软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.006266]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

自动驾驶智能系统测试研究综述*

朱向雷[1,2], 王海弛[1], 尤翰墨[1], 张蔚珩[1], 张颖异[1], 刘 爽[1], 陈俊洁[1], 王 赞[1], 李克秋[1]

[1](天津大学 智能与计算学部,天津 天津 300350)

[2](中国汽车技术研究中心有限公司,天津 天津 300300)

通讯作者: 陈俊洁, E-mail: junjiechen@tju.edu.cn

摘 要: 随着人工智能技术的深入发展,自动驾驶已经成为人工智能技术的典型应用,近十年得到了长足的发展,作为一类非确定性系统,自动驾驶车辆的质量和安全性得到越来越多的关注.对自动驾驶系统,特别是自动驾驶智能系统(如感知模块,决策模块,综合功能及整车)的测试技术得到了业界和学界的深入研究.本文调研了 56 篇相关领域的学术论文,分别就感知模块、决策模块、综合功能模块及整车系统的测试技术、用例生成方法和测试覆盖度量等维度对目前已有的研究成果进行了梳理,并描述了自动驾驶智能系统测试中的数据集及工具集.最后,对自动驾驶智能系统测试的未来工作进行了展望,为该领域的研究人员提供参考.

关键词: 自动驾驶智能系统;测试用例生成;测试覆盖标准;

中图法分类号: TP311

中文引用格式: 朱向雷等,自动驾驶关于智能系统研究测试的研究综述.软件学报. http://www.jos.org.cn/1000-9825/6266.htm

英文引用格式: Zhu XL, Wang HC, You HM, ZhangWH, Zhang YY, Liu S, Chen JJ, Wang Z, Li KQ. Survey on testing of intelligent systems in autonomous vehicles, 2021 (in Chinese). http://www.jos.org.cn/1000-9825/6266.htm

Survey on Testing of Intelligent Systems in Autonomous Vehicles

ZHU Xiang-Lei[1,2], WANG Hai-Chi[1], YOU Han-Mo[1], ZHANG Wei-Heng[1], ZHANG Ying-Yi[1], LIU Shuang[1], CHEN Jun-Jie[1], WANG Zan[1], LI Ke-Qiu

[1](College of Intelligence and Computing, Tianjin University, Tianjin 300350, China)

[2](China Automotive Technology and Research Center Company Ltd., Tianjin 300300, China)

Abstract: With the development of Artificial Intelligence, Autonomous Vehicles have become a typical application in the field of artificial intelligence. In recent 10 years, autonomous vehicles have already made considerable processes. As an uncertain system, their quality and safety have attracted much attention. Autonomous Vehicle testing, especially testing the Intellectual Systems in Autonomous Vehicles (such as Perception Module, Decision Module, Synthetical Functional Module and the Whole Vehicle) gain extensive attention from both industry and academia. This survey offers a systematical review on 56 papers related to autonomous vehicle testing. Besides, this survey analyzes the testing techniques with respect to Perception Model, Decision Model, Synthetical Functional Module and the Whole Vehicle, including test case generation approaches, testing coverage metrics, as well as datasets and tools widely used in autonomous vehicle testing. Finally, this survey brings highlights future perspectives on autonomous vehicle testing and provides reference for researchers in this field.

基金项目: 国家自然科学基金(61872263, 61802275, 62002256, U1836214);天津市智能制造专项资金项目(20193155);天津大学自主科研基金(2020XZC-0042)

Foundation item: National Natural Science Foundation of China (61872263, 61802275, 62002256, U1836214); Intelligent Manufacturing Special Fund of Tianjin(20193155); Innovation Research Project of Tianjin University (2020XZC-0042)

收稿时间: 2020-09-15; 修改时间: 2020-10-26; 采用时间: 2020-12-14; jos 在线出版时间:2021-01-22

Key words: Intelligent systems in autonomous vehicles; Test case generation; Test coverage metric;

在过去的几十年中,随着计算机和通信技术的高速发展,交通系统正逐步引入智能要素,自动驾驶技术应运而生.由于造成交通事故最主要的原因之一是人为因素,引入自动驾驶技术能够大大降低交通事故发生的概率.近十年来,随着感知组件和智能算法的快速发展,全球有多家企业和研究机构正积极开发自动驾驶技术.国际自动化工程师学会/美国汽车工程师学会(SAE)于 2014 年提出了自动驾驶五级分级方案,该方案成为了当前被普遍接受的标准.自动驾驶分级方案从 L0 递增到 L5(L0 为传统驾驶,不算在五级分级方案中)分别代表了自动驾驶汽车的智能程度由浅至深.L1 级别为辅助驾驶,在特定的条件下自动驾驶汽车具有一个或多个自动控制功能,但不能脱离驾驶员的控制.具有 L1 级别的自动驾驶车辆往往包含高级驾驶辅助系统(ADAS),并且目前高级驾驶辅助功能已经逐渐成为中高端车型的标准配置.尽管目前已有越来越多的机构在为 L5 级别自动驾驶功能提供解决方案,但目前的技术尚难以满足 L5 级别自动驾驶即完全自动驾驶的需求.自动驾驶系统的架构一般由三个部分组成:感知模块,决策模块以及控制模块.感知模块负责自动驾驶汽车周围环境认知,往往由摄像头和雷达等组件组成.决策模块需要根据感知模块感知到的信息,根据一定的算法得出自动驾驶汽车下一步行为并传递给控制模块.控制模块负责将决策指令传递给汽车硬件,例如方向盘,油门等,完成对自动驾驶车辆的控制.

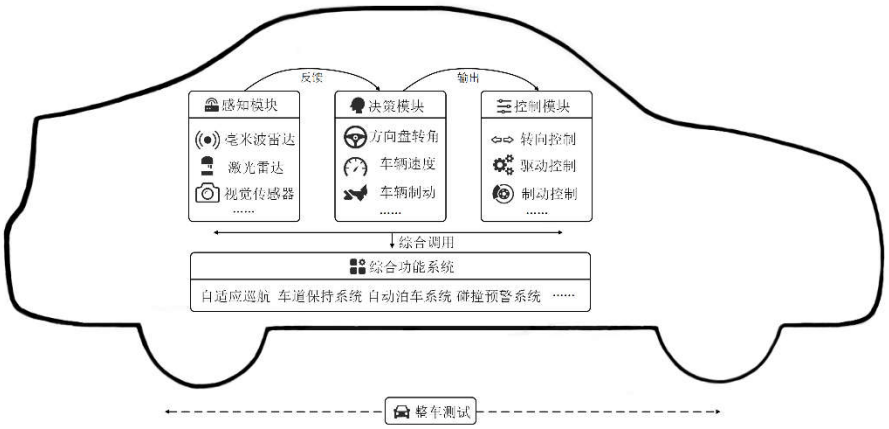

Fig.1 Overall structure

图 1 自动驾驶架构

然而,与任何依赖软件算法的系统一样,自动驾驶系统容易在一些特殊的场景下做出错误的判断并导致事故的发生,同时也容易受到恶意攻击进而导致系统失效.如由 Uber 生产的自动驾驶汽车在 2018 年发生了事故,导致了一名行人死亡.因此,自动驾驶车辆需要经过严格且全面的验证才能保证车辆的安全使用.目前已有多个工作研究自动驾驶系统的测试和验证技术.众所周知,直接采用真实车辆进行检测的成本是非常昂贵的,因此自动驾驶车辆在进行真车测试之前往往会在虚拟环境中进行测试.通过在仿真软件中提供与真实驾驶场景相似的仿真场景,来测试自动驾驶算法的健壮性及正确性.根据测试对象的受控程度,虚拟测试又分为软件在环(Software-in-the-Loop, SiL)、硬件在环(Hardware-in-the-Loop, HiL)、模型在环(Model-in-the-Loop,MiL)和车辆在环(Vehicle-in-the-Loop, ViL).由于自动驾驶测试场景的多样性和真实驾驶事故中场景的突发性,如何生成有效测试用例以及如何评价测试场景有效性成为了目前广泛研究的课题.Koopman 等人[1]总结了自动驾驶的测试和验证的挑战.其认为自动驾驶的测试与验证面临着以下五个挑战领域:驾驶员非在环、复杂需求、不确定性算法、归纳学习算法和故障操作系统.

目前 Li 等人,Kang 等人和 Garcia 等人针对自动驾驶系统的测试进行了总结.Li 等人[2]总结并讨论了人工

智能和人工智能测试之间的关系,并将自动驾驶作为具体案例讨论了人工智能在其中的测试挑战,包括测试任务的定义、测试框架及其形式化定义以及并行测试.Garcia 等人[3]分析了 Baidu Apollo 和 Autoware 两个开源项目的 16851 次提交中的 499 个与自动驾驶相关的错误,并将这些错误进行了分类,从中总结出了 13 种导致错误的根本原因、20 种错误症状和 18 种自动驾驶组件.Kang 等人[4]整理了 37 个开源或者半开源的自动驾驶数据集和 22 个虚拟仿真环境.整理出的数据集全部来自真实采集的数据,且至少包含来自摄像头,激光雷达或毫米波雷达采集的数据.与上述针对自动驾驶系统测试的总结性论文不同,本文充分调研了近年来与自动驾驶测试相关的论文,整理了针对自动驾驶系统与智能相关的模块的测试,总结了相关测试的测试用例生成方法以及覆盖度量指标,并对自动驾驶测试的未来发展做出了展望.本文也是针对自动驾驶智能系统测试的首篇综述,为该领域的相关研究团队提供参考.

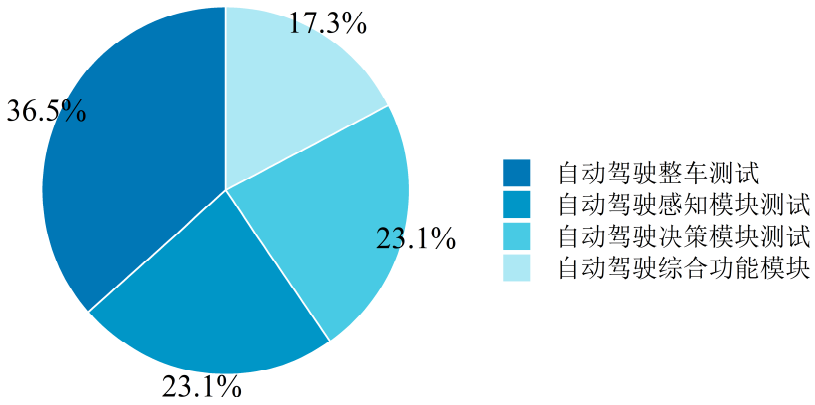

Fig.2 Proportion articles in key chapters

图 2 论文在各个章节中的分布

如前文所述,自动驾驶系统往往由感知模块、决策模块以及控制模块组成.其中控制模块往往与智能无关,因此本文只针对感知模块和决策模块的测试技术进行了总结.目前的研究中部分是单独针对感知模块进行测试,部分针对决策模块进行测试,也有许多研究针对整车进行测试.与此同时,一些高级功能模块同时包含了感知模块和决策模块甚至控制模块的相关组件,例如高级辅助驾驶系统(ADAS),有许多学者致力于针对这样的功能模块进行测试.因此本文针对自动驾驶智能系统测试的总结框架如图 1 所示.

为完成本文的研究问题,我们首先使用“Autonomous Vehicle Testing”, “Automative Systems Testing”, “Self-driving Testing”, “Automated Vehicle Testing”等关键词在国内外知名的学术搜索引擎(例如 CNKI、必应学术、谷歌学术、DBLP 等)上进行搜索,并筛选出与本综述相关的文章.随后,通过检索筛选出来的文章的参考文献和相关作者的发表论文列表,进一步补充相关文献.最终,我们确定了与本文相关的论文 56 篇,其中 19 篇发表在 CCF-B 类及以上或 SCI-2 区及以上,9 篇论文托管在 arxiv 上,其余论文多发表在 IEEE 等知名学术机构下,论文大多发表在 2016 年及以后,其中发表在 2019 年及以后的文章有 20 篇,占全部文章 35.7%.论文在各个章节的分布参见图 2.具体的年份分布参见图 3.

本文余下部分的结构如下:第一节介绍了针对自动驾驶系统中感知模块测试的相关研究,第二节描述了针对自动驾驶系统中决策模块测试的相关研究,第三节总结了针对自动驾驶系统中综合功能模块的测试,第四节梳理了针对自动驾驶整车的测试,第五节整理了当前研究中用到的数据集和模拟器.最后,对自动驾驶智能系统测试的未来工作进行了总结和展望.

