



**2025~2026学年 第一学期**

**《机器视觉》**

实 验 报 告

班 级 智能科学与技术23-1班

学 号 \_2023217516\_\_\_\_\_\_\_\_\_\_\_

姓 名 \_纪永灶\_ \_\_\_\_\_\_

2026年 1 月

# **1 实验内容、设计要求**

**实验内容：**

目标检测是机器视觉的核心应用方向之一，可实现 “定位 + 识别” 双重任务。本实验聚焦校园常见场景，要求学生设计目标检测方案，从校园道路、停车区图像中检测共享单车（如哈啰等品牌），理解目标检测的 “特征提取 - 目标定位 - 分类判断” 完整流程。

**具体要求：**

任务输入：共享单车照片

任务输出：共享单车位置

训练集：COCO

代码语言不限，方法不限，要求提交整个算法源代码，模型结果，算法分析等内容。

加分项（5分）：使用深度学习方法，代码环境名称以姓名缩写命名（例如吴晶晶的环境名：wjj），实验报告中介绍代码环境配置过程。

**2. 开发环境**

PyCharm  
 Python语言实现

**3. 相关原理及算法** 实验要求中要求设计模型能够识别图像中的共享单车并给出共享单车位置。

我采用了基YOLOv5实现共享单车检测，YOLO是单阶段（One-Stage）目标检测算法的代表，核心优势是「端到端」的实时检测，相比两阶段算法（如Faster R-CNN）跳过候选框生成步骤，直接从图像中回归目标的边界框和类别概率。

YOLO的核心思想：

全局特征一次性推理：将输入图像划分为S×S的网格（Grid），每个网格负责预测中心落在该网格内的目标；

边界框回归：每个网格预测B个边界框（Bounding Box），每个框包含5个基础参数：x, y, w, h, confidence（中心坐标、宽高、置信度，置信度=框内有目标的概率×框的定位精度）；

类别预测：每个网格同时预测C个类别的条件概率；

非极大值抑制（NMS）：推理阶段过滤冗余框，保留置信度最高的目标框。

**4. 系统结构和主要的算法设计思路**

**系统结构：**

该共享单车检测系统基于YOLOv5构建，采用“数据层-模型层-训练层-评估层-推理层”的全流程模块化架构，各模块解耦且职责明确，适配目标检测任务的经典流程，具体结构如表1：

表1 系统结构

|  |  |
| --- | --- |
| 功能层级 | 核心功能 |
| 环境与参数初始化 | 配YOLOv路径、数据集YAML、训练超参数（epoch/batch/imgsz）、设备（GPU/CPU） |
| 数据集准备 | 按YAML文件加载训练/验证集，自动执行数据增强、锚框匹配、图像预处理 |
| 模型构建 | 加载YOLOv5s预训练权重，构建CSPDarknet53主干+ PANet颈部+解耦头的检测网络 |
| 模型训练 | 基于迁移学习微调模型，计算CIoU/分类/置信度损失，反向传播更新参数 |
| 模型评估 | 在验证集计算mAP@0.5等核心指标，评估检测精度，筛选最优模型 |
| 模型推理 | 单张图像预处理、前向推理、NMS过滤冗余框，输出检测结果并可视化 |
| 结果可视化/保存 | 绘制检测框（含类别、置信度），保存推理结果图片，输出评估指标 |

