**实 验 报 告**

**（与程序设计有关）**

**课程名称： 人工智能实列与应用**

**实验题目：git操作及利用mmyolo检测目标**

**班级学号： 2303050411**

**姓 名: 吴小松**

**成 绩：**

**沈 阳 理 工 大 学**

**2025年 1 月15日**

|  |
| --- |
| **实验目的及要求：**  1.新建gitee账号，并建立一个公共仓库；  2.在该公共仓库内，执行Git的常见操作：新建文件、修改文件、合并分支；  3.按照课上的要求，安装MMYOLO，并调用摄像头执行实时目标检测，并将检测结果截图；  4.将上述过程形成文档报告，并在报告结尾，附上所建立公共仓库的地址。 |
| **软硬件环境：**  硬件：计算机一台  软件：Pycharm、window11、git |
| **算法或原理分析（实验内容）：**   1. Git      1. git add . 添加全部改动 2. git commit 提交文件 3. git push 推送文件 4. mmyolo 5. 环境安装   安装conda、PyTorch、安装MMYOLO和依赖库   1. 数据集准备   本实验使用mmyolo给的144张猫的原始数据   1. 配置准备 2. 模型训练   Python tools/train.pyconfigs/yolov5/yolov5\_s-v61\_fast\_1xb12-40e\_cat.py   1. 模型测试   pythontools/test.py configs/yolov5/yolov5\_s-v61\_fast\_1xb12-40e\_cat.py \  work\_dirs/yolov5\_s-v61\_fast\_1xb12-40e\_cat/epoch\_40.pth \  --show-dir show\_results |

|  |
| --- |
| **程序代码或实现过程：**  配置代码如下：  \_base\_ = 'yolov5\_s-v61\_syncbn\_fast\_8xb16-300e\_coco.py'  data\_root = './data/cat/' # 数据集根路径  class\_name = ('cat', ) # 数据集类别名称  num\_classes = len(class\_name) # 数据集类别数  # metainfo 必须要传给后面的 dataloader 配置，否则无效  # palette 是可视化时候对应类别的显示颜色  # palette 长度必须大于或等于 classes 长度  metainfo = dict(classes=class\_name, palette=[(20, 220, 60)])  # 基于 tools/analysis\_tools/optimize\_anchors.py 自适应计算的 anchor  anchors = [  [(68, 69), (154, 91), (143, 162)], # P3/8  [(242, 160), (189, 287), (391, 207)], # P4/16  [(353, 337), (539, 341), (443, 432)] # P5/32  ]  # 最大训练 40 epoch  max\_epochs = 40  # bs 为 12  train\_batch\_size\_per\_gpu = 12  # dataloader 加载进程数  train\_num\_workers = 4  # 加载 COCO 预训练权重  load\_from = 'https://download.openmmlab.com/mmyolo/v0/yolov5/yolov5\_s-v61\_syncbn\_fast\_8xb16-300e\_coco/yolov5\_s-v61\_syncbn\_fast\_8xb16-300e\_coco\_20220918\_084700-86e02187.pth' # noqa  model = dict(  # 固定整个 backbone 权重，不进行训练  backbone=dict(frozen\_stages=4),  bbox\_head=dict(  head\_module=dict(num\_classes=num\_classes),  prior\_generator=dict(base\_sizes=anchors)  ))  train\_dataloader = dict(  batch\_size=train\_batch\_size\_per\_gpu,  num\_workers=train\_num\_workers,  dataset=dict(  data\_root=data\_root,  metainfo=metainfo,  # 数据集标注文件 json 路径  ann\_file='annotations/trainval.json',  # 数据集前缀  data\_prefix=dict(img='images/')))  val\_dataloader = dict(  dataset=dict(  metainfo=metainfo,  data\_root=data\_root,  ann\_file='annotations/test.json',  data\_prefix=dict(img='images/')))  test\_dataloader = val\_dataloader  \_base\_.optim\_wrapper.optimizer.batch\_size\_per\_gpu = train\_batch\_size\_per\_gpu  val\_evaluator = dict(ann\_file=data\_root + 'annotations/test.json')  test\_evaluator = val\_evaluator  default\_hooks = dict(  # 每隔 10 个 epoch 保存一次权重，并且最多保存 2 个权重  # 模型评估时候自动保存最佳模型  checkpoint=dict(interval=10, max\_keep\_ckpts=2, save\_best='auto'),  # warmup\_mim\_iter 参数非常关键，因为 cat 数据集非常小，默认的最小 warmup\_mim\_iter 是 1000，导致训练过程学习率偏小  param\_scheduler=dict(max\_epochs=max\_epochs, warmup\_mim\_iter=10),  # 日志打印间隔为 5  logger=dict(type='LoggerHook', interval=5))  # 评估间隔为 10  train\_cfg = dict(max\_epochs=max\_epochs, val\_interval=10) |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **结果分析：**  1、git。地址：https://github.com/mi6661/2303050411-wuxiaosong.git    2、mmyolo | | | |
| **教师签字** |  | **日 期** |  |

*双面打印*