中国矿业大学
CHINA UNIVERSITY OF MINING AND TECHNOLOGY

智能机器人创新设计与实践报告

题目：基于深度学习的香包图案生成、风格迁移与文生图系统

姓　名：杨晓琦

学　号：08222213

任课教师：王冠军　副教授

中国矿业大学

2025 年 1 月

目录

摘　要

作业 1

作业 1 是首先检索一篇近几年高引用文献，《多无人机协同编队飞行控制研究现状及发展》该文献综述了多无人机协同编队技术的现状与进展，重点分析了任务分配、航迹规划、编队控制等方面的研究进展。然后对**主要研究方法与内容**进行了分析总结，其中，**任务规划**研究了基于动态规划、博弈论和粒子群算法等方法来优化多无人机的任务分配。**航迹规划**提出了不同算法来优化多无人机编队在复杂环境下的路径选择，保证飞行效率和安全。**编队控制**包括队形设计、调整与飞行控制方法，研究了 leader-follower、行为法、虚拟结构法等多种控制策略。并对文献相关内容发表个人看法。

作业 2

作业 2 是设计一个**月球多机器人协作系统**，结合月球表面的环境特点以及基地建设的需求，设计了一组不同功能的机器人，将多种技术（如感知、运动、通信和能源系统）与不同类型的机器人结合，它们将在月球基地中承担不同任务，协助完成月球表面的资源开采、基地建设、环境监测、物资运输等任务。

机器人系统包括“**勘探型机器人**”、“**搬运型机器人**”、“**维护型机器人**”和“**监测型机器人**”四种主要类别，其中勘探型机器人负责对月球表面进行地形勘测、资源探索和分析，它使用高精度激光雷达（LiDAR）和摄像头获取月球的 3D 地图，同时利用矿物探测器、土壤分析仪等设备分析月球土壤成分和矿产资源；搬运型机器人负责月球基地建设中的物资运输，尤其是大宗建材（如砖块、石料等）的搬运任务，它需要具备较强的负重能力和灵活的工作能力；维护型机器人负责月球基地设施的日常维护与修复工作，包括清洁太阳能电池板、机械设备检查、故障修复等；监测型机器人负责对月球环境进行实时监测，包括温度、辐射、气体含量等参数，确保基地环境的安全。

作业 3

作业 3 是围绕**香包图案**设计一个**系统**，可实现自动生成关于香包的图案、指定风格应用于现有的香包图案实现风格迁移以及根据文本描述自动生成香包图案等 **3 种功能**。

对于**图像生成**，本文选择 **DCGAN（深度卷积生成对抗网络）**,通过训练生成新的香包图案，利用 GAN 的生成器部分，学习香包图案的特征，并生成创意设计，来自动生成具有不同风格、颜色或形状的香包图案；对于**图像风格迁移**，本文使用基于卷积神经网络的**风格迁移**技术使用 **AdaIN**（自适应实例归一化），将某种艺术风格（如油画、复古风格）迁移到香包图案上，生成新的香包图案，保持原始结构同时应用新的视觉风格；对于**文生图**，使用**文本到图像生成**技术以及 **Stable Diffusion** 模型，结合香包图案数据集进行微调，根据文本输入（如“红色花卉香包”或“复古风格香包”）生成符合描述的香包图案，能够生成多样化的香包设计。

关键词： 图像自动生成 风格迁移 文生图 深度卷积生成对抗网络(DCGAN) AdaIN(自适应实例归一化) VGG16 模型 Stable Diffusion 模型

作业 1

1. 选题

文献题目：《多无人机协同编队飞行控制研究现状及发展》
文献作者：宗群 王丹 邵士凯 张博渊 韩宇
发表期刊：哈尔滨工业大学学报 第 49 卷 第 3 期

2. 文献主要研究方法与研究内容

这篇文献主要采用**综述**和**分析**的研究方法，**首先**对国内外多无人机编队相关技术的**现状和进展**进行综述,**然后**重点对多**无人机编队控制方法**进行分析,并对队形设计、队形调整和队形重构等问题进行归纳总结,**最后**对多无人机协同编队所面临的**机遇和挑战**进行了展望。

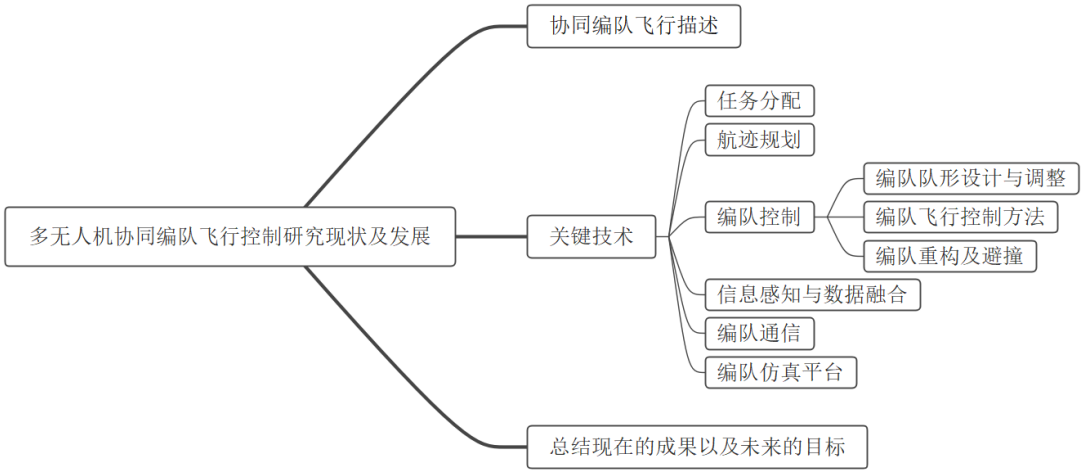


图 1-1 文献主要内容框架

3. 详细解读分析

这篇文献对无人机协同编队飞行中所涉及的**关键技术**描述的非常详细与清楚，主要是目前国内外已经取得的突破性成果以及所设计的方法所以我在这部分重点梳理并且结合个人观点分析一下。(每点都是精华，个人觉得值得精度，受益匪浅)

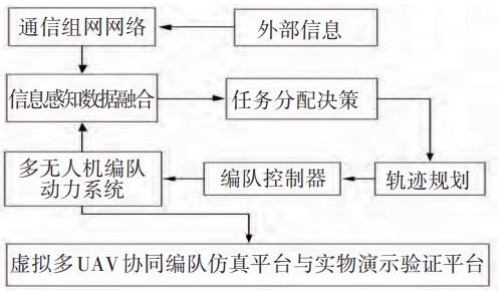


图 1-2 多无人机协同编队飞行控制中各项关键技术之间的关系

3.1 任务分配

对于任务分配这一部分，这篇文献主要对目前国内外对于无人机编队任务分配的研究进展进行了总结，主要有：Mcgrew 等人（2010）使用**近似动态规划技术**解决固定速度一对一作战机动问题，并通过室内飞行试验验证了算法的快速响应能力；Sujit 等人（2011）应用**博弈论**方法解决了两个无人机在未知区域协同搜索的问题，考虑了通信和传感器范围、油量限制等实际约束，提高了搜索效率；Wei 等人（2015）提出了一种**双级任务分配**方法，改善了传统粒子群算法易陷入局部极小值的问题，提升了任务分配的可靠性、精度和速度。

