



# 华北电力大学 毕业设计(论文)

论文题目： 无人机机载边缘模型  
仿真与加速方法研究

院 系： 控制与计算机工程学院

专 业： 智能科学与技术

班 级： 智能（创新）1901

姓 名： 周依然

学 号： 120191070424

指导教师： 吴华

2023 年 6 月

## 摘 要

无人机作为智能移动机器人领域重点研究对象，自主无人机竞速任务中要求无人机以最快速度、最少碰撞自主穿过赛道，对推动无人机自主飞行控制等的相关技术发展具有重要意义。随着人工智能技术不断发展，基于深度学习的自主无人机竞速方法展现了其广阔的应用前景，然而受限于续航、功耗、算力等因素限制，无人机设备通常难以承担复杂模型推理任务，实现边缘模型加速推理，同时利用好云端资源是解决上述问题的重要手段。

本文首先从自主无人机竞速策略出发，基于 GoogLeNet 框架设计了一款自主无人机竞速算法，能够将单一图像输入直接映射为四个控制指令，同时利用分组逐点卷积策略改进网络模型以适应边端无人机设备，此外，为解决数据来源问题，本文提供了仿真环境搭建结果及数据采集策略。进一步，为实现深度学习类竞速算法边侧部署，本文 RK1808 人工智能计算棒为例，围绕 RK1808 计算板卡设计出一套无人机边缘深度学习模型部署方案。最后，为更好发挥云端计算与存储优势，本文基于 KubeEdge-Sedna 边云协同 AI 框架提出了一种边云协同式自主无人机竞速方案，旨在实现云端训练、边端推理模式并提供兼容小样本模型、难例上云、模型增量更新等机制。

**关键词：**自主无人机竞速，边缘加速计算，边云协同

## ABSTRACT

As an important research object in the field of intelligent mobile robots, drones are required to autonomously maneuver through the racetrack with the fastest speed and the fewest collisions in autonomous drone racing tasks, which is of great significance to the development of related technologies such as autonomous drone flight control. With the continuous development of artificial intelligence technology, deep learning-based autonomous drone racing methods have shown promising applications. However, limited by factors such as endurance, power consumption, and computing power, drone devices often find it difficult to undertake complex model inference tasks and realize edge model acceleration inference. Meanwhile, utilizing cloud resources well is an important means to solve the above problems.

Starting from the strategy of autonomous drone racing, this paper designs an autonomous drone racing algorithm based on the GoogLeNet framework, which can directly map a single image input into four control commands and improves the network model using the group-wise point-wise convolution strategy to adapt to the edge drone devices. In addition, to solve the problem of data sources, this paper provides simulation environment setup results and data acquisition strategies. Furthermore, to realize the edge-side deployment of deep learning-based racing algorithms, this paper takes the RK1808 artificial intelligence computing stick as an example and designs a set of drone edge deep learning model deployment schemes around the RK1808 computing board. Finally, to better exploit the advantages of cloud computing and storage, based on the KubeEdge-Sedna edge-cloud collaborative AI framework, this paper proposes an edge-cloud collaborative autonomous drone racing scheme to realize cloud training, edge inference mode, and provide mechanisms compatible with small sample models, difficult cases to be migrated to the cloud, and model incremental updates.

**KEY WORDS:** autonomous drone racing, edge acceleration computing, edge-cloud collaboration

# 目 录

摘 要 .....	I
ABSTRACT .....	II
第 1 章 绪论 .....	1
1.1 研究背景及意义 .....	1
1.2 国内外研究现状 .....	2
1.3 研究内容和文章组织结构 .....	4
第 2 章 GoogLeNet 框架下基于逐点分组卷积策略的无人机竞速算法研究 .....	5
2.1 基础理论 .....	5
2.1.1 GoogLeNet 简介 .....	5
2.1.2 分组卷积（逐点分组卷积）与通道重组 .....	6
2.2 整体算法框架设计 .....	8
2.2.1 算法流程设计 .....	8
2.2.2 输入图像处理 .....	9
2.2.3 网络结构设计 .....	10
2.3 自主无人机竞速仿真环境搭建及模拟飞行数据采集 .....	12
2.3.1 ROS-Gazebo 仿真平台介绍 .....	12
2.3.2 仿真环境搭建及数据采集过程设计 .....	13
第 3 章 自主无人机竞速算法边缘部署研究 .....	17
3.1 边缘 AI 加速设备硬件架构介绍 .....	17
3.2 RK1808 计算板卡模型部署方案设计 .....	17
3.2.1 模型部署整体流程 .....	17
3.2.2 模型量化 .....	19
第 4 章 自主无人机竞速算法边云协同框架研究 .....	21
4.1 边云协同概念阐述 .....	21
4.2 边云协同式无人机自主竞速方法设计 .....	21
4.2.1 Kubeedge-Sedna 框架介绍 .....	21
4.2.2 边云协同式无人机自主竞速方法框架 .....	24
第 5 章 总结与展望 .....	26
5.1 总结 .....	26
5.2 展望 .....	26
参 考 文 献 .....	28
致 谢 .....	29

# 第 1 章 绪论

## 1.1 研究背景及意义

近年来,智能移动机器人领域发展迅速,无人机(Unmanned Aerial Vehicle, UAV)因其体积小、重量轻、部署成本低、活动范围广等特点成为该领域重点研究对象,围绕实现无人机自主飞行控制的一系列相关技术也不断迭代,而自主无人机竞速这一特殊应用场景是推动无人机自主完成环境感知、路径规划等必要智能化步骤的有力因素之一。自主无人机竞速比赛要求无人机(四旋翼)在无人操纵的情况下以最快速度、最少碰撞穿过赛道,赛道中障碍物主体通常是不同种类的障碍门(不限于),目前较为知名的国际无人机竞速赛事有 IROS 2016-2019 Autonomous Drone Racing<sup>[1]</sup>、AlphaPilot AI Drone Racing Innovation Challenge 2019<sup>[2,3]</sup>、NeurIPS 2019 Game of Drones (GoD)<sup>[4]</sup>等。开发出一套优于人类操控的无人机自主导航方案是自主无人机竞速领域一大目标,现已取得许多成果,如 2022 年瑞士无人机日(Swiss Drone Days)苏黎世大学机器人与感知小组所研发的自主无人机与三位世界优秀飞手同场竞技并取得不俗表现,无人机自主飞行速度可达 100km/h,已近乎超越人类最好成绩<sup>[5]</sup>。传统无人机竞速比赛中需要人工手动操控飞机穿越障碍区域,即无人机机载摄像头将第一人称主视角(First-Person-View, FPV)视频流通过无线电推送至显示设备(现在通常为 FPV 眼镜),飞手凭借经验操控飞机运动。主无人机竞速的价值不仅在于满足人类对于速度的极致追求,更可以为其他需要高速导航的应用场景提供技术支撑,如电网巡检、海上巡逻、桥梁检测、搜索救援、农林植保、危险环境建模等,具有深远研究意义。

以深度学习(Deep Learning)为代表的人工智能技术在近年来呈井喷式发展,深度神经网络(DNNs)因其出色的数据表征能力在计算机视觉、自然语言语言处理等领域有着广泛且成功的应用。深度神经网络在处理自主无人机竞速相关问题也有其固有优势,如算法精度上限高、能够省去环境感知、路径规划等复杂环节直接实现端到端输出,但无人机这一特殊边缘设备往往计算资源不足,电池容量有限,无法承担复杂模型训练及推理过程。解决上述困境的一种可行方案是利用传统云计算方法<sup>[6]</sup>,模型推理与训练过程皆在云端进行,无人机端主要负责采集数据、收发及响应控制指令。但这一方案存在许多弊端:其一,受网络带宽限制,两者之间只能传递有限数据;其二,若多台无人机设备同时向云端发出请求,容易造成网络拥堵,通讯延迟增加;其三,无人机与云服务器通常通过无线局域网连接,局域网覆盖范围有限,无人机高速运动状态下时常会出现网络连接断开的情况,云端很难实现可靠服务。由此,边缘计算(Edge Computing)应运而生,边缘计算是一种分布式计算架构,本质是将计算和通信资源从网络核心下沉至网络边缘,减少云端负载。由于边缘节点更加靠近数据源与用户端,能够改

善数据传输频繁，网络延迟大等云计算短板。在如今万物互联时代，海量数据从无人机等各类边缘设备产生，使边缘设备具备处理任务的能力（如边端推理，云端训练）是实现边缘计算乃至边缘智能（Edge Intelligence）的关键。

进一步，尽管边缘计算可以规避云计算带来的许多问题，但两者之间并不彼此排斥。一方面，当今云计算与云服务仍有其不可替代性，例如 Amazon AWS、阿里云、华为云等基础设施即服务（Infrastructure as a Service, IaaS）提供商能够为广大企业与用户提供基础计算资源；另一方面，边缘端作为云端扩展目前无法代替云端完成所有工作，如深度学习等复杂计算任务往往需要云边两端共同协作完成。据此，边缘计算与云计算两者相依而生，协同发展，边云协同对于边缘计算发展具有重大现实意义。在自主无人机竞速与深度学习领域，边云协同策略同样具有广泛研究空间与发展潜力，例如，当无人机在练习或试飞过程中多次发生碰撞时，表明当前模型仍存在缺陷不足以帮助无人机实现自主飞行，此时边端无人机可以将难例数据一并同步至云端，云端进行模型更新或二次训练，提升模型泛化能力，再将新网络模型下放至边端，如此依靠云边协同实现增量学习式自主无人机竞速策略。

## 1.2 国内外研究现状

为方便自主无人机竞速策略研究，Drew Hanover 等人总结了移动机器人领域实现机器自主导航的软件层经典通用范式<sup>[5]</sup>，图 1-1 展示了其完整 workflow。然而，随着许多深度学习方法被应用于感知、规划、控制中一个或多个阶段，上述三个过程也不再显式出现在这一经典范式中。在感知层面，利用门检测方法事先获得障碍门位置信息可以帮助减少视觉惯性里程计（Visual-Inertial Odometry, VIO）所产生的累积漂移。Alphapilot<sup>[2]</sup>根据这一思路利用卷积神经网络检测障碍门，一旦门中某些位置被捕捉到，再调用经典位姿估计算法（如 Perspective-n-Point, PnP）推算相机与障碍门之间的相对位姿。基于深度学习的门检测方法已广泛应用于感知环节，其通常与传统 VIO 方法结合对无人机进行状态估计，由于深度学习模型对于设备算力有较高要求，基于深度学习的视觉里程计系统<sup>[7-9]</sup>在无人机竞速任务中并未大规模取代传统 VIO 方法。在基于深度学习的自主无人机竞速策略中，规划环节往往与感知环节耦合，Kaufmann 等人<sup>[10]</sup>基于 ResNet 架构训练卷积神经网络模型，将单目图像输入直接映射为目标航点的归一化二维图像坐标和当前无人机速度期望值，之后利用最小加速度轨迹优化方法（Minimum-jerk）生成当前位置到目标航点的轨迹片段。Loquercio 等人<sup>[11]</sup>中采用相似思路，但其训练数据均来自仿真环境。近期深度学习也在控制层面为自主无人机竞速领域开辟了新解决思路，Hwangbo 等人<sup>[12]</sup>证明利用强化学习能够成功控制实体四旋翼无人机，且给定状态下能够在  $7\mu s$  内生成控制指令，但由于基于深度学习的控制策略无法提供控制器稳定性分析，像线性模型预测控制（Linear MPC）等传

统控制策略仍是主流方案<sup>[5]</sup>。此外，借助深度神经网络端到端特性可以根据视觉图像帧直接预测控制指令。基于这一思路，有两种不同实现方案：一是分别针对感知、规划、控制环节训练不同神经网络模型，前者输出作为后者输入以此预测控制指令<sup>[13]</sup>，另一种则是单纯的端到端方法<sup>[14,15]</sup>。



