硕士学位论文

基于计算机视觉和深度学习的自动驾驶方法研究

RESEARCH ON AUTONOMOUS DRIVING METHOD BASED ON COMPUTER VISION AND DEEP LEARNING

白辰甲

哈尔滨工业大学

2017 年 6 月

国内图书分类号：TP391　　学校代码：10213
国际图书分类号：004.9　　密级：公开

工学硕士学位论文

基于计算机视觉和深度学习的自动驾驶方法研究

硕士研究生：白辰甲

导　　师：唐降龙 教授

申请学位：工学硕士

学　　科：计算机科学与技术

所在单位：计算机科学与技术学院

答辩日期：2017 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP391
U.D.C: 004.9

Dissertation for the Master's Degree in Engineering

RESEARCH ON AUTONOMOUS DRIVING METHOD BASED ON COMPUTER VISION AND DEEP LEARNING

Candidate: Bai Chenjia
Supervisor: Prof. Tang Xianglong
Academic Degree Applied for: Master of Engineering
Specialty: Computer Science and Technology
Affiliation: School of Computer Science and Technology
Date of Defence: June, 2017
Degree-Conferring-Institution: Harbin Institute of Technology

摘　要

自动驾驶是指车辆通过传感器感知周围环境，在没有人为干预的情况下，实时改变驾驶行为，完成驾驶任务。自动驾驶可以减少交通事故的发生，提高道路交通资源的使用率，节约居民的出行成本，因此对自动驾驶技术的研究具有重要意义。

基于计算机视觉的自动驾驶技术使用视觉传感器的观测图像作为输入，驾驶动作作为输出。现有方法主要分为间接感知型（Mediated Perception）方法、直接感知型（Direct Perception）方法和端到端控制（End-to-End Control）方法。其中，间接感知型方法将自动驾驶任务分为目标检测、目标跟踪、场景语义分割、相机模型和标定、三维重建等子任务。直接感知型方法首先学习交通环境的关键指标，随后由控制逻辑进行控制。端到端控制方法直接建立输入到动作的映射，具有简明的系统结构。

本文设计了一个基于端到端控制和深度学习的自动驾驶算法。算法将自动驾驶作为一个整体的问题进行研究，建立一个端到端的学习系统。学习系统是由 7 层卷积层和 4 层全连接层组成的卷积神经网络（CNN），网络的输入是无人车第一视角的图像，输出是一个浮点数，代表要预测的转向角。相对于预测左转、右转等动作的传统方法，连续的转向角对运动的描述更加精确。为了提升训练效果，算法在训练中使用了网络预训练和防止过拟合等措施。

与间接感知型结构和直接感知型结构相比，本文设计的算法具有明显的优势。首先，避免了间接映射型方法的复杂系统结构，降低了设计的难度。其次，可以在真实场景下高效的完成数据采集和训练，而直接感知型方法中需要学习驾驶相关的指标，例如与障碍物的距离、与标志线距离等，在真实场景中精确采集这些数据需要超声和激光雷达等设备，采集成本高，不易实现。本文算法只需要记录视场图像和转向角作为卷积神经网络的训练样本，在测试时只需采集视场图像，根据卷积神经网络预测的转向角实现对智能车的连续控制。

为了减少真车实验的成本，本文设计实现了一个微缩智能车系统用于数据采集和算法验证。智能车在自行设计的有标志线和障碍物的模拟交通环境中进行数据采集、训练和测试。在对 CNN 的可视化中，发现 CNN 可以自行提取与决策有关的特征。实验结果表明，智能车能够提前规划合理的路线进行避障，能够在常规场景中保持较高的自动驾驶率。

关键词：自动驾驶；计算机视觉；深度学习；卷积神经网络；预训练

Abstract

Autonomous driving means the vehicle change the driving behavior in real time and complete the driving task by observing the surroundings without human intervention. Autonomous driving can reduce the number of traffic accidents, improve the road utilization rate of traffic resource and save travel cost. So the research of automatic driving technology is of great significance.

Autonomous driving technology based on computer vision uses visual sensors as input and drive actions as output. There are three main typical methods as follows: Mediated Perception, Direct Perception and End-to-End Control. Among them, Mediated Perception method separates driving task into some sub tasks, including object detection, tracking, semantic segmentation, camera model and calibration, 3D reconstruction and so on. Direct Perception method learns the key indicators of traffic situation and then be controlled by the control logic. End-to-End Control method establishes the input to the action mapping directly with a concise system structure.

We design an automated driving algorithm based on End-to-End Control and deep learning method, which takes the problem of autonomous driving as a whole to study and establishes an end-to-end learning system. The learning system is a convolution neural network (CNN) consisting of seven convolution layers and four fully connected layers. The input of the network is the image from the first view of vehicle, and the output is a floating point number which represents the steering angle to be predicted. Compared to traditional methods which can only predicts movements like left/right turning etc., the continuous steering angle is a more accurate description of motion. In addition, in order to improve the training effect, we use the network pre-training method and the overfitting prevention measures.

Compared to the Mediated Perception method and the Direct Perception method, the proposed algorithm in this paper has obvious advantages. Firstly, the algorithm avoids the complex system structure of the indirect mapping system, because the Mediated Perception structure needs to be divided into several sub-tasks, and its design and implementation is very difficult. Secondly, the algorithm can accomplish data collection and network training in real situation efficiently. However, the Direct Perception method needs to learn the key indicators of traffic situation like the distance from the barrier, the distance from

the marker line and so on. But the precise collection of these data in real situations often requires equipment such as ultrasound and laser radar, which are costly and difficult to implement. The proposed algorithm only needs to record the current field image and the turning angle as training set. In testing period, only the images from the first view of vehicle are collected, the continuous control of vehicle is implemented according to the steering angle predicted by the convolutional neural network.

In order to reduce the cost of experiment with a real vehicle, a mini intelligent vehicle system was designed to be used for data collection and algorithm verification. The intelligent vehicle performs data acquisition, network training and testing in a self-designed environment with marked lines and obstacles. We also find that the convolution neural network can extract useful features for driving by itself. The result shows that the intelligent car was be able to planning a reasonable route to avoid obstacle in advance and maintain high auto-pilot rates in conventional scenarios.

Keywords: autonomous driving, computer vision, deep learning, convolution neural network, pre-training

目 录

第 1 章 绪论

1.1 课题背景

本课题在国家自然科学基金“主动视觉中的对抗问题研究”（批准号：61672190）的支持下进行，研究基于计算机视觉和深度学习的自动驾驶技术。

近年来，随着人工智能领域的不断发展，自动驾驶技术得到了广泛关注。自动驾驶是指车辆通过观察和感知周围环境，在没有人为干预的情况下，实时的改变驾驶行为，完成驾驶任务。自动驾驶系统一般包括环境感知和决策等两个部分，其中环境感知中使用的传感器包括相机、雷达、超声、定位设备等。本课题研究基于计算机视觉和深度学习的自动驾驶技术，即通过相机获取环境信息，使用计算机视觉和深度学习的方法进行决策。相比于其他传感器而言，视觉信息具有易于采集、信息全面、设备廉价的优势。目前，基于计算机视觉的自动驾驶技术已成为该领域的主流方法[1]，其他传感器往往作为辅助。

随着我国经济社会的快速发展，我国的机动车产量和保有量已经连年高居世界第一。根据公安部发布的汽车保有量相关报告，显示截至 2016 年底，全国机动车保有量达到 2.9 亿辆，其中汽车 1.94 亿辆。机动车驾驶人 3.6 亿人，其中汽车驾驶人超过 3.1 亿。私家车总量达 1.46 亿辆，平均每百户家庭拥有私家车 36 辆，与 2015 年相比，私家车增加 2208 万辆，增长 15.8%。连年的机动车数量增长，给道路交通环境带来了前所未有的压力，同时造成了交通事故频发。自动驾驶技术的出现给解决上述问题带来了希望，具体分析如下。

第一，自动驾驶技术可以有效减少交通事故的发生。交通事故不仅给人民的生命安全造成巨大威胁，同时也造成了巨大的经济损失。道路交通事故发生的原因包括道路环境、天气情况、车辆状况以及驾驶员操作等。其中，由于驾驶员操作不当而引发的道路交通事故占到了总事故的 70% 以上。相比于驾驶员来说，自动驾驶技术的驾驶行为具有可预测性，是无人车辆根据对周围环境的分析得出的。而人类驾驶员可能受到主观驾驶习惯、驾驶经验和情绪的影响。因此，使用自动驾驶技术替代驾驶员是可以显著减少交通事故的一种可行的解决方案。

第二，自动驾驶技术可以提高道路交通资源的使用率，缓解交通压力。自动驾驶车辆可以由城市管理者进行统一调度和分配，路况信息可以随时同步到所有的车辆，车辆可以进行自动分流，从而缓解交通压力。而人类驾驶员往往有自己偏好的交通路线，也不易接收到最新的路况信息，因此容易造成拥堵。

第三，自动驾驶技术可以节约居民的出行成本。统计数据显示，城市有 24% 的路面用于停车，平均每辆车有 95% 的时间处于停车状态，只有 5% 的时间处于行驶状态。因此，如果部分无人驾驶车辆能够实现共享利用和按需分配调度，那么可以有效节约居民的出行成本，提高车辆的利用效率。

自动驾驶技术在国外的研究和应用起步较早，以美国、欧洲、日本为代表的国家在上世纪 90 年代开始相关领域的研究。美国在 1995 年开始了第一个自动驾驶项目[2]，该项目由卡内基梅隆大学主导，美国于同年成立了国家自动公路系统联盟（NAHSC）来支持自动驾驶项目的开展。欧洲、日本等国纷纷效仿，在本国推进相关领域的研究。以谷歌公司为代表的科技公司也纷纷投入自动驾驶领域[3]，谷歌公司于 2009 年开始启动自动驾驶项目，截至 2016 年 4 月，谷歌无人车已经累计安全行驶了 1,498,000 英里，谷歌的自动驾驶汽车如图 1-1 a) 所示。以特斯拉公司为代表的新型汽车厂商也不甘落后，特斯拉于 2015 年推出了新型辅助驾驶系统，并于 2016 年 10 月推出了新型自动驾驶汽车，该车辆装备有 8 个摄像头、12 个超声传感器以及激光雷达，具备完全的自主驾驶能力，特斯拉自动驾驶系统如图 1-1 b) 所示。

a) Google 无人车　　b) Tesla 无人驾驶系统

图 1-1 国外科技公司无人驾驶汽车举例

国内在自动驾驶领域的研究和应用起步较晚，但发展极为迅速。以国防科技大学和清华大学为代表的高校是国内最早开展相关领域研究的机构。2003 年，国防科技大学开发了第一款基于视觉系统的自主驾驶地面车辆 CITAVT-IV[4]，该车辆可以在结构化道路环境下行驶。同年，国防科技大学和一汽集团联合开发的基于红旗轿车的自主驾驶系统，该轿车在高速公路上以最高时速 170km 的速度行驶，如图 1-2 a) 所示。清华大学于 2002 年开发了一款适用于高速公路的，可以进行实时视觉导航的无人车。近年来，国内在相关领域的研究逐步由学术界蔓延到工业界，包括传统汽车制造企业和新型互联网企业在内的众多研究机构纷纷在该领域投入力量。百度公司于 2013 年开始进行无人车项目，以宝马 3 系汽车作为测试平台。百度无人车将车载计算和云计算相结合，建立了“百度汽车大脑”。“百度汽车大

脑”具有国内外领先的十余项核心技术，包括智能互联、人机交互、精确定位、标志检测、场景分割、目标跟踪、目标识别、距离估计等。百度无人车于 2016 年获得了美国加利福尼亚州政府颁发的无人车牌照，该牌照系全球第 15 张。百度无人车如图 1-2 b) 所示。此外，一汽集团联合国防科技大学开展自动驾驶研究，其开发的自动驾驶汽车已经经过多次测试；上汽集团与中航科工进行合作，于 2015 年的上海车展展示了自主研发的智能驾驶汽车 iGS，可以初步实现远程遥控泊车、自动巡航、自动跟车、车道保持、换道行驶、自主超车等功能；东风汽车公司与华为公司于 2014 年 10 月签署合作协议，将逐步开发实现具有情感化自动驾驶的智慧汽车。

a) 我国首辆无人轿车——红旗　　b) 百度无人驾驶汽车

图 1-2 国内无人驾驶汽车举例

在高校科研中，使用真实乘用车进行无人驾驶技术的开发和测试成本太高，使用微缩智能车是一个合理的选择。微缩智能车是指按照一定比例缩小后的自动驾驶技术的载体，其优势在于可以大大减轻物理平台的控制程序的设计工作，从而可以将研究重点放在自动驾驶算法的设计和实现上。此外，微缩智能车平台上的相关研究成果可以轻松的转化到真实乘用车上，因此在高校等科研机构中非常流行。本课题中将使用自行设计实现的微缩智能车进行相关算法的实验和测试。

1.2 国内外研究现状分析

从上世纪 80 年代开始，国内外在自动驾驶和微缩智能车领域开展了一系列研究项目，推动了这一领域的发展。本节将首先介绍国内外具有代表性的自动驾驶项目和微缩智能车项目，随后介绍基于计算机视觉的自动驾驶技术的研究现状。现有的自动驾驶技术主要分为间接感知型（Mediated Perception）方法、直接感知型（Direct Perception）方法和端到端控制（End-to-End Control）方法。其中间接感知型方法包含计算机视觉中的多个子任务，包括目标检测、目标跟踪、场景语义分

割、相机模型和标定、三维重建等，直接感知型方法和端到端控制方法主要使用深度学习技术。本章将对这些技术做简要介绍并讨论现有技术存在的挑战。

1.2.1 国内外自动驾驶项目的发展历史

从上世纪 80 年代起，许多国家开展了研发智能驾驶系统（ITS）的项目。欧美国家从 1986 年开始进行了称为 PROMETHEUS 的自动驾驶项目，该项目联合了超过 13 个汽车制造商，19 所大学以及其他组织。美国研制的自动驾驶车辆完成了从宾夕法尼亚州到加利福尼亚州圣地亚哥市的高速公路的驾驶，取得了在当时令人瞩目的成果。日本随后组织国内多个研究机构于 1996 年成立了“高级辅助驾驶公路系统研究协会”（ACHSRA），用于促进关于自动驾驶方面的研究。Bertozzi 等[5]于 2000 年调研了欧美及日本国家在自动驾驶领域取得的成果，同时指出了该领域面临的挑战。Bertozzi 认为，对于自动驾驶汽车来说，计算资源的已经不再是发展的瓶颈，瓶颈在于计算机视觉的相关算法，这些算法在处理路面反射、潮湿公路、隧道、阴影、遮挡等问题时表现不佳。他建议，一方面要继续深入计算机视觉算法的研究，另一方面要改善视觉传感器的设计，增强视觉传感器的功能。另外，与自动驾驶相关的法律和安全问题也需要进一步讨论。

PROMETHEUS 项目在高速公路的自动驾驶中取得了一定成果，受此启发，Franke 等[6] 于 1998 年进行了针对城市道路自动驾驶的相关研究。他们设计了一个实时视觉系统，适用于城市复杂的交通环境。该视觉系统基于立体视觉获取障碍物的深度信息，进行障碍物和信号灯的检测和跟踪。同时，该系统对城市道路进行了特征提取和结构化描述，如图 1-3 所示。2010 年，来自 VisLab[1]实验室的研究者设计了名为 BRAiVE 的原型车[7]，该原型车综合了 Vislab 实验室之前开发的所有视觉系统，具有障碍物检测、标志线检测、距离测量等功能。该原型车于 2011 年参加了从意大利到中国的自动驾驶挑战赛，取得成功。

Broggi 等[8] 在 2015 年开始了名为 PROUND 的项目，该项目在 BRAiVE 等[7]设计的原型车的基础上进行改造，使其适应在城市道路和高速公路的复杂交通环境。传感器方面，PROUND 无人车不仅使用基于视觉的传感器，同时使用了激光雷达、GPS 定位系统和 IMU 惯性导航设备。架构方面，PROUND 无人车采用分层架构，主要分为环境感知、建立环境地图、控制和规划等部分，如图 1-4 所示。无人车首先使用视觉和激光雷达等传感器进行障碍物检测、道路检测、标志检测、信号灯检测等，随后对多视角的检测结果进行融合，形成完整的环境地图。最后基于

[1]http://www.vislab.it

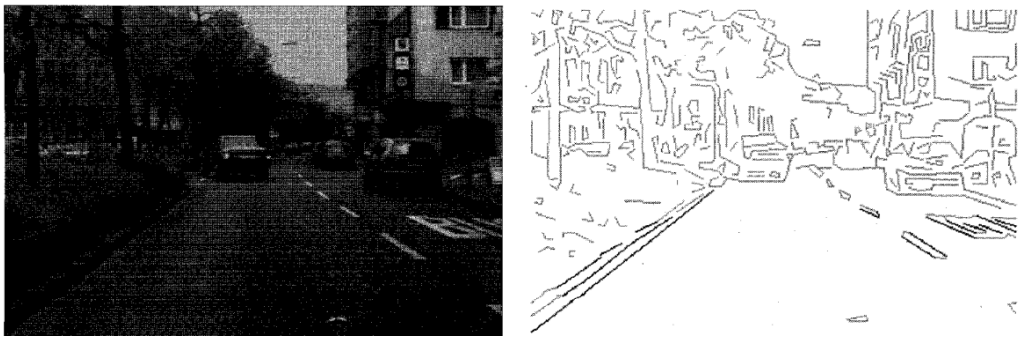


图 1-3 Franke 研究中使用的无人车和道路结构化特征

街景地图和实时路况进行全局路径规划和动力系统控制。PROUND 无人车先后在十余个比赛中进行了性能测试，不断改进、优化和组合多个模块。Broggi 指出，当前的自动驾驶技术已经可以初步应对复杂的真实场景，但为了安全性，无人车往往在速度上有所妥协，因此今后的研究应该更加注重如何在保证安全性的前提下提高无人车的驾驶速度。

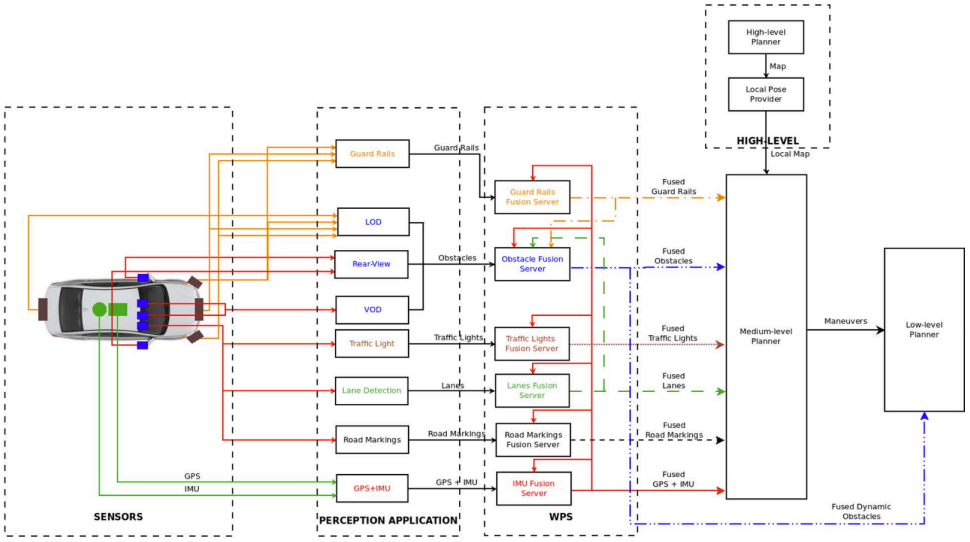
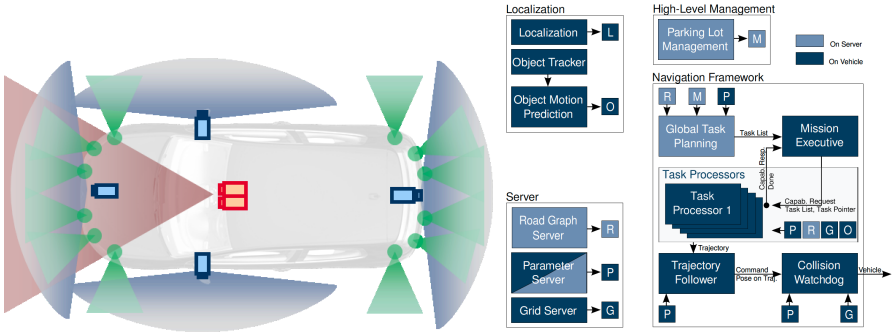