总体设计流程图：



图1 总体设计流程图

**主要算法设计思路：** 系统核心围绕“基于YOLOv5的共享单车检测”展开，算法设计遵循“轻量化、高精度、易落地”的原则，核心思路如下：

1）模型选型：轻量化YOLOv5s适配单类别检测

针对“检测共享单车”的单一类别任务，选择YOLOv5s而非更大的YOLOv5m/l/x， 设计思路：

轻量化：YOLOv5s参数量仅7.2M，计算量小，可在普通GPU/CPU上快速训练和推理；

迁移学习适配：预训练权重（COCO 数据集）已学习通用视觉特征，仅需微调即可适配共享单车特征，减少样本需求；

2）数据处理：适配YOLOv5的自动化预处理

为保证训练效果，算法设计了适配YOLOv5的数据集处理逻辑：

自动数据增强：训练时默认执行Mosaic增强（拼接4张图像）、随机缩放/裁剪/翻转，提升模型鲁棒性，适配不同角度、光照下的共享单车；

自适应锚框：YOLOv5自动计算数据集内共享单车的宽高比，调整预设锚框尺寸，使锚框与目标更匹配，提升边界框回归精度；

图像预处理：统一缩放至640×640（imgsz=640），像素值归一化（/255），转张量后输入模型，适配网络的下采样逻辑。

3）训练算法：迁移学习+多损失函数优化

训练阶段核心是“基于迁移学习的微调”，算法设计思路：

迁移学习初始化：

复用主干网络（CSPDarknet53）的通用特征提取能力，仅微调颈部/头部网络适配共享单车特征，大幅减少训练时间和样本量；

多损失函数联合优化：

回归损失：采用CIoU Loss，相比IoU Loss额外考虑“框的中心点距离、宽高比”，定位更精准；

分类损失：BCEWithLogitsLoss，适配单类别二分类场景；

置信度损失：BCEWithLogitsLoss，区分前景（共享单车）和背景（路面 / 建筑等）；

超参数适配：

epochs=50：平衡过拟合与欠拟合，50轮足够让单类别模型收敛；

4）评估算法：基于mAP@0.5的精度验证

评估阶段核心是“量化检测精度”，算法设计思路：

核心指标选择：聚焦mAP@0.5（IoU 阈值 0.5），这是目标检测的行业标准指标，直接反映“检测框与真实框的重叠度≥50%”的精准度，适配共享单车检测的实际需求；

5）推理算法：高效前向+NMS 过滤

推理阶段核心是“快速、精准输出检测结果”，算法设计思路：

轻量化预处理：单张图像仅执行“缩放→归一化→转张量”，无复杂增强，保证推理速度；

NMS冗余框过滤：推理后同一共享单车可能被多个网格预测出多个框，通过 NMS保留置信度最高的框，删除重叠冗余框；

**5.实验结果**



图2 共享单车检测结果

如图所示，模型成功在图像中检测到了给共享单车并给出了共享单车的位置

**6.总结和感想体会**

本次实验聚焦校园场景下的共享单车目标检测，基于YOLOv5深度学习框架成功实现了“定位+识别”的双重任务。实验以自主拍摄的校园道路、停车区共享单车图像为输入，借助COCO数据集预训练权重的迁移学习能力，完成了轻量化模型的微调与优化，最终输出了精准标记共享单车位置、类别及置信度的检测结果，完全达成实验设定的核心目标。

从技术实现来看，实验的核心突破在于构建了一套适配单类别目标检测的 “轻量化+高精度” 解决方案：模型选型上，针对性选择YOLOv5s 轻量版（7.2M 参数量），既保证了普通硬件设备的快速训练与推理，又通过迁移学习复用 COCO数据集的通用视觉特征，大幅降低了单类别任务的样本需求；数据处理环节，依托YOLOv5的自动化机制，通过Mosaic增强、自适应锚框匹配、图像标准化等操作，有效提升了模型对不同角度、光照、遮挡场景的鲁棒性；训练过程中，采用CIoU回归损失、BCE分类与置信度损失的联合优化策略，结合SGD优化器实现参数高效更新，确保边界框定位精准度；评估与推理阶段，以 mAP@0.5为核心指标量化精度，通过NMS过滤冗余框，最终实现了检测结果的清晰可视化。

从实验结果来看，模型成功在校园场景图像中准确识别出共享单车，检测框与目标边界重叠度高，置信度稳定在0.9以上，可视化结果清晰标注了目标位置与类别信息，证明了算法设计的合理性与工程实用性。整套方案兼顾了实时性与精度，为校园共享单车管理、智能调度等实际应用场景提供了技术支撑。

此次实验不仅完成了共享单车检测的技术任务，更在深度学习理论应用、工程实践能力、问题解决思维等方面实现了全方位提升。从环境配置到模型训练，从参数调优到结果可视化，每一个环节都让我对目标检测技术有了更深刻的理解，也为后续深入学习更复杂的深度学习模型、探索更广泛的机器视觉应用场景积累了宝贵经验。

