3 本章小结	20
第 4 章 基于宽度学习系统的真实道路车辆分类	21
4.1 宽度学习系统模型结构	21

4.1.1 求解伪逆	21
4.1.2 基础模型	22
4.1.3 增强节点的增量学习	23
4.1.4 特征节点的增量学习	24
4.1.5 输入数据的增量学习	25
4.2 实验流程与结果分析	26
4.2.1 实验环境及设备	27
4.2.2 实验流程	28
4.2.3 实验指标	30
4.2.4 实验结果	30
4.3 本章小结	31
第 5 章 多种道路场景下的轻量化道路车辆细粒度检测技术研究	32
5.1 轻量化道路车辆细粒度检测技术	32
5.1.1 读取并处理数据	33
5.1.2 生成候选框	34
5.1.3 车辆分类	35
5.1.4 边缘端部署	36
5.3 实验与结果分析	37
5.3.1 实验环境及设备	38
5.3.2 实验指标	39
5.3.3 实验结果	39
5.4 本章小结	42
第 6 章 总结	43
参考文献	44
在学研究成果	49
致谢	50

第 1 章 绪论

1.1 研究背景及意义

近年来，随着中国经济的高速发展，我国机动车保有量在逐年升高，据公安部统计，截至 2023 年 9 月底，我国机动车保有量达到 4.3 亿辆^[1]；据交通运输行业发展统计，截至 2022 年末，我国公路里程为 535.48 万公里，对比上一年增加 7.41 万公里。公路密度 55.78 公里/百平方公里，增加 0.77 公里/百平方公里。车辆作为交通工具确实给生活提供了便利，但是不可否认，从上述数据来看，我国交通复杂情况日益提升，公路上的交通压力剧增，造成的交通拥堵、部分车道占用率增高，交通安全事故频发。

针对交通环境问题，智能交通系统（Intelligent traffic System, ITS）^[2]应运而生，以此实现人、车、路的紧密配合，智能交通系统融合了包括计算机技术在内的多个先进技术，建立起一种在大范围内应用的综合管理平台。近几年依托人工智能和物联网等技术的兴起，同先进的控制器、传感器、通讯技术相结合，ITS 得到迅猛发展，极大程度上缓解了交通压力，提升交通系统的安全性^{[3][4]}。

在智能交通系统中首要任务是对行驶在公路上的车辆进行快速而准确的检测和分类。在日常交通环境中，车辆本身具有较大的基数并且车辆是造成交通拥堵和交通事故主要原因之一，因此能准确且快速的检测和分类车辆，才能做出快速的决策，进而减少交通事故的发生，有效降低交通拥堵和路径规划不合理等问题的发生频率。

目标检测作为计算机视觉重要组成部分已经广泛应用到视频监控、车联网、机器人等场景，车辆检测作为目标检测中的一部分已经广泛应用在交通流量控制、自动驾驶车辆、智慧停车场、车辆监控等场景^{[5][6][7][8]}，因此研究车辆检测是具有十分重要的现实意义。

车辆检测的目的是要识别出车辆并且对车辆进行细致化分类，也就是需要从图像中获取两部分信息：车辆在图像中的位置；车辆的具体类型，例如轿车，SUV，卡车等等。目前主流的车辆检测方法都将检测的结果发送到云端服务器进行计算，根据计算后的结果做出决策，在将决策指令发送到边缘端设备上执行，但是在自动驾驶等场景中，这种方法存在的问题，当云端和边缘端进行频繁通信的时候势必会产生带宽的拥挤以及数据传输的时间，然后很多时候面对复查的交通情况需要快速做出决策，因此需要一个更加轻量级的车辆检测网络可以部署在计算能力更弱的边缘端设备上，它具有更简单的网络结构、消耗更少的计算资源、花费更短的训练

时间以及可以快速更新。这会避免云端与边缘端数据传输的带宽拥堵，提升车辆检测算法的效率，减少交通事故的发生。给自动驾驶车辆做决策提供足够的反应时间，从而提升自动驾驶汽车的安全性。

1.2 国内外研究现状

在本文中多种道路场景下的轻量化道路车辆细粒度检测技术的主要过程是首先使用在输入图像中找到包含车辆的位置，其次使用宽度学习系统对包含车辆的图像进行细致化分类。具体来说在寻找包含车辆位置图像过程中使用区域建议网络在原始输入图像中绘制若干个边界框，这些边界框被称为感兴趣区域，其次在使用一个分类网络对绘制出的边界框图像进行分类，只区分是背景还是包含车辆的前景，使用边框回归网络计算每一个对象的坐标信息；在车辆分类网络中，将上一步骤的图像输入宽度学习系统中，对图像进行特征提取，最后得到具体的车辆类别。到目前为止宽度学习系统和车辆检测技术在国内外均取得了大量的研究成果，下文将具体介绍二者的发展情况。

1.2.1 车辆检测算法研究

目前车辆检测仍然是各位学者所研究的热点问题。基于深度学习的通用目标价检测方法都可以用于车辆检测^{[9][10][11]}。它们可以被分为两类：二阶段检测算法和一阶段检测算法，两阶段检测算法进行车辆检测主要分为两个过程：生成包含车辆的感兴趣区域和从中找到目标车辆，其代表算法包括：R-CNN^[12]、Fast R-CNN^[13]、Faster R-CNN^[14]等。一阶段检测算法删除了二阶段检测算法中生成包含车辆感兴趣区域这一部分，代表算法包括 SDD^[15]、YOLO^[16]等。文献[17]提出将基于上下文信息融入 R-CNN 中，改进后的网络模型在小型遮挡车辆检测任务中表现良好。文献[18]提出优化 Faster R-CNN 网络的特征提取部分，优化后的模型在无人机采集的车辆图像上准确率有所上升。文献[19]提出对 SSD 网络的损失函数进行增，模型的准确率和速度均有所上升。文献[20]提出使用 YOLO 对不同天气下的车辆进行检测。文献[21]提出使用 Fster R-CNN 对大量的夜间数据集进行训练，提升了夜间检测的准确率。文献[22][23]提出使用 CycleGAN 扩展数据集，生成不同光照的车辆图像，使用 YOLOV4 对扩充后的数据集进行训练提升了检测精度。文献[24]提出将 Faster R-CNN 和 YOLOV5 模型混合训练用于车辆检测和车辆密度估计。

文献[25]提出苏静等人提出为了解决公路上车辆的漏检误检等情况，改进 YOLOV4 算法使用 CSPDarknet53-tiny 作为主干网络，ResBolck 中的残差边与

GhostModule 模块结合作为新网络的特征提取模块,用 ASPPNet 替换 YOLOV4 的 SPPNet 模块结构,扩大感受野,与原算法相比,改进后的方法 AP 提高了 2.69%。文献[26]提出使用改进的 YOLOV5s 对道路车辆中小目标车辆进行检测,具体来说,在特征提取部分增加对小目标车辆的特征提取能力,实验结果表明相比于 YOLOV5 原始网络,作者改进后的网络在小目标车辆检测中 mAP、准确率、召回率均有所提高。文献[27]提出使用改进 Faster R-CNN 检测城市道路中的货车,实验结果表明改进后的方法优于原始 Faster R-CNN。文献[28]提出使用迁移学习和 SSD 对道路车辆进行检测,实验结果表明该方法相比于原始 SSD 算法准确率有所提升,减少了模型内存占有量。

1.2.2 宽度学习系统研究

宽度学习系统^[29]因为其结构简单,收敛速度较快等特性已经被广泛应用在多个场景中^{[30][31][32]}。文献[33]提出使用 BLS 对老化的工业设备进行故障检测,为了应对工业生产过程的多样性,引入亲和力传播(Affinity Propagation, AP)算法来区分不同的生产过程。文献[34]提出使用模糊宽度学习系统对工业机械设备进行故障检测,实验在轴承故障数据集上取得良好的实验效果。在一些工业场景中数据标签很难获得,文献[35]提出使用半监督的宽度学习系统用于这类场景中的故障检测,经实验结果表明该方法在数据标签有限的情况下对于故障检测是有效的。文献[36]提出使用麻雀算法优化宽度学习系统对出现故障的轴承进行检测,由于深度学习存在大量的训练参数,对故障轴承检测时间较长,效率不高,使用优化后的宽度学习系统进行检测可以提升检测精度和速度。文献[37]提出使用迁移宽度学习系统对污水处理过程中存在的故障进行检测。

文献[38]提出使用复杂宽度学习系统来对驾驶员进行疲劳检测。实验结果表明该方法对于驾驶员疲劳检测是有效的。文献[39]提出使用宽度学习系统对车辆数据信息进行析,实现车辆不当行为检测。实验结果表明基于宽度学习系统的准确率和基于深度学习方法相近,但是训练时间提升了数十倍。文献[40]提出改进的宽度学习系统用于提升车联网中数据共享的效率,通过仿真实验结果表明该方法共享数据的速度更快。文献[41]提出使用粒子群优化与宽度学习系统对地铁的客流量进行预测,实验结果表明与传统的人流量预测方法相比,作者所提出的新方法在多项性能指标中均取得优异表现。

文献[42]提出将宽度学习系统和相空间重构相结合在上海风电厂对风速进行预测,经实验结果表明该方法有较快的收敛速度。文献[43]使用改进的宽度学习系统检测热

带气旋的形成。经过实验结果表明该方法模型更新的速度明显优于大多数的深度网络。

1.3 本论文的结构安排

本文主要研究真实道路车辆的检测与细致化分类。自行采集行车记录仪视频作为原始数据，再将视频数据进行预处理保证可以被后续网络直接使用，在本文中，使用宽度学习系统作为多种道路场景下的轻量化道路车辆细粒度检测技术的分类器，在保证检测精度的前提下，轻量化车辆检测模型，提升检测的速度。论文章节安排如下：

第 1 章：绪论。在本章中主要介绍本文研究的必要性和重要性，以及在实际应用中的价值。并且介绍了车辆检测方法以及宽度学习系统两方面的国内外研究现状，最后介绍本文的章节安排情况。

第 2 章：常用车辆检测方法。在本章中首先介绍了卷积神经网络，包括卷积神经网络的组成：卷积层、激活层、池化层和全连接层。介绍每层具体的工作原理和作用。接着介绍现阶段国内外常用的几种车辆检测的方法，主要分为两大类：二阶段检测方法和一阶段检测方法。对不同方法进行分析，讨论优缺点。

第 3 章：真实道路车辆数据集的获取与介绍。在本章中主要介绍本文实验所采用的私有数据集。该数据集是自行采集获取，主要是收集不同车辆的行车记录仪视频。介绍该数据集的基本情况，包括了不同时间节点和不同道路情况的车辆数据，保证数据的完整性。接着介绍数据的预处理方法，由于视频数据无法被后续网络直接处理，需要进行一系列的预处理操作将视频处理成后续网络可以直接使用的形式。最后介绍数据标注的情况，包括数据标注的工具以及标注车辆的具体类别。

第 4 章：基于宽度学习系统的真实道路车辆细致化分类。在本章中主要介绍车辆细致化分类的方法。首先介绍使用的基本方法：宽度学习系统，包括基本模型结构、增加增强节点的增量学习模型结构、增加特征节点的增量学习模型结构、增加输入数据的增量学习的模型结构。接着对车辆细致化分类的具体流程进行介绍，最后介绍实验的进行情况，包括实验环境、实验所使用的设备、实验指标等等，对实验结果进行分析，从实验结果证明提出方法的有效性。

第 5 章：多种道路场景下的轻量化道路车辆细粒度检测技术。在本章中主要介绍多种道路场景下的轻量化道路车辆细粒度检测技术的具体情况。首先介绍使用的基本方法。接着介绍车辆细粒度检测技术的具体执行流程。最后对实验结果进行分析。

第 6 章：总结。在本章中总结现阶段本文已经完成的工作以及未来还需要进一步解决的问题。

1.5 本章小结

在本章中首先对车辆检测相关背景进行了描述，然后根据现实应用场景阐述当前方法中存在问题本文首先对目标重识别相关背景进行了概述，接着阐述了当前车辆检测在应用中存在的困难以及本文的研究意义。接着阐述了国内外关于车辆检测和宽度学习系统的研究进展，最后对全文结构和内容进行总结。

第 2 章 车辆检测相关技术

在本章中主要介绍车辆检测的相关技术，现阶段主流的车辆检测计算大部分都是基于卷积神经网络的，因此在本章的第一部分介绍卷积神经网络的相关内容，第二部分介绍常用的车辆检测算法。

2.1 卷积神经网络

近年来卷积神经网络（Convolutional Neural Networks, CNN）^{[44][45]}经常被用于目标检测、图像识别、语音识别等多种应用场合^{[46][47][48][49]}，卷积神经网络是一种具有深层次网络结构的前馈神经网络（Feedforward Neural Networks, FNN）^{[50][51]}，它也是众多深度学习算法中极具代表性的一种。卷积神经网络的模型如图 2.1 所示。

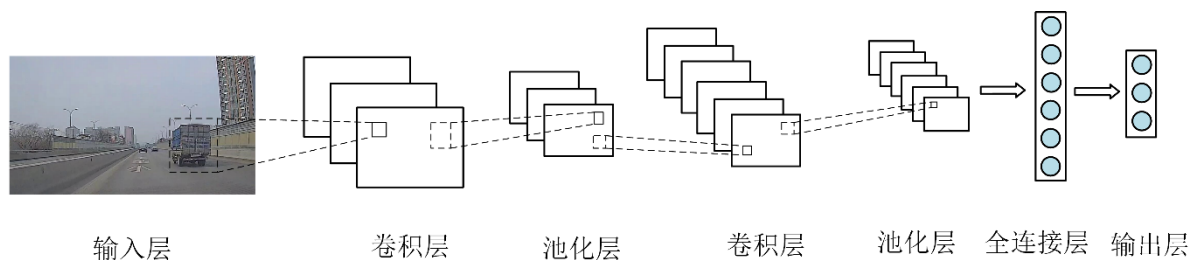


图 2.1 卷积神经网络模型图

Fig. 2.1 The convolutional neural network model

通常来说卷积神经网络主要由四部分组成：卷积层、激活层、池化层和全连接层。卷积层是卷积神经网络的核心所在，主要进行卷积操作。激活层的作用则是通过一些激活函数对上一步骤的输出进行处理，提升网络的表示能力。池化层又叫下采样层，主要用于降低计算量，提升泛化能力。全连接层在整个结构的最后，用于将前面提取到的特征进行组合并输出。下面将详细介绍上述四部分的具体内容。

2.1.1 卷积层

在卷积神经网络中最重要的部分就是卷积层，卷积层的主要作用就是通过卷积操作对输入数据今天特征提取。卷积层的内部由若干个卷积核组成，在卷积核中的每一个元素都对应一个权重系数和一个偏执量。卷积层的输出被映射到输入数据上的范围被称作感受野，感受野的大小通常由卷积核的大小觉得。对于感受野来说我可以类比视觉皮层对事物的感知能力和范围。在卷积层中通常会对输入数据进行卷积计算。例如输入图像的大小为 $4 \times 4 \times 1$ ，卷积核大小为 2×2 ，步长为 1，沿着输入矩阵从左至右、从上至下的顺下进行滑动，滑动窗口的大小设置为 2×2 ，将卷积核和

输入图像相乘在累加得到的结果记作以此卷积计算。计算方式如图 2.2 所示：

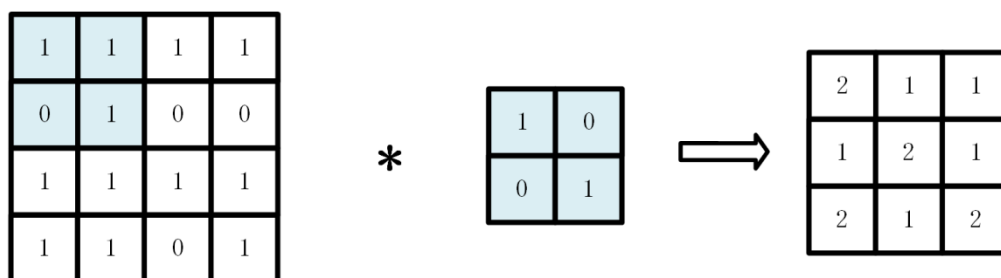


图 2.2 卷积层计算过程

Fig. 2.2 Convolutional layer calculation process

2.1.2 激活层

卷积神经网络的激活层主要就是使用各种激活函数对上一步的输出进行处理。在面对不同的输入数据时可以选择不同的线性函数来学习数据维度间的特征，经常被使用的线性函数包括但不限于 Sigmoid、Tanh、Relu 等函数。本节将详细介绍这三种线性函数的特性。