这些关于任务分配的研究共同推动了无人机编队任务分配技术的发展，从算法创新到实际应用，都为提高无人机在复杂环境中的作业能力和效率提供了坚实的基础。

3.2 航迹规划

为了确保复杂环境中多无人机编队能够安全、快速到达任务区域，降低被敌方雷达捕获、摧毁的概率，需要设计满足一定约束条件及性能指标最优的编队航迹。

因此对于航迹规划这部分，该文献也总结了国内外目前已经拥有的研究成果。Kothari 等人（2013）采用**机会约束方法**结合**快速随机搜索树算法**，解决了多无人机系统中环境不确定性的问题，实现了鲁棒最优路径规划；Shorakaei 等（2014）在研究多无人机协同搜索时，考虑了编队避撞作为性能指标，采用基于概率的**环境建模**和**平行遗传算法**设计二维及三维最优路径； Berger 等（2016）针对异构飞行器静态目标搜索，提出了一种新的**整数线性**和**二次规划模型**，降低了计算复杂度，用线性规划算法获得近似最优解，适用于异构飞行器；梁宵等（2016）为解决复杂环境下移动目标的路径跟踪问题，结合滚**动时域优化**与**人工势场法**，实时生成针对移动目标的最优轨迹。

3.3 编队控制

该文献这部分分析的多，我对内容整理了一下并写一下自己的看法。

3.3.1 编队队形设计与调整

编队队形设计与调整主要包括两部分，首先进行**队形设计**，然后对队形进行**动态调整**。

队形对于无人机协同作战是至关重要的，设计一个合理有效的队形可以延长无人机编队飞行距离、节省燃料消耗、增加编队灵活性，这可以大大提高安全性与任务完成率。我们常见的队形有楔队、梯队、横队、纵队和 V 形等。对于 **V 字形**已经有研究指出最高可节省 12% 的料，这种编队模式用于跨洋飞行或长途飞行的客机编队，大大延长了飞行距离并节省燃料消耗；对于**圆形编队队形**，如下图所示，不仅能扩大探测半径，还能有效提高探测资源区域的效率，有效地完成了多无人机协同探测任务。在实际作战中其实是根据不同的需求来进行变换达到不同的队形。

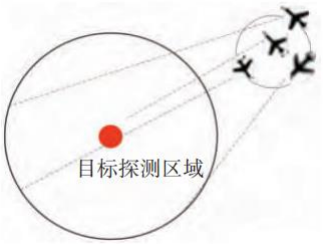


图 1-3 圆形编队

对于队形的动态调整，就是在面对复杂环境及任务的突然变化，能够尽可能短时间快速地生成各项性能指标最优的队形。一个恰当合理的队形变换方案能够提高燃料使用效率、灵活应对突发情况，实现编队的安全快速飞行。这部分文献给大家讲述了 2013 年国加利福尼亚大学 Richert 等针对一对无人机中“Leader”角色的合理配置问题，提出一种**分区协同算法**，

计算出了每架无人机作为“Leader”角色的行程区间如图所示。该算法有效解决了编队队形的动态调整问题，并最小化 leader-follower 角色的转化次数，使得整体编队燃料消耗与单架无人机燃料消耗的同时最少。

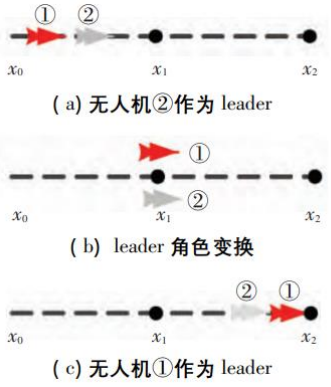


图 1-4 无人机 Leader 变换

3.3.2 编队飞行控制方法(重点)

在多无人机编队执行侦察和防御等任务时，需要多无人机保持一定队形编队飞行到任务执行区域，这部分主要针对不同的实际环境讲了几种方法。

leader-follower 法：Leader-follower 方法是目前多无人机编队控制中最常用的方法之一。leader 跟踪一个预先给定的轨迹，follower 和 leader 轨迹保持一定构型，并速度达到一致。leader 可以看成是目标追踪的对象，或是整个多智能体的共同利益。这种方法将编队问题转化为了经典控制理论中的误差跟踪问题，具有较强的扩展性，而且该方法节能量、减少通信花费、增强群体通信及保证群体方向。

基于行为法：这种编队方法是定义无人机的几种基本控制行为，如跟随、避障和队形构成等，对定义的几种行为进行加权得到编队控制方法。这使系统中的每个单体都具备依据自身决策来协同其他单体完成目标或任务的能力。比如 2015 年，北京航空航天大学提出了一种基于鸽群特性的编队控制方法，这种方法利用图论和势场函数理论对编队中的拓扑结构和群体中的主从关系进行定义，实现了对无人机紧密编队飞行的仿真。

虚拟结构法：这种方法是美国加利福尼亚大学提出的一种集中式控制方法。如下图所示，将编队作为一个虚拟刚体，在编队中设定一个虚拟长机或虚拟几何中心，队中所有无人机都参照虚拟长机或虚拟几何中心运动。

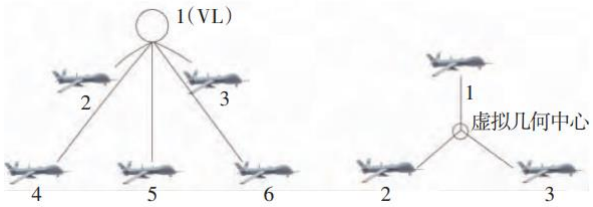


图 1-5 虚拟编队结构图

图论法：它利用拓扑图上的顶点来描述单个无人机，两点之间的边用来表述无人机间的关联/约束拓扑关系，例如感知、通信或控制连接等，将控制理论引入图中，可以构建编队控制策略。

3.3.3 编队重构及避撞

这部分主要包括队形切换及缺少一架或多架无人机时新编队队形的重构，在队形重构过程中必须考虑机间避撞。例如，多无人机编队飞行执行任务时，需要规避雷达、电磁干扰、敌机和较大障碍物，变换合适的队形可以增加任务完成率。实现无人机编队重构的方法有:势能域函数方法;滚动时域法;模型预测法;生物算法;最优控制法。(这部分没有展开细讲..相当

于给大家科普一下)

3.4 信息感知与数据融合

无人机可以利用机载传感器如红外探测仪、摄像机和雷达对周围空地环境进行探测，实现环境感知，并通过编队内部感知能力保持队形，增强协同编队飞行的安全性和可靠性。无人机之间共享感知信息，通过数据融合技术实现协同感知，扩大探测范围并提高环境信息的精确性和全面性，从而有效完成侦察等任务(如下图所示)。

