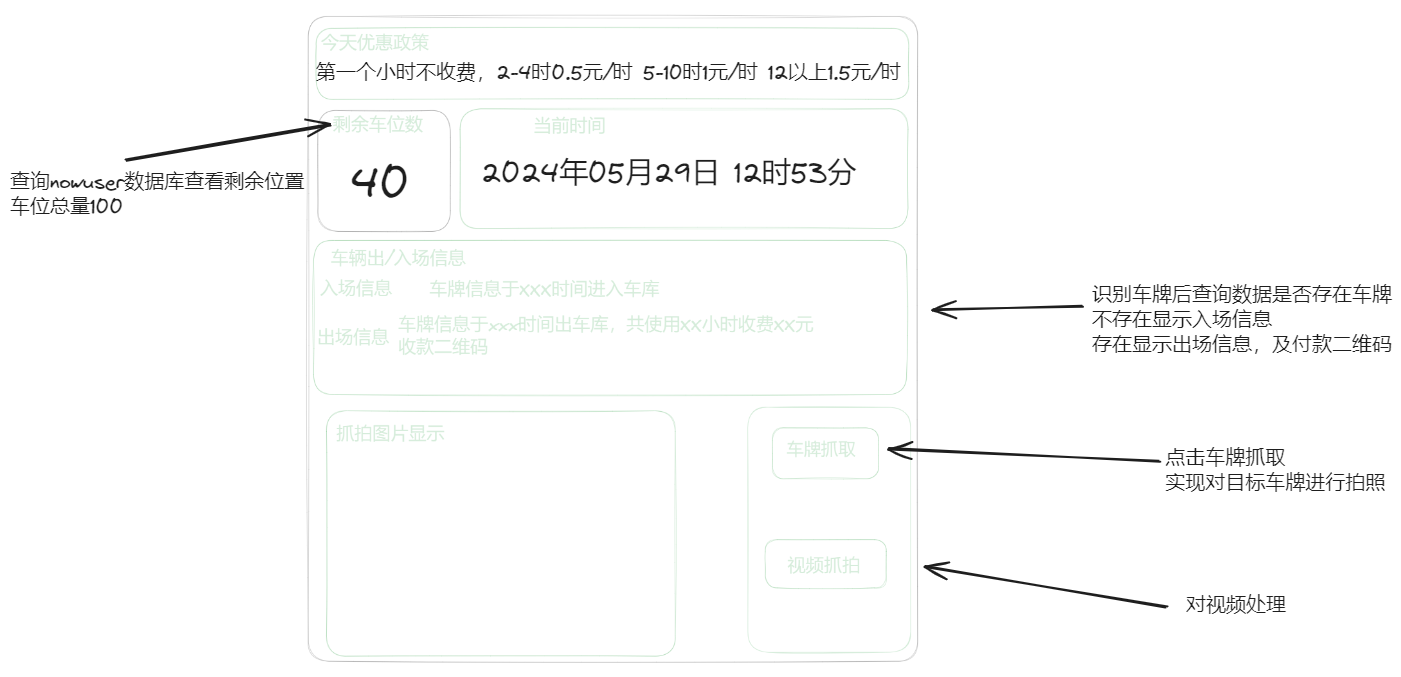
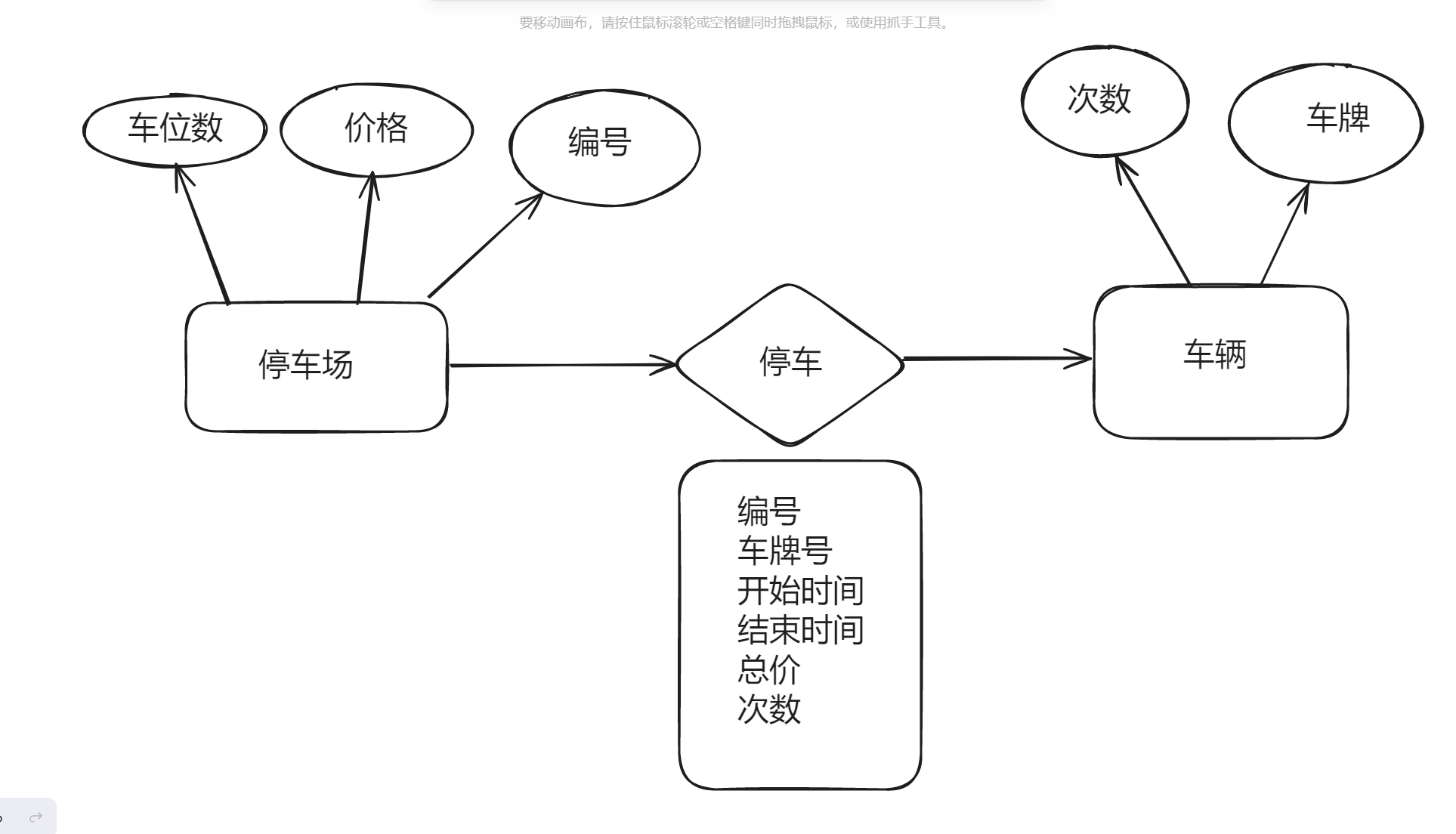
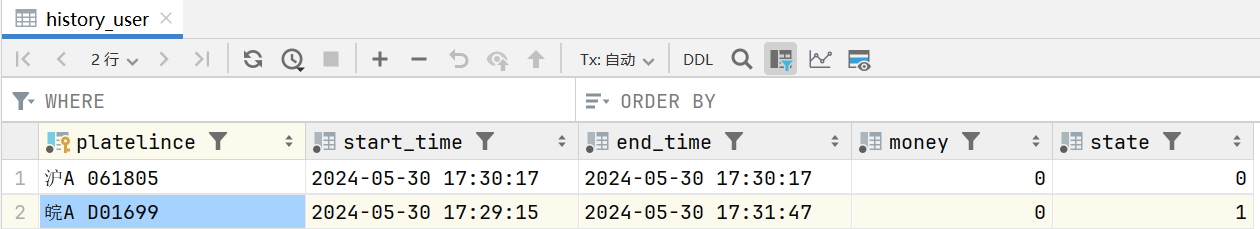
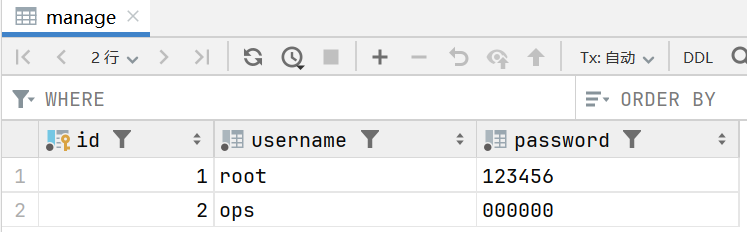
|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 2022020825 | 姓名 | 熊开云 |
| 项目需求 | 实现自动停车登记收费 | | |

