# 面向智能驾驶的多传感器感知融合

# 摘要

# 绪论

## 课题的研究背景和意义

在经济和生活的持续发展之下，汽车已经成为人们生活中不可或缺的一部分，并且其数量也在持续增长。据统计，直至2022年底，我们国家的汽车保有量高达3.021亿辆，汽车驾驶人高达4.81亿，图 1.1是我国近五年来机动车与驾驶人数的变化情况，不难发现，机动车保有量和机动车驾驶人数都在持续稳定增长中，相比之下，我国的城市的道路虽然也在持续建设当中，但是其速度明显无法与汽车增长速度相比[1]，因此，在有限的道路资源下，车辆的驾驶策略与道路的交通状况息息相关。

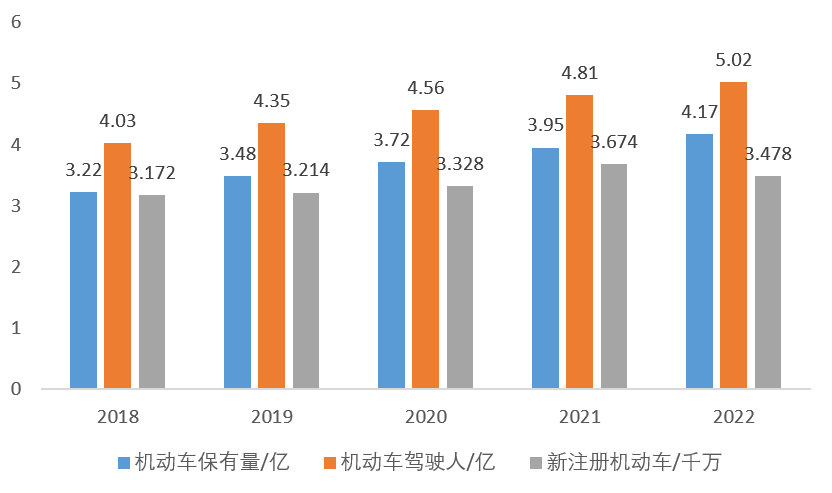


图 1.1 近五年机动车驾驶人数变化

Fig. . Changes in the number of motor vehicles and drivers in the past five years

智能辅助驾驶系统（Advanced Driver Assistance System, ADAS）是一种将人工智能、机器学习、计算机视觉等前沿技术应用于汽车驾驶，用于提高驾驶安全和效率的一类系统。ADAS主要通过检测和分析驾驶员、车辆、环境等信息，来提醒驾驶员注意危险情况和给出安全建议。它以人为主体，通过对环境以及路况的多维感知与预测[2]，然后进行决策和控制，拓展了驾驶员的视野，并提高了驾驶员对环境的感知力，对驾驶员有着重要的辅助作用，在一定程度上减少了因驾驶员感知异常或决策失误而造成的交通事故。随着技术的不断发展，ADAS的研究正在得到越来越多的关注。关于ADAS的研究背景可以追溯到20世纪70年代，当时已经开始研究自动驾驶技术。随着计算机和传感器技术的进步，ADAS在汽车领域的研究和应用也得到了长足发展。许多公司和研究机构都在对ADAS进行研究，并在实际应用中不断改进和完善这类系统。

ADAS的研究具有重要的意义。首先，它可以有效降低驾驶员的疲劳和压力，提高驾驶员的注意力和反应能力，降低驾驶员的错误率和事故风险。其次，ADAS可以有效提高道路交通环境的安全性，减少交通事故的发生，降低交通事故对社会和经济的影响。最后，ADAS的研究也为人工智能和机器学习技术的发展提供了有力的推动力，为未来智能汽车的发展铺平了道路。

然而，ADAS也存在一些挑战和限制。首先，ADAS技术仍然处于不断发展和完善的阶段，存在许多未解决的技术问题。其次，ADAS系统的成本仍然较高，对一般消费者来说仍然不够实用。最后，ADAS系统需要高度可靠的传感器和算法，以保证它们在驾驶过程中的精确性和可靠性。

尽管如此，ADAS研究仍具有广阔的前景。未来，ADAS技术将不断提高，成本将不断降低，并且ADAS系统将被广泛应用于智能汽车和普通汽车中。此外，ADAS技术还将与其他技术，如自动驾驶技术、物联网技术等相结合，形成更加完善和先进的智能交通系统。总之，智能辅助驾驶系统（ADAS）是一项具有重要意义的研究课题，它不仅有助于提高道路交通安全，而且为未来智能汽车技术的发展铺平了道路。因此，未来的研究应该继续探索ADAS的技术难点和挑战，努力实现ADAS技术的更大提高和更广泛的应用。

智能感知是ADAS的重要基础，智能辅助驾驶系统的一切决策和控制都是建立在感知的基础之上，智能感知通过多个传感器对道路信息以及环境状况进行采集，经过过滤和处理后反馈给驾驶员和系统。常用的感知传感器包括摄像头、毫米波雷达、激光雷达、GPS、IMU（Inertial Measurement Unit）等。摄像头作为视觉传感器，可以通过图像处理技术，感知行人，车辆，车道线，障碍物，交通信号灯等，然而摄像头的感知准确度直接受到光照条件的影响，在夜晚或者太阳光照强度较大的条件下，摄像头的感知信息可能会存在异常，激光雷达使用激光束对当前环境进行扫描，用以探测环境中的各种目标[3]，同样地，用激光雷达采集到的点云也存在噪声，通常需要先进行数据预处理，才能获取到需要的目标信息。

不同种类的传感器因为其工作原理，不可避免地会因为环境的干扰或其他因素而出现测量误差，因此，采用多种传感器协同工作，并将它们各自的测量结果进行数据融合分析，可以大大降低测量结果误差，确保感知的准确度。

## 国内外研究现状

针对多传感器模式单一，局限性固定的特点，采用多传感器数据感知融合的方式为汽车提供环境信息是一种比较好的尝试方式，对此，本小节将分别从智能驾驶、智能感知以及数据融合三个方向分析近年来的国内外研究现状。

### 智能驾驶研究现状

当前国内外对于ADAS的研究都处于高速发展阶段，中国对于ADAS的研究起步较晚，因此早期新车的ADAS渗透率处于较低水平，具有较大的市场空间，然而在各国家和地区的相关政策影响下，ADAS获得了充分的发展空间，对应机动车的ADAS方案也在持续提出和更新。Abdallah Moujahid等人在2018年对于机器学习与智能辅助驾驶系统分别进行独立分析，阐述了一些机器学习技术与ADAS组件的匹配程度及原因，以一些有监督学习、无监督学习、深度学习和强化学习作品作为示例，基于他们各自的优势与缺陷进一步讨论其进步空间[4]，这种分析与回顾对后续智能驾驶的深入研究有一定的稳固意义。Jordan Tunnell等人在2018年提出了一种通过ADAS提高车辆燃油经济性的方法，文章使用2010年的丰田普锐斯作为模型，结合ADAS传感器、预测模型和动态规划最佳能量管理控制来优化车辆燃油经济性，经过验证对比，评估四种车辆控制策略对于汽车燃油经济性的影响，结果表明定义的ADAS范围和算法提供了接近与ADAS地面实况的相关性，可以作为基于预测的一部分实现汽车燃油经济性的改进[5]。Ibrar Yaqoob等人在2019年通过设计分类法调查、强调和报告了自动驾驶领域的主要研究进展，列举并讨论了成功部署自动驾驶汽车的一些必不可少的要素，发现并展示了最近关于自动驾驶的协同效应和突出案例研究，最后，确定并讨论了几个势在必行的开放研究挑战，作为未来的研究方向[6]。蔡自兴的研究团队在2019年总结了团队关于智能驾驶领域主要成果和技术，涵盖了感知、规划、控制、导航、通信等多个方面，该团队的自主驾驶技术在我国完成了全程286km的自主驾驶实验，并在期间自主完成超车、被超车、人工干预、汇入车流等功能[7]。杨贵栋等人在2019年讨论了汽车在智能驾驶环境下，乘客的舒适度测评方式，以乘客的主观感受作为主观评价标准，结合车辆性能与驾乘舒适度模型作为客观评价标准，此外还以人因工学、数学模型与机器学习多个角度评价智能驾驶汽车的舒适度[8]，这种多维度的评价方式可以作为汽车驾乘舒适度的一个可靠参考标准。陈锦栋在2020年分析了智能驾驶当中的技术体系与关键技术，主要从车辆的自我感知与规划下的自主式与基于车联网的互联式技术路线出发，包括高性能AI芯片、车用通信、信息安全、车联网标准、传感器等关键技术，并针对当前智能驾驶现状，给出了降低成本、齐头并进、追求稳定的发展建议[9]。David Rojas-Rueda等人在2020年讨论了智能驾驶汽车对于城市公共健康的影响，部分自动驾驶汽车已经开始在美国投入使用，为人们提供各种便利的自动化服务，与之而来的，是使用自动驾驶汽车而产生的一些风险，如空气污染，噪音和汽车的长时间停留，同样地，也不能忽视自动驾驶汽车所带来的各种便利以及事故发生率的降低，因此，自动驾驶汽车的投入使用需要与合适的监管政策相匹配[10]。

### 智能感知研究现状

智能感知是一种新兴的技术，它可以帮助机器获取环境信息，从而实现智能感知。近年来，智能感知技术受到了广泛的关注，并且取得了显著的进展。早在2018年，基于当时现有的车道线识别算法，黄窈蕙提出了一种新的汽车视觉图像处理方法，将图像预处理后，使用透视变换方法将平面图像转换至世界坐标，最后使用Canny边缘检测和Hough变换识别车道线[11]。这种单目车道线检测算法对于传感器的要求不高，但是其识别准确率容易受到图像中车道线的角度影响。张晶晶等人提出了汽车在驾驶场景下动态跟踪目标的图像感知算法，将图像序列预处理后用Harris角点检测方法提取目标角点，继而对目标进行跟踪预测[12]，算法跟踪准确率高，实时性较强，但由于图像的局限性，目标在被遮挡时可能会丢失跟踪并且不容易恢复跟踪。Muhammad Qasim Khan等人在2019年回顾与讨论了关于驾驶员的眼神注意力感知在ADAS中的应用，针对不同类别的注视跟踪技术，分别讨论他们的优点和局限性，从而进一步讨论其各自的实际用途以及在智能辅助驾驶系统中的应用，从而减少因为驾驶员视觉疲劳而造成的交通事故[13]。李银国等人在2019年提出了一种基于双目视觉的大尺度三维场景重建方法[14]，首先通过优化立体匹配策略来提高立体匹配效率,再提出均匀保距的特征点提取算法RSD,减少三维点云计算与三角剖分耗时,提高大尺度智能驾驶场景重建的实时性，这种算法具有良好的大尺度三维场景重建效果。陈晓冬等人在《光电工程》中以激光雷达扫描方式及相关技术为切入点对智能驾驶车载激光雷达硬件关键技术进行了介绍,分别讨论了机械式、混合式和全固态车载激光雷达的原理、特点及现状;以智能驾驶应用任务为导向，对点云分割、目标跟踪与识别、即时定位与地图重建这三类车载激光雷达应用算法进行了归纳总结[15]，车载激光雷达也进一步走向固态化，智能化和网络化。宋绍京等人在2020年对适用于汽车智能驾驶的多光谱激光雷达的波段选择进行了可行性研究,利用主成分分析法对智能驾驶中典型目标进行光谱计算及分析,结合激光光源特性以及光电探测器的特性,综合多光谱激光雷达波段选择方法和智能驾驶应用场景中典型目标地物光谱特性,以及商用激光雷达的可获得性,得出了适用汽车智能驾驶的多光谱激光雷达的波长范围，提升了单一波长激光雷达在物性探测分类和状态上的性能[16]。李一鸣等人在2022年提出了一套用车辆与邻近环境中其他物体能够协作通信(V2X，Vehicle-to-everything)的数据集V2X-Sim，用于 V2X 辅助自动驾驶的综合模拟多代理感知，包括来自路边单元和多辆车的传感器记录、多模态传感器流以及多样的路面条件数据，同时建立了一个开源测试平台，并为最先进的协作感知算法提供了三个任务的基准，包括检测、跟踪和分割，促进了自动驾驶的协作感知研究[17]。

### 点云感知

PartA2点云目标检测算法的背景可以追溯到2013年，当时深度学习技术在计算机视觉领域开始兴起，图像目标检测算法基于深度学习技术也开始取得重大突破。然而，点云数据在自动驾驶和机器人领域中具有重要的应用价值，而且点云数据和图像数据有很大的区别，因此需要针对点云数据进行目标检测算法的研究。

最初，研究人员主要使用传统的计算机视觉方法，例如基于滤波和聚类的方法，来处理点云数据。然而，这些方法的性能受到点云数据的稠密程度、噪声和遮挡等因素的影响，且很难处理点云数据中的局部信息和全局信息。

随着深度学习技术的不断发展，研究人员开始探索使用深度学习方法来处理点云数据，例如使用卷积神经网络（CNN）进行点云特征提取和点云分割等任务。然而，由于点云数据的非结构性，传统的CNN无法直接应用于点云数据。因此，研究人员提出了一些基于三维卷积神经网络（3D-CNN）的点云目标检测算法，例如VoxelNet、SECOND和PointPillars等算法。

