

编号:

2022 - 2023 学年第 2 学期



浙大城市学院

ZHEJIANG UNIVERSITY CITY COLLEGE

# 实 验 报 告



课 程 名 大数据与人工智能综合实践

课 程 号 C01159

学 生 姓 名 樊翔宇 刘书良 王梓霖 王晓圉

学 号 32003028 32003032 32003058 32003230

专业班级 计算 2002 计算 2003 计算 2003 计算 2003

所 在 学 院 计算学院

指 导 老 师 王军 魏李娜 徐超清 李卓蓉

实验报告日期: 2023 年 7 月 10 日

课程名	大数据与人工智能综合实践	课程号	C01159
实验项目名	计算机视觉算法模型设计+可视化理论实践		
实验时间	2023.6.30-2023.7.14 上午 8:30-11:30 下午 13:30-15:50		
小组合作: <input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否	小组成员: 樊翔宇 刘书良 王梓霖 王晓围		

## 一、摘要（300-400 字）

本次实验，我们选择了病理图像细胞分割的课题。

根据数据集 <https://arxiv.org/abs/2003.10778>，我们制作了一个包含前后端的病理细胞分割程序。利用 python 的 streamlit 库做可视化界面，并结合 mmdetection 的 python 代码完成页面与模型的交互。

前端包含四个主要的页面，分别是 训练、预测、展示和画图。其分别可以实现模型在前端一键训练并且展示训练过程的 logging；预测 test 页面可以利用选择的 pth 文件和 config 文件，选择一张测试图片来进行预测；展示 view 页面可以将训练完生成的 log 文件动态地展示出来；画图 draw 页面可以选择不同的参数并画出需要的评价曲线，比如 loss、acc 等。

小组的进度大致是 3 号—5 号是跑模型和前端页面工具熟悉阶段，由于我们选择的是病理细胞分割，而给的数据集转化到 coco 数据格式时有发生格式不匹配的问题，耽误了

半天时间。随后 5-9 号，模型继续调试，直至能正确训练和预测，并在此基础上跑更多效果更好的模型；前后端继续编写代码，分析平台的代码结构，找到参数配置和输出 logging 等相关代码，在前端实现交互，另一方面利用训练完产生的 json 文件生成一系列模型评估的图像。之后的日期撰写实验报告和准备 ppt 与答辩。

## 二、 简介 Introduction（项目背景、研究意义）

医学图像分割是从 CT 或 MRI 等医学诊断中用到的图像中识别器官或病变的像素点，这是医学图像分析中最具有挑战性的任务之一，其目的是传递和提取有关这些器官或者组织的形状和体积的关键信息。而 Mask R-CNN、Cascade Mask-RCNN、Condinst 是实例分割中近些年来比较优秀的模型。它们在目标检测和实例分割任务上取得了显著的成果。然而，针对这些模型的使用和调整通常需要熟练的编程和深度学习经验，限制了广大用户的应用范围。

为了降低模型的使用门槛并提高应用的灵活性，我们设计了一个前后端的程序，结合了 mmdection 平台，实现了在前端一键训练模型、修改模型参数、得到预测结果和展示模型过程等功能。这个项目为用户提供了一个简单而强大的界面，使得使用者无需深入了解模型的底层实现，即可快速搭建、训练和优化目标检测和实例分割模型。

这次的项目得以降低技术使用门槛、加速模型开发和迭代，最重要的是小组成员得以利用这次短学期掌握到在目前流行的框架下模型训练的基础能力，前端后端相结合的能力。

### 三、 方法（模型理解、可视化技术）

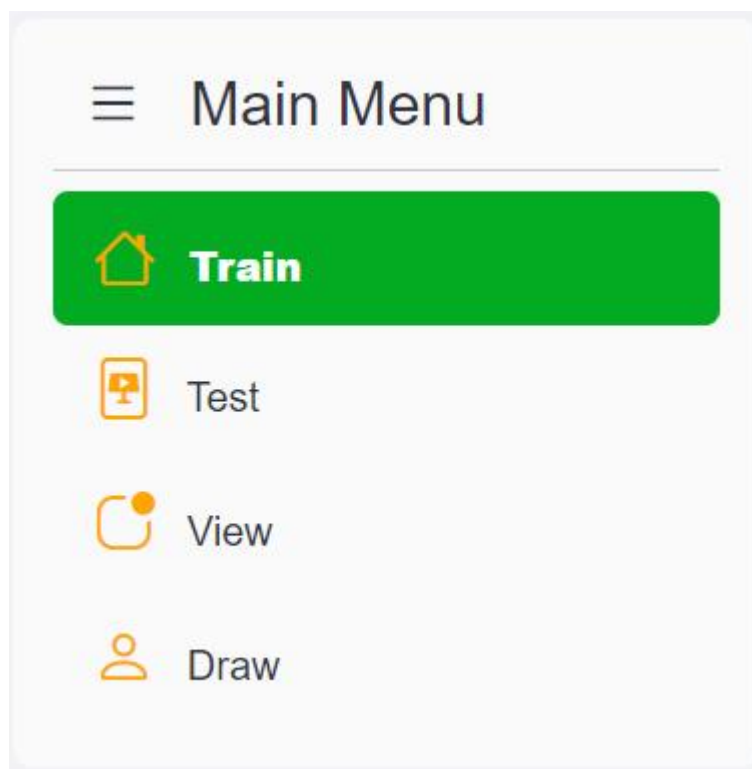
Cascade Mask-RCNN 是对 Mask R-CNN 的改进，采用级联结构来逐步提高目标检测和实例分割的性能。级联过程中，每个阶段利用前一阶段的输出结果提供更准确的候选框和分割结果，从而进一步提升模型性能。Cascade Mask-RCNN 在处理复杂场景和小目标时表现出色，具有更高的检测和分割精度。

