

**课程名称： 深度学习 课程代码： LR3002**

**任课老师： 实验指导老师:**

# 课程设计：经典卷积神经网络应用实战

学生姓名：

学号： 2240919249

教学班： BFV

递交日期：

签收人：

我申明，本报告内的实验已按要求完成，报告完全是由我个人完成，并没有抄袭行为。我已经保留了这份实验报告的副本。

申明人(签名):

**实验报告评语与评分：**

**评阅老师签名：**

## “深度学习”项目大作业：病羊风险预测系统

### 项目简介（背景、目的、意义）

背景：现代农业中，畜牧业的健康管理和疾病预防至关重要。绵羊作为重要家畜，其健康状况直接影响养殖效益和产品质量。传统诊断依赖兽医经验和现场检查，效率低且易受主观因素影响，难以实现大规模、快速、准确的疾病检测。随着计算机视觉和深度学习的进步，图像识别技术为自动检测动物疾病提供了新可能。

目的：我们想开发一个基于深度学习CNN的绵羊疾病自动检测系统，利用卷积神经网络（CNN）和定制化的ResNet架构，通过分析大量标注好的绵羊照片进行高效的疾病分类。

意义：自动化的绵羊疾病检测系统，能够显著提升诊断效率，快速处理大量图像，缩短诊断时间，帮助养殖户及时发现并处理疾病。同时，该系统减少了对专业兽医的依赖，降低了人工成本，特别适合资源有限的地区或小型养殖场。借助深度学习模型，系统通过不断积累数据优化判断逻辑，提供更准确的诊断结果，避免人为误判。

### 二．开发环境与原理

#### 2.1 开发环境

操作系统、IDE、解释器

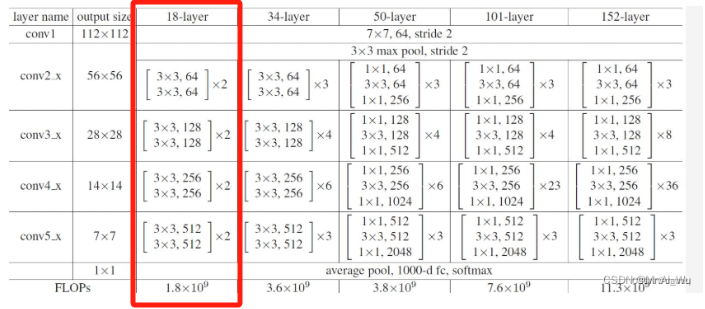
|  |  |  |
| --- | --- | --- |
| Winsows11 | Pycharm | Python3.9 |

Python库

|  |
| --- |
| Tensorflow2.6.0 |
| Pyqt5 |
| Opencv |
| Pillow |
| Numpy |
| Shutil |
| Matplotlib |
| Requests |
| Sns |

#### 2.2 原理

本项目主要使用的是Resnet18残差网络结构训练模型，该模型的网络结构为：



这个项目是基于tensorflow-gpu2.6版本

使用了CNN卷积神经网络的resnet18模型。

前期我尝试使用了resnet101、50、34、18网络，发现由于我手动获取到的病羊的数据集数量比较小（正常羊356张、不正常羊183张），非常容易使训练出来的模型梯度爆炸和过拟合，尝试后发现resnet18的效果最好，一开始的准确率只能达到0.7-0.85，所以我决定在resnet18的基础上再简化模型。

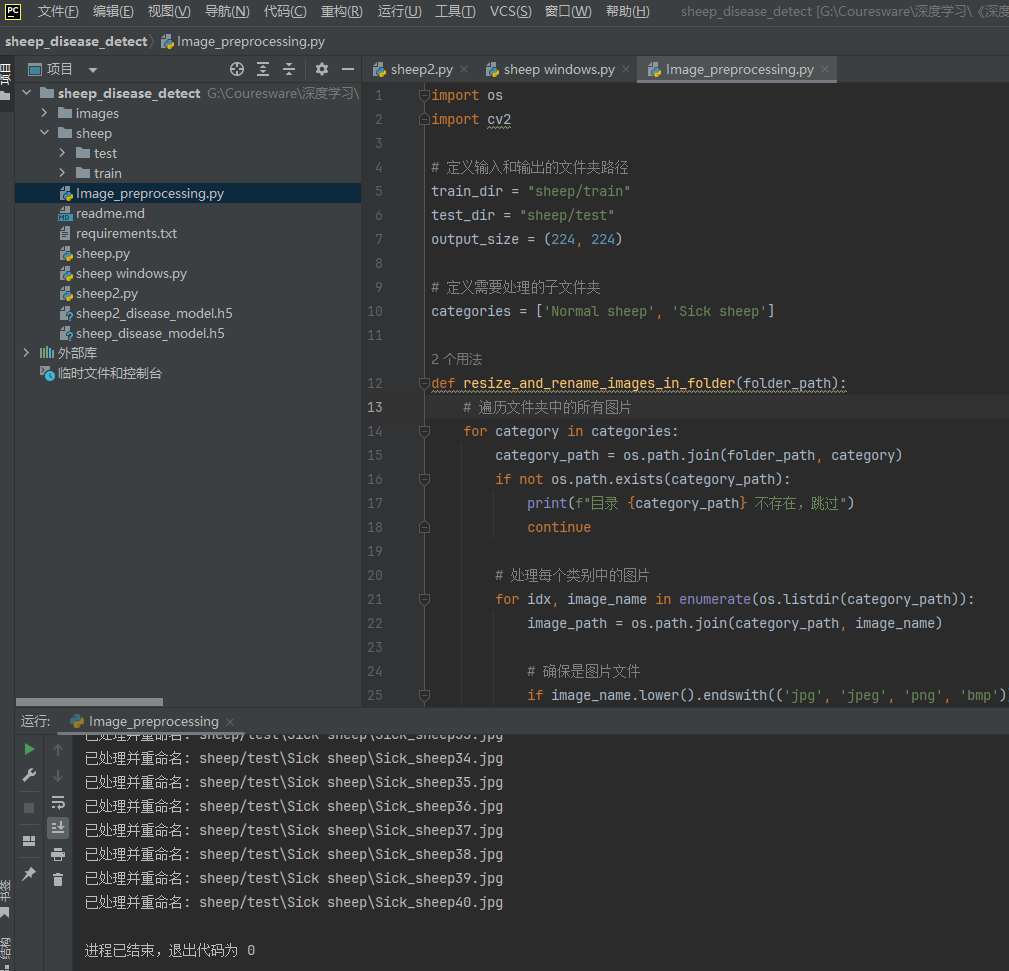
resnet18包含了八个残差块，分成四个阶段，每个阶段有两个残差块，**经过我的简化后，只留下了四个大块，每个大块中有两个基础残差块**。这些改动意味着他的表现能力不如原始的resnet18，但是在小数据集上足够了。