图 1-1 机器人自主导航经典通用范式

如 1.1 所述，要将上述深度学习方法成功落地于自主无人机竞速领域，无人机设备端应具备一定的模型推理能力，这也是边缘计算领域的一个重要研究分支。在硬件加速层面，目前市面上有许多嵌入式边缘计算平台与 AI 硬件加速器可以帮助边端设备卸载计算任务，如 NVIDIA Jetson 系列嵌入式开发平台、英特尔 Neural Compute Stick (NCS)、瑞芯微 RK1808 人工智能计算棒等。在模型适配层面，优化深度学习模型有助于提高模型运行速度、减少计算资源占用和降低功耗，以便其兼容无人机等边端设备<sup>[16]</sup>。模型量化是常见模型优化方法之一，模型量化可以将深度神经网络模型中的浮点计算转化成低比特定点计算从而减小计算开销，如 XNOR-Net<sup>[17]</sup>通过二值化卷积和权重与卷积层输入，在保证模型精度的情况下显著提升了卷积操作速度并大大降低了模型内存占用。此外，可以通过调整模型输入提升推理速度，Zhang 等人<sup>[18]</sup>利用两个前置滤波器和一个小型 YOLO 网络来过滤大量不含目标对象的图像帧，如此能够大幅缩减到达骨干网络（BackBone）的图像数量。ShuffleNet<sup>[19]</sup>则直接通过改进网络结构来提升模型性能，针对小型网络模型中存在大量 $1 \times 1$ 卷积操作导致模型运算量激增这一弊端，ShuffleNet 提出逐点分组卷积和通道重组两种核心操作并据此设计出 ShuffleNet Unit，实验证明 ShuffleNet 在移动设备端可以兼顾运行速度与模型精度。更进一步，要将深度学习方法应用于自主无人机竞速领域也需充分发挥云端节点优势，为实现云边互动，许多企业和组织提出了基于边云协同的开源边缘计算架构（解决方案），如华为在容器化应用程序管理系统 Kubernetes (k8s) 基础之上提出了其边缘计算解决方案 Kubeedge<sup>[20]</sup>，Kubeedge 进一步将 k8s 容器（容器化应用）编排功能由云端扩展至边缘，并为边云网络通信，应用部署，数据同步提供基础技术架构，而 EdgeX Foundry<sup>[21]</sup>则可以将 AI 功能进一步下沉至各种物联网终端设备，通过微服务实现终端设备管理及其与边云节点的通讯。值得一提的是，由 KubeEdge AI SIG 孵化的 Kubeedge 子项目 Sedna<sup>[22]</sup>可以在 Kubeedge 框架下实现 AI 模型云边协同训练推理，这为云边协同共同实现深度学习自主无人机竞速策略提供了坚实基础。

### 1.3 研究内容和文章组织结构

本文主要研究内容为结合人工智能技术发展趋势，充分发挥深度学习方法自主无人机竞速领域的应用价值，深入挖掘基于深度学习的无人机自主竞速方案。针对于边端无人机设备计算资源不足，电池容量有限等问题，探索有效边缘模型部署方案以实现网络模型边侧推理，进一步，为充分利用云端资源，深入研究边云协同模式下自主无人机竞速解决方案。

本文共五章，文章具体结构及各章节内容安排如下：

第一章为绪论。详细分析了本次课题的研究背景，讨论了自主无人机竞速方法、无人机机载边缘加速策略和边云协同计算框架的国内外研究现状，最后介绍了本文研究内容和文章组织结构。

第二章为 GoogLeNet 框架下基于逐点分组卷积策略的无人机竞速算法研究。首先介绍了 GoogLeNet 网络框架和逐点分组卷积策略基本原理，接着提出了一种基于 GoogLeNet 的卷积神经网络，可以将单一图像输入直接映射为四个控制指令 $(\phi, \theta, \varphi, h)$ ，最后展示仿真环境搭建结果及数据采集策略。

第三章为基于深度学习的自主无人机竞速算法边部署研究。为实现深度学习类竞速算法边侧部署，该章节以 RK1808 人工智能计算棒为例，详细介绍了其硬件架构，并围绕 RK1808 计算板卡设计了一套无人机边缘深度学习模型部署方案。

第四章为自主无人机竞速算法边云协同框架研究。详细阐述了边云协同具体概念及其在无人机自主竞速领域的研究价值，接着基于 KubeEdge-Sedna 边云协同 AI 框架设计了边云协同式无人机自主竞速方案。

第五章为总结与展望，对本文所做工作进行系统总结，列举了研究过程中所存在的问题与未来研究方向。



## 第 2 章 GoogLeNet 框架下基于逐点分组卷积策略的 无人机竞速算法研究

### 2.1 基础理论

#### 2.1.1 GoogLeNet 简介

GoogLeNet<sup>[23]</sup>是谷歌公司于 2014 年推出的全新深度神经网络结构，并于同年 ImageNet 图像分类任务中击败 VGGNet 取得冠军。在此之前大多数经典 DNNs 结构（如 AlexNet, VGGNet）都是通过不断堆叠网络深度来提升模型学习效果，但一味增加网络深度会带来过拟合、参数爆炸、梯度消失等问题。据此 GoogLeNet 在网络结构设计时创新性地运用模块化思想，引入 Inception 模块（Inception module），在增加网络模型深度与广度的同时显著减小了网络参数数量，Inception 模块结构如图 2-1 所示。

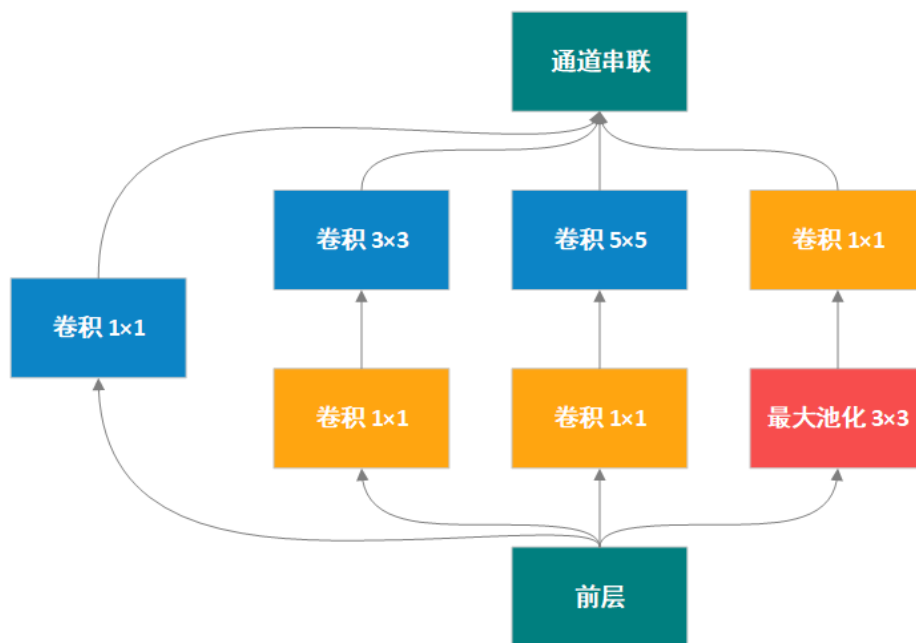


图 2-1 Inception 模块结构图

由上图可以揽储 Inception 模块共有 4 个分支，3 个不同大小卷积核 和 1 个最大池化单元。Inception 模块将不同大小卷积核在各分支中排列组合，使网络能够学习到不同尺度特征，从而增强其对于图像信息的表征能力。在上述三种卷积核中，使用 卷积核进行的卷积操作称为逐点卷积（Pointwise Convolution, PW），其作用主要体现在两方面：一方面，最大池化单元无法降低特征图通道数，若在此分支上不做任何处理，Inception 模块的特征图输出深度必然会大于输入深度，如此网络深度与计算量将会不断累积，引入 卷积核能够对特征图进行压缩降维，减少网络参数数量；另一方面，受 Network in Network（NiN）<sup>[24]</sup>启发，连续卷积

操作（如先进行 $1\times 1$ 卷积，再进行 $3\times 3$ 卷积）能够提升模型非线性拟合能力，逐点卷积与一般卷积操作流程对比见图 2-2。需要注意的是，本文并未采用 GoogLeNet 完整网络结构，而是对其做出了删减与改进以更加适配无人机设备，此外，Inception 模块如今经过了多次发展迭代（Inception v1, Inception v2, Inception v3, Xception），本文后续在自主无人机竞速算法设计中采用 Inception v1 结构。

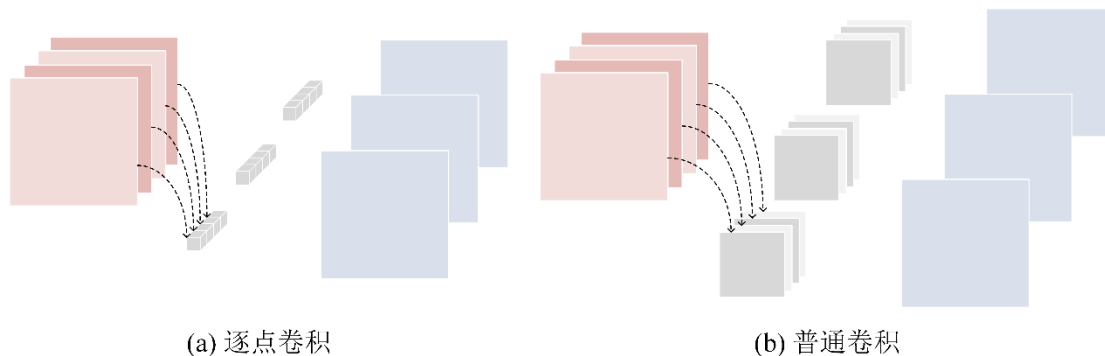


图 2-2 逐点卷积与普通卷积对比图

### 2.1.2 分组卷积（逐点分组卷积）与通道重组

前述内容可知，逐点卷积对降低 GoogLeNet 计算开销具有关键作用，但由图 2-2 易知逐点卷积与常规卷积操作皆为全通道卷积，即特征图通道数等于卷积核通道数，而在轻量化网络结构中逐点卷积运算量在网络总运算量中占比很大，反而成为影响网络性能的一大因素。受 ShuffleNet<sup>[19]</sup>启发，利用分组逐点卷积实现通道稀疏连接能够进一步降低网络模型计算量与参数数量，即先将输入特征图按通道（Channels）分为若干组，之后每组再进行逐点卷积操作，图 2-3 展示了其核心思路。

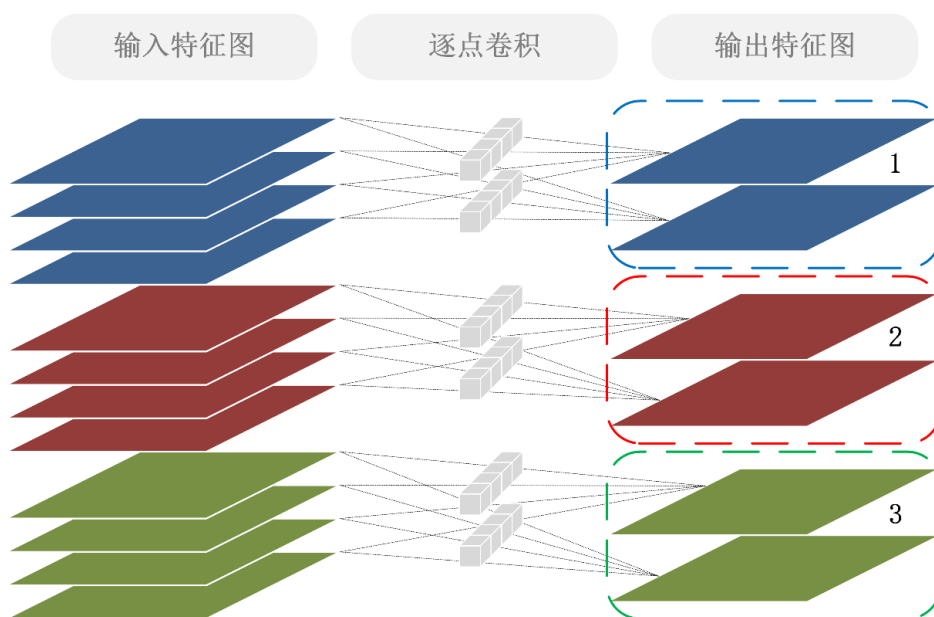


图 2-3 分组卷积（逐点分组卷积）示意图

假设某卷积神经网络中某卷积层输入特征图维度为  $H \times W \times C$ ，输出特征图维度为  $H' \times W' \times C'$ ，单个卷积核大小为  $K \times K \times C$ ，其中  $C$  表示输入特征图通道数， $H \times W$  表示输入特征图大小， $C'$  表示输出特征图通道数， $H' \times W'$  表示输出特征图大小。对于常规卷积操作，整个卷积层（带偏置）在全特征图上的参数量和计算量(FLOPs, 定义一个浮点加法运算和一个浮点乘法运算为一个 flop) 如式(2-1)所示：

$$\begin{aligned} params &= (K \times K \times C) \times C' = (K^2 C + 1) \times C' \\ FLOPs &= K \times K \times C \times H' \times W' \times C' = K^2 C \times H' \times W' \times C' \end{aligned} \quad (2-1)$$

对于分组卷积操作，若共有  $g$  个分组，每个分组有  $n$  个卷积核，则整个分组卷积的参数量和运算量如式(2-2)所示：

$$\begin{aligned} params' &= \left( K \times K \times \frac{C}{g} + 1 \right) \times \frac{C'}{g} \times g = \left( \frac{K^2 C}{g} + 1 \right) \times C' \\ FLOPs' &= K \times K \times \frac{C}{g} \times H' \times W' \times \frac{C'}{g} \times g = K^2 \frac{C}{g} \times H' \times W' \times C' \end{aligned} \quad (2-2)$$