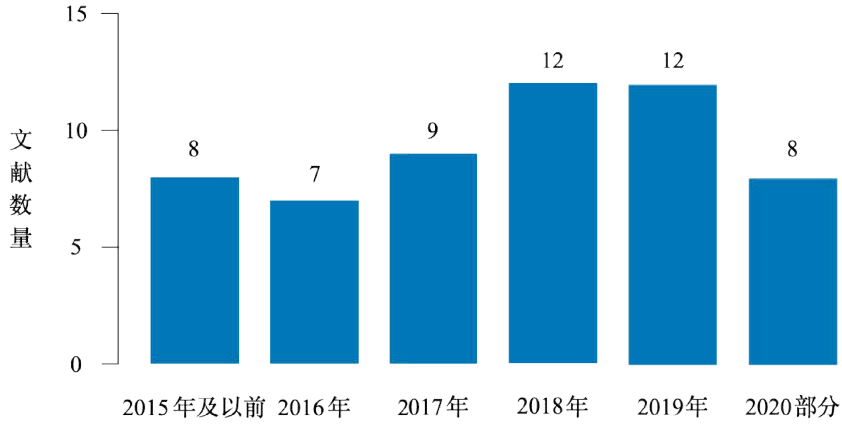

Fig.3 Quantity distribution of articles in different years

图 3 论文年份分布

1 感知模块

在自动驾驶系统中,感知模块作为辨识周边环境、物体的“感官”,起着至关重要的作用,是整个自动驾驶系统能够安全、高效运行的基础.感知模块集成了包括视觉感知在内的多种传感器,例如深度摄像机、毫米波雷达和激光雷达等.感知模块通过各种感知元件采集到周边物体的信息,结合目前发展迅速的深度神经网络模型,将各种收集到的感知数据融合,得到自动驾驶汽车周边环境详实的数据表示.感知模块使得自动驾驶系统能够像人类一样去“认知”周围的环境事物,去分析辨别周围的物体.深度神经网络技术的不断发展加速了感知模块的发展速度,深度卷积神经网络能够很好的辨识出自动驾驶汽车前方的环境,但与此同时,由感知模块带来的安全性问题也引起了人们的注意,目前的感知模型在物体识别上仍具有不同程度的错误率,如何通过测试的方法检验感知模块的正确性成为了相关研究的焦点.目前,许多学者已经对如何测试基于神经网络的感知模块进行了深入研究.经过本文的整理,目前针对感知模块的测试主要分为以下几种方式:生成对抗样本、基于内省的自我评估,以及通过生成真实图像来测试感知模块.

1.1 相关研究现状

生成对抗样本是测试自动驾驶感知模块一种较为常见的方法,与传统软工方法中的基于变异的模糊测试(mutation-based fuzzing)相似,该方法通过对原测试用例(这里以图像为例)进行包括平移、交换和突变等在内的不同程度上的修改,在保留原测试用例的基本外表不变的基础上(人类可以正常识别),使得感知模块产生错误的识别结果.对抗样本在性质上又可以分为实体对抗样本和数字对抗样本.实体对抗样本以“贴纸”的形式可以通过直接粘贴到实体上对感知模块进行攻击.数字对抗样本则针对数字图像进行修改

Eykholt 等人[5]设计出一种产生实体对抗样本的算法以攻击目标识别系统.它可以成功地欺骗包括 YOLO,Faster-RCNN 在内的目标检测器.该算法可以将对抗输入压缩为“对抗贴纸”,被对抗贴纸附着的目标物体(例如停车指示牌)会被目标检测器错误地识别.该团队用附着对抗贴纸的停车指示牌录制了相关视频,并验证该帖纸在大部分视频帧中都会导致 YOLOv2 检测器错误地识别物体.在此基础上,该团队又提出了一个普适的攻击算法:RP2(Robust Physical Perturbations),以在不同的物理条件下生成对抗扰动.Eykholt 等人[6]认为,物理对抗样本必须能够经受住不断变化的物理条件,并能有效地欺骗分类器.物理对抗样本必须要与环境条件相适应,且不能攻击背景图像,要足够微小,并且有一定误差.在实验中,针对 LISA-CNN[7]和

GTSRB-CNN[8]模型(LISA 是一个包含 47 种标志牌的美国交通标志牌数据集,GTSRB[9]是德国交通标志牌数据集.LISA-CNN 和 GTSRB-CNN 分别是在两种数据集上训练生成的卷积神经网络模型.),该团队设置了多种攻击,包括限制物体的海报攻击(针对广告牌的图片扰动,并打印成海报替换广告牌)和贴纸攻击(模拟广告牌被广告和涂鸦遮挡的情况),证明了攻击的有效性.除此之外,作者设置了静态实验(实验室测试)和动态实验(驾车测试)两种实验方法,以验证 RP2 攻击在多种实验条件下都有较高的成功率.该实验证明生成在变化距离和角度上鲁棒的对抗样本是有可行性的.

深度网络模型与传统软件工程系统有着非常大的差异,在自动驾驶中,感知模块的输入往往是一段连续的视频,且视频中图像的角度和距离也时刻发生着变化,这导致了对抗性样本不能时刻保持高效的攻击性.Lu 等人[10]针对此现象提出了新的看法:生成对抗样本对于自动驾驶物体识别分类器或者探测器的影响并不大.Lu 等人使用了几种常见的对抗样本生成方式(FastSign Attack,Iterative Attack,LBFGS Attack)针对道路指示牌进行扰动.为了模拟真实场景,该团队打印了扰动后的停车指示牌图片,并悬挂在不同距离和角度的地点处.在实验中,该文计算在各种距离下的对抗样本失效比率,发现对抗样本失效比非常高,且随悬挂距离增加而增加.这也证明了距离和角度很容易打破对抗扰动对分类器和探测器的影响,对抗样本图片在真实场景下很难对分类器和探测器产生影响.这也使得在大部分真实场景下,使用电子手段生成的对抗样本对分类器和探测器的影响甚微.

外部攻击的方法受到了角度和距离的制约,一些学者尝试从感知模块本身入手,希望其能够自动发现自身存在的错误.感知系统的性能下降会影响决策系统,使其无法做出可靠的决策.这就促使研究人员需要构建具有情境感知能力的系统,以评估系统在本时刻做出的决策是否可靠.这种自我评估能力称为“内省”.Daftry 等人[11]为感知系统中的内省行为提出了一个通用框架,使系统能够通过测量系统状态的合格程度,来识别由于感知系统性能下降而无法做出可靠决策的情况.该框架由 Spatio-temporal CNN 与线性 SVM 组成,根据最后输出的故障预测得分来判断预测的可靠性,以便基于该输入来计划一个动作,或将其丢弃并采取替代行为.Daftry 等人在自主导航任务中进行实验:通过观察无人驾驶的微型飞行器在高、低杂波密度区域内数次自主飞行(1.5m/s)的平均距离来评估避障系统的性能.基于实验得到的故障率曲线等结果证明:内省模型使得无人机能够自主飞行的平均无坠毁距离超过 1000 米,显著提高了风险规避性能.

类似于内省的想法,Gurău 等人[12]尝试了一种新的范式:根据过去在同一工作空间运行所积累的经验,预测感知系统出错的概率.该文假设在相似的驾驶条件下,相同的物理位置会导致相似的感知结果.因此,他们多次遍历相同的路线并沿其收集性能估算值,以创建性能记录.并在测试时利用这些记录来预测当前遍历系统的性能.据此,该文提出了两种模型,第一种方法仅仅特定于位置特征;第二种方法则在合并过去结果时,还额外考虑了外观相似性,在进行位置估计后得到的局部邻域中进行搜索,找到与实时帧最匹配的记录.在实验过程中,该团队反复遍历相同的路线并更改操作条件,利用收集的性能记录来预测两个不同的基于图像的行人检测器在当前遍历中的性能.实验结果表明:这两种方法均可减少两种不同检测器模型的错误决策.此外,利用视觉外观还可以提高感知系统性能预测的准确性.

Ramanagopal 等人[13]结合了软件工程方法中的差异测试方法,设计并实现了一个能够自动检测自动驾驶车辆感知模块识别错误的模型.他们通过两种方法来检测感知模块发生的错误:(1) 基于时间因素,即比较相同检测器在相邻时间的检测结果;(2) 基于空间因素,即比较相同检测器在带有空间信息的图像上的识别情况.该文认为目标检测器在一个连续的视频片段的相邻帧中会发生识别结果不一致的情况,而目标追踪器在这种情况中的表现则较为可靠.于是他们使用两种结果的差异作为判断感知模块在时间因素上是否发生错误的依据.同时,他们发现深度相机采集到的深度图片对目标检测器的检测结果会发生影响,即两张场景完全相同的图片,其中一张具有深度信息,会让目标检测器得到不同的检测结果.于是该团队使用带有深度信息图片的检测结果与普通图片的检测结果的差异,作为判断感知模块在空间因素上是否发生错误的依据,并设计了一个二分类器,依据差异的多种特征进行分类,得到目标检测器可能发生的错误.该团队使用三种先进的目标检测器作为实验对象:SSD,Faster R-CNN,RRC,使用 Sim200k[14]数据集作为检测器的训练集,并通过游戏

引擎生成了一个带有目标追踪信息的数据集 GTA 作为测试集,同时也使用了现实生活中的真实数据集 KITTI 作为测试集.实验结果表明,三种检测器在使用了错误检测器纠正后,F1 得分能够分别上升 4.22%, 0.87%, 3.57%.

无论是“内省”的方式,还是生成对抗样本,都是通过辅助攻击的方式来对以卷积神经网络为主的感知模块进行测试,而不是直接生成完整的测试图像或测试场景.随着对感知模块测试的研究逐步深入,针对感知模块的测试逐渐基于两部分,即生成完整的仿真场景和完整的真实场景.与对抗生成技术相比,该类方法的目标是生成真实的图像,而非将干扰引入到先前存在的图像中.

Dreossi 等人[15]认为前人的验证技术[16]通常会对经过处理的卷积神经网络施加限制,并遭受可伸缩性问题的困扰.故该团队提出了一个通过生成综合数据集来系统测试卷积神经网络的框架,以达到系统地分析用于自动驾驶中汽车分类的卷积神经网络的目的.框架由三个主要模块组成:图像生成器,采样方法,及可视化工具.每次循环中对图像特征进行采样,使用修改后的配置来生成图像,由卷积神经网络返回预测结果,进而检查和可视化 CNN 行为.在实验中该团队使用图像生成器和 Halton 采样序列生成了 1000 张合成图像,并使用生成的图像来分析训练好的 SqueezeDet 与 YOLO 模型.其中,训练好的 SqueezeDet 模型在该方法生成的合成图像下有很高的置信度和 IOU(交并比),并且能够由此训练集有效找出该模型识别的盲点;而 Yolo 模型在该方法生成的合成图像下的置信度和 IOU 则随行车距离增加而降低.该结果表明:文中框架可以有效地生成训练和验证数据集进而评估用于自动驾驶汽车分类的卷积神经网络模型的性能.

基于仿真场景的感知模块测试逐渐兴起,但在模拟场景下训练的模型是否可以很好地适用于真实场景已经成为一个重要问题.对此,Talwar 等人[17]使用 LGSVL[18](LG Silicon Valley Lab)采集了不同保真度的模拟场景,用以训练 YOLOv3 模型,并在真实场景数据集 KITTI 上进行评估.在实验中, 该团队发现了模拟场景和真实场景训练中的各种迁移性问题,例如虚拟场景车密度低,对夜晚条件下的识别不够等.该团队发现,仿真场景下训练的模型在仿真场景中预测效果会更好,在真实场景上效果欠佳,并由此推定在仿真场景下的训练很难广泛应用到真实场景中.实验同样证明了训练集的多样性比仿真场景的真实性在训练过程中起到的作用更大.因此,如何更好地解决虚拟场景和现实场景之间的有效转换将成为自动驾驶测试中下一个需要攻克的难点.

1.2 测试用例生成

为了更好地测试基于神经网络的感知模块,多种针对感知模块生成测试用例的方法被提出.如图 4 所示,现有的针对感知模块的测试用例生成方法主要可分为两类:基于真实场景的测试用例生成方法和基于模拟场景的测试用例生成方法.其中,感知模块中除基于神经网络的视觉模块外,还包含了设有激光雷达,毫米波雷达等多种传感器在内的感知元件.故由于测试所针对的角度差异,基于仿真场景的测试用例生成又可按照测试模块进一步分为:针对视觉模块的测试用例生成和基于多种传感器的测试用例生成.