Sigmoid 函数的数学表达式如下所示：

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

Sigmoid 函数的值域在 0 到 1 之间，它的函数图如图 2.3 所示。

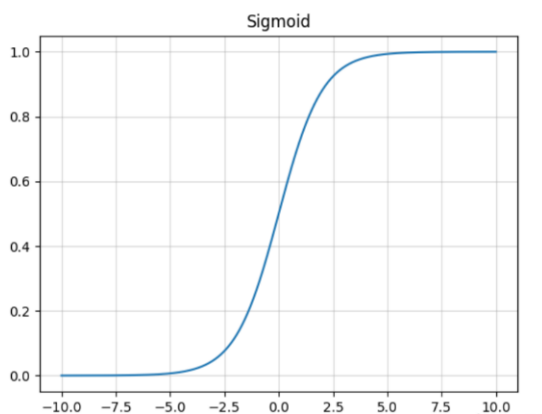


图 2.3 Sigmoid 函数图

Fig. 2.3 Sigmoid function diagram

在网络模型中通常会接收到大量的输入数据，并且从这些数据中找到并且学习数据维度之间的特征，但是在实际情况下，输入的数据会存在较大的差异，这就导致输入数据的特征也会存在巨大的差异，面对这种情况 Sigmoid 函数的优点在于将输出值的范围约束在 0~1 之间，将大范围内的输入特征压缩到 0~1 范围内，这样即可保证数据幅度不会出现较大的波动，由于预测概率通常在 0~1 之间，该函数经常被用于将预测概率作为输出的模型。但是 Sigmoid 函数的缺点也十分明显，当输入

较大或者较小时，Sigmoid 函数的输出基本是一个常数，也就导致变化不太明显，进而导致梯度接近于 0，并且 Sigmoid 函数进行幂运算相对耗时。

Tanh 函数的数学表达式如下所示：

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

Tanh 函数的函数图如图 2.4 所示：

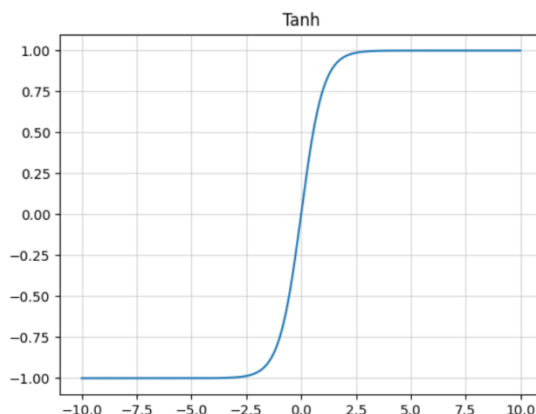


图 2.4 Tanh 函数图

Fig. 2.4 Tanh function diagram

Tanh 函数的输出范围在-1~1 之间，与 Sigmoid 函数相同是一个非线性函数，可以对非线性模式的数据进行学习，与 Sigmoid 函数不同的是，Tanh 函数的值域在-1~1 之间，函数的中心点在坐标原点，这样更有利于适应不同的数据。Tanh 函数的导数的范围在 0~1 之间，相比于 Sigmoid 函数范围更广，在函数图上表示 Tanh 函数比 Sigmoid 函数更加陡峭，一定程度上解决了梯度消失的问题。Tanh 函数更适用于多任务分类的任务。但是 Tanh 函数仍然存在梯度消失的问题。

Relu 函数的数学表达式如下所示：

$$\text{ReLU}(x) = \max(0, x) \quad (2.3)$$

Relu 函数的函数图如图 2.5 所示：

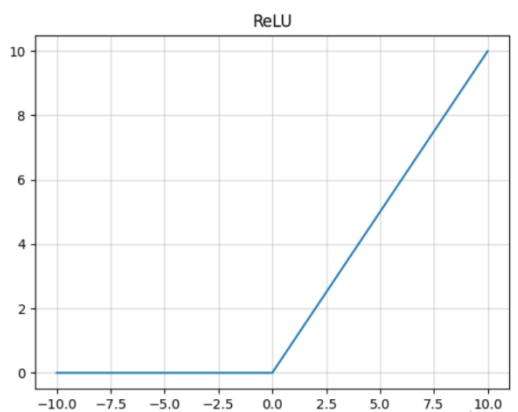


图 2.5 ReLU函数图

Fig. 2.5 ReLU function diagram

Relu 函数的优点在于计算速度很快，它的计算过程主要分为两步：比较输入数据，如果输入数据大于 0，则输出原值，如果输入数据小于 0，则输出 0。由于 Relu 函数的计算量极小可以提升模型的计算速度。其次 Relu 函数可以避免上述两种函数存在梯度消失的情况。但是让输入为正值时，导数为 1 虽然不会出现梯度消失的问题，但是可能会带来梯度爆炸。

2.1.3 池化层

池化层在卷积神经网络中主要作用就是对提取的特征进行选择，池化层会压缩输入的特征图，过滤掉权重较低的特征，提取主要特征，减小特征图的大小。同时池化层可以有效的减小网络的参数防止过拟合的现象出现。池化层具有以下特性：

输入图像在经过池化层后，特征不会发生变化。在实际的应用中，输入的图像相对比较复杂，所包含的对象多且复杂，如果想要得到精确地结果就要将特征和输入图像对应的位置对应起来。

在池化层中会对输入的特征进行降维，只保留主要特征。

在池化层中通过加深网络的深度避免出现过拟合的情况。

池化层中包含两种池化方式：最大池化和平均池化。最大池化是大多数网络所采用的池化方式。这种方式的优点是纹理特征得到更好的保存。在通常情况下最大池化大小是 2×2 。例如对于一个 4×4 的特征图中，步长设置为 2，滑动窗口设置为 2×2 ，在特征图的每一个 2×2 大小的矩阵内选择一个最大的值作为池化值。最大池化计算过程如图 2.6 所示：

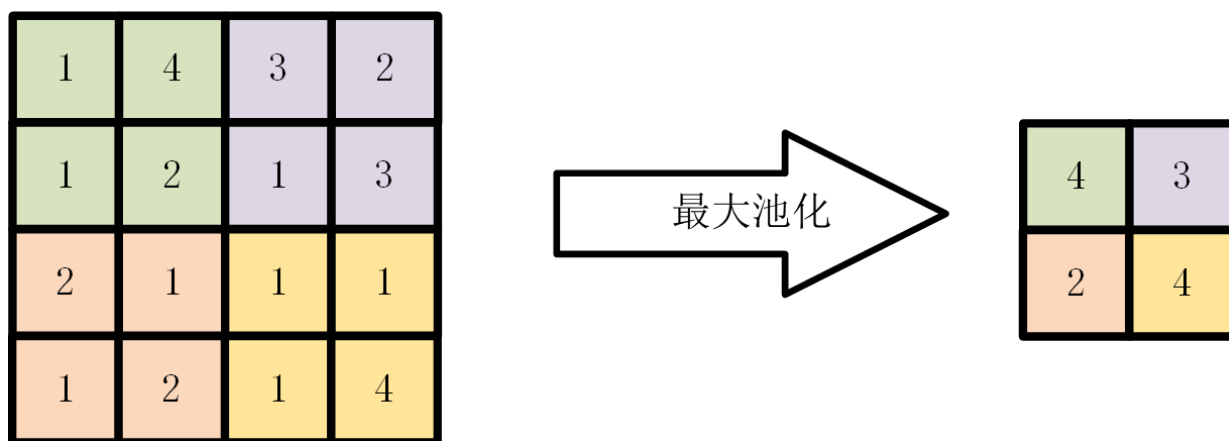


图 2.6 最大池化过程

Fig. 2.6 Max pooling calculation process

平均池化和最大池化的区别是在特征图的矩阵中不在选择最大值而是平均值，平均池化的计算过程如图 2.7 所示：

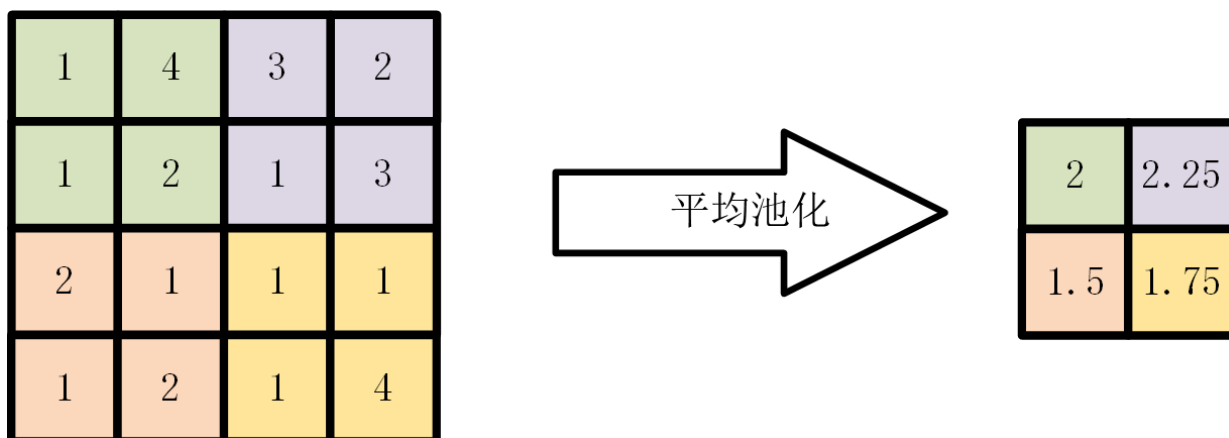


图 2.7 平均池化过程

Fig. 2.7 Average pooling calculation process

2.1.4 全连接层

全连接层的位置是在整个模型的最后部分，作为卷积神经网络的一个分类器，它的主要作用就是对前几步骤中提取到的特征进行组合并且输出，全连接层本身并不具备特征提取的能力，而是利用现有的特征进行组合，将结果送入输出层。

2.2 常用车辆检测算法

车辆检测一般指的是在车辆数据图像中找到车辆的位置并且识别出它的类别。传统的目标检测算法均可以用于车辆检测中。这类方法大致可以分为两种：二阶段检测算法和一阶段检测算法。其中二阶段检测算法要先找到候选区域，其次在对候选区域进行分类。具体来说首先在图像中画出若干个候选框，然后判断在这些候选框中是否存在需要检测的对象，若有则对该对象进行分类和边框回归。二阶段检测算法的代表包括 R-CNN、Fast R-CNN、Faster R-CNN 等。然后一阶段检测算法则只需要一步即可完成检测。也就是直接在图像上获取待检测目标的类别和坐标信息。一阶段检测算法的代表包括 YOLO、SSD 等。

2.2.1 二阶段车辆检测算法

(1) R-CNN:

R-CNN 是第一个使用深度学习方法进行目标检测的算法。传统的目标检测算法通常使用穷举的方法搜索图像中的物体，然后在对其进行分类，R-CNN 仍然按照传统方法的思路，首先从图像中找到所有的候选框，然后进行特征提取，最后进行分类和边框回归。

当 R-CNN 网络接收到输入数据后，会在图像中生成若干个选择框，被检测对象

通常会分布于多个选择框中，但是每个被检测对象所分离出的若干部分的相似性要明显高于其他对象，因此计算相邻的选择框之间的相似性，将相似性较高的选择框合并，这样就可以保证同一对象的不同区域将会合并到一起，重复上述步骤直到无法合并位置。这样就得到了该输入图像的候选区域。

将上一步骤获取到的候选区域送入卷积神经网络进行特征提取，目的是提取到图像中更高级、更抽象的特征信息，为下一步骤的分类和回归提供基础。

该步骤中主要分为两部分：一是将上一步骤提取的特征送入分类网络，对该数据进行分类，二是将上一步骤提取的特征送入边框回归网络，用于获取到物体更准确的边框信息。R-CNN 的网络模型如图 2.8 所示：

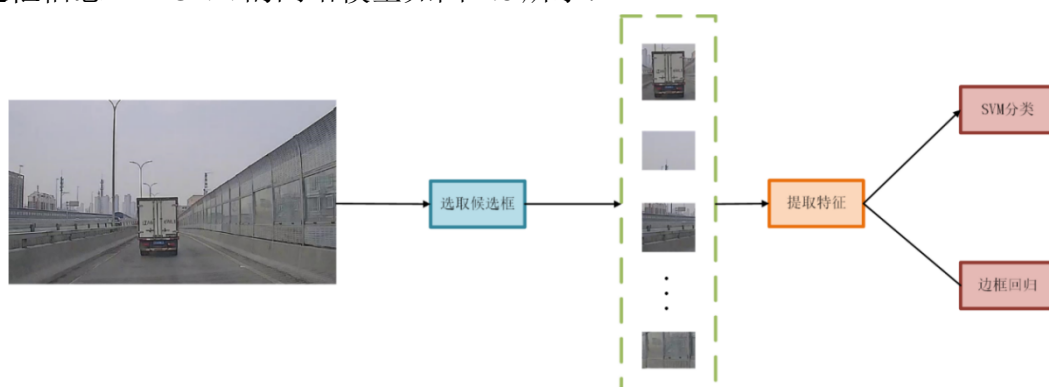


图 2.8 R-CNN 网络模型图

Fig. 2.8 Diagram of the R-CNN network

虽然 R-CNN 是目标检测发展史上的一大重要转折，但是不可否认它仍然存在一些缺点，例如耗时很长、需要很大的存储资源，这主要是由于在提取到候选区域时通常要选择大量的候选框，传统的候选区域提取的方法效率不高等原因造成的。面对上述的问题，Fast R-CNN 随之出现。

(2)Fast R-CNN

在 2015 年提出了 Fast R-CNN 算法，针对 R-CNN 存在的问题主要从两部分进行优化。

在 Fast R-CNN 中作者提出 ROI Pooling 的概念，解决了选择候选区域的时候必须将被选择的图像通过裁剪、缩放等方式统一图像的尺寸。由于卷积神经网络的输入图像必须是统一大小的，目的是为了最后全连接层可以计算选择特征。所以 R-CNN 在选择候选区域时候会将大小不统一的图像通过裁剪、缩放等方式对图像进行尺寸的统一，这个过程会严重扭曲一部分特征信息，导致识别的效率不高。在 Fast R-CNN 中，作者使用 ROI Pooling 来解决这个问题，ROI Pooling 的主要思想就是首先确定要生成图像的长宽，在将特征图按生成图像的长宽进行水平和竖直切分，取每一份中的最大值，组成固定尺寸的特征图。这样可以输出的固定尺寸的图像。

在 Fast R-CNN 中，作者提出将最后一步中的分类损失和回归损失结合在一起训练网络，提出多任务损失函数的思想。Fast R-CNN 的网络结构图如图 2.9 所示：

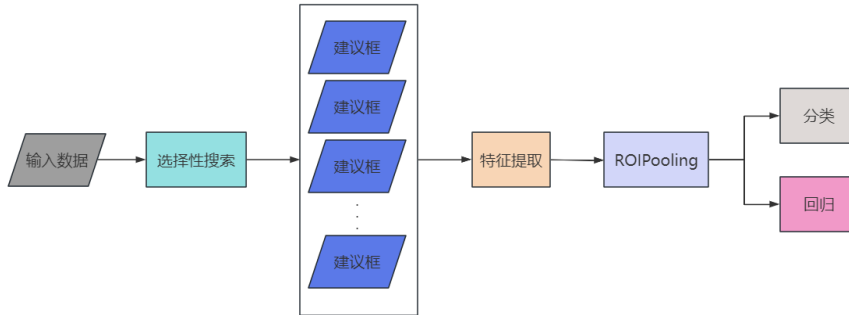


图 2.9 Fast R-CNN 网络模型图

Fig. 2.9 Diagram of the Fast R-CNN network

尽管 Fast R-CNN 在 R-CNN 的基础上已经做出改进，但是由于在选择候选区域的位置仍然使用选择性搜索方法，该方法十分耗时，网络的训练速度仍然需要提升。因此后续 Faster R-CNN 也随之出现。

(3)Faster R-CNN

针对 Fast R-CNN 和 R-CNN 存在问题，Faster R-CNN 提出使用区域建议网络（Region Proposal Networks,RPN）替代选择性搜索方法。Faster R-CNN 算法主要包含四部分：首先将输入图像送入卷积神经网络进行特征提取，与 Fast R-CNN 不同 Faster R-CNN 不需要对每一个候选区域都进行以此特征提取，这样会出现很多重复性的工作，只需要对整张图像进行一次特征提取即可。其次使用 RPN 生成候选区域，并且通过一个分类网络对候选区域进行分类，区分是否是背景，在通过一个回归网络计算候选区域的位置信息。接着进入 ROI Pooling 层，最后在进行分类和边框回归。Faster R-CNN 的网络结构图如图 2.10 所示：