1. 需求分析

当车辆进入摄像头区域进行拍照，对车牌进行识别，并对车位数进行管理。并对用户数据进行持久化处理。

1. 界面设计



1. 数据库设计
   1. 
   2. 数据库名为homework，里面有两个表一个停车用户表，一个是管理用户表。
      1. 停车用户表
         1. 
         2. 字段说明
            1. 车牌号 类型 varchar(20) 主键 非空
            2. 开始时间 类型 datetime 非空
            3. 结束时间 类型 datetime 非空
            4. 总价 类型 int 非空
            5. 停车状态 类型 int 非空
      2. 管理用户表
         1. 
         2. 字段说明
            1. id 类型 主键 非空 自增
            2. 用户名 类型 varchar(20) 非空
            3. 密码 类型 varchar(20) 非空
2. UI设计
   1. 登陆界面
      1. 任务说明

实现用户输入账号密码进行登录

* + 1. 代码展示



* + 1. 界面展示



* + 1. 登录逻辑判断代码

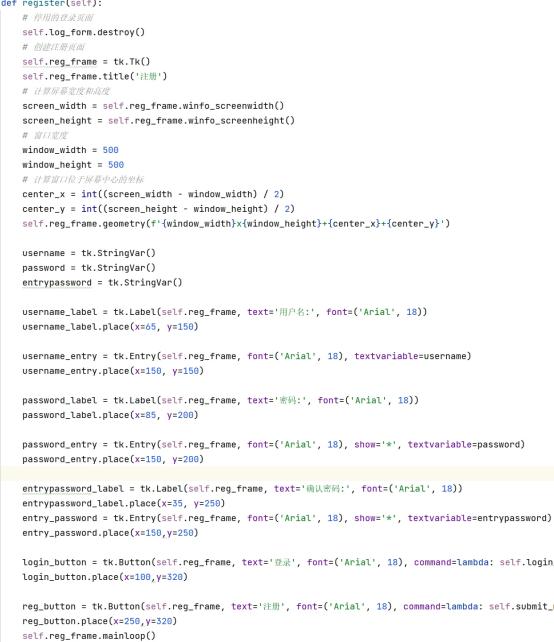
说明：连接数据进行查询查看用户是否注册，若注册检查密码是否正确，给用户提示



* 1. 注册界面
     1. 任务说明

给用户提供注册界面实现注册，并把注册结果提交到数据库

* + 1. 代码展示



* + 1. 界面展示



* + 1. 注册逻辑代码

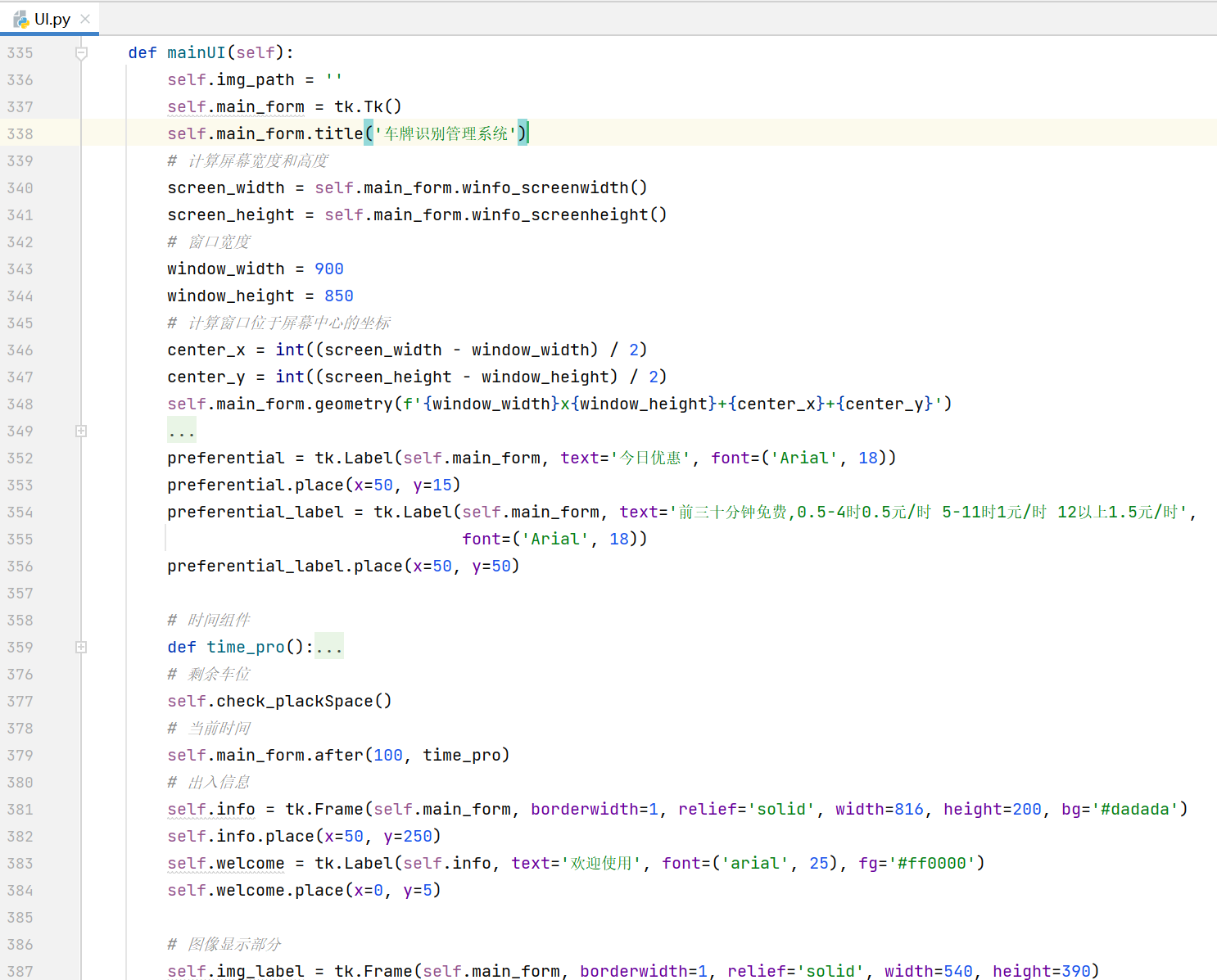