Eykholt 等人[5]和 Araiza-Illan 等人[19]分别提出了基于真实场景的测试用例生成方法.Eykholt 等人[5]将对抗输入压缩为“对抗贴纸”作为测试用例,将对抗贴纸附着到目标物体(例如停车指示牌)上,会导致该目标物体被错误识别或者不被识别.Araiza-Illan 等人认为保证自动驾驶等机器人系统和真实环境的交互行为的正确性至关重要,因此他们使用 BDI(Belief-Desire-Intention)代理方法[20],对人机交互的场景进行建模,使用类人的行为和理性的推理为人机交互任务生成测试用例.代理由初始 Belief(初始知识),Desire(目标)和 Intention(根据初始知识,为达成当前目标确定的动作执行计划)定义.Araiza-Illan 等人使用 BDI 代理建模,表示了人机交互场景中的机器人代码、传感器、执行器和所处环境.之后,BDI 验证代理发送初始知识以激活一系列动作执行计划,并为其他代理提供新的目标,激活更多的动作执行计划,以此类推,每个具有不同初始知识的系统将激活代理中相应的动作执行计划序列.这些不同的动作执行计划将被记录下来,用于生成一个抽象测试的测试用例.该团队使用人机协作制造任务和家居助手任务进行建模和实验,分别生成 131 个和 62 个具体测试用例.结果表明,在上述两个场景中,使用 BDI 代理生成的测试代码覆盖率分别达到 92%和 86%;在人

机协同制造场景中,只有通过 BDI 代理的测试用例可发现违反某特定需求的情况;且 BDI 代理生成的测试用例覆盖了所有单元模块,而对比方法生成的测试用例很难达到预期的覆盖点.因此 BDI 在覆盖率和有效性方面优于伪随机生成的测试用例和通过模型检查自动机生成的测试用例.

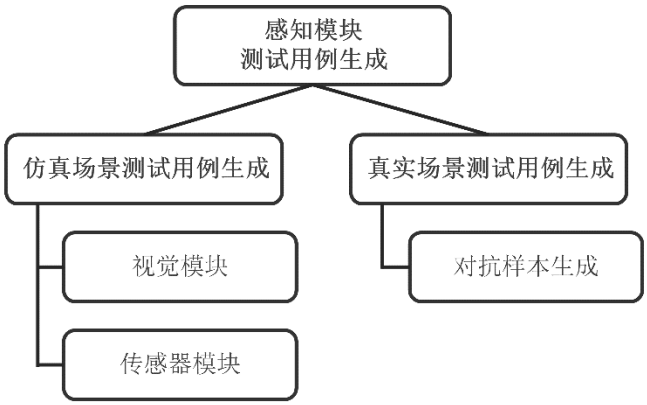

Fig.4 Summary of test case generation methods for perception module

图 4 感知模块测试用例生成方法汇总

尽管虚拟场景与真实场景的转化是一个待攻克的难点,虚拟场景仍具有开发难度低,复用率高,成本低等诸多优势,因此许多学者致力于生成虚拟场景对感知模块进行测试.

Johnson-Roberson 等人[14]认为人工标注大量训练数据这一耗时的过程已经开始阻碍深度学习工作的进展,他们构建了一个从复杂的仿真引擎中提取数据生成测试用例的全自动系统.其使用开源插件 Script Hook V 及 Script Hook V.NET 以 1Hz 的频率捕获有关场景的信息,从本地插件请求所有当前缓冲区,将缓冲区复制到托管内存中,并将其上传到运行 SQL-Server 的云计算机.引擎从主体半径内所有物体的 3D 位置获取粗略的边界框,再在模板缓冲区上进行轮廓检测,使得相互部分遮挡的检测形成单个轮廓,并在深度缓冲区计算检测到的轮廓内的平均深度和平均深度之间的距离,对包含的像素进行阈值化处理,生成一个新的边界框,进一步将边界框进行归类,生成了 3 个不同大小的不同模拟数据集.该团队通过绘制车辆质心的分布热力图比较了该数据集和真实数据集 Cityspaces[21]车辆位置分布情况,证明了其生成的数据集具有更加广的覆盖度.同时,该团队通过比较经过人工标注的数据集训练的模型,和经过自动生成标注的数据集训练的模型,在数据集 KITTI 上的表现,证明:仅使用大量模拟图像训练网络,获得了比仅用真实图像训练的网络更好的性能.表明了仅基于模拟的训练方法对于分类现实世界图像是可行的,无需与真实训练数据集图像进行任何混合.

Dreossi 等人[15]使用由图像生成器、采样方法及可视化工具构成的框架来生成测试用例.通过在图像生成器中布置基本对象(例如,道路背景,汽车)并调整图像参数(例如,亮度,对比度,饱和度)来获得图像.对象和图像参数的所有可能配置都定义了一个修改空间,在 3D 修改空间中抽象出所考虑的特征空间,每个点都对应于特定的图像.并根据用户需求而采用不同的采样技术来为图像生成器提供修改点(如:均匀覆盖修改空间的采样方法和基于主动优化的方法),以生成能够导致错误分类的测试用例,在确保抽象空间的最佳覆盖范围的同时最大程度地减少生成的测试用例数目.

2 决策模块

如果将感知模块比作自动驾驶系统的“眼睛”,那么决策模块就是自动驾驶系统当之无愧的“大脑”.在感知模块感知到外界的情况之后,自动驾驶系统会将多维度的感知信息传递给决策模块并进行数据融合.决策模块依据当前的路况信息进行路线规划和控制,并向相关控制模块(如方向盘、刹车)发出信号.决策模块在自

动驾驶系统中担任着至关重要的角色,其依托环境感知和导航定位系统的输出信息,进行行为预测和路径规划.目前自动驾驶系统中多使用基于数据的深度学习方法和基于运动模型的卡尔曼滤波器对周围的人,车及物体进行预测.在路径规划方面,较为成熟的方法有 Dijkstra 算法、Floyd 算法、A*算法、蚁群算法、基于分层网络的搜索算法、神经网络算法、实时启发式搜索算法、模糊控制以及遗传算法[22]等.

许多学者针对决策模块进行了测试,试图使用测试用例引导决策模块产生错误的决策.目前主要有生成致错场景[23][24][25][26],变异已有场景[28][29][30],预测场景置信度[32]和利用弹性行为[33]等方法.

2.1 相关研究现状

致错场景(Fault Scenario)表示能够使待测模块发生错误的场景,目前许多学者致力于研究出能够快速找出高效的致错场景进而使决策模块发生错误的判断.在寻找致错场景的过程中,一些优化算法和软件工程方法被应用到其中,例如遗传算法和压力测试等.Schultz 等人[23]提出了一种用来检测无人机器(无人机)控制系统的自动化方法,通过自动生成致错场景来验证无人机器的正确性.该团队认为,致错场景可以由机器初始条件(Initial Conditions)和一连串的出错规则(Rule)来定义,通过控制以上因素,就可以控制致错场景的生成.因此该文中使用遗传算法指导致错场景的生成,不断变化以上因素,找到真正使得控制器出错的场景.Schultz 等人使用 AUTOACE 作为模拟器,初始化了 100 种流行的致错场景并迭代调用遗传算法 100 次,总共生成了 10000 种场景.实验结果表明,该方法能够有效地检测无人机器的控制系统.

Tuncali 等人[24]设计并实现了一个自动生成驾驶车辆测试场景的框架,旨在寻找自动驾驶测试场景中能够导致碰撞的临界值,并定义了一个鲁棒性函数,用来表示一个场景的鲁棒性,该值越低,表示自动驾驶汽车在该场景越可能发生碰撞.该框架包含了四个部分:模拟器配置,模拟器引擎,S-TaLiRo 工具箱和鲁棒性函数.使用 MatLab 中的工具箱 S-TaLiRo 来获得使鲁棒性函数最小化的参数,并以此结果来指导模拟器配置的修改,生成下一个测试场景,并不断重复该过程,直到到达了模拟次数限制或者鲁棒性函数得到负值.Tuncali 等人设计了一个由两辆待测车,一辆无关车共三辆车组成的案例分析,实验结果表明该框架在该案例下成功地检测出致错场景.

Mullins 等人[25]提出了一种,用于搜索被测系统的性能临界测试用例的方法,利用自适应采样来智能地搜索状态空间中存在于不同性能边界上的测试场景.此外,使用无监督聚类技术,可以按场景的性能模式对场景进行分组,根据对诊断自治系统行为变化的有效性进行排序.该团队选取多目标导航场景下运行的模拟无人水下航行场景进行试验,令测试系统进行自适应搜索和边界识别.实验结果表明:该团队选择的搜索方法优于空间填充法及 LOLA-Voronoi 自适应搜索[27]等方法,且在基于高斯过程回归(GPR)的搜索算法和聚类边界识别算法应用于无人水下航行仿真的收敛性测试中,GPR 方法能够在 3000 个样本内找到所有边界的情况,而拉丁超立方采样方法需要大约 12000 个样本.

Koren 等人[26]通过自适应压力测试的方法,不断引入随机因素,能够找到使自动驾驶汽车发生碰撞的场景.在已有针对自适应压力测试的蒙特卡洛决策树搜索(Monte Carlo Tree Search)方法的基础上,该团队将问题转换为马尔可夫决策过程,并采用深度强化学习方法寻找目标场景.该文使用 Intelleigent Driver Mode[31]作为模拟器,在同一场景(汽车经过十字路口)下进行了测试,比较了两种方法的效果.实验结果表明,深度强化学习的方法能够在较少的模拟器调用下,得到更多可能错误的场景.与直接生成完整的测试场景不同,一些学者尝试在已有的测试场景上进行扰动,结合软件工程方法中的回归测试与蜕变测试,来达到对决策系统的测试.

现有的图像对抗样本生成技术在自动驾驶中具有局限性,涂黑、对像素进行修改等对抗样本扰动操作在现实中很难出现,Zhou 等人[28]提出了一种更实用的测试技巧 DeepBillboard,即对道路中的“广告牌”进行扰动,以探究其对自动驾驶汽车行驶方向的影响.该团队基于像素的颜色,对广告牌上的每个像素进行扰动,并保证扰动后的广告牌如同真实的广告牌,随汽车的视角等进行变化.该实验证明,扰动广告牌生成的对抗场景确实对自动驾驶汽车的航向系统有较大的影响,这种影响对虚拟场景和现实中的场景都适用.DeepBillboard

的缺陷在于,对广告牌的扰动过大,使得数据集中的广告牌几乎变为色块,在现实中几乎不会存在这样的广告牌.

Tian 等人[29]提出了 DeepTest,以探查在自动驾驶过程中的非自然行为和边界案例(corner case).DeepTest 考虑在驾驶过程中可能出现的各种因素,如雨,雾,光照,对图像进行线性变换(调整光照和对比度),仿射变换(裁剪,平移,放缩和旋转)以及卷积变换(变模糊,加入下雨,起雾效果).该团队证明神经元覆盖的变化和转向角度变化有一定相关性,因此可以用神经元覆盖作为场景生成引导标准.实验证明,DeepTest 生成的组合场景可以大幅提升神经元的覆盖率,并且可以侦察深度神经网络(DNN)模型的错误行为.这些场景同样可以用于修复模型中的问题,以提升模型的准确性.

基于蜕变测试的思想,Zhang 等人[30]提出了 DeepRoad,一个可以大量生成连续测试场景的非监督框架.该团队认为 DeepTest 通过仿射变换生成的场景过于简单,不符合现实中的天气现象,因此改进了生成测试场景的方式.DeepRoad 利用对抗生成网络板块,将晴天的行驶图片转变成为雨雪天气下的图片.实验证明,使用 DeepRoad 生成的场景具有更高的扰动质量,这些图片更容易引发自动驾驶汽车的不协调行为.

类似于“内省”行为,Stocco 等人[32]解决了估计 DNN 响应意外环境行为的置信度的问题,目的是预测潜在的对安全至关重要的不良行为,例如越界事件或碰撞.他们在工具 SelfOracle 中实现了一种基于置信度估计,概率分布拟合和时间序列分析的无监督不良行为预测技术:使用自动编码器和基于时间序列的异常检测重构驾驶场景,并确定不同条件下(如不同光照条件)的置信度边界,通过观察随时间变化的置信度下降趋势及时识别出意外情况,从而预测不良行为.实验过程中,该文使用 Udacity 的 DNN 模型和仿真环境,评估不同变体在预测注入的异常驾驶环境方面的有效性.结果表明,SelfOracle 可以提前 6 秒预测 77%的不良行为,其性能几乎比 DeepRoad 的在线输入验证方法高出 3 倍.