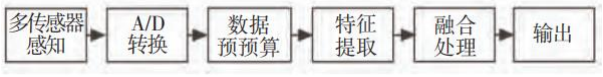
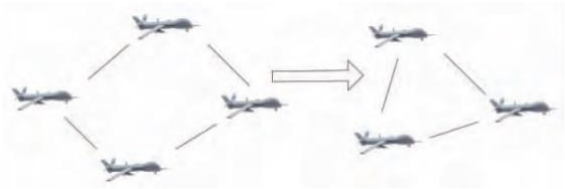


图 1-6 多传感器数据融合过程

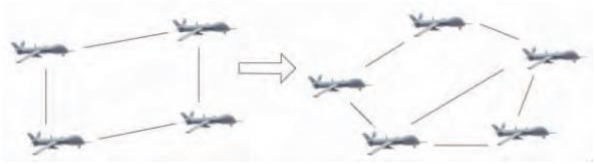
针对多源异质传感器信息融合这一关键问题，文献同样提供了一些研究者们提出了多种解决方案：Lima 等人（2007）采用贝叶斯方法解决自主传感器和机器人网络的目标协同定位问题，提高了观测目标定位的精度。Lee（2008）提出了一种新的不确定信息滤波器算法，利用统计线性误差传播方法处理不确定数据，评估未知信息的不确定性程度。王林等（2010）基于无色变换、交互多模型和信息滤波算法，开发了一种分布式融合估计方法，专门面向多无人机协同感知，该方法保证了更高的估计精度和融合性能。

3.5 编队通信

多无人机通信组网的思想是无人机不完全依赖地面站或卫星等设施的控制，将所有无人机看作一个整体，在多无人机间建立一个无线通信网络，各无人机间相互配合，相互转发指令、交换信息。该网络打破了无人机之间没有任何联系与合作的传统作战思想，可以提高无人机的综合作战能力，减小作战能耗。下图是一种典型的星型拓扑结构。



(a) 通信拓扑变化 1



(b) 通信拓扑变化 2

图 1-7 多无人机网络拓扑变化

3.6 编队仿真平台

搭建满足多无人机协同编队仿真的多无人机仿真平台，对于加快开发周期，降低多无人机编队试验成本，具有十分重要的意义，目前对于无人机仿真平台的研究，主要有 4 类，单系统仿真平台；基于 HLA(high level architecture) 架构的分布式系统仿真平台；自主开发的多飞行器编队分布式虚拟系统仿真平台；自主开发的多无人机编队分布式实物系统仿真平台。我用下面这张图图展示一下仿真平台的具体结构。

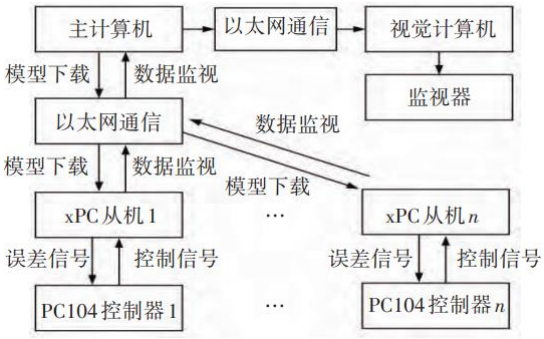


图 1-8 多无人机虚拟仿真平台结构

4. 个人感悟

随着全球安全形势的复杂化，无人机技术真的太重要了!!! 我们可以直观的体会到这种技术在军事、民用和科技发展等领域影响之深远。因此，从一名大学生的角度，要想为我国无人机技术贡献自己的力量，我认为必须首先要**强化数学和物理基础**，因为多无人机系统的运行依赖于复杂的算法和模型，因此扎实的线性代数、微积分、概率论等是力学和电磁学知识是必不可少的。除此之外，作为一名计科学生，还必须**掌握编程技能**，Python、MATLAB 等编程语言来实现无人机相关算法。

希望自己可以更加耐心，扎实走好每一步，不畏惧算法的困难性，可以学以致用，来为祖国无人机事业贡献自己的一份力量!!!

作业 2

1. 方案背景

随着中国探索月球的进一步深入，未来的月球基地建设需要依赖智能机器人来协助完成月球表面的资源开采、基地建设、环境监测、物资运输等任务。为了实现这一目标，本方案设计了一套多机器人协作系统，利用机器人感知、运动、通信系统等的协同作用，确保任务的高效、精准和安全。

2. 方案概述

本设计方案提出了一个多机器人协作系统，结合月球表面的环境特点以及基地建设的需求，设计了一组不同功能的机器人。机器人系统包括“勘探型机器人”、“搬运型机器人”、“维护型机器人”和“监测型机器人”四种主要类别。它们将在月球基地中承担不同任务，并通过协同工作提升任务效率。

3. 机器人主要系统组成

3.1 感知系统

每个机器人都配备了多种传感器和摄像头，来感知月球环境和周围的情况。传感器主要包括：

激光雷达（LiDAR）：用于创建月球表面的 3D 地图，帮助机器人避障和导航。

红外传感器：用于在低光环境中进行探测，确保机器人的导航精度。

摄像头：用于视觉识别、目标跟踪、地形分析等。

气体传感器：用于监测月球环境的气体变化（如月尘）。

温度传感器：监测机器人及其工作环境的温度，避免因温差过大导致故障。

3.2 运动系统

月球表面重力较低，约为地球的 1/6，运动系统应当考虑低重力条件下的适应性：

四轮驱动系统：采用大轮径、宽轮胎设计，确保机器人能稳定行走在月球的松软或岩石表面。

多自由度关节：为机械臂提供更高的灵活性，适应不同任务（如拾取、装载等）的需求。

履带系统：对一些特定的机器人（如勘探型机器人或维修型机器人）采用履带驱动，提供更强的地面附着力，适应复杂地形。

3.3 通信系统

由于月球与地球的距离较远，通信延迟不可避免，因此需要建立可靠的本地通信网络。

地面中继站与卫星链路：通过建立月球轨道卫星链路，保持月球与地球之间的实时通信。

局域通信网络：通过高频率的短距离无线通信系统（例如 Wi-Fi 或 5G），实现机器人之间的互联互通和数据共享。

3.4 能源系统

月球上缺乏大气层和液态水，能量来源有限，因此机器人必须依靠高效的能源系统。

太阳能电池板：机器人配备可展开的太阳能电池板，在月球白昼期间进行充电。

锂电池储能：高效的电池系统用于储存能源，确保机器人在月夜期间继续工作。
热电发电系统：利用月球温差（白天和夜晚的温度差异）进行发电，以提供持续的电力支持。

4. 机器人类型与功能设计

4.1 勘探型机器人

功能：主要负责月球表面地形、矿产、资源的勘探与分析。搭载高精度的激光雷达与摄像系统，分析月球土壤成分、测量地质活动等。
运动方式：四轮驱动，配备高性能机械臂用于抓取样本。
特殊装备：包括激光扫描仪、矿物探测器、土壤分析仪、3D 地图绘制系统。

4.2 搬运型机器人

功能：负责月球基地建设中物资的搬运与转运，特别是大宗建材的运输（如砖块、石料等）。
运动方式：履带式移动，具备较强的负重能力。
特殊装备：搭载大容量货舱、机械臂用于搬运重物。