### 传感器融合研究现状

传感器融合技术将多种传感器的信息结合在一起，以提高系统的性能和准确性。近年来，传感器融合技术在许多领域得到了广泛的应用，如机器人、自动驾驶、室内定位、智能家居等，其相关的研究也越来越深入。

对于多激光雷达的融合检测方法，Muhammad Sualeh等人在2019年提出了一种基于多雷达融合的目标检测和追踪算法(MODT)，考虑到车载嵌入式计算环境的局限性，将三个激光雷达的数据合并后采用高效的 MODT 框架处理，此外，开发了基于 3D 网格的聚类技术来检测高架结构下的物体。使用交互式多模型-无迹卡尔曼滤波器-联合概率数据关联滤波器 (IMM-UKF-JPDAF) 跟踪从对象检测获得的质心测量值[18]。Elif Aksu Taşdelen等人在激光雷达上开发、应用和测试了两种不同的实时传感器融合方法，首先基于多个激光雷达首次提出并实现了高级track-to-track融合算法，此外，分析和比较了track-to-track融合算法与经过充分研究的低级实时融合算法的性能[19]。

对于激光雷达与毫米波雷达的融合，Atsushi Kanno等人演示了通过光纤与远程信号合成器连接的手持式毫米波雷达头，用于实现小型手持式无损成像系统。安装在雷达头上的惯性测量单元 (IMU) 可识别雷达头的方向和方位，为建筑结构检测提供三维点云。还讨论了带有 IMU 的小型激光雷达系统，用于构建建筑物内部结构映射的 3D 点云[20]。党相卫等人在2021年针对智能驾驶中准确鲁棒的感知问题，提出了一种融合毫米波雷达和激光雷达鲁棒的感知算法。使用基于特征的两步配准的空间校正新方法，实现了三维激光点云和二维毫米波雷达点云精确的空间同步；使用改进的毫米波雷达滤波算法,减少了噪声、多径等对毫米波雷达点云的影响，然后对两种传感器的数据进行融合，得到准确鲁棒的感知结果，解决了烟雾对激光性能影响的问题[21]。

对于多相机的融合检测方法，熊辉等人在2020年提出了一种在复杂场景中使用多摄像头融合进行快速车道线检测的方法，集成消失点估计和指定特征拟合策略，实验结果表明，与传统方法相比，多摄像头融合框架有助于提高准确性和鲁棒性[22]。

对于相机与雷达的感知融合，Uttam K. Majumder等人在文献[23]中使用深度学习在合成孔径雷达图像分别进行单目标检测和多目标检测，为摄像头与雷达的传感器融合提供了一定的思路[23]。王章靖等人在2021年提出了一种基于毫米波雷达和相机融合的的鲁棒性目标检测和分类算法，利用融合数据采集平台建立了雷达与摄像头融合数据集，包括白天和夜间的十字路口、高速公路、马路和学校操场，将所提出的融合网络模型与传统的雷达信号算法和Faster R-CNN相比较，其精度比传统雷达算法高于80%，比Faster-R-CNN高于30%[24]。程健康在2022年提出了基于车规级高性能SoC设计了面向汽车后装市场硬件系统，并在此基础上开发了多传感融合的防碰撞预警软件系统，首先，使用成熟的自动驾驶数据集KITTI，结合自采集部分的样本训练深度学习SSD目标检测网络模型，并将模型部署于车规级SoC TDA2S平台，测试表明，系统的动态性能和静态性能准确率都高于90%[25]。

### 图像目标检测研究现状

目标检测算法作为一种计算机视觉技术，旨在通过通过计算机识别和定位特定目标。它已经成为多种应用中不可或缺的技术，包括自动驾驶汽车、安全监控和医学图像处理等。在人工智能领域，目标检测算法一直作为一种热门话题飞速发展，其检测性能也在不断优化。

以最早的卷积神经网络（CNN）为基础，各种目标检测算法百花齐放，主要被分为两种类型，单阶段目标检测算法和双阶段目标检测算法，其主要区别在于双阶段目标检测算法相比单阶段目标检测算法要多出一个生成候选区域阶段，以更久的检测时间为代价，提高检测的精度。

1. 双阶段目标检测

双阶段目标检测算法主要以R-CNN系列为主要代表，首先提出将卷积神经网络用于特征提取[26]，此外提出使用pre-training来完成一些样本量比较少的检测任务。Fast R-CNN基于R-CNN和SPP-net[27]进一步优化，在完成特征的提取之后，使用softmax进行预测，使得分类和回归可以同时进行，此外，利用ROI pooling layer，使得模型支持任意尺寸图像的训练和预测[28]。Faster R-CNN基于Fast R-CNN提出RPN（Region Proposal Network）网络，并将RPN网络生成的区域候选框与原有特征图共同进入池化层，形成了完整的端到端卷积神经网络目标检测模型，并且支持在显卡上运行[29]。

## 论文的主要工作和组织结构

### 论文的主要工作

本文主要面向智能驾驶领域，从传感器的感知与融合角度出发，数据采集点。

### 论文的组织结构

# 传感器数据采集与预处理

## 传感器工作原理与线控底盘

### 激光雷达

激光雷达是一种利用激光束进行测距和建立物体三维模型的设备。它的工作原理是通过发射短脉冲激光，将激光束照射到目标物体上，然后接收被物体反射回来的激光信号，并利用接收到的信号计算目标物体与激光雷达之间的距离和位置信息。

激光雷达发射的激光脉冲很短，通常只有几纳秒至几十纳秒的时间，它们的波长一般在近红外或激光波段。激光束会照射到目标物体上并被反射回来，接收器会接收到这些反射的激光信号并测量它们的时间差，从而计算出目标物体与激光雷达之间的距离。

通过多次发送和接收激光脉冲，激光雷达可以获取目标物体的三维位置信息。通常，激光雷达会以一定的角度扫描整个场景，从而获取完整的三维模型。

需要注意的是，激光雷达的精度和测距范围与其硬件参数有关，例如激光功率、激光束的角度和频率、接收器的灵敏度等等。因此，在实际使用中，需要根据不同的应用场景选择不同类型和参数的激光雷达。

为了完成对车辆周围环境的感知，本文选用镭神智能C16激光雷达作为传感器，如所示，作为一种用于智能驾驶的传感器产品，镭神智能C16采用激光雷达技术，通过向传感器四周发射激光束，然后测量激光束的返回时间和强度，从而感知周围环境。C16一共配备了16个激光雷达传感器，能够实现全方位的环境感知，其中14个传感器安装在设备的圆形底部，另外两个传感器则分别安装在设备的顶部和底部，以实现对垂直方向的感知。

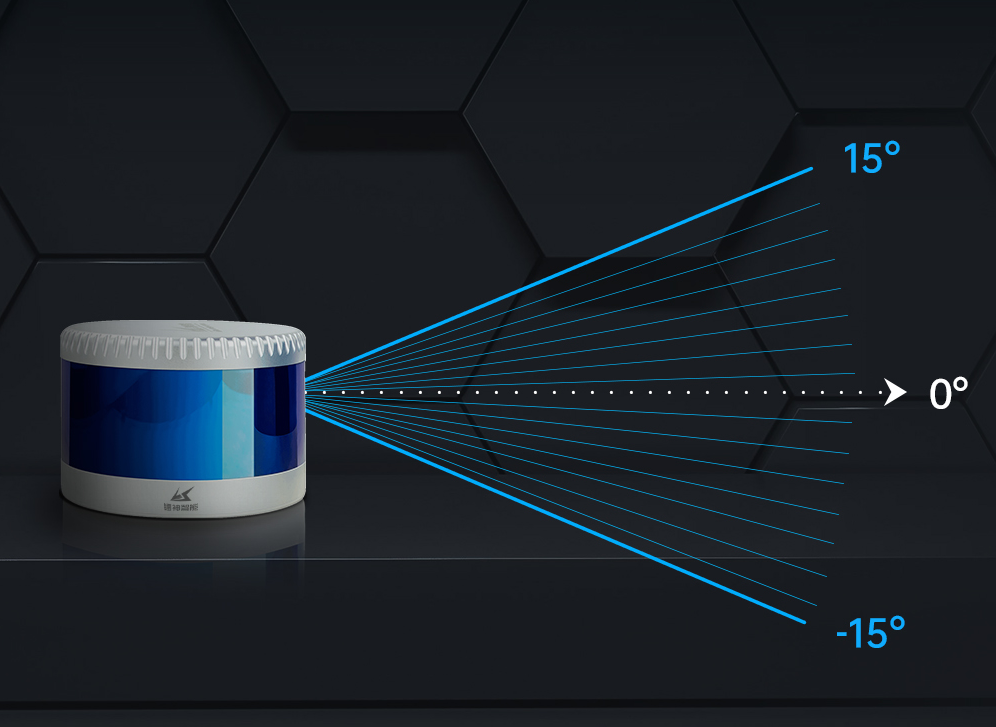


图 2.1 激光雷达产品示意图

Fig. . C16 product appearance

C16感知周围环境之后，它会将感知到的数据以点云的形式输出。这些数据可以提供给车辆控制系统，帮助车辆识别周围的障碍物、道路状况等，并做出相应的行驶决策，从而实现自主控制，是关于镭神C16的产品信息。

表 2.1 镭神C16产品信息

Tab. . C16 product information

|  |  |
| --- | --- |
| 技术参数 | 参数值 |
| 产品名称 | 镭神智能C16激光雷达 |
| 线束 | 16 |
| 波段 | 905nm |
| 测距原理 | TOF（飞行时间法） |
| 探测距离 | 70/120/150/200m |
| 测点速率 | 320000pts/s（单回波） |
| 测距精度 | ±3cm |
| 水平角度分辨率 | 0.09°@5Hz，0.18°@10Hz，0.36°@20Hz |
| 垂直角度分辨率 | 2° |
| 冲击 | 500m/sec2，持续11ms |
| 振动 | 5Hz-2000Hz，3G rms |
| 工作温度 | -20°C~60°C |
| IP等级 | IP67 |
| 外观尺寸 | 直径145mm，高度91mm |
| 重量 | 1.3kg |

### 摄像头

摄像头传感器是一种使用光学传感器技术进行数据采集的设备。它可以通过镜头收集光线，并将其转换为数字信号，然后通过计算机算法处理这些信号。在自动驾驶车辆中，摄像头传感器被广泛应用于感知道路上的障碍物、车辆和行人等。

以摄像头作为视觉传感器的优点之一是它可以提供高分辨率的图像，这对于自动驾驶系统来说至关重要。高分辨率的图像可以提供更多的细节和信息，帮助自动驾驶车辆更准确地感知周围环境。此外，摄像头传感器还具有低延迟的高帧率的特点、这使得自动驾驶车辆可以实时感知和响应道路上的变化。

本文选用USB外置摄像头作为智能感知视觉传感器，摄像头具体外观如图 2.2所示，该摄像头支持以1920\*1080分辨率、30FPS的配置录制视频，其拍摄视场角为90度，物理尺寸为100mm\*50mm\*25mm，其参数配置如表 2.1所示。



图 2.2 USB摄像头

Fig. . USB camera

表 2.2 摄像头产品信息

Tab. . Camera information

|  |  |  |
| --- | --- | --- |
| 名称 | 值 | 描述 |
| 分辨率 | 1920\*1080 | 以像素表示 |
| 帧率 | 30帧/秒 | 每秒钟传输的图像帧数 |
| 传感器类型 | CMOS |  |
| 镜头类型 | 定焦镜头 |  |
| 对焦方式 | 自动 |  |
| 焦距 | 3.6mm | 单位为mm |
| 光圈 | F2.0 | 镜头开口大小 |
| 视场角 | 90度 | 摄像头可以拍摄的水平和垂直角度 |
| 曝光补偿 | 0.0 | -2.0到2.0 |
| 自动白平衡 | 否 | 自动调整白平衡以适应不同光线 |
| 自动曝光 | 是 | 自动调整曝光时间以适应不同光线 |
| 传输接口 | USB2.0 |  |
| 操作系统兼容性 | Windows，Ubuntu |  |
| 麦克风 | 是 | 是否内置麦克风 |
| 摄像头尺寸 | 100\*50\*25 | 摄像头物理尺寸，以毫米为单位 |
| 重量 | 60g |  |
| 附带软件 | 驱动程序 |  |

目前大多数智能驾驶感知系统都配置多个摄像头进行大角度视觉感知，本文在有限的条件下选用单一摄像头作为视觉传感器，此外，考虑到摄像头在某些情况下可能会受到限制，例如，在恶劣的天气条件下（如雨雪天气），摄像头可能无法正常工作，从而影响车辆的环境感知能力。此外，摄像头还可能受到阴影、反光和遮挡等因素的影响，从而影响其检测和跟踪能力。因此，为了提高摄像头的可靠性和准确性，智能驾驶感知系统通常会采用多种传感器进行辅助，并将它们的数据进行融合，以获得更全面和准确的环境感知信息。在这种情况下，本文选用激光雷达作为协同感知传感器。

### FELIS™线控底盘

线控底盘作为一种能够通过远程控制或预先编程的方式实现自主行驶的底盘车辆。与传统底盘车不同，线控底盘不需要驾驶员操纵方向盘、踏板等控制车辆行驶，而是通过先进的计算机和传感器系统来感知和分析道路环境，自主决策行驶路线和速度，并控制车辆的转向、刹车、加速等动作。在智能驾驶领域，线控底盘车辆被广泛应用于自动驾驶技术的研究和测试，帮助开发者更好地理解自动驾驶系统的工作原理、优化自动驾驶算法，从而提高自动驾驶的安全性和稳定性。

