****

**课程报告**

专业班级： 机器人工程2班

学 号： 2023204150

姓 名： 曹峻齐

2024 年06月12日

实验1. 1）输入数字x（代指任意数字），绘制出x角形。2）课本程序练习题2.8。

程序实现思路

1)

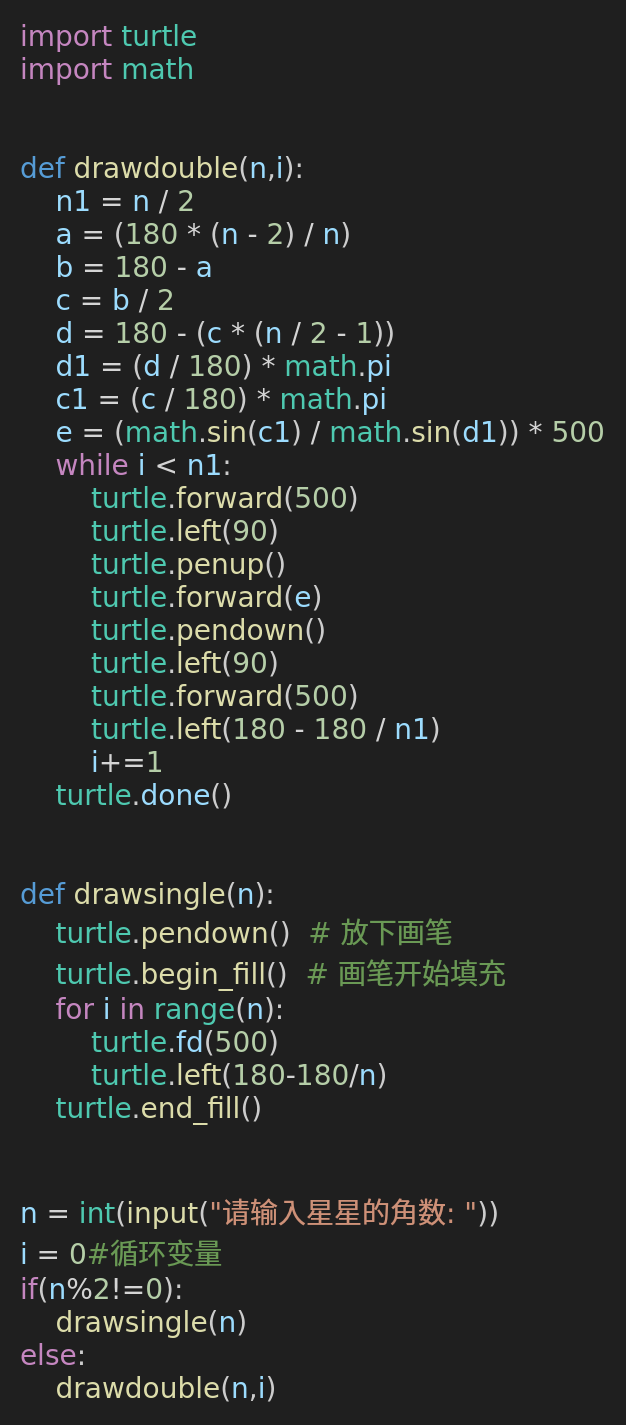
1. 定义绘制星形函数：首先，定义了两个函数drawdouble和drawsingle。drawdouble用于绘制偶数角数的星形，而drawsingle用于绘制奇数角数的星形。
2. 计算角度和距离：在drawdouble函数中，计算了每个内部角和内部边缘的长度。这些值用于在绘制星形时进行适当的旋转和移动。
3. 设置画笔速度和窗口大小：调用turtle.speed方法设置画笔速度，调用turtle.setup方法设置窗口大小。
4. 获取用户输入：使用input函数获取用户输入，询问用户希望绘制多少个角。
5. 检查角数并调用相应函数：根据用户输入的角数，检查它是否是偶数或奇数，并调用相应的绘制函数。
6. 绘制星形：在调用绘制函数之前，确保画笔已经放下并开始填充。在绘制过程中，使用循环来重复绘制每个边缘。