弹性是决策系统应对其正常运行时意外中断的能力,是决策系统当遇到意外的工作条件或不确定性的环境时,可以继续以安全的方式运行的能力,自动驾驶系统在其运行设计领域中的弹性行为至关重要.D’Ambrosio 等人[33]基于模型的系统工程(MBSE)方法开发了具有弹性的安全关键型自动化系统.MBSE 方法提供了对系统行为的保证,并可能通过使用严格的模型和广泛的仿真来减少对车载测试的依赖.MBSE 方法应用于开发弹性系统的两个方面:(1)通过使用弹性合同(Resilience Contracts)进行系统决策来确保弹性行为;(2)应用基于仿真的测试方法来验证系统能否处理所有已知情况,并针对潜在的未知情况验证系统.弹性合同利用基于合同的设计方法和部分可观察的马尔可夫决策过程(POMDP),使系统可以对感知环境中的潜在不确定性进行建模,从而做出更具弹性的决策.基于仿真的测试方法提供了一种结构化的方法,可以评估目标系统在各种运行条件下的运行情况,从而确认待测目标已实现了预期的弹性行为,并在 CARLA 仿真环境中显示了仿真结果.

2.2 测试用例生成

如图 5 所示,决策模块的测试用例生成根据生成类型可分为场景测试用例生成及图片测试用例生成.其中场景测试用例的生成方法主要有两种,第一种方法定义目标函数或奖励函数,通过对该函数取最大值或最小值指导测试场景的生成;第二种方法则是循环生成测试用例,依据遗传算法的思路根据上一个生成的结果来指导测试场景的生成.图片测试用例的生成方法包括基于数字图像的扰动(通过修改图片像素进行扰动)及基于真实环境的扰动(改变真实世界光照,角度等).在诸多生成测试场景的方法中,遗传算法被广泛应用.许多学者都采用遗传算法的思想,将测试场景量化,引入不同的变异算子来得到最优的测试用例.Schultz 等人[23]将场景定义为场景初始条件和一连串的出错规则,场景初始条件包含了车辆的初始位置,初始速度,初始行驶角度等等,初始条件会在模拟器启动时进行载入,将其作为依据初始化场景.每个出错规则包含一系列的触发器和错误类型,当一个规则中的所有触发器都被触发时,则标志着由这个规则定义的错误发生了.Schultz 等人通过遗传算法不断生成测试场景,通过模拟器验证生成的场景的优劣,并指导遗传算法进一步生成测试场景,如此往复,直到到达设置的循环次数.

Tuncali 等人[24]将寻找致错场景的问题转换为最小化函数问题,即在给定的虚拟环境,车辆,无关物体和环境信息下,设定一个鲁棒性函数,进而寻找能够最小化鲁棒性函数的输入参数,即寻找指定的场景初始设置,车辆和物体设置.Tuncali 等人认为,鲁棒性函数的选取对于能否快速准确的找到致错场景起着至关重要的作用.为了找到导致车辆碰撞的临界场景,Tuncali 等人使用碰撞时间(TTC)和车辆的相对速度的和作为鲁棒性函数的定义.Tuncali 等人认为这样的鲁棒性函数能够更有效的找到导致车辆碰撞的致错场景.

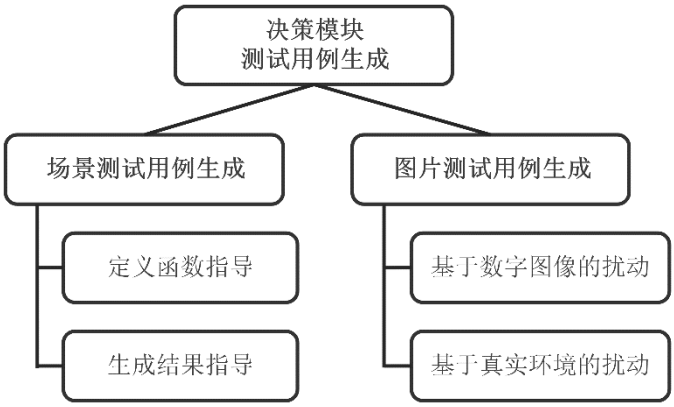

Fig.5 Summary of test case generation methods for dicision module

图 5 决策模块测试用例生成方法汇总

Mullins 等人[25]提出了一个智能搜索和识别测试用例的方法,能够有效筛选并生成自治系统的性能临界测试用例.该方法分为两个主要阶段:搜索和识别.在搜索阶段,通过使用自适应采样或主动学习方法选择自主系统仿真运行的新测试用例.该过程利用高斯回归来建模自治系统的性能,并优先选择可能指示性能边界的区域,充分覆盖边界区域,同时最小化模拟的数量.在识别阶段,利用搜索阶段产生的测试用例,利用无监督的聚类演算法来辨识场景的性能模式,将测试用例按其性能模式分类,识别出性能模式之间的边界,为执行自治系统的不同底层决策过程的多目标无人水下航行任务生成一组测试用例.

Koren 等人[26]将基于场景的自动驾驶测试问题转换为了马尔可夫决策问题并设计了相应的奖励函数,奖励函数将每次测试的输出转化为奖励值,并以此作为依据指导测试用例生成器.该团队采用了两种方法作为测试用例生成器,第一种是蒙特卡洛决策树搜索算法,另一种是深度强化学习算法.蒙特卡洛决策树算法通过奖励值生成新的随机种子并生成场景参数,深度强化学习则利用奖励值生成新的高斯分布,进而得到场景参数.该团队将这两种测试用例生成器应用到压力测试当中,利用马尔可夫决策模型,不断生成新的测试场景.

相比于生成整个测试场景,一些学者致力于生成基于图片的测试用例,一方面基于图片的测试用例更加灵活,另一方面,决策模块在关键帧的决策行为决定了整个场景的决策行为,因此基于图片的测试用例也能很好的测试决策模块的性能.在基于图片的测试用例生成方法中,软件工程中的回归测试和蜕变测试被广泛应用,同一场景在不同的天气,背景下,决策模块应当采取一致的行为,这与回归测试和蜕变测试的思路不谋而合.基于此,Alhaija 等人[34]提出了新的测试用例生成方法,即使用计算机视觉手段生成合成的强化测试用例以提升模型准确率.该团队认为,一个完整的自动驾驶场景图片需要三部分组成,即高清 3D 车模型,周围环境地图和真实物理环境渲染(光照,反射等).文中提供了算法,将平面照片改为鸟瞰图,在鸟瞰图中挑选车辆可以选择的位点,并采样放置车辆,对车辆模型在环境地图的条件下进行渲染.经过实验验证,该算法合成的图片有更高的质量,且在合成图片上训练的模型相比原始数据能达到更高的精度.

Tian 等人[29]提出 DeepTest,旨在生成在现实生活中可能出现的图像输入,因此基于现有的自动驾驶采集的种子图片,利用多种变换模拟现实驾驶中可能应对的情况.线性变化,包括调整图片的亮度和对比度两种途

径,用于模拟遭遇强光、镜头失真等景象;仿射变换包括平移、缩放、水平剪切和旋转,用以模拟物体的位置变化,特殊的拍摄角度和拍摄异常等情况;卷积变换为图片加入雾气和雨天的效果,用于模拟在特殊天气下各种交通情况.该团队使用贪婪算法,寻找有效的图像变换组合,提升生成图片对神经网络的神经元覆盖率,从而保证深度神经网络模型的安全性和测试的充分性.

考虑到 DeepTest 的变化过于简单,生成的图片的较为粗糙,难以模拟驾驶过程中捕捉的真实图像,Zhang 等人[30]提出了 DeepRoad.为了提高生成图片的质量,DeepRoad 框架使用了一种基于对抗生成网络的测试用例生成方法.DeepRoad 框架使用无监督图像转换板块,将两个目标域(如晴天数据集和雪天数据集)投影到相同的潜在空间,经过训练,将晴天图像合成为雪天图像.经过对比,DeepRoad 生成的图片更贴近于实际驾驶过程中采集的视频和图片,比 DeepTest 生成的图片有更高的质量.

在真实场景中捕捉的测试用例通常不会存在全局性的扰动,即真实场景中的扰动通常是由图片中的某个实体或者某部分所引起的.为了模拟真实世界扰动的这一特性,Zhou 等人[28]选取了道路中常见的公告牌进行像素级别的对抗扰动.在扰动过程中,保证对抗不会溢出到公告牌面积之外的区域,且随着视角和距离变化.实验证明,这类基于公告牌扰动的对抗样本确实会对车辆的转向造成一定影响.

2.3 覆盖度量指标

覆盖度量是指利用覆盖率对测试的完整性和有效性进行衡量的手段.覆盖度量指标是传统软件工程测试领域中一个重要的评价指标,其能够评价测试用例集合是否有效且完备,同时也能指导测试用例生成,即通过生成特定的测试用例来达到更高的覆盖度量指标.目前在人工智能领域,已有许多覆盖度量指标被提出[35],包括神经元覆盖,分类准确率,测试覆盖率等,但由于自动驾驶测试的复杂性以及自动驾驶场景的多样性,人工智能领域的覆盖度量很多都无法应用到自动驾驶领域中.Pei 等人[36]引入的神经元覆盖概念是为数不多的能够应用到自动驾驶领域中的覆盖度量指标.现有的实验不能确定其它人工智能领域的覆盖度量是否能够指导自动驾驶测试(即不能确定 AI 测试的覆盖度量是否和自动驾驶测试的充分性相关).此外,从应用的角度看,自动驾驶测试由于拥有特殊的使用场景,覆盖度量指标聚焦于具体驾驶场景的覆盖,因此演化出了精细到情境(situation),场景(scenario)甚至需求(requirement)的高层次覆盖标准,这些标准虽然具有一定指导意义,但是缺乏对驾驶场景国际化统一的定义,且驾驶场景粒度粗糙并难以量化,导致目前提出的标准在实际应用中困难重重.更多的关于自动驾驶系统的覆盖度量指标往往针对整车测试,参见第 4 节关于整车测试的描述.

为了更有效的测试深度网络模型,Pei 等人[36]首先引入了神经元覆盖的概念.一个深度神经网络是由成百上千个神经元构成的,当一个神经元的输出值达到特定阈值,则认定该神经元被激活.神经元覆盖是统计激活神经元在总体神经元的占比.以神经元覆盖率指导生成测试用例,可以挖掘出更多神经网络的边界行为和错误行为.该团队利用 DeepXplore 框架,在 Nvidia DAVE 自动驾驶系统的模型中,找到了诸多导致导向系统错误转向的危险场景.

为了更充分地测试路径规划板块,Laurent 等人[37]提出了权重覆盖.该团队使用了加权损失函数,为路径规划的安全性、合理性、合规性和舒适性进行打分.文中设置了六个与驾驶情境相关的权重常数(例如加速度等),当车辆的某一驾驶决策超出了特定阈值,权重常数将会以累加或累乘的方式计入现有的加权损失值中.该团队按照倍率调整权重常数生成路径规划器,检查该变异路径规划器的路径是否会被现有的测试套件侦测出,一旦侦测出不符合测试预言的路径用例,则意味着调整的权重常数被覆盖.实验证明,权重覆盖能够有效指导对路径覆盖器的测试的充分性.

3 综合功能模块

除了单独针对感知模块和决策模块进行测试,许多学者致力于针对一些综合功能模块进行测试,例如高级驾驶辅助系统(ADAS),这些模块可能包含感知层,决策层和控制层等多个单元,协调完成一项具体的功能,例如自动泊车、自动巡航(ACC)、紧急制动(AEB)、行人保护(PP)、交通标志检测(TSR)、盲点检测系统(BSD)、

前向碰撞预警系统(FCW)、车道偏离告警(LDW)、抬头显示器(HUD)、汽车夜视系统(NVS)、智能车速控制(ISA)、智能大灯控制(AFL)、泊车辅助系统(PA)和行人检测系统(PDS).多数研究通过基于搜索的方法生成测试场景来针对自动驾驶系统进行测试.图 6 总结了基于搜索的测试场景生成的方法.基于搜索的方法或使用进化算法,或使用多目标搜索算法,均通过不断迭代来生成关键测试场景(Key Scenario).在迭代的过程中,相关研究往往使用一个初始测试场景作为生成种子,使用模拟器和适应性函数(Fitness Function)来生成能够表示该场景的优劣的适应度(Fitness Value),接下来通过选择器与搜索器生成变异后的测试用例,并不断重复.相关研究通过辅助预测器和辅助分类器,帮助整个方法更快速更准确的生成相关测试场景.除了基于搜索的方法,一些学者还提出了基于硬件在环,算法在环的测试方法,针对自动驾驶系统中综合功能模块进行测试.