其中满足  $C' = g \times n$ ， $C|g$  且  $C'|g$ ，可以看出输入输出大小相同时，分组卷积计算量是常规卷积的  $1/g$ ，若不考虑偏置项，分组卷积参数量也是常规卷积的  $1/g$ 。而分组逐点卷积作为分组卷积的一种特殊情况（ $K=1$ ），仍满足上述结论。

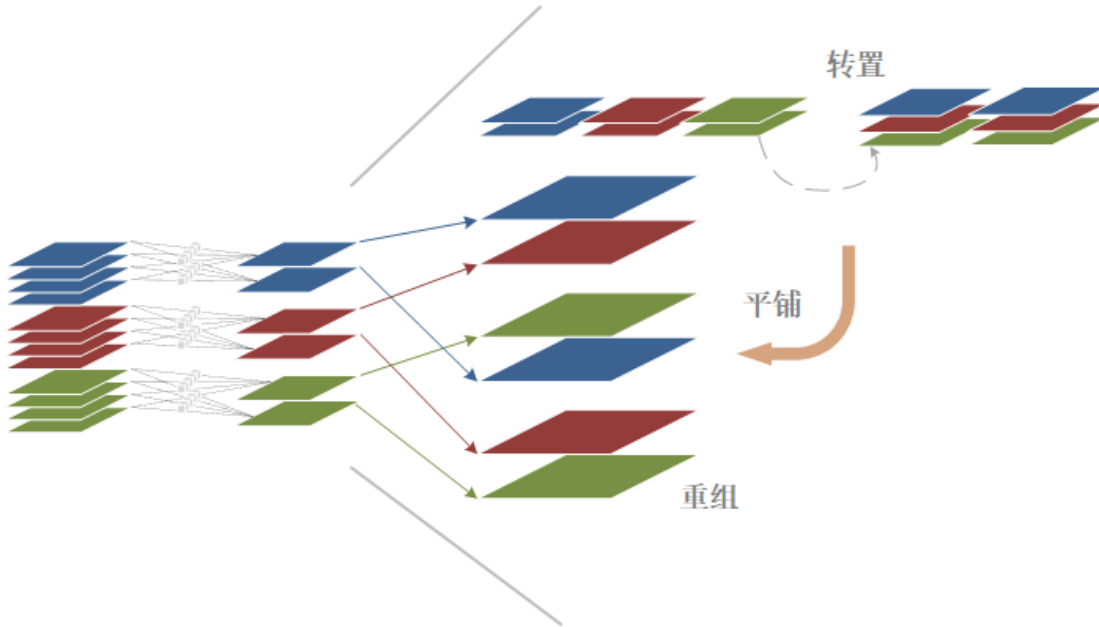


图 2-4 通道重组过程示意图

然而，分组卷积（分组逐点卷积）中每组输出仅和其组内输入相关，即组与组之间无法进行特征通信，以图 2-3 为例，组 1 中 2 个特征图输出（蓝色部分）实际只利用了组 1 中 4 个输入特征图信息，而与其他两组信息无关，这会一定程

度上导致网络特征提取能力降低，进一步导致模型精度下降。针对这一弊端，可以在分组卷积结束后对所有输出特征图进行排列组合以实现组间特征信息交换，这一过程被称为通道重组（Channel Shuffle），通道重组保证了之后的卷积层输入能够利用全组特征信息。具体重组策略为：依据先前假设，分组卷积后输出特征图通道数输出特征图维度为 $C = g \times n$ ，首先将 $C$ 所在维度重构为 $(g, n)$ ，转置之后再重新平铺为 $g \times n$ 个通道，图 2-4 为通道重组过程示意图（基于图 2-3）。

## 2.2 整体算法框架设计

### 2.2.1 算法流程设计

由 1.2 可知，传统自主无人机竞速策略分为感知、规划、控制三个阶段，每阶段有其特定的目标任务与服务范围，若一一针对每个环节设计具体方案无疑会降低任务可实施性，而借助神经网络端到端特性可以在保证精度前提下充分降低设计与实现难度。依据这一思路，本节主要介绍在 GoogLeNet 框架下设计深度神经网络结构以代替传统竞速策略中的三个环节，并基于上述分组逐点卷积策略做出进一步改进，最终实现无人机单目图像输入（第一人称主视角）到控制指令的直接映射，算法整体流程如图所示。无人机所采集的多张连续图像帧首先会排列整合成一张拼贴图像（后续称之为 Mosaic，生成方法详见 2.2.2）作为实际网络输入，最终网络将预测输出四个指令 $(\phi, \theta, \varphi, h)$ ，其中 $\phi, \theta$ 为角度指令，分别表示无人机机架的滚转角和俯仰角， $\varphi, h$ 为速度指令，分别表示无人机偏航角转速和垂直方向速度（也可近似理解为高度）。

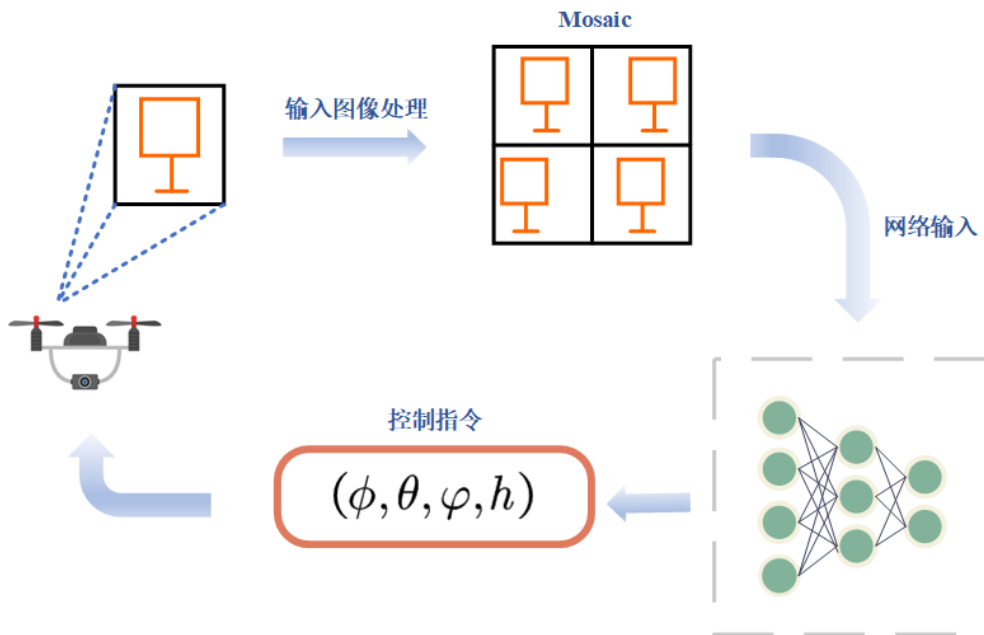


图 2-5 自主无人机竞速算法流程图

### 2.2.2 输入图像处理

上一小节提到在本文所提出的算法框架中网络输入并非常规单一图像，而是由多张连续图像帧拼贴而成的 **Mosaic**，采用这一设计的主要原因有两个方面：一方面，受启发于视频分类任务，此类任务中利用图像帧间的时序信息是其研究领域中重要一环，而视频序列中相邻图像帧之间往往存在大量信息冗余，故通常会事先对视频序列进行采样，再将样本数据一次或多次送入网络。同理，无人机设备所采集到的连续图像帧也一定程度上包含了某些时间维度信息（如无人机相对于障碍物的运动趋势），此类信息若能成功被网络学习将有助于提升网络模型精度以预测出更为有效的控制指令；另一方面，若将多张图像帧打包共同作为网络输入，在后续进行特征融合时容易导致通道数冗余，网络参数过多等问题，这并不符合无人机设备对于轻量化神经网络结构的要求，**Mosaic** 能够在单一图像输入条件下同时为网络提供空间维度（图像本身表层特征）与时间维度信息（运动变化特征）。

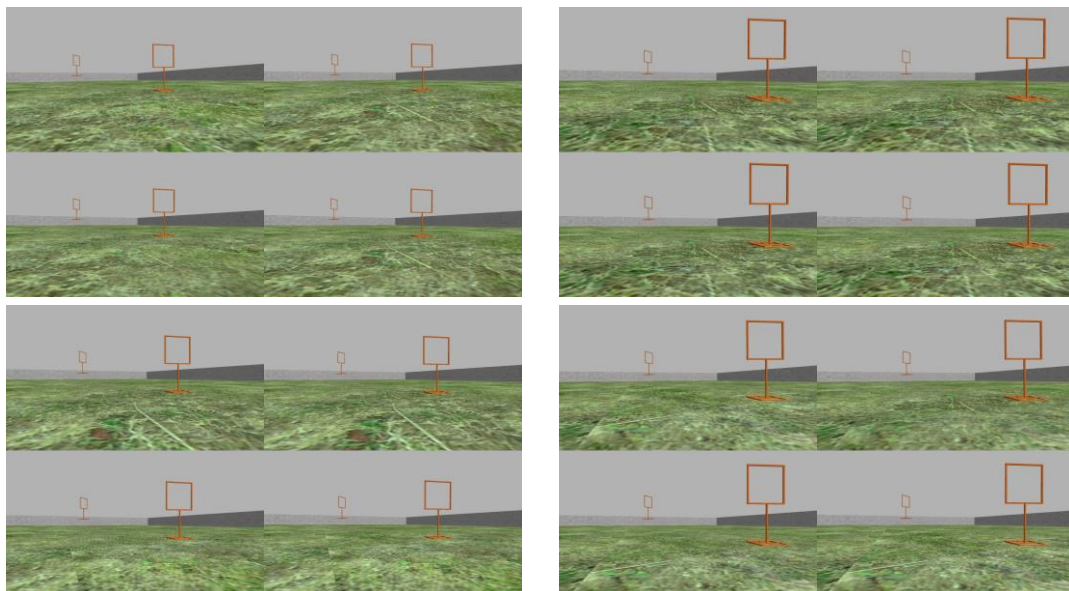


图 2-6 Mosaic 示例图

**Mosaic** 生成方法中需重点解决两个问题：（1）确定组成 **Mosaic** 的图像帧数量；（2）**Mosaic** 更新策略。问题（1）是一个自定义问题，理论上可以以任意数量图像帧生成 **Mosaic**，图像帧数量越多，可以为网络提供更多特征信息，但同时时间成本也更大。确定图像帧数量的一种可行方案是事先定义好若干备选选项，如 4、6、8 个图像帧，之后分别据此构建训练集与测试集，比较模型在不同数据集上的表现以选定最佳方案。问题（2）中 **Mosaic** 更新策略主要用于模型验证阶段，该阶段需要根据无人机机载摄像头采集到的视频流实时生成 **Mosaic**，更新 **Mosaic** 时应按照一定采样频率（如每隔 5 帧对当前视频流采样一次）。此外，由于无人机自主避障时飞行速度具有波动性，速度变化量达到一定阈值时更新频率



也应做出相应调整，例如无人机在起飞阶段速度通常比较缓慢，此时可以采用逐帧采样方法，若 Mosaic 位置已满，应根据先进先出原则移除当前 Mosaic 中第一个图像元素，注意完整 Mosaic 生成后图像大小往往不符合网络输入统一标准，送入网络推理前应先经过缩放处理，Mosaic 更新流程详见图 2-5。

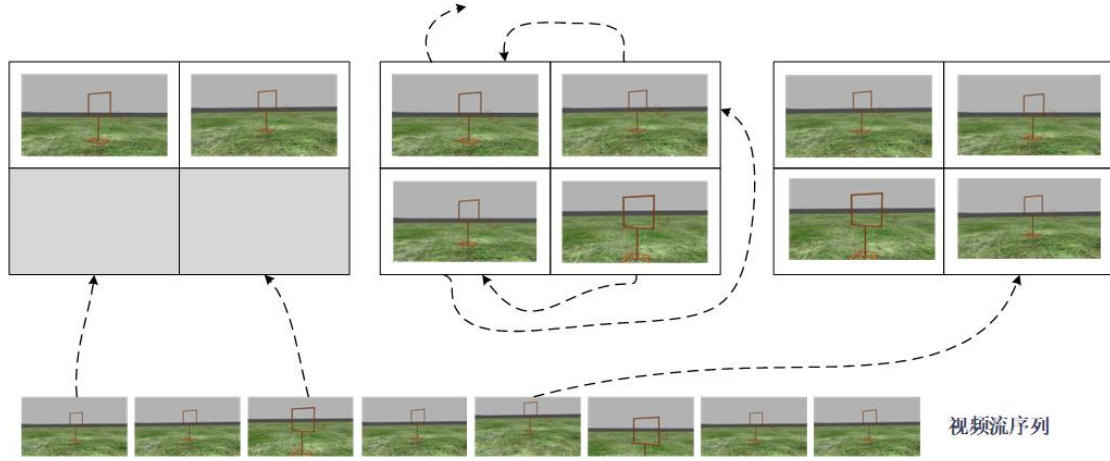


图 2-7 Mosaic 更新流程示意图

### 2.2.3 网络结构设计

在网络结构设计中，相比于单分支网络，多分支网络结构中每个分支可以只预测输出某一项明确指令，这往往有助于提升模型精度，且对于输入数据变化具有较强鲁棒性。据 2.1.1 中所述，本文网络输出为四个控制指令 $(\phi, \theta, \varphi, h)$ ，由于 $(\phi, \theta)$ 皆为无人机角度姿态， $\varphi$ 和 $h$ 为速度指令，故计划将 $(\phi, \theta)$ 作为同组输出， $\varphi$ 和 $h$ 分别作为单组输出。图 2-6 展示了本文网络结构整体框架，该网络共有三个并行分支，每个分支具有相同网络结构不同网络权重，第一分支（从上至下）预测无人机滚转角与俯仰角，第二分支预测偏航角速度，第三分支预测垂直速度。单个分支网络结构参照 GoogLeNet 框架设计，每分支共有 4 个卷积层，3 个 Inception 模块（具体介绍详见 2.1.1），4 个最大池化层，1 个全连接层，2 个批标准化层，为进一步轻量化网络结构以适应无人机设备，本次工作并未采用完整 GoogLeNet 框架，而是对其做出了如下改进：