本文采用北京天隼公司FELIS AHC201自动驾驶底盘车作为测试用底盘车辆，它是一款基于商用车技术架构研制的智能驾驶测试地盘，具有小型化、智能化和重载型的特点，采用天隼自有知识产权的线控零部件为基础，配备前双横臂独立悬架和后整体桥非独立悬架，以及后轮碟刹型式的电子液压行车制动和后桥电磁刹型式的驻车制动，如图 2.2所示。



图 2.3 FELIS AHC201

Fig. . FELIS AHC201

FELIS™底盘架构与乘用车基础架构一致，采用CAN总线通信，并由VCU对转向、驱动以及制动系进行指令解析和转发，实现整车控制总线化。此外，FELIS线控底盘搭载了800W直流无刷电机和整体差速后桥，能够在200kg标准载重下达到15km/h的最高车速，并通过48V/40Ah的磷酸铁锂动力电池保证标准载重下的8小时续航时间。表 2.2为本文测试用线控地盘技术指标。

表 2.3 FELIS AHC201产品参数

Tab. . FELIS AHC201 product parameters

|  |  |
| --- | --- |
| 指标 | 数值 |
| 外形 | 1080\*650\*255mm |
| 轮距 | 575mm |
| 轴距 | 813mm |
| 自重 | 80kg |
| 标准载重 | 200kg |
| 最高速度 | 15km/h |
| 转弯半径 | 1800mm |
| 离地间隙 | 空载83mm/满载60mm |
| 爬坡率 | 20% |
| 驱动形式 | 后整体桥差速驱动 |
| 转向形式 | 前阿克曼转向 |
| 行车制动 | 后轮液压碟刹制动 |
| 驻车制动 | 后桥电磁刹驻车 |
| 悬架系统 | 前双叉臂独立悬架、后整体桥非独立悬架 |
| 无线控制 | 遥控器 GFSK@500KHz/数字跳频 |
| 总线控制 | CAN总线 |
| 接口规格 | CAN总线/WS20-7航插x1、DC12V@30A/WS28-2航插x1 |
| 通信协议 | SAE J1939/定制 |

作为智能驾驶测试地盘，FELIS™线控底盘为研发人员提供了一个实验平台，可以用来测试和验证各种智能驾驶技术，以实现更加安全、智能和高效的智能驾驶系统，通过线控底盘车辆采集传感器数据，可以实现车辆驾驶过程中，各种物理状态的复现，包括但不限于车辆的加减速，路面抖动等，对应地，可以使传感器采集到相应状态下的感知数据，扩大其覆盖范围。

## 传感器数据采集与预处理

### 传感器相关部署

为了完成传感器数据的采集工作，首先需要将对应的传感器部署到线控底盘FELIS AHC201上，其对应的传感器安装位置如图所示

鉴于激光雷达检测原理，为了得到视野开阔的点云数据，需要确保传感器周围没有干扰影响，本文采用合金支架将激光雷达固定于FELIS AHC201顶部。同理，为了保持摄像头视野开阔，减少干扰，需确保摄像头前方无遮挡物存在。

考虑到系统的可扩展性与多个传感器之间数据通信的复杂性，计算机通过交换机与传感器和车辆底盘控制系统建立统一连接，其数据通信结构如图所示，

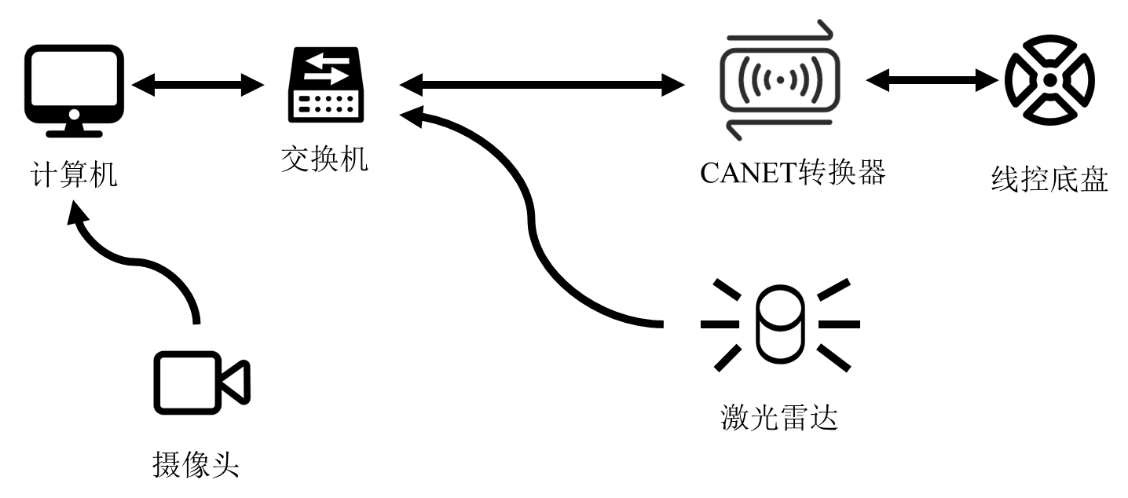


图 2.4 系统数据通信结构

Fig. . System data communication structure

线控底盘控制系统采用CAN总线通信方式，经过CANET转换器转换成TCP/IP协议后与激光雷达一起连接至交换机，从而与计算机建立通信，实现智能驾驶的感知与控制。鉴于现有硬件局限性和所选传感器设备闲置，选用的摄像头只能通过USB串口与计算机进行通讯。

表 2.4 数据通信参数

Tab. . Data communication parameters

|  |  |  |  |
| --- | --- | --- | --- |
| 设备 | 接口地址 | 数据读取方式 | 数据类型 |
| 线控底盘 | 192.168.5.137 | Socket通信 | TCP报文 |
| 激光雷达 | 192.168.5.7. | ROS topic | PointCloud2 |
| 摄像头 | USB串口 | Opencv | Img |

表 2.1是传感器数据格式与通信地址，关于激光雷达数据的采集，在系统层面主要依赖ROS系统，采用TCP通信机制进行数据通信，在交换机配置下，其IP地址为192.168.5.7，确保计算机与激光雷达处于同一网关下，运行激光雷达厂商提供的驱动程序，程序将自动采集激光雷达发送的点云数据，并将点云数据以Topic的形式在ROS系统中发布；关于摄像头感知数据的采集，本文选用的usb网络摄像头不需要驱动程序，将摄像头连接至计算机usb接口后，系统将自动识别摄像头设备，开发者可自行读取端口并采集图像数据。

### 数据采集与预处理

完成传感器与线控底盘的相关部署后，将传感器数据采集系统开到重庆大学校内路面上，并开始相关数据的采集与录制。

1. 摄像头

opencv作为一种流行的计算机视觉库，它可以用来处理图像和视频，opencv提供了多种配置选项，可以在录制视频数据的时候自定义视频的分辨率、帧率、编解码器和其他设置，可在多个操作系统（如Windows、Linux和MacOS）上使用，此外，opencv可以与其他库和框架（如Python、C ++和Java）集成，这使得它可以方便地与现有的应用程序一起使用。因此关于摄像头的数据采集，本文采用opencv作为采集工具，在程序中分别设置视频的编码方式为MPEG-4编码，帧率为30fps，视频帧大小为640\*480，随后开始循环捕捉摄像头数据，将采集到的视频保存到本地，同时，将当前画面实时发送至ROS平台，自定义的摄像头录制程序启动后，将线控底盘车辆开至校内路段行驶一段时间后，终止程序，视频数据采集完成。摄像头发送至ROS的数据格式见表

1. 激光雷达

完成激光雷达的数据采集之前，首先确保激光雷达的硬件与计算机相连接并且软件都配置完成，在激光雷达的驱动程序中，首先创建ROS节点，然后初始化驱动，完成初始化后驱动程序将循环使用UDP获取雷达报文并将获得的点云数据发布到ROS上。

|  |  |  |
| --- | --- | --- |
| 参数 | 变量 | 描述 |
| FrameID | lslidar | 坐标ID |
| Topicname | /lslidar\_point\_cloud | 用于订阅点云数据 |
| Topictype | sensor\_msgs/PointCloud2 | 定义点云数据格式 |
| Frequency | 10Hz | 采样频率 |
| BandWidth | 10±MB/S | 带宽 |
| Distence Threshold | 200 | 采样距离 |

将图像数据以Image形式发布至ROS，同时，以PointCloud2格式发布点云数据，可保持图像与点云数据实时同步，采用rosbag指令录制视频与点云数据，参数指定对应Topic名称，\*\*为数据采集过程，数据采集道路位于重庆大学A区主教附近，采集过程。



## 本章小结

本文的主要研究对象是用于采集数据传感器，包括激光雷达和光学摄像头，对于激光雷达，通过厂商提供的驱动程序将点云数据通过ROS采集，对于摄像头，采用两种方式保存数据，首先通过opencv实时读取摄像头并将视频数据保存至本地，同时通过ROS将视频以Image形式发送至ROS，使其与点云消息保持时间同步，再通过rosbag共同录制消息。

# 基于YOLOv8的实时智能目标检测

## 单阶段实时目标检测模型

### YOLOv5网络

YOLOv5[30]按照不同的模型参数可以分为s、m、l、x，各个版本结构和原理类似，其主要区别在于模型大小，本文以YOLOv5s作为参考，其模型结构图见图 3.1。

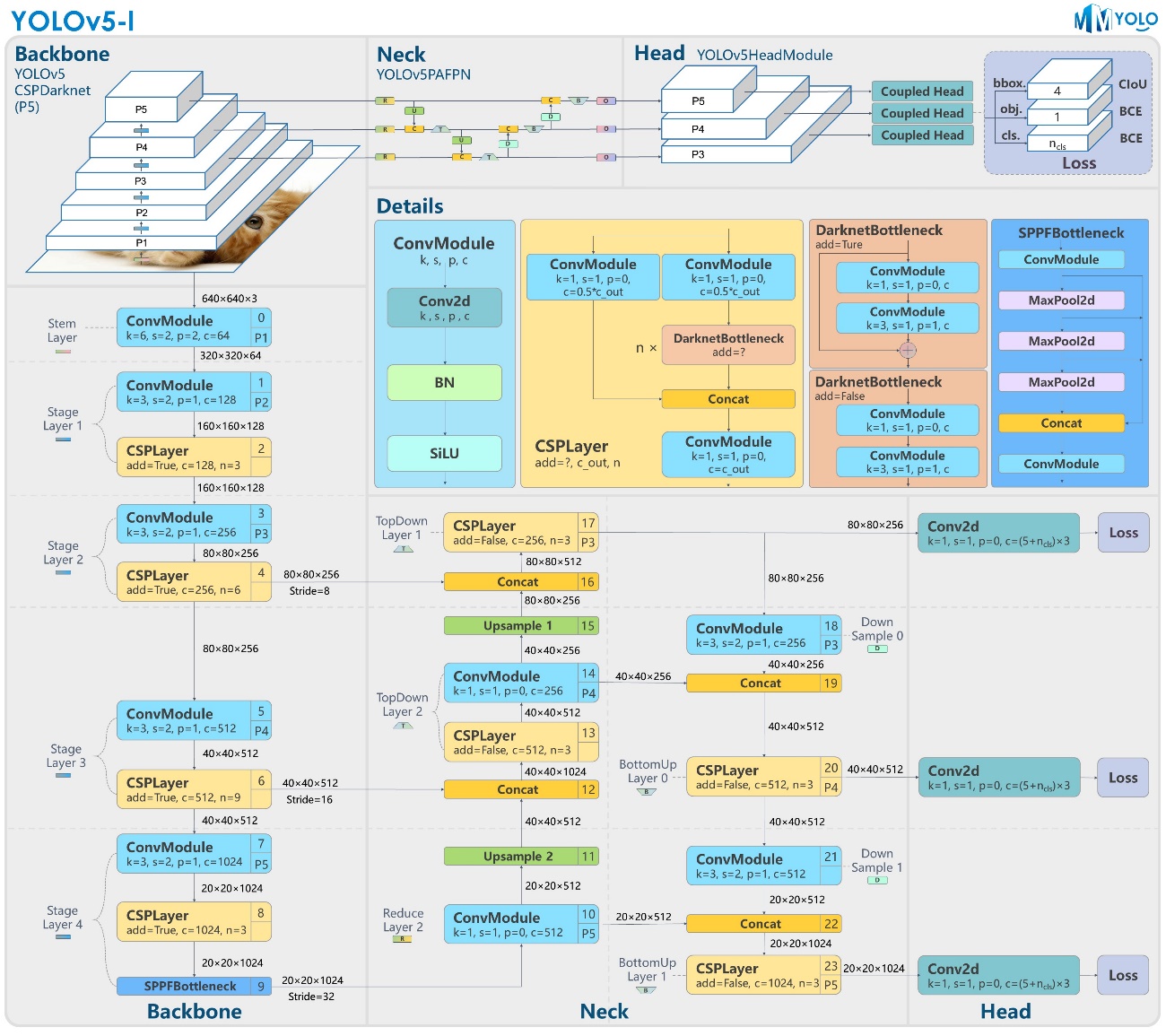


图 3.1 YOLOv5模型结构图

Fig. . YOLOv5 structure diagram

作为YOLO（You Only Look Once）系列算法性能较优的一代模型，YOLOv5的架构主要分为三个部分：骨干网络（Backbone）、特征增强网络（Neck）和检测头（Head）。通过骨干网络提取图像特征，通过特征增强网络增强特征表达能力，再通过检测头预测目标类别和位置，实现目标检测任务。

