﻿基于深度学习车牌识别系统课程测试报告

## 1. 项目概述

本项目实现了一个本地可运行的车牌检测与识别系统，包含以下功能：

车牌检测：支持简单轮廓法、Sobel+形态学增强、YOLO 三种方案。

车牌识别：基于 ResNet18 的多头分类网络 PlateNet，对 7/8 位车牌逐字符识别。

本地 Web 演示：使用 Flask 提供图片上传与可视化识别结果。

## 2. 数据集与标注处理

数据集：CCPD2020（使用 `ccpd\_green` 划分）。

识别训练：从文件名解析车牌字符标签与车牌框，裁剪车牌区域后进行逐字符分类。

检测训练：通过 `prepare\_ccpd\_yolo.py` 将 CCPD 文件名解析的车牌框转换为 YOLO 标签格式，并生成 `ccpd\_yolo/dataset.yaml`。

## 3. 模型与方法

### 3.1 车牌识别（PlateNet）

Backbone：ResNet18（去除分类层）。

多头分类：8 个输出头（省份、字母、字符/数字），按车牌位分别分类。

输入尺寸：96×240（RGB），标准化与常见 ImageNet 统计一致。

损失函数：逐头交叉熵，省份位加权 2.0 以提升汉字首位识别。

### 3.2 车牌检测（YOLOv8n）

框架：Ultralytics YOLOv8n，单类车牌检测。

训练参数来自 `runs/detect/train4/args.yaml`：

- `model=yolov8n.pt`，`epochs=100`，`imgsz=640`，`batch=32`，`device=0`。

数据集配置：`ccpd\_yolo/dataset.yaml`。

## 4. 训练与测试设置

### 4.1 识别训练示例

* CPU 调试：

`python train\_ccpd.py --data\_root CCPD2020/CCPD2020/ccpd\_green --epochs 1 --limit\_train 200 --limit\_val 50 --device cpu`

* GPU 全量：

`python train\_ccpd.py --data\_root CCPD2020/CCPD2020/ccpd\_green --epochs 15 --batch\_size 128 --device cuda:0`

### 4.2 检测训练示例

* 数据准备：

`python prepare\_ccpd\_yolo.py --ccpd\_root CCPD2020/CCPD2020/ccpd\_green --out\_dir ccpd\_yolo`

* YOLO 训练：

`yolo train model=yolov8n.pt data=ccpd\_yolo/dataset.yaml epochs=100 imgsz=640 device=0 batch=32`

## 5. 评估结果（车牌检测）

基于 `runs/detect/train4/results.csv` 统计：

最佳 mAP50-95 出现在第 38 个 epoch：

- Precision = 0.99591

- Recall = 0.97405

- mAP50 = 0.99355

- mAP50-95 = 0.87112

训练结束（第 100 个 epoch）结果：

- Precision = 0.99675

- Recall = 0.96004

- mAP50 = 0.98786

- mAP50-95 = 0.84568

## 6. 系统功能测试（本地 Web 演示）

以下是启动命令的一个示例：

`python web\_app.py --weights ccpd\_recognition\_new.pth --device cuda:0 --port 8000 --detector yolo --det\_weights runs/detect/train4/weights/best.pt`