图 1-4 PROUND 无人车系统结构

欧盟委员会于 2013 年开始 V-Charge[9] 自动驾驶研究项目，旨在服务欧洲未来的交通和环境规划。V-Charge 无人车与之前介绍的无人车相比，有两个比较鲜明的特点。首先，V-Charge 无人车使用电力能源，并建设了一个高效的城市泊车和充电资源调配系统，使所有的 V-Charge 无人车可以在该系统的统一调配下以协作方式为城市居民提供服务；其次，V-Charge 项目之前的大部分自动驾驶项目使用的车辆配备了昂贵的传感器设备，而 V-Charge 无人车仅使用廉价的传感器和计算设备，使得无人车大规模使用成为可能。V-Charge 无人车配备的传感器设备有：12 个用于近距离障碍物检测的声呐传感器，一个立体相机，四个 360 度视角的鱼眼相

机，以及一个标准的 GPS 接收器等，廉价的设备使得 V-Charge 无人车的成本大幅下降。V-Charge 无人车中的传感器设备如图 1-5 a) 所示。架构方面，V-Charge 无人车分为定位和检测、地图服务、导航、控制四大模块，具体结构如图 1-5 b) 所示。该系统组合使用了计算机视觉领域和机器学习领域的多项技术，例如在定位中主要使用了同时定位和建图（SLAM）技术，障碍物检测主要使用双目视觉和 AdaBoost 分类器等。



a) V-Charge 传感器示意图　　b) V-Charge 系统架构示意图

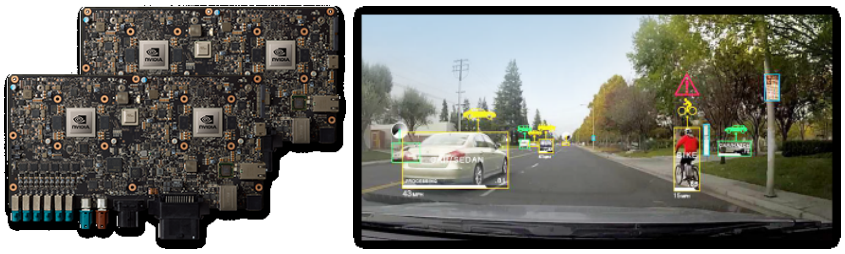
图 1-5 V-Charge 系统示意图

此外，以谷歌（google）、特斯拉（Tesla）、英伟达（Nvidia）为代表的科技公司也在自动驾驶领域进行了许多项目的开展。谷歌目前为自动驾驶项目成立了独立的子公司 Waymo[2]，已经在无人车商业化方面有了一些成果；特斯拉公司最近也推出了自己的自动驾驶汽车[3]，在商业化方面迈出了坚实的一步；英伟达公司最近发布了一款自动驾驶芯片 NVIDIA DRIVE^{TM}PX2，可以让汽车制造商和一级汽车制造供应商加速产品的自主化和无人驾驶车辆的研发如图 1-6 a) 所示。该芯片具有可扩展性，根据具体的设计需求可以进行扩展。该芯片提供了环境探测、自动巡航、高清地图绘制、定位车辆等功能，其中环境探测功能如图 1-6 b) 所示。该芯片使用深度学习技术，融合来自多个摄像头和激光雷达的数据，使无人车可以全方位的探测周围环境，具有高度的准确性和稳定性。NVIDIA DRIVE^{TM}PX2 芯片同时提供了一个基于深度学习的自动驾驶算法的部署教程[10]。

国内近年来具有代表性项目是包括由军事交通学院开发的军用无人车和由西安交通大学研制的“发现号”无人车，分别如图 1-7 a) 和 1-7 b) 所示。在陆军装备部主办的 2016“跨越险阻”地面无人车挑战赛中，军事交通学院研发的“猛狮智能 1 号”车队表现出色，取得了野外战场侦查和编队行进两个比赛的冠军。西安交

[2]http://www.waymo.com

[3]https://www.tesla.com/autopilot



a) NVIDIA DRIVE^{TM}PX2 芯片示意图　　b) NVIDIA DRIVE^{TM}PX2 环境探测功能示意图

图 1-6 NVIDIA DRIVE^{TM}PX2 示意图

通大学从 2005 年开始自动驾驶汽车的研发，其研发的“发现号”无人车在 2016 年“中国智能车未来挑战赛”取得了不错的成绩。

a) 军事交通学院无人车　　b) 西安交通大学无人车

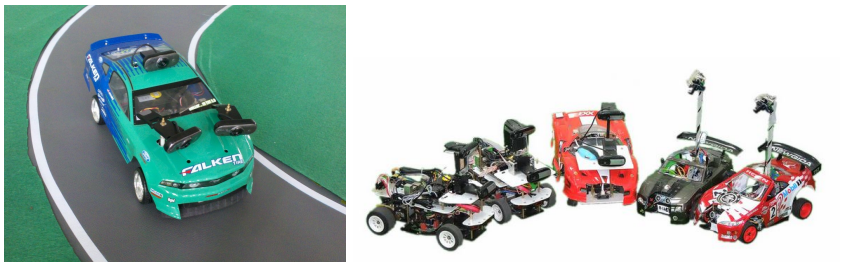
图 1-7 国内高校研制的无人车

1.2.2 国内外微缩智能车的发展历史

缩微智能车是真实车辆的缩小版本，可以方便的用于自动驾驶算法的测试中。在研究中，微缩智能车具有高度的灵活性、可控性、安全性、针对性等优点，不受限与城市道路空间，可以有效减轻研究过程中的平台设计难度。缩微智能车与真车相比在体积上虽然有所降低，但在保证基本的驾驶行为与交通特性条件下，其主要功能上可以与真车相媲美。微缩智能车可以在模拟交通环境中使用计算机视觉、自动控制、机器学习等技术进行自动驾驶，可以方便的进行功能测试和算法验证，相关成果可以轻松转移到真实乘用车的自动驾驶中。同时，微缩智能车的自动驾驶风险是可控的，可以方便的进行成果展示。

国内外许多无人驾驶的研究机构都拥有微缩智能车平台，用于快速进行的算法验证，减少真车实验中的开销。如中国科学院自动化研究所设计的微缩智能车如图 1-8 a) 所示，可以实现交通信号灯识别、超车行驶、跟车行驶、转弯避障等功

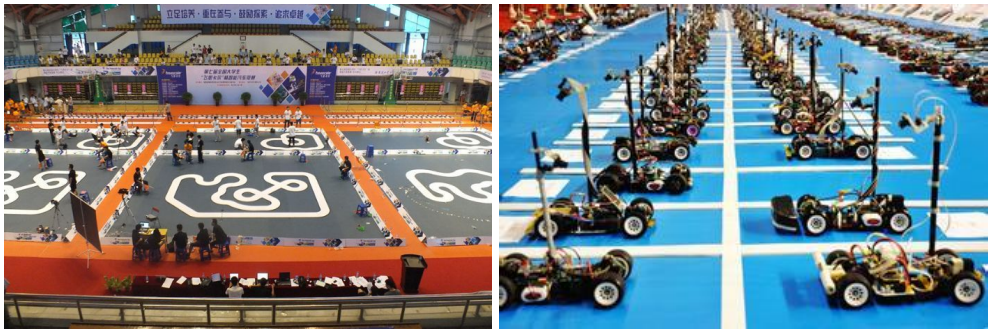
能，硬件结构主要分为传感器、计算系统，控制器和执行机构。类似，上海交通大学智能车实验室也以微缩智能车为实验平台进行算法设计和验证，如图 1-8 b) 所示。该实验室设计的智能车取得了 2012 年全国大学生“飞思卡尔杯”比赛的全国一等奖。



a) 中科院微缩智能车　　b) 上海交通大学微缩智能车

图 1-8 国内研究机构的微缩智能车

全国大学生“飞思卡尔杯”比赛是国内最具代表性的微缩智能车比赛，在比赛中参赛队伍需要自行设计智能车控制单元、算法单元、动力单元、传感器单元、转向单元等，在指定的赛道上完成比赛。该比赛受飞思卡尔公司赞助，由高校自动化委员会承办，每年都会吸引来自国内众多高校的大学生参加。2015 年的比赛赛道和参赛智能车分别如图 1-9 a) 和 1-9 b) 所示。



a) “飞思卡尔杯”比赛赛道　　b) “飞思卡尔杯”参赛智能车

图 1-9 “飞思卡尔杯”比赛情况

1.2.3 自动驾驶技术研究现状

现有的自动驾驶技术主要分为三种，分别是间接感知型（Mediated Perception）方法、直接感知型（Direct Perception）方法和端到端控制（End-to-End Control）方法。间接感知型方法将驾驶任务分为多项子任务，分别作为计算机视觉的标准任务进行计算，随后将计算结果进行转换和整合作为决策的输入。直接感知型方法需要人工设计与自动驾驶相关的关键指标，随后从图像中直接学习这些关键指标，作为

决策的输入。端到端控制方法也称表现反射型（Behavior Reflex）方法，该结构不进行任务拆分，直接从图像中学习转向角等决策信息。三种结构的区别如图 1-10 所示。

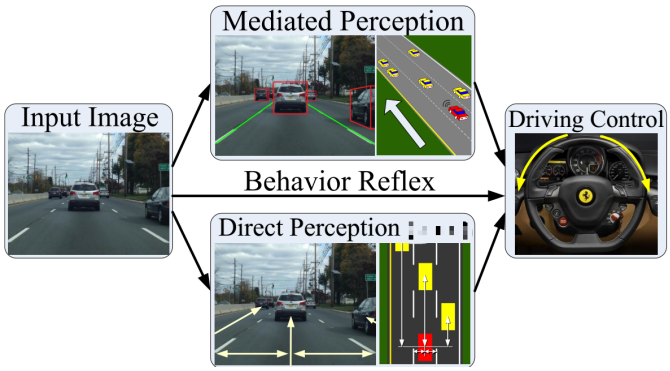


图 1-10 三种自动驾驶系统结构示意图

1.2.3.1 间接感知型方法

间接感知型[11]方法是传统的自动驾驶方法。该方法一般包括计算机视觉的多项子任务，包括目标检测、目标跟踪、场景语义分割、相机模型和标定、三维重建等。算法使用目标检测技术探测与驾驶相关的目标，如道路、交通标志、信号灯、车、行人等，使用场景语义分割技术对交通场景进行分割，使用标定和重建技术计算与障碍物距离等。系统将所有信息进行组合，形成无人车环境的完整表示，决策系统利用这种环境表示进行决策。系统结构如图 1-11 所示。在将多种识别结果合成道路交通环境的完整表示方面，有许多学者进行了研究。如 Zhang 等[12]提出了一种基于概率的生成式模型，该模型将多种检测结果作为输入，输出与交通状况相关的抽象表示。

间接感知型方法的优势在于模块化清晰，可以利用计算机视觉领域内多个子任务的研究成果[13]设计和优化每个模块，各模块之间耦合度低，容易进行组合和调整，在系统出现问题后容易排查故障。但是，这种方法也一定存在缺陷。首先，计算机视觉中的各项子任务均是针对标准数据集设计的，不一定完全适合自动驾驶系统所面临的道路交通环境。其次，系统冗余性和复杂度较高。系统输入由多个子模块组成，输入维度较高，但系统输出一般只有角度和速度等变量，输出维度很低。高维的输入最终转化低维输出，表明冗余性较高，造成了不必要的资源浪费，导致成本较高，不适宜大规模商用。

1.2.3.2 直接感知型方法

直接感知型方法是近年来提出的一种自动驾驶方法[14]，在间接感知型方法的基础上进行了优化，直接学习与驾驶相关的关键指标。这些关键指标需要根据驾

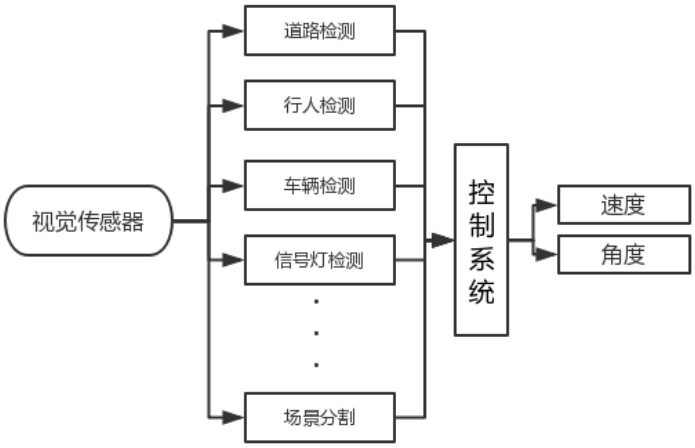


图 1-11 间接感知型自动驾驶系统结构示意图

驶环境进行设计，一般包括与前车的距离、与左侧标志线的距离、与右侧标志线的距离等。该方法可以大大简化系统的复杂度，例如，间接感知型方法包含目标检测模块，该模块的输出是附近车辆和行人等目标在图像中的位置。但实际上，无人车需要的不是目标在图像中的位置，需要的是目标与无人车自身的距离。如果使用目标检测算法，就需要在得到目标位置后通过相机模型转换为世界坐标中的距离。在直接感知型方法中，不再使用目标检测算法，而直接通过神经网络学习目标与自身的距离，简化了步骤。但是，该方法也存在问题。因为算法直接学习距离，因此需要使用带有目标与无人车距离的数据集进行训练。在真实交通场景中，采集带有精确距离信息的数据集往往需要使用超声和激光雷达等设备，采集成本高。因此，多数情况下，算法需要在仿真环境下获得大量样本进行训练，随后在真实交通环境下采集少量样本进行参数微调。另外，由于驾驶相关的关键指标需要人为设计，因此在非结构化的交通场景中存在一定的局限性。

1.2.3.3 端到端控制方法

端到端控制方法的目的是建立一个从传感器到驾驶动作的直接映射关系。这种结构起源于上世纪 80 年代，Pomerleau 等建立了一个基于神经网络的[15] 端到端系统，该系统是早期端到端结构的智能车的代表。系统的训练数据由人类驾驶员采集，受限于当时的理论发展和计算限制能力，ALVINN 系统使用的是浅层网络，仅用少量样本学习，只能适用于较为简单的场景。Lecun 等[16] 对该系统进行了改进，建立了 DAVE 智能车系统用于室外避障。DAVE 智能车是一个自行设计的微缩智能车，如图 1-12 a) 所示，该智能车由远程主机控制，智能车将实时采集的图像传送给远程主机，远程主机经过计算后，将控制命令发送给智能车进行控制。DAVE

在人工远程控制下采集图像，图像传感器使用双目相机，采集的样本需要包括不同的光照条件、障碍物类型、场景类型等，共计采集样本 12 万幅。一幅典型的训练图像如图 1-12 b) 所示。DAVE 中首次引入使用卷积神经网络进行训练，网络以双目相机图像作为输入，以左转、右转、直行三个转向动作作为输出，使用 CPU 训练，在室外场景的避障中取得了不错的效果。

a) DAVE 微缩智能车　　b) DAVE 训练数据的典型场景

图 1-12 DAVE 智能车示意图

综上，三种典型的自动驾驶方法（间接感知型、直接感知型、端到端）都有一些代表性成果，各自存在一定的优缺点，三种方法都在不断的研究和发展中。

1.2.4 自动驾驶技术面临的挑战

自动驾驶技术是一项交叉领域技术。在硬件方面它依赖于视觉传感器的发展水平，在算法方面它依赖计算机视觉和深度学习的研究水平。自动驾驶提供了一个平台来组合使用多种技术，共同完成目标。因此，自动驾驶技术对计算机视觉和深度学习的发展也有很大的促进作用。计算机视觉和深度学习近年来均取得了较大发展，但实现任意复杂环境下进行自动驾驶仍需要很长的时间。主要原因有三个：

第一，现有方法对未知场景的泛化能力问题。许多深度学习和计算机视觉算法基于监督学习和大规模样本训练，但自动驾驶汽车在实际行驶过程中遇到的场景往往不在训练样本库中。因此需要算法在未知场景有良好的泛化能力，能够应对未知的场景。这对于当前技术的发展水平而言仍是一个巨大的挑战。

第二，现有方法的风险和风险控制问题。自动驾驶系统出现错误的代价是昂贵的，涉及到乘客的生命财产安全。目前计算机视觉算法在各项数据集测试中仍存在一定的错误率[13]，实际行驶中仍有一定的风险。

第三，现有方法的稳定性和扩展性问题。常见的自动驾驶结构中，间接感知型结构较为传统，往往组合了目标检测和跟踪、场景分割、三维重建等多个模块，系

统复杂度较高，需要大量的计算资源，投入商用的成本较高。其他两种结构仍处于研究的初期，稳定性和扩展性有待继续论证。

1.3 本文主要研究内容及组织结构

本文设计了一种基于计算机视觉和深度学习的端到端的自动驾驶方法，并针对该方法在未知场景中的泛化能力问题进行了研究。首先介绍典型的自动驾驶方法，随后描述为课题研究设计实现的端到端的自动驾驶系统。本文设计的自动驾驶方法无需使用人工设计的特征，也不依赖与类似“if…else”的规则引擎进行决策。无人车通过学习以车体为第一视角的图像，计算当前合适的转向角度。为了验证算法在真实环境中的有效性，本文设计实现了一个微缩智能车系统。从实验结果来看，算法具有良好的稳定性和泛化能力，能够在较为复杂的有标志线和障碍物的环境下进行自动驾驶和避障。

第一章，介绍自动驾驶技术研究的背景和意义，论证了研究的必要性。叙述了国内外自动驾驶项目的发展历史和微缩智能车的发展历史，以及使用微缩智能车进行自动驾驶技术研究的优势。简要阐述了自动驾驶的典型方法和各自的优缺点，就目前自动驾驶技术面临的挑战进行了分析。同时对本文主要研究内容，即端到端的自动驾驶系统设计进行了简要介绍。

第二章，介绍基于计算机视觉的三种自动驾驶的典型方法，包括间接感知型、直接感知型和端到端控制。间接感知型结构基于计算机视觉的多项子任务，需要组合这些任务进行决策。直接感知型结构基于一些自动驾驶关键指标，需要利用学习模型学习这些指标。端到端控制方法可抽象成图像分类或回归问题，直接对动作进行学习。

第三章，介绍全文算法设计的核心部分。介绍基于计算机视觉和深度学习的端到端的自动驾驶算法的设计和实现，并从理论角度解释算法设计的各要素。提出了基于卷积神经网络的自动驾驶连续转向角预测方法，以及提升训练效果和泛化能力的网络预训练和防止过拟合方法。

第四章，介绍基于微缩智能车的自动驾驶实验及分析。叙述了智能车平台的设计方法、训练数据的采集方法、训练过程和结果、实测中的表现和评价等，并对训练好的卷积神经网络进行了可视化分析。本文在实验过程中采集的训练样本和实验视频见网址：http://pr-ai.hit.edu.cn/research/bai2017intelligent.html

第 2 章 基于计算机视觉的自动驾驶典型方法

2.1 引言

基于计算机视觉的自动驾驶典型方法主要分为三种。分别是间接感知型（Mediated Perception），直接感知型（Direct Perception）和端到端控制（End-to-End Control）。间接感知型方法将自动驾驶任务分为目标检测、目标跟踪、场景语义分割、相机模型和标定、三维重建等子任务。直接感知型方法学习人为设计的驾驶关键指标。端到端控制建立了输入到动作的映射，可转化成图像分类或回归问题。本章介绍自动驾驶数据集，描述三种基于计算机视觉的自动驾驶方法，并对这三种方法的特点进行对比分析。

2.2 自动驾驶数据集

自动驾驶数据集为自动驾驶算法提供了训练数据，同时也为各种算法之间的比较提供了平台。在自动驾驶领域，KITTI[13] 数据集和城市交通数据集 Cityscapes[17] 的使用较为广泛，这两种数据集几乎涵盖了自动驾驶领域的所有子任务，包括三维重建、姿态估计、目标检测等，为自动驾驶的研究提供了真实的交通数据。数据集的收集和标注是一项繁重的工作，特别是在光流计算和图形分割中，需要对每一个像素点进行标注，为此研究人员付出了很多努力。大规模标注数据集的出现使得基于监督学习的模型特别是深层模型能够有效训练，有力的推动了这一领域的发展。

Geiger 等于 2012 年开放了 KITTI 数据集[13]，同时提供了用于光流计算、立体视觉、SLAM、目标检测等任务的算法比较平台[1]，可以进行算法性能的比较。数据集由采集车在真实交通场景中采集，采集车配备了高分辨率彩色和灰度双目相机、Velodyne 3D 激光雷达、高精度 GPS/IMU 导航装置等，如图 2-1 所示。数据集中包括 400 多个典型场景，包括高速公路、城市公路、乡间小路等，涵盖不同的时间和天气条件。收集的交通目标包括真实交通场景中的静态目标和动态目标，所有图像均经过矫正，数据中使用三维 CAD 模型记录目标的运动，同时提供了目标的三维点云信息。该数据集可用于光流计算、三维重建、立体视觉、目标检测、目标跟踪、语义分隔等多个任务中。