2)

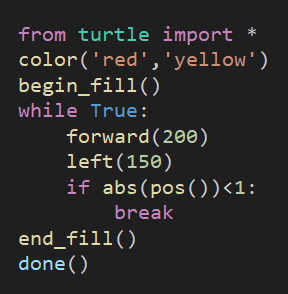
1. 定义绘制星形函数：首先，定义了两个函数drawdouble和drawsingle。drawdouble用于绘制偶数角数的星形，而drawsingle用于绘制奇数角数的星形。
2. 计算角度和距离：在drawdouble函数中，计算了每个内部角和内部边缘的长度。这些值用于在绘制星形时进行适当的旋转和移动。
3. 设置画笔速度和窗口大小：调用turtle.speed方法设置画笔速度，调用turtle.setup方法设置窗口大小。
4. 获取用户输入：使用input函数获取用户输入，询问用户希望绘制多少个角。
5. 检查角数并调用相应函数：根据用户输入的角数，检查它是否是偶数或奇数，并调用相应的绘制函数。
6. 绘制星形：在调用绘制函数之前，确保画笔已经放下并开始填充。在绘制过程中，使用循环来重复绘制每个边缘。

程序代码与说明

1）

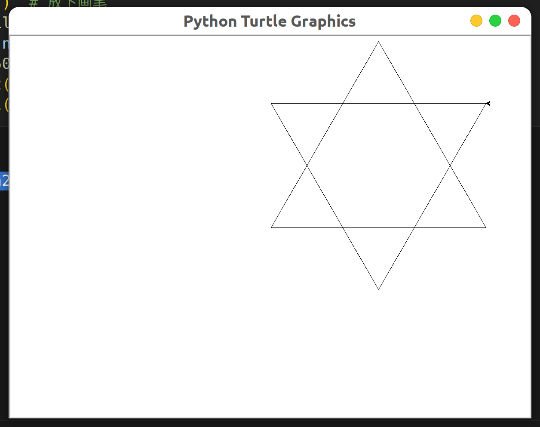
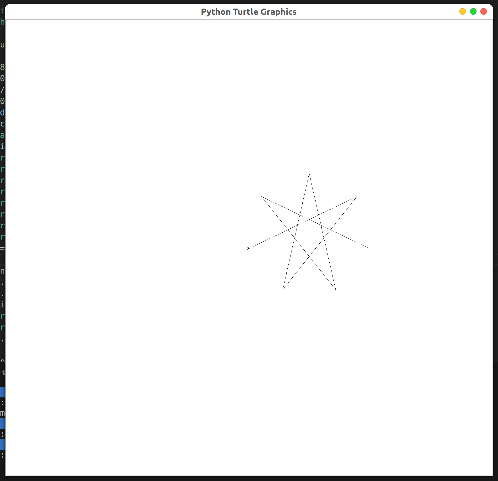


