



3.0 本章导读

本质上讲,自动驾驶系统仍属于一类自动控制系统。对于控制系统,完成闭环是实现系统稳定的重要条件之一。在传统的车辆驾驶中,驾驶员是整个闭环系统的控制核心。自动驾驶系统的功能是替代人类驾驶员成为系统的控制核心,借此实现对车辆行驶的闭环控制。自动控制系统中的控制器可以有多种实现路径,而人工智能只是其中一种。在人工智能技术获得长足发展之前,人们利用经典控制理论来设计传统控制器,特别是在一些比较简单的工况环境中,传统控制器具备更高的效率。然而在自动驾驶应用场景中,车辆所处的道路交通路况相对更加复杂,传统控制器无法胜任。目前最优的解决方案是利用人工智能技术,通过建立基于深度学习的复杂神经网络模型,利用机器学习的方法对模型进行训练和优化,在车辆控制系统中实现模型的实时推理,借此实现车辆的自动驾驶功能。

类似于对人类驾驶员的要求(必须成年才能独立驾驶汽车上路行驶),自动驾驶系统的学习模型也需要达到必要的成熟度才能逐步替代人类驾驶员操控车辆。由于车辆驾驶所面临的复杂路况需要驾驶员具有成年人的心智和能力,而所对应的自动驾驶成熟度模型同样需要进行机器学习生命周期的大量迭代与积累。通过机器学习路径建立的模型需要使用大量的训练数据,如同人类的学习需要大量的阅读和练习一样。因此,在如图 3.1 所示自动驾驶汽车研发过程的 4 个阶段中,数据采集和数据预处理是非常重要的基础和前提。

自动驾驶系统的训练数据主要来源于各种传感器,自动驾驶汽车借助各种传感器感知周围环境,传感器包括雷达、激光雷达、摄像头、声呐和 GPS 等。这些传感器捕获对周围环境的感知信息以识别导航路径、避开障碍物并读取相关标记(例如道路中的各种交通标志)以遵守交通规则。在全球多个城市,不同的自动驾驶汽车开发团队收集数千小时的试驾数据以进行测试。持续 8 小时的测试可以创建超过 100TB 的数据,必须高效地收集、存储、分析和解释这些海量数据,以进行学习算法训练,最终生成车辆的驾驶决策。



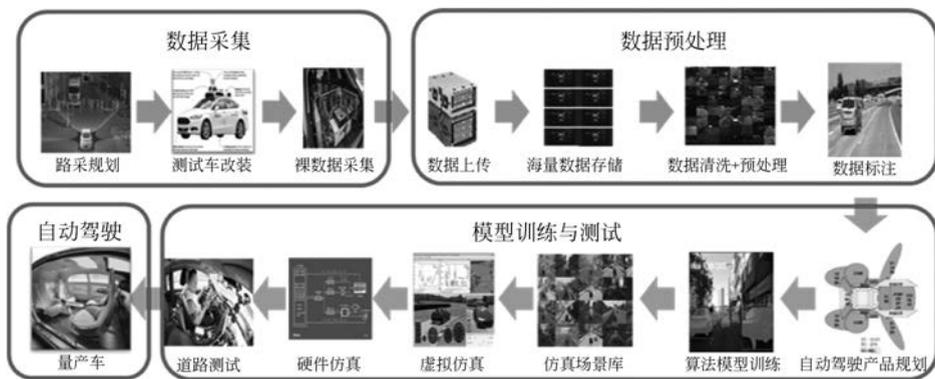


图 3.1 自动驾驶汽车研发过程

通常来说,对于自动驾驶数据集中大量存在的视频数据,会使用基于深度学习模型的计算机视觉技术实现对视频画面内容的分类、目标识别、全景分割等任务,从而实现对驾驶环境的精确感知。在这个“数据—模型—感知”的过程中,模型只能通过学习其“所见内容”(路采的图像数据)获取对环境的感知,因此高质量的图像数据对于深度学习模型应用的成功至关重要。

3.1 机器学习与数据集

3.1.1 人工智能与机器学习

人工智能这一概念最初是在 1956 年被提出的,伴随着电子计算机技术的发展和运用而产生的。由于其功能的实现多依赖于计算,以计算机作为载体运行,因此人们将其归入计算机科学类中。然而,随着研究的不断深入,人工智能涉及自然科学、社会科学和技术科学等更多的领域,被重新定义为一门典型的边缘学科。人工智能在对人的思维模拟上一直存在着两种路径:一种是对人脑的结构模拟;另一种是对人脑的功能模拟。结构模拟仿照人脑的构造和思考机制,以制造“类人脑”机器为目标;功能模拟是抛开人脑结构的特征,仅模仿人脑功能的运行过程。

功能模拟建立在计算机应用的基础上,通过编制指令程序,让计算机完成运算处理的功能,生成人们所期望的结果。功能模拟的优点在于能够运用计算机远超人脑的强大运算能力高效地完成计算功能,但局限性也很明显,其无法达到人脑所能达到的智力水平。然而,人类的智能从何而来?大脑作为智能的容器,其内部结构和功能机理已经成为研究的热点。对人类大脑结构进行模拟,随着现代技术的发展而不断提高,以结构模拟作为技术路径实现人工智能的优势也日趋凸显。

功能模拟路径利用计算机程序指令直接实现预期功能,而结构模拟路径则是通过建立“类人脑”模型,并且用机器学习的方法使模型具备预期功能。机器学习是模式识别领域中的研究热点,随着深度学习技术的发展,也成为目前实现人工智能的重要途径。机器学习相



关理论和方法在解决工程应用类和科学领域的复杂问题中得到了广泛的应用。机器学习有两个研究方向：一个是传统的机器学习方向，注重研究模拟人的学习机制；另一个是侧重对大数据的分析和学习方向，主要研究从海量的数据中获取知识，提高对数据信息的利用率。

从不同的角度出发，机器学习还有不同的分类。例如，从学习策略的角度，机器学习可以分为模拟人脑的机器学习（包括符号学习和连接学习等）和数学统计的机器学习（基于模型和算法的机器学习等）；从学习方法的角度，机器学习可以分为归纳学习、演绎性学习等；从学习方式的角度，机器学习可以分为有监督学习、无监督学习和强化学习等；从数据形式的角度，机器学习可以分为结构化学习和非结构化学习；从学习目标的角度，机器学习可以分为概念学习、规则学习、函数学习、类别学习、贝叶斯网络学习等。

机器学习常见的算法包括决策树算法、朴素贝叶斯算法、支持向量机算法、随机森林算法等。深度学习是机器学习领域中一个新的研究方向，是目前人工智能实现的重要途径，在语音和图像识别方面取得了显著成效，正在被广泛应用于多种实用场景中，例如人脸识别、智能音箱、人机交互等。在自动驾驶领域，采用深度学习方式进行研究是目前常见的研究路径，系统对环境目标进行识别和感知，以此作为决策判断的基础，借鉴人类驾驶员操控车辆的经验和判断，实现车辆驾驶的闭环控制。

3.1.2 机器学习数据集

机器学习所需的数据集是学习模型最终所蕴含的“知识”的来源。机器学习数据集被定义为训练模型和进行预测所需的数据集合。这些数据集分为结构化数据集和非结构化数据集，其中结构化数据集采用表格格式，数据集的“行”对应记录，“列”对应特征，非结构化数据集对应图像、文本、语音、音频等。

数据集通常以人工观察方式进行创建，有时也可能是在某些应用程序或算法的帮助下创建的。数据集中可用的数据可以是数字、类别、文本或时间序列。例如在预测汽车价格时，这些数据会是数字。在数据集中，每一行数据对应一个观察值或样本。

1. 数据类型

从机器学习的角度理解数据集中可用的数据，数据类型分为：

(1) 数值数据：任何作为数字的数据都称为数值数据。数值数据可以是离散的或连续的。连续数据具有给定范围内的任何值，而离散数据则具有不同的值。例如：汽车门的数量是离散的，即 2 个、4 个、6 个等；而汽车价格是连续的，如在 100 000~150 000 元，可能是 100 000 元或 125 000.5 元；数值数据可能为 int64(64 位整型数据)或 float64(64 位浮点型数据)等。

(2) 类别数据：类别数据用于表示特征，例如汽车颜色、制造日期等。它也可以是一个数值，前提是该数值表示一个类别，例如用 1 表示汽油车，用 0 表示柴油车。这里可以使用分类数据形成组，但不能对它们执行任何数学运算，例如可以将汽油车与柴油车归类于机动



车,但是这种分类数据下的 1 和 0 数据并不具有实际的计算意义。它的数据类型是一个对象。

(3) 时间序列数据:时间序列数据是在一定时间段内以固定间隔收集的一系列数值的集合。时间序列数据类型附加有一个时间字段,以便可以轻松查询数据对应的时间戳。

(4) 文本数据:文本数据就是文字,可以为机器学习模型提供某种单词、句子或段落。由于模型难以自行解释这些文字,因此通常需要借助其他自然语言处理技术(例如词频统计、文本分类或情感分析等)进行分析。

2. 数据集划分

在机器学习和深度学习中训练模型时,经常会遇到过拟合与欠拟合的经典问题。为了克服这种情况,更加真实有效地训练与衡量模型的准确性,会将数据集划分为三个不同的部分:训练数据集、验证数据集与测试数据集。每个数据集都在系统中扮演不同的角色,通常按照 60:20:20 的比例进行划分。具体如下:

(1) 训练数据集:该数据集用于训练模型,例如在神经网络模型中,这些数据集用于更新模型的权重。

(2) 验证数据集:用于防止训练过程中可能发生的过拟合,提高模型的泛化能力。使用训练中未使用的验证数据集测试模型,基于训练数据集的模型随着训练过程在准确性上的增加可以认为是模型实际性能的提高。如果基于训练数据集的模型在训练过程中表现出准确度增加而在验证数据集上准确度却在下降,这会导致基于训练数据集的高方差情形,即模型训练的过拟合。

(3) 测试数据集:大多数情况下,当尝试根据验证集的输出对模型进行更改时,会无意中让模型能够窥视到验证数据集,因此,模型有可能会在验证数据集上形成过拟合。为了克服这个问题,通常会使用一个特定的测试数据集,这个数据集在训练过程中对模型来说是不可见的,其仅用于测试模型的最终输出,以确保模型的准确性。

3. 数据质量

数据对于机器学习至关重要,不仅仅关乎数据量,也同样关乎数据质量。对于机器学习模型而言,训练结果好与坏的区别并不在于学习算法或者模型本身的好坏,而往往取决于拥有多少、多好的数据来训练模型。

数据质量有多种定义,其中最主要的两个是:①如果数据符合预期的使用目的,则数据具有高质量;②如果数据正确地代表了数据所描述的真实世界结构,那么数据就是高质量的。

数据质量取决于标注的一致性和准确率。标注的一致性指的是某个标注员的标注和其他标注员的标注一样。标注的一致性通过确保标注员的标注具有相同的准确性或者错误性防止数据标注中的随机噪声。标注的一致性是通过共识算法来衡量的。利用自动化方法与工具可以避免一致性衡量过程采用手动方式的执行,手动方式耗时且会存在安全隐患。但是需要注意的是,由于标注可能始终正确或错误,因此仅靠高一致性并不足以完全保证数据质量。

标注的准确性衡量的是标签与真实值(或“真值”,ground truth)的接近程度。真值数据是由知识专家或数据科学家标记的用来测试标注员准确性的训练数据子集。准确性是通过基准测试(benchmark)衡量的。基准测试使数据科学家能够监视数据整体质量,然后通过深入了解标注员工作的准确性调查和解决可能引起数据质量方面任何导致标注准确性下降的潜在因素。

复查是确保标注准确性的另一种方法。标注完成后,有经验的专家会抽样检查标签的准确性。复查通常通过抽查某些标签来进行,但是某些项目有时也会审查所有标签。复查通常用于识别标注过程中的低准确性与不一致,而基准测试通常用于感知标注员的表现。

基准测试往往是成本最低的质量保证选项,因为它涉及的重叠工作量最少。但是它的局限性在于仅仅能够捕获训练数据集的子集。共识和复查的成本则取决于共识设置和审查比例(两者都可以设置为数据集 0~100%的任意值,并且同时分配给多个标注员)。

3.1.3 多种数据类型的数据标注

数据标注也称为数据标记,是用类别标签标注数据集的过程^[2]。数据标注是对未处理的初级数据,包括语音、图片、文本、视频等进行加工处理,并转换为机器可识别信息的过程。此过程的质量对于监督机器学习算法至关重要,监督学习算法在尝试通过识别未标记数据集中的相同模式预测标签之前,需要先从大量已标记数据中学习模式。常见的数据标注类型有文本标注、音频标注和图像标注。这些经标注的训练数据集可用于训练自动驾驶、聊天机器人、翻译系统、智能客服和搜索引擎等。