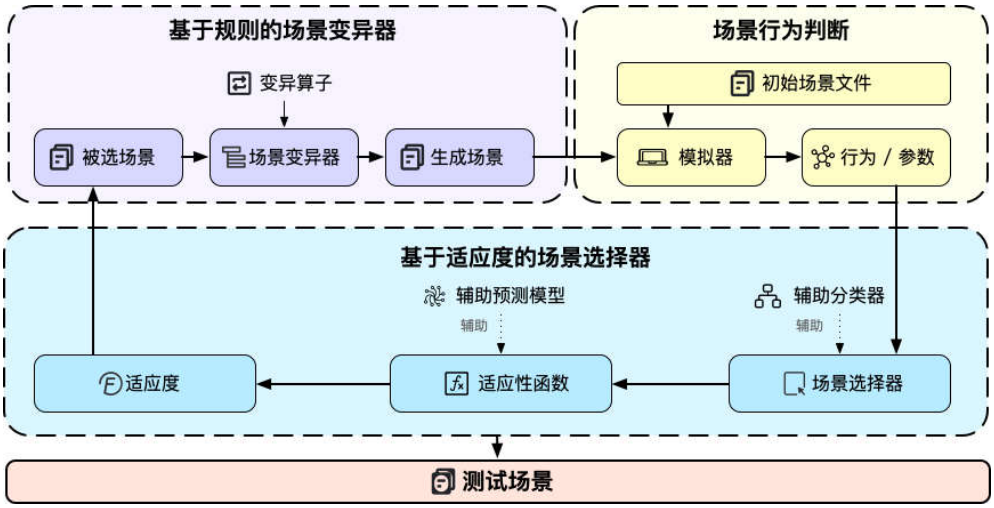

Fig.5 Method of search-based test scenario generation

图 5 基于搜索的测试场景生成方法

3.1 相关研究现状

Bühler等人与Abdessalem所在团队发表了多篇基于搜索算法的针对自动驾驶系统综合功能模块的测试方法.该团队率先使用进化算法对自动泊车[38][39]和协助制动[39]两个自动驾驶系统中的功能模块进行自动化测试.他们通过定义合适的能够表示待测系统质量的适应度函数,将测试问题转换为最优化任务,不断迭代测试用例,不断改进测试用例的最优化目标函数,最终得到致错场景.Bühler 使用自动驾驶系统中的自动泊车系统和协助制动系统作为待测试系统,为两个系统分别定义了目标函数,使用人工测试,随机测试和基于进化算法的测试对两个系统进行评估.实验结果表明,基于进化算法的测试算法能够更有效更快速地生成准确的测试用例.

在 Bühler 等人的相关研究的基础上,Abdessalem 等人提出了一种使用基于进化算法的针对自动驾驶辅助系统的测试算法[40].该团队首先给出了高级驾驶辅助系统(ADAS)的形式化表示,并定义了关键场景,同时给出了进化算法中需要的目标函数.该团队使用 NSGA-II[41]算法作为多目标搜索算法并采用模拟二分交叉算子[42][43]作为进化算法中测试用例迭代变化的交叉算子.与以往的进化算法不同,本文引入了决策树分类模型并定义了一系列分类标准,通过决策树分类的结果来指导进化算法从而得到更加精准的关键场景.本文在实验中使用 NSGA-II 算法作为基准,比较了基准算法和使用决策树模型的改进方法得到的三个进化算法中的评估指标[44],实验结果表明,使用决策树改进后的算法能够比基准算法多生成 78%的更加清晰准确的关键场景,同时改进算法能够得到关键场景的空间特征,为相关研究人员辨认关键场景提供了指导.

类似于进化算法的不断迭代过程,多目标搜索算法也同样是在一个大的搜索空间中不断搜索符合要求的测试用例,其差别在于进化算法需要提供一个适应度函数,用来指导变异的方向,而多目标所搜算法则需要定义距离函数,来指导搜索的方向.

Abdessalem 等人认为现有的仿真工具缺乏必要的智能和自动化能力来指导仿真场景的选择,且执行模拟方案在计算上是昂贵的.因此该团队在 2016 年提出了通过将多目标搜索与基于神经网络开发的替代模型相结合来提供高级驾驶辅助系统的测试方法[45],并使用多目标搜索指导测试高级驾驶辅助系统的关键行为.该文基于行人检测视觉(PeVi)系统进行研究,测试结果表明:基于搜索的测试技术优于随机测试生成策略;将多目标搜索与替代模型相结合,可以在有限的时间内提高生成的测试用例的质量;基于搜索的测试技术能够生成表明 PeVi 系统潜在错误的各种测试用例.

Abdessalem 等人在 2018 年提出了一个基于多目标搜索的寻找自动驾驶车辆中功能覆盖错误的方法并实现了一个工具 FITEST[46].FITEST 旨在解决自动驾驶车辆多个模块在某个场景下针对同一控制器做出不同指令,指令集之间发生覆盖从而导致错误的问题.本文将问题定义为多目标搜索算法,定义了三种距离:覆盖距离,致错距离和致错重写距离.FITEST 基于搜索算法 MOSA[47]开发,通过最小化上述距离来寻找功能覆盖错误.FITEST 分别基于覆盖距离,致错距离和上述三种距离的混合距离,在两种高级辅助自动驾驶工具(包含四种功能:自动紧急刹车、自适应巡航控制、行人保护和交通标志检测)上进行了搜索测试,实验表明基于混合距离的搜索能够在同样时间下得到更多的功能覆盖错误(平均 5.9 和 7.2 个错误).

基于软件在环(Software in the Loop,SiL)和硬件在环(Hardware in the Loop,HiL)的测试方法是自动驾驶测试中常见的方法,不同于寻找基于搜索的方法,在环测试方法往往需要针对测试对象定义出一个完备的综合测试架构,并评估测试对象的各种指标.Gietelink 等人[48]提出了一种用于高级驾驶辅助系统(ADAS)的设计和验证的方法,该方法将测试车辆放在底盘测功机上,这是一种可以通过车辆模型来测试模拟道路行为的滚筒试验台,其他道路使用者用轮式移动机器人表示.他们通过自适应巡航控制系统和前方碰撞预警系统的测试结果,证明了该方法在高级驾驶辅助系统的开发过程中的传感器验证,快速控制原型,模型验证,功能级别验证,控制算法的微调,生产验证试驾等多个阶段具有帮助效果.该方法的优势在于具有可重复性和高灵活性,能够提高测试的安全性并同时节省验证过程的时间和成本,是一种与现有开发过程互补的高级驾驶辅助系统开发新方法,与模型在环(Model in the Loop,MiL)仿真和测试驱动器形成有效联系.

Belbachir 等人[49]使用 Pro-SiVICTM[50][51]作为模拟器,定义了高级驾驶辅助系统(ADAS)的仿真驱动评估架构 Ev-ADA.该架构的评估标准包括车道检测错误、行人检测错误、轿厢位置检测错误、汽车定位错误、路径规划错误、控件/命令错误、驾驶员安全估计和驾驶员舒适度估算,可用于评估任何驱动系统.以上每个标准都可计算出单独的分数,并汇总一个全局分数进行加权求和,分数值越接近值 1,则证明算法的质量越高.

随着自动驾驶测试方法不断进步,一些学者同时将目光放在了自动驾驶技术修复方法上.在 2018 年对高级辅助驾驶系统中功能覆盖错误的检测的基础上,Abdessalem 等人于 2020 年提出了一个能够自动化修复由功能覆盖导致的自动驾驶系统错误的方法[52].该团队首先针对包括行人保护和自动紧急刹车在内的四个辅助驾驶功能模块定义了相关的功能覆盖优先级规则以及安全要求,接着应用成熟的错误定位算法 Tarantula[53],通过重新定义可疑值函数来确认发生错误的代码行数.在确定了错误位置后, 该文采用变异的方法生成补丁候选集,并采用搜索的方法在候选集中选取优秀的修复补丁并归档,接着继续生成新的候选集直到到达设置的搜索条件.最后该文最小化选出来的候选补丁集合作为最终的修复结果.该文采用了工业界的两个高级辅助驾驶系统对方法进行了测试并设置了对比实验,实验结果表明,该方法能够有效地在两种工业产品上找到并修复错误,并优于其他方法.

3.2 测试用例生成

相比于感知模块和决策模块的测试场景,针对综合功能模块的测试场景往往需要更复杂的设置和更多的参与者.因此,针对综合功能模块的测试场景往往拥有更大的数据空间,因此如何快速在庞大的空间中找到

关键的测试场景至关重要.图 5 中提到的基于搜索的方法是解决这一类问题的较为有效的方法.

Bühler 等人[38]使用进化算法来生成针对自动泊车系统的测试用例.该团队为进化算法设计了两种目标函数:1)车辆和碰撞区域的最小距离.该团队采用侧向停车作为测试场景,且前后方均有车辆.其划分出了目标车不可行经的区域并将其定义为碰撞区域,使用碰撞区域与行车轨迹之间的最小欧式距离作为衡量标准;2)车辆与碰撞区域的面积,即行车轨迹和碰撞区域边界包围起来的闭合图形的面积,当车辆轨迹穿越了碰撞区域时,所包含的区域面积为负值.基于这两种目标函数,该团队通过不断使用进化算法找到致错的测试用例.在之前的基础上,该团队于 2008 年的研究[39]加入了对协助制动模块的测试.与先前的进化算法相似,该团队加入了针对协助自动的目标函数的定义:目标车速度,由碰撞时间和刹车制动力的乘积的积分得到.刹车制动力由驾驶员和协助制动模块共同决定.

Abdessalem 等人在 2016 年提出的方法[45]中使用多目标搜索的方式来指导测试用例的生成.为了生成能够导致错误的边缘测试用例,该团队定义了三部分目标函数:1)最小化车辆和行人的欧式距离,文中认为该距离越小,相关的场景越容易出错;2)最小化行人和危险区域的距离,危险区域的定义为待测车辆前方一定距离内的矩形区域,既待测车辆即将通过的区域;3)最小化碰撞时间(TTC).为了解决在模拟环境中计算上述三个目标函数的低效率问题,该团队设计了一个替代模型,并使用已有的模拟环境数据来训练替代模型,使得替代模型能够预测出目标函数的输出.该团队将替代模型应用到 NSGAII 算法中,不断进行搜索,进而生成更多边缘测试用例.

Abdessalem 等人在之前的针对行人检测视觉系统(PeVi)的研究后,又提出了针对高级驾驶辅助系统中自动紧急刹车、自适应巡航控制、行人保护、交通标志检测等模块的基于多目标搜索的测试方法[46].该团队为搜索算法定义了三种距离:1)覆盖距离.由于功能重写模块是白盒的,该文采用分支覆盖的标准来衡量功能重写模块,并以此为依据定义了覆盖距离;2)错误距离.错误距离表示测试用例距离安全标准的距离,测试用例越可能发生错误,则错误距离越近;3)致错重写距离.致错重写距离标志了由功能重写模块引发的距离,若某个测试用例导致的错误的发生不是由于功能模块发生了错误,而是由于功能重写模块发生了错误的优先级覆盖,则该测试用例具有较低的错误距离.该文通过最小化上述三个距离来指导多目标搜索算法,不断生成测试用例.

Abdessalem 等人在 2018 年提出的方法[40]仍使用多目标搜索算法来生成测试用例,与以往的方法不同,该文引入了决策树分类模型来得到更加精准的关键场景.该文将目标函数定义为:1)行人与车辆的距离;2)碰撞时间,通过最小化行人与车辆的距离并最大化碰撞时间,来获得更准确的测试用例.在 NSGA-II 算法的基础上,该文使用决策树模型,通过定义不同参数的范围,来规范出满足关键场景的特征空间,并得到含有较大比例的关键场景的聚类.在每次迭代的过程中,决策树模型能够在保证关键场景占比较高的情况下简化搜索空间,使得搜索算法找到更精准的测试用例.