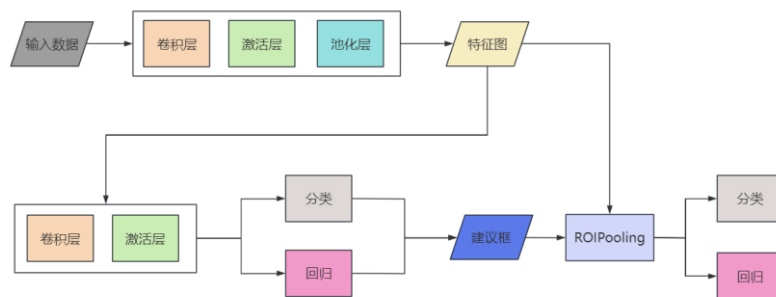


图 2.10 Faster R-CNN 网络模型图

Fig. 2.10 Diagram of the Faster R-CNN network

2.2.2 一阶段车辆检测算法

(1)SSD

SSD 是一个经典的一阶段检测算法，与二阶段检测算法不同的是，SSD 在一个

网络中直接回归出物体的类别和位置，由于减少了网络的数量，因此 SSD 的检测速度有了明显地提高了。SSD 的主干网络采用的是图像分类网络，基线模型采用 VGG16 网络，在 SSD 网络中，经过卷积层、池化层等处理可以得到图像的特征图，下一步直接在特征图上进行回归可以得到被检测对象的类别和位置信息。

考虑在特征图的每个位置上的候选框情况，SSD 对于候选框的位置回归需要给出如下四个参数：检测对象的偏移量 (x,y) ，检测对象的宽度和高度 (w,h) ，但是二阶段检测算法对于一个对象需要针对每一个类都要进行一次位置回归，因此需要计算 $(A+1) \times 4$ 个参数，其中 A 表示检测类别数量。

针对每一个候选框 SSD 都要给出每一个类别加背景的得分，得分最高的记为该候选框的类别。SSD 的模型结构图如图 2.11 所示。

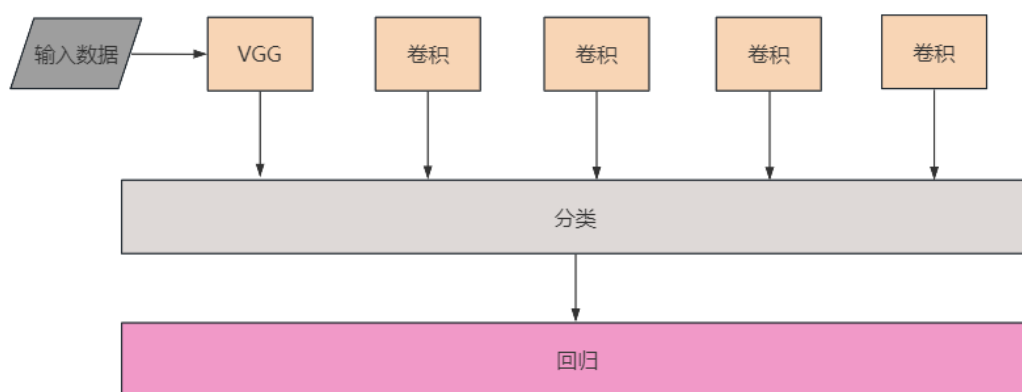


图 2.11 SSD 网络模型图

Fig. 2.11 Diagram of the SSD network

虽然 SSD 在检测速度上相较于二阶段检测算法有了明显的提高，但是仍存在一些问題：对于检测框的初始大小需要认为干预设置，SSD 中的检测框的大小和形状无法通过直接学习获得，而需要人为干预设置，网络中每一层特征所需要的检测框的大小都不一样，过于依赖人工建议的设置。SSD 对于小目标检测仍然不够理想。

(2)YOLO

YOLOV1 是 YOLO 系列目标检测算法的开篇之作，将二阶段检测候选区域和目标识别结合在一起，只需要一步即可完成目标检测，但是 YOLOV1 并没有删除候选区域的选择，而是使用一种新的候选区域选择的方法，将图像分为 $S \times S$ 个网格（为了方便解释，这里 $S=7$ ），也就是将图像划分为 49 个网格，每个网格规定预测出两个候选区域，这样一张图像就被分为 98 个候选区域，但是 YOLOV1 仍然存在一些缺点：每个网格只能预测两个类别，对于密集型目标检测效果不好，目标定位的方法有些粗糙对小目标检测效果不够好。针对 YOLOV1 存在问题，YOLOV2^[52]做出如

下改进：

第一：在 YOLOV2 的卷积层之前加入标准化机制（Batch Normalization），对每个层的数据都进行标准化，这样网络不必学习如何分配数据，进而提升了网络的效率避免过拟合的问题。

第二：YOLOV2 在图像划分的每个网格中设置不同大小的先验框（Anchor Boxes），该框作为预定义的候选框，后续判断该框内是否存在需要检测的对象。并且移除全连接层和一个池化层，让卷积层输出的图像分辨率更高。

第三：在 YOLOV2 中使用 K-means 的方法对标注的候选框进行聚类分析，获取到最终候选框的大小，具体来说是用距离判定相识度，二者距离越近相似度越强，YOLOV2 的目的是希望候选框和真实框相交的面积更大，因此使用距离作为聚类指标更合适。

第四：YOLOv2 将浅层特征和深层特征相融合之后在进行检测，提升了对小目标检测的能力。

2018 年 YOLOV3 对 YOLOV2 网络进行改进，引入残差块，网络变得更深，改进后的网络被命名为 Darknet-53。

除此之外 YOLOV3 从不同尺度进行特征提取，提取最后 3 层的特征图，在每个特征图上分别进行预测，对几个尺度的特征图进行融合，以获取不同的感受野信息，通过三种尺度的特征融合分别获得了小、中、大目标的检测能力，同时在分类网络中，YOLOV3 舍弃了 softmax 分类器，使用逻辑回归。YOLOV4 在 YOLOV3 的基础上在输入端增加数据增强方法，将四张图像拼接在一起进行训练，间接的增加网络训练的速度，同时优化了损失函数，其中 α 是平衡比例的系数， v 是衡量长宽比一致性的参数， d 是计算两个中心点之间的欧氏距离， c 代表闭包的对角线距离， w 和 h 是预测框的宽和高， w^{gt} 和 h^{gt} 是真实框的宽和高。

$$L_{CoU} = 1 - IoU + \frac{d^2}{c^2} + \alpha v \quad (2.4)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (2.5)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (2.6)$$

YOLOV5 在训练的时候会自行学习到输入数据中最佳的检测框，而不再需要依赖 k-means 聚类获取检测框。从整体框架上来说，YOLOV5 相比 YOLOV4 流程更加简单。其次 YOLOV5 为了统一输入图像的大小，对图像进行自适应放缩计算自适应缩放的比例，计算自适应缩放后的尺寸，使用原始图像的长宽和缩放系数相乘得到

待填充图像的长宽，对图像进行填充。YOLOV5 的损失包含三部分，矩形框损失、置信度损失和分类损失。用如下的损失函数定义矩形框损失，该损失函数在考虑到预测框和真实框的堆叠面积基础上增加了一个影响因子将预测框和真实框的长宽比也考虑了进来，其中 d 为真实框和预测框到中心点距离， l 为真实框和预测框所包围矩形最小的对角线长度， h 为真实框和预测框的宽高比相似度， α 为 h 的影响因子， S_1 和 S_2 为两个矩形框的面积；

矩形框损失函数为：

$$L_c = 1 - IOU - \frac{d^2}{l^2} - \alpha h \quad (2.7)$$

$$IOU = \frac{S_1}{S_2} \quad (2.8)$$

$$h = \frac{4}{\pi^2} (\arctan \frac{w_l}{h_l} - \arctan \frac{w_p}{h_p})^2 \quad (2.9)$$

$$\alpha = \frac{h}{1 - IOU + h} \quad (2.10)$$

用如下损失函数定义置信度损失， C 为置信度标签矩阵， V 为预测置信度矩阵其中 x 、 y 、 z 代表置信度矩阵的维度。置信度损失函数为：

$$L_d(z, x, y) = -C(z, x, y) * \log V(z, x, y) - (1 - C(z, x, y)) * \log(1 - V(z, x, y)) \quad (2.11)$$

用如下损失函数定义分类损失， C 为置信度标签矩阵， V 为预测置信度矩， G 代表类别数， τ 为平滑系数，取值范围 0~1。分类损失函数为：

$$L_f = -C_m * \log V - (1 - C_m) * \log(1 - V) \quad (2.12)$$

$$L_m = L(1 - \tau) + \tau/G \quad (2.13)$$

2.3 本章小结

在本章中向大家介绍了一些基于深度学习的常用目标检测算法，根据其检测流程分为一阶段检测算法和二阶段检测算法，其中一阶段检测算法主要包括：R-CNN、Fast R-CNN 和 Faster R-CNN；二阶段检测算法主要包括：YOLO 和 SSD。这些算法在不同方面上取得了显著的成果，为实际应用提供了强有力的支撑，但是随着检测场景的愈发复杂，上述算法仍然面临一系列挑战，例如快速响应、多种场景下的检测问题都是未来研究的重要方向。

第 3 章 真实道路车辆数据集的获取与介绍

本章主要介绍车辆检测和分类所使用的真实道路车辆数据集。该数据集是从若干不同车辆的行车记录仪上收集得到的。其中包括高速公路、快速路、高架桥、省道、国道等不同道路类型和清晨、正午、夜晚等不同时间段下的真实道路车辆数据。从行车记录仪中获取的视频数据无法直接被系统处理，因此对获取到的视频数据做了一些预处理工作，具体内容在下文详细介绍。

3.1 真实道路车辆数据集

在基于有监督深度学习的目标检测与识别算法中，数据集的选取十分重要。在车辆检测任务中，数据集需要具备如下条件：

数据量充足：大部分的车辆检测任务都是基于深度学习的，这类方法通常都需要大量的数据才可以训练网络，如果数据量不充足网络训练的效果不好。

数据场景丰富：在真实场景下，需要在各种路况，各种时间段下都进行车辆检测，这就要求用于训练的数据样本所处的场景尽量丰富，这样模型训练可以在不同场景下都得到充分的训练，更好的提取特征，避免模型存在泛化能力差过拟合等问题。

因此本文选取自行采集的真实道路车辆数据集。通过近十台不同车辆的行车记录仪获取初始视频数据，结果如图 3.1 所示：

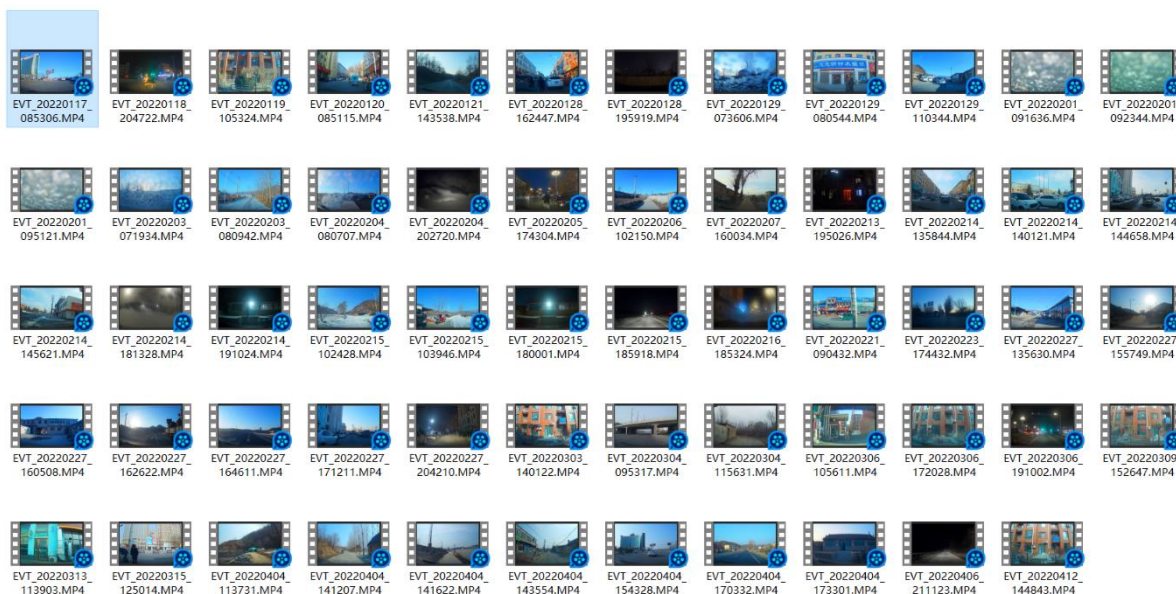


图 3.1 行车记录仪视频数据

Fig. 3.1 Drive Recorder Drive Recorder

为了丰富数据集的场景，选择了不同道路情况下的车辆数据，包括高速公路、高架桥、快速路、国道、省道、县道等。同时选择了不同时间段下的车辆数据，包括清晨、正午、傍晚、夜晚等。由此可见采集的真实道路车辆数据集具备丰富的道路场景以及时间场景，可以用于进行车辆检测。选取部分数据图像如图 3.2、3.3 所示：

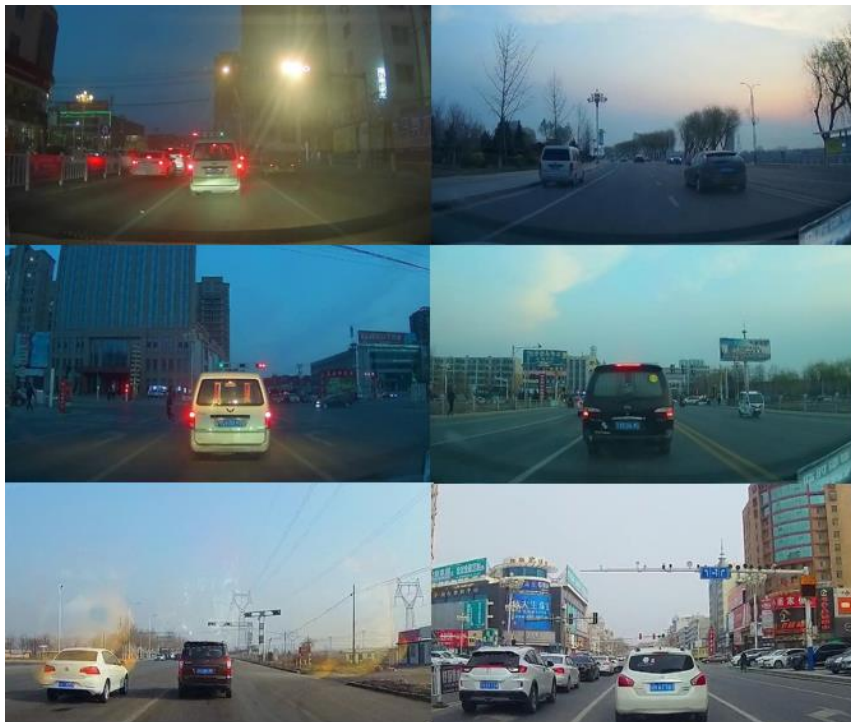


图 3.2 不同天气条件下的数据
Fig. 3.2 Datasets of different weather conditions



图 3.3 不同道路类型下的数据
Fig. 3.3 Datasets of different road types

3.2 数据处理方式

3.2.1 数据预处理

(1) 视频提取关键帧

将视频数据通过提取关键帧的方法将图像提取出来。具体来说使用 Python 中的 OpenCV 库进行处理，OpenCV 是一个具有跨平台性 Python 语言的第三方库，它可以在 Linux、Windows、Mac 等多个系统上执行。OpenCV 是 Python 中最重要的开源机器视觉第三方库，库内包含众多常用且功能丰富的函数，使用 C/C++ 语言编写可以快速处理一些图像相关的任务，使得图像处理与分析变得更加简单。对于将视频切割成图像，使用 OpenCV 的函数方法读取视频，在使用循环语句按照固定的帧率保存图像，这里由于行车记录仪的视频很多都是 25 帧，所以这里也按照 25 帧切割一张图像，也就是时间间隔 1 秒切割一张图像。切割后的结果如图 4 所示：