1. 骨干网络（Backbone）

YOLOv5 使用 CSPDarknet53 作为骨干网络。CSPDarknet53 是 Darknet53 的变体，使用了 Cross Stage Partial Networks（CSP）结构来加速计算和提高准确性。CSP 结构将输入特征图分成两个部分，其中一部分直接通过残差连接到输出，另一部分经过一系列卷积层后再连接到输出，这种结构能够在减少计算量的同时提高准确性。

1. 特征增强网络（Neck）

YOLOv5 使用 SPP（Spatial Pyramid Pooling）结构作为特征增强网络。SPP 层对不同尺度的特征图进行最大池化操作，以捕获不同尺度的物体信息。然后将所有池化结果拼接在一起，作为后续检测头网络的输入。这样可以增强特征表达能力，提高检测准确率。

1. 检测头（Head）

YOLOv5 的检测头结构与 YOLOv4 相似，由多个卷积层、上采样层和预测层组成。其中，预测层是关键部分，它将网络输出的特征值转换为目标类别概率和边界框位置。YOLOv5 采用了一个三层的预测层，每一层都有不同的输出特征图大小和预测比例。具体来说，第一层预测较大的目标，输出特征图的大小是原图的 1/32，预测比例是 2；第二层预测中等大小的目标，输出特征图的大小是原图的 1/16，预测比例是 1；第三层预测较小的目标，输出特征图的大小是原图的 1/8，预测比例是 0.5。预测层使用 sigmoid 函数将输出的特征值转换为目标类别概率和边界框位置。

表 3.1为YOLOv5模型在不同数据集上的性能表现。

表 3.1 YOLOv5性能

Tab. . YOLOv5 performance

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 模型 | mAP@0.5 | FPS |
| COCO | YOLOv5s | 41.2 | 140 |
| COCO | YOLOv5m | 46.0 | 87 |
| COCO | YOLOv5l | 50.3 | 62 |
| COCO | YOLOv5x | 53.4 | 27 |
| PASCAL VOC | YOLOv5s | 85.1 | 280 |
| PASCAL VOC | YOLOv5m | 87.5 | 190 |
| PASCAL VOC | YOLOv5l | 88.6 | 134 |
| PASCAL VOC | YOLOv5x | 89.5 | 68 |

虽然YOLOv5使用了FPN来提高对小物体的检测能力，但在一些场景下，仍然存在漏检的情况。这是因为目标尺寸太小，导致其在高层特征图中难以被准确地定位和检测。因此，需要进一步提高算法对小目标的检测和识别能力。并且，在复杂场景下，目标可能被其他物体遮挡，导致算法无法准确检测。因此，需要采用更加鲁棒的目标检测算法，以处理这种复杂情况。此外，目标检测算法需要处理不同尺度的目标，但在YOLOv5中，仍存在对不同尺度目标的不平等处理问题。因此，需要改进算法对不同尺度目标的处理方式。

### 改进后的YOLOv8

为了进一步提高目标检测的准确性和鲁棒性，继YOLOv5之后，Ultralytics在2023年1月10日发布了YOLOv8[31]，作为目标检测领域最新的一个SOTA模型，在YOLOv5的基础上进行了改进，引入了注意力机制和特征金字塔网络等技术来解决YOLOv5存在的部分问题。其中，注意力机制可以使模型更加关注重要的特征，提高目标检测的准确性；特征金字塔网络可以在不同尺度上提取特征，增强算法对不同尺度目标的处理能力。因此，引入YOLOv8可以进一步提高目标检测的性能，为智能驾驶感知算法提供更加准确和鲁棒的目标检测支持。

在之前版本的YOLO基础之上，YOLOv8提供了更高的准确性以及更快的检测速度。图 3.1为按照YOLOv8官方提供的源码绘制的模型结构图，其主要网络架构主要包括Backbone，Neck，Head三部分，由于YOLOv8是YOLOv5之后的下一个重大更新，因此在引入YOLOv8时，可以将之与YOLOv5做参照对比。

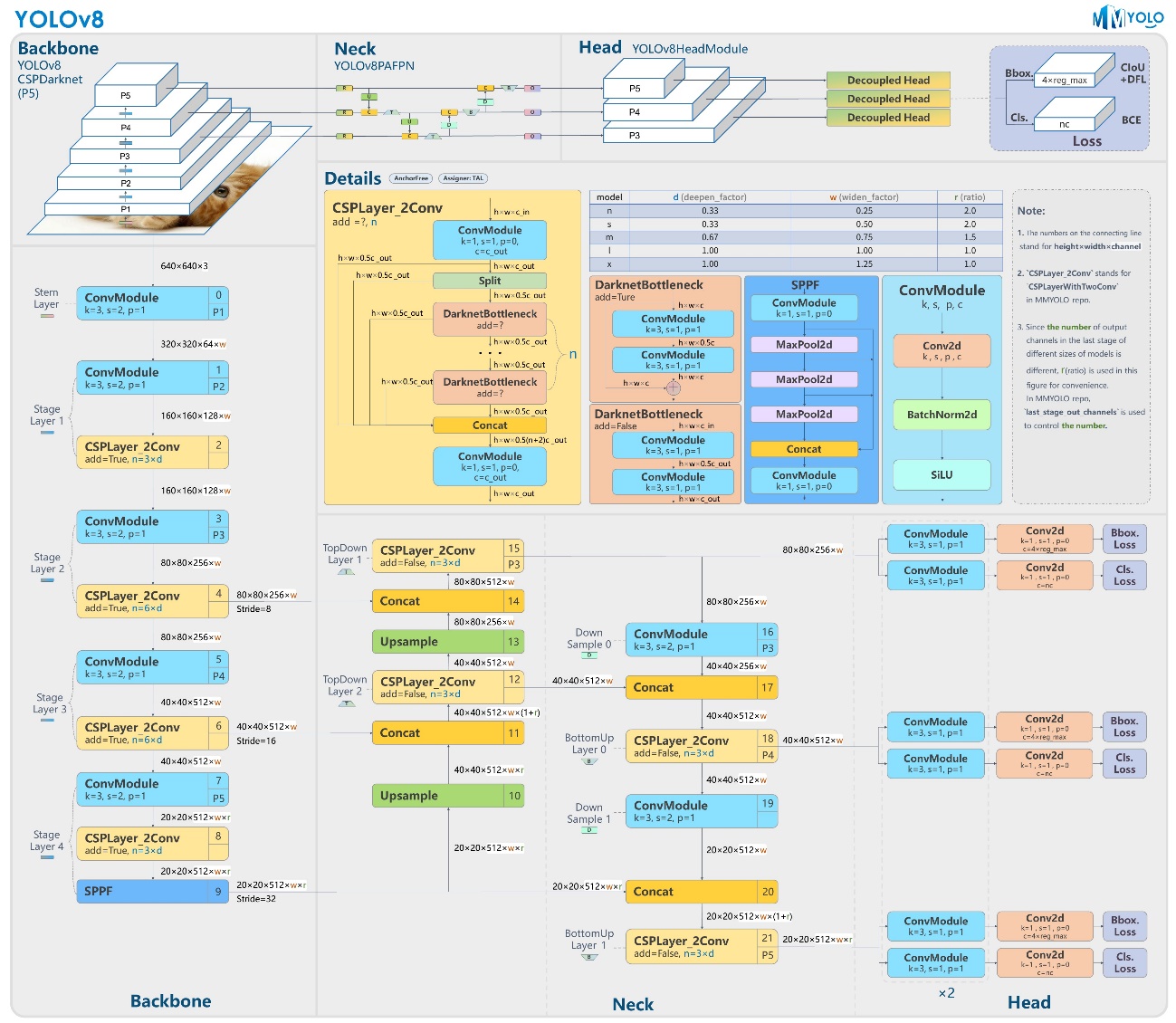


图 3.2 YOLOv8模型结构图

Fig. . YOLOv8 structure diagram

1. Backbone网络

YOLOv8的骨干网络为CSPDarknet，负责从输入图像中提取特征，该网络是 Darknet 的变体，使用了 Cross Stage Partial Networks（CSP）结构来加速计算和提高准确性，对比YOLOv5的CPSDarknet，两者模型结构一致，区别在于第一层的卷积核从6\*6变成了3\*3，此外，YOLOv8将v5的所有C3模块替换成了C2f模块，其结构对比见图 3.3，C3模块的主分支梯度使用若干个残差模块BottleNeck作为过渡，其残差模块个数由模型参数决定，相比C3模块，C2f的BottleNeck结构与C3卷积次数一致，只是在参数上有所差异，C2f还增加了更多的前向连接，并且添加了额外的分离操作，使得模型更加轻量化。

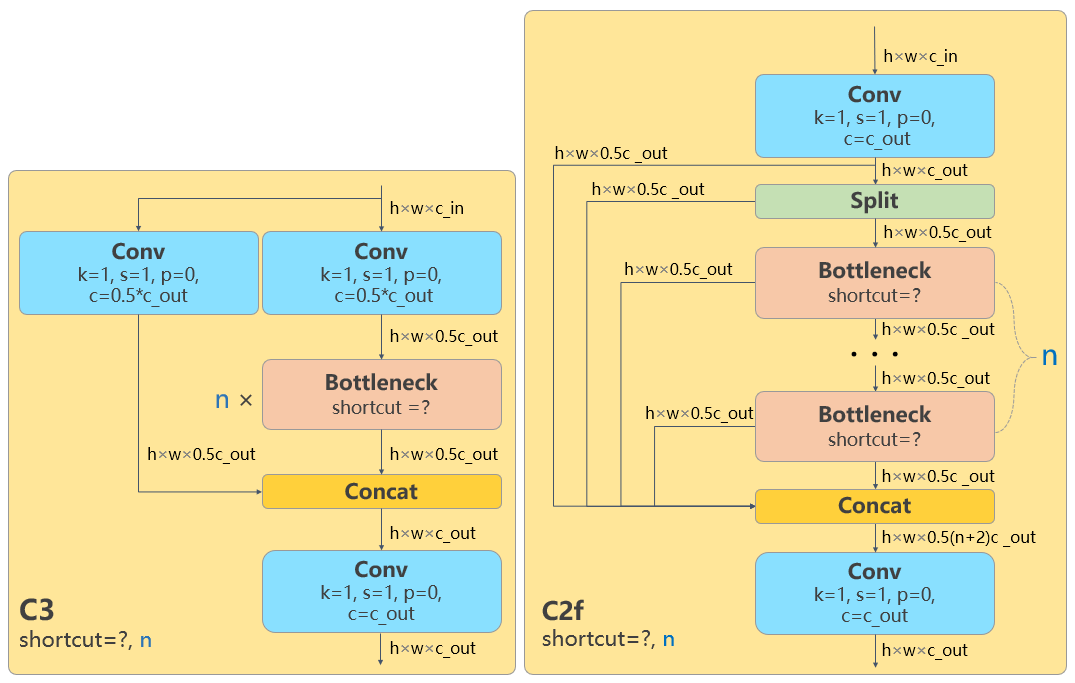


图 3.3 C3&C2f区别

Fig. . C3&C2f diff

1. Neck网络

在YOLOv8的Neck网络部分，选用PAFPN（Path Aggregation Feature Pyramid Network）网络[32]作为其特征提取网络，以改进YOLOv5在多尺度目标检测中的性能，PAFPN网络是一种特征金字塔网络，其主要思想是将不同尺度的特征图通过路径聚合的方式进行融合，从而得到更加丰富的特征表示。具体来说，PAFPN网络主要由两个部分组成：上采样和路径聚合。上采样部分将较低层次的特征图通过反卷积或双线性插值等方法上采样到与高层次特征图相同的分辨率。路径聚合部分将上采样后的特征图与高层次特征图进行聚合，得到更加丰富的特征表示。PAFPN网络通过多次上采样和路径聚合操作，将不同尺度的特征图进行有效的融合，并且可以保留更多的语义信息，从而提高目标检测的准确性和鲁棒性。

在YOLOv8中，PAFPN网络被用作YOLOv5的Neck网络的改进版本。PAFPN网络相比于原有的FPN网络和PANet网络，具有更好的性能和更快的速度。通过使用PAFPN网络，YOLOv8可以更好地处理多尺度目标，并且能够提供更加丰富和准确的特征表示，从而提高目标检测的准确性和鲁棒性。与YOLOv5相比，除了将所有的C3换成C2f之外，YOLOv8跳过了进行上采样之前两个1\*1卷积，将BackBone输出的所有特征直接进行上采样，对应网路结构差异见图 3.4。