（1）舍弃辅助分类器结构。GoogLeNet 在网络不同层次中插入了两个辅助分类器，主要由两个连续全连接层和一个 Softmax 分类器组成，在模型训练阶段辅助分类器能够在网络中间层提供额外分类预测，进一步增强网络梯度流，同时辅助分类器误差也会依权重加入到主损失函数中（相当于加入正则因子）防止网络模型过拟合，但本文中网络只需预测输出 4 个控制指令，并非强分类任务，辅助分类器虽能带来模型精度上的提升，但也引入了大量网络参数，增加了额外计算开销。

（2）Inception 模块个数减至 3 个。与（2）类似，减少 Inception 模块个数也是为了进一步简化网络结构，GoogLeNet 原始结构中共采用了 9 个 Inception 模块，

网络深度达到 27 层（包括池化层），考虑到无人机等边缘设备受计算资源与功耗限制以及后续模型部署难度，在网络结构设计时应尽量减小模型复杂度。

（3）使用分组卷积（分组逐点卷积）与通道重组策略。如图 2-6 所示，本文网络结构除第一个卷积层外，其余三个卷积层皆使用分组卷积策略（第二与第四卷积层为逐点分组卷积），为避免组间特征信息隔离，本文设计在第二与第三两个连续卷积层之间进行通道重组，即在第二层分组卷积操作后重新组合各分组输出特征图（重组策略详见 2.1.2）。分组卷积策略能够进一步减少模型参数量与计算力（详细推导见 2.1.2），而通道重组则能够保证组间信息交互以减少特征损失。

$$loss(I) = \|\hat{c} - c\|_2, \hat{c}, c \in (\phi, \theta, \varphi, h) \quad (2-3)$$

网络损失函数采用平方损失函数，如式(2-3)中所示， $\hat{c}$  表示某一控制指令预测值， $c$  表示对应真值，要完成所有至指令预测，需计算 4 次损失函数。

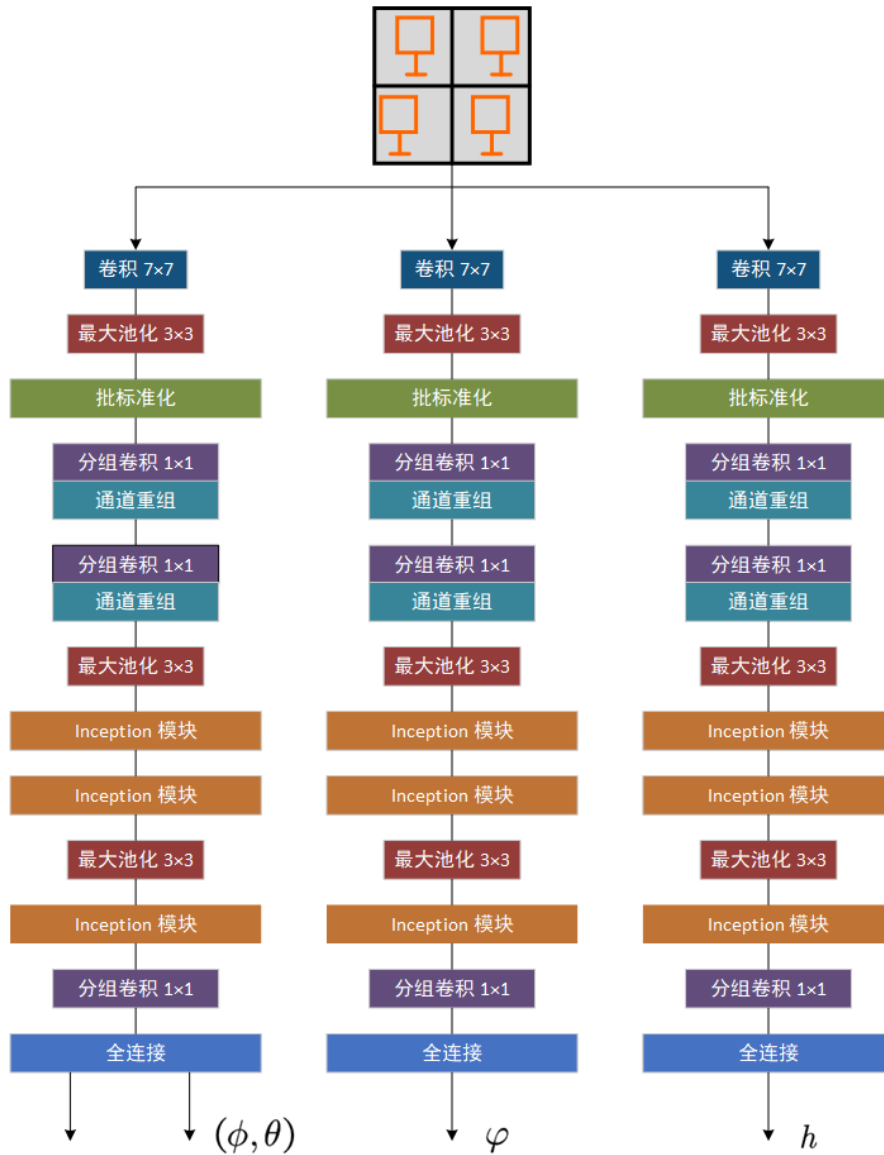


图 2-8 网络结构图

## 2.3 自主无人机竞速仿真环境搭建及模拟飞行数据采集

### 2.3.1 ROS-Gazebo 仿真平台介绍

ROS (Robot Operating System) 是一个应用于机器人开发的开源元操作系统<sup>[25]</sup>，起源于斯坦福大学于 2007 年设立的 STAIR 研发项目，此后 ROS 不断发展壮大，2014 年 OSRF 发布了第一个 ROS 长期支持版 ROS Indigo，为更好地服务于机器人开发领域，OSRF 每两年发布一个 ROS 长期支持版并与 Ubuntu 发行版生命周期（5 年）同步，目前 ROS 最新长期支持版本为 ROS Noetic。ROS 核心功能主要分为通信机制、开发工具、应用功能、生态系统四个方面，ROS 在松耦合分布式通讯框架下通过发布话题消息来连接各个进程节点，并具备物理仿真平台（Gazebo）、三维可视化工具（Rviz）、命令行编译工具（Catkin）等基础开发组件，能够实现机器人导航、建图、控制规划等多种应用功能。此外，ROS 庞大且丰富的生态也是其重要特点之一：个人和企业能够在 Github 等开源代码仓库共享软件包；ROS wiki 提供了完备优质的 ROS 教程与开发文档；全世界机器人开发人员可以在 ROS Answers 中查询并提问相关技术难点。

Gazebo 是一款开源机器人物理仿真平台，用于开发和测试机器人软件。Gazebo 通过自带物理引擎模拟重力、摩擦力、碰撞、惯性等基本物理现象，具体来说，物理引擎根据机器人自身模型和物理参数，能够计算出机器人在各个状态下的运动学和动力学特征，从而模拟机器人和真实环境间的交互过程。同时，它可以集成到 ROS 中，使开发人员可以在 ROS 中快速构建和测试机器人控制，感知和路径规划等算法，Gazebo 界面示意图如图 2-7 所示。总得来说，Gazebo 具有以下特点：

（1）支持多种机器人：Gazebo 支持多种机器人模型，包括四轮驱动车辆、无人机、人形机器人等。用户可以自由地添加和修改机器人模型以满足特定仿真需求。

（2）支持多种传感器：Gazebo 支持多种传感器模型，如激光雷达、RGB-D 相机、摄像头、IMU 等。这些传感器可以用于获取机器人周围的环境信息，从而实现定位、建图和导航等任务。

（3）自定义模拟场景：Gazebo 允许用户自定义模拟场景，包括障碍物、建筑物、贴图等。一方面，Gazebo 官方提供了大量基础场景元素（SDF 文件），用户在 Gazebo 中只需通过简单拖动便可搭建出基础仿真场景（世界文件），另一方面，用户可以根据自身需求将其他外部场景元素模型导入 Gazebo，或者直接修改原有世界文件来创建更加复杂的仿真场景。

（4）良好的性能与可扩展性：Gazebo 使用高性能物理引擎和 OpenGL 渲染引擎，能够处理大量机器人、传感器和环境元素。此外，Gazebo 还支持插件机制，用户可以通过编写插件来扩展系统功能。

Gazebo 和 ROS 两者紧密相关，Gazebo 是 ROS 中最主流的物理仿真器，由



于 ROS 完整版本默认安装 Gazebo，因此 Gazebo 通常被视为 ROS 的一部分。开发人员可以使用 ROS 在 Gazebo 仿真环境中对机器人实施控制，如利用 `roslaunch` 命令可以方便地启动带有机器人的仿真场景；ROS 某些元功能包（`gazebo_ros_pkgs`）能够保证 ROS 进程节点与 Gazebo 仿真实体进行通信。总之，Gazebo 为 ROS 开发机器人应用程序提供重要技术部件，ROS 提供了与 Gazebo 交互的工具和服务，使得开发人员可以更轻松地使用 Gazebo 进行仿真实验。在自主无人机竞速研究中，Gazebo 与 ROS 相结合可以帮助开发人员有效测试和优化自主无人机控制算法，加速研究进展。

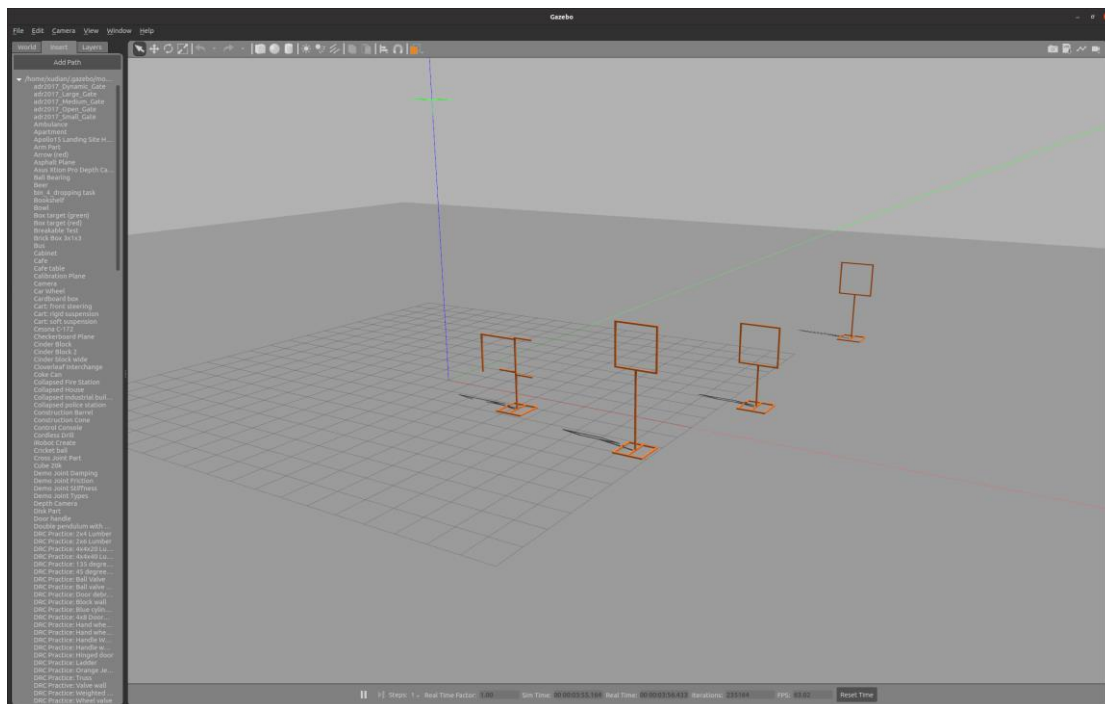


图 2-9 Gazebo 界面展示图

### 2.3.2 仿真环境搭建及数据采集过程设计

模型数据集通常有两种来源，一种是在真实场景中采集，另一种则是搭建仿真环境获取，本文基于第二种方法，以 ROS-Gazebo 仿真平台为例介绍数据采集过程。ROS-Gazebo 仿真环境搭建流程如下：