[1] http://www.cvlibs.net/datasets/kitti/

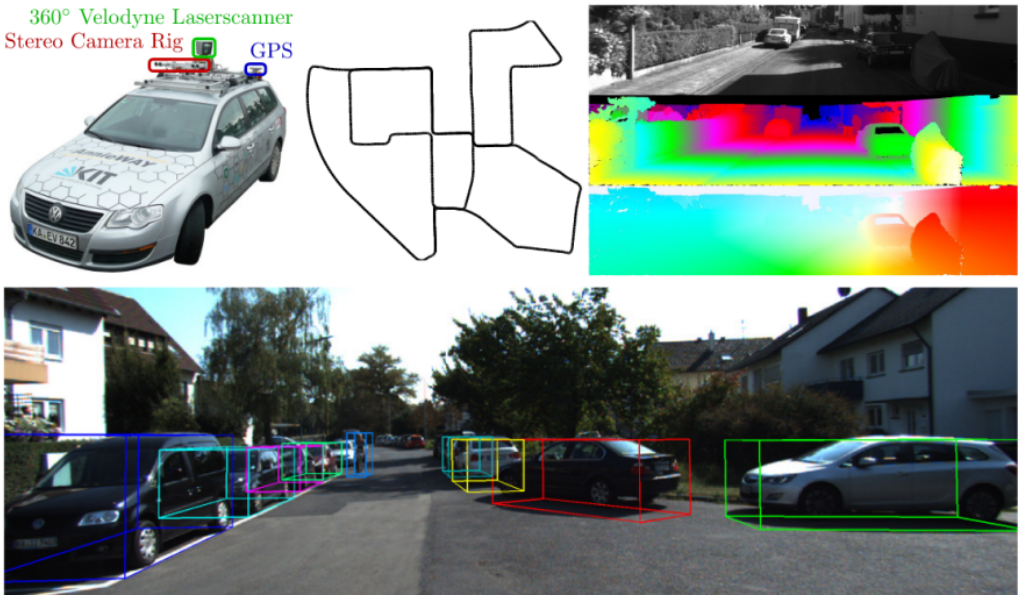


图 2-1 KITTI 数据采集车和采集场景示意图[13]

KITTI 设置了针对多个任务的评测比赛。SLAM 评测包括 22 个立体视觉视频序列，总里程超过 39.2km，评价标准是图像识别结果与激光雷达和测量结果之间的误差。道路检测比赛包括 600 个不同的训练图像和测试图像，包括了多种类型的道路和标志线，使用最大 F1 度量[18] 进行评价。立体视觉评测包括 200 个训练场景和 200 个测试场景，每个场景包括 4 幅彩色图像，使用错误识别的像素占整个图片的比例进行评价。目标识别评测包括 2D 目标识别和 3D 目标识别，其中 3D 目标识别数据包括 7481 个训练样本和 7518 个测试样本，目标分为机动车、行人、自行车三种类型，准确率使用检测结果和真实结果中目标边框的交并比 IOU 衡量，如式 (2-1) 所示，其中 $DetectionResult$ 表示检测结果，$GroundTruth$ 表示真实结果。

$$IOU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth} \tag{2-1}$$

Cityscapes 数据集[17] 是针对城市交通场景的大规模标注数据集。标注内容包括语义分割（Semantic Segmentation）、实体分割（Instance Segmentation）、像素注释等，目标种类达到 50 种。数据采集在欧洲的 50 多个城市进行，历时几个月。数据涵盖了不同的天气条件，大量的运动目标，不同的目标背景等，具有显著的多样性。图形以采集车为第一视角，给出了目标的三维坐标。

KITTI 和 Cityscpace 等数据集主要集中于算法的评价，忽略了自动驾驶中的长距离行驶问题。在长距离行驶中，光照条件、天气条件等环境在不断变化，这也给自动驾驶算法带来了很大挑战。为了使这一问题引起更多关注，Maddern 等于 2017 年发布了 RobotCar 数据集[19]，该数据集主要关注长距离的自动驾驶问题，数据经

过长达 1 年的采集，总里程达到 1000 公里。采集车在固定路线下驾驶 100 余次，共采集了 20TB 的图像、LIDAR 和 GPS 数据。RobotCar 装备的传感器包括一个立体相机、三个单目相机、两个 2D 激光雷达、1 个 3D 激光雷达和 GPS。图 2-2 a) 显示了在不同的天气和季节中，同一个场景可能发生很大的变化，图 2-2 b) 显示了由于人为因素而导致的场景变化。因此，使用长时间、远距离的自动驾驶数据集进行训练和测试很有必要。使用 RobotCar 数据集可以测试算法在应对传感器、照明、季节变化中存在的问题，使算法那能够应对远距离驾驶场景。

a) 场景随天气变化示意图　　b) 场景因人为原因发生变化示意图

图 2-2 RobotCar 数据集举例[19]

2.3 基于间接感知型结构的自动驾驶技术

基于间接感知型结构的自动驾驶技术主要包括目标检测、目标跟踪、场景语义分割、相机模型和标定、三维重建等子任务，该结构通过综合多个子任务的检测结果，建立完整的环境表示。每项子任务都处于计算机视觉的前沿研究领域，都在不断研究和发展中。因此，间接感知型结构能够不断吸收最新的研究成果，不断提高自动驾驶的水平。其缺点在于系统冗余性和复杂度较高，投入商用的成本较高。本节将对间接感知型结构中包含的各项子任务进行介绍。

2.3.1 目标检测

目标检测的可靠性对自动驾驶至关重要。自动驾驶车辆在交通环境中与其他交通工具和行人共享道路资源，尤其在城市交通中，交通工具种类繁多，行人、宠物等目标也经常出现。自动驾驶车辆需要检测这些目标的位置并进行分类，根据目标的种类采取相应措施。在交通环境中，由于目标的种类繁多，且目标之间易发生混淆和遮挡，因此检测难度较大。

视觉传感器方面，无人车使用的视觉传感器除了普通相机之外还有包括红外相机、热感应相机等，这些相机可以直接检测到温度较高的植被和行人。使用多个不同类型的视觉传感器可以进行优势互补，防止特殊天气条件下某种视觉传感器信息不准。无人车通常需要对多个视觉传感器图像进行拼接和融合，已有一些学者在融合多类型的视觉传感器信息方面进行了研究[20]。

传统的目标检测方法一般包括图像预处理，兴趣区域（ROI）提取，候选区域分类和微调边框等。

图像预处理方法一般包括图像滤波、去噪、直方图均衡化、白化等，还可能要根据相机模型进行畸变矫正和重投影。具体采用的图像预处理方法与应用密切相关，预处理对整个目标检测有重要作用。

ROI 提取可以使用多尺度的滑动窗口，对图像中的兴趣区域进行滑动搜索。全局搜索效率低下，许多学者提出了启发式的搜索方法，通过目标比例、尺寸、位置等特征进行筛选，减少候选区域数量。Broggi 等[21] 提出了一种针对交通场景中的行人目标的 ROI 提取方法，该方法利用行人的形态学特征和行人目标的对称性来确定候选区域。Dollar 等[22] 对使用滑动窗口进行行人检测的方法进行了综述，指出这些方法在中等以下分辨率的图像中能够取得较好的效果。Uijlings 等[23] 提出了选择性搜索方法（selective search），该算法结合了穷举搜索和图像分割，使用多种图像特征来衡量候选区域之间的相似性，同时给出了使用选择性搜索和 SVM 分类器来进行目标识别的算法框架，如图 2-3 所示。

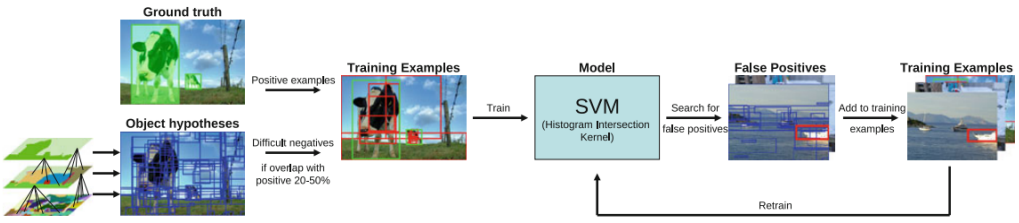


图 2-3 基于选择性搜索的目标检测算法[23]

候选区域分类中，由于候选区域数量庞大，因此需要从候选区域中去除位于图像背景的区域，只保留含有目标的区域。早期的相关工作需要首先对候选区域进行特征提取，随后由分类器分类。例如 Viola 等[24] 提出了一种基于 AdaBoost 分类器进行候选区域分类的算法，取得了不错的效果。AdaBoost 二分类公式如式 (2-2) 所示。

$$G(x) = sign(\sum_{m=1}^{M} \alpha_m G_m(x)) \tag{2-2}$$

其中 α_m 是第 m 个分类器的权重，衡量分类器的准确程度，$G_m(x)$ 是第 m 个分类

器的分类结果，$sign$ 是符号函数，$G(x)$ 是最终分类结果。在迭代过程中，训练数据的权重不断变化，误分类样本的权重不断增大，在下一轮分类中起更大的作用。AdaBoost 通过组合多个弱分类器，达到了比强分类器更好的效果。

Dalal 等[25] 提出了一种结合方向梯度直方图特征（HOG）和线性支持向量机(SVM) 进行候选区域分类的方法，该方法先进行特征提取，随后进行分类。另外一种比较成功的算法是 DPM 方法[26]，该算法首先计算梯度直方图，然后用 SVM 训练得到物体的梯度模型，最后进行模板匹配和分类。Sermanet 等[27] 将卷积神经网络引入交通场景中的行人分类，提出了一种使用无监督的预训练特征和卷积神经网络结合的方法，取得了不错的效果。

近年来，随着深度学习[28] 技术的兴起，使用卷积神经网络进行目标检测取得了许多成果。其中代表性成果是 R-CNN 算法[29]，算法结构如图 2-4 所示，后续的许多算法是由 R-CNN 改进而来的。R-CNN 首先提取 ROI 区域，对每个候选区域按照与标记边框的 IOU 值打分，如式（2-1）所示。根据 IOU 值和阈值，每个候选区域被标记为某个物体类别或背景类别。随后，对 AlexNet 网络[30] 的全连接层进行调整，使其输出层与类别数一致，用有标记的候选区域和类别对网络权重进行微调。训练完成后，将每个候选区域通过 AlexNet 提取特征，第五个池化层权重作为该图片的特征进行保存，这些特征被输入到 SVM 分类器中进行分类。R-CNN 在当时突破了传统方法的最好效果，其缺点在于消耗的计算资源较大。随后有很多学者对 RCNN 的计算效率进行了优化，提出了 fast-RCNN[31]，faster-RCNN[32]，YOLO[33] 等算法。其中 YOLO 算法不再使用 ROI 提取，而直接学习目标位置信息，速度得到很大提升，甚至可用于实时视频目标检测。

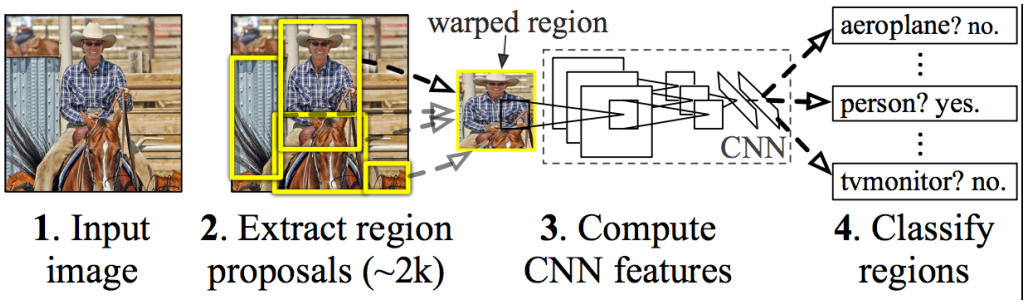


图 2-4 RCNN 目标检测算法结构图[29]

R-CNN[29]，fast-RCNN[31]，faster-RCNN[32]，YOLO[33] 等算法在通用型目标检测数据集 PASCAL VOC[34] 上取得了不错的成绩，但是在自动驾驶数据集 KITTI 上的表现略差一些。主要原因是交通场景包含大量不同尺度的目标，同时还有严重的遮挡和截断现象，因此基于候选区域的算法会出现一些问题。近年来许多学者针对自动驾驶数据集对候选区域的选择进行了改进。Chen 等[35] 于 2015 年提出了一种

针对利用立体视觉图像生成 3D 目标候选区域生成方法，可以用于 RGB 和 RGB-D 图像，于 2016 年[36] 又提出了针对单目视觉图像的 3D 目标检测方法，在 KITTI 数据集上均取得了不错的效果。Xiang 等[37] 针对交通图像中的目标遮挡问题，提出了一种编码目标重要性质的目标表示方法，可用于 2D 和 3D 目标检测，于 2016 年[38] 又提出一种使用类别信息的目标检测方法，该方法使用子类信息引导候选区域生成，是目前 KITTI 目标检测的最好成绩。KITTI 中关于车辆检测和方向估计的评测排名如表 2-1 所示。

表 2-1 KITTI 车辆检测和方向估计评测排名表

方法	简单	中等难度	难	运行时间
SubCNN[38]	88.62%	90.67%	78.68%	2s/GPU
Mono3D[36]	86.62%	91.01%	76.84%	4.2s/GPU
3DOP[35]	86.10%	91.44%	76.52%	3s/GPU
3DVP[37]	74.59%	86.92%	64.11%	40s/8 cores
SubCat[39]	74.42%	83.41%	58.83%	0.7s/6 cores

2.3.2 目标跟踪

自动驾驶中目标跟踪的目的是实时掌握交通环境中车辆、行人等目标的位置、速度和加速度等信息，并预测目标未来可能的位置，预测可能发生的碰撞，这些信息对于自动驾驶汽车至关重要。例如，自动驾驶汽车需要根据自身和其他目标的速度、距离等信息判定何时采取刹车动作，如果对方高速行驶，则应提前采取刹车动作。目标跟踪面临着诸多挑战。例如，目标被其他目标遮挡，多个目标聚集在一起难以分辨，行人目标的形态多样，光照对视觉传感器的成像产生影响等，都会对目标跟踪造成困难。传统的目标跟踪方法主要分为三类：基于实时检测、基于模板匹配和基于贝叶斯滤波。

基于实时检测的方法利用分类器对目标和背景进行分类，置信度较高的区域被认为是目标区域，这种方法的缺点在于依赖于目标特征，在交通场景中目标经常相互遮挡的情况下表现较差。

基于模板匹配的目标跟踪主要分为全局模型跟踪、区域模型跟踪和特征跟踪等方法。其中，基于全局模型的方法针对目标的外形特征建立模型，通过目标与模型的匹配程度不断更新模型，具有较好的鲁棒性，缺点在于不适用于几何性质经常发生变化的目标。基于区域模型的方法将目标划分为不同的部件，分别进行建模和跟踪，如将人体划分为头部、四肢、躯干等，在目标不受遮挡的情况下跟踪较为

稳定。基于特征跟踪的方法一般包括提取特征和匹配特征两步，首先建立运动目标的特征模型，随后利用该模型对当前帧的目标进行匹配，代表方法是 MeanShift 均值漂移[40]，该方法的优点在于跟踪稳定性好，对目标的形状和尺度等不敏感。

基于贝叶斯滤波的目标跟踪方法的原理是，给定当前观测和之前的目标状态，估计后验概率密度函数，典型的滤波算法有卡尔曼滤波[41]、粒子滤波算法[42] 等。卡尔曼滤波器可以根据一组有限的、带有噪声的观测序列预测目标实际的位置，但是只适用于线性高斯系统。粒子滤波算法使用蒙特卡罗方法，通过随机样本调节粒子的权重，从而近似真实的概率分布。

传统的目标跟踪方法仍在不断发展中，该领域也不断涌现新的解决方法。主要包括基于数据关联的目标跟踪和基于能量最小化的目标跟踪。基于数据关联的目标跟踪方法可以用于多目标跟踪，该方法以目标跟踪的结果与连续多帧图像中的轨迹计算关联概率，判定每条轨迹的真实性，从而获得真实的目标轨迹。相关的方法包括多假设检验方法、动态规划方法、联合概率数据关联方法等。数据关联方法的优点在于抗干扰能力较强，利用多帧图形信息，降低了单帧检测错误而带来的误差，缺点在于算法对计算能力和存储性能的要求较高。基于能量最小化的目标跟踪方法认为目标在运动过程中具有连续性，其外观、运动参数、运动范围在多帧图像中具有一致性，因此可以建立合理的能量函数对轨迹进行迭代求解，该方法在实际应用中需要综合考虑对象的运动特征、外观以及其他约束条件，使得能量函数能够准确建模目标的运动过程。

在 KITTI 目标跟踪评测中，Lenz 等[43] 提出了一种基于最小能量函数的目标跟踪方法，针对在多目标跟踪中联合数据优化而导致的计算资源昂贵的问题，提出了在有限计算资源和存储空间上进行视频目标跟踪的近似在线算法，达到了当时 KITTI 车辆检测的最好效果。Yoon 等[44] 提出了一种使用单目相机进行在线多目标跟踪的方法，该方法构造了一个目标相对运动网络来建模目标之间的相对运动，使用相对运动信息保证跟踪的鲁棒性。Choi 等[45] 提出了一种近似在线的多目标跟踪方法，使用一种基于光流关键点特征描述子 ALFD，描述每个目标块之间的相对运动关系，衡量目标块之间的相似程度。例如，对于两个目标块 $box1$ 和 $box2$，计算 $box1$ 投影到 $box2$ 的相对运行表示 $P'(d_1, d_2)$ 和 $box2$ 投影到 $box1$ 的相对运动表示 $P'(d_2, d_1)$，则 ALFD 特征如式 (2-3) 所示。其中 $n(d_1, d_2)$ 是两个目标块中总的关键点的个数，λ 是一个正则项，用来约束关键点个数的影响，文中 $\lambda = 20$，该方法在 KITTI 数据集上达到了很好的效果。

$$P(d_1, d_2) = \frac{P'(d_1, d_2) + P'(d_2, d_1)}{n(d_1, d_2) + \lambda} \tag{2-3}$$

相对于贝叶斯滤波和最小能量函数法，Xiang[46] 等于 2015 年提出了一种基于马尔科夫决策过程（MDP）的多目标跟踪方法，将强化学习（Reinforcement Learning）[47] 引入目标跟踪。该方法将学习目标区域的相似性函数看做学习 MDP 中的最优策略，MDP 根据当前状态和目标的历史状态进行决策。MDP 中将目标分为激活（Active）、非激活（Inactive）、跟踪（Tracked）、丢失（Lost）等状态，如图 2-5 所示。同时，该方法使用反向强化学习[48] 构造奖励函数，学习到的最优策略即区域相似性函数。该方法取得了 KITTI 数据集上的最好成绩之一。

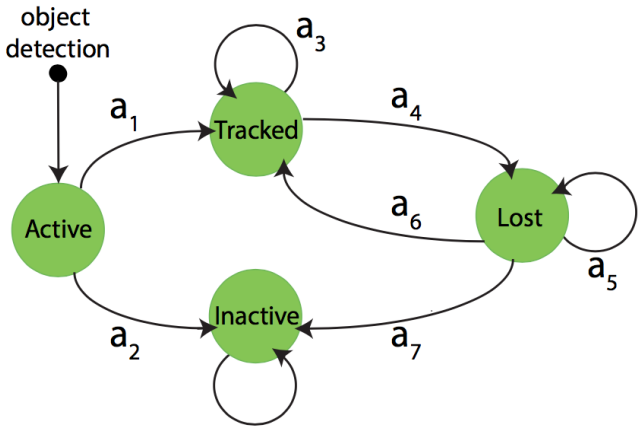


图 2-5 Xiang 等建立的目标跟踪 MDP 示意图[46]

2.3.3 场景语义分割

场景语义分割是自动驾驶技术的一个重要分支，其目的是将一幅场景图像中的每个像素点都归属到某个类别中，典型的分割结果如图 2-6 所示。交通场景中一般将图像分割成车、行人、道路等，为自动驾驶车辆理解环境提供了重要参考。该问题的难点在于交通场景较为复杂且种类繁多。