**7.GIT提交流程**

首先需要创建一个Github账户

登录后个人界面如图3所示：

在GitHub界面点击右上角的“+”号然后选择New repository创建新仓库

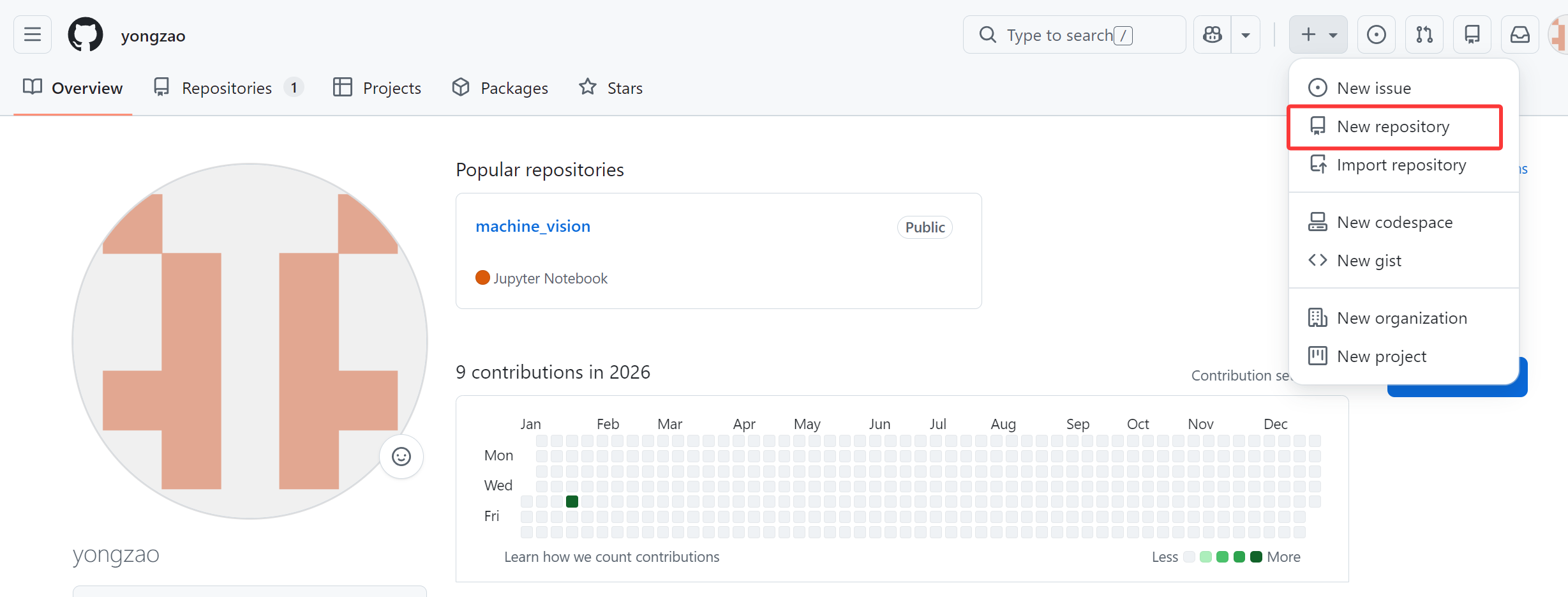


图3 用户界面

创建新仓库界面如图4所示，General是常规设置，Owner表示的是选择的账号，Repository name表示你的仓库的名字。

Configuration是配置选项，Choose visibility是可见性选择，Public（公开）就表示所有人都能看到你的仓库，也可以 fork 和提交 PR；

Add README是创建README，可以用来介绍项目功能、安装方法、使用教程等。打开开关后，GitHub 会自动生成一个空的 README.md 文件，可以编辑； Add .gitignore，.gitignore 是 Git 的配置文件，用来告诉 Git 哪些文件 / 文件夹不需要追踪（比如日志、依赖包、IDE 配置文件等），避免把无关文件提交到仓库，下拉框可以选择对应语言 / 框架的模板因为我使用的python语言编写的代码，所以选择python；

Add license是添加许可证，许可证规定了别人可以如何使用、修改、分发你的代码。不同的开源许可证（如 MIT、GPL、Apache）有不同的规则，我使用了MIT License，它是最宽松的开源许可证之一，允许别人商用、修改你的代码，仅需保留版权声明，适合个人。