类似于上述的基于搜索的方式生成测试用例的方法,Gambi 等人[54]将视频游戏中生成连续内容的技术和搜索算法相结合,设计并实现了一个针对自动驾驶系统中自动巡线模块的自动化测试方法,该方法旨在生成更有效的测试场景.该文首先将视频游戏中的连续内容生成技术应用到场景衍生中,根据一定的规则生成道路网络.接下来使用车辆行驶轨迹和车道线中央的距离作为目标函数进行搜索,并通过变异得到新的测试用例.变异方法包括道路片段变异,切断并分别组合两段道路以及道路迁移覆盖.该文使用 BeamNG.research 作为模拟器,对方法进行了测试,实验结果表明,该方法能够有效地生成测试用例并且暴露出与安全相关的问题.

4 整车测试

相比于针对个别模块的测试,许多学者将研究目标放在了整车测试上.基于整车的测试不考虑单独模块或者某个功能,而是对整个自动驾驶系统做出综合判断.因此基于整车的测试不考虑具体哪个模块发生了错

误.针对整车的测试往往使用场景作为测试用例,在虚拟环境中进行测试.关于场景的定义,Ulbrich 等人[56]给出过总结.Ulbrich 等人整理了多篇文章对于场面(scene)、情景(situation)和场景(scenario)的定义,并对这三个概念总结出了一个整体的描述.Scene 由环境,动态物体,所有参与者和观察者以及上述物体之间的关系组成.Situation 总是从一个物体的主观视角出发,根据某个信息的选择,依据某个短暂且临时的目标从 Scene 中衍生出来.与 Scene 相比,Situation 中的物体往往是与主观物体相关的.Scenario 包含了 Scene,同时还有一些相关物体的动作、事件、目标以及属性.Scenario,功能范围,系统边界和期待行为共同组成为一个用例(use-case).本文提到的场景均为 Scenario.

目前国内外针对测试场景的定义给出了相关的标准[55].测试场景要素往往分为测试车辆要素、静态环境要素、动态环境要素、交通参与者要素和气象要素等.其中测试车辆要素又包含了待测车辆详细的物理信息,例如重量、性能、位置和运动状态等.静态场景要素中包含了障碍物、交通标识和道路等路况信息.由于人工标注真实场景的开销和代价过于庞大,而针对自动驾驶的整车测试往往需要使用相关的研究方法生成能够导致待测自动驾驶系统发生错误的测试场景,于是自动化地生成测试场景成为了该方向相关学者们的研究目标.

4.1 研究现状

目前针对整车测试方法主要分为两种,一种是通过结合真实场景和交通数据来生成仿真场景的测试方法,一种则是通过构建完备的测试框架,对自动驾驶系统做出评估.基于真实场景的仿真场景衍生能够更加贴合真实世界中的驾驶场景,更加具有真实性,相关学者通过引入在真实世界中采集的地理信息,例如 GPS 等,衍生出更加具有说服力的测试场景.

Zhang 等人[57]提出了一种基于图像序列和道路地理信息系统(GIS)数据的交通场景建模方法,并在此基础上,设计了一个交通场景模拟器,利用该模拟器对自动驾驶汽车进行性能测试和自动驾驶任务评估.他们利用车载摄像机从实际交通场景中获取视频序列,然后将其与 GPS 获取的视点位置序列相结合,得到视点图像序列,随后按时间顺序组织视点图像序列.同时,该文引入道路 GIS 数据,定义交通场景的三维结构.实验过程中,通过避障驾驶及自动驾驶能力等级评定来对生成的场景进行检验.实验结果显示,该方法生成的场景能够有效地评价出待测车的自动驾驶能力.

Althoff 等人[58]提出了一种自动创建临界场景测试用例的方法,该方法首先通过显式计算自动驾驶车辆的可驾驶面积来量化解决方案空间.通过指定运动规划问题,对包括自动驾驶车辆在内的其他交通参与者的初始状态进行优化,来减小求解空间,直至得到所需的可驾驶区域大小,实现测试用例所需的临界度.实验过程中,该团队针对一个刻意简单化的场景及两个来自 CommonRoad[59]的不同基准场景,分别使用不同数量的其他交通参与者来进行实验.该文对简单场景中自动驾驶汽车的速度,两个基准场景的速度及交通参与者的初始状态进行了优化,三种场景中自动驾驶汽车的可驾驶区域均得到了相应减小,得到了具有所需可驾驶区域临界度的情况.

Gambi 等人[60]认为真实车祸场景非常适合作为测试场景,但真实碰撞中收集到的数据不足以完全再现场景,故该文提出了基于碰撞报告的自动碰撞场景构造器 AC3R 来生成测试用例.此框架依靠自然语言处理(NLP)方法从车祸现场的警察报告中提取有关信息,将文本概念映射到场景细节,创建车祸的抽象表示,生成可在驾驶模拟器中实现车祸动态的代码,并从模拟中导出测试用例.实验过程中, 该文检查自动驾驶汽车在避免碰撞的同时是否达到其最终预期位置,并预测多个驾驶能力指标,以调查 AC3R 成功生成场景模拟的数量及与报告的匹配程度.实验结果表明:AC3R 能处理过半(62.6%)的数据库样本报告,且多数情况下(78.0%),AC3R 生成的用例的预期损坏报告与碰撞有关的所有虚拟汽车的预期损坏报告相同.

Fremont 等人[61]提出了一种基于形式化验证的自动驾驶汽车测试方法,能够有效地将形式化模拟测试转变为真实世界中的整车测试.本文首先使用形式化语言来表示测试场景和自动驾驶汽车的安全属性,并使用相关模拟器,验证工具和算法来判断自动驾驶汽车在测试场景中是否会发生错误.最后将生成的虚拟测试场

景映射到真实场景中,并使用整车进行测试.本文使用 2018 Lincoln MKZ Hybrid 作为测试车辆,选取的场景为自动驾驶汽车转弯经过十字路口,遇到过马路的行人.实验表明,62.5%的虚拟致错场景转变的真实场景在测试中发生了错误,93.3%的不发生错误的虚拟场景转变的真实场景在测试中也没有发生错误.

与基于场景生成的测试方法不同,基于构建完备的测试框架的测试方法能够对自动驾驶系统做出系统的评估,可以在更高的层级对自动驾驶系统做出综合的判断.

Tuncali 等人[62]提供了一个测试用例生成与自动伪造方法兼容的测试框架 Sim-ATAV,该框架可用来检查包括机器学习组件的自动驾驶系统闭环特性.该文提出了一种测试方法,称为覆盖数组和需求伪造方法,以自动识别有问题的测试方案.该框架使用 Webots[63]进行环境建模,SqueezeDet[64]作为感知系统,选择一个特定的自动驾驶场景进行测试.实验结果表明 Sim-ATAV 框架可用于提高自动驾驶系统的可靠性.Tuncali 等在 Sim-ATAV 中实现了三种测试策略,分别为全局统一随机搜索、覆盖数组和全局统一随机搜索、覆盖数组和模拟退火.由于全局统一随机搜索不能被引导到测试用例的关键区域,该文使用 ACTS[65]工具生成的覆盖数组选取最小数量的测试用例以确保对所有组合进行测试.然而覆盖数组仅能对离散变量的选取进行指导,所以提出第三种方法使用覆盖数组评估离散变量并使用模拟退火搜索连续变量.实验证明覆盖数组和模拟退火生成的测试用例在保证覆盖范围的同时拥有更低的鲁棒性.

Rocklage[66]提出了一种对自动驾驶汽车进行验证的方法.该文引入了一个新的验证空间概念,该概念基于时空状态网格来生成动态交通参与者的运动.时空状态网格基于车辆所在空间的高清地图或人为手动操作绘制,网格中格点表示位置-时间元组,这个空间一开始是稀疏的,但随着车辆遇到新的情况,它变得越来越密集.研究人员可以对形成的有向图应用深度优先,广度优先或回溯等算法测试图形边缘与本车计划轨迹相交的场景,并逐步覆盖验证空间.该文建议自动驾驶车辆的一般控制体系结构中添加一个新的安全模块,安全模块将使用验证空间数据库中的信息来确定当前情况是否得到验证及当前情况附近的连续区域被覆盖的程度.该文使用人类睡觉时做梦进行类比,建议自动驾驶车辆在停车时“睡觉”,自动驾驶车辆的安全模块还包含一个仿真模块,用来在“睡觉”时模拟白天无法自动处理的情况和场景.该验证方法的优势在于可以通过迭代达到 L5 自动化水平,车辆能够将干预请求发送给驾驶员,从而消除驾驶员始终处于循环状态的需求.

Aramrattana 等人[67]提出了一种用于测试和评估协作智能交通系统(C-ITS)应用程序的仿真框架,该框架结合了驾驶,交通和网络模拟器.协作智能交通系统(C-ITS)的目标是更安全更高效的交通系统.该文给出了协作式自适应巡航控制(CACC)应用程序的示例,并将其用于仿真框架的测试,分别为单排车辆增大车距,双排车辆合并为单排车辆,驾驶员驾驶三种场景.测试结果说明了在循环中使用人工驱动程序测试不同控制策略的能力和仿真框架的灵活性,并证明了仿真框架中的模拟器是同步的.该文认为由于涉及人类驾驶员切换驾驶模式,交通模拟器未考虑横向加速度以及仿真框架正确性验证等原因导致该框架中的每个仿真器都没有被充分利用.

4.2 测试用例生成

针对整车测试的测试场景生成方法主要以结合真实场景数据为主,通过一系列的转化和衍生方法生成新的仿真场景.部分学者也尝试了使用基于搜索的方法,基于回归测试的方法和基于多实体(Agency)的方法生成相关测试场景.Zhang 等人,Gambi 等人,Fremont 等人结合了真实场景去生成测试用例.

Zhang 等人[57]提出了一种基于视点图像序列的自动驾驶测试用例生成方法:首先,使用 GPS 坐标确定场景要素的地理空间位置,使用车载摄像机获取路面上任意一点的图像数据,然后将其与 GPS 获取的视点位置序列相结合,得到视点图像序列.按时间顺序组织视点图像序列,引入道路 GIS 数据,通过道路网络确定道路的三维几何信息,并进一步进行视图转换,将交通场景可视化,生成自动驾驶测试用例.

Gambi 等人[68]提出了使用基于碰撞报告的自动碰撞场景构造器 AC3R 来生成测试用例,其生成过程分为四个阶段:信息提取,轨迹规划,仿真生成及测试生成.其中,信息提取阶段通过语法相关性分析,计算碰撞描述的每个语句中每对单词之间的语法相关性,并匹配特定领域的本体(如环境、交通参与者、行为及事故),

提取道路属性及车辆属性,将文本中的概念充分映射到撞车场景的细节.在轨迹规划阶段,AC3R 将冲击点放在参考坐标空间的原点上,并为每个模拟车辆的每个驾驶动作置一个新的航路点,不断添加航路点至包含所有驾驶行为,最后调整冲击点的位置并更新道路几何形状,创建出车祸的抽象表示.仿真生成阶段则使用了专门用于模拟交通事故的仿真引擎 BeamNG.research,使得 AC3R 能够使用在规划虚拟汽车轨迹时计算出的几何数据来以程序方式生成道路并放置路标,生成可在驾驶模拟器中实现车祸动态的代码.并在测试生成阶段,由模拟中导出测试用例.

Fremont 等人[61]使用形式化语言来描述仿真测试的过程,通过形式化测试场景和安全因子并使用 VERIFAI[68]工具对相关形式化表示进行验证,区分出安全和不安全的测试用例,进一步通过聚类算法选择出合适的测试用例进行虚拟场景测试.该文使用 pulley-based 4Active surfboard platform (SB)来模拟行人,以将选择出的测试场景转换成为了真实的整车测试.

Rocklage 等人[69]提出了一种在虚拟仿真环境中为自动驾驶回归测试自动生成测试用例的方法,着重研究了生成其他交通参与者的运动而不损失一般性的问题.该团队将组合交互测试方法与一个简单的轨迹规划器相结合作为可行性检验器,以生成有效的可变覆盖测试集.底层约束满足问题采用回溯算法解决:首先定义了不同类型的交通场景及虚拟环境中回归测试的相关术语,使用“四层概念”作为基础,创建静态或混合场景,使用组合互动测试 CIT[65]相互组合找到参数值,实现道路、静止物体、气象效应、动态对象及本车的参数化,使用以上获得的参数来生成测试用例.该方法为有关自动驾驶回归测试领域的研究提供了基础,但仍需要优化以获得更高的效率.且该方法只能沿预定义的路径创建运动,若能在一般概念中加入路径发现问题,将允许更多的自动化.