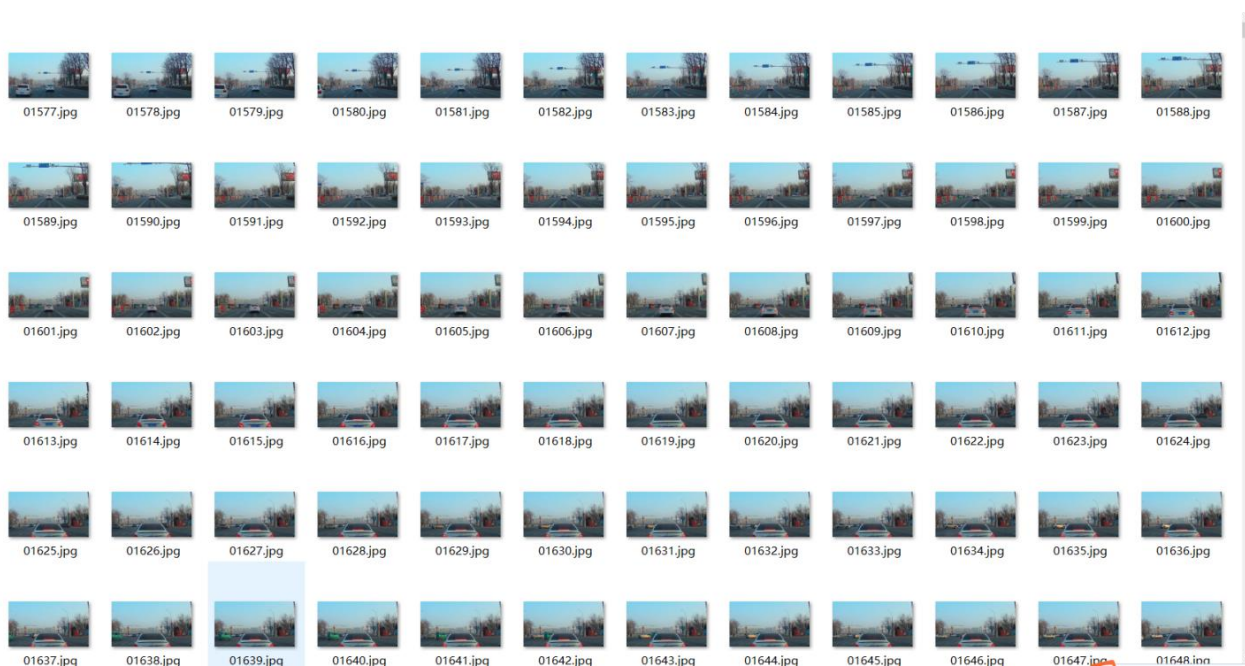


图 3.4 在视频中按帧提取图像

Fig. 3.4 Images are extracted image by frame in the video

(2) 统一图像大小

从理论上来说任意大小的输入图像都可以被网络处理，但是这里的理论上指的是计算机有足够大的算力处理任意大小的图像，但是从实际情况出发如果输入图像过大对于算力的负载过大，因此从不同行车记录仪中切割的图像具有不同的尺寸，包括 1920×1080 和 1280×720 。通过程序将二者统一成 960×540 分辨率上两种尺寸图像被重构成统一大小。由于在实验中原始数据两种分辨率宽高比例相同，直接重构大小即可，考虑到后续系统可能会接受其他不同长宽比例的输入数据，直接重构大小会损坏原始图像的比例，扭曲特征导致检测效率降低，所以这里对于图像大小的重构方式采取按比例缩放，保证输入图像的原始比例，在图像尺寸不够的地方使用黑色像素点进行填充，这样既可以将图像重构成统一大小也不会改变原始图像的宽高比例，保证特征提取的完整和正确性。

(3)数据增强

数据增强是人为的不改变数据的实质性下对数据集进行扩充。通过提升数据量使系统可以学习到更多具有鲁棒性的特征，进而提高系统的检测能力和泛化能力。在本文中采用几何变换、增加亮度、增加噪点等方式对数据进增强。

3.2.2 数据标注

数据标注指的是对模型需要进行训练的数据中各个元素进行标记，辅助模型学习该数据中的特征，然后在训练过程中应用这些标记好的数据。在有监督学习中，数据标注是模型取得好的训练效果的前提，机器学习模型是否可以精准预测出结果往往取决于它训练所以的数据情况，如果数据标注不够标准，数据质量不高，模型很难准确的提取出数据之间的特征，学习错误的模式。因此在数据标注时要尽可能保证精准。

使用 labeling 工具进行标注，主要标注类型包括四类，具体类别信息如表 3.1 所示：

表 3.1 数据标注信息
Tab. 3.1 Data annotation information

标签	解释	数量
Sedan	轿车	7261
SUV	越野车	5075
Microbus	多人商务车	4299
Truck	卡车	4179

数据标注过程如图 3.5 所示：

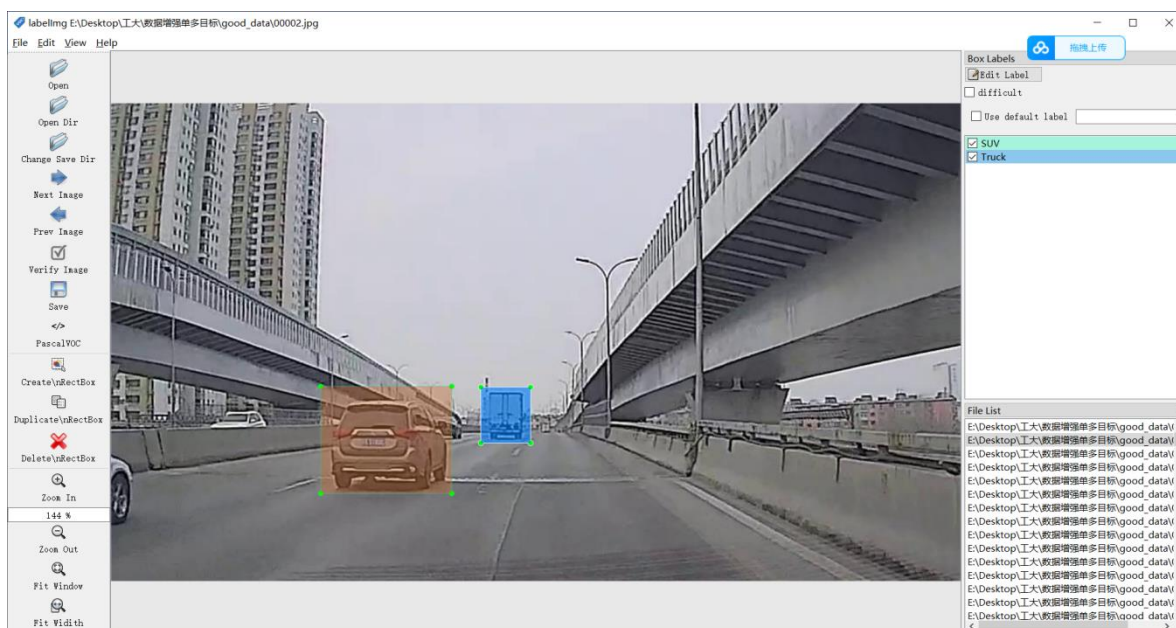


图 3.5 使用 labeling 进行数据标注
Fig. 3.5 Data annotation by labeling

标注后的数据会对应生成一个 xml 文件，在文件中会标注出数据存储的路径、名字、标签信息、坐标信息等，标注后生成的结果图如图 3.6 所示：

```
1 <annotation>
2   <folder>good_data</folder>
3   <filename>00006.jpg</filename>
4   <path>E:\Desktop\good_data\00006.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>960</width>
10    <height>540</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>Truck</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>621</xmin>
21      <ymin>259</ymin>
22      <xmax>796</xmax>
23      <ymax>438</ymax>
24    </bndbox>
25  </object>
26 </annotation>
27
```

图 3.6 数据标注结果

Fig. 3.6 Data annotation results

3.3 本章小结

在本章中深入探讨了数据集的获取和处理，这是十分重要的一环，数据的质量直接影响车辆细粒度检测技术的性能和泛化能力。详细介绍了数据获取与处理的过程，通过采集行车记录仪的视频作为初始数据集，接着对视频数据进行提取关键帧、重构大小、数据增强等工作，获取到最终的数据集。使用 `labelimg` 工具对数据进行标注。

第 4 章 快速更新的轻量化车辆分类方法

车辆检测的目的是要识别出车辆并且对车辆进行分类，也就是需要从图像中获取两部分信息:车辆在图像中的位置；车辆的具体类型，例如轿车，SUV，而车辆分类是车辆检测的基础工作。因此在本节中将介绍在多种道路场景下的轻量化道路车辆细粒度检测技术中的快速更新的轻量化车辆分类方法，首先介绍宽度学习系统的基础模型，其次介绍宽度学习系统的增量模型，最后介绍快速更新的轻量化车辆分类方法以及实验结果分析。

4.1 宽度学习系统模型结构

深度学习网络已经被证明使用更深层次的网络结构可以在很多工作上有着很好的表现，但是深层次的网络结构具有模型复杂，参数众多等问题，导致网络难以训练，要付出很大的计算资源和时间资源。在自动辅助驾驶领域中希望能够在保证模型效果的前提下尽可能的提高训练速度。

宽度学习系统是一种不依赖于模型深度的具有扁平式网络结构的神经网络结构，宽度学习系统可以通过增加网络的宽度的方式来优化网络。宽度学习系统将接收到的输入数据送入特征节点用于特征映射，在将结果送入增强节点计算网络最后的输出结果。此外宽度学习系统可以通过使用增量学习的方式增加增强节点的数量，扩展网络结构，通过计算伪逆的方式实现网络的快速迭代而非重新训练，因此宽度学习系统经常被用于需要快速反应的分类和回归任务中。

4.1.1 求解伪逆

在扁平式网络结构中求解伪逆的方式是一种计算输出层权重常用且有效的方法。伪逆又可以成为广义的逆矩阵，计算方法有很多，例如正交投影法、正交化法、迭代法等等，但是这些直接计算伪逆的方法往往十分耗时，特别是应对维度较大的图像数据输入的时候。

在宽度学习系统中提出另一中伪逆的计算方式：

$$\arg \min W: \| AW - Y \|_v^{\sigma_1} + \lambda \| W \|_u^{\sigma_2} \quad (4.1)$$

其中 $\sigma_1 > 0$ ， $\sigma_2 > 0$ ， u 、 v 代表范数正则化系数， λ 代表权重 W 的系数，当 $\lambda = 0$ 时，逆问题就可以看成是最小二乘问题，这时即可求得原始的伪逆；当 $\lambda = \infty$ 时解具有强约束性且趋于 0，因此矩阵的伪逆可以表示为：

$$P^+ = \lim_{\alpha \rightarrow 0} (\alpha I + P^T P)^{-1} P^T \quad (4.2)$$

4.1.2 基础模型

在本节中，将详细介绍宽度学习系统的基础模型，其模型结构图如下图 4.1 所示：

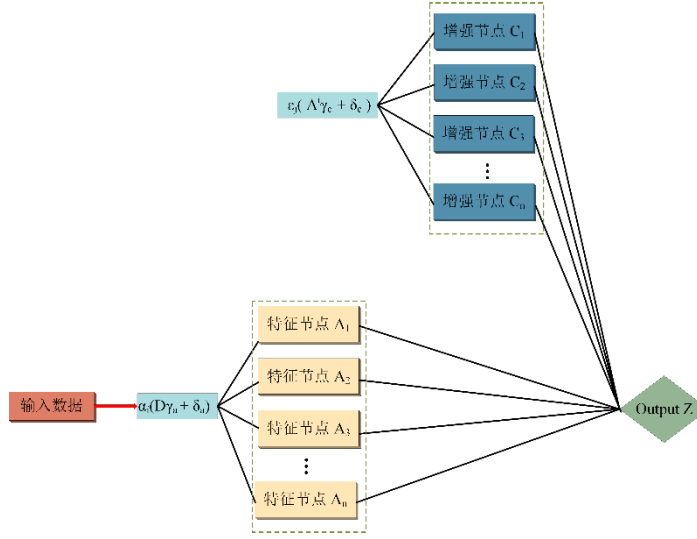


图 4.1 宽度学习系统基础模型

Fig. 4.1 Basic Models for Broad Learning systems

将 A 作为网络的输入数据， Z 作为网络的最终输入，其中 A 是一个 $n \times m$ 的矩阵， n 代表输入数据的样本数量， m 代表每个样本数量的维数， Z 是一个 $n \times s$ 的矩阵， s 代表样本的类别数量。将 A 在网络中进行特征映射，可以生成 t 个特征映射，每一个特征映射都存在 k 个节点。可以表示为：

$$M_a = \omega_a(AH_{r_a} + \mu_{r_a}), a=1,2,3,\dots,t, \quad (4.3)$$

其中 H_{r_a} 和 μ_{r_a} 均为随机生成的矩阵，表示特征节点间相互连接的权重和偏执。 ω_a 是线性函数，这里不唯一，可以根据不同的输入数据或者不同的特征映射组合选择不同的线性函数。

得到 t 个不同的 M 组合起来得到 s 组特征节点， $M^t=[M_1, M_2, M_3, \dots, M_t]$ ，它代表通过网络映射从输入 A 中提取到的全部映射特征。

接着根据 M^t ，随机生成的 H_{u_b} 和 μ_{u_b} ，非线性函数 φ_b 计算对应的增强节点，可以表示为：

$$N_b = \varphi_b(M^t H_{u_b} + \mu_{u_b}), b = 1,2,3,\dots,r \quad (4.4)$$

H_{u_b} 和 μ_{u_b} 依然是随机生成的矩阵，表示增强节点间相互连接的权重和偏执。 φ_b 代表非线性函数，这里同样也可以选择不同的 φ ，通过上式得到的 N_b 也是矩阵形式，将不同的 r 个 N 组合在一起可以得到最终的 r 组增强节点 $N^r=[N_1, N_2, N_3, \dots, N_r]$ ，它代表从映射特征中得到的所有增强节点

将映射特征 M^t 和增强节点 N^r 合并在一起成为一个矩阵，将合并后矩阵和网络连

接的权值相乘就可以得出最后的输出。可以表示为：

$$Z = [M^t | N^r] G = PH \quad (4.5)$$

其中 P 为 M^t 和 N^r 合并后的矩阵 $P = [M^t | N^r]$ ， H 作为连接网络的权重，它可以表示为：

$$H = \lim_{\alpha \rightarrow 0} (\alpha I + P^T P)^{-1} P^T Z \quad (4.6)$$

其中需要让 $\alpha \rightarrow 0$ ， I 为单位矩阵， P 的伪逆乘以输出矩阵 Z 得到网络的连接权重 H 。

将计算得到的 H 代入 (4.5) 中即可得到网络最终的输出 Z 。

4.1.3 增强节点的增量学习

宽度学习系统具有易扩展的特性，在大多数的情况下宽度学习系统的基础模型难以达到期望的分类准确率，这时可以通过增加增强节点数量的方式扩展模型，提高模型的分类能力，以获取更高的分类精度。下面将详细介绍增加增强节点的增量学习算法流程。其模型结构图如下图 4.2 所示：

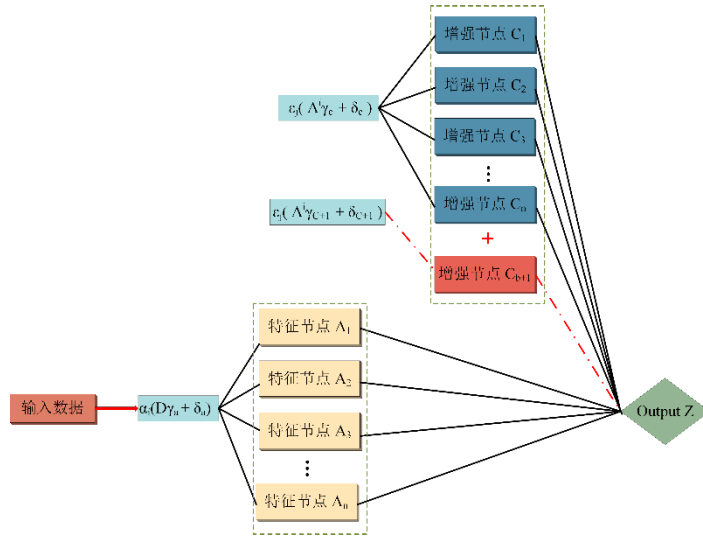


图 4.2 宽度学习系统增强节点增量学习模型

Fig. 4.2 Incremental learning model of enhancement nodes for broad learning systems

首先获取到基础模型的输入矩阵 $P = [M^t | N^r]$ ，对基础模型的增强节点进行增加，增加后的输入矩阵可以表示为：

$$P^b = [P | \varphi(M^t H_{u_{b+1}} + \mu_{u_{b+1}})] \quad (4.7)$$

$H_{u_{b+1}}$ 和 $\mu_{u_{b+1}}$ 均为随机生成的矩阵，表示新增加的增强节点间相互连接的权重和偏执。通过 4.1.1 中的讨论可以推导出新矩阵 P^a 的伪逆，具体如下表示：

$$(P^b)^+ = \begin{bmatrix} (P)^+ - QV^T \\ V^T \end{bmatrix} \quad (4.8)$$

$$Q = (P)^+ \varphi(M^t H_{u_{b+1}} + \mu_{u_{b+1}}) \quad (4.9)$$

$$F = \varphi(M^t H_{u_{b+1}} + \mu_{u_{b+1}}) - PQ \quad (4.10)$$

$$V^T = \begin{cases} (F)^+ & F \neq 0 \\ (1 + Q^T Q)^+ V^T (P)^+ & F = 0 \end{cases} \quad (4.11)$$