2）

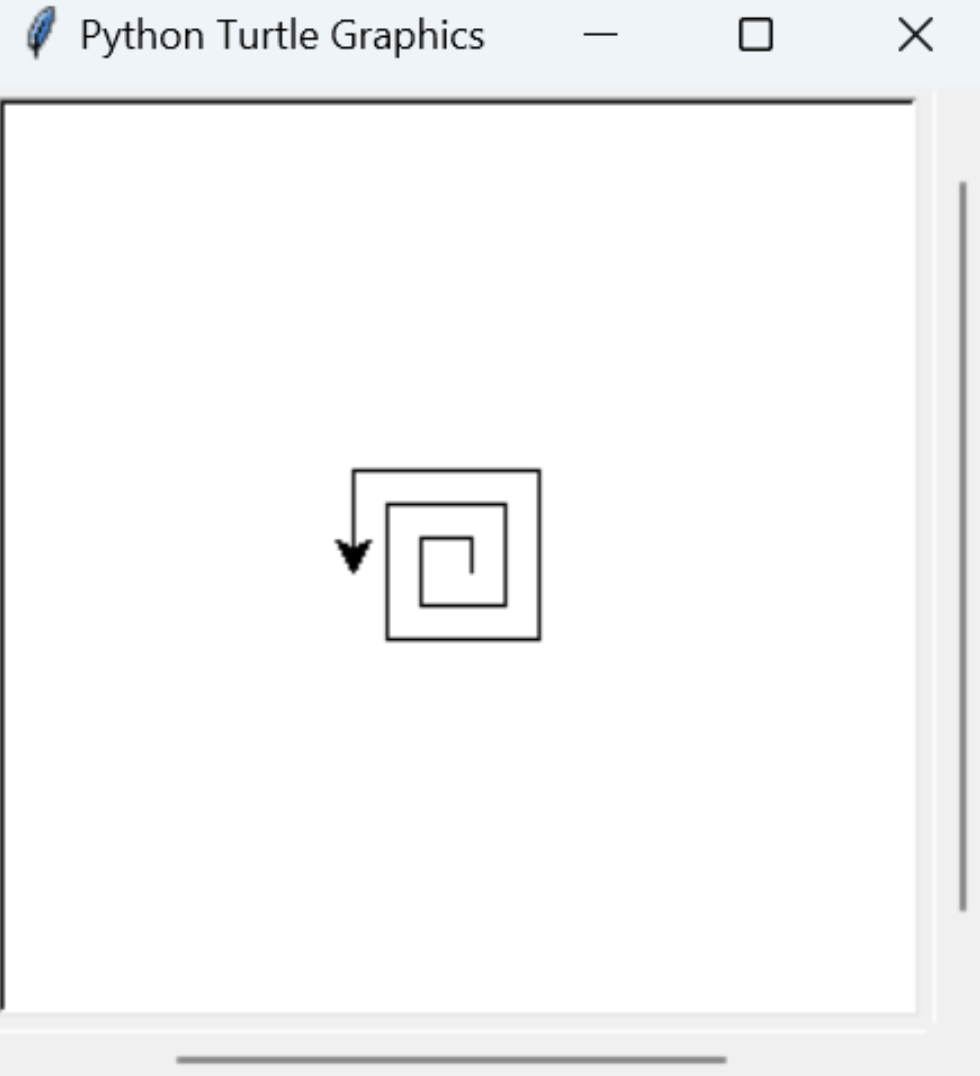


程序运行结果

1）

2）

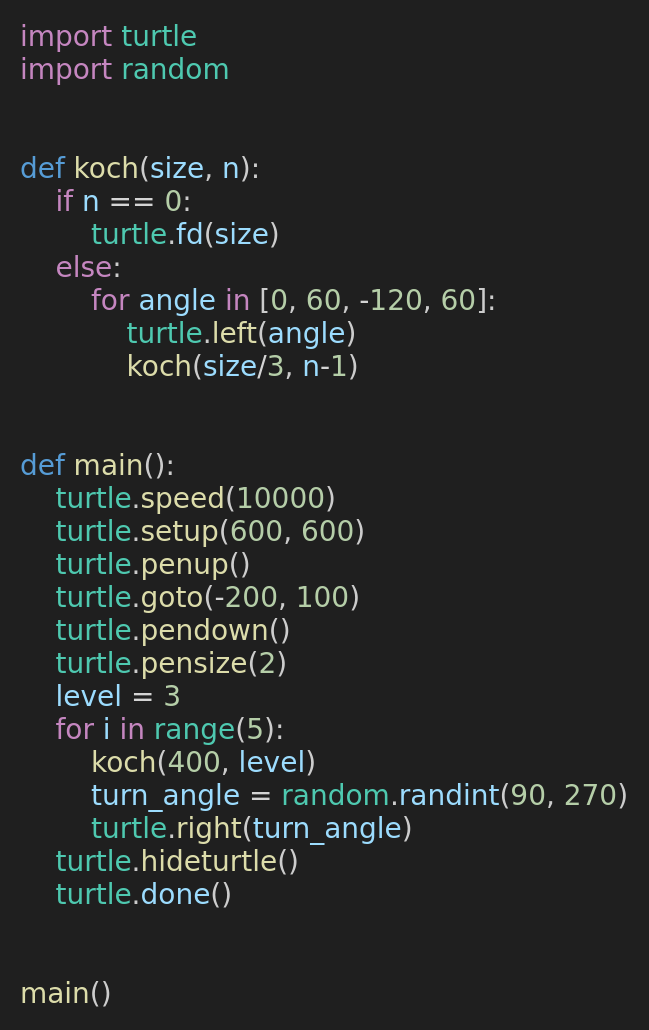


实验2 利用分形绘制海岸线

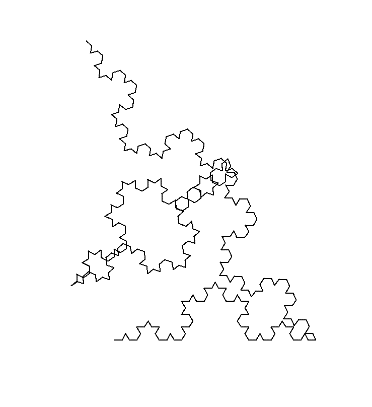
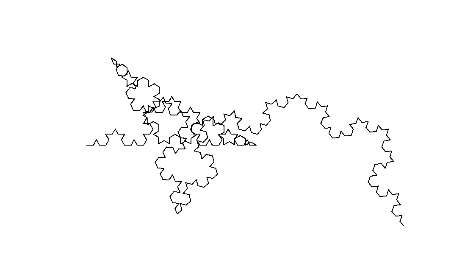
程序实现思路

1. 定义Koch曲线函数：首先，定义了一个名为koch的函数，用于绘制Koch曲线。这个函数接受两个参数：size（线段长度）和n（迭代次数）。如果n等于0，则直接画一条线段；否则，递归地绘制四个较小的Koch曲线。
2. 主函数：定义了一个名为main的函数，这是程序的入口点。
3. 设置画笔速度和窗口大小：调用turtle.speed方法设置画笔速度，调用turtle.setup方法设置窗口大小。
4. 移动画笔到起点：使用turtle.penup和turtle.goto方法移动画笔到起始位置。
5. 开始绘图：调用turtle.pendown方法开始绘图，并设置画笔宽度。
6. 绘制Koch曲线：使用一个循环来绘制多个Koch曲线，并在每次迭代后随机旋转画笔。
7. 隐藏画笔：调用turtle.hideturtle方法隐藏画笔。
8. 结束程序：调用turtle.done方法结束程序。