类似于针对综合功能模块的基于搜索的测试方法,Althoff 等人[59]提出了通过显式计算自动驾驶车辆的可驾驶面积来量化解决方案空间的自动生成临界场景测试用例的方法,该方法结合可达性分析和优化技术来确定解空间的大小并缩小它,以得到满足需求的测试用例:首先指定运动规划问题,然后定义自动驾驶车辆的可驾驶区域.为了求得该区域,需要对包括自动驾驶车辆在内的其他交通参与者的初始状态进行优化,直至得到所需的可驾驶区域大小.其中,当不仅考虑静态障碍物,还同时考虑其他交通参与者时,须计算其初始状态变化对其未来占用率的影响,将其他交通参与者未来可能占用的空间从自动驾驶车辆的允许空间中移除,以避免碰撞,进一步有效计算可行驶面积.为了求解该有限优化问题,该论文将时间等距离散,并用矩阵表示法近似表示优化问题的离散时间,再将其重写为一个二次规划问题,然后使用二进制搜索对其进行扩展.最终得到满足临界度的测试用例.

Chance 等人[70]提出了基于多实体(Agency)的测试用例生成方法. 实体是指在测试环境中与自动驾驶车辆交互的动态实体,在这篇文章中,该团队将行人作为实体,对行人设置各种行为动作,生成有效的测试用例.行人的动作分为两类,一类是随机动作,即行人有一定的随机行为,并会在随机时刻穿过行车道;第二类是实体指向行为,其中临近行为(Proximity)是指行人在车辆到达一定半径范围内穿越行车道,选举行为(Election)是指选择离车辆最近的行人穿越行车道.为了提升测试的效率,作者以车辆和行人行进时长为标准给测试用例打分,耗时越长分数越低,以促进短时长测试用例形成.实验证明,基于实体的测试用例生成方法比伪随机方法生成两倍的有效测试用例.

4.3 覆盖度量指标

对于整车的质量测试和安全保证往往拥有更宏观的测试标准.Tahir 等人[71]将常见的基于整车的覆盖标准分为三类:即需求覆盖(Requirement Coverage),场景覆盖(Scenario Coverage)以及情境覆盖(Situation Coverage).无论是需求,场景还是情境测试,都是在自动驾驶质量和安全保证方面的宏观定义,这也意味着,上述三种覆盖标准会随驾驶和测试场景的不同而改变,而非遵循一套统一的定义.

需求覆盖是指,待测系统必须满足一系列的可被接受的需求.需求覆盖通常需要相关领域专家以启发性的方式指定相关需求,这可以有效指导自动驾驶产品的构建,但是,需求覆盖在测试中的使用相对困难.首先,

粗粒度的需求对指导测试帮助不大,例如“自动驾驶汽车不可以违反交通法规”的需求很难实现,也没有标准能验证是否实现该需求.其次,大部分的意外情况是不出现在既定的需求中的,即,需求覆盖的测试更倾向软件验证(verification),希望测试能够满足需求说明书的指标.需求覆盖无法指导软件确认(validation),即无法满足其他在真实环境中遇到的情况,例如需求“自动驾驶汽车需保证可以在积水路段行驶”,如果未能提前定义该需求,需求覆盖就难以指导对积水路段相关错误行为的甄别.

为了更好的提出需求覆盖的准则,Fujikura 等人[72]提出了基于 KAOS[73]的需求覆盖方式,即对于自动驾驶汽车这类与安全相关的领域,根据安全要求,将危险状态指定为场景的最终状态,并利用这种方法生成更多揭露车辆问题的危险场景. KAOS 是需求工程领域使用的面向结果的需求捕捉方法,配合穷尽寻路方法使用,实验结果表明该方法与现有方法相比,能够实现较高的需求覆盖率,并且保证需求到测试方案的高度可追溯性.

根据 Schuldt 等人[23]的最新的研究结果,Scenario 和 Situation 得到了详细定义.Schuldt 等人认为:Situation 是交通参与者在特定时间点,用于选择合理行为模式的全部环境信息,通常出自某一交通参与者的主观视角.Scenario 是数个 scene 的时序发展组合,通常是一个确定的时间片段.

基于最新的定义,现有的研究致力于将上述抽象的覆盖标准,具体应用到实际的基于自动驾驶整车的测试当中.

Alexander 等人[74]最早提出情境覆盖的定义,并认为现有的“需求覆盖”是不足以全面地指导测试自动驾驶系统的.因为现有的需求覆盖严重依赖于需求说明书,它能有效应用于基于需求说明书的验证软件,却不能应用于现实场景下的软件确认.这意味着需求覆盖很难发现真实场景中不常见的错误,从而造成测试中的疏漏.相比而言,情境覆盖会更加可靠,更加细致,考虑更多的实际情况.Alexander 等人提出了宏观和微观两种常见的覆盖选择方式,并由 Hawkins 等人[75]在用例研究中给出了详细的例证.在用例研究中,Hawkins 等人建立了虚拟地图,选定自动驾驶汽车的初始位置和目的地,并在地图上放置障碍物和其他交通参与者,检测在车辆前进路线中是否会出现危险交通情况.该团队选取了三个距离标准作为情境覆盖的指标,实验证明这种情境覆盖标准可以比随机方法指导发现更多危险场景,但是效果有限.因为所选取的覆盖标准可能和危险相关性不强,因此覆盖标准的选取可能会对实验造成很大影响.

Situation 是由交通参与元素和相关参数组成的,在实际测试中,若覆盖所有可能的组合结果则会出现指数爆炸的问题.Cheng 等人[76]提出了量化的 k 投影覆盖率,要求将数据集投影到任意选择的 k 条件形成的超平面时,以不小于关联权重的数据点数覆盖每个类别,该指标大大减少了所需测试的场景数量.相较在每个类中至少覆盖一个数据点的朴素铺盖度量指标,该覆盖率指标解决了不同情况相对重要性的问题.此外 Cheng 等开发了一种高效编码到 0-1 整数的方法,该方法允许增量地创建场景以最大程度地增加覆盖范围.但该指标对于较大的 k 值,要实现完整的投影覆盖范围仍然面临挑战.

根据 Schuldt 等人[23]的定义,场景是由几个 scene 组成的长时间片段,通常具有一定时效性和动态性.因此,对场景的覆盖难以量化.现有的覆盖框架仍局限于以枚举的方式实现覆盖最大化.相关研究[77][78]中的枚举方式虽然简单,但是需要测试人员具有一定的经验,且容易造成漏选和遗忘.

目前比较有效的量化场景覆盖的方法是将场景抽象成图进行覆盖. Iqbal 等人[79]提出了动态测试世界的方法,他们使用 UML 类图构建参与者与其关系的结构模型,将交通参与者的行为抽象扩展为有限状态机(CEFSM),然后通过使用图覆盖标准以覆盖动态世界模型来生成抽象世界行为测试用例(AWBTC).实验结果表明可执行测试用例的数量取决于生成的测试数据数量及执行中涉及的参与者过程数量.该方法将传统软件测试图覆盖中的边覆盖、节点覆盖和路径覆盖方法引入自动驾驶测试.

在自动驾驶测试过程中,通过在模拟环境和封闭测试设施中设计适当的方案,可以减少进行道路测试的工作量.利用模拟或封闭设施测试主要优势是为每个运行设计域生成测试方案库.Feng 等人[80][81]为具有不同运行设计域(ODD),联网自动化车辆模型和性能指标的测试方案库生成的问题提出了一个公用框架,在给定运行设计域的情况下,测试场景库被定义为可用于联网自动化车辆测试的关键方案集.该团队利用机动性挑

战和暴露频率组合的辅助目标函数等指标,结合发生频率和机动性,评估场景的重要性和危险性,以便采样到更重要,更有测试意义的场景,尽可能覆盖到各种危险场景,将现实世界中发生概率较高的场景且机动性挑战较高的场景优先用于联网自动化车辆的测试.

5 数据集与模拟器

为了更好地支持该领域的研究人员,本文总结了相关文献中的数据集和模拟器的使用情况,并标注了数据集的规模以及出处,结果在表 1 与表 2 中展示.其中名称列表示数据集的名称,数据集规模列标注了每种数据集的大小,引用文献列标注了使用了这些数据集的文章,出处列标注了该数据集的出处.

Table 1 Summary of datasets

表 1 数据集

名称	数据集规模	引用文献	出处
KITTI	180G 路况采集视频	[17][14][28][34]	[82]
GTSRB	51840 张图片	[6]	[9]
UCF-101	13320 视频,101 动作类别	[11]	[83]
HMDB-51	6766 视频,51 动作类别	[11]	[84]
BIRD	60k	[11]	[11]
Sim200k(10k,50k)	200k(10k,50k)	[13]	[14]
Cityscapes	5000 高质量图片和 20000 粗略注解图片	[14]	[85]
Udacity	与使用版本有关	[28][29][30]	[86]
Dave testing dataset	45568 张图片	[28]	[87]

Table 2 Summary of simulators

表 2 模拟器整理

名称	引用文献	出处
Intelleigent Driver Model	[26]	[88]
AUTOACE	[23]	[23]
Udacity	[32]	[89]
Pro-SiVICTM	[49]	[90]
Prescan	[46]	[91]

6 总结与展望

本文总结了几年来针对自动驾驶智能系统测试的相关研究 56 篇,相关数据集 8 个以及相关模拟器 5 个.本文依据自动驾驶智能系统测试的对象将整篇文章划分成了 4 部分:自动驾驶感知模块的测试、自动驾驶决策模块的测试、自动驾驶综合功能模块的测试以及自动驾驶整车测试,并针对这四部分展开了详细的综述.

尽管目前已经有许多团队,学者参与到了该领域的研究当中,但该领域仍有许多难题等待着人们去攻克,本文总结了目前该领域仍然存在的挑战,为相关研究人员提供接下来的研究方向,推动自动驾驶系统测试进一步发展.

1. 由于自动驾驶智能系统的复杂性和难解释性,已有的自动驾驶智能系统测试方法也面临着可解释性较差的问题.而较差的可解释性会对进一步提升自动驾驶智能系统的鲁棒性,安全性等带来新的挑战.因此,提升自动驾驶智能系统测试的可解释性是一项重要的研究工作,也将为后续的自动修复带来新的机会.

2. 由于自动驾驶智能系统往往需要大量的测试样本,而标注测试样本是一项十分耗时耗力的任务,因此,如何选择有效的测试样本进行优先标注,对于提升自动驾驶智能系统的测试效率具有重要意义.因此,解决自动驾驶智能系统测试中的样本标注问题是一个重要的研究方向.

3. 尽管目前有许多针对整车的测试覆盖度量指标,但是针对单个模块(如感知模块和决策模块)的测试覆盖度量指标仍然欠缺.而单一模块的测试是整车测试的基础,因此,结合单一模块的具体特性,设计针对单一模块的测试覆盖度量指标,对自动驾驶智能系统的测试充分性具有重要意义,该方向也是未来研究方向之一.

References:

[1] Koopman P, Wagner M. Challenges in autonomous vehicle testing and validation. SAE International Journal of Transportation Safety, 2016, 4(1): 15-24.

[2] Li L, Lin Y L, Zheng N N, et al. Artificial intelligence test: a case study of intelligent vehicles. Artificial Intelligence Review, 2018, 50(3): 441-465.

[3] Garcia J, Feng Y, Shen J, et al. A comprehensive study of autonomous vehicle bugs. Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. 2020: 385-396.

[4] Kang Y, Yin H, Berger C. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. IEEE Transactions on Intelligent Vehicles, 2019, 4(2): 171-185.

[5] Eykholt K, Evtimov I, Fernandes E, et al. Note on attacking object detectors with adversarial stickers. arXiv preprint arXiv:1712.08062, 2017.

[6] Eykholt K, Evtimov I, Fernandes E, et al. Robust physical-world attacks on deep learning visual classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1625-1634.

[7] Mogelmose A, Trivedi M M, Moeslund T B. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4): 1484-1497.

[8] Stallkamp J, Schlipsing M, Salmen J, et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 2012, 32: 323-332.

[9] Stallkamp J, Schlipsing M, Salmen J, et al. The German traffic sign recognition benchmark: a multi-class classification competition. The 2011 international joint conference on neural networks. IEEE, 2011: 1453-1460.

[10] Lu J, Sibai H, Fabry E, et al. No need to worry about adversarial examples in object detection in autonomous vehicles. arXiv preprint arXiv:1707.03501, 2017.

[11] Daftry S, Zeng S, Bagnell J A, et al. Introspective perception: Learning to predict failures in vision systems. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 1743-1750.