2007年,斯坦福大学教授李飞飞等启动 ImageNet 项目,该项目主要借助亚马逊劳务众包平台(Amazon Mechanical Turk,AMT)来完成图像的分类和标注,以便为机器学习算法提供更好的数据集。截至2010年,已有来自167个国家的4万多名工作者提供了14 197 122张标记过的图像,共分成21 841种类别。从2010年到2017年,ImageNet项目每年举办一次大规模的计算机视觉识别挑战赛,各参赛团队通过编写算法正确分类、检测和定位物体及场景。ImageNet项目的成功,促使大众理解了数据对于人工智能研究的核心作用。

不同的数据标注类型适用于不同的标注场景,不同的标注场景也针对的是不同的AI应用场景。

1. 文本标注

文本数据是最常用的数据类型。根据2020年度《AI和机器学习全景报告》,70%的公司进行数据标注时离不开文本标注。文本标注包括情绪标注、意图标注等,具体如下:

(1) 情绪标注:情绪分析包括评估态度、情绪和观点,因此拥有正确的训练数据非常重要。为了获得这些数据,经常需要人工标注者,因为他们可以评估所有网络平台(包括社交媒体和电子商务网站)上用户的情绪和评论内容,并能够标记和报告亵渎、敏感的关键字或新词。



微课视频 34

(2) 意图标注: 随着人们越来越多地进行人机交互, 机器必须能够理解人类自然语言 and 用户意图。根据多种意图数据收集, 可将意图划分为若干关键类别, 包括请求、命令、预订、推荐和确认。

(3) 语义标注: 语义标注既可以改进产品列表, 又可以确保客户找到想要的产品。这有助于把浏览者转化为买家。语义标注服务通过标记产品标题和搜索查询中的各个组件, 帮助训练算法, 以识别各组成部分, 提高总体搜索相关性。

(4) 命名实体标注: 命名实体识别(NER)系统需要大量手工标注的训练数据。一些企业在用例中会应用命名实体标注功能, 例如帮助电子商务客户识别和标记一系列关键描述符, 或帮助社交媒体公司标记实体, 例如人员、地点、公司、组织和标题, 以帮助他们更好地定位广告内容。

2. 音频标注

音频标注是对语音数据的转录和加时间戳, 包括特定语音和语调的转录, 以及语言、方言和说话者人口统计数据识别。各用例都各不相同, 有些用例需要一个非常具体的方法, 例如攻击性的语音指示器以及安全和紧急热线技术应用中标记玻璃破碎等非语音声音等。

3. 图像标注

图像标注在应用中至关重要, 包括计算机视觉、机器人视觉、面部识别以及依赖机器学习来解释图像的解决方案。要训练这些方案, 必须以标识符、标题或关键字的形式为图像分配元数据。图像标注除了标注图像外, 也包括标注视频, 因为视频也是由连续播放的图像所组成的。图像标注一般要求标注人员使用不同颜色来对不同的目标标记物进行轮廓识别, 然后给相应的轮廓打上标签, 用标签来概述轮廓内的内容, 以便让算法模型能够识别图像中的不同标记物。

常见的数据标注方式则包括分类标注、拉框标注、区域标注、锚点标注等, 具体说明如下:

(1) 分类标注: 分类标注是从给定的标签集中选择合适的标签分配给被标注的对象。通常一张图可以有多种分类标签, 例如运动、读书、购物、旅行等。对于文字, 又可以标注出主语、谓语、宾语, 或者标注出名词和动词等。此项任务适用于文本、图像、语音、视频等不同的标注对象。

(2) 拉框标注: 拉框标注就是从图像中框选出要检测的对象, 此方法仅适用于图像标注。拉框标注一般为多边形拉框。多边形拉框是将被标注元素的轮廓以多边形的方式勾勒出来, 不同的被标注元素有不同的轮廓, 除了同样需要添加单级或多级标签以外, 多边形标注还有可能涉及物体遮挡的逻辑关系, 从而实现细线条的种类识别。其中四边形拉框主要是用特定软件对图像中需要处理的元素(例如人、车、动物等)进行一个拉框处理, 同时用一个或多个独立的标签来代表一个或多个需要处理的元素。

(3) 区域标注: 与拉框标注相比, 区域标注的要求更加精确, 而且边缘可以是柔性的, 并仅限于图像标注, 其主要的应用场景包括自动驾驶中的道路识别和地图识别等。

(4) 锚点标注：锚点标注是指将需要标注的元素(例如人脸、肢体)按照需求位置进行点位标识,从而实现特定部位关键点的识别。例如,采用锚点标注的方法对图示人物的骨骼关节进行标识。锚点标注的适用场景包括姿态识别、人脸识别、手势识别等。

3.1.4 高维数据可视化技术

高维数据是一种十分常见的数据类型。其数据样本拥有多个属性,如何高效地分析这类数据,对分析人员来说始终是一个巨大的挑战。其中的关键在于,如何同时展示多个属性并挖掘它们之间的联系,这在数据拥有成百上千维度时会变得尤为困难。

过去的数十年中,在可视化领域已经产生了大量优秀的技术,如数据降维、散点图矩阵、平行坐标等,以帮助用户分析这类数据。但这些技术都还有很大的完善空间,而且研究和应用领域中也存在着尚未发掘的潜力与空缺。

1. 数据降维

数据降维(dimensionality reduction)是把高维数据转化为二维或三维数据,从而可以通过散点图等展示方法对数据进行分析。降维的目标是尽可能多地在低维空间保留高维数据的关键结构。传统降维方法如 PCA(principal component analysis,主成分分析)算法是线性方法,在降维过程中主要保证了不相似的点尽量远离的结构特征。但是当高维数据处于低维非线性流形上时,保证相似的点尽量接近则变得更为重要。流形数据是像绳结一样的数据,虽然在高维空间中可分,但是在人眼所看到的低维空间中,绳结中的绳子是互相重叠的,是不可分的。

随机近邻嵌入(stochastic neighbor embedding, SNE)算法把高维数据点之间的欧几里得距离转化为表示相似度的条件概率。SNE 算法是通过仿射变换(affine transformation)将数据点映射到概率分布上,主要包括两个步骤:

(1) 通过 SNE 算法构建一个高维对象之间的概率分布,使得相似的对象有更高的概率被选择,而不相似的对象有较低的概率被选择。

(2) 通过 SNE 算法在低维空间里构建这些点的概率分布,使得这两个概率分布之间尽可能相似。

尽管 SNE 已经可以给出较好的数据可视化,但它仍然受限于优化问题自身的难度和“拥挤问题”(crowding problem)。拥挤问题是指各个簇聚集在一起,无法区分。例如存在这样一种情形,高维度数据降维到十维时,可以有很好的表达,但是降维到二维后却无法得到可信映射,例如降维到十维中有 11 个点之间两两等距离,在二维下就无法得到可信的映射结果(最多 3 个点)。

t -SNE(t -distributed stochastic neighbor embedding, t 分布随机近邻嵌入)算法是对 SNE 算法的改进,由 Laurens van der Maaten 和 Geoffrey Hinton 在 2008 年提出,在低维空间下使用更重长尾分布的 t 分布来避免拥挤问题和优化问题。 t -SNE 算法与 SNE 算法的不同点主要在于:

(1) 使用对称版的 SNE 算法,简化梯度公式。



(2) 低维空间下,使用 t 分布替代高斯分布表达两点之间的相似度。

t -SNE 算法将样本点间的相似度关系转化为概率:在原始空间(高维空间)中转化为基于高斯分布的概率;在嵌入空间(二维空间)中转化为基于 t 分布的概率。这使 t -SNE 算法不仅可以关注局部(SNE 算法只关注相邻点之间的相似度映射而忽略了全局之间的相似度映射,可视化后的边界不明显),还可以关注全局,可视化效果更好(簇内不会过于集中,簇间边界明显)。

如图 3.2 所示, t -SNE 算法非常适用于高维数据降维到二维或者三维,进行可视化。当想要对高维数据进行分类,又不清楚这个数据集有没有很好的可分性(即同类之间间隔小,异类之间间隔大)时,可以通过 t -SNE 算法投影到二维或者三维空间中进行观察。如果在低维空间中具有可分性,则数据是可分的;如果在高维空间中不具有可分性,可能是数据不可分,也可能仅仅是因为不能投影到低维空间。

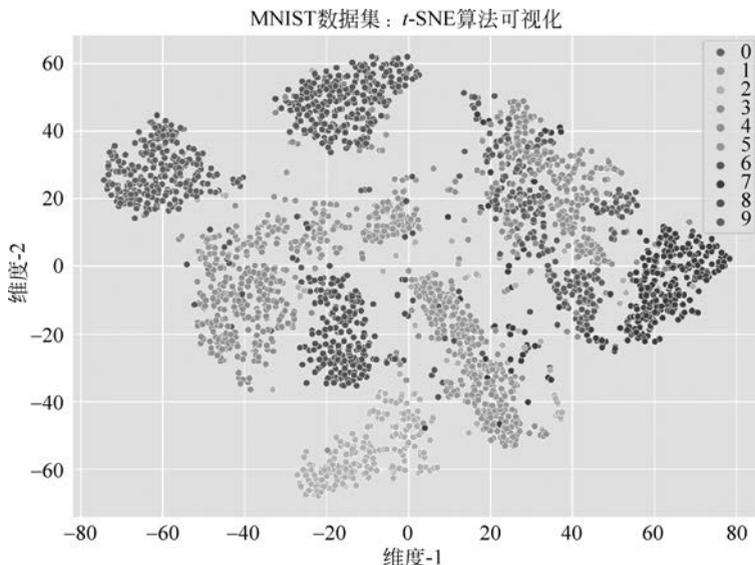


图 3.2 MNIST 数据集在二维平面中的聚类图

通过原始空间和嵌入空间的联合概率的 K-L 散度(Kullback-Leibler Divergence)评估可视化效果的好坏,也就是说用有关 K-L 散度的函数作为损失函数,然后通过梯度下降最小化损失函数,最终获得收敛结果。需要注意的是,该损失函数不是凸函数,即具有不同初始值的多次运行将收敛于 K-L 散度函数的局部最小值中,以获得不同的结果。因此,可尝试不同的随机数种子(在 Python 中通过设置种子获得不同的随机分布),并选择具有最低 K-L 散度值的结果。

K-L 散度是一种量化两种概率分布 P 和 Q 之间差异的方法,也被称为相对熵。在概率学和统计学上,经常会使用一种更简单的、近似的分布替代观察数据或太复杂的分布。K-L 散度能帮助度量使用一个分布来近似另一个分布时所损失的信息量。

对称 SNE 算法实际上是在高维度中的另一种减轻“拥挤问题”的方法:在高维空间中

使用高斯分布将距离转换为概率分布,在低维空间中使用长尾分布将距离转换为概率分布,使得高维度中的距离在映射后能够有一个较大的距离。

t -SNE 算法的梯度更新有两大优势:

- (1) 对于不相似的点,用一个较小的距离会产生较大的梯度让这些点排斥。
- (2) 这种排斥又不会无限大,避免不相似的点距离太远。

值得注意的是:未能在二维空间中用 t -SNE 显示良好分离的均匀标记的组不一定意味着数据不能被监督模型正确分类,还可能是因为二维空间不足以准确地表示数据的内部结构。

t -SNE 算法的不足之处包括:

(1) 主要用于可视化,很难用于其他目的。例如测试集合降维,因为没有显式的预估部分,不能在测试集合直接降维;又例如降维到十维,因为 t 分布偏重长尾分布,1 个自由度的 t 分布很难保存好局部特征,可能需要设置成更高的自由度。

(2) t -SNE 算法倾向于保存局部特征,对于本征维数(intrinsic dimensionality)本身就很高的数据集,不可能完整地映射到二维或者三维空间。

(3) t -SNE 算法没有唯一最优解,且没有预估部分。如果想要做预估,可以考虑降维之后,再构建一个回归方程之类的模型去做。但是要注意, t -SNE 中的距离本身是没有意义的,都是概率分布问题。