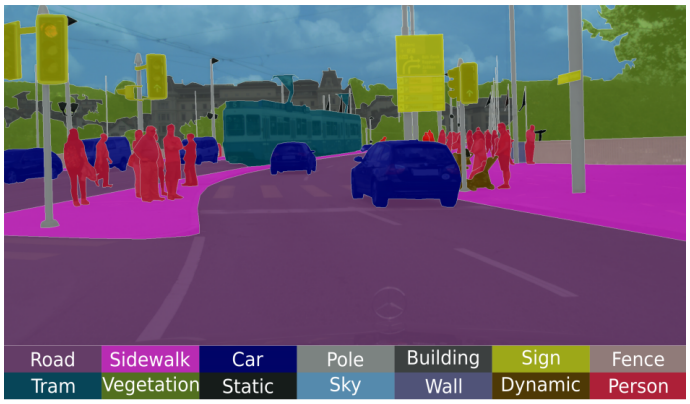


图 2-6 Cityscape 数据集上典型的场景语义分割结果

传统的语义分割方法基于概率图模型，概率图模型利用图来表征变量之间的依赖关系，计算变量之间的条件概率分布，广泛用于概率推理。条件随机场（CRF）是一种典型的概率图模型，语义分割被抽象成 CRF 的最大化后验概率（MAP）问题。定义能量函数为单点势能和成对势能之和，如式 (2-4) 所示。

$$E(x)=\sum_{i\in V}\psi_i(x_i)+\sum_{i\in V,j\in N_j}\psi_{ij}(x_{ij}) \tag{2-4}$$

其中 N_i 表示所有与变量 x_i 相邻的点的集合，V 表示图 $G(E,V)$ 中的节点的集合，成对势能表示使得相邻像素点标记为同种类别的平滑函数。Shotton 等[49] 提出了一种纹理滤波的图像特征表达方法，将其与 CRF 中的低级图形特征进行组合，取得像素级的分割效果。但像素表达能力极为有限，根据像素亮度、颜色和坐标等无法准确判断类别，只能反映图像的局部信息且缺乏稳定性。因此，有人提出了基于超像素的 CRF 来进行图像分割，超像素具有较好的区域表达能力，使同一区域的像素具有某种共同的视觉特征，该基于无监督学习。典型的成果有基于 K-均值[50] 的图像分割方法和基于均值漂移（Mean-shift）[51] 的图像分割方法等。此外，有一些方法考虑目标之间的共现关系，例如车与马路共同出现的概率比车与办公室共同出现的概率高。Krahenbuhl 等[52] 提出了一种高效的全连接 CRF 模型，该模型考虑了像素之间的层次化关系和连接关系，取得了不错的效果。

近年来，有学者将卷积神经网络（CNN）引入图像语义分割任务中。使用 CNN 进行图像语义分割的最原始的想法是对每个像素点为中心的图像块进行分类，从而得到全部像素的类别，但这种方法非常消耗计算资源。一个改进的想法是，在 CNN 中不使用全连接层，使用全卷积网络输出分割结果，这种方法的问题在于 CNN 中随着层数的递增，输出的分割结果的尺度会越来越小。farbet 等[53] 提出了一种改进的分割方法，使用图像金字塔将原图转换为多种尺度的图片，分别用 CNN 进行分割，随后用对象之间的关系树对结果进行调整，得到分割结果。Long 等提出了一种[54] 使用全卷积网络进行图像分割，使用 CNN 中的多个采样层特征进行上采样，最后对所有上采样结果进行微调。Noh[55] 于 2015 年提出了一种使用卷积和反卷积层（deconvolution）结合的图像语义分割方法，使用一个 CNN 和一个镜像 CNN 结合进行图像分割，在 VOC 2012[56] 数据集上取得了当时的最好效果，算法结构如图 2-7 所示。

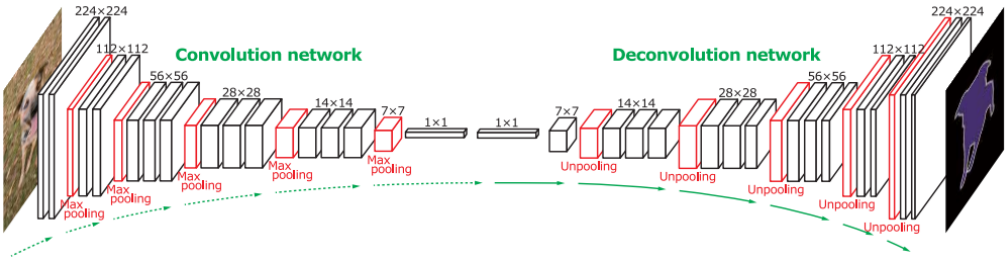


图 2-7 Noh 等提出的反卷积网络结构示意图[55]

2.3.4 相机模型和标定

相机观察到的光线由物体发出，通过透镜到达相机平面，关于这个过程的几何关系的研究是计算机视觉和自动驾驶技术的一个重要方面。针孔相机模型是一个简单实用的相机模型，它假设光线只能从墙中的小孔中穿过，在墙的后面成像。相机标定的目的是估计相机内参数和外参数，这些参数是联系像平面和三维世界的纽带，同时可以矫正使用透镜带来的误差。在自动驾驶技术中，建立相机和真实物理世界的关系尤其重要。

根据 zhang 等[57] 提出的“张氏标定法”，相机标定中一般定义四个坐标系，各坐标系之间的关系如图 2-8 所示。

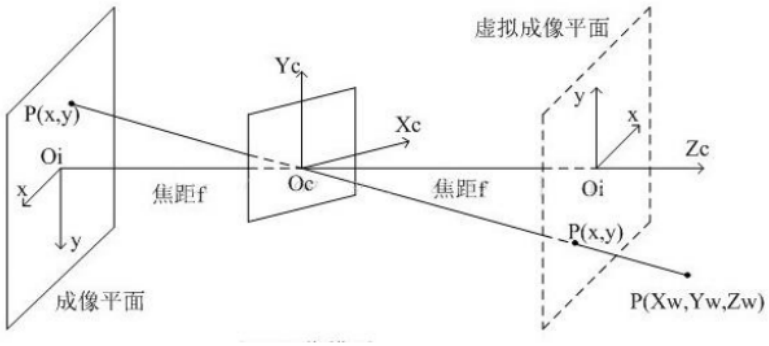


图 2-8 相机标定中四个坐标系之间的关系

（1）图像坐标系 (u, v)

图像坐标系以像素为单位。以图像左上方为原点，向右为 u 的正方形，向下为 v 的正方形，建立图像坐标系 O_0uv。

（2）成像平面坐标系 (x, y)

成像坐标系以毫米为单位，表示像素在图像中的物理位置。x 轴和 y 轴的正方向与坐标系 O_0uv 的定义相同，成像坐标系的原点在相机光轴和图像平面的交点处，记为 O_1xy。

（3）世界坐标系 (X_w, Y_w, Z_w)

世界坐标系是自定义的真实世界的三维坐标系，用于描述物体相对空间关系和相对位置。在标定过程中，为了方便，世界坐标系往往以标定板平面为 XOY 平面，单位为 mm。

（4）相机坐标系 (X_c, Y_c, Z_c)

以相机光心为原点 O_c，通过原点垂直于成像平面的光轴为 Z_c 轴，X_c 轴和 Y_c 轴与成像平面的坐标一致，建立相机坐标系，单位为 mm。

在线性针孔相机模型下，位于世界坐标系的点转换到图像坐标系下需要经过四个步骤，如图 2-9 所示。

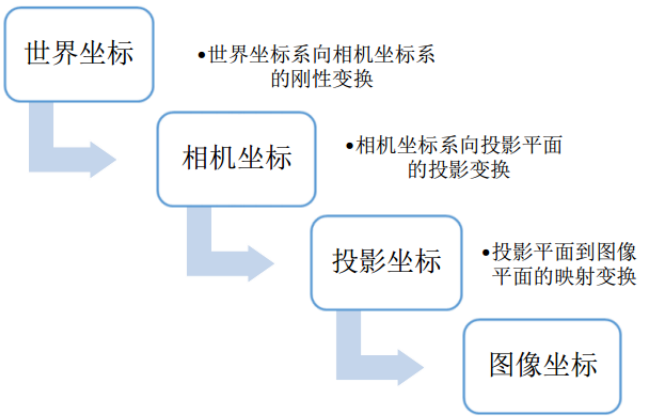


图 2-9 世界坐标系到图像坐标系的转换步骤

位于世界坐标系的点，通过旋转矩阵 **R** 和平移矩阵 **T** 可以完成由世界坐标系 (X_w, Y_w, Z_w) 到相机坐标系 (X_c, Y_c, Z_c) 的刚性变换，如式（2-5）所示。

$$\begin{bmatrix} X_c \\ Y_c \\ X_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2-5}$$

相机坐标系通过乘以由焦距 f 构造的矩阵，可以转换到成像平面坐标系下，如式（2-6）所示。

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{2-6}$$

成像平面坐标系通过式（2-7）可以转换到图像坐标系下，其中 dx 和 dy 分别是 x 方向和 y 方向的像素尺寸，(u_0, v_0) 为主点位置坐标，s 为缩放因子。

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}\frac{1}{dx}&0&u_0\\0&\frac{1}{dy}&v_0\\0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix} \tag{2-7}$$

综合式（2-5），（2-6），（2-7），线性相机模型的总体表达是如式（2-8）所示。

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x&0&c_x\\0&f_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}\mathbf{R}\ \mathbf{T}\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \tag{2-8}$$

或

$$s\mathbf{m}'=\mathbf{A}\begin{bmatrix}\mathbf{R}\ \mathbf{T}\end{bmatrix}\mathbf{M}' \tag{2-9}$$

其中 **A** 为相机的内参数矩阵，**R** 和 **T** 为相机的外参数矩阵。

透镜由于制造的原因会产生径向和切向的畸变，使实际成像平面偏离理想投影平面，从而使投影坐标产生误差。因此，在线性投影模型的基础上还需要对畸变进行修正。相机标定可以通过棋盘格进行，通过拍摄多种姿态的棋盘格照片，对棋盘格进行角点检测，得到多组图像坐标系与世界坐标系的对应关系，从而估计内参数矩阵和外参数矩阵。相机标定的主要评价指标是反向投影误差，除此之外，标定的速度、鲁棒性等也是要考虑的因素。

自动驾驶车辆中常使用全景相机，这种相机可以对 360° 视角进行全方位观测。全景相机不可以使用线性投影模型，因为全景相机对图像的扭曲很大，应考虑由镜头引起的镜面反射和折射。Geyer 等[58] 针对所有的中心全景相机系统提出了统一的投影模型，被广泛用于标定工具包中[59]。事件相机（Event Cameras）在最近被引入自动驾驶系统中，该相机的曝光需由事件触发，事件包括亮度变化、位置变化、标记出现等。事件相机可以减少普通相机拍摄图片中的冗余，关注环境中的变化信息。

2.3.5 三维重建

自动驾驶中的常见的三维重建技术主要基于立体视觉。基于立体视觉的三维重建模仿人眼成像原理，由两台前向平行对准的相机进行成像，从两幅图像中寻找匹配点来估计深度信息，从而重建三维环境。深度信息在自动驾驶中可以用来探测障碍物距离、探索安全区域等。

立体视觉系统主要包括图像获取、相机标定、特征提取、立体匹配、深度计算、重建等部分。其中，图像获取主要通过双目立体相机；相机标定用于计算相机内

参数、外参数、畸变系数等；特征提取是指提取二维图像的角点、边缘、轮廓等特征，要求特征具有明显的区分性和独立性，在匹配中需要使用提取到的特征；立体匹配将两幅图像的相同像素点进行对应，是立体视觉技术的关键；深度计算是指从匹配的 2D 图像中获取深度信息，主要依据三角测量原理从视差图中估计深度，影响深度计算的因素有标定误差、特征提取精度和匹配精度等；重建主要目的是恢复三维场景的表面信息，视差图中仅包含部分点的视差，因此需要由视差图插值来进行重建。

立体匹配是三维重建的关键，是立体视觉技术中最为复杂的部分。立体匹配算法可以分为基于区域的匹配算法、基于特征的匹配算法和基于相位的匹配算法三类，每种算法都建立在一定的约束条件下。基于区域的匹配算法主要利用窗口之间灰度的相关程度，在纹理丰富的场景中有较好的表现，缺点是算法对畸变较为敏感，且运行速度较慢。基于特征的匹配算法依赖于角点、边缘等特征，其前提是特征提取的效果较好，在图像发生畸变的情况下仍有较好的效果，缺点是由于特征具有稀疏性，所以得到的深度图也是稀疏的，对后续的三维重建有一定的影响。基于相位的匹配方法利用图像局部结构，假设局部结构之间的相位应该相等。其优点在于对畸变不敏感，对噪声也有一定的抑制能力，缺点是当局部结构存在的假设不成立时会存在相位奇点问题。立体匹配策略可以分为全局最优搜索策略和分层匹配策略。全局最优搜索策略是指在匹配中，使用多个全局约束条件，以能量方程的最小化的方式搜索，常用的方法有动态规划法、松弛法、分割法等。分层匹配策略是从全局到局部的多尺度匹配策略，主要利用图像金字塔等方法。

近年来，深度学习也开始应用到立体视觉技术中。Mayer 等[60] 于 2016 年提出了一种编码-解码的深度学习模型用来进立体匹配。Sergey[61] 提出了多种通用型的卷积网络（CNN）结构来解决两个图像块的匹配问题，能够在匹配图像之间存在遮挡、相机设置差异、光照差异等情况下完成匹配，网络结构共同点在于均使用两个 CNN 提取特征，随后将提取的特征进行组合，不同之处在于两个 CNN 之间是否共享权重以及两幅图片之间是否有预处理的差异。网络最终输出以 l_2 正则和折页损失（hinge loss）衡量误差，如式（2-10）所示。

$$\min_{w} \frac{\lambda}{2}\|w\|_2 + \sum_{i=1}^{N} max(0, 1 - y_i o_i^{net}) \tag{2-10}$$

其中 N 表示样本个数，o_i^{net} 表示第 i 个样本通过 CNN 的输出，$y_i \in \{-1, 1\}$ 表示第 i 个样本的真实标签，w 代表网络权重，$\|w\|_2$ 为正则项，λ 为常数。该方法在多个任务集上取得了不错的效果。

Luo 等[62] 在此工作的基础上，提出了改进的图像匹配方法。由于左右两幅图

像的输入尺寸不同，因此在卷积神经网络提取特征后不对特征进行连接，而是采用内积的形式输出二者的相似概率，网络结构如图 2-10 所示。采用这种方法的好处是极大提升了速度，而且得到了图像之间的匹配概率分布，该方法在 KITTI 2012 和 KITTI 2015 的立体视觉评测中取得了当时最好的成绩。

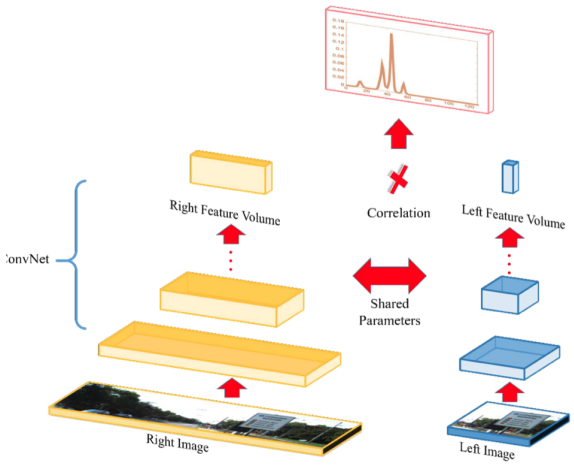


图 2-10 Luo 等提出的基于深度学习的立体匹配算法示意图[62]

2.4 基于直接感知型结构的自动驾驶技术

直接感知型结构在间接感知型结构的基础上，对减少系统复杂度方面进行了改进。该结构的思路是，传统的间接感知型结构需要将各模块的输出结果转换为道路交通环境的完整表示，这种转换比较困难且存在误差，因此转而直接学习与自动驾驶相关的关键指标。例如，无人车需要获取与附近车辆的距离。在间接感知型结构中，需要首先利用目标检测方法对车辆进行检测，随后由相机模型转换为距离。直接感知型结构不使用标准的目标检测方法，而利用神经网络直接学习与附近车辆的距离值。因此，直接感知型结构可以直接从图像中学习道路交通环境的表示，减轻了系统复杂度。

chen 等[14] 提出的 DeepDriving 算法是直接感知型结构的典型代表。DeepDriving 中设计了一个卷积神经网络[28]，网络的输入是无人车的第一视角图像，输出是与交通环境相关的十余个指标，包括与左侧标志线的距离、与右侧标志线的距离、与前车的距离等，具体如表 2-2 所示，这些指标共同构成了高速公路交通环境的完整表示。决策系统通过学习这些关键指标，根据一定的逻辑控制无人车的前进方向和速度。

DeepDriving 网络需要通过监督学习进行训练，但由于这些关键指标是人工设计的，没有专门针对这些指标的交通数据集。因此，DeepDriving 网络的训练依赖

表 2-2 直接感知型结构 DeepDriving 设计的高速公路指标

名称	意义
angle	无人车与道路的夹角
toMarking-LL	无人车与最左侧标志线的距离
toMarking-RR	无人车与最右侧标志线的距离
toMarking-ML	无人车与当前车道左侧标志线的距离
toMarking-MR	无人车与当前车道右侧标志线的距离
distLL	无人车与左侧车道前车的距离
distMM	无人车与当前车道前车的距离
distRR	无人车与右侧车道前车的距离
toMarking-L	无人车在超车中与左侧标志线的距离
toMarking-M	无人车在超车中与当前标志线的距离
toMarking-R	无人车在超车中与右侧标志线的距离
dist-L	无人车在超车中与左侧前车的距离
dist-R	无人车在超车中与右侧前车的距离

赛车仿真环境 TORCS[2]。训练模型基于 AlexNet[30]，在该模型的基础上对输出层节点个数进行修改，输出自行设计的关键指标。训练时使用 Torcs 仿真器采集多个场景中的图像共 484,815 幅，迭代 140000 次后收敛。测试结构表明，训练好的网络不仅在 Torcs 仿真环境中很好的表现，在真实交通环境中也有不错的表现。在 Torcs 环境下训练的系统结构如图 2-11 所示。

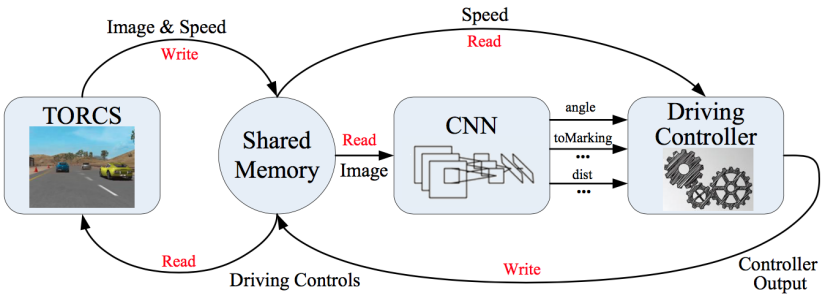


图 2-11 DeepDring 使用 Trocs 进行训练的系统结构示意图[14]

直接感知型结构的优势在于没有单独设计各项任务的检测模块，而直接学习与当前交通环境相关的各项指标，因此在构建完整的交通环境表示的同时拥有较为简单的系统结构。其缺陷在于，由于需要人工设计与交通状况相关的指标，因此难以应对非结构化的交通场景。例如表2-2中设计指标是针对于高速公路环境的，高速公路场景的结构化特征较为明显，但是这种设计不易迁移到普通交通环境，如

[2]http://torcs.sourceforge.net

普通公路和城市道路等。因此，直接感知型结构更适用于特定的交通场景，迁移到一般性的交通场景中仍存在一定的困难。另外，直接感知型结构所需的训练数据在真实交通环境中较难获取，因为精确的距离测量往往需要超声和激光雷达等设备，采集成本高，所以数据采集需要依赖仿真环境。仿真环境到真实交通环境的算法迁移的可靠性需要进一步研究。

2.5 基于端到端控制的自动驾驶技术

基于端到端控制的自动驾驶技术采取另一种思路，不对系统进行任务划分，而直接利用监督学习的方法学习驾驶动作。这种方法的典型代表是 Lecun 等建立了 DAVE 智能车系统[16]，该系统以双目相机图像作为输入，输出左转、右转、直行三个控制命令。因此，DAVE[16] 将自动驾驶问题转为图像分类问题进行研究。近年来，研究者陆续提出了一些基于深度学习的图像分类方法，主要包括 AlexNet[30]，VGGNet[63]，ResNet[64] 等。