使用到的可视化技术是 python 的 streamlit 库。Streamlit 是可以用于快速搭建 Web 应用的 Python 库。好处是 API 简单明了，易上手；无须学习前端知识（html、css、javascript）Streamlit 基于 tornado 框架，封装了大量互动组件，同时也支持大量表格、图表、数据表等对象的渲染，并且支持栅格化响应式布局；支持 markdown 和 html 文本的渲染。同时 streamlit 有相应结合 d3.js 和 echarts 的包。在可视化数据方面，用这些 python 包就可将图像美观的展示在 streamlit 页面上。

### 四、 实验（模型性能评估、可视化系统框架、功能模块介

绍，不能只贴图，还要有文字分析）

该系统可视化采用 streamlit 平台展示数据，主要分为 4 个页面，采用功能按钮来切换页面。



默认页为训练页面，页面包含有参数设置和训练过程展示细节，训练之前需要把数据按要求放在指定位置并采取固定命名即可，对于 coco 类别信息手动处理，由于使用 mmdetection 平台训练，所以无法进行参数修改，对于可以修改的参数做出了相应的可视化选择，比如学习率和轮次，也可以通过切换配置来训练不同的模型，训练过程的各种参数也会随着训练进度展示在主页面上。

## 训练参数设置

训练轮数

1 5 10

学习率

0.00 0.02

批次大小

4

请选择一个配置文件

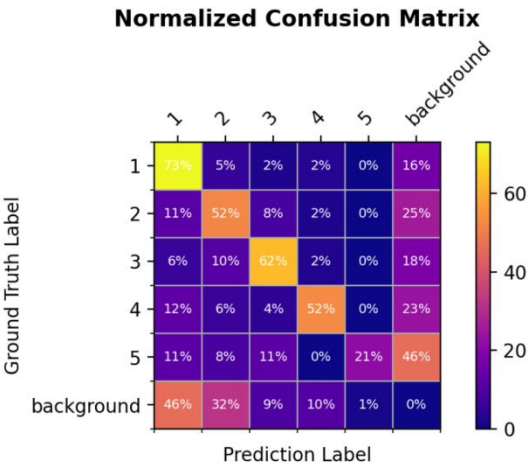
cascade.py

测试参数

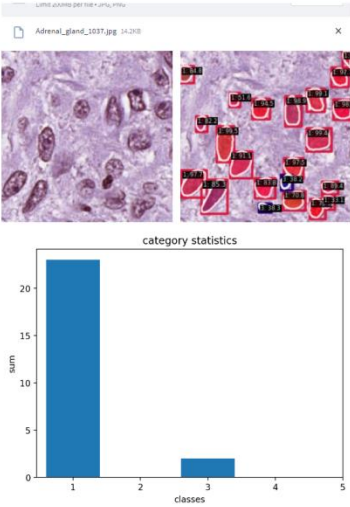
开始训练

第二页为测试页面，侧边栏为各项参数选择以及相应的操作，其一是绘制混淆矩阵，我们提前把训练过程中的 pkl 文件、配置文件以及权重文件放到相应的 work\_dir 和 config\_model, 我们只需要把文件放到对应的文件的夹即可