- (1) 无人机建模。如今许多企业、组织和无人机自动驾驶平台开源无人机模型，如 RotorS、PX4、Parrot AR.Drone 及 Parrot bebop2 等。
- (2) 搭建仿真赛道场景，主要包括赛道地形、飞行障碍物等。
- (3) 编写 launch 文件，将无人机模型与赛道场景集成至 Gazebo。
- (4) 利用 ROS 编程工具包 (rospy) 实现 Gazebo 中无人机模型和控制程序 (自主无人机竞速算法) 桥接。

图 2-11、图 2-12、图 2-13 和图 2-14 展示了 ROS-Gazebo 仿真环境搭建结果，共有 2 个竞速赛道场景（椭圆型赛道与扁圆形赛道），赛道中障碍门高度与朝向

皆为任意值，无人机模型采用 Parrot bebop2，其仿真示意图如图 2-10 所示。可以实现仿真环境手动控制飞机。



图 2-10 Parrot bebop2 无人机仿真示意图

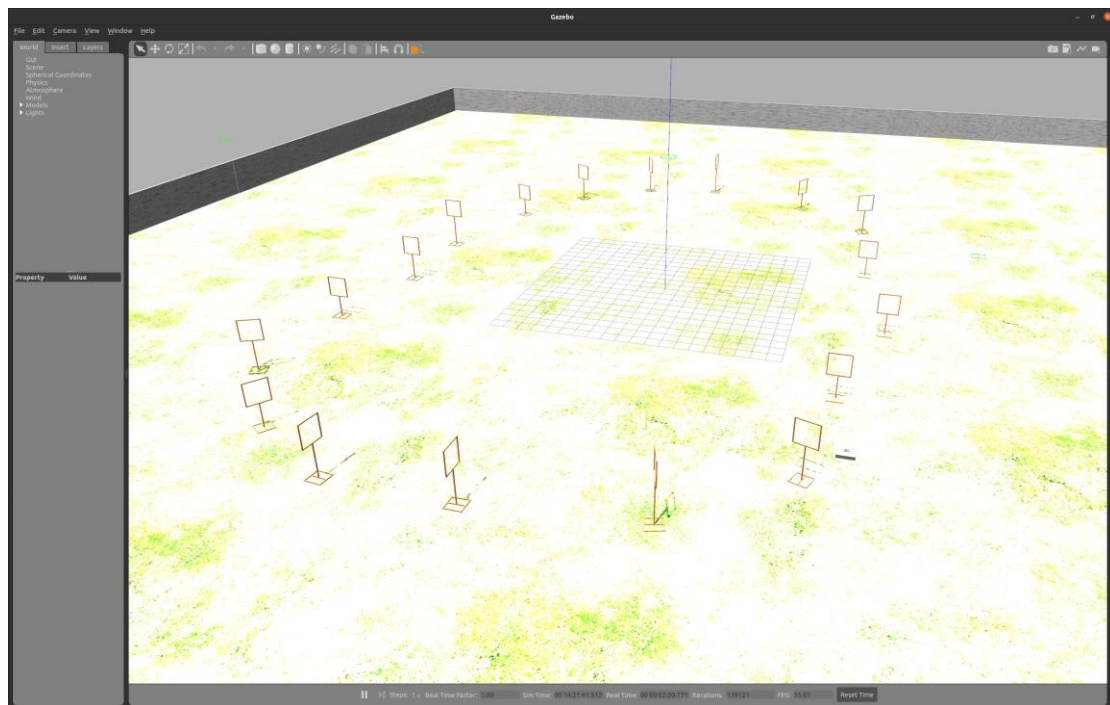


图 2-11 椭圆形赛道仿真环境展示图

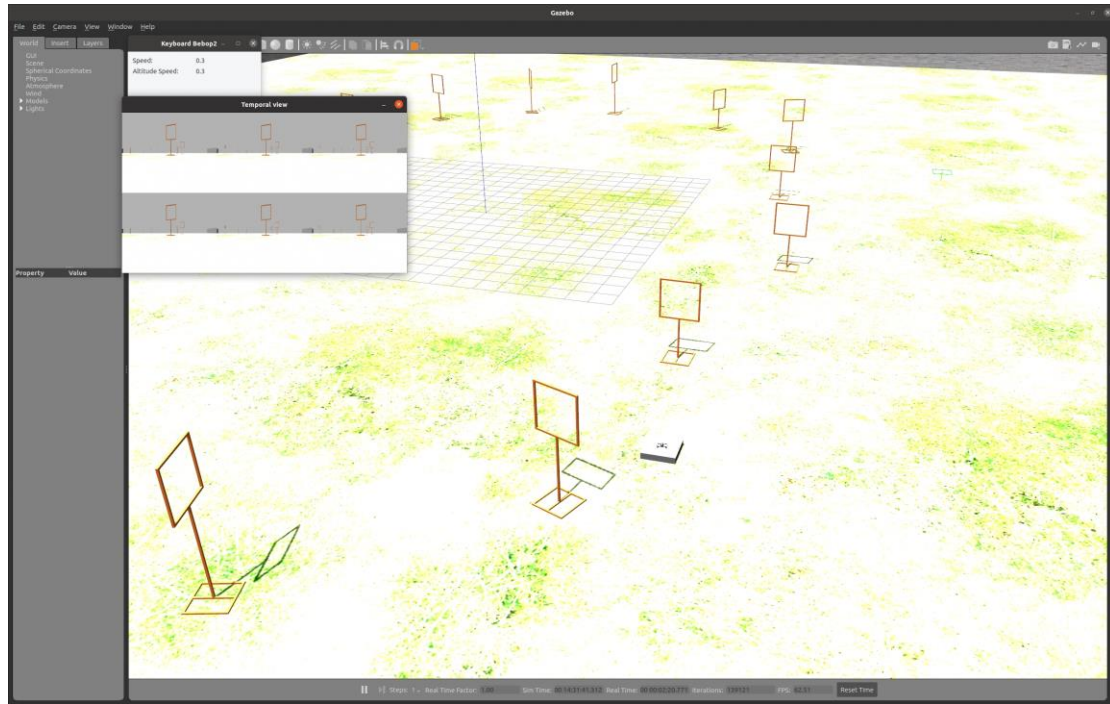


图 2-12 椭圆形赛道仿真环境数据采集示意图

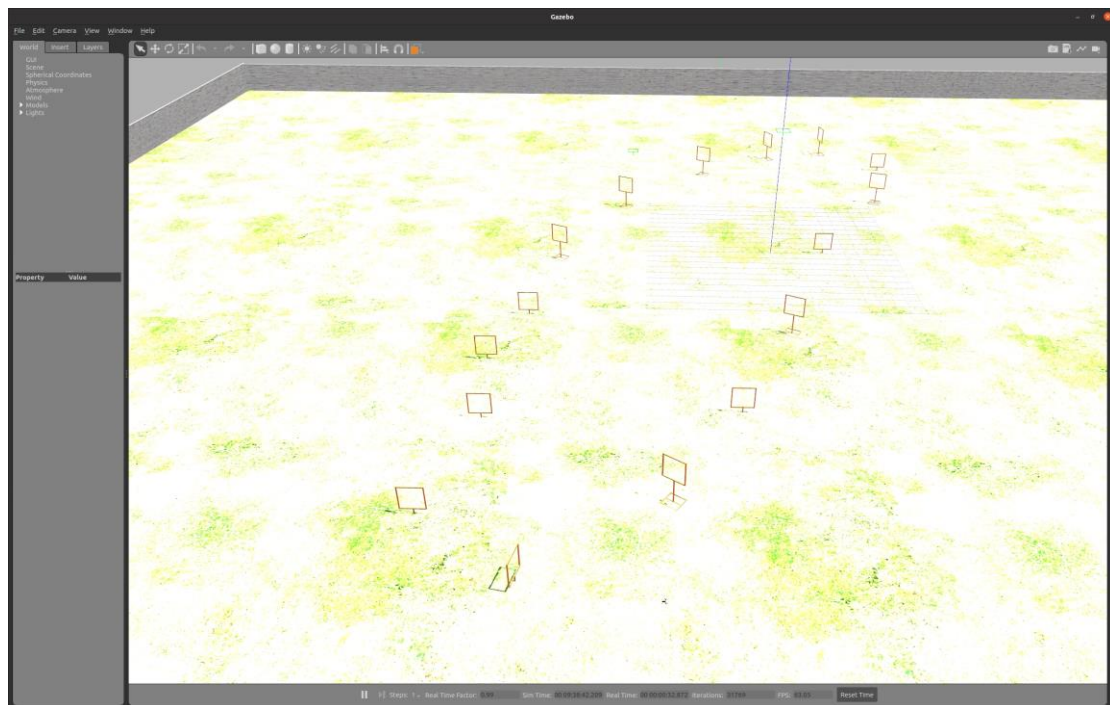


图 2-13 扁圆形赛道仿真环境示意图



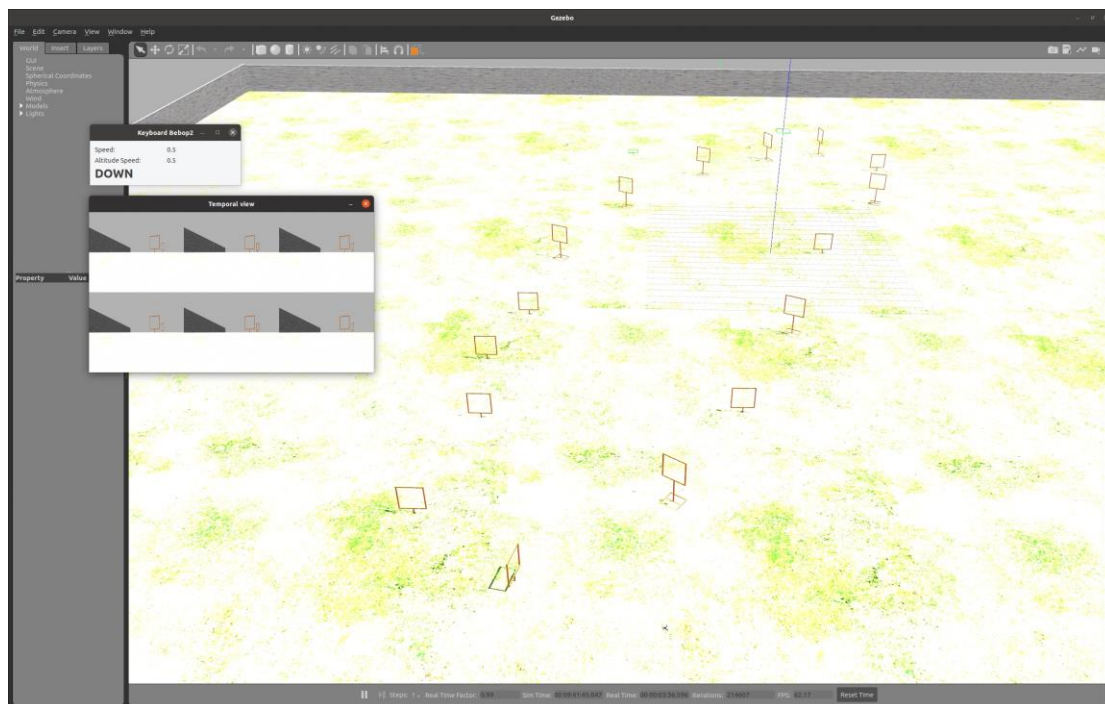


图 2-14 扁圆形赛道仿真环境数据采集示意图

为简化数据集构建过程，无人机在穿越障碍门时的动作响应可以大致分为 4 类（七个基本动作），分别为直行，左右平移，上下运动，顺时针逆时针旋转，整个数据集构建围绕上述七个基本动作。具体来说，不同动作响应中障碍门位于无人机视野的不同位置，同样也对应不同控制指令。以向右运动为例，此时无人机实际位于障碍门左侧，障碍门位于无人机视野右侧，人手动操控无人机运动，一旦指令发出，该指令 $(\phi, \theta, \varphi, h)$ 与其对应视野图像将被同时记录，其中图像作为样本数据，指令作为标签数据，其余六个基本动作数据采集过程与之类似。记录完成后，需要每张图片所对应的指令标签做手动调整，这是由于 $(\phi, \theta, \varphi, h)$ 中每个指令元素对于某一基本动作贡献不同，如左右平移运动时，起支配作用的指令为 $\phi$ ；上下运动时，支配指令为 $h$ ，为避免模糊指令，手动调整原则应尽量增加支配作用的指令元素值，于此同时弱化其他指令元素，图 2-9 展示了向右运动时数据采集及处理过程。