AlexNet[30] 于 2012 年提出，在大规模图像分类比赛 ILSVRC-2012（ImageNet Large-Scale Visual Recognition Challenge 2012）[65] 中取得第一名，top-5 错误率仅为 15.3%，远低于第二名。AlexNet 将模型拆分到两个 GPU 上，共训练 6 天时间。激活函数使用受限的线性单元（RELU），显著提升了训练速度，同时使用 DropOut 来防止过拟合。其网络结构如图 2-12 所示。

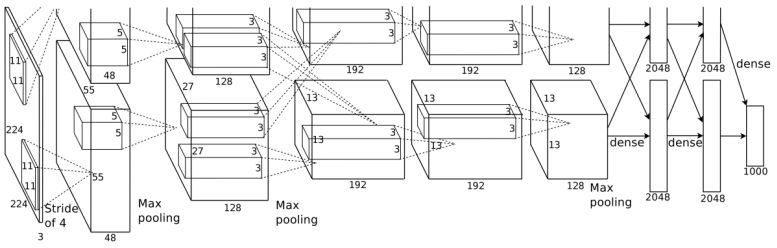


图 2-12 AlexNet 网络结构示意图[30]

VGGNet[63] 在 ILSVRC-2013 上取得第一名，VGGNet 中引入小卷积核，证明了小卷积核相比于大卷积核而言具有更强的表达能力，且可以减少参数个数。VGGNet 在卷积层保持特征大小不变，仅在池化层缩小特征尺度。VGGNet 根据层数的不同分为多个而版本，根据实验表现来看，层数越深实验效果越好。另外，对训练数据进行增广、进行多模型融合也可以提升训练结果。

ResNet[64] 获得了 ILSVRC-2015 的第一名，ResNet 设计的动机是解决深层网络的退化问题。随着网络层数的加深，网络的学习能力增强，但深层网络的错误率有时比浅层网络高，原因是当模型复杂时 SGD 优化困难，从而达到了不好的效

果。ResNet 中增加了一个恒等映射，将原始需要学习的函数 $H(x)$ 转换为 $F(x)+x$，这两种表达的效果相同，但优化的难度却不同。这一想法源于残差向量编码，将问题分解成多尺度的残差问题，起到了加速训练的效果，一个典型的残差模型如图 2-13 所示。ResNet 最深的模型达到了 152 层，相比于 AlexNet 和 VGGNet 而言，在网络深度方面有了很大提升，提升了训练效果。

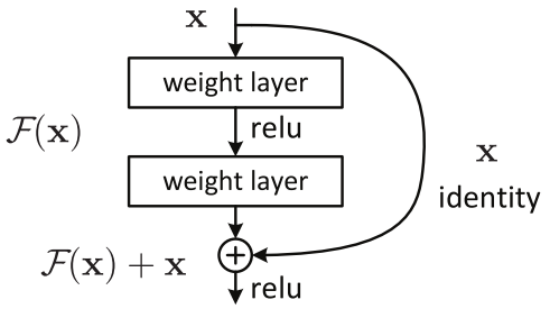


图 2-13 ResNet 残差模块示意图[64]

2.6 基于计算机视觉的自动驾驶方法对比分析

基于计算机视觉的三种自动驾驶典型方法分别为：间接感知型、直接感知型和端到端控制型。三种方法的优缺点对比总结如表 2-3 所示。

表 2-3 三种自动驾驶典型方法对比

方法	模块化程度	数据获取难易程度	系统复杂度
间接感知型	模块化，将任务划分为多个模块	较易，每个子任务均有标准数据集	复杂度高，需转换和集成子任务
直接感知型	无模块化，但输出多个关键指标	较难，精确测量关键指标需要精密传感器	复杂度低
端到端控制型	无模块化，直接输出动作	较易，需采集数据记录图像和动作	复杂度低

2.7 本章小结

本章介绍了基于计算机视觉的三种自动驾驶的典型方法，分别是间接感知型结构，直接感知型结构和端到端控制结构，并详细介绍了每种方法的主要思路和典型成果。三种方法各有优缺点，均在不断的研究和发展中。

第 3 章 基于端到端控制的自动驾驶算法设计

3.1 引言

本章内容是全文的核心，介绍基于计算机视觉和深度学习的端到端自动驾驶算法的设计和实现过程，并从理论上解释算法设计的各要素。提出了基于卷积神经网络的自动驾驶连续转向角预测方法，以及提升训练效果和泛化能力而采取的网络预训练和防止过拟合的方法。

与传统端到端控制方法的不同点在于，传统方法将该问题抽象成一个分类问题来研究，用方向来描述运动，粒度较为粗糙。本文提出的方法将其作为一个回归问题，以转向角来描述运动，对运动的描述更加精确，适应能力更强。

3.2 算法概述

本文设计的算法基于深度学习和端到端控制，将自动驾驶作为一个整体的问题进行研究，建立一个端到端的学习系统。学习系统是由 7 层卷积层和 4 层全连接层组成的卷积神经网络（CNN），网络的输入是安装在无人车前方的单目相机拍摄的图像，输出是一个浮点数，代表要预测的转向角，损失函数由平方误差衡量。CNN 训练完成后，相机拍摄的图像经过 CNN 会被映射成转向角，无人车在前进过程中不断拍摄当前环境的第一视角图像，由 CNN 预测转向角，随后无人车根据预测的转向角实时调整方向。算法整体流程如图 3-1 所示。

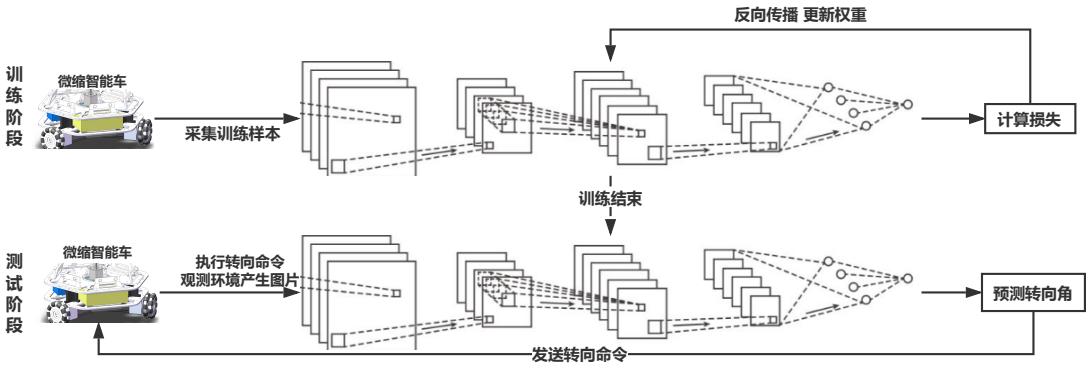


图 3-1 基于端到端控制的算法结构示意图

本文工作在一定程度上受到了 Lecun 等设计的 DAVE[16] 智能车的启发，该系

统以双目相机图像作为输入，输出转向命令。与 DAVE 相比，本文的主要贡献在于：

第一，DAVE 系统建立在立体相机对深度测量的基础上，因此采集数据时使用双目相机。但根据实验发现，使用单目相机与双目相机的效果相当，且可以减少计算量，因此本系统中仅使用一个单目相机采集数据。

第二，DAVE 系统所使用的智能车自身的计算资源有限，需要与远程主机之间通过无线通信的方式传递图像和数据，因此会受到距离、信号噪声和传输速度的影响。而本文自行设计实现的智能车系统其自身的计算能力就可以满足需要，在运动过程中不需要远程主机的协助，具备自主决策的能力。

第三，DAVE 系统将自动驾驶作为一个分类问题，预测转向动作。而本文将其当做一个回归问题，预测转向角度。相比于预测左转、右转等动作而言，转向角对运动的描述更加精确，但训练也更为困难。因此，本文提出了一些提升训练效果和泛化能力的解决方案。

第四，从实际应用的角度看，本文方法降低了对系统实时性的要求，明确了学习目标和应用场景，减轻了 DAVE 训练样本中场景的语义分歧。

3.3 卷积神经网络（CNN）设计与实现

本文设计的卷积神经网络输入时无人车第一视角的图像，输出是转向角，是一个连续值。设计要素主要包括损失函数、网络结构、卷积层、激活函数、优化方法、参数初始化等，本节将分别进行阐述。

3.3.1 损失函数

CNN 输出的转向角度属于连续值，因此使用平方误差衡量损失。网络优化的目标是最小化 CNN 预测的转向角度和人工采取的转向角度之间的平方误差。记 m 为训练样本数量，n 为特征数量，d 为类别数目，损失为 L。则 m 个样本构成的特征矩阵 $\mathbf{X} \in (m, n)$，权重矩阵 $\mathbf{W} \in (n, d)$，样本标签 $\mathbf{y} \in (m, d)$，损失函数表达式如式 (3-1) 所示。

$$\mathbf{L} = \|\mathbf{XW} - \mathbf{y}\|^2 \tag{3-1}$$

3.3.2 网络结构

CNN 共有 11 层，包含 7 个卷积层以及 4 个全连接层，最终输出节点代表转向角。原始图像经过预处理后转为 129×225 大小的图片作为网络输入，接着经过五个卷积核大小为 5×5 的卷积层，通过每层卷积层后特征的尺寸都在缩小，但特征的数目不断增多，分别为 24、36、48、64、64。随后经过两个卷积核大小为 3×3 的卷积层，这两个卷积层对特征进行进一步提取，不进行尺寸的放缩。卷积层结束后进入全连接层，全连接层共有 3 个隐含层，神经元个数分别为 1164、100、50。最终输出为一个节点，代表转向角。网络结构如图 3-2 所示。

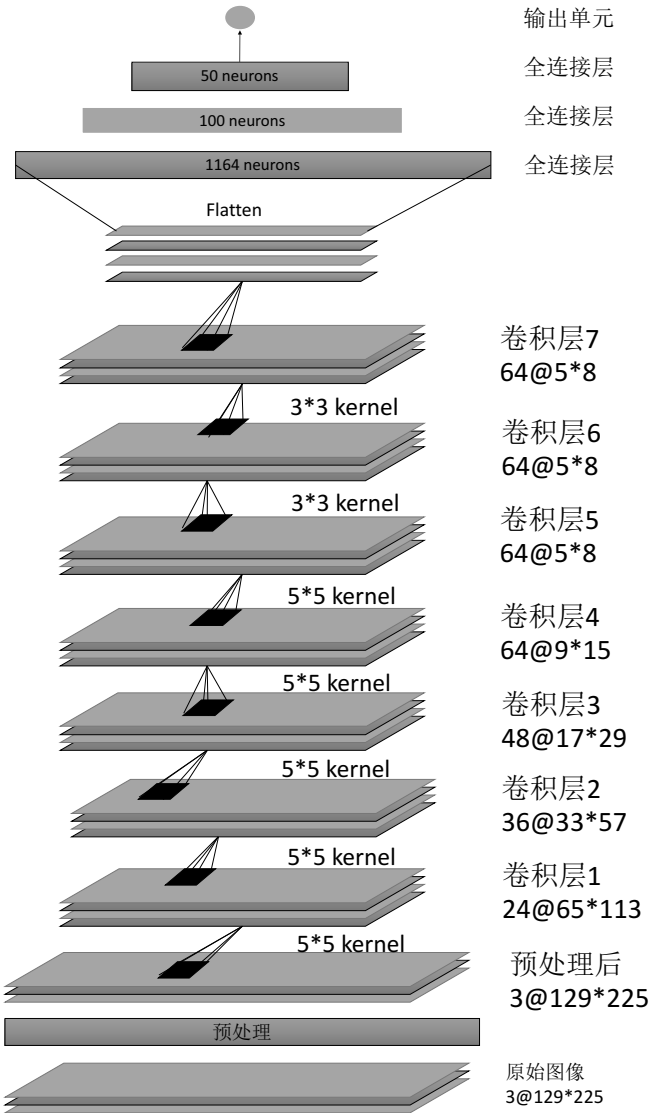


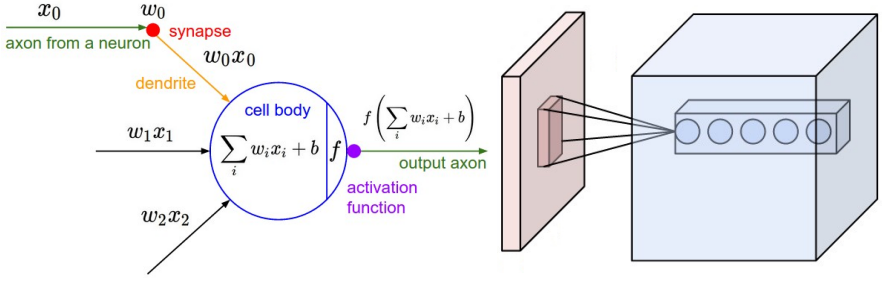
图 3-2 CNN 网络结构示意图

3.3.3 卷积层

卷积层是构建卷积神经网络的核心层，它承担了神经网络中的大部分计算量，其作用是逐层对图像特征进行抽取。卷积层的参数由一些可学习的卷积核构成，每个卷积核在宽度和高度上都比较小。算法中采用的卷积核大小分别为 3×3 和 5×5。相比于较大的卷积核来说，使用较小的卷积核使网络的非线性更强，同时可以减少参数的数目。卷积核在深度上和输入数据体一致，算法中网络第一层卷积核的尺寸为 $5\times5\times3$，分别代表卷积核的宽度、高度和深度，由于输入图像有 3 个通道，因此卷积核的深度也为 3。

网络前向传播时，卷积核会在输入数据体的宽度和高度上滑动，计算与滑过的数据块之间的内积，随后经过激活函数作为下一层神经元激活值，神经元激活值的计算方法如图 3-3 a) 所示。当卷积核滑过整个输入数据体后会产生一个二维的激活图（activation map），每个元素是一个激活值。直观来说，当卷积核学习到对决策有用的特征时（如障碍物、标志线等），表示该特征的神经元就会呈现较高的激活值，上层卷积层学到的特征比底层卷积层的特征更具抽象性和概括性。每个卷积层上有不止一个卷积核，本算法中第一个卷积层共有 24 个卷积核，每个卷积核对输入做卷积会产生一个激活图，因此共产生 24 个激活图。这些激活图在深度方向进行叠加就形成了输出，输出仍在宽度、高度和深度三个方向上排列。

卷积层中神经元之间采用局部连接。在处理图像等高维数据时，让每个神经元都与前一层中的所有神经元进行全连接是不现实的，会使网络可学习的参数太多而达不到好的效果。因此，每个神经元与前一层神经元进行局部连接，连接空间的大小称为感受野（receptive field），表示卷积核的空间尺寸，是一个超参数，连接方式如图 3-3 b) 所示。



a) 单个神经元结构示意图　　b) 神经元局部连接示意图

图 3-3 神经元示意图

输出数据体尺寸除了与输入数据尺寸 W、卷积核大小 F 等因素有关外，还与深度（depth）、步长（stride）和零填充（zero-padding）等参数有关。深度是指卷积

层中卷积核的数目，步长是指在滑动滤波器时每次移动的像素个数，一般情况下步长为 1 或 2，如果滤波器滑动步长 $S > 1$，则输出数据体在空间上会缩小。零填充值 P 用来控制输入数据体的尺寸，常见用法是通过在边缘填充 0 值来使得输出数据体在宽度和高度上大小不变。输出数据体的尺寸 W' 通常情况下计算如式（3-2）所示。

$$W' = (W - F + 2P)/S + 1 \tag{3-2}$$

本算法中输入数据体宽度 $W = 129$，高度 $H = 225$，通道数为 3。第一个卷积层中，卷积核大小 $F = 5$，步长 $S = 1$，零填充 $P = 2$，共有 $K = 24$ 个卷积核。根据公式（3-2），输出数据宽度 $W' = (129 - 5 + 2 \times 2)/2 + 1 = 65$，高度 $H' = (225-5+2\times2)/2+1 = 113$，因此输出数据体尺寸为 $65\times113\times24$。后续卷积层的尺寸可依次推算。需要注意，在设计卷积层的超参数时，应该保证 $W - F + 2P$ 的值能够被步长 S 整除，即卷积核能够整齐的滑过输入数据体。通常可以适用零填充来保证这种整除关系。

卷积层中使用参数共享来减少参数数量。参数共享基于一个假设，即如果一个特征算子在像素 (x_1, y_1) 处是有用的，那么在像素 (x_2, y_2) 处也是有用的。因此，在深度方向上每个深度切片的神经元使用相同的权重和偏置，一组权重参数对应深度方向的一个深度切片。因此，设卷积层的输入数据体深度为 D_1，则卷积层共有 $F \cdot F \cdot D_1 \cdot K$ 个权重和 K 个偏置。例如，本算法的一个卷积层在深度方向上共有 24 个深度切片，共有 $129 \times 225 \times 24 = 696600$ 个神经元，但只有 $5 \times 5 \times 3 \times 24 = 1800$ 个权重参数（+24 个偏置参数）。因此，卷积神经层通过权值共享的方式大大减少了参数的数目。

卷积神经网络中常用的层还包括汇聚层、归一化层、局部连接层等，在本算法中未使用这些层。

3.3.4 激活函数

本算法的激活函数使用 ELU（Exponential Linear Units）函数。由于在回归训练中回传的梯度可能过大，容易导致 RELU 神经元死亡。而 ELU 函数具有左侧软饱和性，因此可以在一定程度上避免死亡，加速收敛。

激活函数是对输出值的非线性映射，Bengio 等[66] 给出了激活函数定义是：激活函数是一个映射 $h : R \to R$，且几乎处处可导。假设一个网络仅包含线性卷积核，那么再深的网络也只能表达线性映射，难以建模非线性分布的数据。激活函数为神经网络提供了非线性建模的能力，是深度网络不可或缺的部分。

从定义来看，几乎所有的连续可导函数都可以作为激活函数。深度网络出现之前，最常用的激活函数是 sigmoid 函数，函数图像如图 3-4 a) 所示。其表达式如式（3-3）所示，其中用 $\sigma(x)$ 代表 sigmoid 的函数值。

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3-3}$$

可以推知，sigmoid 函数的导数公式如式（3-4）所示。

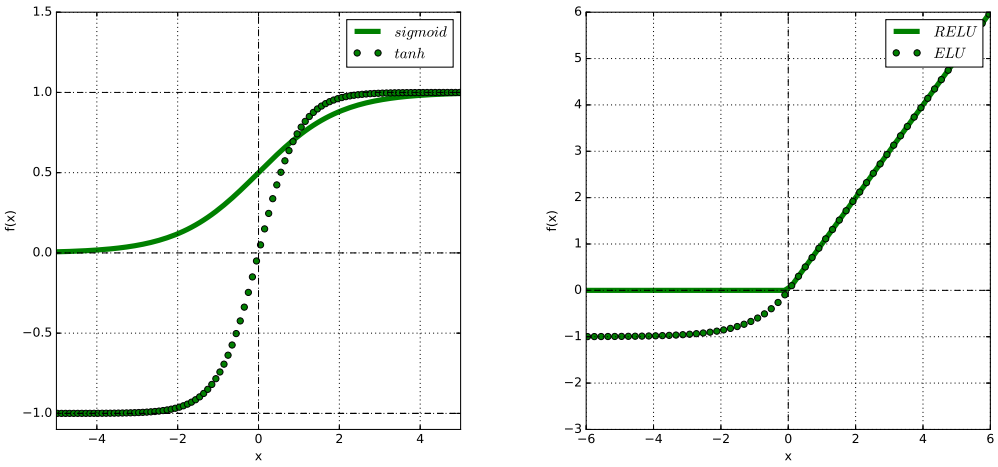
$$\sigma'(x) = \sigma(x)(1-\sigma(x)) \tag{3-4}$$

当 $\sigma(x)$ 的值接近于 0 或 1 时会使梯度 $\sigma'(x)$ 接近于 0，即具有软饱和性。在误差反向传播时，激活函数的局部梯度将会与整个损失函数关于该门单元输出的梯度相乘。因此，接近于 0 的局部梯度会使得回传的梯度信号接近与 0，即“杀死”了梯度。另外，sigmoid 函数的输出值大于 0，即神经元得到的数据不是零中心的，这一情况也会影响训练过程。因此在深层网络中，使用 sigmoid 函数不太合适。