(4) t -SNE 算法的计算复杂度很高,在数百万个样本数据集中可能需要几小时,而 PCA 算法可以在几秒钟或几分钟内完成。

(5) 算法是随机的,具有不同种子的多次实验可以产生不同的结果。虽然通常倾向于选择损失最小的结果,但可能需要多次实验以选择合适的超参数设置。

(6) 全局结构未明确保留。

2. 散点图矩阵

如图 3.3 所示,散点图矩阵是对散点图的扩展。对于 N 维数据,采用 N^2 个散点图逐一表示 N 个属性之间的两两关系。这些散点图根据它们所表示的属性,沿横轴和纵轴按一定的顺序排列,从而组成一个 $N \times N$ 的矩阵。关于散点图矩阵的对角线位置,可以替换为对应属性的直方图或相关形式表示。

散点图矩阵方法存在的问题是:当维度较多时,散点图矩阵数量呈几何级数增长,难以有效发现其中的规律或模式。有限展示重要的散点图可以在一定程度上缓解空间的局限,目前已经有致力于自动化识别有价值散点图的研究,例如采用分类模式、相关模式将散点图进行针对性的高亮展示。

3. 平行坐标

如图 3.4 所示,平行坐标是一种经典的高维数据可视化技术。它将多个维度的坐标轴并列摆放,并利用穿过各轴的折线来表示数据点的取值。因其形式的紧凑性和表达的高效性,平行坐标被广泛应用在各学科、各行业的数据分析中。然而,这种形式也存在缺陷,如容易产生视图遮挡、交互不方便等问题。

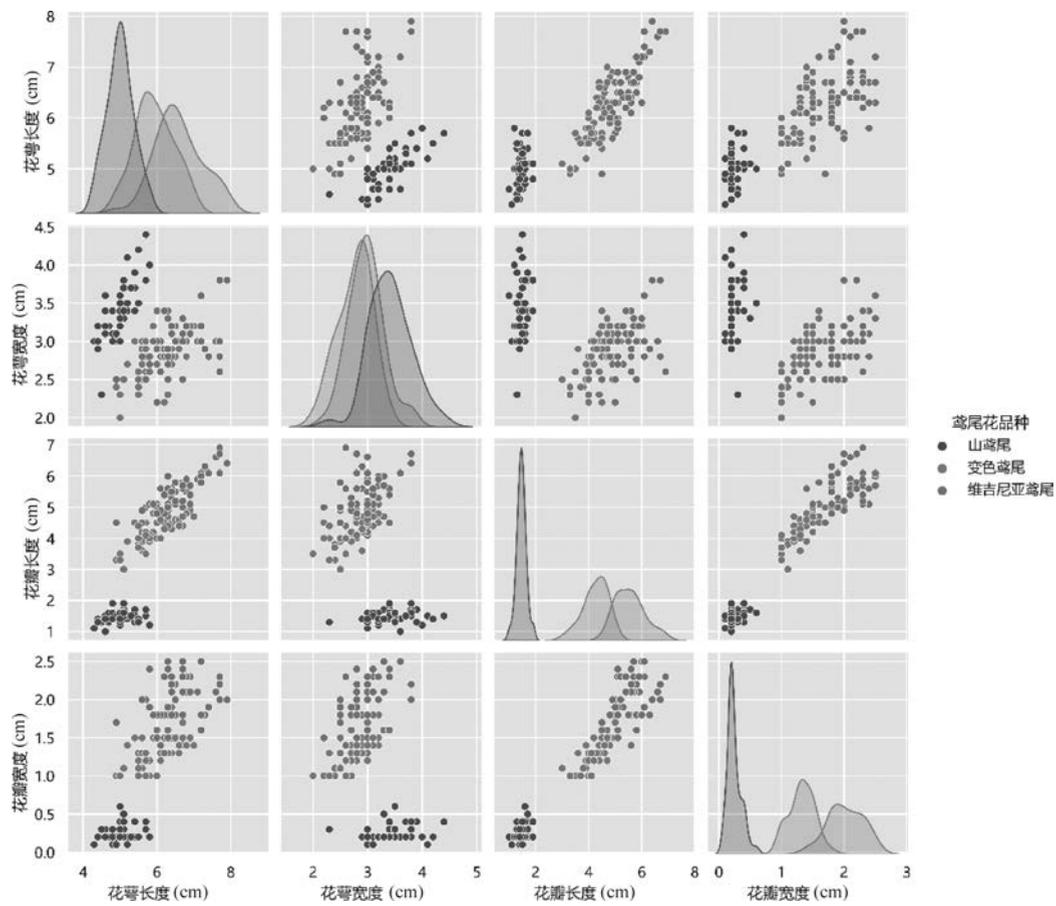


图 3.3 散点图矩阵

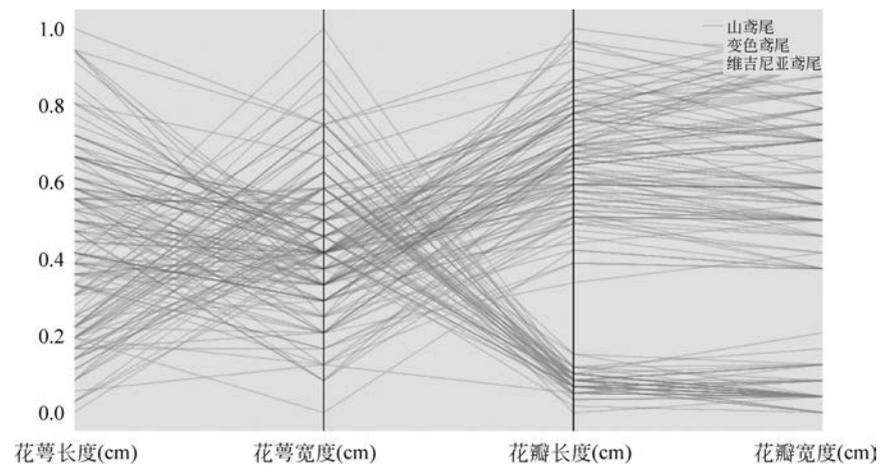


图 3.4 鸢尾花数据集的平行坐标

4. 子空间分析

在高维数据中，一部分数据称作一个子集，而一部分维度称作一个子空间。很多数据特征（例如数据结构、维度相关性等）会在不同的子空间里呈现不同的面貌。然而，这些子空间的特征往往隐藏很深，无法通过全局的数据分析发现。用户需要深入探索各个子空间来发掘其中隐含的信息。

5. 交互式的数据可视化

可视化在数据分析中起着重要的作用。设计良好的交互式的数据可视化定制工具可以帮助用户无须编程，即可通过单击、拖曳等简单的交互手段，快速地构建数据的可视化。此外，用户还可以创造各种新颖的可视化形式，并与其他用户分享、交流。

3.2 自动驾驶数据收集与处理

预测能力是人类智能的重要组成部分。人们在开车时，总是会观察周围环境的动向，以识别潜在风险并做出更安全的决策。从数据的角度来说，人类的感知系统将驾驶环境视为连续的以视觉为主的信息流。

自动驾驶技术也需要解决“感知”和“决策”两方面的问题。感知是使用多种技术融合的传感器，通常包括摄像头、激光雷达、毫米波雷达、GPS/IMU 等设备感知路面、车辆和行人；决策则是使用感知到的信息判断应该如何行动。所以有效的感知是做出可靠决策的前提。

目前解决感知问题的主要方式就是“利用海量数据训练深度学习模型”，通过监督学习的方法将大量各种情况的训练数据提供给深度学习算法，让生成的模型具备感知能力。基于自动驾驶技术需求对海量原始数据进行框选、提取、分类等一系列处理，将异构数据转化为监督学习算法可识别的机器学习数据集，帮助自动驾驶系统更好地感知实际道路、车辆位置和障碍物等信息，实时感知在途风险，实现智能行车、自动泊车等预定目标。

那么，如何训练自动驾驶汽车更深入、清晰和精准地感知与理解周围的世界？机器学习模型能否从过去的经验中学习以识别帮助它们安全地应对新的和不可预测的情况的未来模式？要想回答这样的问题，自动驾驶领域的开放数据集必然是推进未来解决方案的核心要素之一。

迄今为止，学术界与工业界提供的自动驾驶数据主要由大量静态单一图像组成，通过使用“边界框”，这些图像可用于识别和跟踪道路上和道路周围发现的常见物体，例如自行车、行人或交通灯。

相比之下，大量关于驾驶场景的连续视频数据也正在开放出来，这些数据中包含许多常见道路对象的更精确的像素级表示，为基于视频驾驶场景感知提供了动态、真实驾驶情况的数据流。基于这种连续数据类型的全场景分割对于识别更多无定形对象（例如道路建设和植被）特别有帮助，因为这些对象并不总是具有明确和统一的形状。它还允许研究

人员探索随时间推移的数据模式,这可能会带来潜在的机器学习、场景理解和行为预测的进步。



微课视频 36

3.2.1 自动驾驶数据特征

在执行自动驾驶的方案中,无论是测试阶段还是实际运行阶段都会产生并使用大量的多种类型的复杂数据。在测试阶段,需要使用大量的测试数据验证自动驾驶的功能,并对自动驾驶未来的功能进行预研。在对数据进行标注后,感知和决策模型开始利用数据进行训练,同时提取自动驾驶场景数据构建虚拟仿真模型以提升车辆的自动驾驶能力,保证自动驾驶车辆的安全性和鲁棒性。在实际运行阶段,自动驾驶车辆的正常运行不仅依赖于车端传感器采集的大量数据,同时也依赖于高精度地图数据、实时交通数据、天气数据等,而自动驾驶车辆在运行过程中也会产生或接收大量的有关车辆的数据、控制数据、用户驾驶数据等。

自动驾驶数据集一般需要满足几方面的要求:规模性、多样性、在道路上获取并包含时间信息。其中数据的多样性对于验证感知算法的鲁棒性十分关键。自动驾驶在测试和实际运行过程中产生的数据具有多样性,包括感知数据、决策与控制数据、测试与仿真数据以及用户个人数据等。数据集的多样性也涵盖了不同的天气状况(晴天、阴天、雨天)、一天中不同的时间段的光照状况。在感知数据中主要包含自动驾驶传感器原始数据、动态交通数据、自动驾驶地图数据和车联网数据。

自动驾驶数据高度还原真实世界。自动驾驶数据不仅包含道路及其两旁的静态信息,还包括道路上动态信息,如车辆、行人、交通信号等,以及部分敏感的地理信息,诸如军事设施、核设施、港口、电力设施等。

自动驾驶数据包含用户个人数据。例如用户操作习惯数据和驾驶习惯数据(包括行程轨迹,用户导航、历史及即时地理位置等),此外还可能包括用户虹膜、指纹、声纹等生物特征数据。

自动驾驶数据与车联网数据存在众多差异,因此需要针对自动驾驶数据的特点进行分级分类以全面考量其安全性及保障手段。综合考虑自动驾驶的人工智能属性以及自动驾驶数据的多样性、规模性、非结构性、流动性、涉密性的特点。除此之外,自动驾驶车辆还具有车本身的安全属性和智能网联下跨产业技术融合的特点^[3]。

(1) 数据的多样性:根据不同自动驾驶级别,数据来源的不同,数据不仅包括汽车基础数据(车牌号、车辆品牌和型号、车辆识别码、车辆颜色、车身长度和宽度、外观等相关数据),也包括基础设施数据、交通数据、地理信息数据(红绿灯信息、道路相关基础设施、道路行人的具体位置、行驶和运动的方向、车外街景、交通标志、建筑外观等真实交通数据),以及车主的用户身份类数据(姓名、手机号码、驾照、证件号码、支付信息、家庭住址、用户的指纹、面部等生物特征信息等)、用户状态数据(语音、手势、眼球位置变化等)、行为类数据(登录、浏览、搜索、交易等操作信息等)等。

(2) 数据的规模性:自动驾驶车辆作为跨产业技术的融合载体,融合了来自汽车、道路、天气、用户、智能计算系统等多方面的海量数据,涉及数据类型多,需要统计分析的数据总量大。

(3) 数据的非结构性：数据多样性决定了不同来源的数据格式不同，数据的非结构性和非标准性对数据聚合或拆分技术以及权限管理和安全存储都带来了巨大的挑战。

(4) 数据的流动性：大量自动驾驶数据在用户端、车端、云端等多场景的交互使得数据的流动性增大。除此之外，自动驾驶数据还具有跨行业共享交换的特点。因此，如何确保交互数据的安全性，是一个亟待解决的问题。

(5) 数据的涉密性：自动驾驶汽车在公开道路驾驶过程中，会采集大量地理信息数据，根据中国法律法规要求，采集地理信息数据可能涉及涉密测绘成果，因此需要按照《中华人民共和国保守国家秘密法》中的相关规定要求进行分级管理。