在页面中展示出来，方便选择。（文件夹不存在会自动创建）  
比如直接展示混淆矩阵：



预测图片也是相应的操作，选中图片以及相应的模型即可，预测出来的包含原图与预测图、类别统计柱状图：



可自定义 iou 阈值。

由于是借助于 mmdection 平台，所以该技术实现主要针对与该平台代码的改动，至于前后端传值则相对简单，比如混淆矩阵的绘制采用的是 mmdection 平台中的命令 “python tools/analysis\_tools/confusion\_matrix.py”

```
work_dirs\cascade-mask-rcnn_r50_fpn_1x_coco\cascade-  
mask-rcnn_r50_fpn_1x_coco.py
```

```
work_dirs\cascade-mask-rcnn_r50_fpn_1x_coco\cascade-  
new\result.pkl” 。我们通过源码发现具体实现是使用函数读取控制台参数然后进行 if 判断实现的，因此我们直接对参数进行赋值：
```

```
# 下面为可修改参数  
args.__dict__['config'] = data["config_path"]  
args.__dict__['prediction_path'] = data["pkl_path"]  
args.__dict__['save_dir'] = "./work_dirs/"
```

具体画图调用的是 `plot_confusion_matrix()` 函数，该函数源代码是直接把文件保存在所提供的文件中，而通过阅读发现我们不需要保存，直接把参数 `fig` 返回传递给 `streamlit`，使用 `st.pyplot()` 即可画图。

预测生成图像也是阅读源文件 `test.py`，对其进行修改将必要条件传给 `test` 文件，至于阈值控制我们发现“`model = init_detector(config_file, checkpoint_file, device='cpu')`”在构建模型的时候使用到了该函数，推断应该是这个函数来控制的，查看源代码发现他有第四个参数：

```
Defaults to cuda:0.  
cfg_options (dict, optional): Options to override some settings in  
the used config.
```

根据名字可知是修改配置文件，即将我们提供的配置信息覆盖在原文件，因此我们只需要的阈值传给该参数即可：



```

cfg_options = {
    'model.test_cfg.rcnn.score_thr': iou
}
model = init_detector(config_file, checkpoint_file, device='cpu',
    cfg_options=cfg_options)

```

第三页为动态数据可视化页面，主要用来解决训练时间过程，无法在答辩过程演示，所以我们通过读取训练过程中的日志文件来动态的演示训练过程。

第三页主要采用了 `receptplotly` 这个可以使条形图动态变化的库。代码逻辑是首先获得 json 文件，对不同的结构进行不同的筛选，并存入 `dataframe` 。

```

{"lr": 0.01799799599198394, "data_time": 0.048844280242919924, "loss":
{"coco/bbox_mAP": 0.078, "coco/bbox_mAP_50": 0.132, "coco/bbox_mAP_75":

```

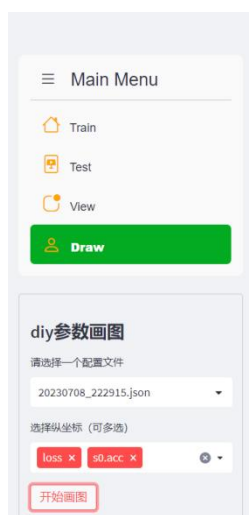
将每个 json 都拆分出自变量、因变量，添加几个可选的参数如表格高宽等，核心代码如下

```

raceplot = barplot(df, item_column=item_column, value_column=value_column, time_column=time_column)
fig=raceplot.plot(item_label = item_label, value_label = value_label, time_label = time_column+'s:',

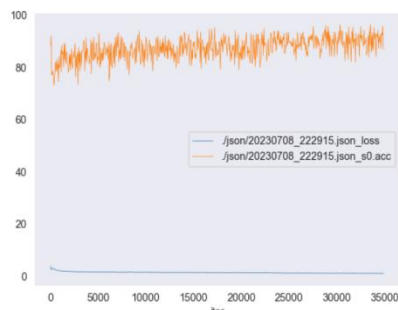
```

第四页为画图页面，使用日志文件，调用 `mmdection` 的画图工具进行展示相应的数据，比如可以选择一个日志的 json 然后选择需要展示的纵坐标即可，也可以多选：



## MMDETECTION VIEW

这是一个简单的模型训练展示示例



## 五、 总结（课程项目总结、心得体会）

通过这一次的短学期，主要学习到了如何使用可视化工具和技术去分析和呈现模型数据和视觉数据。重新复习了一下如何使用 mmdetection 平台实现检测和分割。本次项目背景是对病理图像进行分割，但重点在分析数据和可视化数据。一开始对分析数据还抱有乐观的态度，直到实际上手才发现困难重重，比如说如何选取数据，数据怎样清洗，选取哪些合适的数据，使用什么图像能合理的展现出想要的效果等等都是要考虑的事。

本次的课程对我们的启发就是：学习要理论结合实践，该项目帮助我们将所学的计算机视觉算法和可视化理论应用于实际场景。培养了实际应用的导向思维，使我们能够将所学知识和技能应用于解决实际问题。通过不断思考数据呈现的设计等。也激发了我们解决问题的能力，在面对困难

学会并寻找解决方案的能力。