与 sigmoid 函数的类似的函数还有 tanh 函数，如图 3-4 a) 所示，其函数的表达式如式（3-5）所示。

$$tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{3-5}$$

可见 $tanh(x) = 2\sigma(2x) - 1$，$tanh$ 函数也具有软饱和性，在深层网络中同样会导致梯度消失。



a) $sigmoid$ 函数和 $tanh$ 函数　　b) $RELU$ 函数和 ELU 函数

图 3-4 神经网络中的激活函数

RELU 激活函数[67] 于 2010 年被提出，函数图像如图 3-4 b) 所示。其表达式如式（3-6）所示。

$$RELU(x)=\begin{cases}x & if\ x>0\\ 0 & if\ x\le 0\end{cases} \tag{3-6}$$

RELU 激活函数具有非饱和性，有效的避免了梯度消失现象，对网络收敛有很强的加速作用。有学者在 ImageNet 分类任务中对 RELU 函数和 sigmoid 函数进行了对比，发现使用 RELU 激活函数的神经网络收敛速度是后者的 6 倍左右。同时，RELU 函数避免了在 sigmoid 和 tanh 函数中出现的指数运算，指数运算在网络前向传播和反向传播时非常消耗计算资源。但是，RELU 函数也有一定的缺点，主要的缺点被称为神经元“死亡”现象。当一个很大的梯度流过 RELU 神经元时，可能使 x 更新到横轴的负半轴区域，此时神经元的激活值和梯度都变为零，这种情况下神经元将永远无法被再次激活。通过降低学习率可以在一定程度上避免这种现象。

由于算法用于转向角度预测问题，使用平方误差损失函数时很有可能得到较大的梯度。经测试，如果使用 RELU 神经元会导致大约 40% 的神经元死亡。因此本算法中使用 ELU（Exponential Linear Units）[68] 激活函数。ELU 激活函数融合了 RELU 函数和 sigmoid 函数的优势，具有左侧软饱和性，如图 3-4 b) 所示。其表达式如式（3-7）所示。

$$ELU(x)=\begin{cases}x & if\ x\ge 0\\ \alpha(exp(x)-1) & if\ x<0\end{cases} \tag{3-7}$$

其中 α 是一个常数，可通过交叉验证选取，算法中设为 0.1。ELU 函数右侧的线性部分可以缓解梯度消失，左侧软饱和性可以减少神经元的“死亡”现象，使收敛速度更快，适合于本问题。

3.3.5 优化方法

本算法中采用小批量梯度下降法（mini-batch gradient descent）和动量（Momentum）更新相结合的优化方法。其中批量大小设置为 64，使用 Nesterov 动量，动量系数为 0.9。学习率设为 0.01，并逐步衰减。

神经网络的迭代求解一般使用梯度下降算法。根据每次迭代使用样本数，梯度下降算法可分为批量梯度下降、小批量梯度下降和随机梯度下降。批量梯度下降每次需要计算所有样本梯度的平均值，然后更新权重，这种方法对于大规模数据来说资源消耗大，只适用于小规模问题。小批量梯度下降每次选择部分样本作为一个批量，计算这个批量中所有样本梯度的平均值更新权重，在效率上更具优势。随机梯度下降是一种特殊版本，每次只使用一个样本进行更新。

批量大小的设置与实际样本个数有关，所选择的批量大小一般应使全部样本在 10-20 轮内参与迭代一次。同时，由于多数框架使用高效的矩阵方法实现，因此如果将批量大小设置成 2 的指数级，将会对计算有加速作用。另外，批量大小的设置还要考虑实际使用的 GPU 的显存大小，批量越大，每次迭代需要装入 GPU 的数据越多，对显存的要求也更高。

动量（Momentum）更新对深度网络的训练有很好的加速作用。在优化中，可以将损失值理解成山的高度，高度势能为 $U = mgh$，因此 $U \propto h$。参数初始化后等同于在某个位置给质点设置初始速度为 0，因此最优化过程可以看做质点在地形上滚动的过程，质点所受的力是损失函数的负梯度（$F = -\nabla U$）。此外，由于 $F = ma$，所以梯度与质点的加速度成正比。梯度先影响速度，再由速度影响位置。动量更新公式如式 (3-8) 所示。

$$\begin{cases} v = \mu \cdot v - lr \cdot dx \\ x+ = v \end{cases} \tag{3-8}$$

其中，lr 是学习率，dx 为当前的梯度值，μ 控制动量的大小，是一个超参数。动量可以抑制速度，降低系统动能，使质点能够在山底停下来。μ 的值可以通过交叉验证选择，一般设置为 $\{0.5, 0.9, 0.95, 0.99\}$ 中的一个，也可以开始时将 μ 值设置为 0.5，随着学习过程慢慢提升至 0.99。

Nesterov 动量在普通动量的基础上进行了优化。思路是，当参数位于位置 x 时，梯度 dx 不在位置 x 处计算，而先计算 $x + \mu \cdot v$ 作为未来的近似位置，然后在该位置处计算梯度。这样在计算梯度时“向前看”一步，使算法收敛到更好的位置。Nesterov 动量和普通动量的比较如图 3-5 所示。

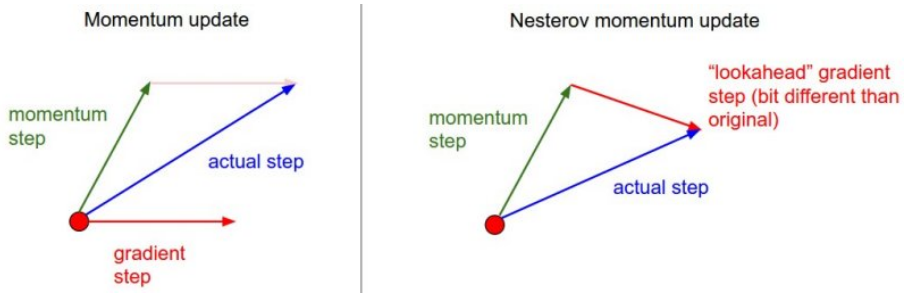


图 3-5 普通动量和 Nesterov 动量的更新示意图

3.3.6 参数初始化

算法中采用的参数初始化方法是 He 等[69] 于 2015 年提出的深度网络初始化方法。参数初始化方式对模型能否收敛有很大影响，近年来相关方法有许多研究和改进，但目前还没有形成统一标准。

神经网络的权重初始化应该打破对称性。如果将网络中所有权重初始化为相同的值，则每个神经元的激活值相同。在本算法中，由于输出节点只有一个，因此如果隐含层中神经元的激活值相等，则回传的梯度也相等，每个参数的更新量也相等，网络将无法学习。一种简单的打破对称性的方法是，将权重初始化为很小的数值，一般使用 $\mu = 0$，σ 较小的高斯分布生成随机数进行初始化，这种方法能够使网络开始学习。但在深层网络中，这种初始化方式会出现一定的问题，因为权重初始化为较小的值，而梯度与权重是成比例的，因此会导致反向传播时梯度很小，阻碍深层网络的学习。

在输入数据量较大时，输入数据的方差也在增长。一般来说当神经元输入数据的方差与输出数据的方差一致时，收敛速度最快，因此需要通过初始化来对方差进行校准。假设权重 w 与输入 x 之间的内积为 $s = \sum_i^n w_i x_i$，其中 n 为输入神经元的个数，内积表示在通过激活函数之前的值，则 s 的方差推导如式（3-9）所示。

$$\begin{aligned} Var(s) &= Var(\sum_i^n w_i x_i) \\ &= \sum_i^n Var(w_i x_i) \\ &= \sum_i^n [E(w_i)]^2 Var(x_i) + E[(x_i)]^2 Var(w_i) + Var(w_i)Var(x_i) \\ &= \sum_i^n Var(x_i)Var(w_i) \\ &= (nVar(w))Var(x) \end{aligned} \tag{3-9}$$

在第三步中假设输入数据 x 和权重 w 都服从均值为 0 的正态分布，即 $E[x_i] = E[w_i] = 0$。根据结果，如果想要 $Var(s) = Var(x)$，即输入和输出的方差相同，则必须保证权重 $Var(w) = 1/n$。在实现中，可以先使用标准正态分布初始化权重，随后除以标准差 $\sqrt{n}$。

Glorot 等[70] 对输入和输出的方差进行了类似的分析，推荐的初始化公式为 $Var(w) = 2/(n_{in} + n_{out})$，其中 n_{in} 代表权值连接的前一层神经元个数，n_{out} 代表权值连接的后一层神经元个数。He 等[69] 针对 $RELU$ 神经元的特点单独进行了分

析，推荐的初始化公式为 $Var(w) = 2/n_{in}$。

在本算法中，使用的神经元 ELU 是 RELU 神经元的改进，因此也需要对权重的方差进行校准。经过交叉验证，最终选择 $Var(w) = 2/n_{in}$ 的权重初始化方式。

3.4 提升泛化能力和训练加速

本文设计的卷积神经网络结构要完成一个回归学习问题，相比于分类问题来说，回归问题更加难以学习。由于损失函数使用平方误差函数，回传的梯度可能过大或过小，因此易导致神经元死亡，神经元死亡会导致网络收敛速度过慢或产生过拟合。本节提出了防止过拟合和网络预训练的方法来解决上述问题。

3.4.1 防止过拟合

深度模型需要从数据集中提取特征来完成任务。数据特征分为两类：全局特征和局部特征。在自动驾驶中，全局特征对应于所有自动驾驶场景图片都具备的特征，如障碍物、标志线等。局部特征对应于仅在实验采集的训练数据中的特征，不具有通用性。例如采集到的样本中转弯处均放置了一个箱子，模型可能利用箱子这个局部特征进行转弯，而不是利用标志线这个全局特征。在测试中，如果转弯处没有放置箱子，无人车就不会转向，这表明模型只学习到了局部特征。在模型学习过程中，数据的全局特征和局部特征都会被学习。学习的全局特征比例越高，模型的泛化能力越强。相反，学习的局部特征的比例越高，模型就越趋于过拟合。

模型过拟合的原因通常有三种：（1）训练集和测试集的分布不一致。（2）训练样本中存在数据不一致，即存在很多噪声数据。（3）模型容量太大但样本数相对较少。在一个精心构造的学习系统中，前两点原因通常可以避免。在深度学习模型中，由于模型的容量较大，即表征能力太强，通常会由于原因（3）发生过拟合。在微缩智能车实验环境中，由于人工采集的图片有限，因此容易引发过拟合。本算法中提出了三种方法来防止模型过拟合。

3.4.1.1 训练图像增广

训练图像增广可以变相增加样本数量，消除样本噪声，从而防止过拟合。首先对图像按 RGB 通道进行均值减法，即减去训练数据中每个颜色通道的平均亮度，使每个颜色通道的均值归零。随后对训练图像进行变换，在不影响图像语义的前提下扩大样本数量，图像变换中主要采取的措施如下：

（1）在水平和竖直方向随机移动最大比例为 10% 的像素。

（2）对图像进行 0.9-1.1 倍的随机放大或缩小。

（3）对图像的颜色通道进行小范围随机抖动，模拟环境亮度变化。

将以上三种图像增广方法进行任意组合，理论上可以获的无限多的训练样本，从而提高模型的泛化能力。

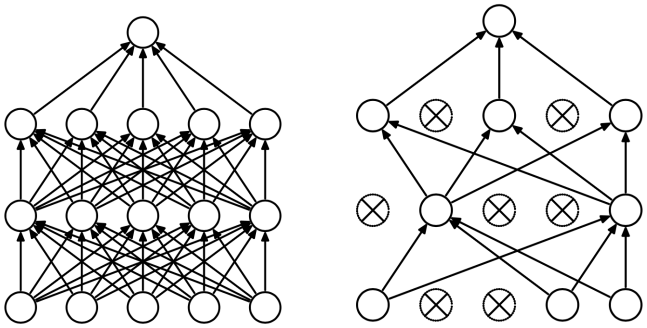
不能对图像使用镜面翻转、随机剪裁等方式进行增广。镜面反射可能使图像中的道路、标志线等改变方向，图像对应的理想转向角会发生变化，即图像的语义特征被改变。随机剪裁容易将图像中的障碍物剪裁掉，造成障碍物不完整，也会对图像语义产生影响。

3.4.1.2 正则化和随机失活

正则化和随机失活（DropOut）可以对模型容量进行限制。本算法在第一个全连接层中对权值 W 添加了正则项因子 $\lambda = 0.01$ 的 $L2$ 正则，同时在全连接层之间添加了概率 $p = 0.5$ 的 DropOut[71]。这些操作可以对权重的范围进行限制，减少过拟合的发生。

正则化方法主要分为 $L2$ 正则和 $L1$ 正则。$L2$ 正则通过惩罚神经网络中参数的平方项实现。对每一个权值 w，在损失函数增加一项 $\frac{1}{2}\lambda w^2$，其中 λ 为正则项因子，代表正则强度。在反向传播中，除了平方误差损失函数产生的梯度外，还要加上值为 $-\lambda w$ 的梯度。$L2$ 正则项可以减少网络中权重 w 之间的差距，使决策可以均匀的使用所有特征，而非小部分特征。另外还有一种 $L1$ 正则，它通过在损失函数中增加值为 $\lambda|w|$ 的正则项对权值进行约束，$L1$ 正则可以使权重在优化过程中变得稀疏，有利于消除噪声，有特征选择的作用。但本算法中并不关注特征选择，因此选择 $L2$ 正则。λ 越大，模型的偏差（bias）越大，方差（variance）越小；λ 越小，模型的偏差越小，方差越大，模型趋于过拟合。λ 的值通过交叉验证进行选择。

随机失活（DropOut）[71] 是一种高效的模型集成方法。在训练过程中相当于从完整的神经网络中抽样出一些子集独立的进行训练，在测试过程中对这些独立训练的多个模型的预测结果取平均值。传统的模型集成方法需要训练多个模型，非常耗时。而 DropOut 通过训练一个模型达到了集成多个模型的效果，在计算上具有优势。DropOut 在训练中以概率 p 使每个神经元失活，失活的神经元激活值设为 0。未失活的神经元正常参与网络的前向传播和反向传播，失活的神经元不参与本次迭代。由于 DropOut 中每个特征都有可能失活，因此可以显著减少神经网络对某些特征的依赖，提高网络的鲁棒性和泛化能力。DropOut 中神经元连接方式如图 3-6 所示，可以看出，失活的神经元相当于断开了与网络中其他神经元之间的连接。DropOut 的概率 p 一般设为 0.5，也可以通过交叉验证选取。



a) 普通神经元连接示意图　b) 添加随机失活的神经元连接示意图

图 3-6 DropOut 连接示意图[71]

3.4.1.3 转向角预处理

本文中采集的自动驾驶训练集具有一定的特殊性，需要对转向角进行一定的预处理。训练数据包括图片和标签，其中标签为转向角，记为 r。在交通场景中，无人车直行时转向角 $r=0°$，在道路转弯处、超车时、规避行人和其他车辆时会采取转向动作，此时 $r\neq 0°$。根据驾驶经验来看，大部分时间车辆是按照直线行驶的，即 $r=0°$，而 $r\neq 0°$ 的情况只占少数。本文在微缩智能车环境中采集到的样本情况也是如此，多数样本的标签 $r=0°$，$r\neq 0°$ 的样本只占少数。

理想情况下，转向角 r 在训练集中应当均匀分布于值域范围内，这样有利于神经网络的学习。但实际上由于 $r=0°$ 的样本较多，导致了分布不均匀。关于如何在样本分布不均匀的数据中进行学习是一个非常活跃的研究领域，称为“代价敏感学习”[72]。由于本算法中仅 $r=0°$ 的样本较多，并非全部样本都分布不均匀，因此使用较为简单的转向角预处理方法来解决该问题。

具体方法为，使用均值 $\mu=0$，方差 $\sigma=0.5$ 的正态分布对 $r=0°$ 的样本标签进行震荡，震荡后的转向角分布如图 3-7 所示。可以看出，$r=0°$ 的角度在震荡后范围扩展至 $[-1°,1°]$，这种小范围的数据震荡不会改变训练数据的一致性。经实验验证，这种方法可以在一定程度上避免过拟合，提升训练效果。

此外，由于 ELU 激活函数的输出在 $(-1,+\infty)$ 范围内，而样本中 $r\in[-45°,45°]$，因此将转向角按比例缩放到 $[0,2]$ 的范围内。

3.4.2 网络预训练

许多深度学习模型，如深度信念网络（deep belief networks）、栈式自编码器（stacks of auto-encoder）等都得益于无监督学习对网络的预训练，Bengio 等[73] 对预

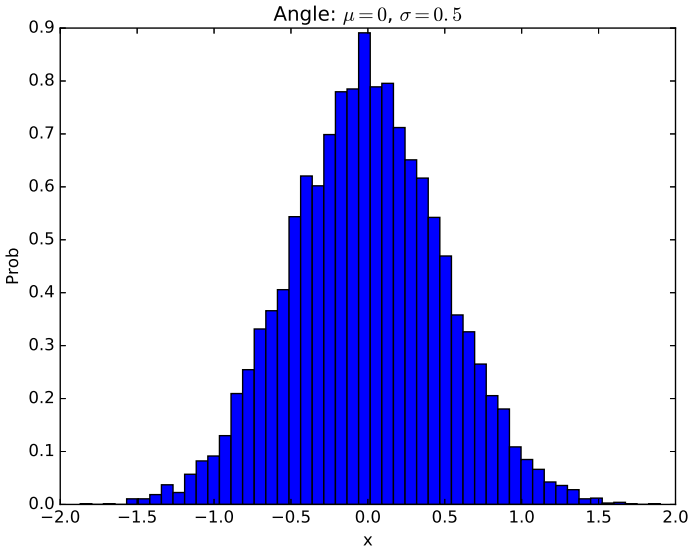


图 3-7 对转向角 $r = 0$ 的样本进行震荡

训练在深度学习中的作用进行了分析，指出通过预训练，可以对网络的参数进行规约，使参数到达一个合理的分布，增强网络的泛化能力，加速网络的收敛速度。受此启发，本算法使用了预训练方法来加速训练。训练步骤分为两步。

（1）从训练样本中人工挑选 500 张图片，要求这些图片语义清晰，即可以从图片中清晰的推断出转向角度的近似值。例如，可以选择光照充足条件下的转弯处，或有明显的障碍物。人工挑选的图片被用来进行网络的预训练，由于图片数量较少而模型容量较大，经过迭代后，网络的损失值应该接近于 0，即网络对这 500 张图片达到“过拟合”。此时，将网络中的权重保存下来，作为后续训练的权重初始值。

（2）使用预训练的权重初始化网络，在全部训练集上对模型进行训练，调整网络权值。

经过测试，使用语义清晰的部分图片进行网络预训练可以明显加快网络收敛速度。

3.5 本章小结

本章首先介绍了基于端到端控制的卷积神经网络的设计和实现方法，从理论角度解释了算法设计的各要素，包括损失函数、网络结构、卷积层、激活函数、优化方法、参数初始化方法等。随后，针对网络训练速度较慢的问题提出了网络预训练的方法，针对过拟合的问题提出了训练数据增广、正则化和随机失活、转向角预处理等解决方法，这些方法均可以在一定程度上提升训练效果。

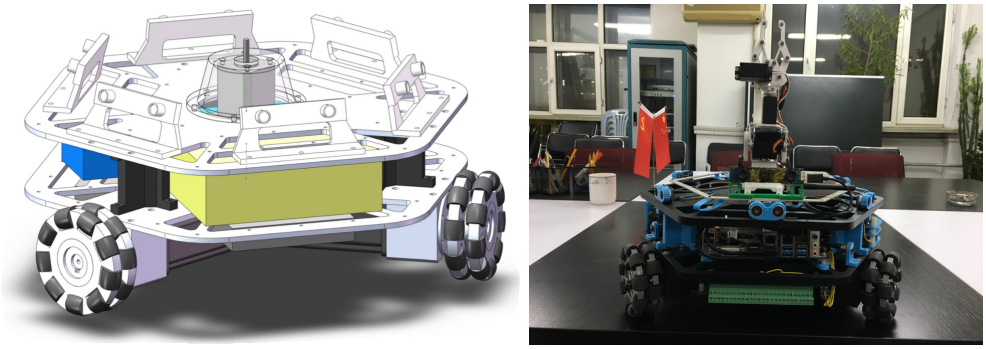
第 4 章 基于微缩智能车的自动驾驶实验和分析

4.1 引言

微缩智能车具有高度的灵活性、可控性，可以减轻真车实验带来的风险，在实验中被广泛使用。因此，课题组设计实现了一个微缩智能车用于图像采集和算法验证。本节将首先介绍微缩智能车平台的设计方法，随后介绍使用微缩智能车进行数据采集和算法测试的实验细节，并对实验结果进行评价和分析。