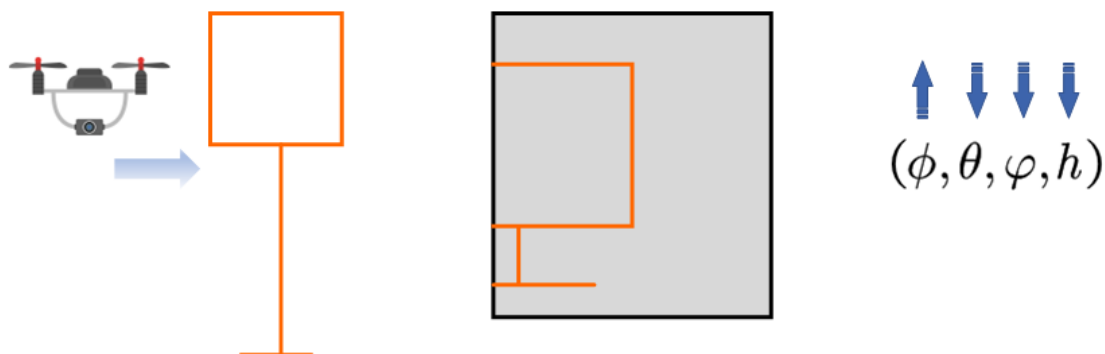


图 2-15 数据采集策略示意图（以向右运动为例）

## 第 3 章 自主无人机竞速算法边缘部署研究

### 3.1 边缘 AI 加速设备硬件架构介绍

RK1808 人工智能计算棒是瑞芯微旗下 Toybrick 系列产品之一，其外观尺寸如图 3-1 所示。RK1808 人工智能计算棒搭载 RK1808 芯片，RK1808 芯片由双核 Cortex-A35 和内置 NPU 处理器组成，采用 22 纳米工艺制造，集成了高速 DDR3/LPDDR3 内存控制器与大量接口，同时支持多种加密算法，具有低能耗、高性能的特点，能够为 RK1808 人工智能计算棒提供高效 AI 推理性能。计算棒内置 2GB LPDDR3 内存和 16GB EMMC 存储，提供高速内存访问速度与一定存储空间。在外部接口方面，RK1808 计算棒仅包括一个 USB3.0 Type-A 接口，该接口既用于供电和数据传输，也可连接外部摄像头和其他设备。简单来说，RK1808 人工智能计算棒被设计成一个简单 USB 设备，具有 USB 接口的边缘计算嵌入式平台可利用计算棒推理性能在网络边缘侧部署人工智能算法，使计算资源受限的嵌入式设备完成人工智能升级。在自主无人机竞速领域，无人机机载电脑通过结合 RK1808 计算棒能够加速网络模型推理，从而使得深度学习类自主无人机竞速算法在边缘本地部署成为可能。此外，RK1808 人工智能计算棒既可用于辅助推理计算，也支持通过二次开发完成独立人工智能计算功能。需要注意的是，RK1808 计算棒只能运行自身 RKNN 模型文件，故使用 RK1808 计算棒必须将常规框架(Pytorch/Tensorflow/Caffe)下所导出的网络模型转换为 RKNN 模型。



图 3-1 RK1808 人工智能计算棒外观尺寸图

### 3.2 RK1808 计算板卡模型部署方案设计

#### 3.2.1 模型部署整体流程

在本次工作中模型部署具体指使已经过训练的神经网络模型能够在 RK1808 AI 计算平台中运行，卸载无人机机载电脑（Onboard Computer）模型推理任务，

从而使深度学习类竞速算法能够成功部署于无人机设备。RK1808 计算板卡支持主动和被动两种开发模式，被动模式下，RK1808 作为其上位机的一个外接 USB 计算设备，被动接收网络模型和输入数据，整个过程中 RK1808 与上位机数据传输需依靠 RKNN-Toolkit 完成；主动模式下，RK1808 计算棒可视为一款具备 AI 推理能力的特殊主机，且内部默认预装 RKNN-Toolkit 与 RKNN-API，上位机无需再次安装。其作为主设备，经过配置后能够集成所需网络模型，上位机只需将输入数据（如图片或视频流）通过 USB 接口输入至计算棒，RK1808 计算棒将自动完成模型推理并将输出结果再次通过 USB 接口返回至上位机。目前被动模式仅支持有限平台，诸如 NVIDIA Jetson、Raspberry Pi 等常用移动端嵌入式计算平台（包括无人机机载电脑）皆无法使用被动模式开发，而主动模式具有平台无关性，不限制上位机系统架构且可搭载任意操作系统，故后续无人机竞速模型部署流程皆基于主动模式设计。

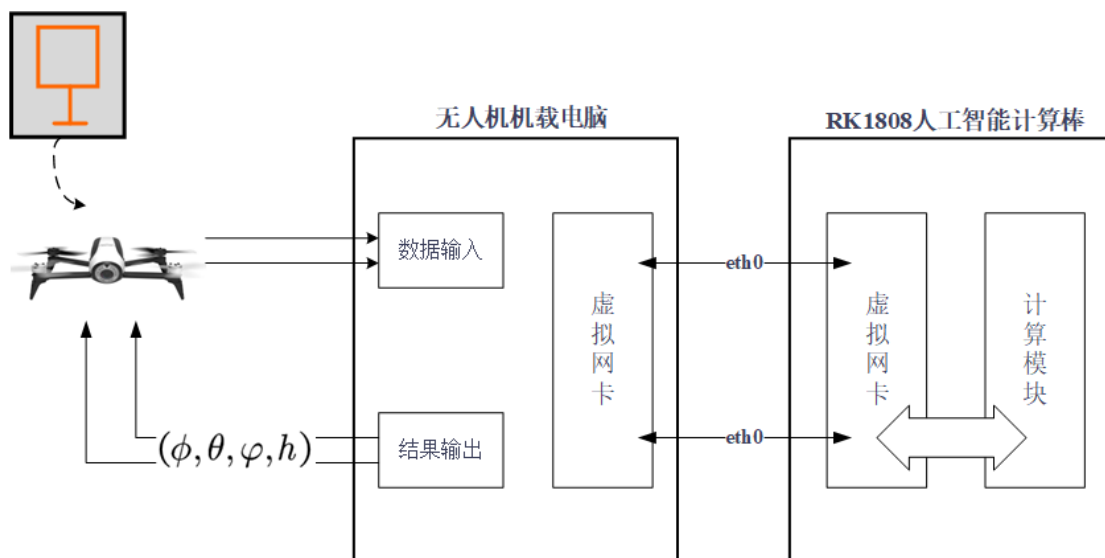


图 3-2 自主无人机竞速模型部署方案示意图

图 3-2 展示了单计算棒条件下无人机竞速模型部署整体流程，此时上位机特指无人机机载电脑。基于 3.1 中内容，RK1808 计算板卡只支持运行 RKNN 模型，故模型部署首先应利用 RKNN-Toolkit 进行模型转换，将开源深度学习框架导出的模型转成 RKNNPU 所能识别的 RKNN 模型，其中模型量化是模型转换中的重要步骤，具体细节详见 3.2.2。模型转换完成后，需建立机载电脑与 RK1808 之间的网络连接，具体来说，RK1808 计算棒通过 USB 接口插入机载电脑后，会作为虚拟网卡设备在机载电脑端生成网络节点，机载电脑端需利用网络管理工具对该虚拟网络节点配置 IP、子网掩码等基本属性信息，保证机载电脑与 RK1808 之间能够进行正常网络通信。之后，机载电脑端可以利用上述 IP 远程登录 RK1808 计算棒（实际中内置操作系统为 Fedora），将所需 RKNN 模型固化于 RK1808。最后，完善模型推理数据流，即 RK1808 计算棒运行服务端程序以接

收网络输入数据并自动进行模型推理；与此同时，机载电脑端运行客户端程序向计算棒发送推理计算请求与网络输入数据，待计算棒完成推理后返回输出结果。由此无人机机载电脑与 RK1808 人工智能计算棒之间形成了典型 CS 模式，两者得以相互配合完成复杂网络模型推理任务。

### 3.2.2 模型量化

模型量化是一种深度神经网络模型优化常用技术，其基本原理是建立浮点数与定点数的数据映射关系，从而将网络模型中浮点参数转化为定点参数，减少存储与计算所需资源。具体来说，量化模型使用较低精度（如 int8/uint8/int16）保存模型权重信息，在部署时可以使用更少存储空间获得更快推理速度。由于各深度学习框架训练、保存模型时通常使用浮点数据，故模型量化是 RK1808 计算棒模型转换过程中非常重要的一环。RKNN-Toolkit 是 RK 系列计算平台经常使用的模型转换工具，用户可以通过安装 RKNN-Toolkit 直接调用 Python 接口完成进行模型转换、推理运行和性能评估等环节。目前 RKNN-Toolkit 主要支持两种模型量化形式：一种是根据用户提供的量化数据集，对加载的浮点模型进行量化，生成量化 RKNN 模型；另一种是由深度学习框架导出量化模型，RKNN-Toolkit 加载并利用已有的量化信息生成量化 RKNN 模型。

根据执行量化操作的时间不同，可以将模型量化分为训练后量化（Post-Training Quantization）和量化感知训练（Quantization-Aware Training, QAT），后训练量化是指将预先训练好的浮点模型转换为定点模型。在后训练量化过程中，首先需要对预训练好的浮点模型进行推理，记录模型中权重和激活值的统计信息，并根据这些信息将浮点数转化成定点数，RKNN-Toolkit 中可以通过调用 config 接口指定特定训练后量化方法与校准数据集。量化感知训练旨在模型训练过程中考虑后续量化操作，这一过程通常会降低训练数据精度，使模型在低精度数据上进行训练，同时对权重和激活值使用动态或静态量化方法。该种量化方法能够让模型在训练过程中尽可能地学习低精度权重和激活值的特征表示，模型。RKNN-Toolkit 目前支持 TensorFlow 和 PyTorch 这两种框架量化感知训练得到的模型。

使用 RKNN-Toolkit 进行模型量化时可能会导致原有模型精度下降，主要可以归因为以下三种情况：一是 RKNN-Toolkit 对该层算子的匹配有问题，或者 NPU 驱动对该算子的实现有问题（此时该层推理结果通常会和浮点模型结果差距巨大，余弦距离小于 0.5）；二是该层数据分布对量化操作不友好（例如数据分布范围比较广，用低精度表达后精度损失严重）；三是 RKNN-Toolkit 或 NPU 驱动的某些图优化可能导致精度下降，例如用 conv 替换 add 或者 average pool 等。对于第一种场景，可以尝试用混合量化的方式进行规避；对于第二种场景，可以尝试用 MMSE 或 KL 散度方法对量化参数进行调优。若还无法改善模型精度，可以考虑使用混合量化，用更高精度方法去计算对量化操作不友好的算子，或者使用量化

感知训练直接得到量化模型，此时 RKNN-Toolkit 将使用模型自身量化参数，几乎不会造成精度损失；对于第三种场景，可以尝试将优化等级下调（设置 config 接口中 optimization\_level 参数）。



## 第 4 章 自主无人机竞速算法边云协同框架研究

### 4.1 边云协同概念阐述

边云协同是指将云端计算和边缘计算相结合，以实现数据处理、分析和决策快速响应与高可靠性的计算模式。在传统的云计算架构中，大多数计算任务都是在云端进行处理，然后将结果返回给终端设备。然而，某些应用场景下（如实时视频流处理、物联网或其他移动设备数据分析等），终端设备所处环境往往网络信号覆盖度低且数据体量大，在此类问题中云计算模式容易产生网络延迟和带宽占用等问题，导致数据实时性与准确性降低，从而影响用户体验。为解决上述问题，边云协同将云计算与边缘计算优势相结合，此时计算资源移动至更靠近数据源的边缘设备，网络延迟大幅降低，能够保证计算结果实时反馈，于此同时也可以借助云端节点的计算与存储资源。例如，在处理视频流任务中，可以考虑将部分计算工作部署与摄像头或视频录像机等边缘设备上，边端设备只需执行图像分割、运动检测等简单计算任务，之后再将预处理后的数据传输至云端进行更复杂和高级的计算（如人脸识别、物体跟踪等）。通过边云协同此类分布式计算模式，可以在保证应用程序准确性与实时性的同时，减少网络带宽和终端响应延迟所带来的影响。与传统的云计算架构不同，边云协同框架需要考虑更多复杂问题，例如，如何协调分布式计算任务，如何处理边缘设备的限制性资源，如何保证传输数据的安全性和隐私性等等。此外，边云协同与人工智能领域联系紧密，两者彼此耦合，协同发展。一方面，人工智能领域可以为边云协同框架中计算卸载、网络服务部署等约束优化问题提供多元解决方案；另一方面，边云协同可以为人工智能技术提供更加广阔的应用平台，如云边大小模型协同推理、边云协同终身学习等。总之，边云协同作为一种新兴计算框架，将云端技术与边缘计算相结合，使得数据处理和分析过程更加实时、高效和安全，在物联网、智能交通、机器人控制等各个领域都有研究与应用价值。