4.3 维护型机器人

功能：负责基地设施的日常维护与修复，包括太阳能电池板的清洁、机械设备的检查、故障修复等。
运动方式：小型四轮移动平台，便于在狭窄区域操作。
特殊装备：集成多种维修工具，如焊接设备、螺丝刀、激光清洁装置等。

4.4 监测型机器人

功能：负责环境监测，包括月球表面温度、辐射、气体含量等重要参数的实时监测。
运动方式：类似巡逻型机器人，定期在基地周围移动。
特殊装备：辐射监测仪、温湿度传感器、气体检测仪。

5. 协作机制

信息共享与环境感知：所有机器人都配备了先进的感知系统，包括激光雷达、摄像头和其他传感器。它们能够创建高分辨率的 3D 地图，并实时监测周围的环境变化。勘探型机器人负责探索新的区域，收集关于地形、矿物分布等信息。它会将获得的数据发送给中央调度系统以及其它需要这些信息的机器人。监测型机器人持续监控基地及其周边的环境参数，如温度、辐射水平和气体含量。如果发现异常情况，它会立即向中央调度系统报告，以便及时采取措施。
任务规划与分配：中央调度系统根据接收到的信息和当前的任务需求，智能地规划每个机器人的工作路径和任务优先级。例如，当勘探型机器人发现了潜在有价值的资源后，中央调度系统可以快速评估并派遣搬运型机器人前往该地点进行样本采集或物资运输。如果维护型机器人检测到某项设备出现故障或者需要定期保养，它会向中央调度系统发出请求，然后由系统安排最合适的时机进行维修作业。
动态协作与实时响应：在执行任务过程中，机器人之间可以通过局域通信网络保持密切联系，实现动态协作。比如，在进行大规模建筑材料的运输时，多个搬运型机器人可以同时行动，形成一个临时的物流链，提高工作效率。如果监测型机器人检测到某个区域存在安全隐患（如高辐射区），它可以即时通知正在接近该区域的其他机器人，引导它们绕行或者暂停操作，

从而保障所有机器人的安全。

能源管理与支持：为了保证长时间稳定运行，各类型机器人均采用了高效的能源管理系统。太阳能电池板和热电发电系统为机器人提供了必要的电力补充。维护型机器人还可以协助检查和维护其他机器人的能源系统，如清洁太阳能电池板上的灰尘，确保其效率最大化。

数据整合与决策支持：收集到的所有数据都会被上传至中央数据库，供科学家和技术人员分析使用。这不仅有助于优化现有任务流程，也为未来的任务规划提供了宝贵的经验参考。通过对历史数据的学习，中央调度系统可以更好地预测可能遇到的问题，并提前做出相应的调整，进一步提升整个多机器人系统的智能化水平。