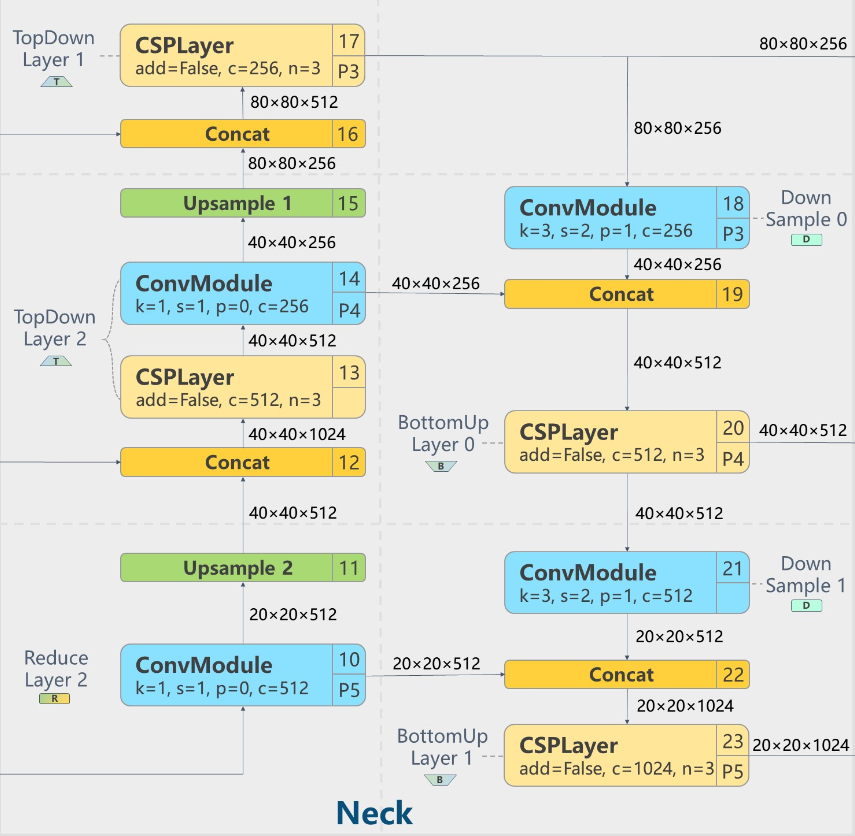
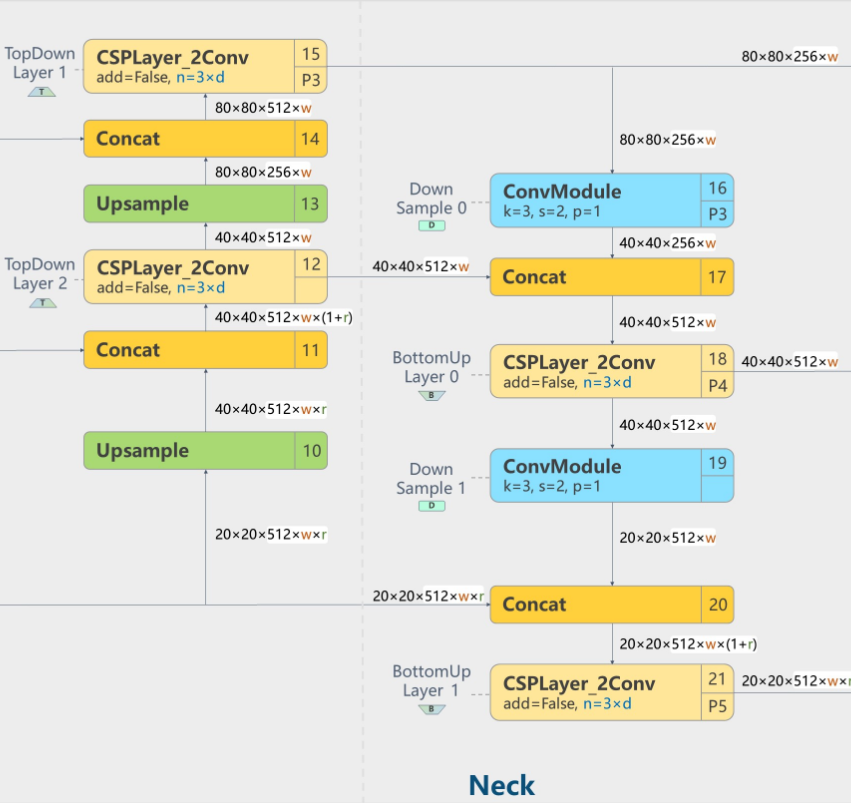
 

图 3.4 Neck网络差异

Fig. . Neck module diff

1. Head网络

对于YOLOv8的Head网络部分由原来的耦合结构改为解耦结构，如图 3.5。风格由YOLOv5的Anchor-Based改为Anchor-Free。

在Anchor-Based的目标检测方式中，首先需要在图像中选取一些先验框（即Anchor），然后利用卷积神经网络预测每个先验框内是否存在目标，并调整每个先验框的位置和大小以更好地匹配真实目标框。这种方式的优点是能够处理目标尺度变化较大的情况，并且对于较小的目标可以取得较好的检测效果。不过，Anchor-Based方法需要事先选取一些先验框，并需要对先验框的尺度、长宽比等参数进行调整，因此在复杂场景下需要较大的计算量和调试难度。

而在Anchor-Free的目标检测方式中，不需要预先选取先验框，而是直接根据像素点的特征预测目标框的位置和大小，并对每个像素点进行分类。这种方式的优点是能够更好地适应目标形状的多样性，并且在处理密集目标时能够取得更好的效果。不过，Anchor-Free方法需要预测每个像素点的坐标和分类信息，因此在计算效率上可能存在一定的问题。

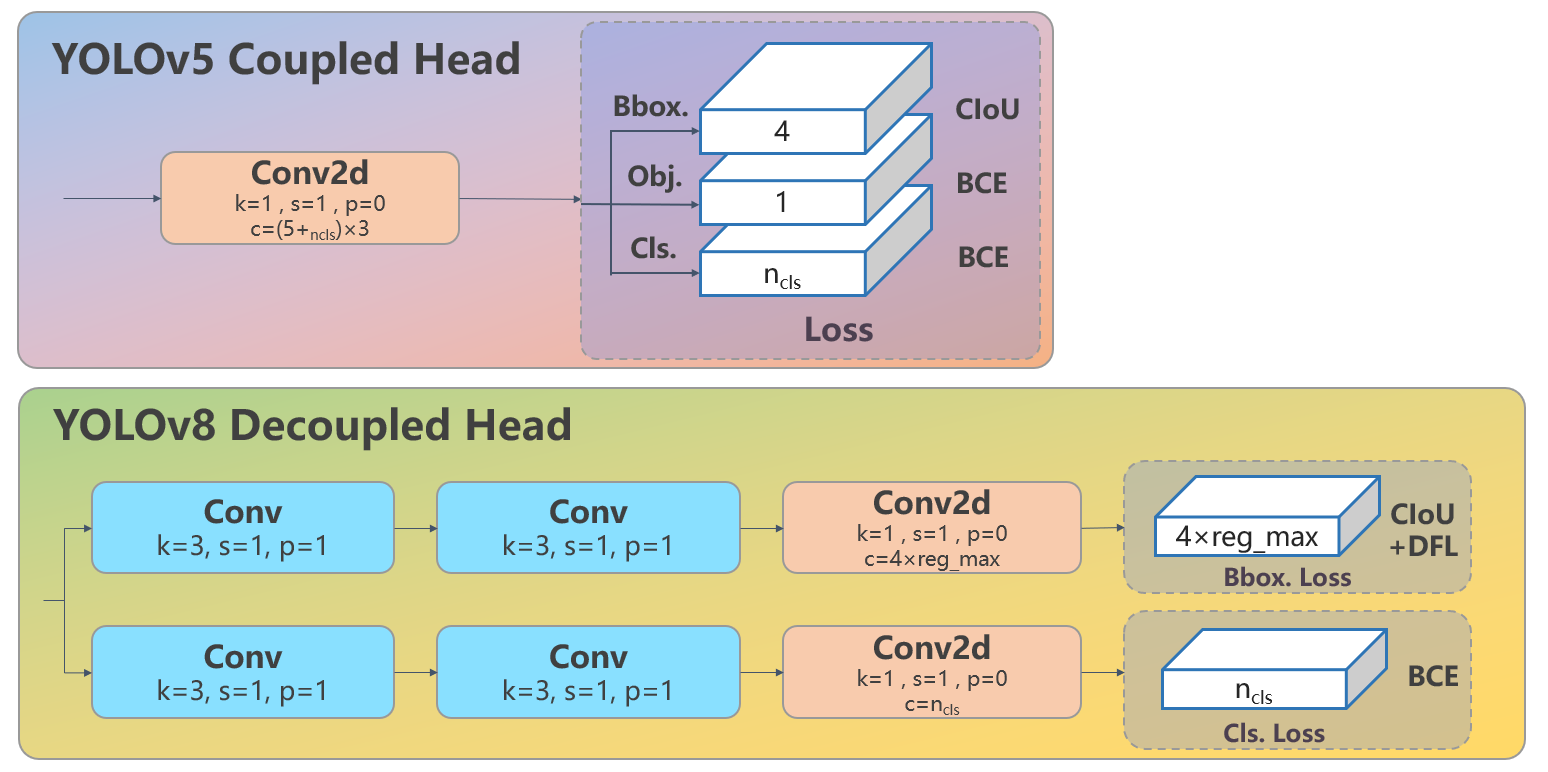


图 3.5 Head网络差异

Fig. . Head module diff

1. 损失值计算

YOLOv8的损失计算包括样本分配策略和损失计算。

大多数现代检测器都采用动态样本分配策略，例如 YOLOX 的 simOTA[33]、TOOD 的 TaskAlignedAssigner[34] 和 RTMDet 的 DynamicSoftLabelAssigner[35]。鉴于动态分配策略的优越性，YOLOv8 算法直接结合了 TOOD 的 TaskAlignedAssigner 中使用的算法。

TaskAlignedAssigner的核心思想是将目标检测任务分解为多个子任务，每个子任务关注特定的目标类别和难度级别。具体而言，TaskAlignedAssigner根据目标难度和类别将真实目标划分为多个任务，并将预测框与相应的任务进行对应。然后，TaskAlignedAssigner使用匈牙利算法对预测框和真实目标进行匹配，以获得最优的匹配结果。

对于第i个预测框和第j个真实目标，假设它们之间的IoU为，分类得分为，预测框坐标为，真实目标坐标为，任务权重为，则第k个子任务的得分为：

其中，是一个常数，用于控制匹配得分的平滑度，表示第i个预测框和第j个真实目标之间的空间距离，用于衡量目标框位置的匹配精度。最终的匹配结果为匹配得分最高的对应关系。

这种策略在YOLOv8中可以概括为：根据分类和回归的加权分数来选择正样本。

s是ground truth类别对应的预测分数，u是预测边界框和gt边界框的IoU。

将VFL Loss作为分类损失，主要改进是提出了非对称的加权操作，FL和QFL都是对称的。而非对称加权的思想来源于论文PISA[36]，该论文指出首先正负样本有不平衡问题，即使在正样本中也存在不等权问题，因为mAP的计算是主正样本。

q是label，正样本时候q为bbox和gt的IoU，负样本时候q=0，当为正样本时候其实没有采用FL，而是普通的BCE，只不过多了一个自适应IoU加权，用于突出主样本。而为负样本时候就是标准的FL了。可以明显发现VFL比QFL更加简单，主要特点是正负样本非对称加权、突出正样本为主样本。

对应的回归损失为CIOU Loss+DFL的形式，针对这里的DFL（Distribution Focal Loss），其主要是将框的位置建模成一个 general distribution，让网络快速的聚焦于和目标位置距离近的位置的分布。

DFL的含义是以交叉熵的形式去优化与标签y最接近的一左一右2个位置的概率，从而让网络更快的聚焦到目标位置的邻近区域的分布；也就是说训练出来的分布理论上是在真实浮点坐标的附近，并且以线性插值的模式得到距离左右整数坐标的权重。

1. 模型性能

Ultralytics给出了YOLOv8预训练模型的性能，参考表 3.2。

表 3.2 YOLOv8目标检测模型性能

Tab. . YOLOv8 performance in prediction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型 | 尺寸  (像素) | mAPval  50-95 | 推理速度  CPU ONNX | 推理速度  A100 TensorRT | 参数量  (M) | FLOPs  (B) |
| YOLOv8n | 640 | 37.3 | 80.4 | 0.99 | 3.2 | 8.7 |
| YOLOv8s | 640 | 44.9 | 128.4 | 1.20 | 11.2 | 28.6 |
| YOLOv8m | 640 | 50.2 | 234.7 | 1.83 | 25.9 | 78.9 |
| YOLOv8l | 640 | 52.9 | 375.2 | 2.39 | 43.7 | 165.2 |
| YOLOv8x | 640 | 53.9 | 479.1 | 3.53 | 68.2 | 257.8 |