3.2.2 自动驾驶传感器数据

为了实现技术方案的可靠性与鲁棒性，自动驾驶汽车通常使用多种类型的传感器。这些传感器有多种分类方法，按照测量数据的来源可以分为两大类：外感受传感器(proprioceptive sensor)和本体感受传感器(exteroceptive sensor)。

在自动驾驶的技术场景中，外感受传感器是用来观察环境的，包括道路、建筑物、汽车、行人等。自动驾驶汽车最常见的外感受传感器是摄像头和传感器。如图 3.5 所示，为了观察和感知自身周围的一切，自动驾驶汽车通常使用三种类型的外感受传感器：摄像头、雷达和激光雷达。这些传感器的原始数据类型包括点云、视频、照片、高精度定位坐标等。

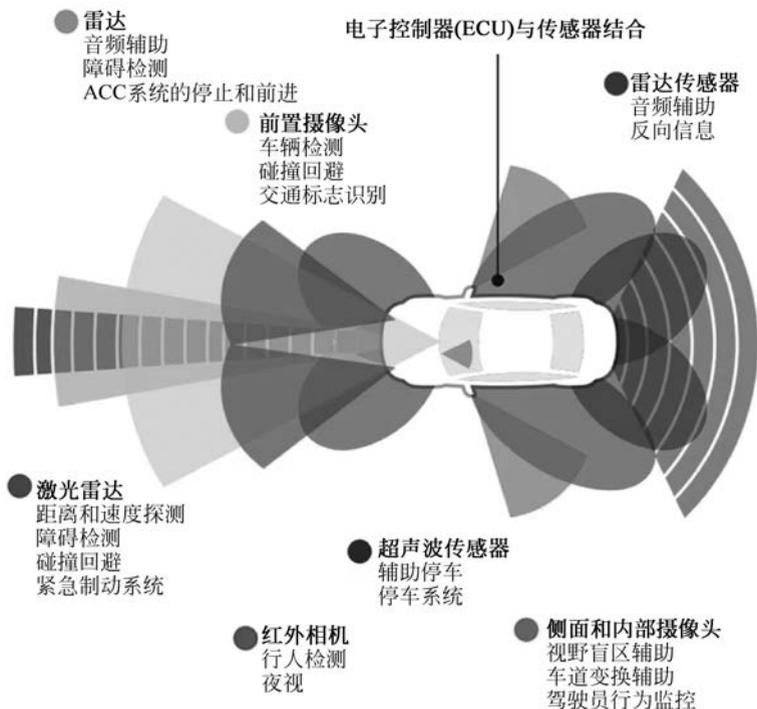


图 3.5 自动驾驶汽车传感器探测范围示意图



微课视频 37

本体感受传感器测量给定系统的内部值。大多数现代汽车已经配备了非常多的本体感受传感器,例如车轮编码器用于里程测量,转速计用于监测加速度变化。这些传感器通常可以通过车辆 CAN 总线访问。

1. 摄像头数据

摄像头可帮助车辆获得周围环境的 360°全景影像。不仅如此,如图 3.6 所示,现代摄像头还可以提供逼真的 3D(三维)图像,识别物体和人,并确定它们之间的距离。由于车辆行驶时,摄像头一直都会进行拍摄,其实就相当于在录制高清电影。根据该高清电影数据,可以基于位置、颜色等识别对象,或者使用来自多个摄像机的视角差异计算出距离并识别障碍物、他车、行人等。



图 3.6 KITTI 数据集中的单目摄像头图像示例

(图片来源: KITTI 数据集官方网站 <http://www.cvlibs.net/datasets/kitti/index.php>)

摄像头属于被动传感器,这意味着它们不需要发出信号捕捉信息,从而限制了与其他传感器的可能干扰。由于被动性质,它们会受到光照和天气条件的负面影响。如图 3.7 所示,为了弥补夜间或低照度下糟糕的摄像性能,红外相机常被用于行人检测等任务。



图 3.7 KAIST 数据集中的红外相机图像示例

(图片来源: KAIST 数据集官方网站 <http://multispectral.kaist.ac.kr/pedestrian/data-kaist/images/set00.zip>)

另一种令人感兴趣的相机是事件相机,如图 3.8 所示,它输出像素级别的亮度变化,而不是标准的亮度帧。它们提供了出色的动态范围和非常低的时延,这在高度动态的场景中非常有用。然而,大多数已经开发的视觉算法并不容易应用到这些相机上,因为它们输出的是异步事件序列,而不是传统的光亮强度图像。



图 3.8 DAVIS 数据集中的事件相机图像示例

(图片来源：DAVIS 数据集 <https://davischallenge.org/>)

极化传感器图像示例如图 3.9 所示。极化传感器——索尼 Pregius 5.0 MP IMX250 传感器也达到了更好的性能,可以提供更多的细节。极化传感器的偏振通道受光照变化和天气的影响较小。它们对表面粗糙度也很敏感,这有助于车辆的检测。然而,目前暂时还没有发布使用偏振相机的公共自动驾驶数据集。



图 3.9 极化传感器图像示例

(图片来源：DAVIS 数据集 <https://davischallenge.org/>)

2. 激光雷达数据

激光雷达使用激光代替无线电波,并且可以创建周围环境的 3D 图像并绘制地图,具有很强的空间覆盖能力,能在汽车周围创建 360°的视图。激光雷达擅长距离测量,通过照射激光束并测量其从物体返回所需的时间来测量到物体的距离和方向等。

一个激光束会带回一个“点”数据,通过发射无数个激光束会返回无数个点,从而形成一组“点云”数据。对于每个点,根据激光发射和返回所花费的时间测量到物体的距离,确定距离和物体的形状。

通过不断实时生成数据,激光雷达除了测量识别到周围障碍物的距离和道路形状外,也可以通过计算表面的反射率来识别道路上的白线等。

激光雷达比雷达准确得多,但由于雾、雨或雪等天气条件的影响,它们的性能会下降。它们有时在近距离探测物体时也有困难。

3. 雷达数据

雷达旨在检测移动物体,实时测量距离和速度。天气条件不会影响短程和远程雷达。短程波有助于消除盲点,并有助于车道保持和停车。远程雷达可以测量汽车与其他行驶中车辆之间的距离,并有助于制动。

为了减轻激光雷达在恶劣天气或近距离传感方面的局限性,雷达也被用作距离传感技术。作为一种比激光雷达更成熟的传感器,雷达通常更便宜、更轻,同时也能确定目标的速度。然而,它们的空间分辨率很低,难以解释接收到的信号,而且精度比激光雷达差得多。

4. 位置信息数据

位置信息数据包括卫星定位系统数据和每秒更新的动态地图数据。卫星定位数据是通过诸如 GPS 之类的全球卫星导航系统(GNSS)获得的,全球卫星导航系统依靠从人造卫星发射的信号测量自动驾驶车辆的位置信息。

与卫星系统配合使用的高精度地图也在不断发展。在汽车导航系统上显示的常规地图是 2D 平面地图,主要用于为驾驶员提供参考。而当前正在开发的 3D 高精度地图则覆盖了诸如各个车道的曲率和坡度之类的道路形状等许多信息,还包含各种动态信息,例如道路结构、车道信息、路面信息、交规信息、下落的物体、故障车辆和信号显示信息等。

5. 定位数据

自动驾驶一般用组合定位技术。首先本体感受传感器,如里程计(odometer)、陀螺仪(gyroscope)等,通过给定初始位姿测量相对于初始位姿的距离和方向,确定当前车辆的位姿,也称为航迹推测。然后用激光雷达或视觉感知环境,用主动或被动标识、地图匹配、GPS 或导航信标进行定位修正。位置的计算方法包括三角测量法、三边测量法和模型匹配法等。从这个角度而言,惯性测量单元(IMU)也是自动驾驶必备的传感器。



微课视频 38

3.2.3 数据融合与车辆定位

在自动驾驶中,车辆自身所处的位置是首先需要解决的问题。通过各类传感器观测环境、感知环境,通过测量数据来获得自身的定位,是自动驾驶系统操控车辆的基础。

1. 传感器数据处理与车辆定位

车辆利用不同类别的传感器获得测量数据,例如可以利用车辆自身的状态传感器部分测量数据,包括用惯性测量单元获取车辆的运动加速度和转动角度;用车轮传感器获取车轮转速;用助力系统传感器获取方向盘角度、油门力度、制动力度等数据。这类传感器提供的是车辆内部测量数据,相关信息容易获得,能够帮助车辆实现自身定位,然而会有比较大的定位误差。为了减小这种误差,车辆还可以利用外感知传感器测量外部数据,通过摄像头、激光雷达和毫米波雷达、超声波等传感器对外部环境进行感知,通过标定环境中的特征点,对照地图等参照数据标定自身的位置,这是一类采用相对位置进行感知定位的方法。另外一种常见的方法是全球卫星定位。在全球导航卫星构建的坐标系中,定位系统接收导航

卫星提供的电信号,通过三边测量法或三角测量法确定自身的绝对位置。

车辆自身的位置可以通过其在世界坐标系中定义坐标向量的数值确定,车辆自身的姿态可以在世界坐标系中定义车体本身的方向角,由于车辆的局部移动基本属于二维运动问题,因此用坐标和方向角就可以将定位问题转化为对数学问题的求解。对车辆位置与姿态的估算相当于在世界坐标系里求解坐标数值和方向角度。利用多种传感器估计车辆的位置与方向角,可以采用贝叶斯滤波器、卡尔曼滤波器(扩展卡尔曼滤波器、无迹卡尔曼滤波器)和粒子过滤器等相关算法。

这些算法解决的是在估算车辆自身位置时精确度的问题。当车辆从起始点出发并在行进过程中,位置随时间不断变化,在每一时刻车辆自身位置的估算是否准确?是否会有测量和计算误差?这些都是自动驾驶可能会遇到的问题。造成位置估算错误的原因有很多,例如车辆自身驱动模型不精准、行驶环境干扰、传感器测量误差等。驱动模型中对于油门踏板、方向盘和制动器的建模是否准确,将直接影响车辆移动位置估算的结果。环境条件中遇到风向变化和风力大小、道路路面结冰的不同摩擦力等也会干扰位置估算结果。

例如让车辆从起始点出发,直线行驶 100 米后停住。有如下几个问题要考虑:一是需要给车辆提供多少燃油,在行进过程中的油门如何控制?二是方向盘是否能够保证车辆完全沿直线行驶,方向转动是否存在误差?三是行进过程中对风阻变化的判断是否准确?四是行进过程中路面是否会积水或结冰,路面摩擦力是否恒定?这些问题都必须解决才能完成让车辆直行 100 米后停止的任务。然而当由驾驶员操控车辆时,则根本无须特别关注上述问题,这是因为驾驶员能够主动观测道路环境(知道是否直线行进了 100 米)并调整车辆行进姿态,但是对于自动驾驶的车辆,观测环境的任务却并无驾驶员参与,而是由传感器承担。

任何传感器都有测量误差(可以把传感器测量的数据看成一个有均值的高斯分布),测量误差会随着时间的推移出现结果偏差。尤其是 IMU 传感器,它所测量的物理量并不是位置和角度,而是高阶物理量(加速度、角速度等),用这些高阶物理量需要对时间进行积分后才能够转换成位移距离和转动角度等信息。对时间积分将使传感器自身很微小的误差会随着时间的推移而急剧增大。因此对于传感器数据需要进行一定的处理才能避免估算结果出现偏差过大。

提高车辆定位的精度,可以从三方面着手处理:一是改善汽车驱动模型,通过建立更精准的汽车驱动模型估算车辆位移距离,在这一点上电动汽车控制的精度是优于燃油车的;二是可以提高传感器的测量精度,但对于惯性测量元件等传感器,仍然无法完全克服其误差随时间推移的问题;三是融合更多种类传感器的数据,例如融合全球卫星导航系统提供的定位数据,其定位测量精度(民用)可达到分米量级。对于车辆定位,有些环境因素会成为必然的干扰而无法克服,例如风力风向、路面摩擦力不均匀等,目前通过使用路网协同等技术能够尽量提前预知部分环境因素的变化,对车辆的位置估算进行补偿修正。

通过算法处理,可以将车辆驱动模型输出的结果与传感器测量的数据进行融合,来获得

位置估算的最优解。贝叶斯滤波器运用贝叶斯条件概率方程,提高位置估算的准确度;卡尔曼滤波器通过观察方程和状态方程动态权衡测量值与状态输出值,从而得出最(次)优解。由于卡尔曼滤波器的使用范围是针对线性模型,其分布假设符合高斯分布,虽然其计算成本比较低,但是由于车辆行驶中以非线性因素为主,因此卡尔曼滤波器会出现较大的误差。为此,可以采用扩展卡尔曼滤波器和无迹卡尔曼滤波器进行改善,对于非高斯分布的模型还可以采用粒子滤波器进行计算。