6. 设计图

图 2-1 月球机器人系统架构设计图

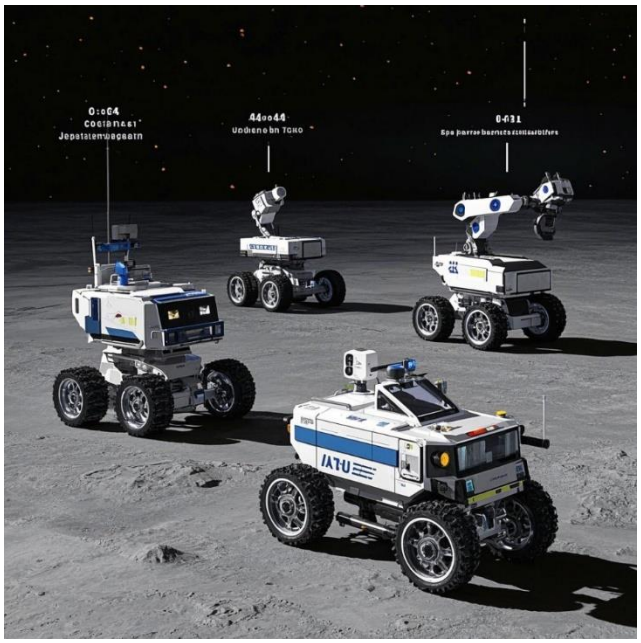


图 2-2 AI 生成四种机器人协作完成任务

勘探型机器人：四轮驱动，带有可伸缩机械臂、激光雷达、摄像头和土壤分析装置。

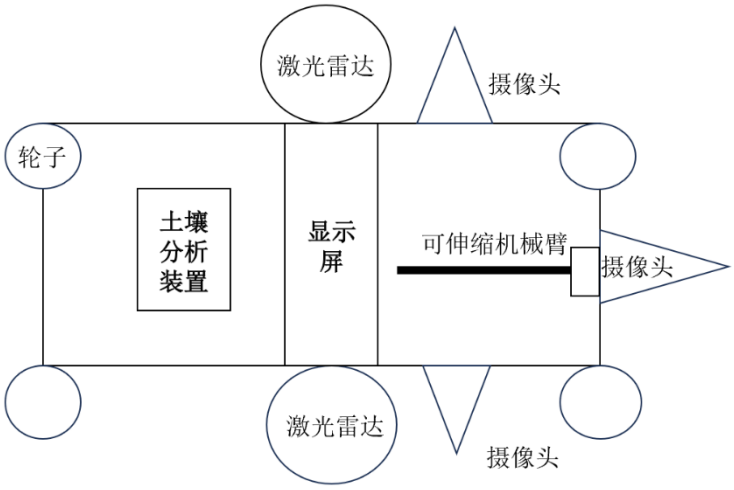


图 2-3 勘探型机器人平面设计图

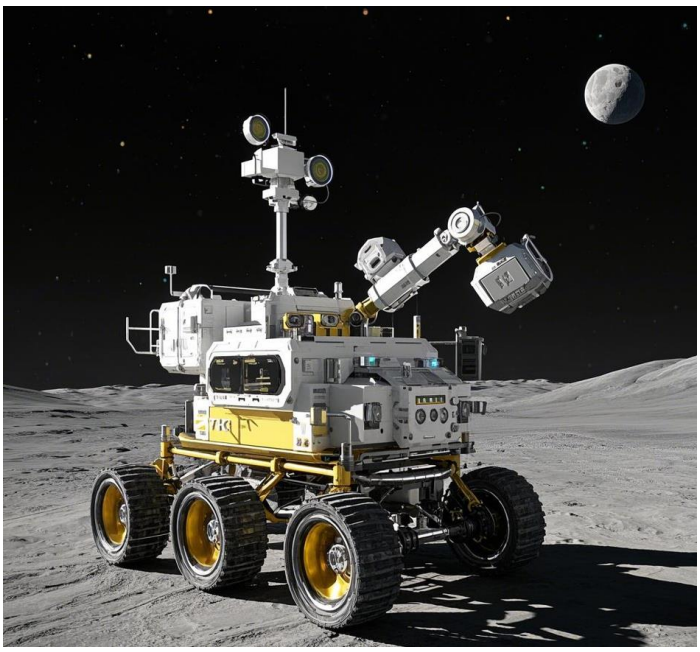


图 2-4 AI 生成勘探型机器人

搬运型机器人：履带式平台，带有大型货舱和多个机械臂，能够快速卸货和搬运重物。

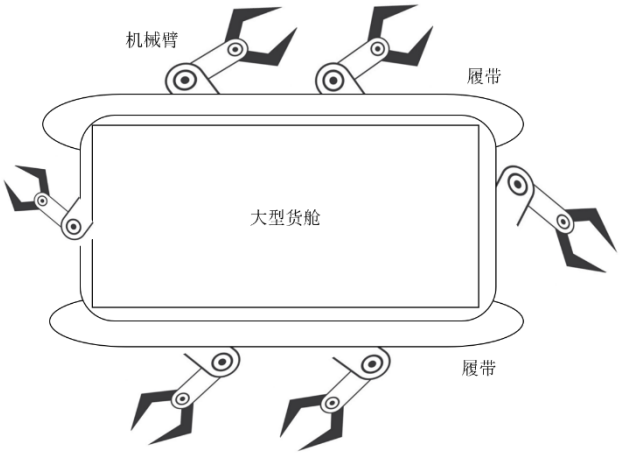


图 2-5 搬运型机器人平面设计图

图 2-6　AI 生成搬运型机器人

维护型机器人：小型四轮，配备维修工具和摄像头，灵活可达各类基地设施。

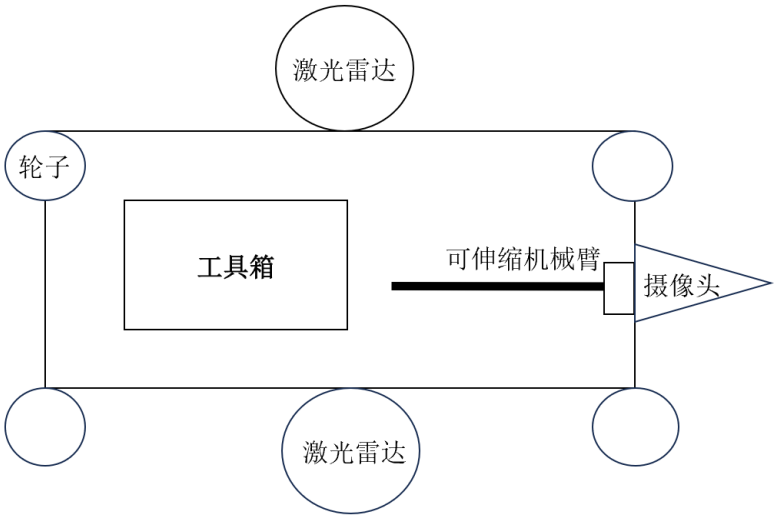


图 2-7 维护性机器人平面设计图

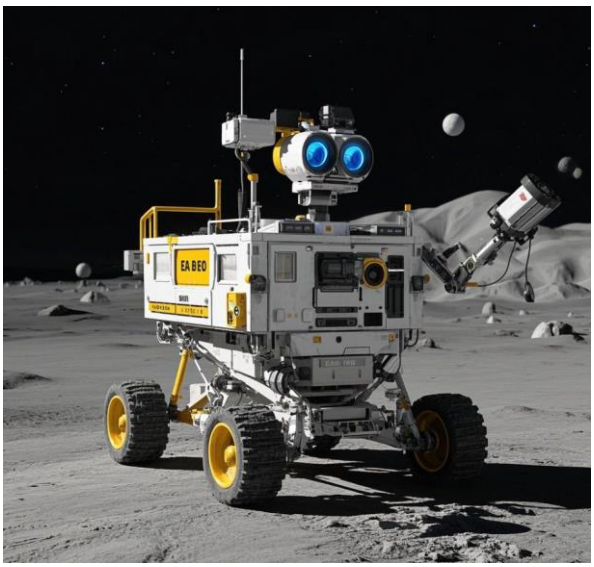


图 2-8　AI 生成维护型机器人

监测型机器人：类似巡逻机器人，多足，行动快，配备各种传感器用于环境监测，具有高精度定位能力。

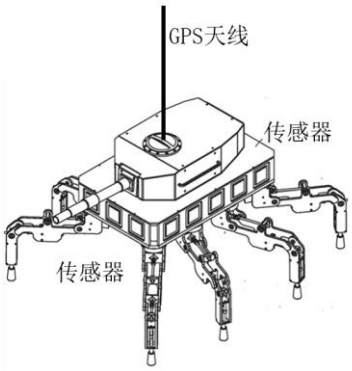


图 2-9 检测型机器人 3D 设计图

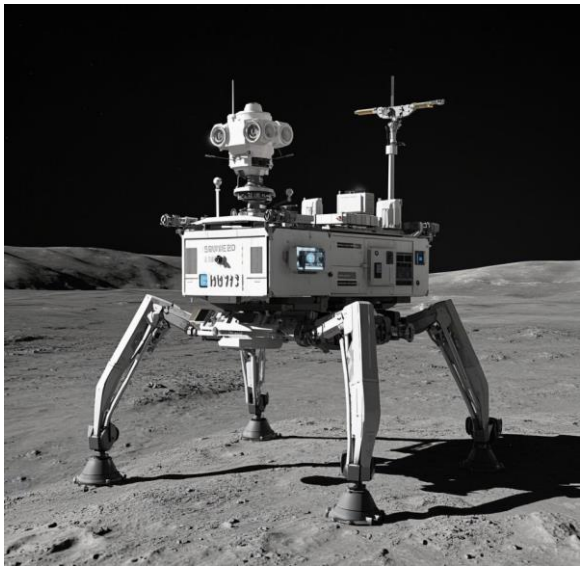


图 2-10　AI 生成监测型机器人

作业 3

1. 开发环境、开发平台与开发框架

开发环境：Python3.7
开发平台：PyCharm
开发框架：PyTorch

2. UI 界面设计

图 3-1 系统首页

图 3-2 风格迁移界面

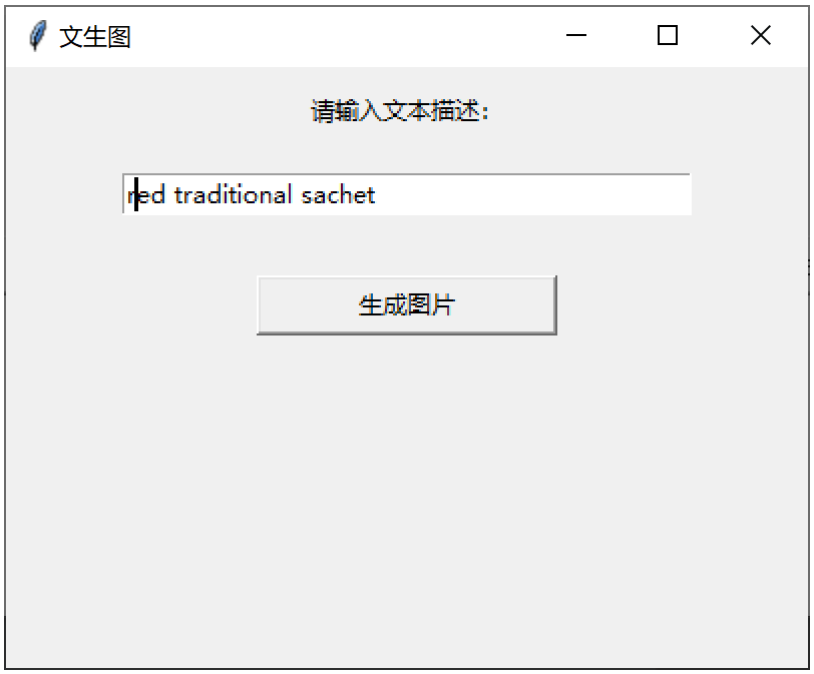


图 3-3 文生图界面

3. 关键代码

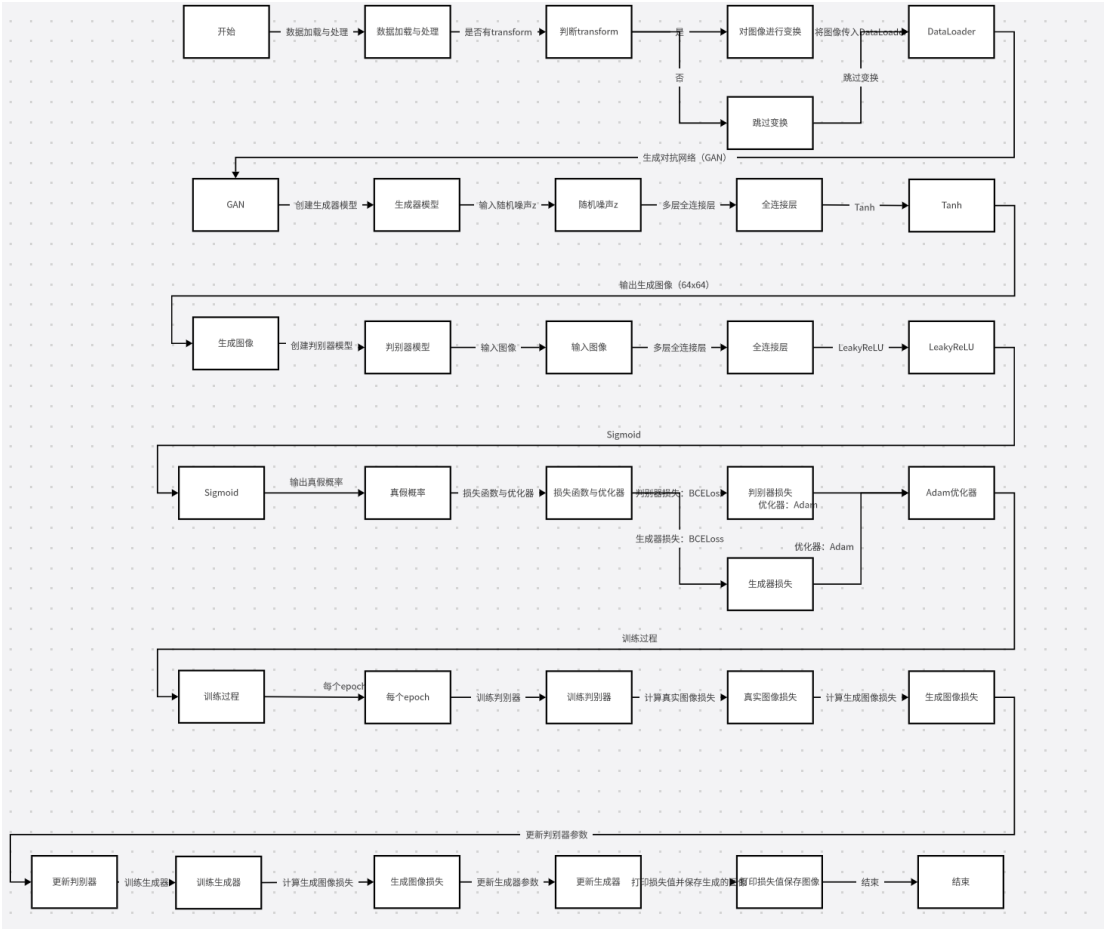


图 3-4 算法框架

3.1 数据加载与处理

将这些图像数据加载并转换为合适的格式

```
class FloralDataset(Dataset):
    def __init__(self, root_dir, transform=None):
        self.root_dir = root_dir
        self.transform = transform
        self.image_paths = [os.path.join(root_dir, f) for f in os.listdir(root_dir) if f.endswith('.jpg')]

    def __len__(self):
        return len(self.image_paths)

    def __getitem__(self, idx):
        img_path = self.image_paths[idx]
        image = Image.open(img_path)
        if self.transform:
            image = self.transform(image)
        return image

transform = transforms.Compose([
    transforms.Resize(64),
    transforms.ToTensor(),
    transforms.Normalize([0.5], [0.5])
])