程序代码与说明



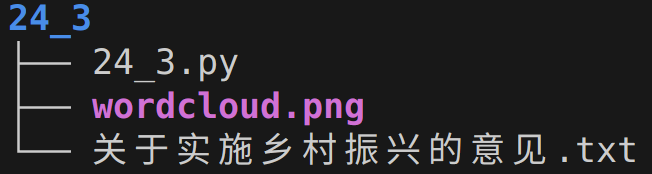
程序运行结果

实验3

程序实现思路

工作区树层结构



1. 加载库：首先，加载了`jieba`库用于中文分词，以及`wordcloud`库用于生成词云图像。

2. 读取文本文件：打开一个名为`关于实施乡村振兴的意见.txt`的文本文件，并读取其内容。

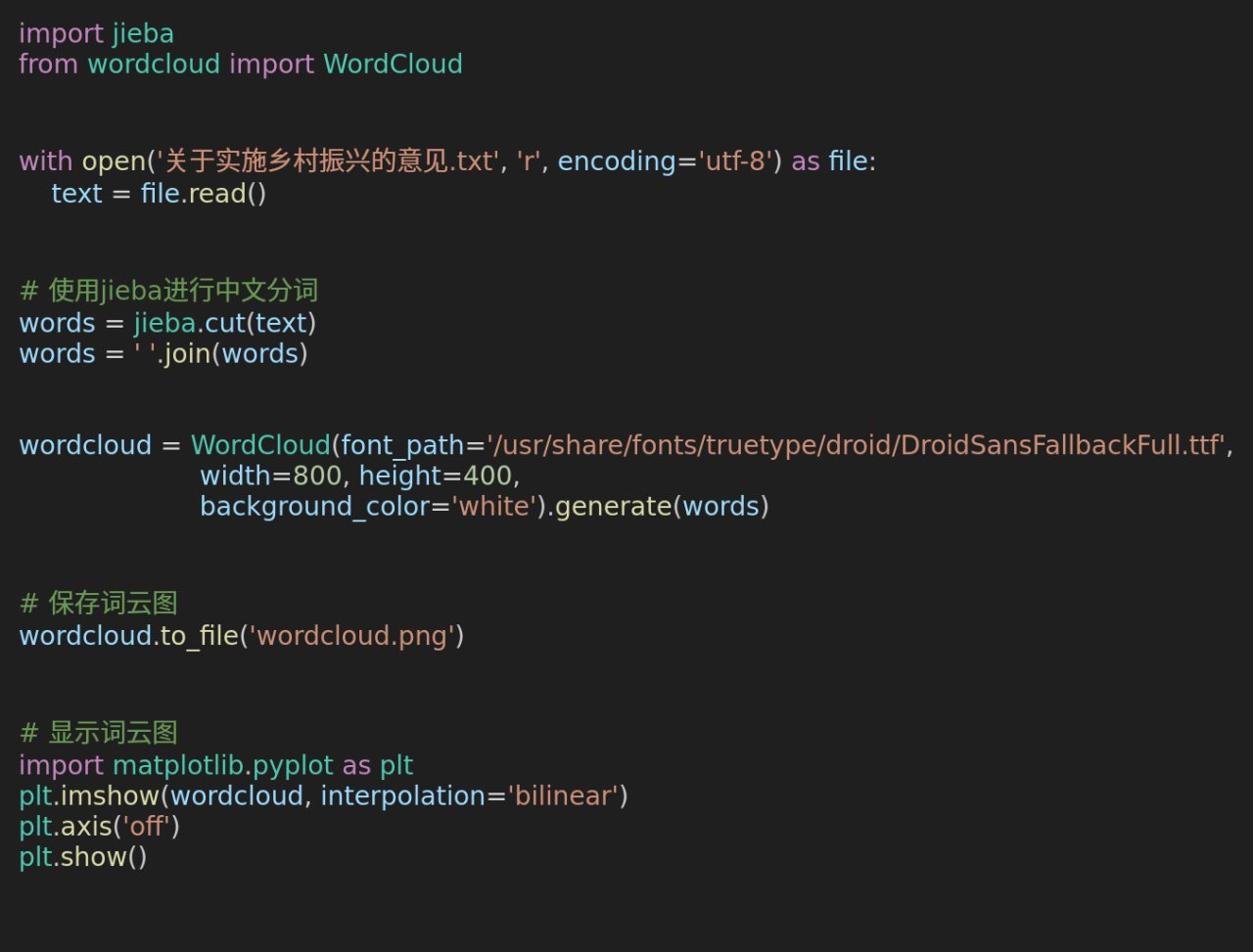
3. 中文分词：使用`jieba.cut`函数对文本进行中文分词，并将分词后的结果合并成一个字符串。