上述性能参数由模型在COCO val2017[38]数据集上进行单模型单尺度测试而得出。

## 基于改进yolo模型的传感器实时目标检测

### 实验平台

实验采用改进的YOLOv8网络模型对路面采集的视觉信息进行目标检测，在硬件方面，使用Nvidia Jetson TX2作为系统控制器，其硬件参数信息如表 3.2。

|  |  |
| --- | --- |
| GPU | NVIDIA Pascal 架构 GPU，256 CUDA 核心，1.3 TFLOPS |
| CPU | ARMv8 64-bit 六核心 Denver 2 + ARM Cortex-A57 |
| RAM | 8 GB LPDDR4 1866MHz |
| 存储 | 32 GB eMMC 5.1 存储器 |
| 视频编解码器 | 4K @ 60Hz H.265, H.264, VP9 |
| 摄像头输入 | 12 CSI lanes，最高支持6个摄像头，最高分辨率2x 4K @ 30 FPS |
| 显示器输出 | HDMI 2.0 或 DP 1.2 |
| USB | 1个USB 3.0，2个USB 2.0 |
| 网络 | 千兆以太网 RJ45，双频 802.11ac 网络和蓝牙 4.1 |
| 其他接口 | 1个M.2 Key-E 插槽，1个M.2 Key-M 插槽，GPIO，I2C，I2S等接口 |
| 尺寸 | 87 x 50mm |
| 电源 | 5V/4A DC |

表 3.3 TX2参数

Tab. . TX2 parameters

Nvidia Jetson TX2作为高性能、低功耗人工智能超级计算机模块，显然不适合用来进行模型的训练，因此，本次实验选用AMD R5 3600中央处理器搭配NVIDIA GTX 1660 super显卡来完成实验中的训练部分，对应训练实验环境参考表 3.3。

表 3.4 训练实验环境

Tab. . Lab environment for training

|  |  |
| --- | --- |
| 操作系统 | Ubuntu 20.04 |
| CPU | AMD R5 3600 |
| 内存大小 | Kingston FURY 16GB DDR4 3200MHz |
| GPU | GTX 1660 super |
| 显卡驱动 | 530.30.02 |
| 显卡个数 | 1 |
| 显存容量 | 6G |
| CUDA | 10.2 |
| CuDNN | 8.7.0 |
| Python | 3.10.9 |
| Pytorch | 1.12.1 |

### 模型训练样本

结合目标检测模型应用场景，为了提高其检测精度，本文在现有公开自动驾驶数据集Argoverse基础上，将训练好的目标检测网络模型应用到第二章采集到的图像数据并进行迁移学习。从而减少因采集到的样本量较少而对模型结果产生的影响。

1. Argoverse自动驾驶数据集

Argoverse是一种用于自动驾驶技术开发的公共数据集和API工具包，由Argo AI公司创建并维护。Argoverse的目标是提供一个完整的、高质量的数据集，以帮助研究人员和开发人员进行自动驾驶技术的研究和开发。

Argoverse数据集包括城市驾驶场景中的各种传感器数据，如车辆相机、激光雷达、GPS、惯性测量单元和环境传感器等。这些数据可以用于训练和评估各种自动驾驶技术，包括目标检测和跟踪、路径规划和控制等。Argoverse数据集目前覆盖了美国多个城市的各种驾驶场景，包括城市街道、高速公路、交叉口和停车场等。

作为图像检测网络模型，本次实验选择提取Argoverse1.1中的图像数据作为图像样本，文献[37]基于Argoverse1.1，构建了带有高帧率注释的数据集Argoverse-HD，可用于图像或视频的对象检测、多对象跟踪和预测等。整理好的数据集结构如图 3.6所示。其中的图像和标注内容都按照训练、验证和测试进行了分类。

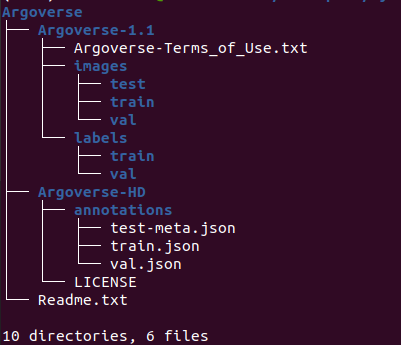


图 3.6 Argoverse目录结构

Fig. . Argoverse file tree

数据集总共包含66953张图像，其中60%作为训练样本（图 3.7），剩下的分别当作验证样本（图 3.8）和测试样本（图 3.9）。样本将目标按照10个类别分别进行标注，包括行人、自行车、汽车、摩托车、公共汽车、货车、交通灯和停车标志。



图 3.7 训练样本

Fig. . Samples for training

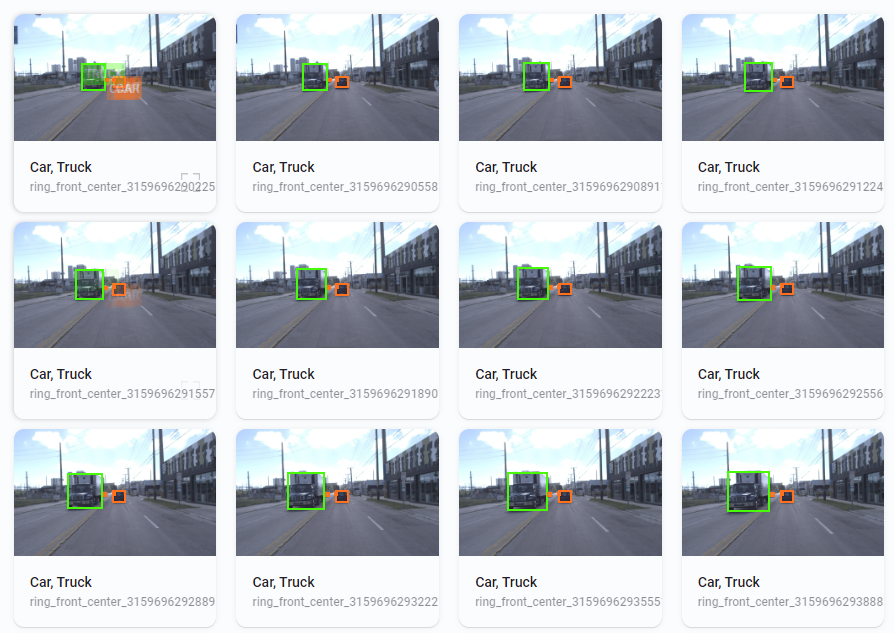


图 3.8 验证样本

Fig. . Samples for validation



图 3.9 测试样本

Fig. . Samples for test

1. 个人小样本数据集

本次实验采集的数据集样本主要以校园内道路环境为目标，在Argoverse数据集的分类标注基础上，进一步优化预测目标，以智能驾驶元素作为主要对象，同样对应Argoverse数据集分类：行人、自行车、汽车、摩托车、公共汽车、货车、交通灯和停车标志，自主完成数据集的制作。

关于自制样本数据集，首先需要将训练用的图像准备好，将第二章采集到的视频按照一定的频率截取成图像，为了避免图像的内容、角度过于单一，本次实验还在网络上收集清晰、高分辨率、光线充足、背景干净并且符合实际应用场景的图像，以便模型更好地识别物体和场景，从而提高模型的鲁棒性和泛化能力。

，总共1382张图像作为样本，按照8：2的比例拆分为训练集与验证集，由于采集的样本数量相对来说不够充分，并且推荐样本数量大于1500，因此本次实验训练样本不提供测试集，同时，在样本标注过程中，

### 实验结果分析

## 本章小结

# 基于PartA2的点云目标感知算法

激光雷达通过传感器的激光反射计算空间坐标，光束经过空间的延申后变得稀疏，对应的点云数据也呈现出稀疏的状态，此外，激光雷达生成的点云并没有按照一定的规律排序，因此，对于稀疏、无序的点云数据，几乎无法提取相邻点之间的几何特征，而在图像领域应用广泛的卷积神经网络，将其应用到点云检测，同样也因为点云数据的三维特性而面临巨大挑战。本章将对第二章采集的激光雷达点云数据做目标检测处理，与图像的目标检测相比，点云的目标检测数据更加庞大，并且还需要一些额外的数据预处理操作。

## 基于深度学习的三维点云目标检测基础

### 点云数据表示

点云是由大量三维点构成的数据结构，因此需要对点云数据进行有效的表示。常见的点云表示方法包括点的坐标、法向量、颜色等。另外，基于深度学习的三维目标检测算法还需要考虑点云数据的稠密性、不规则性等特点。在ROS中，点云主要以PointCloud2格式编解码，表 4.1为PointCloud2数据格式定义。

表 4.1 PointCloud2数据格式

Tab. . PointCloud2 data definition

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| Header | std\_msgs/Header | 消息头部信息，包括时间戳、帧ID等 |
| height | uint32 | 点云数据的高度 |
| width | uint32 | 点云数据的宽度 |
| fields | sensor\_msgs/PointField[] | 描述每个点的数据类型和相对偏移量的数组 |
| is\_bigendian | bool | 点云数据的字节序，true表示大端序，false表示小端序 |
| point\_step | uint32 | 每个点的字节长度 |
| row\_step | uint32 | 每行点云数据的字节长度 |
| data | uint8[] | 点云数据的字节流 |
| is\_dense | bool | 点云数据是否密集，即是否存在NaN或INF等无效数据 |

其中，sensor\_msgs/PointField类型的fields字段包含以下三个成员变量：

表 4.2 fields字段说明

Tab. . membership of fields

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 描述 |
| name | string | 该点的数据类型名称 |
| offset | uint32 | 该数据类型在点数据结构中的偏移量 |
| datatype | uint8 | 该点数据类型的编码值，如FLOAT32、UINT16等 |

需要注意的是，PointCloud2数据格式可以存储各种类型的点云数据，因此fields数组中的成员变量可以根据具体情况进行调整和扩展。同时，需要保证数据的字节序和数据类型的编码值与实际数据的格式一致，否则会导致数据解析错误。

### 点云的地面分割处理

地面分割是三维点云处理的一个基本任务，它是将点云中属于地面的点与非地面的点进行区分的过程。在许多应用中，例如机器人导航、无人驾驶车辆和建筑物三维建模等，地面分割是非常重要的。本文使用RANSAC对点云数据进行地面分割。

RANSAC（Random Sample Consensus）是一种经典的随机抽样一致性算法，常用于估计数据中的模型参数。其基本原理包括以下步骤：

首先从数据集中随机选择一小部分样本，用于估计模型参数。样本数量应该足够大，以保证估计出来的模型参数的准确性。假设样本数量为 n。使用估计的模型参数，计算所有数据点与模型之间的误差。对于一个数据点，其到平面的距离可以表示为：

根据一定的误差阈值t，判断哪些数据点被归类为模型内点，哪些被归类为模型外点。如果某个数据点的距离小于t，则将其归类为内点，否则归类为外点。

计算内点数量m，如果m大于某个阈值M，则使用所有内点重新拟合模型，并计算模型的拟合误差。如果拟合误差小于预设的误差阈值，则认为已找到最优模型，算法终止。否则回到步骤1，重复执行若干次，直到满足停止条件。

最终在所有迭代中得到的模型中，选择拟合度最好的模型作为最终的模型参数。

RANSAC算法的主要优点是可以处理有噪声和异常点的数据，具有一定的鲁棒性。但是其缺点是在迭代次数较少的情况下，可能无法找到最优的模型，同时需要事先确定样本数量和误差阈值等参数。

将RANSAC算法应用到点云地面分割任务，首先，从点云数据中随机采样一小部分点作为初始种子点，用最小二乘法拟合平面模型。然后，计算所有点到平面的垂直距离，取阈值为0.2米，将距离小于阈值的点归类为地面点，其余点为非地面点。

接下来，使用所有被归类为地面点的点重新拟合平面模型，并重新计算所有点到新平面的垂直距离。将上述步骤迭代50次，直到地面模型不再发生显著变化。

最后，将所有被归类为地面点的点提取出来，即可得到地面点云。非地面点可以进一步用于障碍物检测、物体识别等任务。实验结果见图 4.1。

图 4.1 点云地面分割结果

Fig. . Point cloud segmentation

实验选用激光雷达采集的实时点云作为分割目标，将点云分割前后进行对比，可以明显地看到，图中地面环形点云被成功分割，分割后的点云数据为后续点云处理任务提供基础数据支持。

## PartA2网络模型

PartA2点云目标检测算法是基于VoxelNet算法的改进版本，它在VoxelNet算法的基础上，进一步优化了模型结构和算法流程，提高了检测精度和速度。PartA2网络结构包括两个部分，如图 4.2，它是一个双阶段网络模型，主要包括part-aware stage以及part-aggregation stage两部分，前者主要用于检测具有明显部件（Part）的物体，后者主要作用对象时没有明显部件的物体。