主要功能：

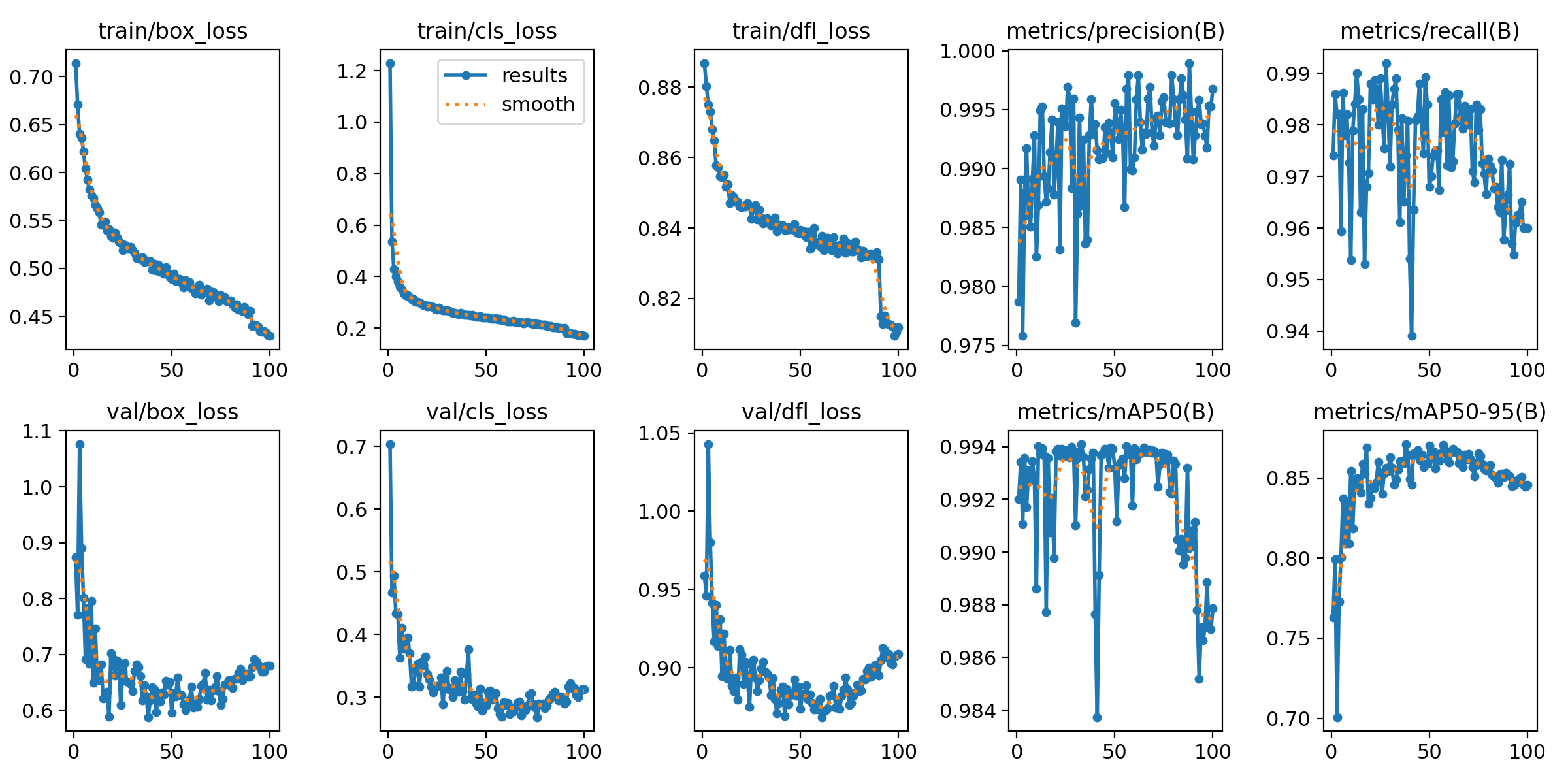
上传车辆图片后，系统返回车牌检测框与识别结果。

可切换 `simple` / `enhanced` / `yolo` 三种检测方式对比效果。

## 7. 可视化结果（训练输出）

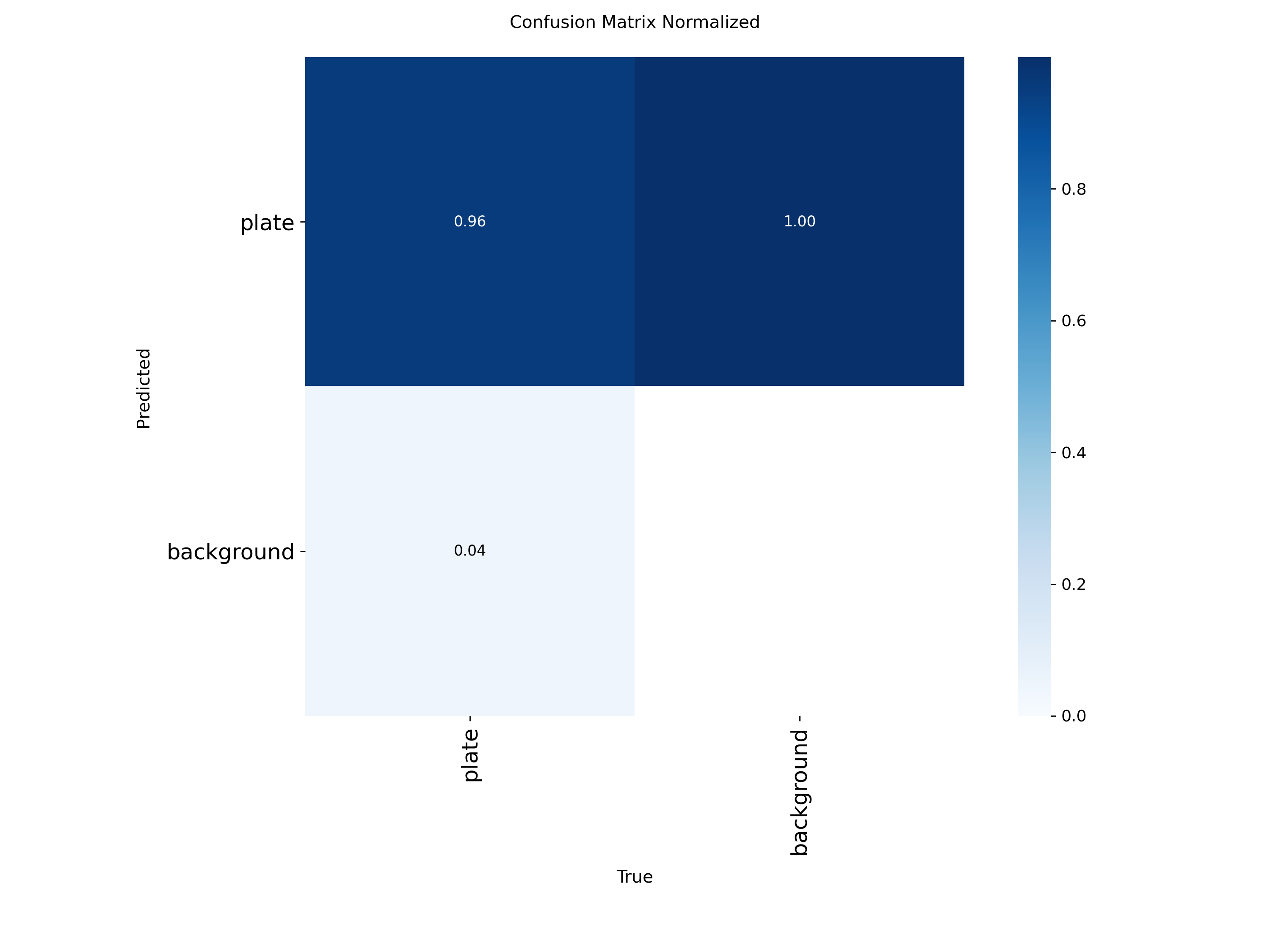
> 以下图片来自 `runs/detect/train4/` 目录。