4. 生成词云图：使用`WordCloud`类创建一个词云对象，指定字体路径、图像宽度、高度和背景颜色。然后调用`generate`方法生成词云图像。

5. 保存词云图：使用`to\_file`方法将生成的词云图像保存到文件中。

6. 显示词云图：使用`matplotlib.pyplot`库来显示词云图像。调用`imshow`方法显示图像，设置插值方式为线性插值，并将轴设置为不可见。最后调用`show`方法来展示图像。

程序代码与说明



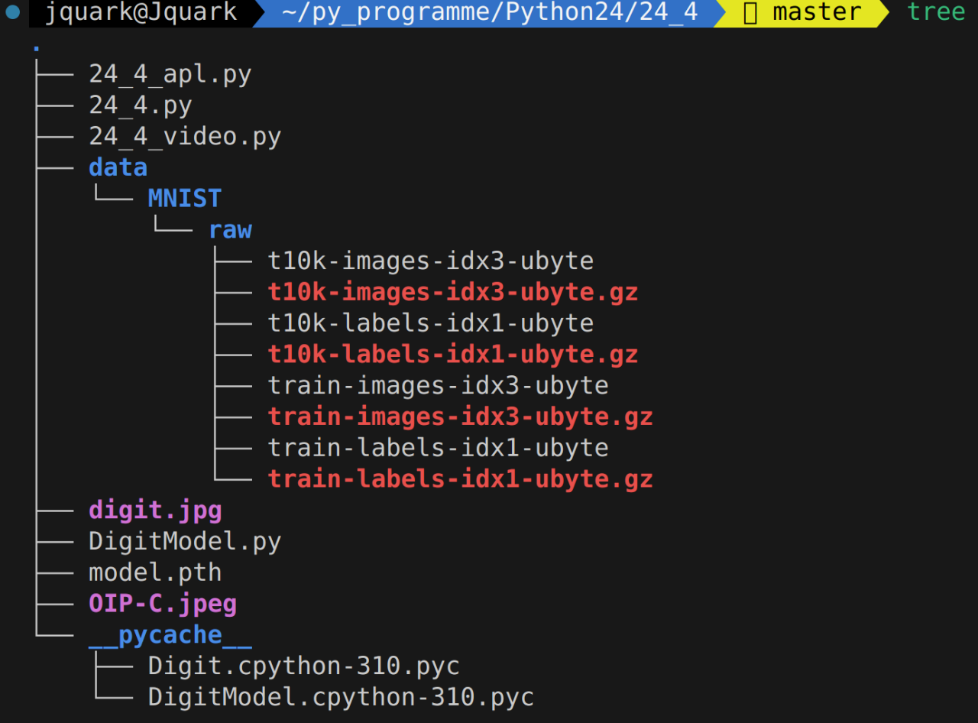
程序运行结果



实验4

程序实现思路

工作区树层结构



程序一: MNIST数据集手写数字模型的训练

1. 加载库：首先，加载了必要的Python库，包括torch、torch.nn、torch.optim、torchvision、matplotlib等。
2. 超参数设置：定义了训练过程中使用的超参数，如批量大小、设备选择（CPU或GPU）、训练轮次等。
3. 数据预处理：使用transforms.Compose构建了一个数据预处理流程，将图片转换为张量并进行正则化。
4. 数据加载：下载并加载MNIST手写数字的训练集和测试集。
5. 模型构建：定义了一个名为Digit的神经网络模型，包含卷积层和全连接层。
6. 优化器定义：选择了Adam优化器来更新模型的参数。
7. 训练函数：定义了一个函数来训练模型，包括前向传播、损失计算、反向传播和参数更新。
8. 测试函数：定义了一个函数来测试模型的性能，计算测试损失和准确率。
9. 模型保存：保存模型的状态到文件中。