除了上述处理方法之外,利用环境感知的定位方式也能有效改善因车辆内部传感器精度所带来的定位准确度的问题。环境感知的定位需要融合处理多种感知传感器的数据,例如可以利用激光雷达的点云阵建立车辆周边的局部环境结构,融合摄像头等视觉传感器对环境的语义分析理解局部环境中标志信息的含义以及识别道路、判定目标及障碍物等,再对照高精度地图寻找到匹配的坐标位置,借此提高车辆自身的定位准确度。

2. 地图表示与多传感器数据融合

车辆定位需要对自身位置进行估算并确定与环境特征的相对关系(方位和距离)。环境信息无法直接使用,需要通过地图进行表达,地图表示法是实现自动驾驶的重要基础之一。地图信息包含多种传感器的数据和这些数据的融合。地图表示分为全局地图和局部地图:全局地图主要用于做全局规划,确定车辆出发位置和目的地位置,关注的是路径和距离;局部地图关注的是通行环境和行驶安全。

地图表示法包括格栅占用地图、特征地图、语义地图以及高精度地图表示法等。格栅占用地图表示法比较简单,常见的是激光雷达所形成的二维点云图。格栅占用地图表示法的优点是:较直观易懂,能够直接用于路径规划,创建比较容易。它的缺点是:无法进行三维显示,定位精度很有限,离散分辨率固定等。如果需要生成三维地图,可以使用三维激光雷达,但三维激光雷达成本较高,定位精度有限,测量的数据量很大,会包含很多无关的信息。

特征地图可以通过摄像头建立,通过在图像当中提取目标的特征信息,并将其保存在三维地图中。特征地图表示法的优点是:适合用于二维地图和三维地图,不需要进行离散化,兼容不同传感器的数据,计算时消耗的内存比较少。其缺点也同样明显:地图信息人类无法直观理解,地图上的特征含义,人类很难直观解读,缺少位置占据信息,无法判断哪些是可通行的路径。

语义地图表示法通常与其他地图表示法结合使用,包含对环境中路径、障碍物、交通标识等的语义信息。语义地图的优点是:包含语义信息,能够用不同的对象分类来替代抽象的特征,并且能够附加路径规划的相关信息。其缺点是:需要保留更多关于周围环境的信息,语义信息判断依赖于对目标的识别和检测,需要占用大量的计算资源,需要有足够的计算处理时间。

高精度地图是离线生成的可用于在线定位的复杂地图,它是由多种不同信息组合而成的,包含了大量传感器的探测数据和环境先验信息,例如道路和路径的详情、限速标识、交通灯等交通标识信息,限号等交通政策法规的附加信息。高精度地图表示法的优点是:代表了多传感器的数据融合,有很高信息含量,针对各种应用可以进行完美的调整。其缺点是:创建高精

度地图的成本高,针对路况变化实时更新比较困难,使用时对所占用的内存和带宽要求较高。

在地图中,特别是高精度地图中会包含多种传感器的数据融合。例如用激光雷达形成点云图与结合摄像头通过目标检测识别形成的特征信息,可以为车辆的行驶提供精细的微观环境感知;通过融合路网协同传感器探测的道路信息与带有卫星导航系统坐标信息的全局地图,可以为车辆的行驶提供宏观的决策规划指引。在行驶过程中,自动驾驶系统以地图中的信息数据为参照,结合自身姿态的测量数据和现场道路的感知数据综合决策,为最终实现车辆的自动驾驶奠定基础。

3.2.4 自动驾驶数据标注

无论建立地图数据还是驾驶现场对道路环境的感知,自动驾驶系统都需要预先采集大量的道路实景数据。一般而言,道路采集数据结束后,所有数据都将从车辆中提取到数据中心,并对有益的数据进行分析和标记。原始数据本身对于处理器系统核心的学习系统没有多大价值,数据对象包括行人、骑自行车的人、动物、交通信号灯等变量。在将传感器数据用于训练或测试学习模型之前,所有这些目标都需要进行手工标记和注释,以便系统可以理解其“所见”。

研究人员根据传感器的读数操作生成地图和行人姿态数据,包括3D地面反射率地图、3D点云地图、六自由度地面真实姿态和局部姿态传感器信息。这些数据要能够反映天气差异(例如晴天、下雪和多云的情况),并且涵盖多种驾驶环境(例如高速公路、立交桥、桥梁、隧道、建筑区域和植被覆盖区等)。

如今,大多数感知系统都严重依赖机器学习或深度学习算法,感知系统需要处理传感器的信号信息并尝试对车辆周围的物体进行分类。为了能够完成此任务,必须使用经过彻底标记和注释了所有道路的相关数据才能更好地发挥出数据的价值。值得注意的是:标记过程可能比原始数据收集还要耗时。

自动驾驶场景中的数据标注包括:

(1) 点云标注(激光雷达、雷达获取的数据):通过识别和跟踪场景中的对象,了解汽车前方和周围的场景。将点云数据和视频流合并到要标注的场景中,点云数据可帮助模型了解汽车周围的世界。

(2) 2D标注(包括语义分割):帮助模型精确理解来自可见光摄像头图像的信息。该信息能够在创建用于自定义本体的可扩展边界框时,或进行高清像素掩膜时为数据处理提供帮助。

(3) 视频对象和事件跟踪标注:帮助自动驾驶算法模型了解关注的对象如何随时间移动并清楚获得相应事件的时间标注。其难点在于:在多帧视频和激光雷达场景中需要保持跟踪进入或离开视线区域的对象(例如其他汽车、行人等),更困难的是需要在整个视频中无论对象进入和离开视线区域的频率如何,都要保持对其特性的一致性理解(不能发生混淆)。

为了保证驾驶安全,自动驾驶训练数据的要求非常严格,需要达到高质量、高量级、高效率等要求。在标注过程中,人工智能辅助的标注平台往往扮演了更为核心的角色,辅助平台进行人工智能的预标注,大大降低了人工的耗时,并且可以进行高质量快速的质检,这是纯





微课视频 40

人工标注所无法达成的。

3.2.5 自动驾驶公开数据集

虽然由自动驾驶测试(路采)生成的所有数据对于车辆感知其周围环境都是非常有价值的,但实际上只有其中的特定部分对研发和改进自动驾驶系统有用。例如在典型城市街道上—天的测试中,重复经过的道路所采集的数据对于整体改善自动驾驶系统性能并没有很大的帮助,所以车辆中的工程师和技术人员会有选择性地记录发生细微变化或重新寻找具有挑战性的场景。这说明自动驾驶数据的采集需要更加多样化和精细化,这样形成的数据集对相关从业人员来说才更有使用的价值^[4]。本节将为读者介绍一些比较有价值的典型公开数据集。

1. KITTI 数据集

KITTI 数据集是一个用于自动驾驶场景的计算机视觉算法测评数据集,由德国卡尔斯鲁厄理工学院(KIT)和丰田工业大学芝加哥分校(TTIC)共同创立。数据集包括:

- (1) 立体图像和光流图: 389 对。
- (2) 视觉测距序列: 39.2km。
- (3) 3D 标注物体的图像: 超过 20 万个。
- (4) 采样频率: 10Hz。

如图 3.10 所示,KITTI 数据集的数据采集平台装配有 1 个惯性导航系统、1 个 64 线 3D 激光雷达、2 个灰度摄像机、2 个彩色摄像机以及 4 个光学镜头。具体的传感器参数如下:

- (1) 全球定位及惯性导航系统(GPS/IMU): OXTS RT 3003×1(开阔环境下定位误差小于 5 cm)。
- (2) 3D 64 线激光雷达: Velodyne HDL-64E ×1(10Hz,64 个激光束,范围为 100m)。
- (3) 灰度摄像机: Point Grey Flea 2(FL2-14S3M-C)×2(10Hz,分辨率为 1392×512 像素,opening: 90×35)。
- (4) 彩色摄像机: Point Grey Flea 2(FL2-14S3C-C)×2(10Hz,分辨率为 1392×512 像素,opening: 90×35)。
- (5) 光学镜头(4~8mm): Edmund Optics NT59-917×4。

KITTI 数据集主要包括以下基准数据集:

(1) 立体评估(stereo evaluation): 基于图像的立体视觉和 3D 重建,从一个图像中恢复物体的 3D 结构,只能得到模糊的结果,此时通常需要从不同角度的多张图片恢复图像中物体的 3D 结构。这对自动驾驶的场景应用是非常有帮助的,例如可以得到汽车的形状和周围环境的形状等信息。

(2) 光流(flow): 光流是关于视域中的物体运动检测有关的概念,用来描述相对于观察者的运动所造成的观测目标表面或边缘的运动情况。其应用领域包括运动检测、对象分割、接触时间信息、扩展计算焦点、亮度、运动补偿编码和立体视差测量等。

(3) 场景流(scene flow): 场景流表现的是场景密集或半密集的 3D 运动场,是光流的

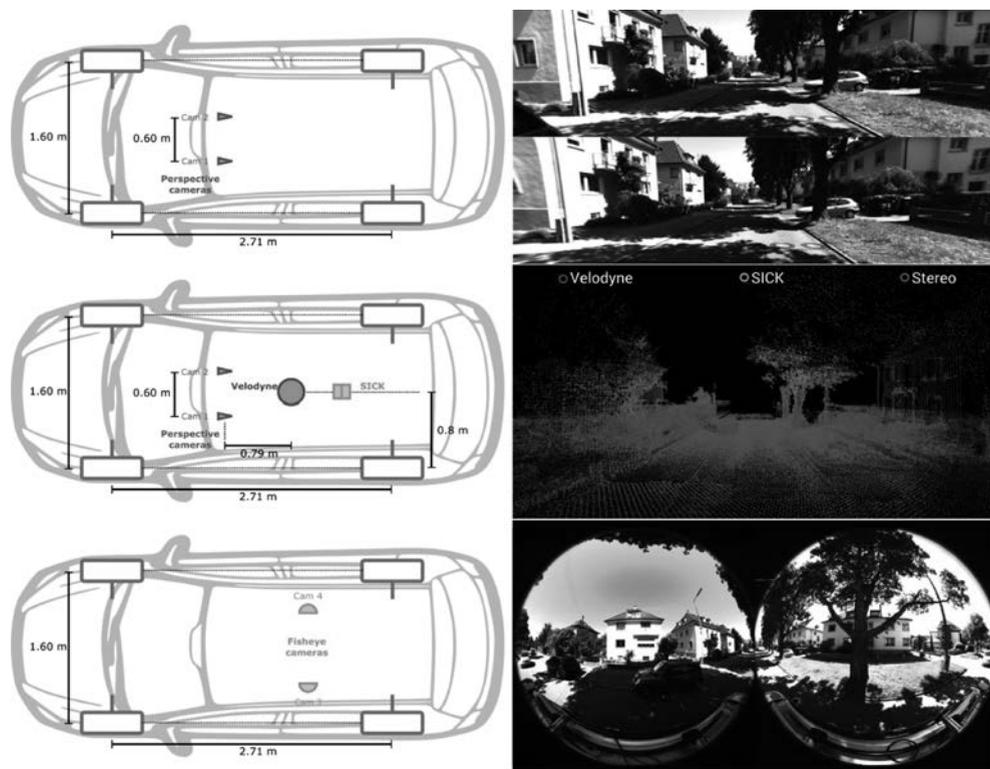


图 3.10 KITTI 数据集示例

(数据来源: <http://www.cvlibs.net/datasets/kitti-360/>)

三维版本。场景流的潜在应用很多：在机器人技术中可以用于预测周围物体的运动，在动态环境中进行自主导航或遥控；可以补充和改进最先进的视觉测距和 SLAM(simultaneous Localization and Mapping, 即时定位与地图创建)算法，这些算法通常假设在刚性或准刚性环境中工作；可以用于机器人或人机交互以及虚拟和增强现实等技术。

(4) 深度评估(depth evaluation)：视觉深度在视觉 SLAM 和里程计方面应用广泛，如果是基于视觉的里程计，那么就需要用到视觉深度评估技术，其中包括 2 项基准数据，即深度补全(depth completion)和深度预测(depth prediction)。