dataset = FloralDataset(root_dir='植物纹样共 330 个', transform=transform)
dataloader = DataLoader(dataset, batch_size=32, shuffle=True)
```

3.2 自动生成香包图案的核心逻辑

生成对抗网络（GAN）包含一个生成器（Generator）和一个判别器（Discriminator）。生成器生成新的香包图案，判别器判断图像是否为真实图像。

3.2.1 生成器模型

```
class Generator(nn.Module):
    def __init__(self, z_dim):
        super(Generator, self).__init__()
        self.fc1 = nn.Linear(z_dim, 256)
        self.fc2 = nn.Linear(256, 512)
        self.fc3 = nn.Linear(512, 1024)
        self.fc4 = nn.Linear(1024, 3 * 64 * 64)
        self.tanh = nn.Tanh()
```

```
    def forward(self, z):
        x = torch.relu(self.fc1(z))
        x = torch.relu(self.fc2(x))
        x = torch.relu(self.fc3(x))
        x = self.fc4(x)
        x = self.tanh(x)
        return x.view(-1, 3, 64, 64)
```

3.2.2 判别器模型

```
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.fc1 = nn.Linear(3 * 64 * 64, 1024)
        self.fc2 = nn.Linear(1024, 512)
        self.fc3 = nn.Linear(512, 256)
        self.fc4 = nn.Linear(256, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = x.view(x.size(0), -1)    # Flatten the image
        x = torch.leaky_relu(self.fc1(x), 0.2)
        x = torch.leaky_relu(self.fc2(x), 0.2)
        x = torch.leaky_relu(self.fc3(x), 0.2)
        x = self.fc4(x)
        return self.sigmoid(x)
```

3.2.3 损失函数与优化器

```
# 定义优化器
lr = 0.0002
beta1 = 0.5
beta2 = 0.999

generator = Generator(z_dim=100)
discriminator = Discriminator()

optimizer_g = optim.Adam(generator.parameters(), lr=lr, betas=(beta1, beta2))
optimizer_d = optim.Adam(discriminator.parameters(), lr=lr, betas=(beta1, beta2))

# 使用二元交叉熵损失
criterion = nn.BCELoss()
```