[12] Gurău C, Rao D, Tong C H, et al. Learn from experience: probabilistic prediction of perception performance to avoid failure. The International Journal of Robotics Research, 2018, 37(9): 981-995.

[13] Ramanagopal M S, Anderson C, Vasudevan R, et al. Failing to learn: autonomously identifying perception failures for self-driving cars. IEEE Robotics and Automation Letters, 2018, 3(4): 3860-3867.

[14] Johnson-Roberson M, Barto C, Mehta R, et al. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?. arXiv preprint arXiv:1610.01983, 2016.

[15] Dreossi T, Ghosh S, Sangiovanni-Vincentelli A, et al. Systematic testing of convolutional neural networks for autonomous driving. arXiv preprint arXiv:1708.03309, 2017.

[16] Papernot N, McDaniel P, Jha S, et al. The limitations of deep learning in adversarial settings. 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016: 372-387.

[17] Talwar D, Guruswamy S, Ravipati N, et al. Evaluating Validity of Synthetic Data in Perception Tasks for Autonomous Vehicles. 2020 IEEE International Conference On Artificial Intelligence Testing (AITest). IEEE, 2020: 73-80.

[18] "LGSVL Simulator," https://github.com/lgsvl/simulator-2019.05-obsolete.

[19] Araiza-Illan D, Pipe A G, Eder K. Model-based test generation for robotic software: Automata versus belief-desire-intention agents. arXiv preprint arXiv:1609.08439, 2016.

[20] Bordini R H, Hübner J F, Wooldridge M. Programming multi-agent systems in AgentSpeak using Jason[M]. John Wiley & Sons, 2007.

[21] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213-3223.

[23] Schultz A C, Grefenstette J J, De Jong K A. Adaptive testing of controllers for autonomous vehicles. Proceedings of the 1992 Symposium on autonomous underwater vehicle technology. IEEE, 1992: 158-164.

[24] Tuncali C E, Pavlic T P, Fainekos G. Utilizing S-TaLiRo as an automatic test generation framework for autonomous vehicles. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016: 1470-1475.

[25] Mullins G E, Stankiewicz P G, Gupta S K. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017: 1443-1450.

[26] Koren M, Alsaif S, Lee R, et al. Adaptive stress testing for autonomous vehicles. 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1-7.

[27] K. Crombecq, L. De Tommasi, D. Gorissen, and T. Dhaene, "A novel sequential design strategy for global surrogate modeling," in Winter Simulation Conference. Winter Simulation Conference, 2009, pp. 731–742.

[28] Zhou H, Li W, Zhu Y, et al. Deepbillboard: Systematic physical-world testing of autonomous driving systems. arXiv preprint arXiv:1812.10812, 2018.

[29] Tian Y, Pei K, Jana S, et al. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. Proceedings of the 40th international conference on software engineering. 2018: 303-314.

[30] Zhang M, Zhang Y, Zhang L, et al. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2018: 132-142.

[31] Treiber M, Hennecke A, Helbing D. Congested traffic states in empirical observations and microscopic simulations. Physical review E, 2000, 62(2): 1805.

[32] Stocco A, Weiss M, Calzana M, et al. Misbehaviour Prediction for Autonomous Driving Systems. arXiv preprint arXiv:1910.04443, 2019.

[33] D'Ambrosio J, Adiththan A, Ordoukhanian E, et al. An MBSE Approach for Development of Resilient Automated Automotive Systems. Systems, 2019, 7(1): 1.

[34] Alhaija H A, Mustikovela S K, Mescheder L, et al. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision, 2018, 126(9): 961-972.

[36] Pei K, Cao Y, Yang J, et al. Deepxplore: Automated whitebox testing of deep learning systems. proceedings of the 26th Symposium on Operating Systems Principles. 2017: 1-18.

[37] Laurent T, Arcaini P, Ishikawa F, et al. A mutation-based approach for assessing weight coverage of a path planner. 2019 26th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2019: 94-101.

[38] Bühler O, Wegener J. Automatic testing of an autonomous parking system using evolutionary computation[R]. SAE Technical Paper, 2004.

[39] Bühler O, Wegener J. Evolutionary functional testing. Computers & Operations Research, 2008, 35(10): 3144-3160.

[40] Abdessalem R B, Nejati S, Briand L C, et al. Testing vision-based control systems using learnable evolutionary algorithms. 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE). IEEE, 2018: 1016-1026.

[41] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 2002, 6(2): 182-197.

[42] Beyer H G, Deb K. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Transactions on evolutionary computation, 2001, 5(3): 250-270.

[43] Deb K, Agrawal R B. Simulated binary crossover for continuous search space. Complex systems, 1995, 9(2): 115-148.

[44] Knowles J D, Thiele L, Zitzler E. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report, 2006, 214.

[45] Abdessalem R B, Nejati S, Briand L C, et al. Testing advanced driver assistance systems using multi-objective search and neural networks. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. 2016: 63-74.

[46] Abdessalem R B, Panichella A, Nejati S, et al. Testing autonomous cars for feature interaction failures using many-objective search. 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2018: 143-154.

[47] Panichella A, Kifetew F M, Tonella P. Reformulating branch coverage as a many-objective optimization problem. 2015 IEEE 8th international conference on software testing, verification and validation (ICST). IEEE, 2015: 1-10.

[48] Gietelink O, Ploeg J, De Schutter B, et al. Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations. Vehicle System Dynamics, 2006, 44(7): 569-590.

[49] Belbachir A, Smal J C, Blosseville J M, et al. Simulation-driven validation of advanced driving-assistance systems. Procedia-Social and Behavioral Sciences, 2012, 48: 1205-1214.

[50] Gruyer D, Glaser S, Pechberti S, et al. Distributed simulation architecture for the design of cooperative ADAS. First International Symposium on Future Active Safety Technology toward zero-traffic-accident. 2011.

[51] Hiblot N, Gruyer D, Barreiro J S, et al. Pro-sivic and roads. a software suite for sensors simulation and virtual prototyping of adas. Proceedings of DSC. 2010: 277-288.

[52] Abdessalem Ben R, Panichella A, Nejati S, et al. Automated Repair of Feature Interaction Failures in Automated Driving Systems. Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2020). 2020.

[53] Jones J A, Harrold M J, Stasko J. Visualization of test information to assist fault localization. Proceedings of the 24th International Conference on Software Engineering. ICSE 2002. IEEE, 2002: 467-477.

[54] Gambi A, Mueller M, Fraser G. Automatically testing self-driving cars with search-based procedural content generation. Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis. 2019: 318-328.

[56] Ulbrich S, Menzel T, Reschka A, et al. Defining and substantiating the terms scene, situation, and scenario for automated driving. 2015 IEEE 18th International Conference on Intelligent Transportation Systems. IEEE, 2015: 982-988.

[57] Zhang C, Liu Y, Zhao D, et al. RoadView: A traffic scene simulator for autonomous vehicle simulation testing. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014: 1160-1165.

[58] Althoff M, Lutz S. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1326-1333.

[59] Althoff M, Koschi M, Manzinger S. CommonRoad: Composable benchmarks for motion planning on roads. 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017: 719-726.

[60] Gambi A, Huynh T, Fraser G. Generating effective test cases for self-driving cars from police reports. Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019: 257-267.

[61] Fremont D J, Kim E, Pant Y V, et al. Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World. arXiv preprint arXiv:2003.07739, 2020.

[62] Tuncali C E, Fainekos G, Ito H, et al. Simulation-based adversarial test generation for autonomous vehicles with machine learning components. 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1555-1562.

[63] Michel O. Cyberbotics Ltd. Webots™: professional mobile robot simulation. International Journal of Advanced Robotic Systems, 2004, 1(1): 5.

[64] Wu B, Iandola F, Jin P H, et al. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017: 129-137.

[65] Kuhn D R, Kacker R N, Lei Y. Introduction to combinatorial testing[M]. CRC press, 2013.

[66] Rocklage E. Teaching self-driving cars to dream: A deeply integrated, innovative approach for solving the autonomous vehicle validation problem. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 1-7.

[67] Aramrattana M, Larsson T, Jansson J, et al. A simulation framework for cooperative intelligent transport systems testing and evaluation. Transportation research part F: traffic psychology and behaviour, 2019, 61: 268-280.

[68] Dreossi T, Fremont D J, Ghosh S, et al. Verifai: A toolkit for the formal design and analysis of artificial intelligence-based systems. International Conference on Computer Aided Verification. Springer, Cham, 2019: 432-442.

[69] Rocklage E, Kraft H, Karatas A, et al. Automated scenario generation for regression testing of autonomous vehicles. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017: 476-483.

[70] Chance G, Ghobrial A, Lemaignan S, et al. An Agency-Directed Approach to Test Generation for Simulation-based Autonomous Vehicle Verification. arXiv preprint arXiv:1912.05434, 2019.

[71] Tahir Z, Alexander R. Coverage based testing for V&V and Safety Assurance of Self-driving Autonomous Vehicle: A Systematic Literature Review. The Second IEEE International Conference On Artificial Intelligence Testing. York, 2020.

[72] Fujikura T, Kurachi R. A test scenario generation method for high requirement coverage by using kaos method. 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2019: 542-543.

[73] Van Lamsweerde A. Requirements engineering: From system goals to UML models to software[M]. Chichester, UK: John Wiley & Sons, 2009.

[74] Alexander R, Hawkins H R, Rae A J. Situation coverage–a coverage criterion for testing autonomous robots. 2015.

[75] Hawkins H, Alexander R. Situation Coverage Testing for a Simulated Autonomous Car--an Initial Case Study. arXiv preprint arXiv:1911.06501, 2019.

[76] Cheng C H, Huang C H, Yasuoka H. Quantitative projection coverage for testing ML-enabled autonomous systems. International Symposium on Automated Technology for Verification and Analysis. Springer, Cham, 2018: 126-142.

[77] Alnaser A J, Akbas M I, Sargolzaei A, et al. Autonomous vehicles scenario testing framework and model of computation. SAE International Journal of Connected and Automated Vehicles, 2019, 2(12-02-04-0015): 205-218.

[78] Antkiewicz M, Kahn M, Ala M, et al. Modes of Automated Driving System Scenario Testing: Experience Report and Recommendations[R]. SAE Technical Paper, 2020.

[79] Iqbal M Z, Arcuri A, Briand L. Empirical investigation of search algorithms for environment model-based testing of real-time embedded software. Proceedings of the 2012 International Symposium on Software Testing and Analysis. 2012: 199-209.

[80] Feng S, Feng Y, Yu C, et al. Testing scenario library generation for connected and automated vehicles, part i: Methodology. IEEE Transactions on Intelligent Transportation Systems, 2020.

[81] Feng S, Feng Y, Sun H, et al. Testing scenario library generation for connected and automated vehicles, part II: Case studies. IEEE Transactions on Intelligent Transportation Systems, 2020.

[82] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.

[83] Soomro K, Zamir A R, Shah M. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.

[84] Kuehne H, Jhuang H, Garrote E, et al. HMDB: a large video database for human motion recognition. 2011 International Conference on Computer Vision. IEEE, 2011: 2556-2563.

[85] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3213-3223.

[86] Udacity, https://github.com/udacity/self- driving- car/tree/master/datasets.

[87] Chen S. Autopilot-tensorflow, 2016. URL https://github. com/SullyChen/Autopilot-TensorFlow, 2016, 1(5): 6.

[88] Treiber M, Hennecke A, Helbing D. Congested traffic states in empirical observations and microscopic simulations. Physical review E, 2000, 62(2): 1805.

[89] Udacity. 2017. A self driving car simulator built with Unity. https://github.com/udacity/self-driving-car-sim. Online.

[90] Bellet T, Mayenobe P, Bornard J C, et al. COSMO-SIVIC: a first step towards a virtual platform for Human Centred Design of driving assistances. IFAC Proceedings Volumes, 2010, 43(13): 210-215.

[91] TASS-International. 2018. PreScan. https://www.tassinternational.com/prescan. (2018).

附中文参考文献:

[22] 中国汽车技术研究中心有限公司.智能网联汽车技术,第二版,北京: 社会科学文献出版社,2020.

[35] 王赞,闫明,刘爽,陈俊洁,张栋迪,吴卓,陈翔.深度神经网络测试研究综述.软件学报,2020,31(05):1255-1275.

[55] 中国汽车技术研究中心有限公司.自动驾驶测试场景技术发展与应用,第一版,北京: 机械工业出版社,2020.