(5) 目标识别(object recognition)：目标识别包括 2D 场景、3D 场景和鸟瞰视角 3 种方式的基准数据，其中 2D 场景不仅要能正确标注坐标，还要能在鸟瞰视图中标注出相应的位置。2D 场景增加了除了有汽车、行人和自行车等分类，还有对它们的目标检测与方向估计。3D 场景主要是对激光雷达点云的标注，有汽车、行人、自行车等分类。

(6) 语义分割(semantic segmentation)与实例分割(instance segmentation)：语义分割对自动驾驶的信息处理非常关键，例如人会根据语义分割区分车道与周围的环境以及其他汽车，然后针对不同的场景进行决策。如果没有语义分割，系统会将所有的像素同等对待，对识别和决策都会引入不必要的干扰。实例分割侧重于检测、分割和分类对象实例，包括像

素级别的分割和实例级别的分割。

2. nuScenes 数据集

nuScenes 数据集是由 Motional 团队开发的用于无人驾驶的公共大型数据集。为了支持公众对计算机视觉和自动驾驶的研究, Motional 公开了 nuScenes 的部分数据, 其摄像头布局如图 3.11 所示。

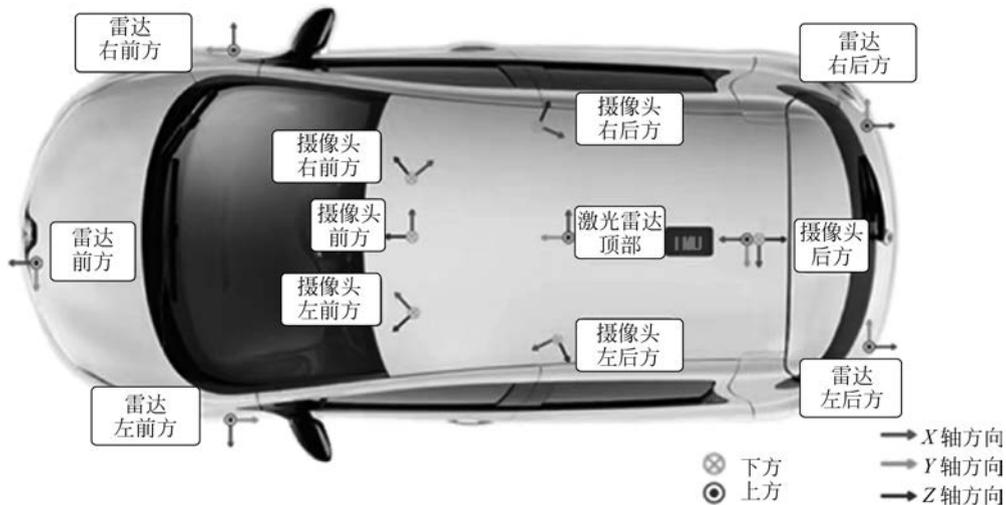


图 3.11 nuScenes 数据集的摄像头布局

(原始图片来源: <https://www.nuscenes.org/>)

nuScenes 数据集在波士顿和新加坡这两个城市收集了 1000 个驾驶场景, 这两个城市交通繁忙而且驾驶状况极具挑战性。nuScenes 手动选择 20 秒长的场景, 以显示各种驾驶操作、交通状况和意外行为。nuScenes 收集不同的数据进一步研究计算机视觉算法在不同位置、天气状况、车辆类型、植被、道路标记以及左右手交通之间的通用性。

Motional 于 2019 年 3 月发布了完整数据集, 包括约 40 万个关键帧中的 140 万个摄像头图像、39 万个 LiDAR 扫描数据、140 万个雷达扫描数据和 1.4 万个对象边界框。创建 nuScenes 数据集的灵感来自于 KITTI 数据集。nuScenes 是第一个大规模数据集, 该数据集使用的自动驾驶车辆的传感器套件包括 6 个摄像头、1 个激光雷达、5 个雷达、GPS 和 IMU, nuScenes 包含的对象注释信息是 KITTI 的 7 倍。之前发布的大多数数据集(例如 Cityscapes, Mapillary Vistas, Apolloscapes, Berkeley Deep Drive)都是基于摄像头的对象检测, 而 nuScenes 是研究整个传感器套件。

为了在激光雷达(LiDAR)和摄像头之间实现良好的跨模态数据对齐, 当顶部 LiDAR 扫过摄像头视野中心时会触发摄像头的曝光。图像的时间戳为曝光触发时间, LiDAR 扫描的时间戳是当前 LiDAR 帧完成旋转的时间。鉴于摄像头的曝光时间几乎是瞬时的, 因此这种方法通常会产生良好的数据对齐效果。但是摄像头的曝光频次为 12Hz(降低到 12Hz 是为了降低对计算带宽和存储的要求), 而 LiDAR 的扫描频率是 20Hz, 12 次的摄像头曝光需要尽

可能均匀地分布在 20 次 LiDAR 扫描中,因此并非所有 LiDAR 扫描都有相应的相机帧。

收集驾驶数据后,nuScenes 以 2Hz 的帧率采样同步良好的关键帧(同时包含图像、激光雷达、雷达数据的图像帧),发送给合作伙伴 Scale 进行标注。使用专家注释器和多个验证步骤实现了高度准确的标注。nuScenes 数据集的所有对象都带有语义类别以及它们出现的每一帧的 3D 边界框和属性等。与 2D 边界框相比,3D 边界框更能够准确推断对象在空间中的位置和方向。

2020 年 7 月,Motional 发布了 nuScenes-lidarseg。其中使用了语义标签(激光雷达语义分割)对 nuScenes 中关键帧的每个激光雷达点进行注释。因此 nuScenes-lidarseg 包含 40 000 个点云中的 14 亿个带标注的点和 1000 个场景(用于训练和验证的 850 个场景以及用于测试的 150 个场景)。

3. Waymo 数据集

如图 3.12 所示,Waymo 开放数据集是^[5]由 Waymo 公司自动驾驶汽车在各种条件下收集的高分辨率传感器数据组成。在与 KITTI、nuScenes 等数据集的对比中,其传感器配置、数据集的大小都有很大的提升。



图 3.12 Waymo 数据集示例

(图片来源: Waymo 官方网站 <https://waymo.com/open/>)

Waymo 数据集的传感器包含 5 个激光雷达和 5 个摄像头,激光雷达和摄像头的同步效果也更好。更重要的是,Waymo 数据集包含 3000 段驾驶记录,时长共 16.7 小时,平均每段长度约为 20 秒。整个数据集一共包含 60 万帧素材,共有大约 2500 万个 3D 边界框、2200 万 2D 边界框。

此外,在数据集的多样性上,Waymo Open Dataset(开放数据集)也有很大的提升,该数据集涵盖不同的天气条件,白天、夜晚不同的时间段,市中心、郊区不同地点,行人、自行车等不同道路对象等,例如:

(1) 规模和覆盖范围:数据集包含 3000 个驾驶片段,每一片段包含 20 秒的连续驾驶画面。连续镜头内容可以使研究人员能够开发模型,跟踪和预测其他道路使用者的行为。

(2) 多样化的驾驶环境：数据采集的范围涵盖凤凰城、柯克兰、山景城、旧金山等地区以及各种驾驶条件下的数据,包括白天、黑夜、黎明、黄昏、雨天和晴天。

(3) 高分辨率的特点和 360°的视图：每个分段涵盖 5 个高分辨率 Waymo 激光雷达和五个前置和侧面摄像头的的数据,这些数据可以构成 360°的环视视角图像。

(4) 密集的标签信息：车辆、行人、自行车、标识牌等图像都经过精心标注,一共有 2500 万个 3D 标签和 2200 万个 2D 标签。

(5) 摄像头与激光雷达同步：Waymo 团队致力于融合多个摄像头和激光雷达的数据生成 3D 感知模型,为此 Waymo 设计了全套的自动驾驶系统(包含硬件和软件),用以无缝地协同工作(包括选择传感器的位置和高质量的时间同步等)。

4. 毫米波雷达数据

图 3.13 为 Oxford 数据集传感器安装示意图,图 3.14 为 Oxford 雷达测量数据样例,Oxford 发布的雷达数据集 Oxford Radar Dataset^[6]收集了城市环境中所采集的多段车辆行驶数据,每段数据采集车辆行驶里程约 9km。Oxford 使用的雷达是 Navtech 开发的一款 76~77GHz 毫米波雷达,这款毫米波雷达不同于目前车载市场常用的宽波束雷达,而是采用天线阵列组成了具有波到达方向(direction of arrival,DOA)特性的窄波束雷达。雷达通过窄波束进行机械扫描可以达到类似机械式激光雷达的效果,只是相对机械式激光雷达,其分辨率较低。毫米波雷达的波束宽度仅为 1.8 度,机械扫描每次间隔 0.9 度,即每旋转一圈可以获得 400 个角度方向的测量值,机械旋转速度约 4Hz,毫米波雷达的距离分辨率为 4.32cm,最大测距为 163m。

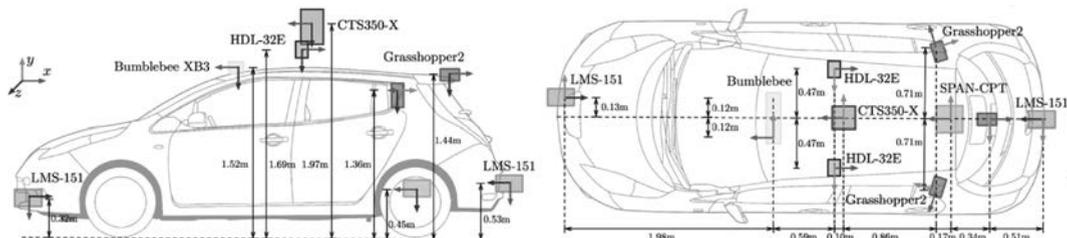


图 3.13 Oxford 数据集传感器安装示意图

(图片来源：网址 <https://oxford-robotics-institute.github.io/radar-robotcar-dataset/documentation>)

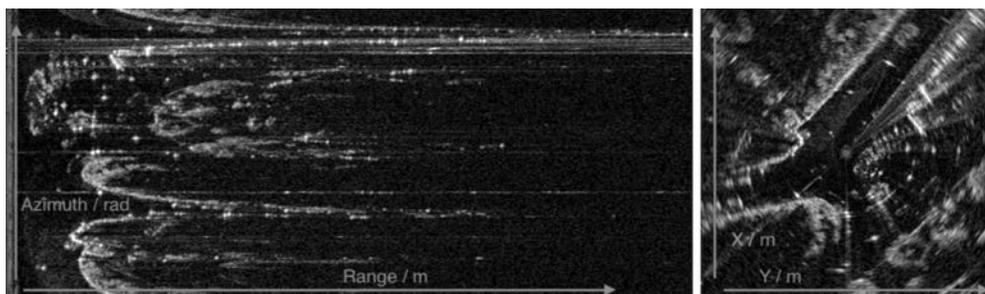


图 3.14 Oxford 雷达测量数据样例

3.3 智能小车数据收集与处理

为了收集数据,智能小车系统包括仿真的车道沙盘,它是对实际道路系统的模拟。如图 3.15 所示的车道沙盘可以灵活组装成不同大小、不同道路、不同网络情况的交通环境。车道系统设计有专用的交通标志、道路标线,转角和直道由特殊颜色的部件组成,此举能够简化智能小车的图像分类或识别算法。车道周围可设有黑色挡板,挡板可以屏蔽周围环境的干扰,便于生成更为精简的数据,能够减少数据预处理的时间,这些措施可以加快实验开发周期,令使用者的精力集中在人工智能算法的设计和训练优化上。智能小车使用的仿真车道沙盘较为简单,也省略了前期为实验建立地图的时间成本。

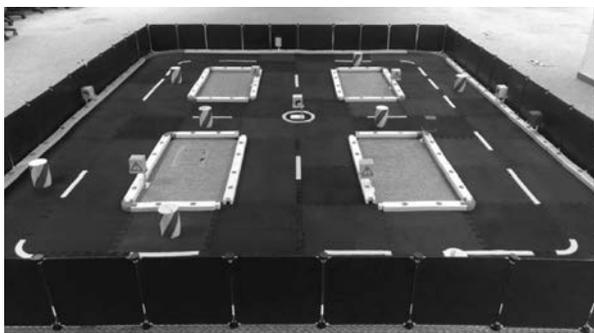


图 3.15 车道沙盘

在搭建好的车道环境中,使用者通过手动遥控智能小车在车道上行驶的方式完成数据路采。如图 3.16 所示,需要先将遥控器与智能小车进行蓝牙连接,在智能小车的 GUI(图形用户界面)或者命令行下,用手动模式启动智能小车,此时便可通过操作遥控器上的按键操作智能小车的前进、后退、停止、左转和右转。根据实际需要,使用者还可以进一步开发出智能小车其他的操作功能,例如横移、加速、减速和原地掉头等。