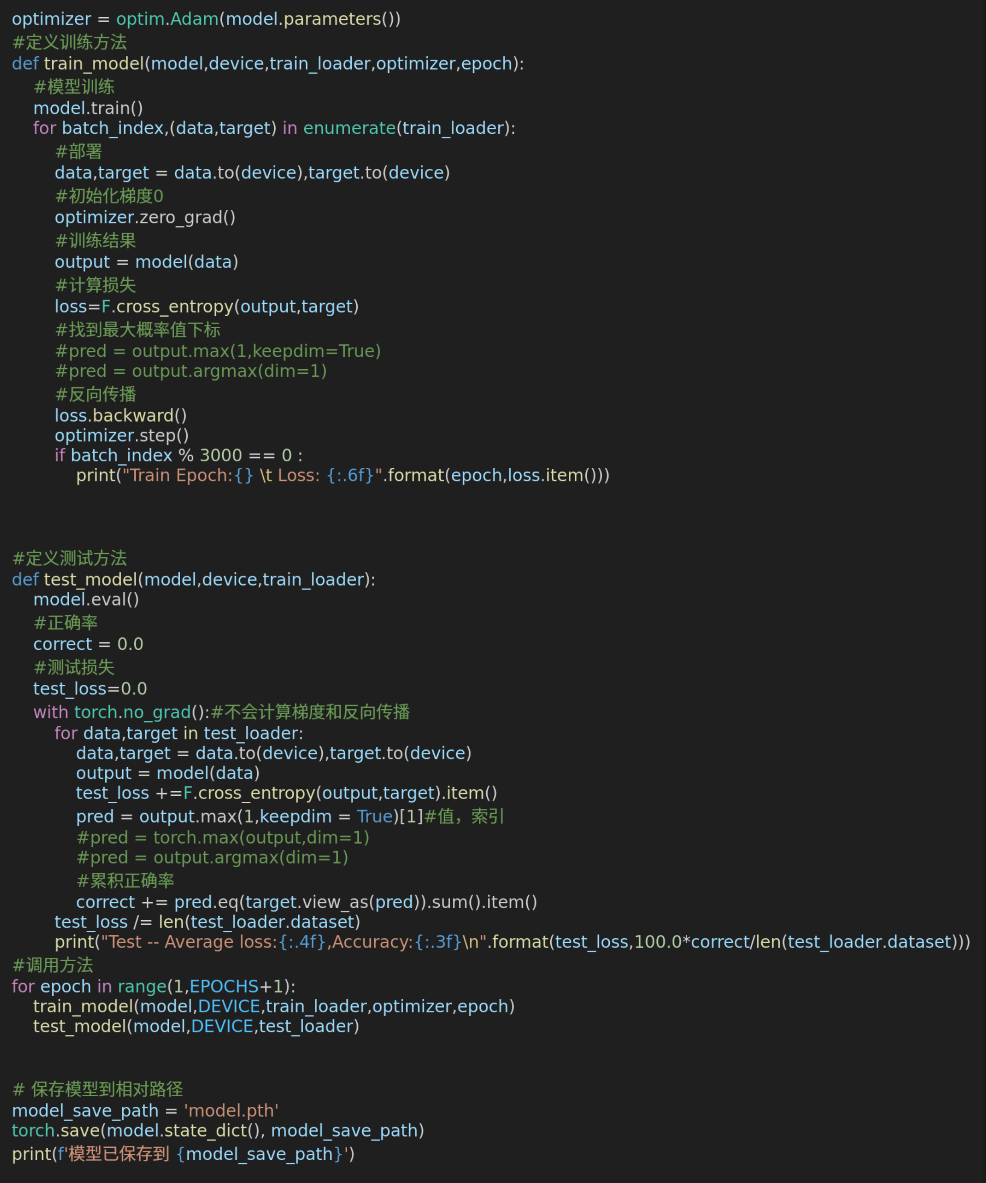
程序2与3为模型的应用，借助输出的pth模型文件，结合已有OpenCV基础，查阅网络资料，实现了对静态图像数字的分析和捕获视频中数字(动态效果仍然需要优化)