本文第 3 章中提出利用边缘 AI 加速设备能够帮助边端无人机独立推理运行深度学习类自主竞速算法，结合上述陈述，可以考虑将边云协同这一基本构想进一步扩展至自主无人机竞速领域。具体来说，在边云协同框架下，边缘无人机节点可以与云端进行互动，借助云端算力进一步更新完善边缘竞速模型，本文在 4.2.1 中将基于现有边云协同解决方案设计自主无人机竞速框架。

### 4.2 边云协同式无人机自主竞速方法设计

#### 4.2.1 Kubeedge-Sedna 框架介绍

KubeEdge 是华为云原生团队开发的一个开源边缘计算解决方案，基于 Kubernetes (k8s) 开发。Kubernetes 是由 Google 公司发起并维护的开源容器集群管

理系统，提供大量的工具和 API 方便用户调度容器资源以实现各种应用功能。KubeEdge 对 Kubernetes 进行了扩展和优化，将云端容器化应用程序和边缘设备集成在一起，使 Kubernetes 具备在边缘场景下处理问题的能力。KubeEdge 由边缘部分和云端部分组成。对于边缘部分，KubeEdge 支持在多种边缘设备（包括传统 PC 端和各种嵌入式计算平台）上部署 Kubernetes 集群以运行容器化应用程序，并提供连接边缘设备和云端的网关功能，使得云端和边缘设备可以进行通信，并提供针对边缘设备的数据采集和处理功能。对于云端部分，KubeEdge 对原有的 Kubernetes 功能进行二次开发，能够在云端节点实现对整个边缘集群的管理和监控，并支持用户在云端部署和管理应用程序以及对边缘节点进行管理。

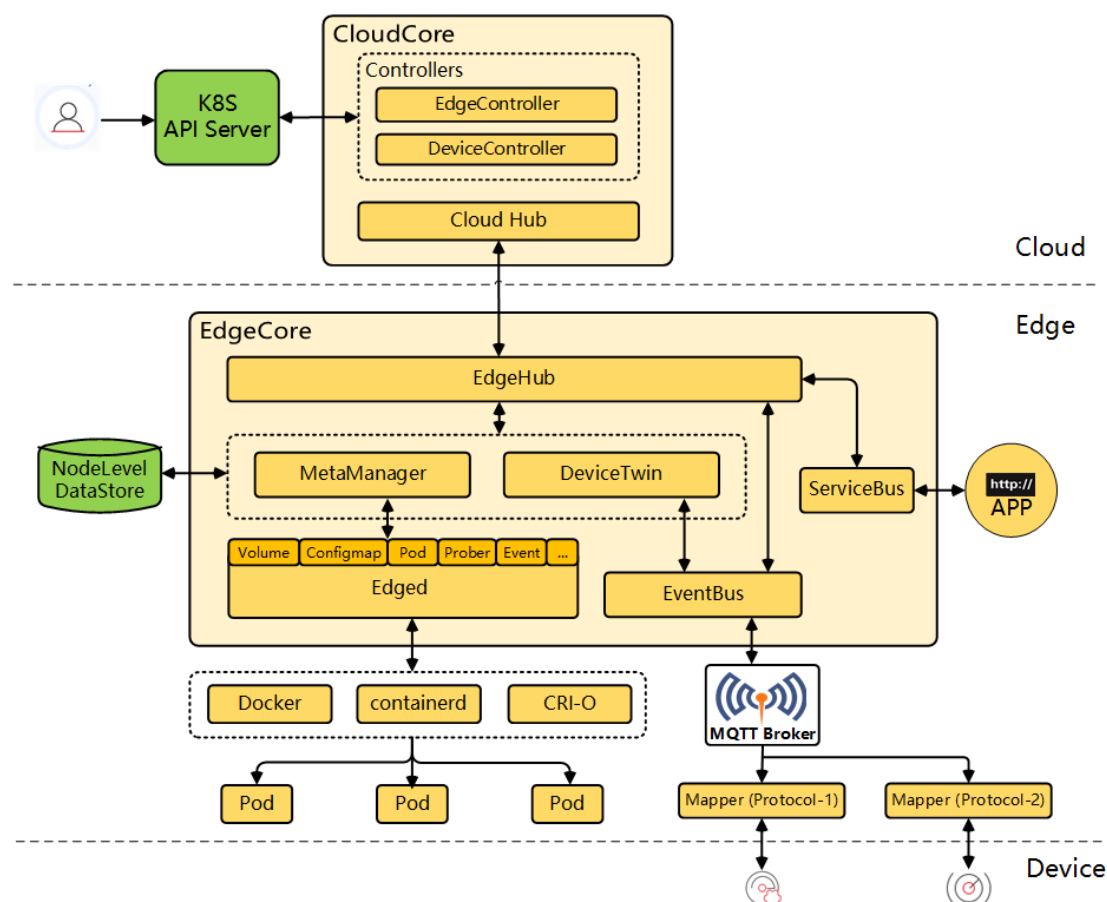


图 4-1 KubeEdge 整体框架

KubeEdge 是一个基于 Kubernetes (k8s) 开发的华为云原生边缘计算解决方案，其整体框架如图 4-1 所示。KubeEdge 并未对 Kubernetes 控制面做出修改，用户在 KubeEdge 中使用 kubectl 等原生 Kubernetes 工具北向调用 k8s API 与在 Kubernetes 中使用体验一致。CloudCore 是 KubeEdge 自开发组件，原生 Kubernetes 中工作节点与 K8s API 之间采用 List-Watch 通讯机制，而由于边缘场景下网络状况并不稳定，边缘节点与云端节点网络连接长时间断开后会触发 List-Watch 机制中 Re-List 操作以保证边云两端数据一致性，该操作会重新获取边端全部资源对象，若此时边缘节点数目较多，便会对云端控制面造成剧烈带宽冲击。基于这一

背景下，Kubedge 利用 CloudCore 实现云边网络增强，即边端不直接与 k8s API 通信，CloudCore 会先调用 k8s API 获取源数据，进一步通过 CloudHub 将数据下发至边端（默认采用 WebSocket 协议），该条云边链路采用轻量化 List-Watch 机制，而是通过一系列消息可靠性方法来记录边云通信状态，从而避免资源信息重复获取。边缘部分主要组间为 EdgeCore，EdgeCore 中 EdgeHub 与上游 CloudHub 进行交互实现云边通信，Edged 是原 Kubernetes 工作节点中 kubelet 组件的简化版本，用于实现边缘 Pod 生命周期管理，支持各类容器运行时接口（CRI）、容器网络接口（CNI）和容器存储接口（CSI）。此外，KubeEdge 边端通过 Mapper 组件能够接入各种物联网终端设备，支持 MQTT 协议与管控面交互。

KubeEdge 是当前最活跃的云原生边缘计算社区之一，吸引了世界各地的用户和开发者参与并付诸实践。2020 年下半年，KubeEdge 成立了 AI SIG(Special Interest Group)，旨在解决边缘人工智能在实现中面临的挑战。KubeEdge AI SIG 将前期边云协同应用于 AI 场景的优秀实践经验予以固化，并于 2020 年 12 月发起了 Sedna 子项目。Sedna 基于 KubeEdge 提供的边云协同能力，实现了人工智能模型跨边云协同训练和协同推理，并支持如 TensorFlow、PyTorch、PaddlePaddle 和 MindSpore 等业界主流人工智能框架，同时 Sedna 也支持现有人工智能应用程序无缝下沉到边缘，快速实现跨边云增量学习、联邦学习和协同推理等功能，最终达成降低模型成本、提升模型性能和保护数据隐私的效果，Sedna 整体框架如图 4-2 所示。

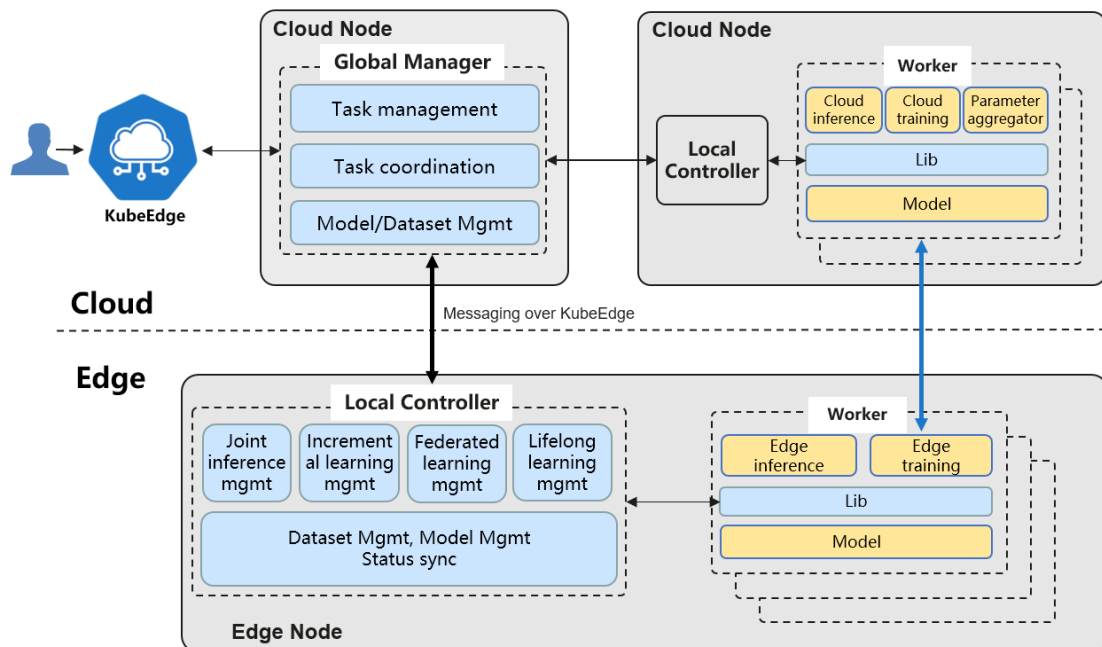


图 4-2 Sedna 整体框架

Global Manager 是基于 Kubernetes Operator 实现的边缘 AI 控制器。用户通过创建 CRD 来实现模型和数据集的管理，以及实现边云协同的 AI 任务管理和跨边

云协调。目前支持以下 AI 任务：增量训练、联邦学习、联合推理和终身学习。其中，增量训练针对小样本和边缘数据异构的问题，模型可以在云端或边缘进行自适应优化，边学边用使模型越来越智能；联邦学习针对数据集规模大、原始数据不出边缘和隐私要求高等困难场景，AI 模型在边缘训练，参数在云端聚合，解决了数据孤岛问题；联合推理针对边缘资源需求大或者边侧资源受限等情况，任务可在云端进行，实现整体推理性能的提升；终身学习旨在解决 AI 应用在边缘落地时所面临的样本基数小、边缘数据异构等难题，通过云端知识库提供记忆功能，使边缘累积样本信息在持续更新的同时实现数据持久化，从而解决传统 AI 模型灾难性遗忘问题。同时，终身学习结合了增量训练和多任务训练的方式来实现跨时间和跨情景的知识迁移，以便更好地处理未知任务。Worker 用于执行训练或推理任务，基于现有 AI 框架开发训练，不同 AI 特性对应不同 Worker 组，Worker 可同时部署在边缘或云端，并实现协同工作。Local Controller 实现增量训练、联邦学习和联合推理模式的本地闭环管理，包括数据集管理、模型管理和 AI 任务状态同步等。Lib 可为应用程序提供边云协同 AI 特性接口，用户可以通过调用 Lib 库来实现边云协同式模型训练、推理和评估。

#### 4.2.2 边云协同式无人机自主竞速方法框架

随着人工智能领域蓬勃发展，深度学习类无人机自主竞速方法展现出了其固有优势，近几年也涌现出大量研究成果，但无人机等边端设备计算资源、电池容量有限，难以推理复杂深度神经网络模型，传统云计算方式并不适用于无人机自主竞速此类延迟敏感应用场景，这导致大多数深度学习类无人机自主竞速方法难以真正落地。而受益于目前人工智能与边缘计算领域双向联动发展趋势，许多轻量级 AI 加速计算平台逐步面世，本文第三章中提出了基于 RK1808 边缘 AI 加速设备模型部署方案设计，帮助边端无人机独立完成深度学习类自主竞速算法推理过程，但正如 1.1 中所述，边缘计算与云计算两者并非是相互割裂，此消彼长的关系，边缘计算与云计算在各自领域都有其不可替代的优越性，在无人机竞速场景下，如何利用好云端资源优势对实现自主竞速方法与人工智能技术深度融合至关重要，具体来说，云端节点往往计算资源丰富，无需考虑功耗限制，网络状况稳定，可以承担起模型训练、模型更新、模型评估等复杂计算任务，而边缘无人机节点只需具备一定模型推理能力，便能在保证指令响应速度的同时不断优化资深网络模型，从而帮助无人机高质量完成自主飞行任务，上述过程中需要边云两端协同互动，完成边缘模型下发、模型更新请求等任务。基于 4.2.1 中内容，借助 KubeEdge 边云协同能力与 Sedna 人工智能特性，本节提出 KubeEdge-Sedna 框架下边云协同式无人机自主竞速方法设计方案。