4.2 微缩智能车平台设计

设计实现的微缩智能车是一个小型、便携、具有独立运算能力的三轮全向智能车，可用于数据采集和算法测试。智能车的设计图和实景图分别如图 4-1 a) 和 4-1 b) 所示。



a) 智能车设计图　　b) 智能车实景图

图 4-1 智能车示意图

智能车具备全向运动能力，其底盘由三个全向轮组成，径向对称安装，各轮互成 120° 角。三个全向轮的大小和质量完全相同，由性能相同的电机驱动。智能车全向运动系统的坐标系如图 4-2 所示。其中，xoy 是世界坐标系，XOY 是智能车坐标系，θ 是智能车坐标系和世界坐标系之间的夹角，ϕ 为驱动轮之间的夹角，这里 $\phi = 120°$，L 为智能车中心到全向轮中心的水平距离。

控制智能车运动时，给智能车发出的指令是世界坐标系中的速度和角度，智能车需要将其转换为三个全向轮的线速度。设 v_1，v_2，v_3 分别是三个全向轮的线速度，v_x，v_y 分别是智能车在 XOY 坐标系下的 X 轴和 Y 轴的速度分量，ω 为智能

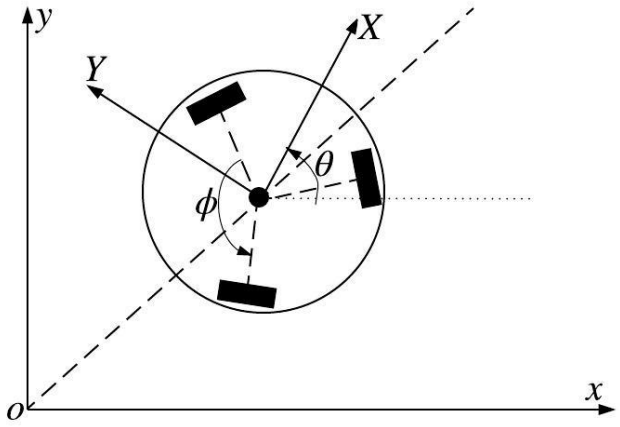


图 4-2 三轮全向智能车运动模型示意图

车自转角速度。根据坐标转换关系可推得，$(v_1, v_2, v_3)^{\mathrm{T}}$ 与 $(v_x, v_y, \omega)^{\mathrm{T}}$ 之间的关系如式（4-1）所示。

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} sin(\phi/2) & cos(\phi/2) & L \\ -sin(\phi/2) & cos(\phi/2) & L \\ 0 & -1 & L \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \tag{4-1}$$

以三轮全向智能车的中心 O 为参考点，取广义坐标系 $q = (\dot{x}, \dot{y}, \dot{\theta})$，其中 $(\dot{x}, \dot{y})$ 是智能车中心 O 在世界坐标系 xoy 中的坐标，$\dot{\theta}$ 是智能车坐标和世界坐标系之间的夹角。根据坐标系的建立情况，世界坐标系与智能车坐标系之间的关系如式（4-2）所示。

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \tag{4-2}$$

综合式（4-1）和式（4-2），可推知智能车在世界坐标系中的速度与驱动轮速度之间的关系，如式（4-3）所示。

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} sin(\phi/2-\theta) & cos(\phi/2-\theta) & L \\ -sin(\phi/2+\theta) & cos(\phi/2+\theta) & L \\ sin\theta & -cos\theta & L \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{4-3}$$

因此，给智能车发出速度和角度指令，根据式（4-3），智能车可以将其转换为三个全向轮的线速度，随后驱动电机运动。智能车运动控制界面如图 4-3 所示，只需拖拽标记或输入 x 方向和 y 方向的运动距离和角度，就可以驱动智能车进行运动。

由智能车实景图可以看到，智能车上安装了相机和超声传感器。相机安装在

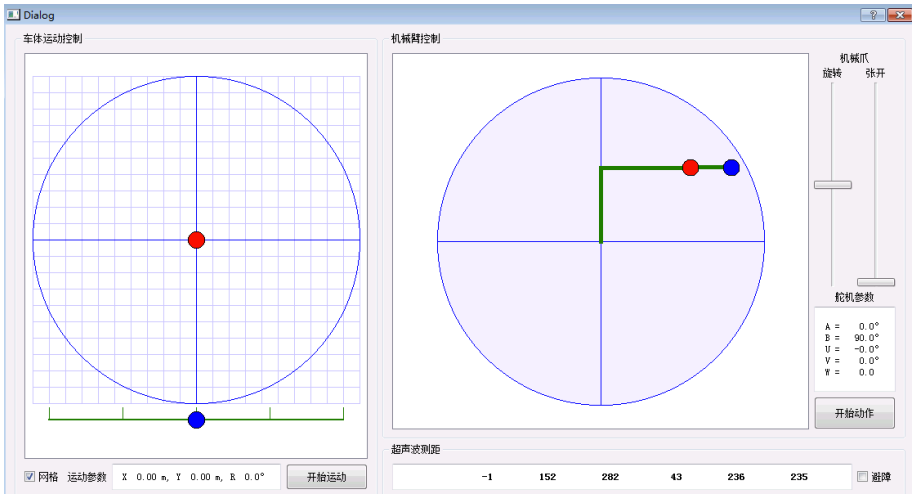


图 4-3 智能车运动控制程序界面示意图

智能车前部，可采集以智能车为第一视角的图像，在训练和测试过程中作为卷积神经网络的输入。超声传感器用于测量与环境障碍物的距离，但距离信息并不作为算法的输入，只用于紧急避障。智能车在不同方向共配置有 6 个超声传感器，观测范围可以 360° 覆盖周围环境。在控制界面的右下角可以看到 6 个超声传感器测量距离的实时变化情况。当超声传感器检测到的障碍物最小距离小于安全距离时，智能车会采取紧急规避动作，随后需由人工进行恢复。

智能车搭载了自行设计的数据处理平台，因此智能车具有一定的数据处理能力，可与远程主机进行协作。采集样本时，智能车的数据处理平台控制相机拍照，将图片保存在处理平台上，并同步到远程主机；样本采集结束后，远程主机进行卷积神经网络的训练，远程主机配备有高性能的 NVIDIA GTX-1070 显卡，可依靠显卡加速功能完成训练，训练结束后远程主机将网络参数同步至智能车的数据处理平台；测试中，数据处理平台运行训练好的卷积神经网络，根据相机采集的环境图像，卷积神经网络可实时预测转向角度，驱动智能车运动。智能车和远程主机之间通过网络进行通信。智能车的主要硬件配置如表 4-1所示。

表 4-1 智能车主要硬件配置表

硬件名词	性能指标
全向轮	三个，轮径 101.6mm
步进电机	1.8° 步进角，16 细分，移动精度 0.1mm
主板	华擎 Z170M-1TX/ac
固态硬盘	128G SSD
锂电池	36V 8Ah
电源板	TP240 DC-ATX，峰值功率 240W

远程主机主要完成 CNN 的训练工作。训练由 GPU 和 CPU 协作完成，其中 CPU 进行数据预处理等准备工作，GPU 进行神经网络的前向传播和反向传播计算。选择 GPU 时要考虑核心数和存储空间等因素。GPU 核心数决定了网络的训练时间，由于本算法在线下训练，所以对训练速度要求不高。GPU 存储空间限制了网络规模和训练批量的大小，神经网络每层的激活值和误差都要存储在 GPU 中，需要大量的存储空间。减少训练中数据批量的大小，或改用步长较大的卷积核都可以减少 CNN 所需的存储空间，但可能会带来一定的精度损失，因此在实际应用中需要进行权衡。本文使用的远程主机的主要硬件配置如表 4-2 所示。

表 4-2 远程主机的主要硬件配置表

硬件名词	性能指标
主板	x58
CPU	Xeon E5506@2.13GHz 四核
GPU	Nvidia GTX-1070 8GB 显存

4.3 实验设计与分析

4.3.1 训练数据采集

每条训练数据应该包括以智能车为第一视角的图片和该图片对应的合适的转向角度。在数据采集中，合适的转向角度由人工选择，选择转向角度后智能车按照该角度前进一段距离，随后再次拍摄图片并再次由人工选择转向合适的转向角。如此循环直至样本数量达到预期，则训练样本采集完成。因此，算法为监督学习算法，其中监督者即为样本采集者。为了保证训练样本的多样性和代表性，样本采集需要在不同的地图、地面环境、光照条件下进行。本文的样本采集工作在哈尔滨工业大学新技术楼 907 和 401 进行，采集中用人工设置的标志线和障碍物来模拟交通环境，共采集样本 8120 幅。

采集时由采集者使用遥操作的方式控制智能车。采集者对当前智能车前方的环境进行观察，综合考虑障碍物距离、标志线距离、前进方向和自身位置等因素，选择一个合适的转向角度作为样本标签。数据处理平台会记录当前拍摄的图像和采集者选择的转向角度，二者构成一条训练样本。数据采集时使用的控制界面如图 4-4 所示，控制界面中可以实时显示相机拍摄的图像，同时提供了滑块供采集者选择合适的转向角。

图 4-4 智能车数据采集界面示意图

采集者在采集样本中应遵循一定的策略，该策略应使智能车能够平稳行驶和避障。同时，在整个采集过程中策略应具有一致性。策略的一致性可以保证训练数据的一致性，从而保证学习过程的合理性，防止出现噪声数据引发过拟合。实验中采集样本的基本策略如下：

（1）当前方中没有障碍物或智能车已经越过障碍物时，智能车应保持在道路中央，沿标志线行驶。

（2）当智能车视场内出现障碍物时应开始调整转向，随后采取一系列转向动作，以尽可能平滑的运动轨迹绕开障碍物，避免急转。其原因是，相机只能观测到位于智能车前方的障碍物，当位于智能车前部的相机越过障碍物后，智能车侧面的障碍物就无法被观测到。因此，如果智能车在靠近障碍物处急转，那么即使智能车前部可以避开障碍物，智能车后部也很容易触碰障碍物，因此智能车应该提前计划一条平滑的运动轨迹进行避障。

（3）样本采集时需变换不同的地图类型。在实验中，布置了 S 形、八边形、五边形等多种地图类型，以保证样本的多样性。实验过程中的部分地图形状和实景图如图 4-5 所示。

（4）样本采集时需变换不同的光照条件，实验中利用自然光的明暗变化、灯光的颜色和亮度变化来模拟不同的光照条件。

（5）样本采集时需变换不同的障碍物类型，包括纸杯、乒乓球、网球、各种玩具等，用于模拟交通环境下的障碍物。

（6）采集极端情况下的避障行为，例如智能车距离障碍物较近时进行急转，在没有标志线的道路上进行避障等，模拟非结构化交通环境中的行驶，使算法更具鲁棒性。

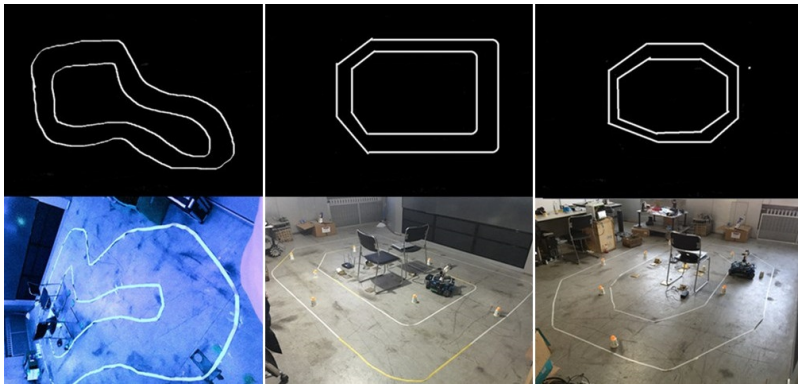


图 4-5 智能车数据采集时的部分场景地形图和实景图

4.3.2 训练结果与分析

4.3.2.1 训练结果

样本采集结束后，可以对设计的卷积神经网络进行训练，样本经过数据增广和转向角预处理等步骤后作为网络的输入。全部训练样本分割为训练集、验证集和测试集，其中训练集包含 6000 个样本，验证集包含 1000 个样本，测试集包含 1120 个样本。

验证集用于在网络训练中进行超参数选择，包括卷积核大小、正则项系数、DropOut 率等。同时，在训练过程中验证集可以防止模型过拟合。具体方法是，在每一轮训练结束后都计算验证集的数据损失，当验证集的损失不再下降时停止训练，称为提早停止（early stopping）。这是因为当训练集损失下降而验证集损失不再下降时，模型会逐渐趋于过拟合，此时应当停止训练。

卷积神经网络结构基于 Tensorflow[74] 实现。Tensorflow 是一个高效的深度学习框架，采用数据流图描述计算过程，节点表示数学操作，线表示在节点间相互联系的多维数据数组，即张量（tensor）。Tensorflow 在神经网络反向传播中可以自动计算自动计算微分，提供了 Python/C++ 等程序接口，可以方便的部署在多 CPU 和多 GPU 的分布式环境中。

训练过程共耗时 7 个小时。在迭代 200 轮后，验证集误差不再下降，此时停止训练，使用测试集对训练好的神经网络进行测试，最终误差为 0.012。由于样本标签（即转向角）经过了缩放处理，因此需将误差换算成缩放前的值，换算后误差为 4.9°，即神经网络预测的转向角与人工选择的转向角之间的平均差值为 4.9°。训练过程中训练集和验证集的损失变化如图 4-6 所示。

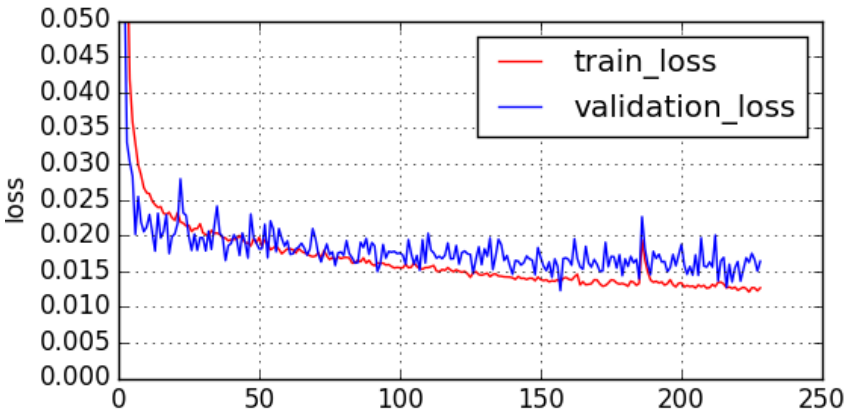


图 4-6 训练集和验证集的损失变化图

4.3.2.2 对比分析

本文在第三章提出的防止过拟合的三种方法包括训练数据增广、正则化和随机失活、转向角预处理，加速训练的方法为网络预训练。通过单独去除其中的某种方法，可以观察该方法对训练结果和训练速度的影响。表 4-3 对比了单独去除每种方法后，训练中的验证集损失的变化情况，“—”是去除的意思。考虑到验证集的损失存在抖动，表中以 40 轮为一个单位，跟踪每 40 轮的平均损失并换算为角度。“stop”表示验证集损失已停止下降，训练停止。

表 4-3 防止过拟合和网络预训练方法在验证集上的损失变化对比

方法描述	1-40	41-80	81-120	121-160	161-200	201-240	241-280
—训练数据增广	6.64	6.21	5.90	5.78	**5.59**	stop	stop
—正则化与损失失活	6.34	5.93	5.62	5.51	**5.33**	stop	stop
—转向角处理	6.16	5.76	5.46	5.36	**5.18**	stop	stop
—网络预训练	6.10	5.71	5.40	5.30	5.13	5.11	**5.10**
全部使用	6.04	5.65	5.35	5.25	**5.08**	stop	stop

从表 4-3 中可以得出以下结论：

（1）防止过拟合和加速训练的每种方法都能在一定程度上提升训练速度，改善训练效果。

（2）按照对最终损失的影响排序，四种方法由大到小分别为训练数据增广、正则化与随机失活、转向角预处理、网络预训练，对最终损失的影响比例分别约为 10%，5%，2%，0.5%。可见，在样本有限的情况下，使用训练数据增广的方法可以有效防止过拟合。正则化与随机失活可以对样本容量进行限制，在一定程度提高模型泛化能力。转向角预处理也能一定程度上解决样本分布不均衡的问题。

（3）网络预训练虽然对最终损失只有微弱的影响，但对训练速度的提高有很大帮助。不使用网络预训练的至少需要多迭代 80 轮左右才能达到相同的效果。原因在于，图像分类可以使用大规模数据集（如 ImageNet）上训练好的网络权重初始化，而回归问题则没有合适的预训练权重。使用语义清晰的小规模数据对网络进行预训练可以在一定程度上解决这一问题。

4.3.3 CNN 可视化分析

训练结束后，可以通过卷积层的可视化分析较为直观的看出 CNN 学习到了哪些特征。图 4-7 是两种不同输入下 CNN 第一个卷积层中 24 个激活图的可视化效果。其中左图的输入图像有障碍物和标志线，右图则没有。

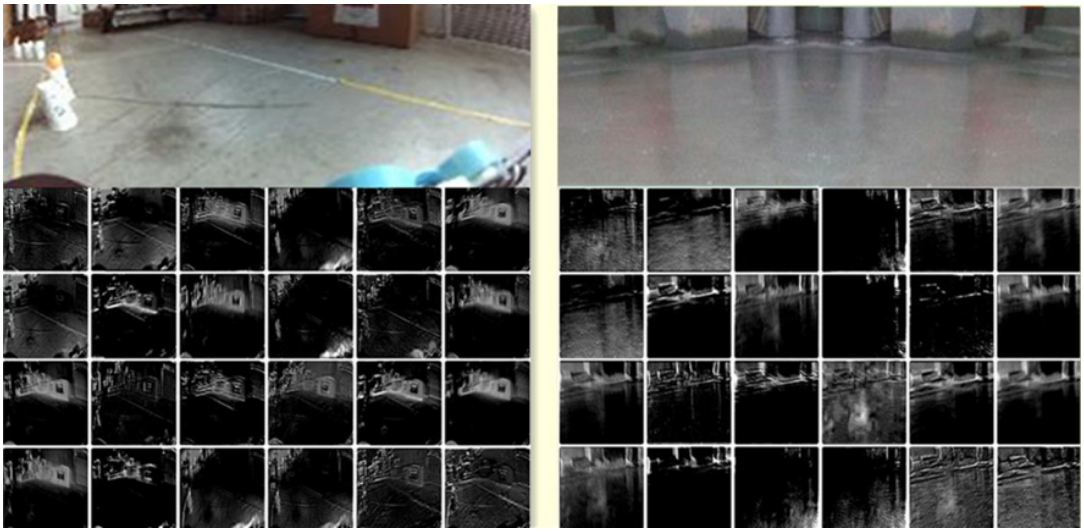


图 4-7 两种输入图片下卷积层可视化示意图

可以看出，左图中的箱子、乒乓球、纸杯、标志线、地面等与自动驾驶紧密相关的物体在不同的激活图中有较大的激活值，所有激活图都有较强的纹理，右图则不具备这些特征。激活图是 CNN 从图片中提取的特征，这些特征是 CNN 为了优化人工选择的转向角和网络预测的转向角之间的误差而自行从图像中提取的，从可视化效果来看，CNN 提取的这些特征确实与转向角的选择密切相关。因此，基于端到端控制的 CNN 学习系统并不需要人工设计障碍物、标志线等检测功能，却可以自行学习到这些对决策有价值的关键特征，从而证明了本文设计的结构在具有简明性的同时也具有合理性和可解释性。

4.4 实测结果与分析

4.4.1 实测结果

智能车在实际测试中的效果令人满意，在实验过程中拍摄的智能车避障视频见网址：http://pr-ai.hit.edu.cn/research/bai2017intelligent.html

实验中智能车的避障表现总结如下：

（1）智能车在没有障碍物的环境中能够跟随标志线行驶，且运动轨迹一直处于道路中央。即使初始状态下智能车处于道路的一侧，在运动过程中智能车也会渐渐转移到道路中央。

（2）智能车能够规划一条平滑的运动轨迹避障，同时能够适应变化的环境，即在智能车运动过程中，地图和障碍物位置可以发生变化。障碍物可以在智能车反应距离外随机出现（反应距离约为 $1m$）或随机改变自身位置，智能车能够根据情况对路线做出调整。

（3）在一些极端情况下智能车仍然表现良好。在没有标志线的场景中，智能车遇到障碍物仍然可以采取规避行为；初始时如果将智能车放置在标志线外侧，智能车能够找到标志线并逐步运动到内侧；可以更换训练样本中没有用到的其他类型障碍物，只要障碍物纹理清晰，智能车就可以进行规避。