最终新的的网络模型权重 H^b 可以表示为:

$$H^b = [\begin{matrix} H & -QV^T Z \\ V^T Z & \end{matrix}] \quad (4.12)$$

宽度学习系统可以通过增加增强节点的方式对模型就行扩展，以此来提升模型的分类能力，提高分类准确率。但是值得注意的是，在进行模型更新的时候只需要计算增加的那部分阶段的伪逆，不需要计算网络中的整个增强节点的伪逆，从而实现模型的快速更新，提高模型的速度。

4.1.4 特征节点的增量学习

在不同的应用场景中，随着增强节点数量的增加模型仍然无法满足预期要求，这通常是因为特征节点数量不足，映射的特征不足以网络学习进而进行分类。在当前热门的深度学习网络中，当模型无法学习到更高的数据特征时，一般通过增加网络的深度来扩展模型，增加更多的参数约束网络，提升网络的性能，但是这总扩展方式会导致模型更加复杂，训练时间更长。相反宽度学习系统在增加新的特征节点时可以十分容易的构建的网络结构，并且进行增量训练，通过计算新增加节点的伪逆更新权重，而非训练整个模型。下面将详细介绍宽度学习系统特征节点的增量学习。其模型结构图如下图 4.3

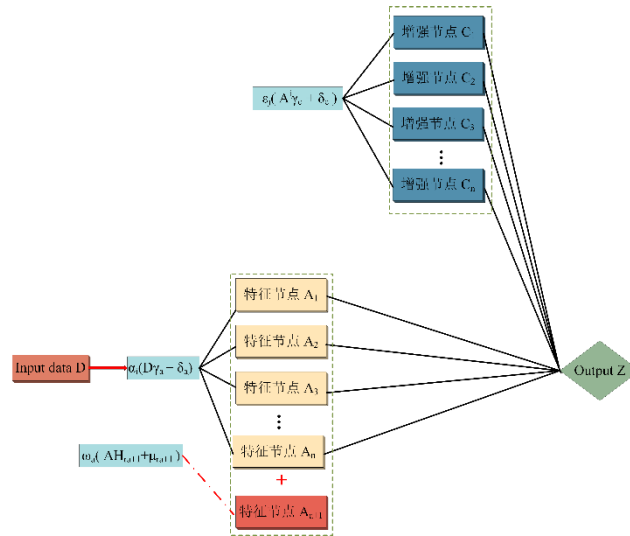


图 4.3 宽度学习系统特征节点增量学习模型

Fig. 4.3 Incremental learning model of feature nodes for broad learning systems

用如下的公式定义新增加的特征节点:

$$M_{a+1} = \omega_a(AH_{r_{a+1}} + \mu_{r_{a+1}}) \quad (4.13)$$

其中 H_{r_a} 和 μ_{r_a} 均为随机生成的矩阵，表示特征节点间相互连接的权重和偏执。接着将特征节点的计算结果送入增强节点中，增强节点对于新增加的特征节点的计算方式如下所示：

$$N_{nb} = [\varphi_b(M_{a+1}H_{u_1} + \mu_{u_1}), \dots, \xi(M_{a+1}H_{u_r} + \mu_{u_r})] \quad (4.14)$$

其中 H_{u_b} 和 μ_{u_b} 均为随机生成的矩阵，表示新增加的特征节点在增强节点中相互连接的权重和偏执。用如下公式将基础模型的输入矩阵与新增加的特征节点、增强节点计算后的结果拼接在一起：

$$P_{a+1}^b = [P|M_{a+1}|N_{nb}] \quad (4.15)$$

通过上述讨论计算新矩阵 P_{a+1}^b 的伪逆，具体计算方式如下所示：

$$(P_{a+1}^b)^+ = \begin{bmatrix} (P)^+ - QV^T \\ V^T \end{bmatrix} \quad (4.16)$$

$$Q = (P)^+[M_{a+1}|N_{nb}] \quad (4.17)$$

$$F = [M_{a+1}|N_{nb}] - PQ \quad (4.18)$$

$$V^T = \begin{cases} (F)^+ & F \neq 0 \\ (1 + Q^TQ)^+V^T(P)^+ & F = 0 \end{cases} \quad (4.19)$$

最终新的的网络模型权重 H^b 可以表示为：

$$H^a = \begin{bmatrix} H - QV^TZ \\ V^TZ \end{bmatrix} \quad (4.20)$$

同增加增强节点的增量学习相同，该方法仅需要计算增加的那部分阶段的伪逆，不需要计算网络中的整个增强节点的伪逆，从而实现模型的快速更新，提高模型的速度。

4.1.5 输入数据的增量学习

通常，在一些特殊的应用场景中，尤其是在边缘计算场景中当模型被训练并且部署在边缘端使用后，经常会出现未出现在训练样本中的新数据出现，例如现在模型针对猫和狗的图像进行了训练，但是当有一张不是猫和狗的图像给到训练好的模型时经常会由于训练特征的不足导致分类失败，常规方法在面对这种情况下只能重新训练模型，加入新的数据再次对模型进行训练更新，这样会消耗大量的计算时间和计算资源，但是宽度学习系统则可以通过对输入数据增量学习的方式快速更新模型，而不需要训练整个模型。其模型结构图如下图 4.4 所示：

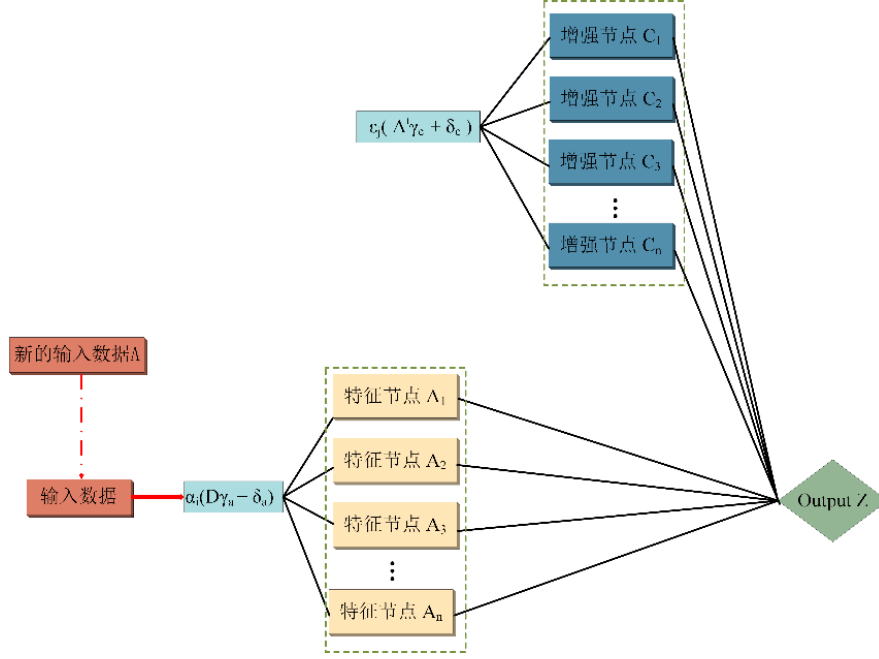


图 4.4 宽度学习系统输入数据增量学习模型

Fig. 4.3 Incremental learning model of input data for broad learning systems

下面将详细介绍宽度学习系统输入数据的增量学习方法。

$$P_a = [\omega_a(X_a H_{r_t} + \mu_{r_t}) \mid \varphi_b(N_b^a H_{u_r} + \mu_{u_r})] \quad (4.21)$$

$$N_b^a = [\omega_a(A_{new} H_{r_1} + \mu_{r_1}), \dots, \omega_a(X_a H_{r_t} + \mu_{r_t})] \quad (4.22)$$

其中 H_{r_t} 、 H_{u_r} 、 μ_{r_t} 、 μ_{u_r} 是随机生成权重和偏执。因此可以用如下公式表示新的输入矩阵：

$$P_{new}^b = \begin{pmatrix} P \\ P_a \end{pmatrix} \quad (4.23)$$

通过上述讨论计算新矩阵 P_{new}^b 的伪逆，具体计算方式如下所示：

$$(P_{new}^b)^+ = [(P)^+ - VQ^T \mid V] \quad (4.24)$$

$$Q^T = P_a P^+ \quad (4.25)$$

$$F = P_a - Q^T P \quad (4.26)$$

$$V^T = \begin{cases} (F)^+ & F \neq 0 \\ (1 + Q^T Q)^+ V^T (P)^+ & F = 0 \end{cases} \quad (4.27)$$

最终新的网络权重可以通过下面公式进行计算：

$$H^{new} = H + (Z_{new} - PH)V \quad (4.28)$$

同增上述两种增量学习相同，该方法仅需要计算增加的那部分阶段的伪逆，不需要计算网络中的整个增强节点的伪逆。其模型结构如图 4.4 所示：

4.2 实验流程与结果分析

4.2.1 实验环境及设备

本节基于宽度学习系统的车辆分类实验环境配置如表 4.1 所示：

表 4.1 车辆分类实验环境信息

Tab. 4.1 Information on the environment of the vehicle classification experiment

名称	信息
云端操作系统	Windows10
编程语言	Python3.8
云端 CPU	I7-11700k
云端内存	32G
边缘端操作系统	Linux
边缘端设备	树莓派 4B
边缘端内存	4G

在本实验中使用树莓派来模拟边缘端设备，树莓派是一种只有信用卡大小的微型计算机，在树莓派主板中用于大量丰富的硬件接口，例如网线接口、HDMI 接口、USB 接口、摄像头接口等等，丰富的接口也使得树莓派适用于大多数的边缘场景。树莓派可以在 SD 卡中或者外置硬盘中安装 ARM 平台的操作系统，例如 Linux 大多数的发行版本，Android、Windows IoT 等。在本实验中使用的是 Linux 操作系统。树莓派的运算能力以及存储资源和大多数边缘端设备相差不多，其次体积更小，但是更小的体积带来的代价就是 CPU 的算力不高，因此树莓派要求部署的算法更加轻量级，所需的计算量更小。并且树莓派上用于 Python 开发的库更加完善，对于不同版本以及不同系统的兼容性更好，由于 Python 语言本身具备良好的跨平台性，在云端训练好的模型可以在树莓派上部署。本实验的宽度学习系统刚好满足上诉要求。本实验所采用的的树莓派如下图 4.8 所示：

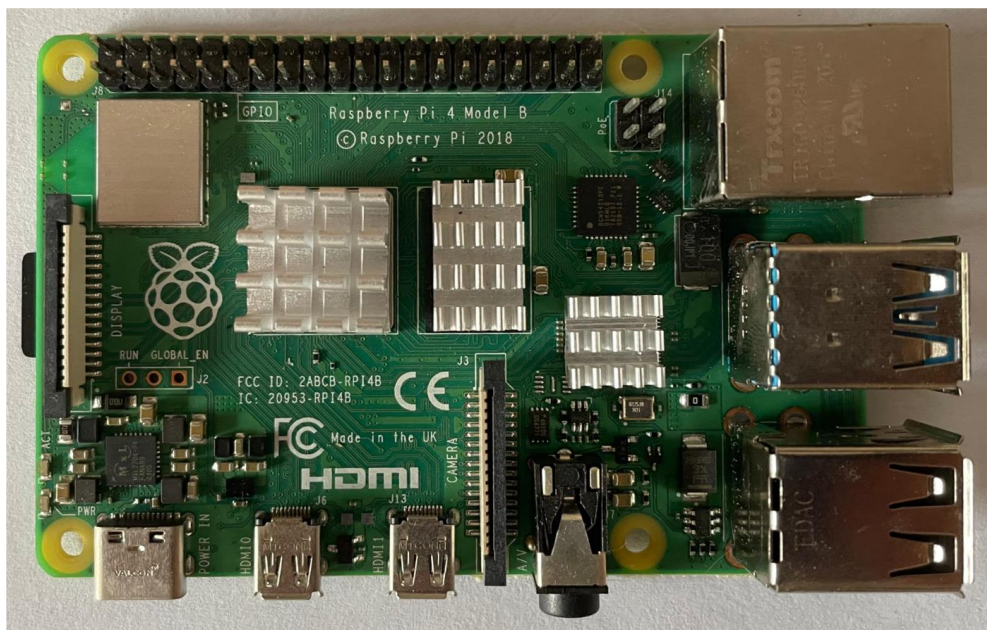


图 4.8 树莓派 4B

Fig. 4.8 Raspberry Pi 4B

4.2.2 实验流程

(1) 获取车辆数据

本次实验主要是对不同的车辆类型进行识别和分类，在实验开始之前首先要获取实验需要的数据，需要的数据集为现实道路的车辆数据，本次实验的数据来源是上述章节中已经介绍的自行采集的数据集。经过处理后的数据集如图 4.5 所示：



图 4.5 车辆分类数据

Fig. 4.5 Vehicle classification data

(2) 云端训练

用于车辆分类的宽度学习系统的训练过程是在以 i7-11700k 为 CPU，运行内存 32G 的 windows 系统模拟的云端服务器上进行的，首先宽度学习系统接收到了上一步骤提到真实道路车辆数据，其次，将真实道路车辆数据送入宽度学习系统的特征节点中用于特征映射，然后将得到的结果在送入增强节点中进行计算，将上述两部分拼接在一起乘以网络的权重得到最终的车辆分类结果。到此为止，一个用于真实道路车辆数据分类的基础宽度学习系统就训练完成，但是经过对实验结果的观察，车辆分类的准确率无法达到对实验结果的预期，因此使用增加增强节点数量的方法对宽度学习系统进行扩展，以此来提高车辆分类的准确率，由于宽度学习系统自身的特性，在基础模型的基础上，使用循环的方法逐次增加增强节点的数量，每次循

环都会计算一次新的伪逆和权重，以此更新网络模型寻找到更好的车辆分类结果。由于宽度学习系统的特性，仅需要对新增加的节点计算伪逆，进而更新网络权重并非重新训练整个模型，本实验的训练速度很快，模型更加轻量化。最后将训练好的模型保存。整个训练过程如图 4.6 所示。

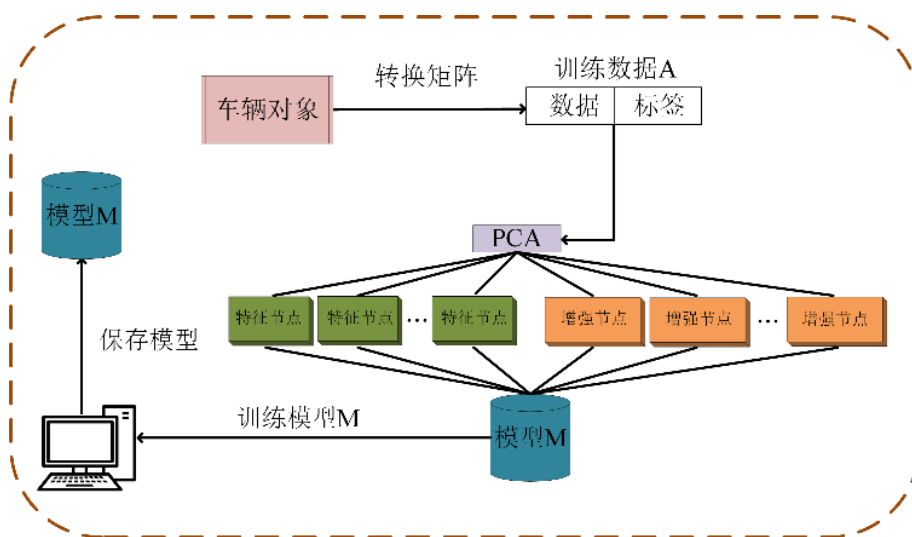


图 4.6 车辆分类训练流程

Fig. 4.6 Vehicle classification training process

(3)边缘端部署

本文希望可以将模型部署到一些边缘端的设备中去，所以在本实验中使用树莓派来模拟边缘端设备。在云端训练基础模型之后，将模型保存，在树莓派中接收并加载已经训练好的模型，将其转换成开放神经网络交换（Open Neural Network Exchange, ONNX）格式，接着加载新的验证数据，将验证数据送入训练好的宽度学习系统模型中，根据训练好的权重直接计算分类结果。推理过程如图 4.7 示。