图 4-3 展示了 KubeEdge-Sedna 框架下边云协同式无人机自主竞速方法框架，无人机边端节点首先会部署一个初始模型，该模型并不要求预测精度与数据集质

量，无人机开始竞速任务后边侧节点推理模块（Inference Worker）将会调用该初始模型生成控制指令。无人机在执行竞速任务期间会碰到难例样本（如无人机在某个障碍门前发生碰撞或者出现路线偏移），具体判断标准可以根据 Sedna Lib 中相关接口进行自定义操作，之后上述难例样本数据将会上传至云端难例存储数据库，进一步，难例数据需要通过外接标注服务或者人工标注方法进行打标。云端 Local Controller 会依据指定时间监测已打标难例样本数，当样本数目达到指定阈值时，云端 Local Controller 会通知 Global Manager 以触发下一轮训练，新模型训练完成后会传递给评估模块（Evaluation Worker），当模型精度满足指定触发策略时，Global Manager 捕获到评估状态信息，会将指定模型下发至边端，并通过边端 Local Controller 通知推理模块进行模型推理，如此重复这一过程，可实现增量学习式模型更新、评估、推理策略。

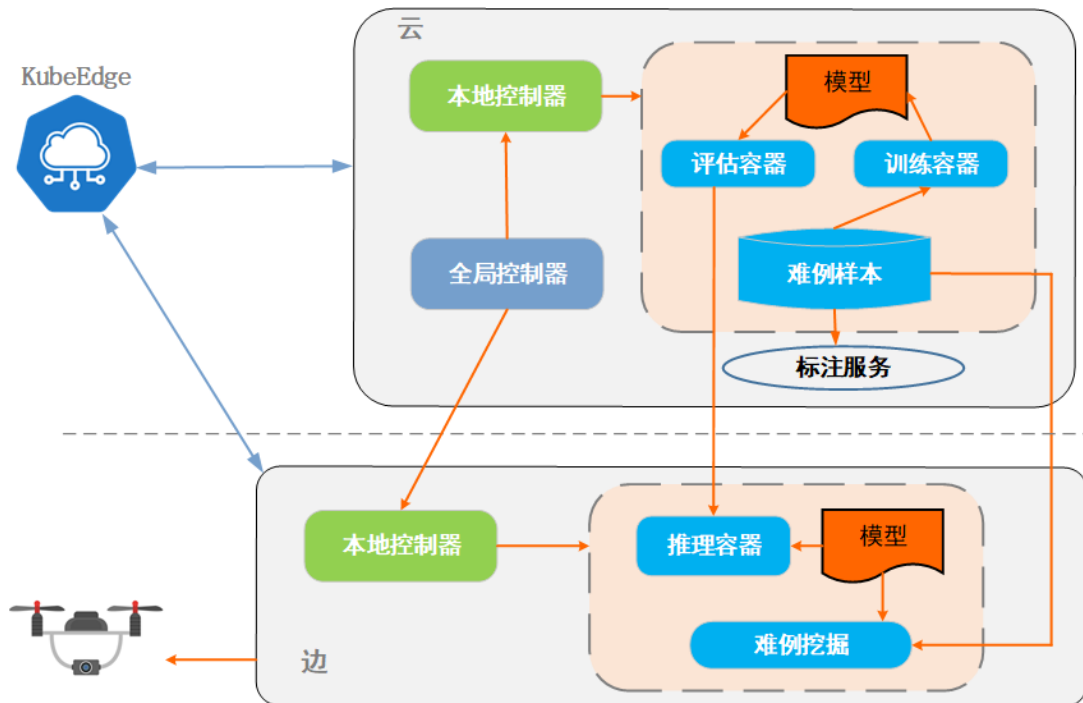


图 4-3 边云协同式无人机自主竞速方法设计方案示意图

基于以上陈述，本文设计方案主要有以下两点优势：

- （1）小样本模型兼容。对于深度学习类无人机自主竞速算法，不论仿真环境或是真实场景中，数据采集采集过程往往十分繁琐耗时，上述设计方案中初始数据集可以只包含少量样本，后续难例数据上传后可以进一步扩充训练样本，提升模型精度。
- （2）支持难例上云。本次方案中，仅将推理输入中难例数据上传至云端，从而降低数据标注工作量，同时能够减小边端与云端数据传输体量，减少网络带宽占用。

## 第 5 章 总结与展望

### 5.1 总结

无人机作为一种常用边缘移动设备，是智能移动机器人领域重点研究对象。自主无人机竞速是推动无人机自主导航领域发展的重要途径之一，近年来随着人工智能技术不断发展，基于深度学习的自主无人机竞速方法展现了其广阔的应用前景，然而受算力与功耗限制，无人机设备通常难以推理深度网络模型，受益于边缘计算领域与边云协同计算框架相关研究不断成熟，深度学习类自主竞速方法得以进一步走向实际应用。基于此，本文围绕自主无人机竞速、边缘模型加速推理、边云协同式无人机竞速方案展开研究，现将工作成果总结如下：

（1）基于 GoogLeNet 框架设计了一种自主无人机竞速方法，旨在实现第一人称视角图像到控制指令的端到端输出，本文所设计网络采用三支结构，网络输入采用 Mosaic 形式，最终预测输出四个控制指令 $(\phi, \theta, \varphi, h)$ ，同时，本文设计采用分组卷积与分组逐点卷积策略进一步减小网络模型参数量与计算量。此外，为解决数据来源问题，本文在 ROS-Gazebo 仿真环境中搭建了两种赛道环境，并围绕 7 种基本动作设计数据采集策略。

（2）为赋予边侧无人机设备模型推理能力，引入 RK1808 人工智能计算板卡卸载无人机机载电脑计算任务，并基于该硬件加速设备设计出一套模型部署方案。

（3）详细阐述了边云协同计算框架对于推进深度学习类无人机自主竞速算法实现落地的深远研究意义，对 KubeEdge-Sedna 边云协同 AI 框架做出详细介绍，并基于该框架设计出一套边云协同式自主无人机竞速方法。

### 5.2 展望

本文设计了一种基于边云协同的自主无人机竞速算法框架，该框架具有较高系统复杂度，涉及无人机控制算法、可用于竞速任务的无人机硬件平台、机载嵌入式计算平台、边云设备协同架构，考虑到时间与工作量限制，本文主要给出设计思路，暂未实现在真实情境下的方法验证，以下列举本次工作存在的不足与未来研究方向：

（1）在无人机自主竞速策略方面，本文基于 GoogLeNet 框架设计了一个深度神经网络结构用于预测控制指令，但该方法目前并未在仿真环境中实施验证；此外，为减小模型复杂度，本文为在网络结构设计中选择直接缩减 Inception 模块数量，但也可以考虑对 Inception 模块本身进行改进，如减小模块中分支数与卷积核数量等。

（2）在边缘模型加速推理方面，本文以 RK1808 人工智能计算板为例，设计了一套网络模型部署方案，但并未对本文所设计无人机自主竞速模型进行部署验证，

此外，除 RK1808 计算办卡外，还可以考虑其他边缘计算硬件平台。

（3）在边云协同式无人机竞速策略方面，本文介绍 KubeEdge-Sedna 框架，并基于此提出了边云协同式自主无人机竞速设计方案，受实验条件限制本文未能对所提方案可行性进行系统验证和分析。此外，就调研结果来看，KubeEdge-Sedna 框架对竞速无人机设备的支持有待改进。

## 参 考 文 献

- [1] Moon H, Martinez-Carranza J, Cieslewski T, et al. Challenges and implemented technologies used in autonomous drone racing[J]. *Intelligent Service Robotics*, 2019, 12(2): 137-148.
- [2] Foehn P, Brescianini D, Kaufmann E, et al. AlphaPilot: autonomous drone racing[J]. *Autonomous Robots*, 2022, 46(1): 307-320.
- [3] De Wagger C, Paredes-Vallés F, Sheth N, et al. The Artificial Intelligence behind the winning entry to the 2019 AI Robotic Racing Competition[J]. *Field Robotics*, 2022, 2(1): 1263-1290.
- [4] microsoft/AirSim-NeurIPS2019-Drone-Racing: Drone Racing @ NeurIPS 2019, built on Microsoft AirSim[EB/OL]. [2023-05-29]. <https://github.com/microsoft/AirSim-NeurIPS2019-Drone-Racing>.
- [5] Hanover D, Loquercio A, Bauersfeld L, et al. Autonomous Drone Racing: A Survey[A]. arXiv, 2023.
- [6] Navardi M, Humes E, Mohsenin T. E2EdgeAI: Energy-Efficient Edge Computing for Deployment of Vision-Based DNNs on Autonomous Tiny Drones[C]//2022 IEEE/ACM 7th Symposium on Edge Computing (SEC). Seattle, WA, USA: IEEE, 2022: 504-509.
- [7] Wang S, Clark R, Wen H, et al. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). 2017: 2043-2050.
- [8] Wang W, Hu Y, Scherer S. TartanVO: A Generalizable Learning-based VO[A]. arXiv, 2020.
- [9] Teed Z, Lipson L, Deng J. Deep Patch Visual Odometry[A]. arXiv, 2022.
- [10] Kaufmann E, Loquercio A, Ranftl R, et al. Deep Drone Racing: Learning Agile Flight in Dynamic Environments[A]. arXiv, 2018.
- [11] Loquercio A, Kaufmann E, Ranftl R, et al. Deep Drone Racing: From Simulation to Reality With Domain Randomization[J]. *IEEE Transactions on Robotics*, 2020, 36(1): 1-14.
- [12] Hwangbo J, Sa I, Siegwart R, et al. Control of a Quadrotor with Reinforcement Learning[J]. *IEEE Robotics and Automation Letters*, 2017, 2(4): 2096-2103.
- [13] Müller M, Li G, Casser V, et al. Learning a Controller Fusion Network by Online Trajectory Filtering for Vision-based UAV Racing[A]. arXiv, 2019.
- [14] Rojas-Perez L O, Martinez-Carranza J. DeepPilot: A CNN for Autonomous Drone Racing[J]. *Sensors*, 2020, 20(16): 4524.
- [15] Fu J, Song Y, Wu Y, et al. Learning Deep Sensorimotor Policies for Vision-based Autonomous Drone Racing[A]. arXiv, 2022.
- [16] Deng S, Zhao H, Fang W, et al. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 7457-7469.
- [17] Rastegari M, Ordonez V, Redmon J, et al. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks[A]. arXiv, 2016.
- [18] Zhang C, Cao Q, Jiang H, et al. FFS-VA: A Fast Filtering System for Large-scale Video Analytics[C]//Proceedings of the 47th International Conference on Parallel Processing. Eugene OR USA: ACM, 2018: 1-10.
- [19] Zhang X, Zhou X, Lin M, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[A]. arXiv, 2017.
- [20] Xiong Y, Sun Y, Xing L, et al. Extend Cloud to Edge with KubeEdge[C]//2018 IEEE/ACM Symposium on Edge Computing (SEC). Seattle, WA, USA: IEEE, 2018: 373-377.
- [21] EdgexFoundry-中文[EB/OL]. [2023-05-18]. <https://cn.edgexfoundry.org/>.
- [22] Zheng Z. KubeEdge-Sedna v0.3: Towards Next-Generation Automatically Customized AI Engineering Scheme[A]. arXiv, 2023.
- [23] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[A]. arXiv, 2014.
- [24] Lin M, Chen Q, Yan S. Network In Network[A]. arXiv, 2014.
- [25] Quigley M, Gerkey B, Conley K, et al. ROS: an open-source Robot Operating System[J].



## 致 谢

首先十分感谢吴华老师对本次毕业设计的悉心指导，毕设一路走来跌跌撞撞，中间经历了很多次沮丧、崩溃的时刻，感谢吴华老师每一次不遗余力的、坚定的鼓励，同时我要感谢 Amaze 实验室所有老师和同学，主 E 8 层的走廊有时候长得走不到尽头，但在电梯转角处的第一扇门背后你可以走进一个温暖的大家庭。

感谢爸爸、妈妈和姐姐一路走来的支持与陪伴，感谢你们包容我的任性与幼稚，二十多年来周围的面孔不断变换，只有你们始终守候，愿你们身体健康，平安顺遂。

此外，我想感谢我的室友们，这四年来没有什么狗血剧情，有的只是大家彼此照顾，相互关心，感谢你们让我看到了女孩子身上的一切美好。

我的四年大学时光掺杂于千百万的人毕业故事中，同它们一起在这个夏日终结。未来的日子，不要踟躇，停止抱歉，快快长大吧。