3.2.4 训练过程

训练过程包含生成器和判别器交替训练。生成器生成图像，判别器判断图像真假，并通过损失函数优化。

```
# 训练过程
epochs = 50
for epoch in range(epochs):
    for i, real_images in enumerate(dataloader):
        batch_size = real_images.size(0)

        # 真实标签和假标签
        real_labels = torch.ones(batch_size, 1)
        fake_labels = torch.zeros(batch_size, 1)

        # 训练判别器
        optimizer_d.zero_grad()

        real_images = real_images.cuda()
        outputs = discriminator(real_images)
        d_loss_real = criterion(outputs, real_labels)

        z = torch.randn(batch_size, 100).cuda()
        fake_images = generator(z)
        outputs = discriminator(fake_images.detach())
        d_loss_fake = criterion(outputs, fake_labels)

        # 总判别器损失
        d_loss = d_loss_real + d_loss_fake
        d_loss.backward()
        optimizer_d.step()

        # 训练生成器
        optimizer_g.zero_grad()
        outputs = discriminator(fake_images)
        g_loss = criterion(outputs, real_labels)
        g_loss.backward()
        optimizer_g.step()

    print(f'Epoch [{epoch+1}/{epochs}], D Loss: {d_loss.item()}, G Loss: {g_loss.item()}')
    if (epoch + 1) % 10 == 0:
        save_image(fake_images, f"generated_images_epoch_{epoch+1}.png", nrow=4,
normalize=True)
```

3.3 实现风格迁移的核心逻辑

3.3.1 提取图片特征

```
def get_features(image, model, layers=None):
    if layers is None:
        layers = {
            '0': 'conv1_1',
            '5': 'conv2_1',
            '10': 'conv3_1',
            '19': 'conv4_1',
            '21': 'conv4_2',
            '28': 'conv5_1',
        }

    features = {}
    x = image
    for name, layer in model._modules.items():
        x = layer(x)
        if name in layers:
            features[layers[name]] = x
    return features

def load_vgg19_model(device='cuda'):
    vgg = models.vgg19(pretrained=True).features.to(device).eval()
    return vgg
```

3.3.2 不断训练，减少损失

```
def run_style_transfer(content_img, style_img, model, num_steps=500,
style_weight=1000000, content_weight=1, device='cuda'):
    content_features = get_features(content_img, model)
    style_features = get_features(style_img, model)
    generated_img = content_img.clone().requires_grad_(True).to(device)

    optimizer = optim.LBFGS([generated_img])

    for step in range(num_steps):
        def closure():
            generated_img.data.clamp_(0, 1)

            # 计算特征
            generated_features = get_features(generated_img, model)
```

```
                content_loss=compute_content_loss(content_features['conv4_2'],
generated_features['conv4_2'])


                style_loss = 0
                for layer in style_features:
                    style_loss+=compute_style_loss(style_features[layer],
generated_features[layer])

                # 总损失
                total_loss = content_weight * content_loss + style_weight * style_loss

                optimizer.zero_grad()
                total_loss.backward()

                return total_loss

            optimizer.step(closure)

            if step % 50 == 0:
                print(f'Step {step}/{num_steps}, Loss: {total_loss.item()}')
                imshow(generated_img, title=f'Step {step}')

        generated_img.data.clamp_(0, 1)
        return generated_img
```

3.3.3 输出图片

```
def main():
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

    content_img = load_image('植物纹样共 330 个/1-10 曹晓雅/图片 4.jpg', device=device)
    style_img = load_image('img.png', device=device)

    vgg_model = load_vgg19_model(device=device)

    output = run_style_transfer(content_img, style_img, vgg_model, num_steps=500,
style_weight=1000000, content_weight=1, device=device)

    imshow(output, title="Generated Image")

if __name__ == "__main__":
    main()
```

3.4 文生图的核心逻辑

3.4.1 加载 Stable Diffusion 模型和生成图像的逻辑

```
class TextToImageModel:
    def __init__(self, model_name="CompVis/stable-diffusion-v1-4-original",
device="cuda"):
        self.device = device

        self.pip=StableDiffusionPipeline.from_pretrained(model_name,
torch_dtype=torch.float16)
        self.pipe = self.pipe.to(self.device)

    def generate_image(self, prompt, guidance_scale=7.5):

        with torch.no_grad():
            generated_image = self.pipe(prompt,
guidance_scale=guidance_scale).images[0]
        return generated_image
```

3.4.2 加载数据并微调模型

```
def train_model(train_data_path, text_prompts, batch_size=4, epochs=5):
    image_paths = [train_data_path + f"/{i}.jpg" for i in range(len(text_prompts))]

    dataset = CustomSachetDataset(text_prompts, image_paths)
    dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=True)


    model = TextToImageModel()
    optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)

    for epoch in range(epochs):
        model.train()
        for text, images in dataloader:
            optimizer.zero_grad()
            loss = model(images, text)
            loss.backward()
            optimizer.step()

        print(f"Epoch {epoch+1}/{epochs} completed")
    torch.save(model.state_dict(), "fine_tuned_model.pth")
    print("Model saved to fine_tuned_model.pth")
```

3.4.3 利用训练好的模型生成香包

```
def generate_image_from_text(prompt, output_path="generated_image.png"):
    model = TextToImageModel()
    generated_image = model.generate_image(prompt)

    # 保存生成的图像
    generated_image.save(output_path)
    generated_image.show()
```

4. 系统实现的功能及展示

4.1 香包图案自动生成。可根据所给的330个香包数据集自动生成个性化香包，如下图所示。

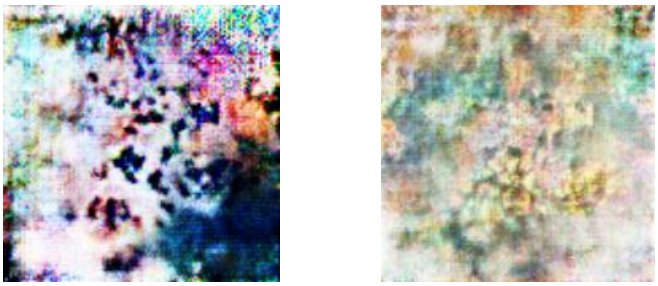


图 3-5 自动生成的多种颜色的**花卉**图案的香包

4.2 风格迁移。可将某种艺术风格（如油画、复古风格）迁移到香包图案上，生成新的香包图案，保持原始结构同时应用新的视觉风格。如下图所示，将图片的现代简美风格添加到已有香包之中。