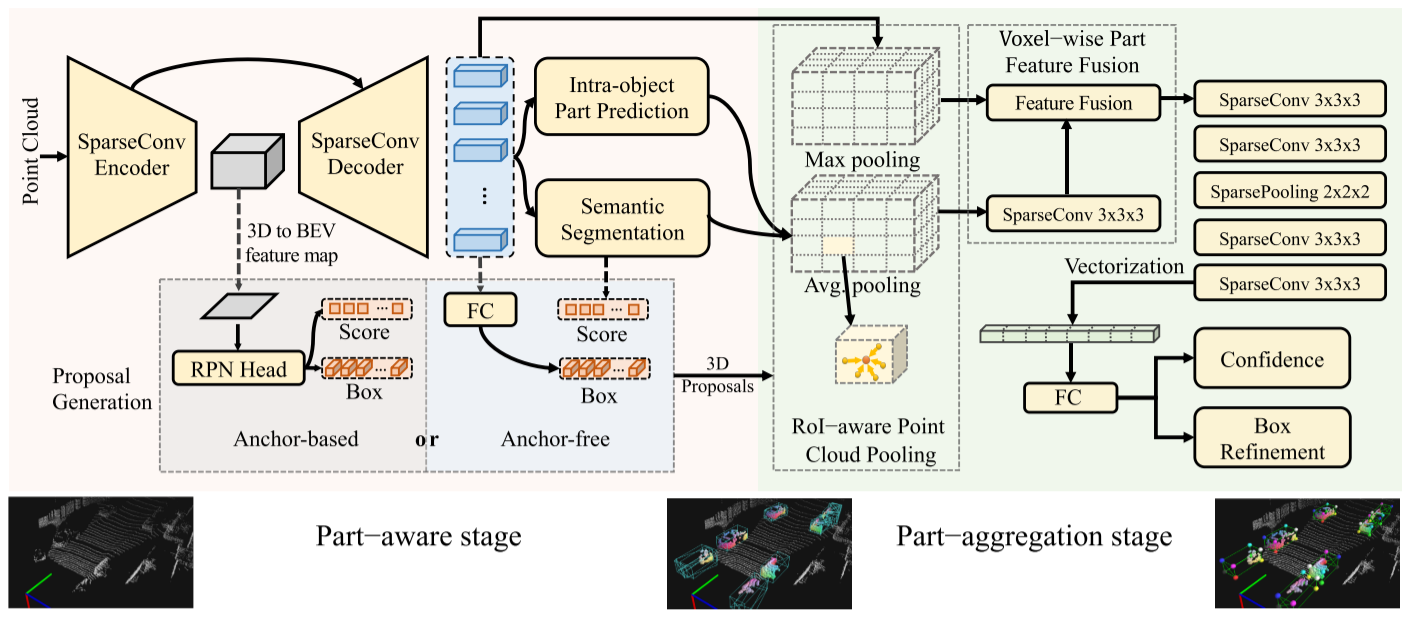


图 4.2 PartA2网络结构

Fig. . PartA2 structure diagram

### Part-aware stage

Part-aware stage 主要是为了解决一些复杂物体的检测问题，例如汽车、行人和自行车等。这些物体通常具有明显的部件，例如车轮、车灯、挡风玻璃、车门等，因此在检测时需要关注这些部件。

首先使用一个基于稀疏卷积的Encode-Decode骨干网络（Backbone）对输入点云进行特征提取，如图 4.3所示，主要通过3d稀疏卷积和3d稀疏反卷积实现网络的特征提取。经过步长为2的一些卷积层将输入将采样至原本大小的八分之一，然后再上采样至原来的分辨率，这样就得到了voxel-wise的特征，后续将基于此特征进行局部位置预测和分割掩码预测。

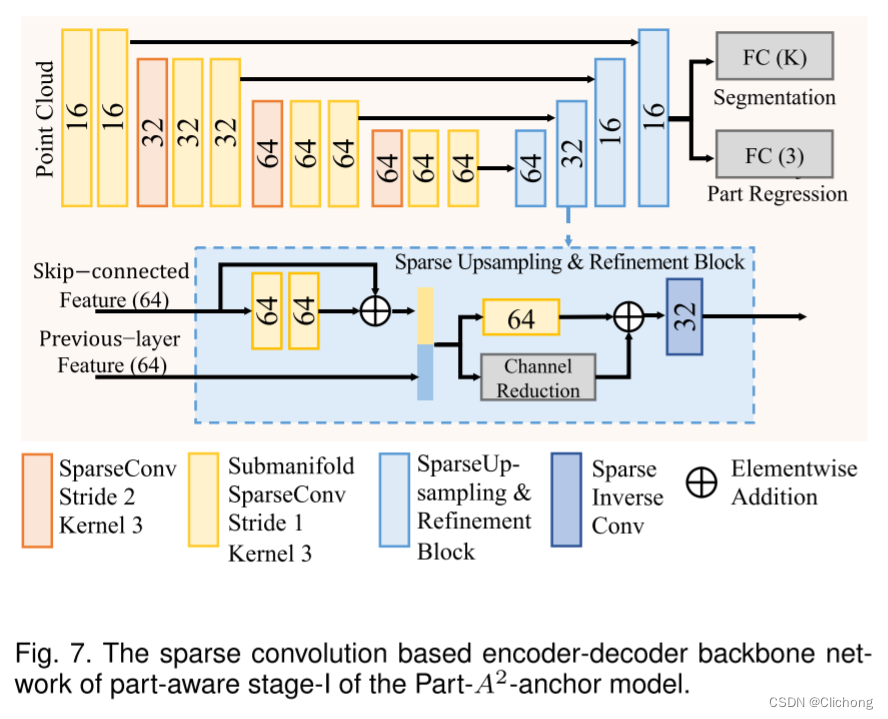


图 4.3 基于稀疏卷积的encoder-decoder backbone

Fig. . The sparse convolution based encoder-decoder backbone

将得到的特征输入到Part-aware RPN（Region Proposal Network）中进行目标检测。与传统的RPN不同，Part-aware RPN还考虑了物体的部件信息，将物体分解为若干部件，并在各个部件上生成候选框，从而提高检测精度。

具体来说，Part-aware RPN包括两个子网络，即Part-aware Proposal Subnet和Part-aware Refinement Subnet。Part-aware Proposal Subnet用于生成候选框，首先将点云数据划分为若干个三维体素（Voxel），并对每个体素中的点云进行特征提取，然后将提取的特征输入到Part-aware Proposal Subnet中，生成候选框。Part-aware Refinement Subnet则用于对生成的候选框进行修正，通过进一步优化候选框的位置、大小和朝向等参数，提高检测精度。

其中，关于候选框的生成，在PartA2中设计了两种候选框生成策略：基于anchor-free和anchor-based生成候选框。

1. Anchor-free候选框生成

在PartA2-free结构中，对于Encoder-Decoder提取出来的特征，可以通过语义分割的掩码获取前景点，随后对这些前景点使用另外的一个分支来预测生成3d候选框。对于每个前景点如果直接回归标注框，那么回归的目标可能在一个比较大的范围内变化。比如GT角点上的前景点，其到对象中心的相对偏移远大于对象侧面上的前景点的相对偏移，所以如果对每个前景点直接预测相对偏移量，那么总损失将被角点除的前景点的偏移损失所支配，这样会导致损失不平衡。

为了避免直接回归带来的损失平衡，PartA2-free采用bin-based的回归方式，与PointRCNN中一致。也就说，对于xy坐标以及角度θ来说，先分类缩小回归距离然后对残余距离进行回归操作，也就是先确定位置大概的区域后再进行小范围回归（其中角度的范围是2π，划分为w个区域）。而对于位置坐标z以及尺寸信息hwl，可以直接进行smooth-L1回归操作。详细见我在PointRCNN中的笔记。损失组成如下所示：

1. Anchor-based候选框生成

采用anchor-free生成候选框策略的PartA2称为PartA2-free，在此结构中，对Encoder中输出的编码进行8倍下采样同时将特征投影到鸟瞰图中，也就是将M\*N\*H\*D转换为，相当于对于z轴的特征信息全部拼接了起来，转为为了2d的特征图。之后的操作与之类似，在生成的2d特征图中的每个特征点，对于每个类别设计2种互相垂直方向但尺寸一样的3d先验框（prios）。正负样本通过设置在鸟瞰图上不一样的iou实现，比如对于汽车、行人和骑自行车的人，正样本IoU阈值根据经验分别设置为0.6、0.5、0.5，负样本IoU阈值分别设置为0.45、0.35、0.35。

通过iou筛选后的anchor一般都在GT附近，所以可以直接进行回归处理，此时的回归距离是比较小的因此不需要采用anchor-free的那种分区加上残差距离预测来实现。这里的损失设置与SECOND方法完全一致，使用了来编码方向损失，消除方向的模糊性。此处，如果使用其他方法，会将两个相反的方向编码为相同的值，所以需要第三个分支来预测方向。由于gt的方向属于-π~π，所以这里的分支预测如果为正，则方向目标为1；否则方向目标为0。损失组成如下所示：

这两种3D候选框策略都有其优点和局限性。所提出的无锚点策略通常是轻量级的并且存储器高效的，因为它不需要评估3D空间中的每个空间位置处的大量锚点。由于3D对象检测中的不同类别通常需要不同的锚框，而无锚点方案可以共享逐点特征以生成多个类别的建议，因此对于多类别对象检测的效率更明显。第二种基于锚点的建议生成策略通过为每个类别预先定义锚点来覆盖整个鸟瞰图特征图，实现了略高的召回率，但具有更多的参数，需要更多的GPU内存。

Part-aware stage的另一个关键之处在于利用Part-aware Non-Maximum Suppression（NMS）算法来处理检测结果，它主要用于过滤掉检测结果中的冗余框，从而提高检测的精度和效率。具体流程如下：

* + 1. 将检测结果按照类别进行分组。
    2. 对于每个类别，按照置信度对检测结果进行排序。
    3. 选择置信度最高的检测框，并将其与其余检测框进行重叠度（IoU）计算。

其中，（Area of Intersection）表示两个检测框的交集面积，（Area of Union）表示两个检测框的并集面积。

* + 1. 对于与置信度最高的检测框重叠度大于一定阈值的检测框，将其从候选框集合中删除。
    2. 重复步骤3和步骤4，直到所有检测框都被处理完毕。
    3. 将处理后的检测框按照置信度进行排序，并输出前K个最终检测结果。

总的来说，Part-aware stage通过考虑物体的部件信息，提高了检测的精度和鲁棒性，对于具有明显部件的物体具有较好的检测效果。

### Part-aggregation stage

Part-aggregation stage主要用于对proposal中的所有点进行聚合，得到整个proposal的part-aware特征表示。Part-aggregation stage包含了两个主要的步骤：ROI-aware feature pooling和Part-aware Feature Aggregation，分别表示ROI的特征池化和候选框的特征聚合操作。

1. ROI-aware feature pooling

在进行roi池化之前，需要先进行坐标系的规范化，这与PointRCNN中同样一致，所谓的坐标系规范化就是将每个候选框中的汇集点（pooled points）转换到各个相应候选框中的坐标系中。对于每一个候选框的规范坐标系的具体的操作是：

1. 原点位于框方案的中心；
2. 局部X和Y轴近似平行于地平面，其中X指向建议的头部方向，而另一Y轴垂直于X；
3. Z轴与整体坐标系的Z轴保持一致。候选框中的所有汇集点坐标p都进行适当的选择和平移操作转化到规范坐标系p`上。正样本的候选框与其对应的GT同样被变换到规范坐标系以计算用于回归距离。

规范坐标系基本上消除了不同3D方案的大量旋转和位置变化，并提高了特征学习的效率，以用于随后的框位置修正。

在PointRCNN的工作中，点汇集操作直是将一个候选框中的所有点特征进行聚合来对候选后进行修正，但这种3d候选框汇总引入了模糊性的问题，如图 4.4所示，不同的候选框之间可能包含的是一样的汇集点，这种相同的汇集特征pooling feature可能会导致后续修正框的性能。

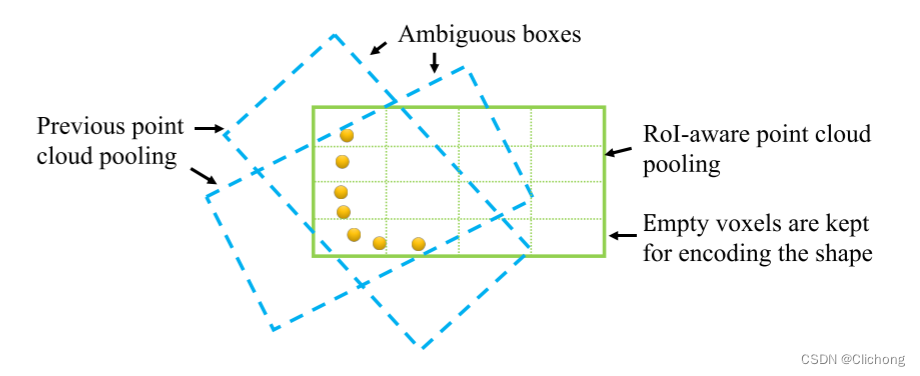


图 4.4 多候选框特征汇集

Fig. . Multi-boxes feature aggregation

解决这种模糊性问题的关键是需要利用空体素的信息，来捕获完整的候选框空间位置。在PartA2的完整结构图中可以看出，这里对每一个候选框均匀的划分为一个固定的空间形状（这里是14\*14\*14的大小）。也就是说，对于任意不同尺寸的候选框，都划分成14\*14\*14的大小，对proposal进行voxel化。那么，对于候选框内的每个voxel，都会包含Part-aware Stage中的一些点（其实也是体素中心）。其中，对于每个voxel中的点，将其Encoder-Decoder输出的point-wise特征，进行max pooling池化操作；将其point-wise的内部信息（point-wise part location）以及语义分数（semantic scores）两个部分的特征拼接起来，进行avg pooling池化操作。那么，对一个切分为14\*14\*14的候选框来说，分别构建了两个池化特征，维度组成为14\*14\*14\*C1与14\*14\*14\*C2。

将候选框内部点分别根据不同的Point-wise信息构建成两个池化特征层既完成了ROI-aware feature pooling操作，同时考虑了空体素信息以及非空体素的信息。