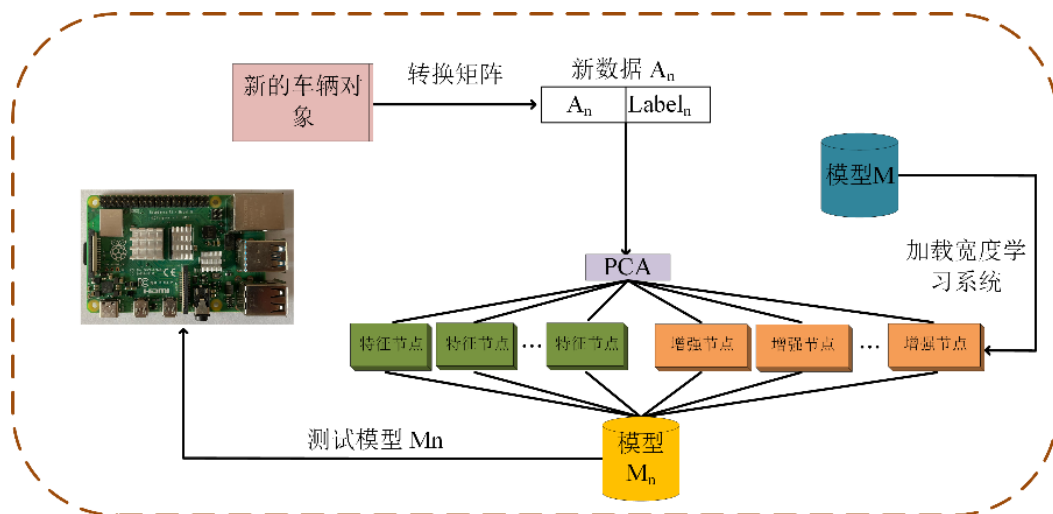


图 4.7 车辆分类边缘端部署流程

Fig. 4.7 Vehicle classification edge end deployment process

4.2.3 实验指标

在本实验中，采用准确率（Accuracy）作为车辆分类实验的评价指标，准确率的定义是在给的测试集中分类器正确分类的样本数与总样本数之间的比值。用如下数学公式表示：

$$Accuracy = \frac{TP+TN}{SUM} \quad (4.29)$$

其中 TP 代表正样本被预测成正样本，TN 代表负样本被预测为负样本，TP+TN 表示所以预测正确的样本数，SUM 代表总样本数，用准确率来记录车辆分类实验中宽度学习系统的表现。

4.2.4 实验结果

为了验证本文所提出方法的有效性，在采集的数据集上进行了实验。将数据集的 70% 作为训练集，20% 作为测试集，10% 作为验证集。将数据集的训练部分送入网络在云端进对基础模型进行训练。基础模型的准确率为 69.2%，由于模型的准确率无法达到最初的预期，因此对模型进行增加增强节点的增量学习，循环增加增强节点的数量，每次增加 100 个，循环 120 轮，分类准确率从 69.2% 上升到 77.2%。实验结果如图表 4.2 所示：

表 4.2 增加增强节点的增量学习实验结果

循环轮次	累计增加增强节点数	准确率
10	1000	70.2%
20	2000	72.4%
30	3000	73.4%
40	4000	74.7%
50	5000	74.7%
60	6000	75.3%
70	7000	75.6%
80	8000	75.9%
90	9000	76.4%
100	10000	76.5%
110	11000	76.8%
120	12000	77.2%

为了证明增加增强节点的增量学习模型的有效性，将基础模型的增强节点数量调整至和增加增强节点的增量学习模型的增强节点数相同，进了实验，基础模型与增量模型的实验结果图如下图 4.9 所示，通过实验结果可知，即使宽度学习系统的基础模型与增量模型保持相同的增强节点的数量，增量学习模型在车辆分类实验中的准确率仍然高于基础模型的车辆分类准确率。在对宽度学习系统进行增加增强节点的增量学习实验中，随着增强节点数量逐渐增加，分类准确率在前几轮的实验中增长比较明显，在接近 120 轮的时候准确率基本持平，为了平衡准确率和训练时间在本

实验中选择循环增加增强节点的轮次为 120 轮。

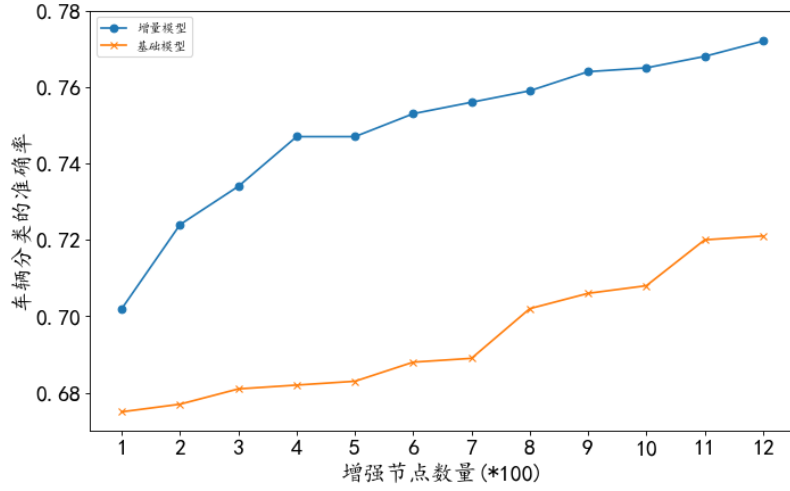


图 4.9 基础模型与增量模型的实验结果

Fig. 4.9 Experimental results of the base and incremental models

4.3 本章小结

在本章中使用宽度学习系统对真实道路车辆进行分类，并且部署在树莓派上。为了验证宽度学习系统的有效性，开展了关于宽度学习系统对于真实道路车辆的分类能力的研究，接收到真实道路车辆数据集后，训练基础模型，由于其准确率无法达到预期使用增加增强节点的增量学习方法对基础模型进行扩展训练，分类准确率从 69.2% 上升到 77.2%。基本满足预期的分类准确率。成功证明了该方法可以作为多种道路场景下的轻量化道路车辆细粒度检测技术的分类。

第 5 章 多种道路场景下的轻量化道路车辆细粒度检测技术研究

在本章中将详细深入探讨多种道路场景下的轻量化道路车辆细粒度检测技术研究的整体架构情况，专注于不同道路场景下的应用。介绍针对多种复杂环境的轻量级车辆细粒度检测技术的研究方案，并着重展示在这些方案中数据处理的关键作用。具体来说，将剖析经过数据处理后的车辆细粒度检测技术在云端的训练过程，以及在边缘端的实际测试效果。通过这一全面的架构分析，旨在为实现在多样化道路场景下高效可靠的车辆检测技术提供深入的理解和有效的方法。

5.1 轻量化道路车辆细粒度检测技术

在本节中将详细介绍轻量化道路车辆细粒度检测技术的整体框架，首先进行数据预处理，获取行车记录仪的数据，通过数据预处理得到清洗后的数据和对应的标签，接着获取处理后的数据训练车辆检测模型，当模型准确率不符合预期的时候对其参数进行微调，若还是不符合预期则进行增量训练，反之生成 pt 模型，最后一步将 pt 模型转换成 onnx 模型将其部署在边缘端设备上。其模型结构流程图如图 5.1 所示，

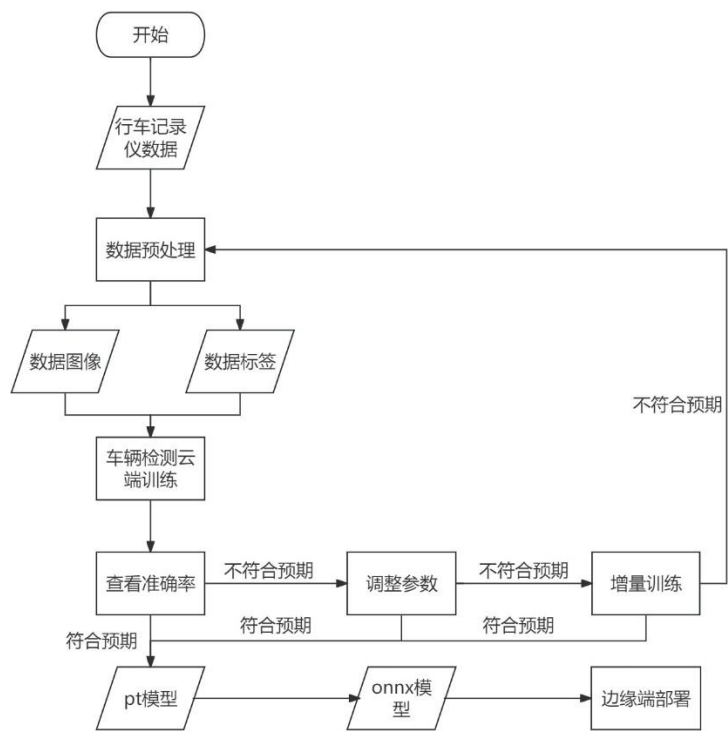


图 5.1 轻量化道路车辆细粒度检测技术流程图

Fig. 5.1 Flowchart of lightweight road vehicle fine-grained detection technology

算法流程如表 5.1 所示，与图 5.1 不同的是，在表 5.1 中详细介绍了整个轻量化道路车辆细粒度检测技术的算法流程。

表 5.1 算法流程

Tab. 5.1 Algorithm flow

车辆细粒度检测算法

输入：行车记录仪数据 N ，感兴趣区域数量 L ，增强次数 E 。

输出：感兴趣区域 X_j ，类别 Z

1. 创建列表 N 存放行车记录仪视频
2. For N do
3. 提取关键帧 m
4. 重构图像 m 的大小
5. End for
6. 创建列表 X 存放图像 m
7. For X do
8. For L do
9. 提取感兴趣区域 x_j
10. End for
11. End for
12. 创建列表 J 存放 x_j
13. 通过公式 (4.3) 计算映射特征 M_a
14. 通过公式 (4.4) 计算增强特征 N_b
15. 通过公式 (4.2) 计算伪逆 P^+
16. 通过公式 (4.6) 计算权重 H
17. 保存模型 M
18. 转换成 ONNX 模型 M_l
19. 通过公式 (4.5) 得到分类结果 Z
20. 通过 J 获得检测框 X_j

5.1.1 读取并处理数据

首先轻量化道路车辆细粒度检测技术的第一步是要获取训练数据，采集到的初始数据是车辆的行车记录仪的视频数据，因此在该步骤中，要对行车记录仪视频进行处理，处理后得到最终模型需要的训练数据。

通过提取关键帧的方式从视频中提取到想要的图像，然后对数据进行清洗，对于重复图像进行去重，对于异常数据进行删除处理，保证数据质量。接着对数据进行集成，将每个行车记录仪中清洗后得到的数据集成到一起组成新的数据样本。最后对数据进行转换，因为不同行车记录仪的不同，每张图像的分辨率不同，存在差异性，要对存在差异的数据统一分辨率大小，最后通过随机反转、选择等方式对数据进行增强，提升模型的泛化性。

接着将得到的数据集进行随机切割，按照固定比例分成训练集、测试集、验证集。其中训练集和测试集将会发送至云端用于模型的训练和测试，验证集将会发送至边缘端用于模型的推理，并且训练集、测试集和验证集不会存在交集。

数据处理模型结构图如下图 5.2 所示：

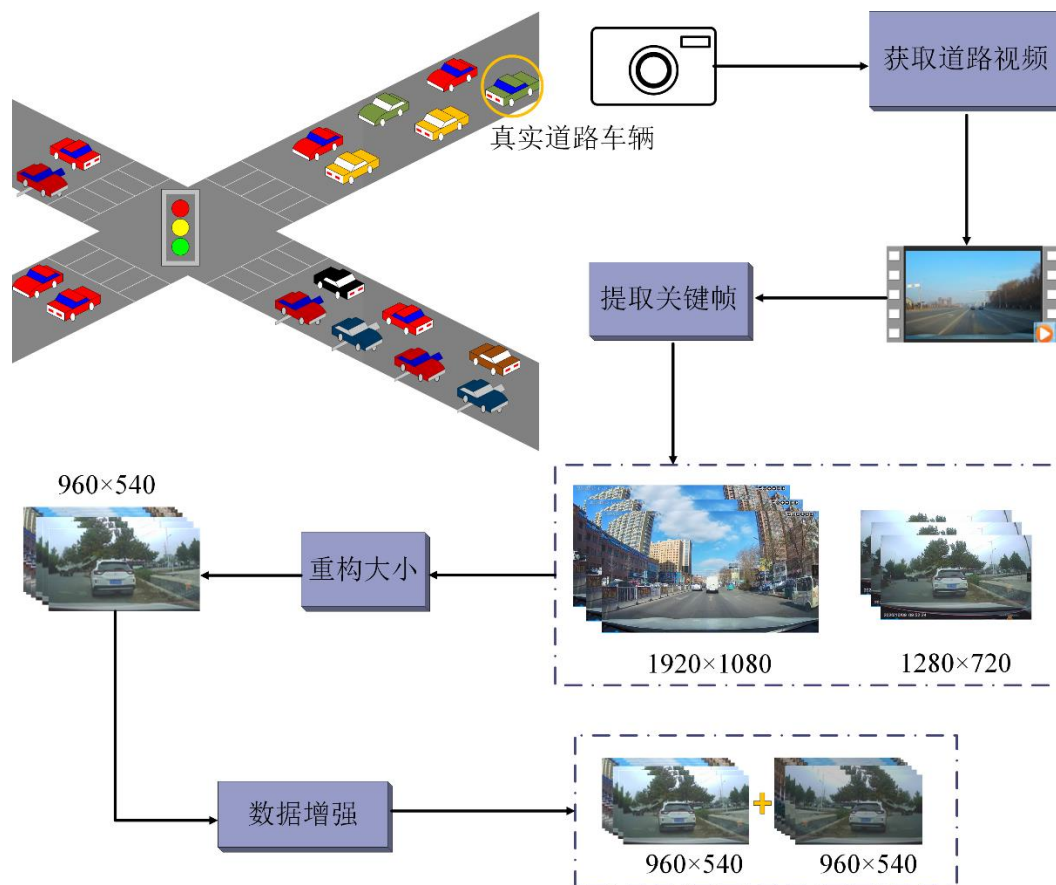


图 5.2 数据预处理模型结构图

Fig. 5.2 Data preprocessing model structure diagram

最终将处理后的训练集和测试集发送到云端准备进行训练。

5.1.2 生成候选框

在云端接收到训练的数据首先通过卷积层、激活层、池化层等网络层进行特征提取。这一步骤的设计是为了能够学习到图像中车辆的关键特征。例如纹理、形状等。通过这些方法可以获取到更高维度的特征、更加有利的特征。这为后续的车辆检测提供了强有力的基础。

接着将提取到的特征图送入区域建议网络，生成候选框，在这一步骤中，生成的物体框的尺寸和大小是固定的，以此确保在不同的检测场景中都可以成功获取检测对象。具体来说，使用一种映射方法，通过将特征图中的每一个点与原图的感受野中心点做映射，生成一个基准点，然后以基准点为中心选取若干个不同尺寸和比例的物体框。接着要为每一个生成的物体框标定一个标签。具体来说，标签标定规则如下：1 代表前景物体，在本文中指车辆，0 代表背景，而-1 则表示无效。因此，在这一步骤中进行的是对物体框内的物体进行二分类，即该物体框中是否为车辆。

最后要对物体框的位置进行调整，使其更贴近真实框的位置，更完整的将前景

对象包裹起来。使用位置平移的方法确定中心点坐标，使用尺度放缩的方法确定高宽。具体过程如下：

真实框可能获取到了很多个预测框，但是这些框也能存在各种各样的问题，最后希望每一个物体只有一个最合理的预测框，因此需要对多个预测框进行选择删除，常见的两种处理方式：

(1) 超出原始图像边界的预测框：将预测框映射回原图上，计算一下预测框是否存在大面积超出边界范围的情况，将这类预测框删除。

(2) 存在重合区域的预测框：对重合部分的预测框进行非极大值抑制操作，具体来说就是首先将所有预测框按照置信度进行排序，选取一个置信度最高的预测框，遍历其他预测框，如果与该预测框的重叠面积大于一定的面积，删除此预测框。

根据候选框的位置信息，从原图中切割下相应的车辆图像，切割后的数据如下图 5.3 所示。为车辆分类训练所需的数据样本，也就是单独的车辆图像。将这些图像和标签信息一起用于后续车辆分类的训练，以至于模型可以学习到准确的车辆细致化特征。



图 5.3 数据预处理模型结构图

Fig. 5.3 Data preprocessing model structure diagram

5.1.3 车辆分类

在上一步骤中，获取到了包含车辆对象在内的单一图像，只有车辆对象信息而没有具体类别信息，因此在这一步中，要将上一步得到的车辆对象进行分类，获取到

最终的车辆细粒度类别。

将获取到的图像数据首先进行归一化，归一化可以将图像的像素值转换到 0-1 之间，这样有利于模型加速迭代。当图像的像素值分布在很大的范围中，模型需要多次迭代才能找到最优解，但是经过归一化处理后的结果则会减少迭代次数，提升模型的效率，同时，归一化处理后的数据像素值分布更加稳定，模型训练也更加稳定，图像的像素值的量级通常也是不同的，未进行归一化处理会导致模型在训练的时候很难学习到不同量级间的相同特征，会导致特征提取不精确，从而影响训练结果，而将图像归一化后所有像素值均在一个十分稳定的范围内，更有利于模型学习到特征之间的相关性。归一化的处理也在很大程度上避免了模型过拟合的问题，不进行归一化图像数据的方差可能会很大，这会导致模型过拟合，泛化能力会下降。