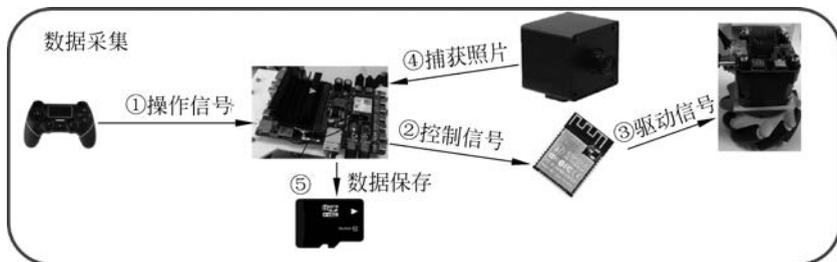


图 3.16 智能小车数据收集框图

在智能小车 mycar 软件库框架内的程序控制下,摄像头能以 20Hz 的帧率收集路况图片和操作信息,收集的信息数据将存储在智能小车的存储卡中。



微课视频 41

3.3.1 操控智能小车行驶

1. 启动数据收集

在 mycar 软件命令行方式中,智能小车的启动程序是~/mycar/drive.py,但需要先打开遥控手柄,成功建立蓝牙连接后,方可使用下面的命令启动智能小车的开车程序:

```
1 cd ~/mycar
2 sudo python3 drive.py - models js
```

或者是用默认参数启动:

```
1 cd ~/mycar
2 sudo python3 drive.py
```

2. 手柄按键映射

开车程序启动后,可以通过遥控器方向摇杆控制智能小车行驶,只有在前行或转向时,道路图像数据才会被持续保存,如果停止或者倒车,道路数据将不会被保存。

如果需要修改默认手柄的功能映射,可以编辑智能小车软件系统中的~/mycar/dellcar/parts/controller.py 文件进行修改(参见 2.4.3 节)。如果需要增加更多遥控功能,只需通过简单的代码修改就可以实现,例如使用 R2 按钮控制智能小车横移,则只需将配置文件中的 SMOOTH 值改为 True 即可,在重新启动开车程序之后按下 R2 按钮后,再使用遥控器方向摇杆就能令智能小车进行左右横移。

3.3.2 智能小车行驶数据收集

1. 数据收集

智能小车只要行驶的速度不为 0,数据收集程序就会不断地以 20Hz 的帧率收集路况图片和智能小车的遥控操作指令。如图 3.17 所示,路况图片在智能小车内保存为 jpg 格式,操作指令保存为 json 文件。默认情况下收集的数据将会存储在智能小车的~/mycar/tub 目录中。

```
lanqin@desktop-7:~/mycar$ ls tub/
1_cam-image_array_.jpg 14_cam-image_array_.jpg 27_cam-image_array_.jpg 77_cam-image_array_.jpg record_125.json record_175.json record_52.json
100_cam-image_array_.jpg 150_cam-image_array_.jpg 28_cam-image_array_.jpg 78_cam-image_array_.jpg record_126.json record_176.json record_53.json
101_cam-image_array_.jpg 151_cam-image_array_.jpg 29_cam-image_array_.jpg 79_cam-image_array_.jpg record_127.json record_177.json record_54.json
102_cam-image_array_.jpg 152_cam-image_array_.jpg 2_cam-image_array_.jpg 7_cam-image_array_.jpg record_128.json record_178.json record_55.json
103_cam-image_array_.jpg 153_cam-image_array_.jpg 30_cam-image_array_.jpg 80_cam-image_array_.jpg record_129.json record_179.json record_56.json
104_cam-image_array_.jpg 154_cam-image_array_.jpg 31_cam-image_array_.jpg 81_cam-image_array_.jpg record_12.json record_17.json record_57.json
105_cam-image_array_.jpg 155_cam-image_array_.jpg 32_cam-image_array_.jpg 82_cam-image_array_.jpg record_130.json record_180.json record_58.json
106_cam-image_array_.jpg 156_cam-image_array_.jpg 33_cam-image_array_.jpg 83_cam-image_array_.jpg record_131.json record_181.json record_59.json
107_cam-image_array_.jpg 157_cam-image_array_.jpg 34_cam-image_array_.jpg 84_cam-image_array_.jpg record_132.json record_182.json record_5.json
108_cam-image_array_.jpg 158_cam-image_array_.jpg 35_cam-image_array_.jpg 85_cam-image_array_.jpg record_133.json record_183.json record_60.json
109_cam-image_array_.jpg 159_cam-image_array_.jpg 36_cam-image_array_.jpg 86_cam-image_array_.jpg record_134.json record_184.json record_61.json
110_cam-image_array_.jpg 15_cam-image_array_.jpg 37_cam-image_array_.jpg 87_cam-image_array_.jpg record_135.json record_185.json record_62.json
110_cam-image_array_.jpg 160_cam-image_array_.jpg 38_cam-image_array_.jpg 88_cam-image_array_.jpg record_136.json record_186.json record_63.json
111_cam-image_array_.jpg 161_cam-image_array_.jpg 39_cam-image_array_.jpg 89_cam-image_array_.jpg record_137.json record_187.json record_64.json
112_cam-image_array_.jpg 162_cam-image_array_.jpg 3_cam-image_array_.jpg 8_cam-image_array_.jpg record_138.json record_188.json record_65.json
113_cam-image_array_.jpg 163_cam-image_array_.jpg 40_cam-image_array_.jpg 90_cam-image_array_.jpg record_139.json record_189.json record_66.json
114_cam-image_array_.jpg 164_cam-image_array_.jpg 41_cam-image_array_.jpg 91_cam-image_array_.jpg record_13.json record_18.json record_67.json
115_cam-image_array_.jpg 165_cam-image_array_.jpg 42_cam-image_array_.jpg 92_cam-image_array_.jpg record_140.json record_190.json record_68.json
116_cam-image_array_.jpg 166_cam-image_array_.jpg 43_cam-image_array_.jpg 93_cam-image_array_.jpg record_141.json record_19.json record_69.json
117_cam-image_array_.jpg 167_cam-image_array_.jpg 44_cam-image_array_.jpg 94_cam-image_array_.jpg record_142.json record_1.json record_6.json
118_cam-image_array_.jpg 168_cam-image_array_.jpg 45_cam-image_array_.jpg 95_cam-image_array_.jpg record_143.json record_20.json record_70.json
119_cam-image_array_.jpg 169_cam-image_array_.jpg 46_cam-image_array_.jpg 96_cam-image_array_.jpg record_144.json record_21.json record_71.json
```

图 3.17 智能小车数据集图片文件



微课视频 42

由于数据质量对于机器学习至关重要,在数据收集阶段要考虑数据集的多样性,因此对于如何收集数据应该进行充分的设计和思考。如图 3.18 所示,收集数据时需要充分考虑不同的光线条件、智能小车与车道上标线的不同相对位置、障碍物的不同位置、与交通标志的不同接近方式等路况的图像,并对它们进行收集。

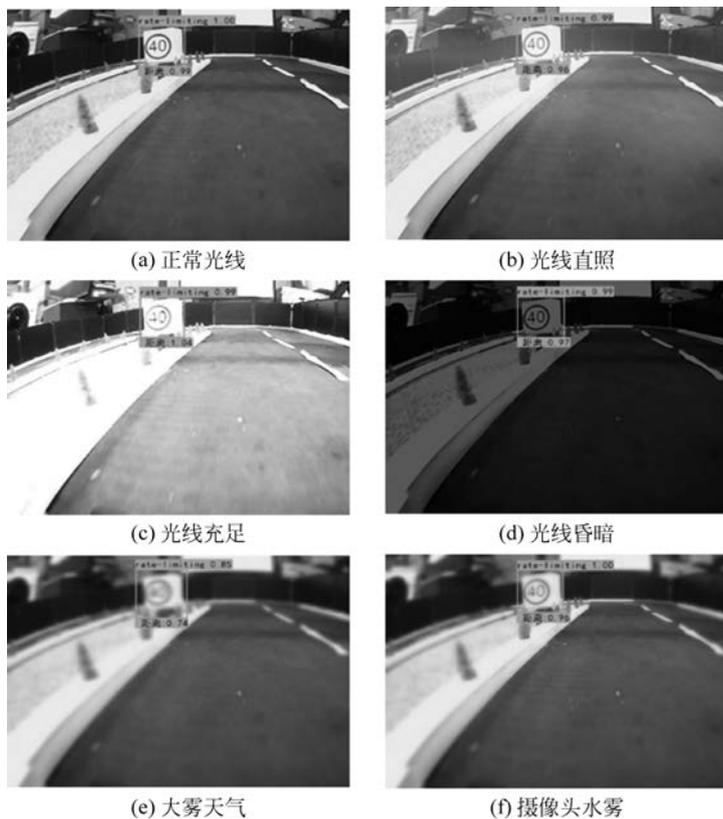


图 3.18 智能小车数据集图片

2. 数据传输

当数据收集完成,需要将智能小车收集的数据传输到后台服务器进行预处理和训练。模型训练完成后还需要将后台服务器中的模型文件传输回智能小车中进行推理。数据文件的传输方法有很多种,下面是一个简单的示例:

```
1 rsync -aP ~/mycar/tub/gpu@192.168.x.x:mycar_server/tub
2 rsync -aP gpu@192.168.x.x:mycar_server/models/mymodel ~/mycar/models/
```

3.3.3 智能小车数据标注

把数据上传到服务器后需要根据算法要求对路况图片进行标注。如图 3.19 所示,可以使用软件库自带的 LabelImg 工具标注数据并导出标记文件。



图 3.19 对路况中的交通标志进行标注

与数据收集相当,数据的标注过程也是非常重要,而且都是耗费人力的过程。在标注前,先要对数据图片进行清洗,清洗的过程就是将不合格(例如智能小车压线、过界、碰触障碍物等)的路况图片删除。这是数据预处理的一部分工作,数据预处理还包括对路况图片的修饰、编辑和校正等工作,这些都需要人工完成,其处理的效果会对后续的模式训练产生很大影响。

清洗后的图片数据可以划分为训练数据集、验证数据集和测试数据集。对于这些数据需要认真地进行高质量标注。默认情况下,智能小车以摄像头图像为主,所以一般是对图片类型的数据进行标注,而随着智能小车选用不同的传感器,需要标注的数据类型也会不一样。这些标注工作会由 mycar 软件库中的工具自动完成,当完成标注后,将自动形成如图 3.20 所示的数据集。

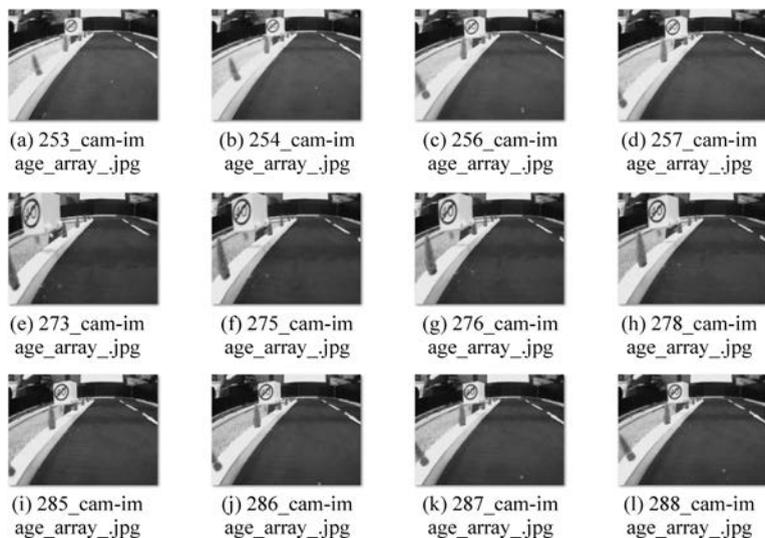


图 3.20 图片完成标注后形成的数据集



微课视频 43

3.3.4 智能小车数据分析

如果智能小车只配置了摄像头,其所产生的数据就会相对简单。主要为 jpg 格式的路况图片和记录驾驶操作的 json 文件。如图 3.21 所示,在默认配置下,智能小车的驾驶操作有前进、后退、停止、左转、右转等,这些操作将作为标注自动记录在数据集中。