两次输入密码是否一致的校验，提交数据库。



* 1. 主界面设计
     1. 任务说明

能对车辆的车牌进行检测，并记录进出时间和使用的价格。

* + 1. 代码展示



* + 1. 界面展示



1. 模型训练
   1. 任务说明

使用yolov8训练目标检测模型

* 1. 步骤说明
     1. 采集数据集
     2. 使用labelimg对数据集进行标注
     3. 下载ultralytics模块
     4. 对数据集进行合理分割(可选)
     5. 划分训练集和测试集
     6. 下载模型及对应yaml文件
     7. 编写训练脚本
     8. 编写data.yaml
     9. 开始训练
     10. 训练完毕查看生成的results.png，查看评估标准的趋势走向(判断是否需要在进行训练)
  2. 对应步骤的说明
     1. 
     2. 标注说明
        1. 标注前请先下载labelimg下载命令

pip install labelimg -i <https://mirrors.aliyun.com/pypi/simple/>

* + - 1. 
    1. pip install ultralytics -i <https://mirrors.aliyun.com/pypi/simple/>
    2. 数据集中一共有很多张。但是由于设备限制我们只挑选200进行训练
       1. 代码展示



* + - 1. 代码说明

正如上面是代码所示，我只挑选了训练的图片的随机二百张图片。保证数据的离散型，保证模型对于不同车牌的图片识别的准确率。

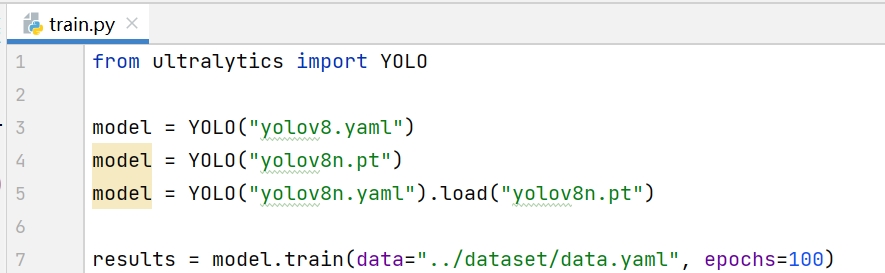
* + 1. 我们按照合理的比例进行划分数据集和测试集
       1. 代码展示