* YOLO 训练指标汇总曲线



YOLO Results

* 混淆矩阵（归一化）



Confusion Matrix

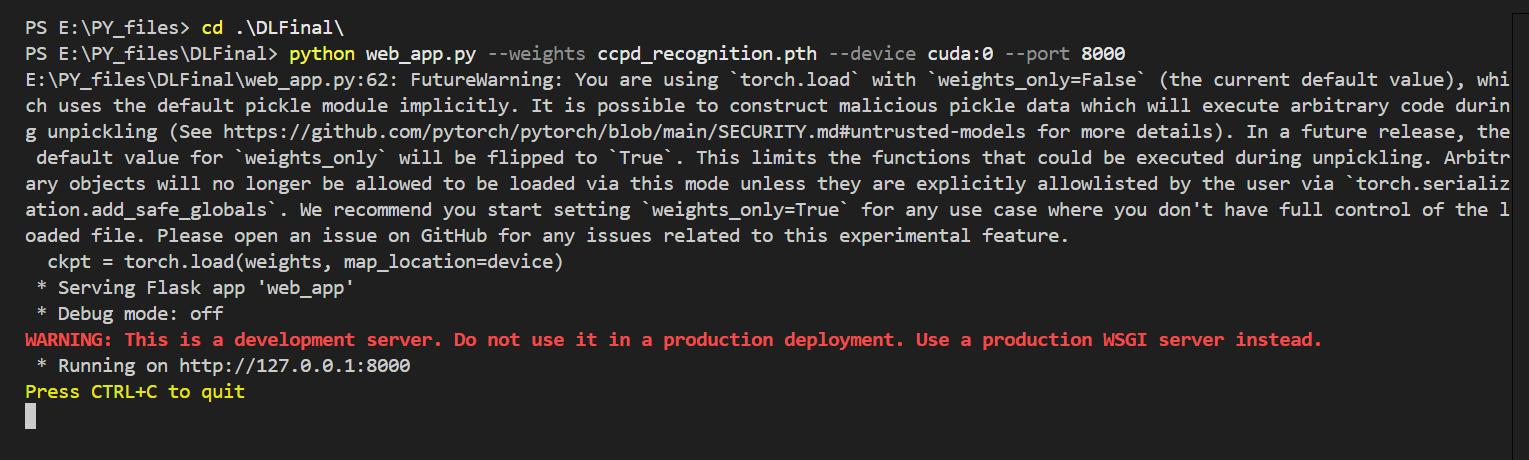
* 验证集预测可视化示例



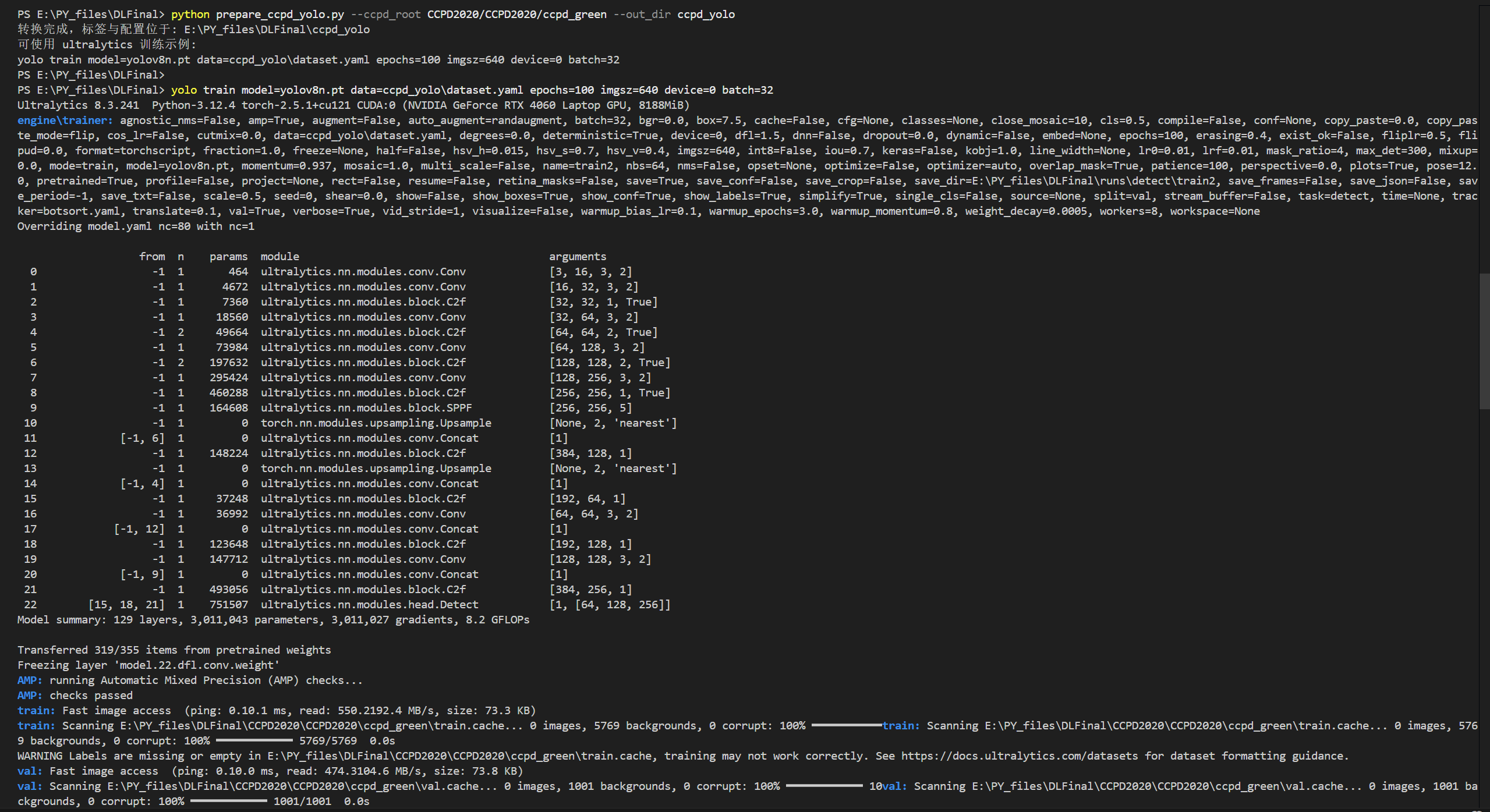
测试集预测