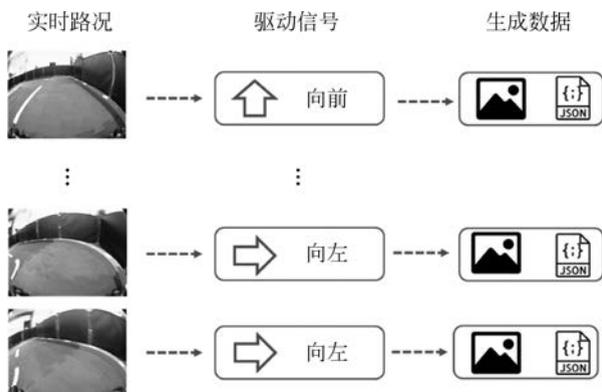


图 3.21 智能小车数据分析

智能小车收集的驾驶操作信息将包含如下内容:

- (1) user/angle(用户/转向角): 1 左转,0 直行,-1 右转。
- (2) user/throttle(用户/油门): 1 前进,-1 后退。
- (3) cam/image_array(相机/图像陈列): 对应的路况图片。

具体代码如下:

```

1 {
2   "cam/image_array": "100_cam - image_array. jpg",
3   "user/angle": -1,
4   "user/throttle": 1,
5   "user/mode": "user"
6 }
```

3.3.5 智能小车数据清洗

如图 3.22 所示,通过智能小车的路况图片可以直接查看图片和其文件的时间戳,如果存在不合格图片数据则需要进行手工清洗(删除)。例如在手动遥控过程中产生了操作失误,此时产生的路况图片和操作信息是不正确的,不应该用作训练数据,因此可以把这个对应时间段的数据完全删除。



微课视频 44



图 3.22 在数据集中挑选并清洗不良图片数据

3.3.6 智能小车数据可视化

数据可视化可以帮助使用者更好地分析数据、理解数据,从而更好地运用数据训练出高质量的模型。智能小车默认配置以摄像头为主,驾驶操控动作主要是前进、左转、右转,数据比较简单,因此可以采用比较简单的方法进行分析。

如图 3.23 所示,从这个智能小车路采数据的标签分布可以直观看出,相对于右转信息,左转信息相对较少,应该再多收集一些左转的路况信息。更明显的是,直行信息远比转向信息多得多,这从车道沙盘的实际环境来看是合理的,但是从训练的角度来看,这种不平衡的数据会带来模型训练的偏差。根据数据分析,使用者可以采取一些措施弥补这种数据不平衡的问题,例如:

- (1) 补充更多的左转、右转路采数据;
- (2) 对左转、右转数据进行数据增强处理,以扩展其数量;
- (3) 选择让每个训练的数据集中的左转、右转、直行的数据量相当,即不同迭代中重复使用转向数据与不同的直行数据组成当前训练集进行本轮迭代的训练。

随着引入更多的传感器、更复杂的控制参数,数据的维度将不断上升。复杂数据的分析可以使用前面提到的降维可视化工具进行数据可视化分析。

图 3.24 是通过 t -SNE 工具分析了 3000 张智能小车路况图片的结果,其中左转、右转、直行分别有 1000 张。通过降维到二维平面来观察(其中标注的 0、1、2 三种标签分别代表了左转、右转和直行三类图片)可以看到每种不同的数据具有一定聚合分类的特征,因此可以预测用这些数据去训练的分类模型,应该能取得较好的结果。

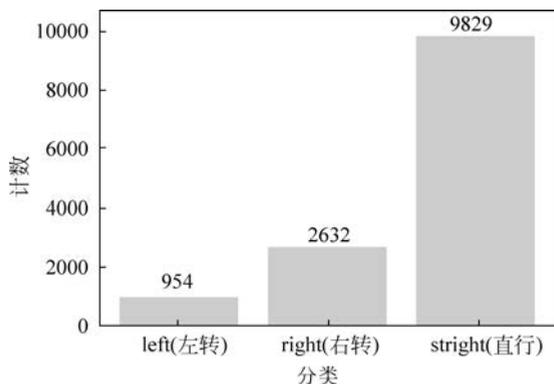


图 3.23 智能小车数据集标签分布

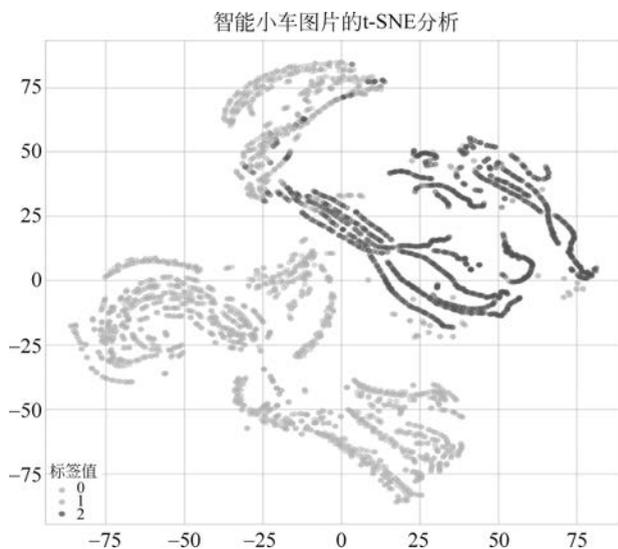


图 3.24 智能小车转向数据聚合情况分析



彩图

3.3.7 智能小车数据处理工具

由于数据量大,采用不同的读取方法将数据输入训练算法中对训练速度的影响会有非常大的不同。如图 3.25 所示,采用不同的软件工具处理 json 文件的效率存在明显的差别。

除了选择正确的软件工具,数据处理的流程也是优化的重点。例如若训练的每一个循环迭代中都从磁盘读入原始数据进行训练计算,存储原始数据的磁盘 I/O 操作则会消耗大量的时间,加上 json 文件解析所消耗的时间,会让训练过程变得非常漫长。因此在通常情况下,需要提前处理相关数据,例如将数据都转化成 TFRecord 格式保存,在训练过程中使用预读技术提前读入内存中,可以极大减少每个训练循环的时间,提高训练效率,如图 3.26 所示。



彩图

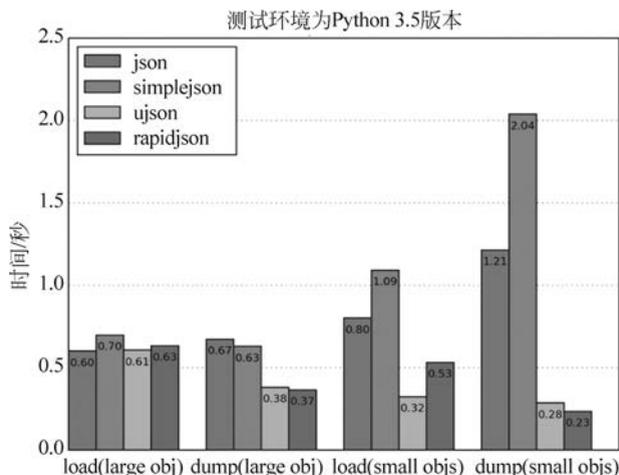


图 3.25 采用不同的软件工具处理 json 文件的效率对比

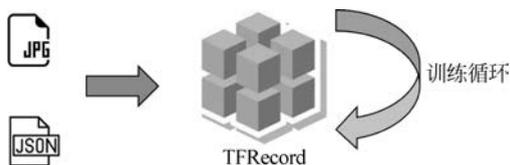


图 3.26 通过 TFRecord 工具加快数据集读取效率示意图



微课视频 45

3.4 开放性思考

从自动驾驶数据的角度考虑,各种传感器的优势和局限性都很明显。激光雷达因为成本问题,目前仍然未能实现大规模商用,因此基于视觉的自动驾驶解决方案成为部分自动驾驶厂商探索的主要方向。基于视觉的自动驾驶方案的背后有这样一个假设:认为驾驶员就是通过视觉感知驾驶环境的信息,做出驾驶行为的决策,故自动驾驶在技术上应该可以同样仅依据视觉传感器数据实现准确的环境感知,为自动驾驶系统提供决策支持。请读者思考自己是否支持这样的观点?

但也有厂商和研究者认为这个假设事实上并不完整,因为对于驾驶员而言,驾驶环境中人类大脑的感知和决策会利用到自身多年积累的各种常识和驾驶经验,这些经验能够有效帮助驾驶员理解道路环境中正在发生的场景,预测场景中人和车辆将要发生的行为。而这些通过多年学习获得的常识和经验恰是单纯的视觉数据所不可能具有的。目前的机器学习和深度学习模型都是学习数据中已经具有的隐含模式,因此如何有效利用传感器数据形成对驾驶环境的感知,以人类大脑的方式或者机器智能的方式理解驾驶环境并做出决策,仍然是极具挑战性的开放性问题的。也许人工智能的价值就在于形成类似人类的常识和经验积累。这些问题有待于读者进一步思考和分析。

传感器是汽车感知周围环境的硬件基础,而自动驾驶离不开感知层、控制层和执行层的相互配合。摄像头、雷达等传感器获取图像、距离、速度等信息,扮演人类“眼睛”和“耳朵”的角色。目前自动驾驶的事故原因绝大多数出现在传感器这个环节,将各类传感器融合在一起,能否起到 $1+1>2$ 的效果呢?

激光雷达获取信息时会遇到很多现实环境中的具体问题,包括:①极端环境的干扰(例如雨点打到地面溅起水雾就可能会产生误判);②远距离感知;③特殊钢铁物体的感知;④多激光点云拼接等。特别是如何对拼接部位的变形进行正确覆盖?这也是一个极大的挑战。

通过增加传感器的数量,并让多个传感器融合来提高汽车自动驾驶能力。多个同类或不同类传感器分别获得不同的局部和类别信息,这些信息之间可能相互补充,也可能存在冗余和矛盾,而控制中心最终只能下达唯一正确的指令,这就要求控制中心必须对多个传感器所得到的信息进行融合,综合判断。在使用多个传感器的情况下,要想保证安全性,就必须对传感器进行信息融合。多传感器融合可显著提高系统的冗余度和容错性,从而保证决策的快速性和正确性,多传感器融合已成为自动驾驶的必然趋势。

实现传感器融合是有前提条件的,例如数据对准的问题,随着传感器越来越多,时间同步是最大的难题,如何进行多传感器同步和效率同步,是未来的挑战之一。又例如异构传感器多目标跟踪问题,包括虚假目标和不感兴趣目标的剔除问题,这些都需要在融合过程中进行处理。还有自动驾驶车辆在环境感知中的个体局限性,包含自动驾驶车辆自身搭载传感器的固有局限,也有复杂交通状况下普遍存在的各种障碍物与遮挡问题等,如何寻找有效的算法以应对感知数据中隐藏的固有局限?这些都是需要读者思考的开放问题。

3.5 本章小结

在哲学范畴内,形式逻辑主要有归纳逻辑和演绎逻辑两种。人类本身具有学习能力,因此能够对于客观世界进行观察和分析,并在学习中归纳总结出一些基本的运行规律,这是归纳逻辑的体现。人类可以制造工具,人类将自身所认识到的运行规律纳入高级工具中,这些工具就能够代替人类高效率地生产或改造自然环境,这是演绎逻辑的体现。而机器学习是人类将学习的能力赋予机器,让机器具备类似于人类观察和分析世界并归纳内在规律的能力。机器学习的出现是人类制造工具水平的一个质的飞跃。

机器学习是让机器模仿人类具有学习的能力,人类的学习需要有学习对象或者学习资料。对于机器学习,其需要的学习资料就是数据。机器学习的优势在于能够充分利用计算机大量并行计算的能力,能同时处理大量的数据信息,而其劣势在于目前尚无法像人类那样建立起全社会通用的机器学习知识体系和资料库,对问题理解的深度和关联解决问题的能力也有不足,缺乏创造力和适应能力等。

然而在自动驾驶领域中,机器学习所涉及的问题虽然十分复杂,但是仍然有希望在充分利用计算机庞大并行处理计算能力的基础上进行求解。而机器学习庞大算力的运用基础是要为其提供庞大的有效数据,也就是机器学习所需的学习资料。因此对数据的收集、处理是



自动驾驶系统机器学习的必要前提。

如图 3.27 虚线框所示,本章在介绍现有数据处理技术的基础上,以智能小车为例向读者介绍了如何收集自动驾驶训练所需的数据及对数据集的处理方法。应当指出,为引导读者逐步熟悉并进入自动驾驶人工智能领域,这些案例所展示的都是最简单的处理过程,而业界实际的处理远比本书中所介绍的复杂,建议愿意深入了解相关内容的读者自行查阅最新的自动驾驶行业报告,在进行开放性思考的过程中完成学习的提升。



图 3.27 章节编排