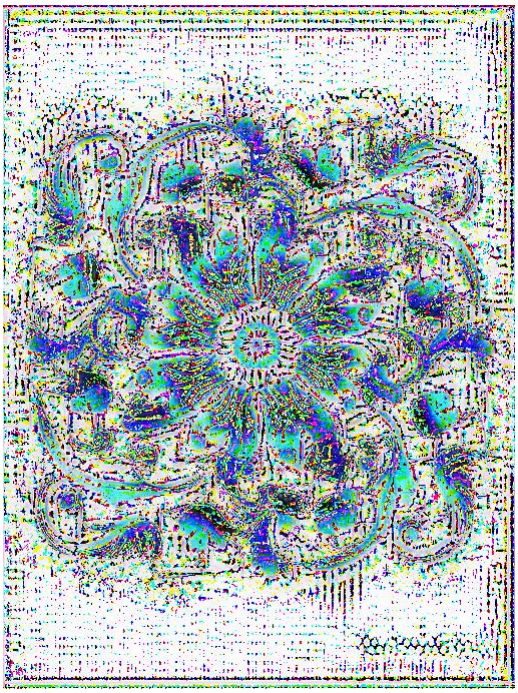


图 3-6 **简美风**添加到香包图案中

4.3 文生图。根据文本输入（如“红色花卉香包”或“复古风格香包”）生成符合描述的香包图案，能够生成多样化的香包设计。生成结果如下图所示：

图 3-7 **蓝色花卉**香包

5. 实践结果

首先利用 DCGAN，训练了一个深度卷积生成对抗网络模型来学习数据集中的香包图案特征，在多轮训练后，模型能够生成形态丰富、色彩多样的香包图案。然后使用 AdaIN 进行图像风格迁移时，将特定艺术风格迁移到香包图案上，目标是保持香包图案的原始结构同时应用新的视觉风格，风格迁移成功地将所选艺术风格的特征应用于香包图案。最后在采用 Stable Diffusion 模型进行文生图功能时，通过结合香包图案数据集的微调，模型能够根据不同的文本输入生成多样化的香包设计，生成的香包图案能够较好地匹配描述，体现出色彩、形态和风格上的变化。

6. 模型优缺点分析与改进方法

6.1 DCGAN（用于香包图案生成）

优点：

结构合理： 通过使用深度卷积神经网络，DCGAN 在图像生成中能够较好地保留图案的结构特征，生成的图案与数据集中的香包图案相似，能够表现出香包的典型特点。

训练过程稳定： DCGAN 相对其他 GAN 变种，如 WGAN 或 LSGAN，训练过程较为稳定，且容易调整超参数。

缺点：

小数据集问题： 由于香包图案数据集较小（仅 330 个样本），训练 DCGAN 时容易出现过拟合或生成模式单一的问题。虽然 DCGAN 能够生成一些有创意的图案，但这些图案的多样性和创新性可能受限。

模糊和重复： 在训练数据不足的情况下，生成的图案可能显得模糊，尤其是在图案细节上有时会显现出重复性或失真问题，缺乏高质量的细节表现。

改进：

增加数据集的样本数量或通过数据增强技术扩展数据集，能有效提高生成图案的多样性和细节质量。采用更复杂的生成对抗网络（如 StyleGAN 或 BigGAN）来提高图案生成的质量和创新性。

6.2 AdaIN（用于香包图案风格迁移）

优点：

保留结构： AdaIN 在风格迁移过程中能够有效地将目标艺术风格（如油画、复古风格）应用到香包图案上，同时保留图案的原始结构。生成的图案既有视觉风格的独特性，又保持了香包图案的设计完整性。

缺点：

细节丢失： 在一些复杂的风格迁移过程中，香包图案的细节（如花纹、边缘等）可能会丢失或变得模糊，尤其是当目标风格与香包图案的原始设计差异较大时。

过度风格化： 在某些情况下，风格迁移可能会导致过度风格化，使得原本简洁的香包图案变得不自然或失去其传统设计的特征。

改进：

使用更多的风格数据集来增强模型的风格迁移能力，使其能够适应不同类型的香包图案。调整风格迁移的强度和参数，以平衡结构保持和风格应用之间的关系，避免图案细节丢失。

6.3 Stable Diffusion（用于文生图）

优点：

多样性和创新性： 相较于 DCGAN，Stable Diffusion 生成的图案能够展现更高的多样性和创意，特别是在描述复杂、独特的设计时，模型能够生成出具有创新性的香包图案。

适应性强： Stable Diffusion 能够根据不同的文本描述生成与之匹配的香包图案，生成过程能够适应多样的输入要求，具有较好的灵活性。

缺点：

文本和图像匹配问题： 尽管 Stable Diffusion 可以根据文本生成图像，但有时生成的图案与描述不完全匹配，尤其在描述复杂或抽象的香包图案时，生成的图案可能会与文本描述有所偏差。

小数据集的影响： 微调过程中，数据集的规模较小可能导致模型对香包图案的学习不足，从而影响生成结果的质量和多样性。在某些情况下，生成的香包图案可能显得重复或缺乏创意。

改进：

扩展数据集，增加更多不同风格、形状和颜色的香包图案样本，帮助模型更好地理解香包设计的多样性。加强文本与图像之间的关联，结合更高质量的文本描述数据进行训练，以提高生成图案与文本描述之间的匹配度。

7. 实践体会与感想

通过使用 DCGAN 实现香包图案的自动生成，我体会到生成对抗网络的强大潜力。它能够通过学习数据集的潜在特征来生成全新的香包图案，展现了生成网络的多样性和创意性。然而，DCGAN 也存在一定的局限性，特别是在数据集较小的情况下。由于训练样本的数量限制，生成的图案有时显得过于简单，且缺乏细节。通过这次实践，我认识到数据集的规模对模型训练的重要性，并且在未来的项目中，我会更加注重数据集的质量和数量，甚至考虑使用数据增强等方法来弥补样本不足的问题。

AdaIN 在风格迁移方面的应用让我深刻体会到，如何平衡结构保留与风格应用之间的关系是风格迁移技术的关键。在迁移油画、复古等艺术风格到香包图案时，AdaIN 能够较好地保留原图的结构，同时引入目标风格的视觉特征。这让我认识到风格迁移不仅仅是技术上的挑战，更是如何在保持原有设计的基础上进行创新。在实践中，我也遇到了一些风格迁移效果不理想的情况，如某些图案的细节模糊或风格迁移过度。通过这次经验，我学习到如何调整迁移强度以及如何选择合适的风格，使得迁移后的图案既具艺术感，又不失原有的设计元素。

Stable Diffusion 为香包图案的文本生成提供了一个创新的解决方案。通过微调模型，我能够根据不同的文本描述生成符合要求的香包图案，这让我体验到文本与图像之间的紧密关系以及生成模型在创造性设计中的潜力。然而，文本与图像的匹配问题也给我带来了挑战。有时，生成的图案无法完全与描述一致，尤其是在描述较为复杂或抽象的图案时。此外，微调模型需要较高的计算资源和时间成本，这使我更加理解了深度学习中计算资源和时间管理的重要性。

总之，此次香包系统的制作让我意识到深度学习技术在跨领域应用中的巨大潜力。将图像生成和风格迁移技术结合，能够为香包设计带来新的创意和灵感；而文生图技术的引入，则使得香包设计的定制化和个性化变得更加可能。作为一个大三的计算机专业的学生，应该扎实自己的专业知识，才能为更多领域来贡献自己的力量!!!

致谢

感谢王老师！这门课虽然只是两学分的考查课，但是由于您时不时的课堂随机提问让我不得不仔细听讲，当然正是讲课形式也通俗有趣，我们才能听得进去！除此之外，您的结课大作业也是自认为入学以来最难的一次大作业，(第三个香包大作业多亏了各种大模型的“指导”)，相比较于复习某些专业考试大课“更加痛苦更加折磨”，当然也提高了我的动手实践能力。虽然只有短短两个月的相处，最后再次感谢王老师，祝科研顺利!!! 工作顺利!!! 新年快乐!!!