程序代码与说明

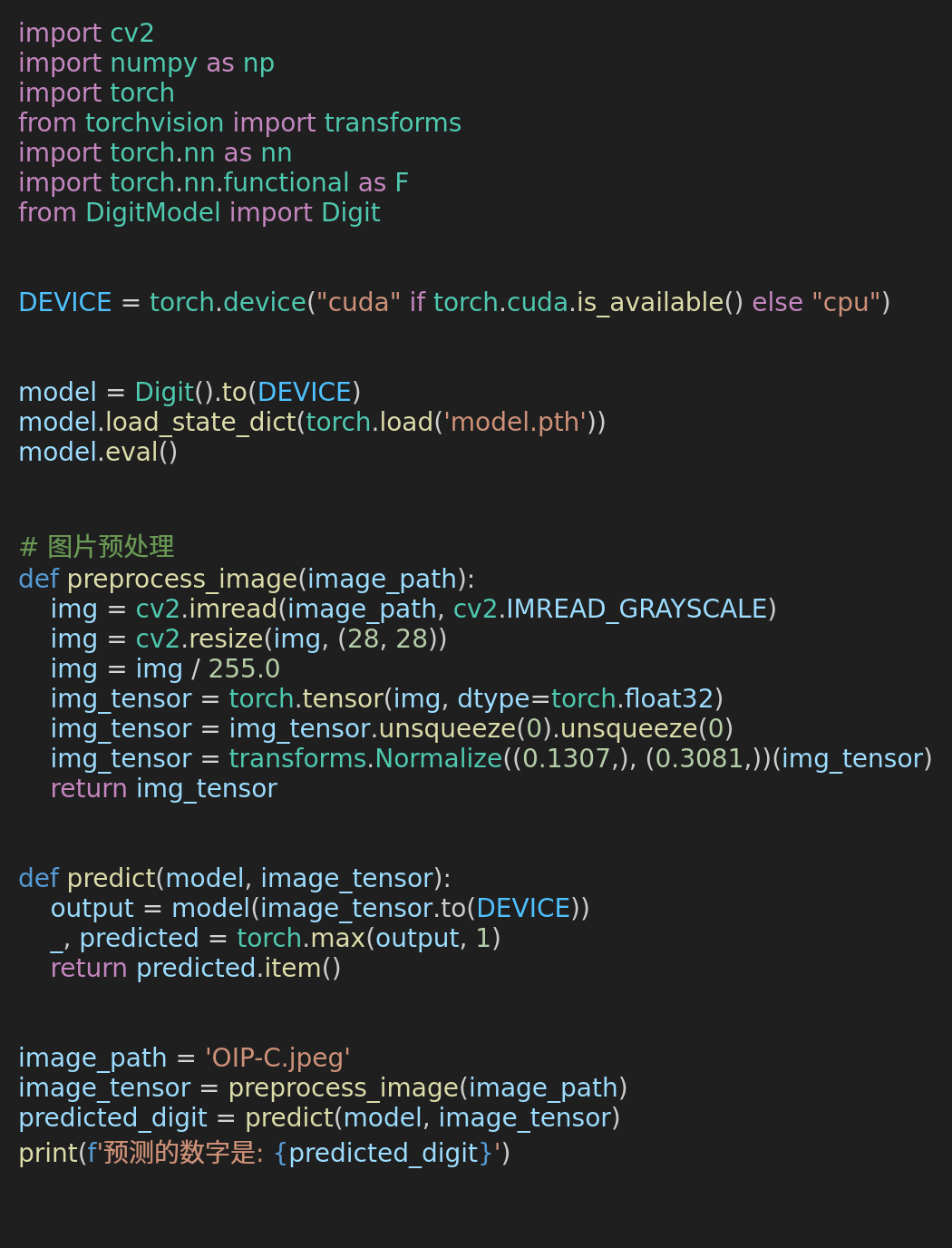
1)模型训练







2)本地图片数字识别



3)视频数字识别



程序运行结果

1)

2)



(识别对象)



(输出结果)

3)