将得到归一化后的图像数据转换成矩阵，并且将矩阵从二维转换成一维矩阵，将随之而来的标签类别信息和一维图像矩阵绑定，一起送入宽度学习系统对车辆进行细致化分类，数据首先进入特征节点通过特征节点和增强节点对数据的学习，计算权重，最终得到车辆的分类结果。

车辆检测模型结构图如下图 5.4 所示

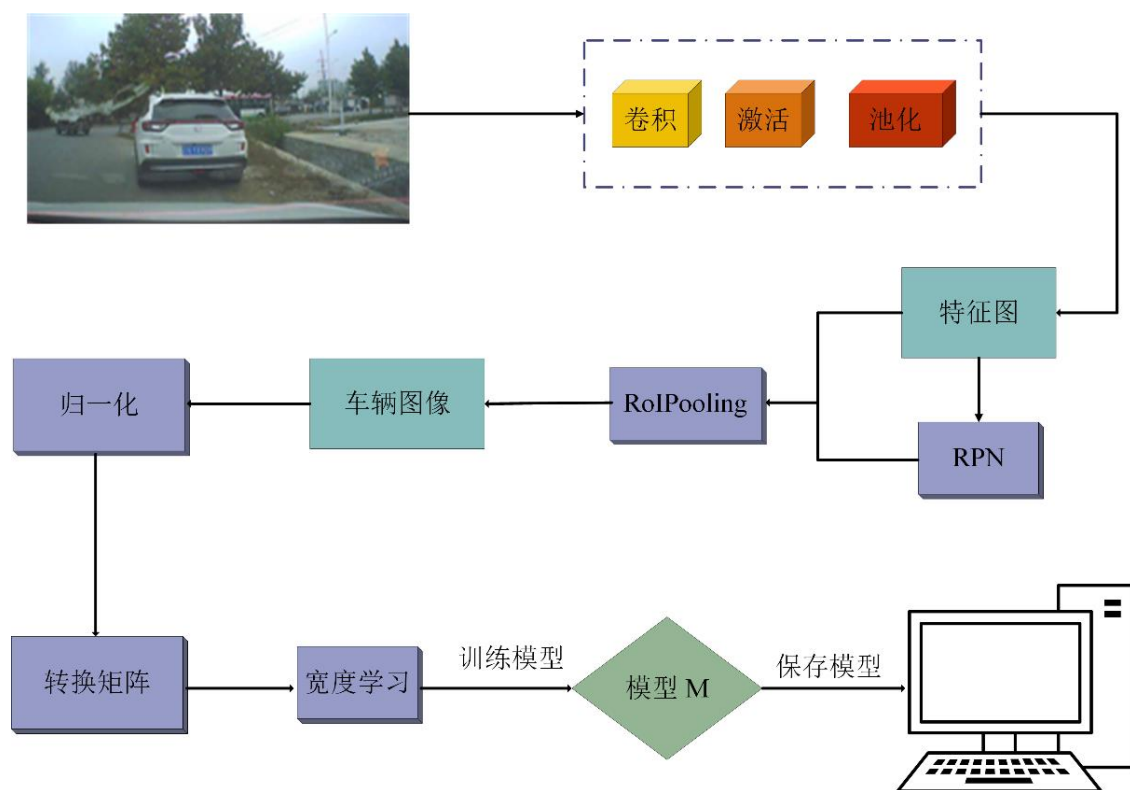


图 5.4 车辆检测模型结构图

Fig. 5.4 Structure diagram of vehicle detection model

5.1.4 边缘端部署

最终，经过上述步骤训练好的模型将被保存，以便在边缘场景中中进行有效的车辆细粒度检测。边缘端部署模型结构图如下图 5.5 所示：

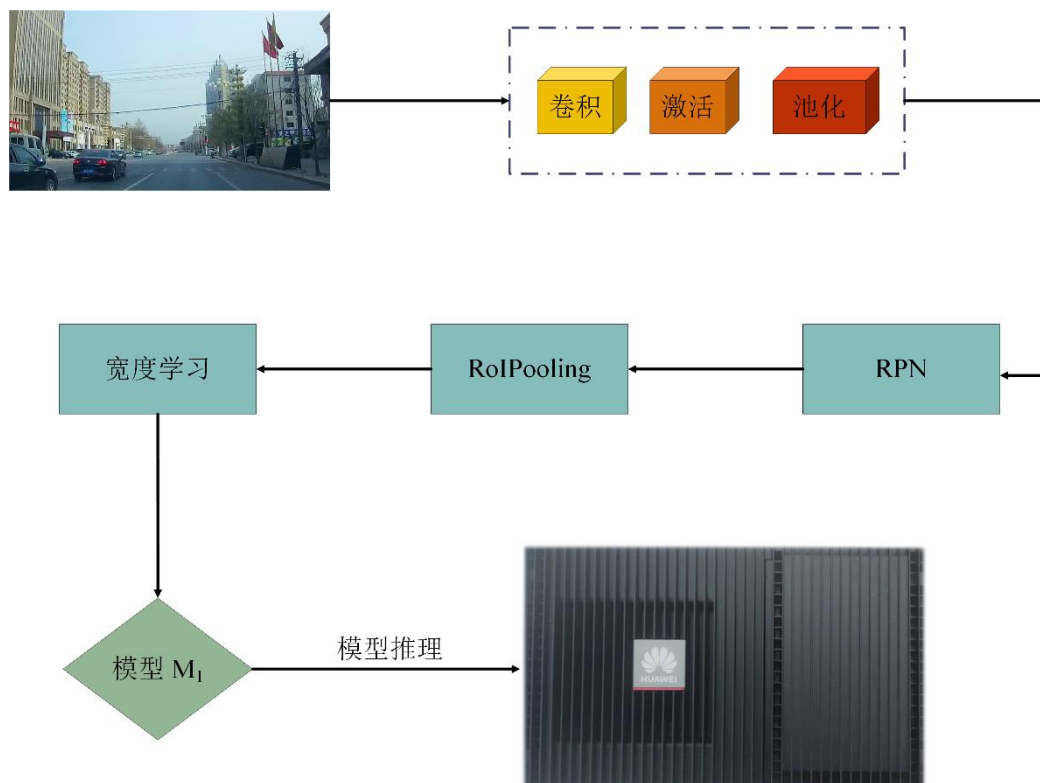


图 5.5 边缘端部署模型结构图

Fig. 5.5 Structure diagram of the edge-side deployment model

模型保存的过程是深度学习流程中至关重要的一环，它保留了模型在训练数据上学到的权重和参数，使其能够在未来的推理任务中发挥作用。在该步骤中将系统部署在边缘端设备上，用于评估系统的性能。由于车辆检测技术经常需要部署在一些计算能力更弱，存储资源更少的边缘端设备上，因此迫切的需要一个更加轻量级的车辆检测技术，提出的方案可以满足上诉要求。

在进行推理之前，不可或缺的首要步骤是获取用于推理的数据以及已经经过充分训练的模型。这两者经过一番精心的预处理后，会一同传送至边缘端，为后续搭建起坚实的基础。一旦边缘端成功接收到模型和数据，将迅速启动加载模型的过程，并精准地从数据中提取特征。这些特征的获得为接下来的车辆细粒度检测提供了关键支持，确保车辆检测能够在高效、准确的基础上进行推理操作。

在该步骤中，首先接收到已经训练好的模型和验证集，由于模型不能直接部署到 HUAWEI Atlas 500 上，要先将其转化成通用神经网络的 onnx 模型格式，然后启动模型，对验证集进行推理。

5.3 实验与结果分析

在本节中介绍多种道路场景下的轻量化道路车辆细粒度检测技术的云端训练以及边缘端推理所需要的实验环境，衡量实验结果所使用的实验指标以及所取得的实验结果

5.3.1 实验环境及设备

本文多种道路场景下的轻量化道路车辆细粒度检测技术实验环境配置如表 5.2 所示：

表 5.2 实验环境配置

Tab. 5.2 Experimental Environment Configuration

名称	信息
云端操作系统	Windows10
编程语言	Python3.8
云端 CPU	I7-11700k
云端内存	32G
边缘端操作系统	Linux
边缘端设备	Atlas 500
边缘端内存	4G

在本实验中，云端训练在 Windows 系统下进行，使用 HUAWEI Atlas 500 来模拟边缘端设备。其参数如下表 5.3 所示：

表 5.3 HUAWEI Atlas 500 参数

Tab. 5.3 HUAWEI Atlas 500 parameters

名称	信息
AI 算力	22/16 TOPS INT8
内存规格	LPDDR4X, 8GB/4GB, 最大 51.2GB/s
图像编解码能力	1080P24FPS
图像最大分辨	4096*2160
接口	HDMI 接口和 USB 接口
典型功耗	40W
环境条件	-40 ℃~ 60 ℃
结构尺寸	45mm*355 mm*220mm

HUAWEI Atlas 500 是华为公司推出得用于边缘计算场景中新一代人工智能推理设备，它采用华为公司自研得 Ascend 处理器，这是专为人工智能计算优化得处理器架构，支持多种深度学习框架，包括 TensorFlow、PyTorch 以及华为公司自研的 MindSpore 框架。对于使用者来说可以更快选择熟悉的框架进行开发部署。

HUAWEI Atlas 500 设计初衷是希望可以在边缘端设备上执行人工智能推理任务，这样可以避免云端与边缘端相互通信所产生信道堵塞、延迟等问题，更有利于提升人工智能任务的响应速度，这刚好符合需求。HUAWEI Atlas 500 具有十分强大的应用场景，例如智慧变电站、智慧交通、智慧社区、环境监控、智能制造、智慧营业厅、无人零售等场景。

本实验所采用的的 HUAWEI Atlas 500 如下图 5.6 所示



图 5.6 HUAWEI Atlas 500

Fig. 5.6 HUAWEI Atlas 500

5.3.2 实验指标

在本实验中采用平均准确率均值（mAP）作为轻量化道路车辆细粒度检测技术性能的度量指标，mAP 是每个类别平均准确率（AP）的平均值。在这一度量标准中，AP 是由精确率-召回率曲线（PR 曲线）和 X 轴、Y 轴所围成的面积计算得出的，其中 X 轴和 Y 轴分别表示召回率和精确率。PR 曲线下方的面积即为 AP，而 mAP 则为所有类别 AP 的平均值。这种度量方法能够全面评估在多种道路场景下车辆检测算法的性能，为准确率和召回率提供了直观的综合指标。具体计算精确率和召回率的公式如下：

$$precision = \frac{TP}{TP+FP} \quad (5.9)$$

$$Recall = \frac{TP}{TP+FN} \quad (5.10)$$

其中，TP 代表被模型正确预测为阳性的正样本数量，FP 表示被模型错误预测为阳性的负样本数量，而 FN 则表示被模型错误预测为阴性的正样本数量。

5.3.3 实验结果

多种道路场景下的轻量化道路车辆细粒度检测技术在云端进行训练，基础模型 mAP 为 0.68。加入增强节点对系统进行增量学习，mAP 从 0.654 提高到 0.712，增量

模型的性能优于基本模型。随着增强节点数量的增加，mAP 的结果如表 5.4 所示。N3 是增加的增强节点数量。

表 5.4 随着增强节点数量的增加 mAP 的值

Tab. 5.4 The value of mAP as the number of enhanced nodes increases

循环轮次	累计增加增强节点数	mAP
10	1000	65.3%
20	2000	66.8%
30	3000	67.2%
40	4000	67.5%
50	5000	67.7%
60	6000	67.8%
70	7000	68.5%
80	8000	68.8%
90	90000	69.0%
100	10000	69.3%
110	11000	69.5%
120	12000	70.2%

为了证明增加增强节点的增量学习模型的有效性，将基础模型的增强节点数量调整至和增加增强节点的增量学习模型的增强节点数相同，进了实验，基础模型与增量模型的实验结果图如下图 5.7 所示，

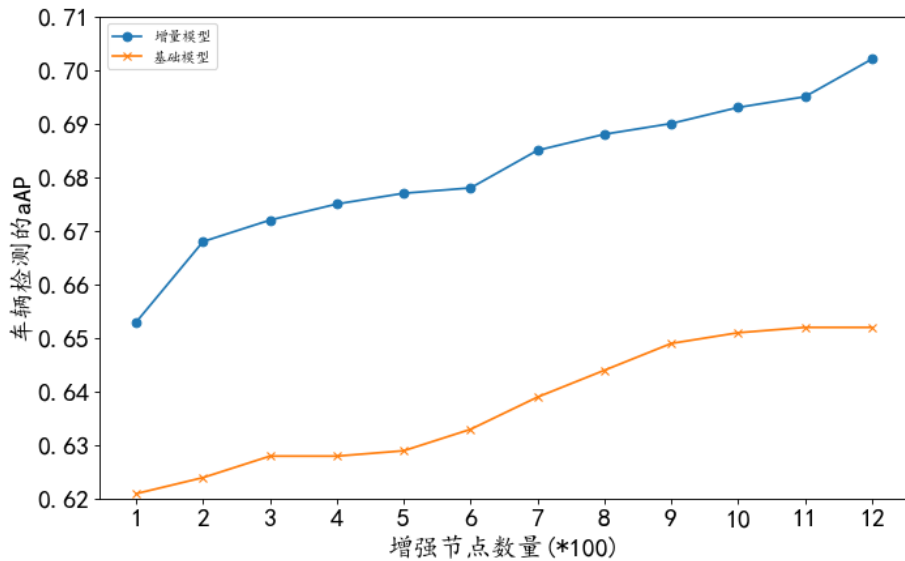


图 5.7 基础模型与增量模型的实验结果图

Fig. 5.7 Graphs of experimental results for the base model and the incremental model

通过在 HUAWEI Atlas 500 上进行性能评估，将训练好的模型保存在云端，随后将保存的模型连同用于推理的车辆数据一起发送到 HUAWEI Atlas 500。之后用保存的模型对数据进行推理，最终模型的 mAP 达到了 0.712。

在将模型训练好后对验证集进行推理，并且生成可视化结果，从可视化结果从来看，轿车、SUV、商务车、卡车等均可以被成功检测到。

最终的可视化结果如下图 5.8 所示：

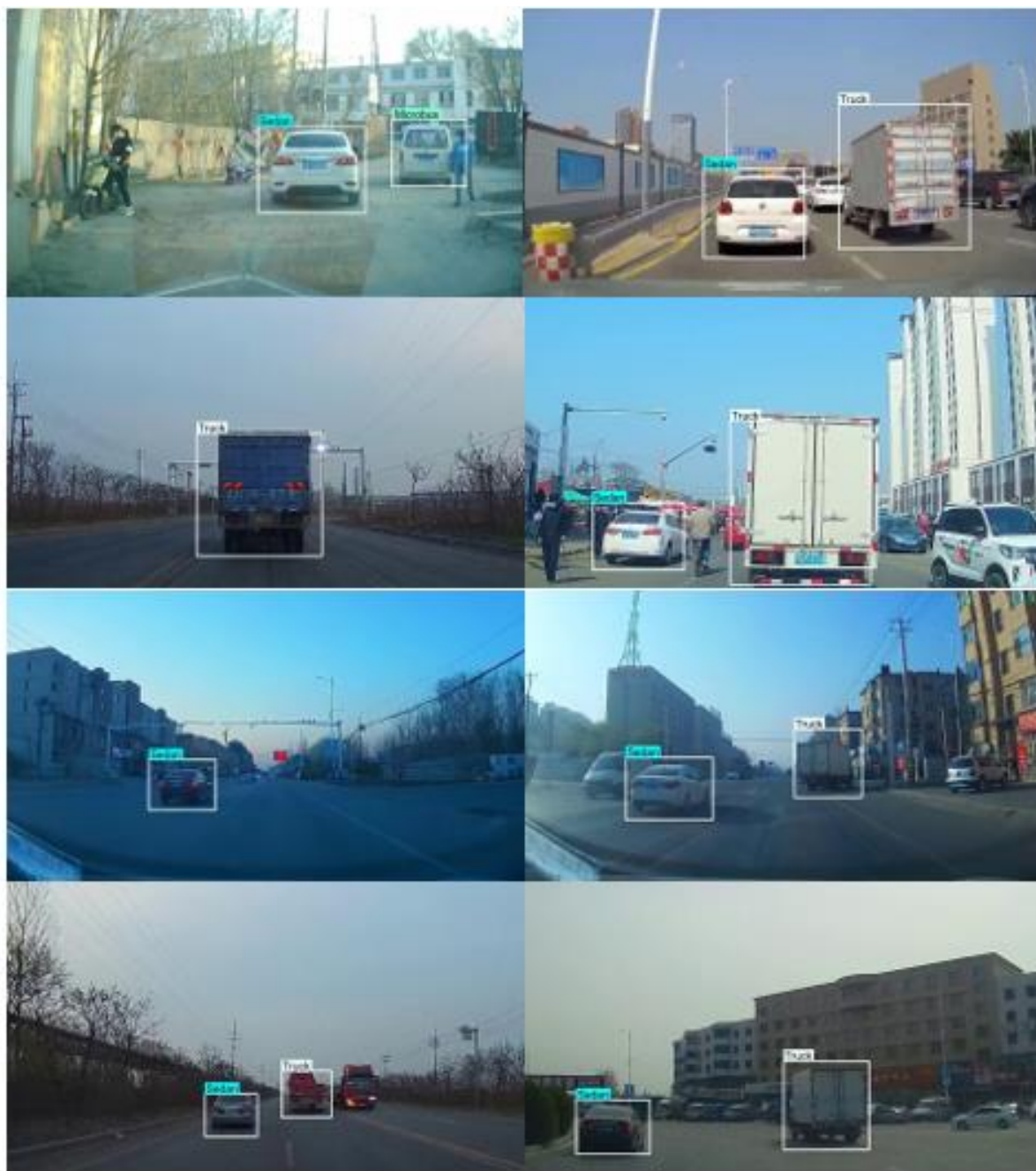


图 5.8 可视化结果

Fig. 5.8 Visualizing the results

在保证实验条件相同的前提下，在相同的数据集上分别对 Faster R-CNN、SSD 和 YOLOV5 进行实验，通过实验结果分析上述方法在 mAP 和训练时间方面的表现，具体结果可见于图 5.9，其中在 mAP 方面，虽然 YOLOV5 比本文的方法高出 0.05，Faster R-CNN 也略胜一筹，超过了 0.01。然而，在训练速度方面，本文方法的训练速度比它们快了二到三倍。另外，虽然 SSD 在具有较快的训练速度，但由于其模型特性，其 mAP 仅达到 0.532。