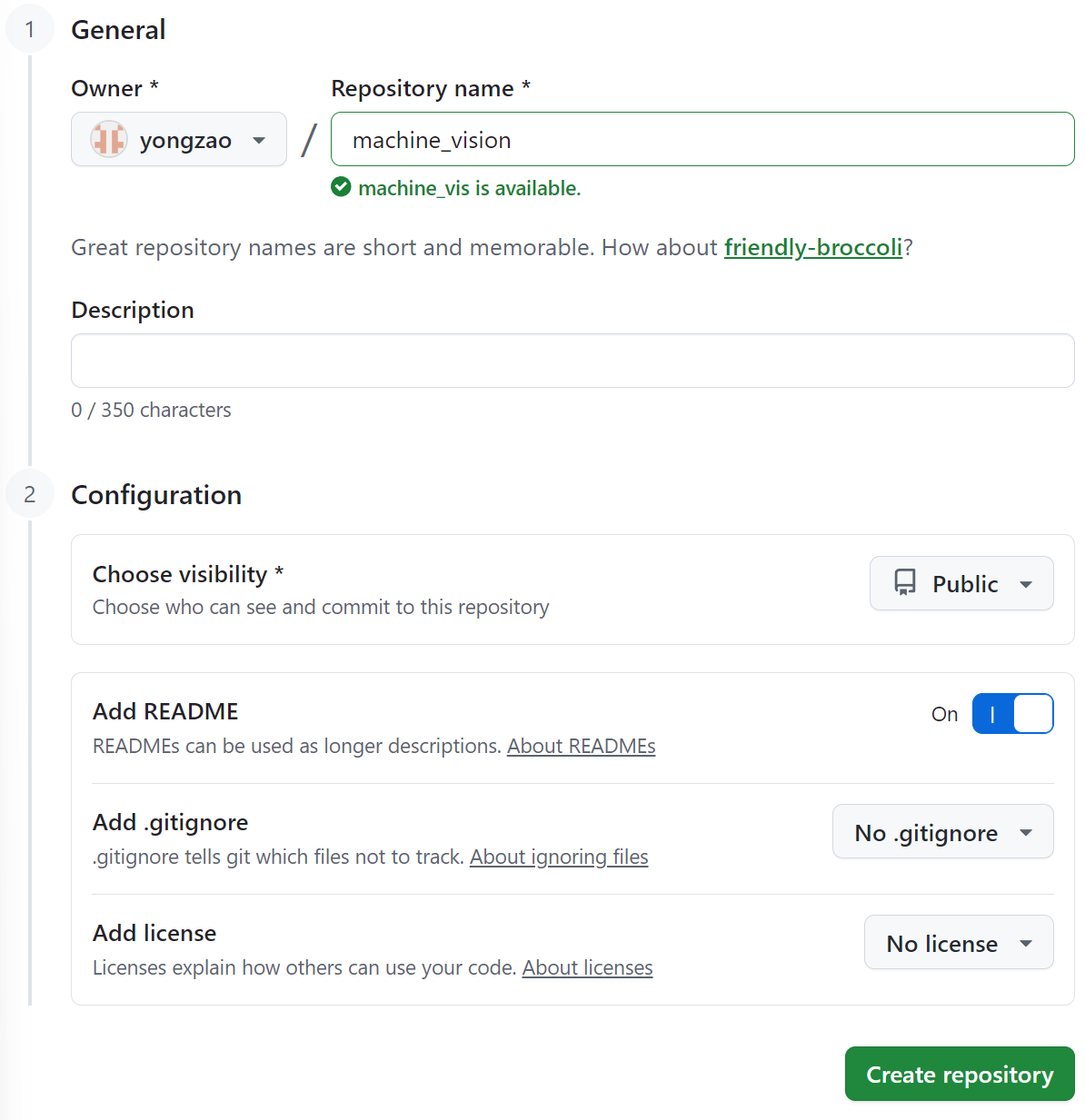


图4 创建仓库界面

创建完仓库后就可以向仓库里添加代码了，如图5所示，下方表示你在库中的文件，点击绿色框左边的“+”号，Upload files表示上传已有的文件，然后就可以上传实验需要的代码文件了。