1. Part-aware feature aggregation

继ROI-aware feature pooling操作之后，将两个3d的特征层在channels上进行拼接（先对avg pooling特征进行一个稀疏3d卷积），也就是变换成14\*14\*14\*(C1+C2)。随后堆叠几个卷积核大小为3\*3\*3的稀疏卷积进行特征提取，再进行一个2\*2\*2的稀疏池化进行下采样操作，将特征体积降采样到7\*7\*7，以节省计算开销和参数。最后将特征转为为一个向量（feature vector），空体素保持为0，这里也是按照特定顺序对聚合的候选框voxel进行排列的。随后送入两个分支进行框评分以及修正。

与直接对点特征进行聚合相比，这里的part location聚合方法实现了局部信息以及全局信息的聚合，有效的学习到物体内部的空间分布信息，并且利用3d稀疏卷积节省了大量的计算量和参数量。同时，这里的置信度预测利用了soft label来进行交叉熵回归计算。如果iou>0.75，置信度的标签设置为1；iou<0.25，置信度标签设置为0；其余范围设置为2Iou-0.5（这里的iou是只候选框与GT的3d Iou计算）。

### 损失函数

对于part-aware的损失主要包括三个部分：用于前景点分割的焦点损失、用于部分位置回归的二元交叉熵损失以及用于3D候选框生成的平滑L1损失

其中,对于3d候选框生成的损失中，对PartA2-free采用的是bin-based的框生成损失（bin-based box generation loss），对PartA2-anchor模型采用基于直接回归的框生成损失（residual-based box regression loss）。

对于part-aggregation的损失主要包括两个部分：候选框质量（objectness）回归的二元交叉熵损失项、用于3d候选框修正细化的平滑L1损失项：

在第二阶段的损失函数中，还额外添加了8个角损失来进行正则化。虽然这种损失计算出来的信息是冗余的，但它是一种很好的正则化手段，在3DSSD中也使用上了这个附加损失，以实现更好的方向限制，提高准确率。此外，二阶段的方向损失不再需要新添加一个正负类别判断，而是直接回归即可，与PartA2-anchor稍有不同，这得益于iou评分的限制。

总的损失是以上两个阶段损失的叠加：

## 点云目标检测实例

## 本章小结

# 基于多传感器的智能感知系统

在第三章和第四章，本文分别完成了基于摄像头的目标检测算法和基于激光雷达的目标检测算法，其中使用到的传感器也是智能驾驶领域中非常常见的传感器，基于此但不限于此，本章以目标感知为目的，设计了基于多传感器的智能感知系统，在传感器与对应算法的驱动下，系统结合已知的环境信息，与用户进行交互，提供智能感知的结果，以实现前文算法的应用。

## 智能感知系统需求分析

1. 数据交互
2. 数据采集

在智能感知系统中，数据的来源非常广泛，智能感知系统需要从多个传感器中收集数据，包括但不限于车载摄像头、激光雷达、毫米波雷达、GPS、IMU等。此外，传感器所反馈的数据又因传感器的不同，其对应的通信协议以及数据格式也各不相同，因此，这些传感器采集到的数据需要进行统一处理和管理，以便为系统提供准确和实时的数据支持。

1. 数据传输

在数据传输过程中，尤其需要考虑到数据的大小，在智能感知系统中，图像数据和点云数据是数据量的主要部分，其中，点云数据里每一帧都包含上万个点的三维空间坐标，鉴于此，网络的传输带宽，延迟是保证数据传输稳定的重要依赖。

1. 数据存储

系统需要将采集到的数据进行存储，以便后续的数据分析和使用。数据存储需要考虑数据的规模、存储结构、数据安全等问题。在数据存储过程中，需要保证所存储的数据在时间和空间上保持同步，此外，由于点云数据占用内存巨大，在数据存储过程中还需考虑数据的压缩，在有限的空间内优先存储价值更大的数据。

总之，为了保证智能感知系统中的数据交互部分的正常运行，需要采用多种技术和方法，例如分布式系统技术、云计算技术、大数据技术、人工智能技术等。同时，还需要采用合适的数据交互协议和数据安全措施，以保障数据的安全和稳定性。最终，智能感知系统中的数据交互部分需要通过优化和改进，不断提升系统的性能和可靠性，为智能驾驶提供更加准确和可靠的数据支持。

1. 人机交互

在智能感知系统中，系统将传感器与计算的感知结果可视化，使用户能够更加直观地看到系统的感知结果，为了使用户易于理解和分析，本文从以下几个方面出发，设计智能感知系统，帮助用户快速地发现模式、趋势、异常和规律。

1. 提供更直观的表达方式

将数据转化为图表、图形等视觉元素可以更直观地展示数据和信息，让用户更容易理解和分析。通过视觉化的方式，用户可以更直观地看到数据之间的关系，从而更容易发现其中的规律和趋势。

1. 突出重点信息

数据可视化可以突出重点信息，使用户更快速地理解和把握信息的核心内容。通过调整图表和图形的颜色、大小、形状等属性，可以将重要的数据和信息突出显示，从而使用户更容易理解和记忆。

1. 提高效率和准确性

数据可视化可以帮助用户更快速、更准确地分析和处理数据。通过视觉化的方式，用户可以快速地浏览和比较数据，从而更快地发现数据之间的关系和趋势，减少错误和误解的可能性。

1. 改善交互体验

数据可视化可以提升用户的交互体验，使用户更加愿意使用和参与到数据分析和决策过程中。通过交互式的数据可视化工具，用户可以更自由地探索和分析数据，从而更深入地了解数据和信息。

1. 系统的多平台兼容性

系统的多平台兼容性指所设计的智能感知系统能够在不同的设备上运行、展示，系统采用响应式设计，可以根据不同设备的屏幕大小和分辨率，自适应地调整布局和显示效果，使用户在任何设备上都可以获得最佳的用户体验，进一步，在系统开发过程中，还需要进行浏览器测试和兼容性测试，确保系统能在主流浏览器上运行，包括Chrome、Firefox、Safari等。

为了保证智能感知系统的多平台兼容性，对应平台的数据通信协议和数据格式也应该多平台兼容。在数据传输过程中，需要采用通用的数据格式或者进行数据转换，以确保不同平台之间的数据交互能够正常进行。

以系统兼容性为需求设计智能感知系统，进一步扩展了该系统的未来可塑性，可在智能感知系统的基础之上，进一步与物联网相结合，从而实现多车协同，远程控制等功能。

## 智能感知系统架构

基于智能感知系统的需求分析，分别从硬件和软件两个维度设计系统架构。

### 智能感知硬件架构设计

智能感知系统的硬件架构主要在于传感器及其通信，将多种传感器安装在车辆上，并通过数据转换和传输设备连接至计算机，计算机再将必要的控制信息通过通信电缆发送至车辆底盘。其硬件结构见图 5.1。

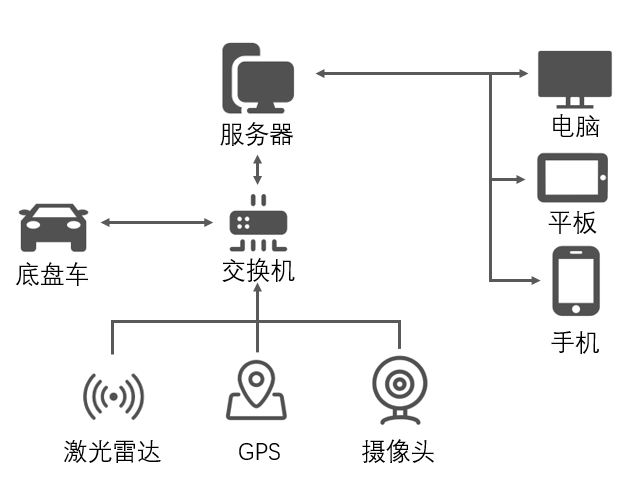


图 5.1 系统硬件架构

Fig. . System hardware architecture

### 智能感知软件架构设计

## 系统功能设计

## 智能感知系统实现

## 功能测试

## 本章小结

# 总结与展望

## 课题总结

## 课题展望

# 参考文献

1. 哈全财. 城市机动车道路拥堵治理方案研究[J]. 科技传播, 2013(1):2.
2. 李克强. 智能网联汽车的发展现状与对策建议[J]. 机器人产业, 2020(6):8.
3. 黄思源, 刘利民, 董健,等. 车载激光雷达点云数据地面滤波算法综述[J]. 光电工程, 2020, 47(12):12.
4. Moujahid A, Tantaoui M E A, Hina M D, et al. Machine learning techniques in ADAS: a review[C]//2018 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2018: 235-242.
5. Tunnell J, Asher Z D, Pasricha S, et al. Toward improving vehicle fuel economy with ADAS[J]. SAE International Journal of Connected and Automated Vehicles, 2018, 1(12-01-02-0005): 81-92.
6. Yaqoob I, Khan L U, Kazmi S M A, et al. Autonomous driving cars in smart cities: Recent advances, requirements, and challenges[J]. IEEE Network, 2019, 34(1): 174-181.
7. 蔡自兴.智能驾驶技术研究的重要进展[J].机器人技术与应用,2019,0(3):37-39
8. 杨贵栋,张帅乾,王亚飞,籍庆辉,郑欲锋.智能驾驶汽车驾乘舒适性测评方法研究综述[J].上海汽车,2019(08):13-21.
9. 陈锦栋. 浅析智能驾驶及其关键技术[J]. 中国科技纵横,2020(23):23-24,59.
10. Rojas-Rueda D, Nieuwenhuijsen M J, Khreis H, et al. Autonomous vehicles and public health[J]. Annual review of public health, 2020, 41(1): 329-345.
11. 黄窈蕙.智能驾驶汽车视觉图像处理技术[J].公路交通科技：应用技术版,2018,0(8):323-326
12. 张晶晶,杨鹏,刘元盛,梁军.基于智能驾驶的动态目标跟踪研究[J].计算机工程,2018,44(7):14-19
13. Khan M Q, Lee S. Gaze and eye tracking: Techniques and applications in ADAS[J]. Sensors, 2019, 19(24): 5540.
14. 李银国,周中奎,白羚.基于双目图像的大尺度智能驾驶场景重建[J]计算机科学,2019,46(S11):251-254+259
15. 陈晓冬,张佳琛,庞伟凇,艾大航,汪毅,蔡怀宇.智能驾驶车载激光雷达关键技术与应用算法[J].光电工程,2019,46(7):28-40
16. 宋绍京,陈育伟,胡海江,胡金艳,龚玉梅,邵慧.适用汽车智能驾驶的多光谱激光雷达波长选择可行性研究[J]红外与毫米波学报,2020,39(1):86-91
17. Li Y, An Z, Wang Z, et al. V2x-sim: A virtual collaborative perception dataset for autonomous driving[J]. arXiv preprint arXiv:2202.08449, 2022.
18. Sualeh M, Kim G W. Dynamic multi-lidar based multiple object detection and tracking[J]. Sensors, 2019, 19(6): 1474.
19. Taşdelen E A, Sezer V. Comparison and application of multiple 3D LIDAR fusion methods for object detection and tracking[C]//2020 5th International Conference on Robotics and Automation Engineering (ICRAE). IEEE, 2020: 64-69.
20. Kanno A, Takaoka R, Otani S, et al. Handheld millimeter-wave radar and lidar systems using an IMU device[C]//Passive and Active Millimeter-Wave Imaging XXII. SPIE, 2019, 10994: 62-67.
21. Xiangwei D, Fei Q I N, Xiangxi B U, et al. A Robust Perception Algorithm Based on a Radar and LiDAR for Intelligent Driving[J]. Journal of Radars, 2021, 10(4): 622-631.
22. Xiong H, Yu D, Liu J, et al. Fast and robust approaches for lane detection using multi‐camera fusion in complex scenes[J]. IET Intelligent Transport Systems, 2020, 14(12): 1582-1593.
23. Majumder U K, Blasch E P, Garren D A. Deep learning for radar and communications automatic target recognition[M]. Artech House, 2020.
24. Wang Z, Miao X, Huang Z, et al. Research of target detection and classification techniques using millimeter-wave radar and vision sensors[J]. Remote Sensing, 2021, 13(6): 1064.
25. 程建康. 视觉感知与毫米波雷达融合的防碰撞技术研究[D]. 四川:电子科技大学,2022.
26. 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.
27. 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.
28. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
29. 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.
30. Jocher G, Stoken A, Borovec J, et al. ultralytics/yolov5[J]. Github Repository, YOLOv5, 2020.
31. Jocher, G., Chaurasia, A., & Qiu, J. (2023). ultralytics/ultralytics[J]. Github Repository, YOLOv8 2023.
32. Li Y, Pei X, Huang Q, et al. Anchor-free single stage detector in remote sensing images based on multiscale dense path aggregation feature pyramid network[J]. IEEE Access, 2020, 8: 63121-63133.
33. Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.
34. Feng C, Zhong Y, Gao Y, et al. Tood: Task-aligned one-stage object detection[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021: 3490-3499.
35. Lyu C, Zhang W, Huang H, et al. RTMDet: An Empirical Study of Designing Real-Time Object Detectors[J]. arXiv preprint arXiv:2212.07784, 2022.
36. Cao Y, Chen K, Loy C C, et al. Prime sample attention in object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11583-11591.
37. Li M, Wang Y X, Ramanan D. Towards streaming perception[C]//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer International Publishing, 2020: 473-488.
38. Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[J]. arXiv preprint arXiv:1405.0312, 2014.