定义自动驾驶率 R 来衡量智能车的避障水平。当智能车在行进过程中接触到标志线或障碍物时，记录其出现故障一次。故障需要人工进行恢复，在实测中发现人工处理一次故障平均需要 $6s$。故定义自动驾驶率如式（4-4）所示，其中 C 代表故障次数，T 代表测试总时间。

$$R = (1 - \frac{C * 6s}{T}) * 100\% \tag{4-4}$$

自动驾驶率会受地图中障碍物数量的影响，因此定义障碍物密度来衡量当前环境的困难程度。智能车对障碍物的反应距离约为 $1m$，故定义障碍物附近 $1m$ 的范围为障碍物区域。定义障碍物密度如式（4-5）所示，其中 ρ 代表障碍物密度，N 代表障碍物数量，L 代表路线总长度。

$$\rho = (\frac{N * 1m}{L}) * 100\% \tag{4-5}$$

实测中使用总长 20m 的地图，障碍物位置随机。智能车在不同的障碍物密度下测试避障性能，同时记录自动驾驶率。每种障碍物密度下均经过多次测试，每次测试中变换地图和障碍物位置。每种障碍物密度下的平均自动驾驶率统计如表 4-4 所示。

表 4-4 智能车实测中自动驾驶率随障碍物密度变化表

障碍物密度（ρ）	自动驾驶率（R）
20%	100%
30%	100%
40%	98%
50%	90%
60%	78%
70%	62%

从表 4-4 可以看出，当障碍物密度大于 60% 时，自动驾驶率才会有明显下降。而常规场景中障碍物密度不会高于 40%，因此，该智能车能够在常规场景中保持较高的自动驾驶率。

4.4.2 对比分析

从实际应用的角度看，本文工作与 DAVE[16] 智能车相比，主要提高在以下两个方面。

第一，降低了对系统实时性的要求。DAVE 智能车控制左转或右转，不指定角度。由于控制粒度较为粗糙，DAVE 必须具有很高的实时性，即两次采样预测之间的时间间隔应该在毫秒级，以不断适应变化的环境。本文设计的智能车系统使用粒度更细的转向角描述运动，可以对动作有提前的规划，从而降低了对系统实时性的要求。同时，由于本文设计的卷积神经网络层数较多，达到毫秒级的控制间隔需要消耗大量的计算资源，在实现上也是不现实的。

第二，减轻了场景的语义分歧。DAVE 智能车在结果分析中提到，其训练好的网络在测试集上的错误率是 35.8%，并指出错误率较高的原因是同一场景有语义分歧。即在某些场景下，左转或者右转都可以完成避障，但在图像标注和测试中只能选择左转或右转中的一个动作，因此造成歧义。本文认为，虽然测试集上的错误率不一定会影响智能车在实测中的表现，但是语义分歧会对卷积神经网络的训练造成不利影响。造成语义分歧的原因是，DAVE 并不限定训练和测试场景，训练数据中包含大量的非结构化场景，因此容易造成语义分歧。本文将智能车的应用限制在模拟交通环境中，利用标志线作为引导标记，智能车的目标是在沿标志线行驶的同时进行避障，在一定程度上消除了场景的语义分歧。

4.5 本章小结

本章首先介绍了微缩智能车的平台设计，随后介绍了实验的设计与分析，包括训练数据采集的过程以及训练的结果，并对训练结束后的 CNN 进行了可视化分析，最后对智能车的实测表现进行了评价。结果表明，智能车能够提前规划合理的路线进行避障，能够在常规场景中保持较高的自动驾驶率。同时，相对于 DAVE 智能车来说，降低了对系统实时性的要求，减轻了场景的语义分歧。

从微缩智能车实验中可以发现，与间接感知型结构和直接感知型结构相比，本文设计的算法具有明显的优势。首先，避免了间接映射型方法的复杂系统结构，降低了设计的难度。其次，可以在真实场景下高效的完成数据采集和训练。而直接感知型方法中需要学习驾驶相关的指标，例如与障碍物的距离、与标志线距离等，在真实场景中精确采集这些数据需要超声和激光雷达等设备，采集成本高，不易实现。本文算法只需要记录视场图像和转向角作为卷积神经网络的训练样本，在测试时只需采集视场图像，根据卷积神经网络预测的转向角实现对智能车的连续控制。

结　论

基于计算机视觉的自动驾驶技术主要分为间接感知型结构、直接感知型结构和端到端控制结构。本文改进了传统的端到端控制结构，设计了一个基于深度学习的由图像到转向角的学习算法，并对算法的有效性进行了实验验证。本文主要工作和取得的结果包括：

（1）设计了基于计算机视觉和深度学习的端到端的自动驾驶系统。该系统将自动驾驶作为一个整体的问题进行研究，使用卷积设计网络（CNN）建立图像到动作的直接映射。CNN 由 7 层卷积层和 4 层全连接层组成，通过学习以无人车为第一视角的图像，预测转向角度。算法的设计要素主要包括损失函数、网络结构、卷积层、激活函数、优化方法和参数初始化，各要素的设计需要综合考虑样本类型、输出类型、样本数量、可视化结果等。本文从理论角度分析了算法中各要素的设计方法。

（2）提出了提升 CNN 泛化能力和加速训练的方法。由于 CNN 输出连续的转向角度，因此是一个回归学习问题，容易导致神经元死亡，导致网络收敛速度过慢或引发过拟合。针对过拟合问题，本文提出了训练图像增广、正则化和随机失活、转向角预处理等方法，有效的抑制了过拟合，提升了网络的泛化能力。针对网络收敛速度过慢的问题，本文提出了选择部分语义清晰的图片进行模型预训练的方法，通过预训练初始化网络权值，从而加速收敛。

（3）设计了一个微缩智能车系统验证算法的有效性。微缩智能车具备全向运动能力，自身搭载有数据处理平台，可以控制相机、超声等传感器进行数据采集，同时可以远程主机进行协作。微缩智能车在人工设计的模拟交通环境中完成训练数据的采集，随后由卷积神经网络进行训练，最终测试误差为 4.9°。实测结果表明，智能车能够提前规划合理的路线进行避障，能够在常规场景中保持较高的自动驾驶率。与 DAVE 智能车相比，降低了对系统实时性的要求，减轻了场景的语义分歧。在可视化分析中可以看出 CNN 为了优化最终目标而自行从图像中提取了障碍物和标志线等特征，证明了设计的合理性。

自动驾驶是一个庞大的系统，本文仅对其中的端到端控制方法进行了初步探索，今后的研究工作可以从下面三个方向进行：

（1）提高基于卷积神经网络的回归学习的效果。回归学习中可能会由于回传的梯度过大而导致神经元死亡或梯度消失。虽然在算法设计中加入了预训练步骤，

使用了改进的激活函数，但并不能完全避免该问题。如何在回归问题中进行更有效的训练是有待研究的问题。

（2）提高自动驾驶系统的稳定性。由于本文使用单帧图像作为训练数据，没有考虑帧与帧之间的关联，因此可能出现两帧的预测角度差值较大的情况。当差值较大时，无人车可能会出现颠簸等不稳定情况。因此，如何将两帧或多帧图像综合进行分析是有待研究的问题。

（3）对错误原因进行有效的分析和处理。基于端到端控制的方法虽然简洁，但当无人车发生故障时却很难对故障原因进行分析。间接感知型结构和直接感知型结构则不存在该问题。其中，间接感知型结构由多个模块组成，可以对每个模块的输出结果进行监控。直接感知型结构输出了与驾驶相关的关键指标，也可以对每个指标进行监控。因此，如何使用可视化技术对错误进行分析和处理是有待研究的问题。

参考文献

[1] Janai J, Guney F, Behl A, et al. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art[J], 2017.

[2] Thorpe C, Hebert M H, Kanade T, et al. Vision and navigation for the Carnegie-Mellon Navlab[C] // Transactions on Pattern Analysis and Machine Intelligence. 1988 : 362 – 373.

[3] 凌春霞, 刘劲华. 谷歌无人汽车: 人工智能快速占领交通高地 [J]. 汽车运用, 2016(10) : 30 – 31.

[4] 孙振平, 安向京, 贺汉根. CITAVT—IV——视觉导航的自主车 [J]. 机器人, 2002, 24(2) : 115 – 121.

[5] Bertozzi M, Broggi A, Fascioli A. Vision-based intelligent vehicles: State of the art and perspectives[J]. Robotics & Autonomous Systems, 2000, 32(1) : 1 – 16.

[6] Franke U, Gavrila D, Gorzig S, et al. Autonomous driving goes downtown[J]. IEEE Intelligent Systems & Their Applications, 1998, 13(6) : 40 – 48.

[7] Grisleri P, Fedriga I. The BRAiVE platform[C] // Procs.Ifac Symposium on Intelligent Autonomous Vehicles. 2010.

[8] Broggi A, Cerri P, Debattisti S, et al. PROUD—Public Road Urban Driverless-Car Test[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(6) : 3508 – 3519.

[9] Furgale P, Schwesinger U, Rufli M, et al. Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge Project[C] // Intelligent Vehicles Symposium. 2013 : 809 – 816.

[10] Bojarski M, Testa D D, Dworakowski D, et al. End to End Learning for Self-Driving Cars[J]. CoRR, 2016, abs/1604.07316.

[11] Ullman. Against direct perception[J]. Behavioral & Brain Sciences, 1980, 3(3) : 373 – 381.

[12] Zhang H, Geiger A, Urtasun R. Understanding High-Level Semantics by Modeling Traffic Patterns[C] // IEEE International Conference on Computer Vision. 2013 : 3056 – 3063.

[13] Urtasun R, Lenz P, Geiger A. Are we ready for autonomous driving? The KITTI vision benchmark suite[C] // IEEE Conference on Computer Vision and Pattern Recognition. 2012 : 3354 – 3361.

[14] Chen C, Seff A, Kornhauser A, et al. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving[C] // IEEE International Conference on Computer Vision. 2015 : 2722 – 2730.

[15] Pomerleau D A. Alvinn: An Autonomous Land Vehicle in a Neural Network[J]. Advances in Neural Information Processing Systems, 1989(4) : 595 – 599.

[16] Lecun Y, Muller U, Ben J, et al. Off-road obstacle avoidance through end-to-end learning[C] // International Conference on Neural Information Processing Systems. 2005 : 739 – 746.

[17] Cordts M, Omran M, Ramos S, et al. The Cityscapes Dataset for Semantic Urban Scene Understanding[C] // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

[18] Fritsch J, Kuhnl T, Geiger A. A new performance measure and evaluation benchmark for road detection algorithms[C] // International IEEE Conference on Intelligent Transportation Systems. 2013 : 1693 – 1700.

[19] Maddern W, Pascoe G, Linegar C, et al. 1 year, 1000 km: The Oxford RobotCar dataset[J]. The International Journal of Robotics Research, 2017, 36(1) : 3 – 15.

[20] Chen X, Ma H, Wan J, et al. Multi-View 3D Object Detection Network for Autonomous Driving[J]. CoRR, 2016, abs/1611.07759.

[21] Broggi A, Bertozzi M, Fascioli A, et al. Shape-based pedestrian detection[C] // Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE. 2000 : 215 – 220.

[22] Dollár P, Wojek C, Schiele B, et al. Pedestrian Detection: An Evaluation of the State of the Art[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34 : 743 – 761.

[23] Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective Search for Object Recognition[J]. International Journal of Computer Vision, 2013, 104 : 154 – 171.

[24] Viola P A, Jones M J, Snow D. Detecting Pedestrians Using Patterns of Motion and Appearance[C] // International Journal of Computer Vision. 2003.

[25] Dalal N, Triggs B. Histograms of oriented gradients for human detection[J]. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, 1 : 886 – 893 vol. 1.

[26] Felzenszwalb P F, McAllester D A, Ramanan D. A discriminatively trained, multiscale, deformable part model[J]. 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008 : 1 – 8.

[27] Sermanet P, Kavukcuoglu K, Chintala S, et al. Pedestrian Detection with Unsupervised Multi-stage Feature Learning[C] // Computer Vision and Pattern Recognition. 2013 : 3626 – 3633.

[28] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553) : 436 – 444.

[29] 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.

[30] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C] // NIPS. 2012.

[31] Girshick R. Fast r-cnn[C] // Proceedings of the IEEE International Conference on Computer Vision. 2015 : 1440 – 1448.

[32] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C] // Advances in neural information processing systems. 2015 : 91 – 99.

[33] Redmon J, Divvala S K, Girshick R B, et al. You Only Look Once: Unified, Real-Time Object Detection[J]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016 : 779 – 788.

[34] Everingham M, Van Gool L, Williams C K, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2) : 303 – 338.

[35] Chen X, Kundu K, Zhu Y, et al. 3d object proposals for accurate object class detection[C] // Advances in Neural Information Processing Systems. 2015 : 424 – 432.

[36] Chen X, Kundu K, Zhang Z, et al. Monocular 3d object detection for autonomous driving[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016 : 2147 – 2156.

[37] Xiang Y, Choi W, Lin Y, et al. Data-Driven 3D Voxel Patterns for Object Category Recognition[C] // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.

[38] Xiang Y, Choi W, Lin Y, et al. Subcategory-aware convolutional neural networks for object proposals and detection[J]. arXiv preprint arXiv:1604.04693, 2016.

[39] Ohn-Bar E, Trivedi M M. Learning to Detect Vehicles by Clustering Appearance Patterns[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(5): 2511 – 2521.

[40] Comaniciu D, Meer P. Robust analysis of feature spaces: color image segmentation[C] // Conference on Computer Vision and Pattern Recognition. 1997: 750.

[41] Jang D S, Choi H I. Active models for tracking moving objects[J]. Pattern Recognition, 2000, 33(7): 1135 – 1146.

[42] Yoon C, Cheon M, Park M. Object tracking from image sequences using adaptive models in fuzzy particle filter[J]. Information Sciences, 2013, 253(18): 74 – 99.

[43] Lenz P, Geiger A, Urtasun R. FollowMe: Efficient Online Min-Cost Flow Tracking with Bounded Memory and Computation[J]. 2015 IEEE International Conference on Computer Vision (ICCV), 2015: 4364 – 4372.

[44] Yoon J H, Yang M-H, Lim J, et al. Bayesian Multi-object Tracking Using Motion Context from Multiple Objects[J]. 2015 IEEE Winter Conference on Applications of Computer Vision, 2015: 33 – 40.

[45] Choi W. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor[C] // IEEE International Conference on Computer Vision. 2016: 3029 – 3037.

[46] Xiang Y, Alahi A, Savarese S. Learning to Track: Online Multi-object Tracking by Decision Making[C] // IEEE International Conference on Computer Vision. 2015: 4705 – 4713.

[47] Sutton R, Barto A. Reinforcement Learning: An Introduction[J]. Trends in Cognitive Sciences, 1999, 3(9): 360 – 360.

[48] Ng A Y, Russell S J, others. Algorithms for inverse reinforcement learning.[C] // Icml. 2000: 663 – 670.

[49] Shotton J, Winn J, Rother C, et al. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context[J]. International Journal of Computer Vision, 2009, 81(1): 2 – 23.

[50] Forgy E W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications[J]. Biometrics, 1965, 21 : 768 – 769.

[51] Cheng Y. Mean shift, mode seeking, and clustering[J]. IEEE transactions on pattern analysis and machine intelligence, 1995, 17(8) : 790 – 799.

[52] Koltun V. Efficient inference in fully connected crfs with gaussian edge potentials[J]. Adv. Neural Inf. Process. Syst, 2011, 2(3) : 4.

[53] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8) : 1915 – 1929.

[54] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015 : 3431 – 3440.

[55] Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation[C] // Proceedings of the IEEE International Conference on Computer Vision. 2015 : 1520 – 1528.

[56] Everingham M, Van Gool L, Williams C K, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2) : 303 – 338.

[57] Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations[C] // Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on : Vol 1. 1999 : 666 – 673.

[58] Geyer C, Daniilidis K. A unifying theory for central panoramic systems and practical implications[J]. Computer Vision—ECCV 2000, 2000 : 445 – 461.

[59] Heng L, Furgale P, Pollefeys M. Leveraging Image-based Localization for Infrastructure-based Calibration of a Multi-camera Rig[J]. Journal of Field Robotics, 2015, 32(5) : 775 – 802.

[60] Mayer N, Ilg E, Hausser P, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016 : 4040 – 4048.

[61] Zagoruyko S, Komodakis N. Learning to compare image patches via convolutional neural networks[C] // Computer Vision and Pattern Recognition. 2015 : 4353 – 4361.

[62] Luo W, Schwing A G, Urtasun R. Efficient Deep Learning for Stereo Matching[C] // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

[63] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[64] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C] // Computer Vision and Pattern Recognition. 2015 : 770 – 778.

[65] Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision (IJCV), 2015 : 211 – 252.

[66] Gulcehre C, Moczulski M, Denil M, et al. Noisy activation functions[C] // International Conference on International Conference on Machine Learning. 2016 : 3059 – 3068.

[67] Nair V, Hinton G E. Rectified Linear Units Improve Restricted Boltzmann Machines[C] // International Conference on Machine Learning. 2010 : 807 – 814.

[68] Clevert D-A, Unterthiner T, Hochreiter S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)[J]. Computer Science, 2015.

[69] He K, Zhang X, Ren S, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[C] // The IEEE International Conference on Computer Vision (ICCV). 2015.

[70] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks.[C] // Aistats : Vol 9. 2010 : 249 – 256.

[71] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting.[J]. Journal of Machine Learning Research, 2014, 15(1) : 1929 – 1958.

[72] Zhou Z-H. Cost-Sensitive Learning[C] // MDAI. 2011.

[73] Erhan D, Bengio Y, Courville A, et al. Why Does Unsupervised Pre-training Help Deep Learning?[J]. Journal of Machine Learning Research, 2010, 11(3) : 625 – 660.

[74] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. arXiv preprint arXiv:1603.04467, 2016.

攻读硕士学位期间发表的论文及其他成果

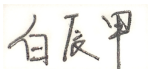
（一）参与的科研项目及获奖情况

[1] 主动视觉中的对抗问题研究，国家自然科学基金项目. 批准号：61672190.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于计算机视觉和深度学习的自动驾驶方法研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：  日期：2017 年 06 月 28 日

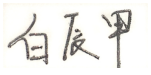
学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：  日期：2017 年 06 月 28 日

导师签名：  日期：2017 年 06 月 28 日

致　谢

硕士生活即将结束，回顾两年来的学习和成长，收获良多。我要感谢学校的培养，感谢师长的关心，感谢同学的帮助。学校为我营造一流的学习环境，使我能够获取宝贵的学习资源；师长为我指明了前进的方向，指导我一步一步在学业上取得进步；同学们营造了一个温馨的集体，在交流和学习中共同进步。

首先我要感谢我的指导老师刘鹏副教授，刘老师是我学习和工作的领路人。刘老师谦虚严谨的治学态度，朴实乐观的生活作风都深深影响着我。刘老师对科学研究有着深刻的认识，对学生的培养兢兢业业，不仅引导我在科研中不断进步，还教给了我很多做人的道理。当我在研究中遇到困难时，刘老师总是鼓励我独立克服困难，坚持不放弃。我会经常与刘老师就自己近期的学习情况进行交流，每次交流都能开阔我的视野，加深我对问题的认识。谢谢您对我的指导，让我受益良多！

感谢我的导师唐降龙教授。唐老师是领域内非常资深的研究者，至今仍奋斗在科研工作的第一线。唐老师培养了很多计算机领域内的人才，可以说桃李满天下。唐老师在学术方面高屋建瓴，见解独到，给了我很大启发。唐老师作为研究中心主任，关心和爱护每一位学生，能够及时发现和解决学生困难。同时，唐老师是一个老党员，在思想信仰上的坚定执着也深深影响着我。

感谢薛怡然师兄，从进入实验室开始，薛怡然师兄就开始指导我的项目和研究工作。他扎实的算法设计能力，精确查找问题的能力，开放敏捷的思维，谦虚谨慎的生活态度都深深影响着我，是我学习的榜样。每次与师兄的合作与讨论，我总能学到一些新思想、新方法，谢谢师兄的热心帮助，帮我解决了很多困难，谢谢！

感谢肖婷师姐，在生活和学习上给了我很多指导，坦诚的与我交流思想，谢谢！

感谢实验室的同学，你们都非常优秀，能够与大家在一起工作和学习是我的荣幸。所有同学都在为研究中心的建设发光发热，大家共同营造了一个温暖的集体，谢谢！

特别要感谢我的父母和姐姐，感谢我能在一个温馨的家庭中成长，感谢你们无私的付出，鼓励我从每一次挫折中走出来，鼓励我不放弃希望，鼓励我乐观生活，鼓励我独立思考，坚持自己的决定，谢谢！

感谢我的所有朋友，是你们帮助我成长，让我的生活不再孤单，谢谢！

本课题承蒙国家自然科学基金“主动视觉中的对抗问题研究”（批准号：61672190）的资助，特此致谢。