* + - 1. 代码说明

我按照0.75：0.25的比例进行划分75%的数据为训练集25%是数据是测试集

* + 1. <https://github.com/ultralytics/ultralytics/tree/main/ultralytics/cfg/models/v8>从这个网址下载此文件夹下的yolov8.yaml
    2. 编写训练脚本
       1. 代码如下



* + - 1. 代码解释

上面代码分别加载了yolov8的模型及v8的参数设置，我们运行模型，训练100次

* + 1. 编写data.yaml
       1. 内容如下

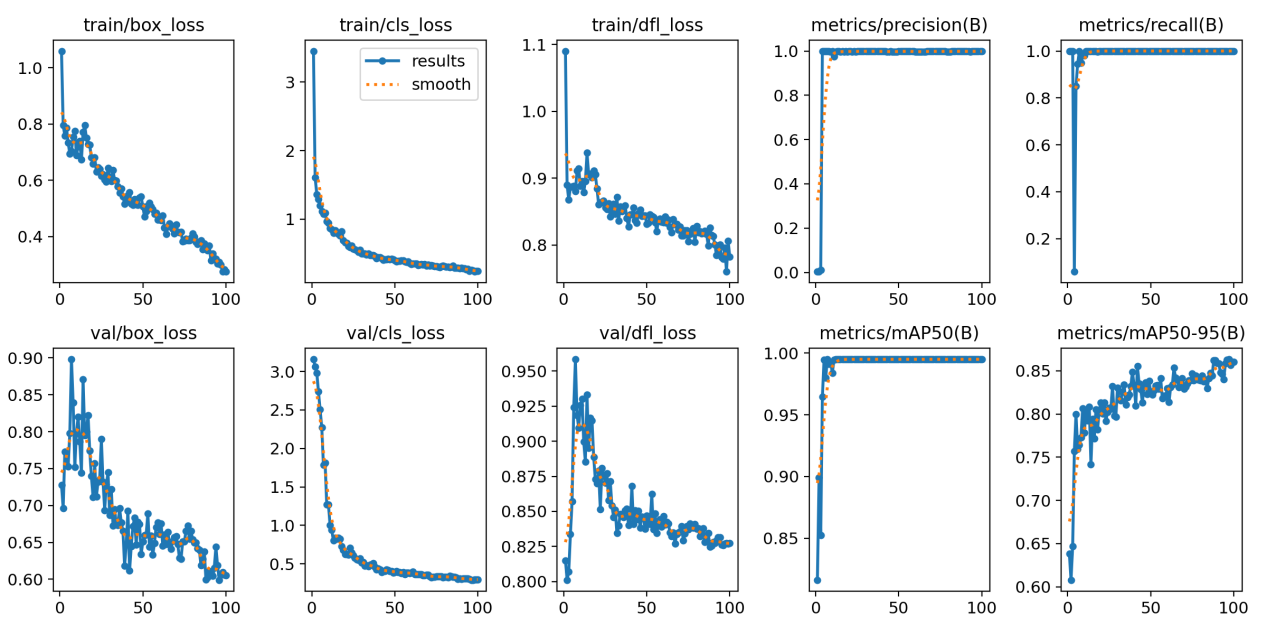
train: 训练集的绝对路径

val: 测试集的绝对路径

nc: 1

names: [‘platelince’]

* + 1. 开始训练运行我们上面编写的train.py
    2. 运行之后我们会看到在train.py同级目录下会出现一个runs文件夹里面还有一个detect文件夹里面还有一个文件夹train里面有weights，results.csv……我们只查看results.png



我们看到精确率和召回率已经达到100%，很好了我们就不在训练了。

1. 车牌识别模块
   1. 识别模块我是用paddleocr模块对车牌上的文字进行检测
   2. 流程大概就是使用yolov8检测出车牌的左上角和右下角坐标，使用cv2对图片进行裁剪，把裁剪图片让给ocr识别返回车牌上的文字
2. 效果展示
   1. 
   2. 