从这一对比实验中看出，本文提出的多种道路场景下的轻量化道路车辆细粒度检测技术在综合性能上表现优异，通过在 mAP 和训练时间之间取得平衡，为解决实际

问题提供了更为可行和高效的解决方案。

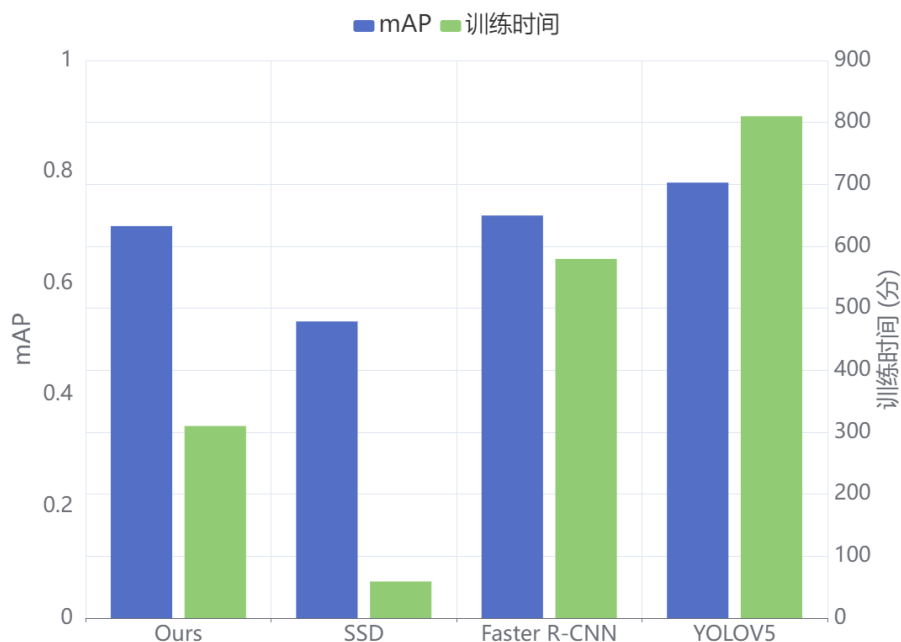


图 5.9 对比实验结果图

Fig. 5.9 Comparison of experimental results

5.4 本章小结

在本章中讨论关于多种道路场景下的轻量化道路车辆细粒度检测技术的整体架构，深入探讨其在云端的训练过程，包括数据的准备、特征提取、获取检测框、分类等步骤。其次将其部署在 HUAWEI Atlas 500 上来模拟真实的车辆检测场景。为了验证本文提出方法的有效性进行了一系列的实验，首先通过增加增强节点来扩展模型，与基础模型相比，增量模型的 mAP 从 70.2% 上升到 77.2%。同时和 Faster R-CNN、YOLOV5、SSD 等优秀模型进行对比，本文的方法的速度比它们快了二到三倍。

第 6 章 总结

本文提出了一种多种道路场景下的轻量化道路车辆细粒度检测技术研究，使用更简单的模型结构，提升车辆检测的响应速度，并且可以将模型部署到边缘端设备上，避免云端与边缘端相互通信产生的时延，避免自动驾驶车辆由于反应速度慢而产生的交通事故。

本文的多种道路场景下的轻量化道路车辆细粒度检测技术主要包括三部分：数据获取与预处理、云端训练、边缘端推理。数据获取的来源主要是真实车辆的行车记录仪视频，这样更加贴合实际的车辆检测场景，对数据进行预处理后获取到包括高速、快速路、高架、省道等多种道路场景以及清晨、正午、傍晚、夜晚等多种时间下的车辆数据。在云端训练时，通过增加增强节点的增量学习方法扩展模型，提升系统的检测精度。值得注意的是，当需要训练增量模型时，仅需要对增加的部分进行训练，而非训练整个模型。本文将多种道路场景下的轻量化道路车辆细粒度检测技术部署在 HUAWEI Atlas 500 上以此来模拟边缘端设备，避免模型在云端推理时和边缘端通信产生的时延问题。

与其他车辆检测方法相比，本文的方法在训练速度上提升了二到三倍。但是在训练速度方面仍有提高的可能，在未来的工作中将考虑对系统的其他部分做轻量化处理，在保证检测精度的同时提高训练速度。

参考文献

- [1]. 吴博峰. 全国机动车保有量达 4.3 亿辆 [N]. 中国消费者报, 2023-10-13(003). DOI:10.28867/n.cnki.nxfzb.2023.002049.
- [2]. Pyykönen P, Laitinen J, Viitanen J, et al. IoT for intelligent traffic system[C]//2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2013: 175-179.
- [3]. Chen B, Sun D, Zhou J, et al. A future intelligent traffic system with mixed autonomous vehicles and human-driven vehicles[J]. Information Sciences, 2020, 529: 59-72.
- [4]. Modi Y, Teli R, Mehta A, et al. A comprehensive review on intelligent traffic management using machine learning algorithms[J]. Innovative infrastructure solutions, 2022, 7(1): 128.
- [5]. López-Sastre R J, Herranz-Perdigüero C, Guerrero-Gómez-Olmedo R, et al. Boosting multi-vehicle tracking with a joint object detection and viewpoint estimation sensor[J]. Sensors, 2019, 19(19): 4062.
- [6]. Ng S C, Kwok C P. An intelligent traffic light system using object detection and evolutionary algorithm for alleviating traffic congestion in hong kong[J]. International journal of computational intelligence systems, 2020, 13(1): 802-809.
- [7]. Feng D, Harakeh A, Waslander S L, et al. A review and comparative study on probabilistic object detection in autonomous driving[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(8): 9961-9980.
- [8]. Feng D, Haase-Schütz C, Rosenbaum L, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(3): 1341-1360.
- [9]. 邹伙宗, 邓守城. 基于深度学习的车辆目标检测算法综述[J]. 时代汽车, 2023(15): 16-18.
- [10]. Zou Z, Chen K, Shi Z, et al. Object detection in 20 years: A survey[J]. Proceedings of the IEEE, 2023.
- [11]. Zhao Z Q, Zheng P, Xu S, et al. Object detection with deep learning: A review[J]. IEEE transactions on neural networks and learning systems, 2019, 30(11): 3212-3232.

- [12].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.
- [13].Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [14].Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [15].Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [16].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.
- [17].Han X. Modified cascade RCNN based on contextual information for vehicle detection[J]. Sensing and Imaging, 2021, 22(1): 19.
- [18].Wang L, Liao J, Xu C. Vehicle detection based on drone images with the improved faster R-CNN[C]//Proceedings of the 2019 11th International Conference on Machine Learning and Computing. 2019: 466-471.
- [19].Cao J, Song C, Song S, et al. Front vehicle detection algorithm for smart car based on improved SSD model[J]. Sensors, 2020, 20(16): 4646.
- [20].Ghosh R. A modified yolo model for on-road vehicle detection in varying weather conditions[J]. Intelligent Computing and Communication Systems, 2021: 45-54.
- [21].Leung H K, Chen X Z, Yu C W, et al. A deep-learning-based vehicle detection approach for insufficient and nighttime illumination conditions[J]. Applied Sciences, 2019, 9(22): 4769.
- [22].Ho N, Pham M, Vo N D, et al. Vehicle detection at night time[C]//2020 7th NAFOSTED Conference on Information and Computer Science (NICS). IEEE, 2020: 250-255.
- [23].Lin C T, Huang S W, Wu Y Y, et al. GAN-based day-to-night image style transfer for nighttime vehicle detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(2): 951-963.

- [24].Mittal U, Chawla P, Tiwari R. EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models[J]. Neural Computing and Applications, 2023, 35(6): 4755-4774.
- [25].苏静,刘伊智.基于改进 YOLOv4 的车辆目标检测算法[J].天津科技大学学报,2023,38(03):61-67.DOI:10.13364/j.issn.1672-6510.20220180.
- [26].陈秀锋,王成鑫,吴阅晨等.改进 YOLOv5s 算法的车辆目标实时检测方法[J/OL].哈尔滨理工大学学报:1-9[2023-07-11].<http://kns.cnki.net/kcms/detail/23.1404.N.20230601.1430.030.html>.
- [27].任杰,李钢,赵燕姣等.基于改进 Faster RCNN 的城市道路货车检测[J].计算机系统应用,2022,31(12):316-321.DOI:10.15888/j.cnki.csa.008900.
- [28].廖慕钦,周永军,汤小红等.基于 SSD-MobilenetV3 模型的车辆检测[J].传感器与微系统,2022,41(06):142-145.DOI:10.13873/J.1000-9787(2022)06-0142-04.
- [29].CHEN C P, LIU Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture[J]. IEEE transactions on neural networks and learning systems, 2017, 29(1): 10-24.
- [30].苏嘉铭. 基于正则化技术的宽度学习系统网络结构改进方法 [D]. 中国矿业大学,2022.DOI:10.27623/d.cnki.gzkyu.2021.000855.[18]傅刚, 赵承, 李佳路. 大风沙过后的思考 [N/OL] . 北京青年报 , 2001-12-19 [2005-09-28]. <http://www.bjyouth.com.cn/Bqb/20000412/GB/4216%5ED0412B1401.htm>.
- [31].陈鹏. 基于池计算的宽度学习系统研究 [D]. 华东交通大学,2021.DOI:10.27147/d.cnki.ghdju.2020.000459.
- [32].褚菲,苏嘉铭,梁涛等.基于 lasso 和 elastic net 的宽度学习系统网络结构稀疏方法[J].控制理论与应用,2020,37(12):2543-2550.
- [33].Peng C, Lu R W, Kang O, et al. Batch process fault detection for multi-stage broad learning system[J]. Neural Networks, 2020, 129: 298-312.
- [34].Wang X, Wang C, Zhu K, et al. A Mechanical Equipment Fault Diagnosis Model Based on TSK Fuzzy Broad Learning System[J]. Symmetry, 2022, 15(1): 83.
- [35].Pu X, Li C. Online semisupervised broad learning system for industrial fault diagnosis[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 6644-6654.
- [36].陈光林,于丽娅,张成龙等.基于麻雀算法优化宽度学习系统的轴承故障诊断[J].工业工

- 程,2023,26(03):151-158.
- [37].姜午恺,周平.非平衡数据下基于自适应迁移与宽度学习的多工况故障诊断[J].中国科学:信息科学,2023,53(05):946-971.
- [38].Yang Y, Gao Z, Li Y, et al. A complex network-based broad learning system for detecting driver fatigue from EEG signals[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 51(9): 5800-5808.
- [39].Wang X, Zhu Y, Han S, et al. Fast and progressive misbehavior detection in internet of vehicles based on broad learning and incremental learning systems[J]. IEEE Internet of Things Journal, 2021, 9(6): 4788-4798.
- [40].Yuan X, Chen J, Zhang N, et al. A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles[J]. China Communications, 2021, 18(7): 117-133.125342.
- [41].付建广,尤斌,林毅等.基于粒子群优化与宽度学习系统的地铁客流预测模型[J].城市轨道交通研究,2023,26(05):23-26+33.DOI:10.16037/j.1007-869x.2023.05.005.
- [42].Xu X, Hu S, Shi P, et al. Natural phase space reconstruction-based broad learning system for short-term wind speed prediction: Case studies of an offshore wind farm[J]. Energy, 2023, 262:
- [43].Wang S, Yuen K V, Yang X, et al. Tropical Cyclogenesis Detection from Remotely Sensed Sea Surface Winds Using Graphical and Statistical Features-based Broad Learning System[J]. IEEE Transactions on Geoscience and Remote Sensing, 2023.
- [44].Li Z, Liu F, Yang W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021.
- [45].Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern recognition, 2018, 77: 354-377.
- [46].余潇智,杨静,李兴超等.基于卷积神经网络的目标检测算法综述[J].电子技术与软件工程,2022(24):171-176.
- [47].韩向阳.基于卷积神经网络的中文语音识别人机交互系统设计[J].自动化与仪器仪表,2023(07):201-204+209.DOI:10.14016/j.cnki.1001-9227.2023.07.201.
- [48].Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection[J]. Advances in neural information processing systems, 2013, 26.

- [49].Dhillon A, Verma G K. Convolutional neural network: a review of models, methodologies and applications to object detection[J]. Progress in Artificial Intelligence, 2020, 9(2): 85-112.
- [50].Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural networks[J]. Chemometrics and intelligent laboratory systems, 1997, 39(1): 43-62.
- [51].Eldan R, Shamir O. The power of depth for feedforward neural networks[C]//Conference on learning theory. PMLR, 2016: 907-940.
- [52].Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

在学研究成果

一、发表论文、出版专著

[1] Peng X, Zhao N, Xu L, et al. Vehicle Classification System with Mobile Edge Computing Based on Broad Learning[C]//2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2022: 1611-1617.

[2] 彭曦霆,赵乃贤,张笑宇,杨海波,徐乐西 基于 OSNet 的可疑人员跟踪定位方法: 申请号: 202310596858.8, 2023-05-25 (发明专利已受理)

二、科研项目

致谢

时光荏苒，转眼间我即将完成我为期三年的硕士研究生学习生涯，在即将毕业之际我回首看望这段旅程，心中感慨万千。在此我向三年硕士研究生求学过程中给予我帮助和支持的人们表达最真挚的感谢。

首先我要感谢我的二位导师，彭曦霆副教授和张笑宇老师。感谢二位老师把我招进课题组。在整个研究生生涯中对我的耐心指导以及辛苦付出。您们渊博的学识以及严谨的科研态度一直是我学习的榜样。是您们的悉心教导让我从一个懵懵懂懂的新生逐渐成熟，找到学习、奋斗的目标。我也学到了很多对我今后工作、生活很有帮助的好习惯。在每一个阶段您们都耐心的解答我的问题，在科研上为我指明方向，提供很多宝贵的建议。

除此之外，我还要感谢董冕雄教授、白石副教授、徐乐西老师在科研上对我的帮助，在您们的帮助下，我才得以不断提升。

在此，我要向我的家人表示深深的谢意。感谢你们一直以来对我无条件的支持和关爱，你们是最坚强的后盾，是你们的辛苦付出才能让我专心完成学业，你们的支持是我不断前行的动力。

感谢我的同窗好友，是你们陪我走过这段丰富多彩的研究生时光，我们互相学习，共同成长，一起面对困难。

最后我要感谢所有教导过我的老师，以及在我求学过程中遇到的每一位朋友们。再次向大家表示感谢，愿我们在今后的人生路中继续追寻属于自己的梦想。