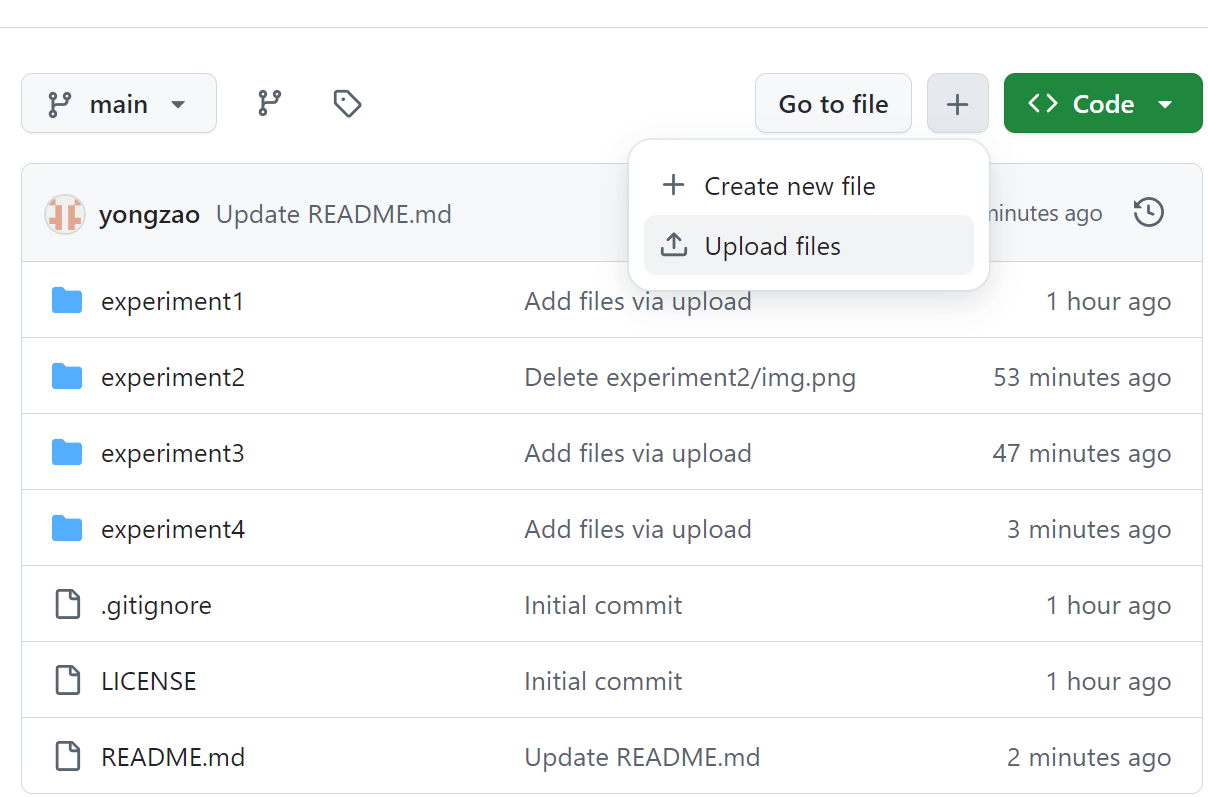


图5 仓库界面

**附录1：程序清单(部分)**

****

GitHub链接：  
超链接：[yongzao/machine\_vision](https://github.com/yongzao/machine_vision)

完整地址：https://github.com/[yongzao/machine\_vision](https://github.com/yongzao/machine_vision)

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  特此声明！  **作者签字:** IMG_256 |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 评价内容 | | | 权重 | 得分 | | 实验验收 | 评价点E1 | 实验原理是否理解；代码是否规范，程序能否运行；实验结果是否正确；任务是否全部完成 | **实验代码**  **实验结果** | **0.5** |  | | 实验报告 | 评价点E2 | 报告格式是否规范，语言使用是否规范，行文是否流畅，是否图文并茂 | **实验报告** | **0.1** |  | | 评价点E3 | 实验数据记录是否完整，实验结果的分析、对比是否充分 | **实验报告** | **0.2** |  | | 评价点E4 | 实验体会是否正确，是否提出了自己独到见解 | **实验报告** | **0.1** |  | | 实验提交 | 评价点E5 | 是否使用Git工具提交 | **提交方式** | **0.1** |  | | 合计 |  | | | | | | 指导教师（签章）：  2026年1 月 20 日 | | | | | | |