### 7.1 系统测试截图

> 以下图片来自 `e:\PY\_files\深度学习` 目录。



本地网页程序运行中



使用yolo识别车牌位置



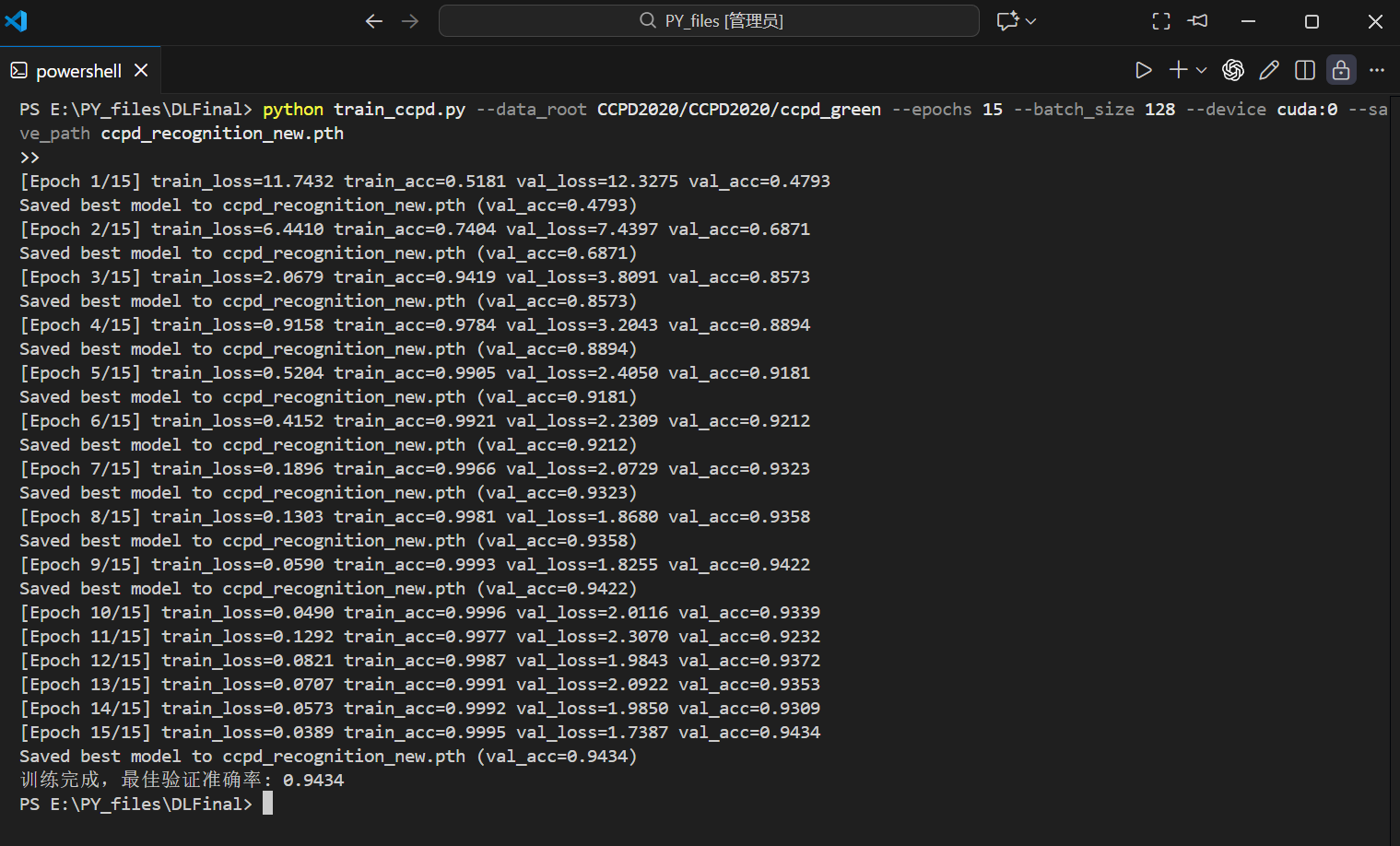
识别成功1



识别成功2



正确识别



最佳模型v2

### 8. 结论与改进建议

检测模型在 CCPD2020 上表现稳定，mAP50 维持在 0.99 左右，mAP50-95 达到 0.87 左右。

识别模型结构清晰，支持 7/8 位车牌，但缺少量化评估记录，建议补充测试集准确率与端到端识别指标。

可进一步增加真实场景数据测试（雨雪、夜间、遮挡）以及端到端流程（检测+识别）的综合评测。

### 附录：依赖与环境

Python 依赖：’torch’、’torchvision’、’opencv-python’、’flask’、’ultralytics’、numpy