### 三．项目实现及结果

#### 3.1项目总体框架结构图

#### 3.2项目实现

1. 数据处理

Image\_preprocessing.py 使用了opencv工具把所有手动保存的图片更改为设定的图片格式和大小，并重新命名。把数据格式与模型保持一致。

1. 模型构建及训练
2. **数据预处理与增强**

为了提高模型的泛化能力和数据利用率，我们对图像数据进行了预处理和增强。使用了ImageDataGenerator来创建一个图像生成器，该生成器不仅能够将像素值归一化到[0, 1]区间，还实现了多种图像增强技术，包括随机水平翻转、垂直翻转、旋转、亮度调整、错切变换以及缩放等。此外，为了确保模型的评估不受训练过程的影响，我们将20%的数据划分作为验证集，用于监控模型在未见过的数据上的表现。对于测试集，我们只进行了归一化处理，以保证其原始特性不被改变，从而能真实反映模型的预测能力。

电脑萤幕截图

描述已自动生成

1. **模型架构设计**

模型基于ResNet（残差网络）结构构建，这是一种广泛应用于图像分类任务的深度学习模型。ResNet通过引入残差块解决了深层网络中的梯度消失问题，并加速了训练过程。我们的模型由四个大块组成，每个大块包含两个基本残差块。第一个大块中，如果它是整个网络的第一个大块，则会使用1x1卷积进行维度调整；其他大块在首个残差块处执行降采样操作，以逐步减少特征图的空间尺寸并增加通道数，从而捕获更高层次的特征。网络开始时有一个7x7的卷积层，随后是最大池化层，用于提取初步特征。最后，全局平均池化层将特征图转换为固定长度的向量，连接一个全连接层，使用sigmoid激活函数输出二分类结果。

**小残差块**

文本

描述已自动生成

**大残差块**

文本

描述已自动生成

**模型主体**

文本

描述已自动生成

1. **模型编译与优化**

模型使用Adam优化器，它是一种自适应学习率的方法，可以有效地处理稀疏梯度和非平稳目标。损失函数选择了二元交叉熵（binary\_crossentropy），这是因为我们的任务是二分类问题。同时，我们选择准确率（accuracy）作为评估指标，用以衡量模型预测的正确性。

电脑萤幕的截图

描述已自动生成

1. **学习率调度**

为了防止过早收敛或训练停滞，我们引入了学习率衰减策略。具体来说，当验证集上的损失值在连续5个epoch内没有显著下降时，学习率将以0.1的比例降低，以此帮助模型跳出局部最优解，继续探索更优的参数空间。



1. **模型训练**

模型训练共设置了500个epochs，即迭代次数。每次迭代，模型都会从训练集中抽取一定数量的样本（batch size = 32）进行前向传播和反向传播，更新网络参数。同时，在每个epoch结束时，模型会在验证集上评估性能，以监测是否出现过拟合现象。通过动态调整学习率，确保模型能够在不同阶段保持良好的学习效率。

文本

描述已自动生成

1. **模型评估**

训练完成后，我们在独立的测试集上评估了模型的表现，获得了最终的准确率。这个准确率反映了模型在新数据上的预测能力，是我们判断模型是否有效的重要依据。

图形用户界面, 文本

描述已自动生成

1. **模型保存**

训练好的模型被保存为H5文件格式，方便后续加载和部署。这使得我们可以随时恢复模型状态，而无需重新进行长时间的训练过程。



1. **训练过程可视化**

为了直观地了解模型训练过程中的变化趋势，我们绘制了训练集和验证集的准确率与损失曲线。这些图表有助于分析模型的学习动态，例如是否存在过拟合或欠拟合的问题，以及何时达到了最佳性能点。

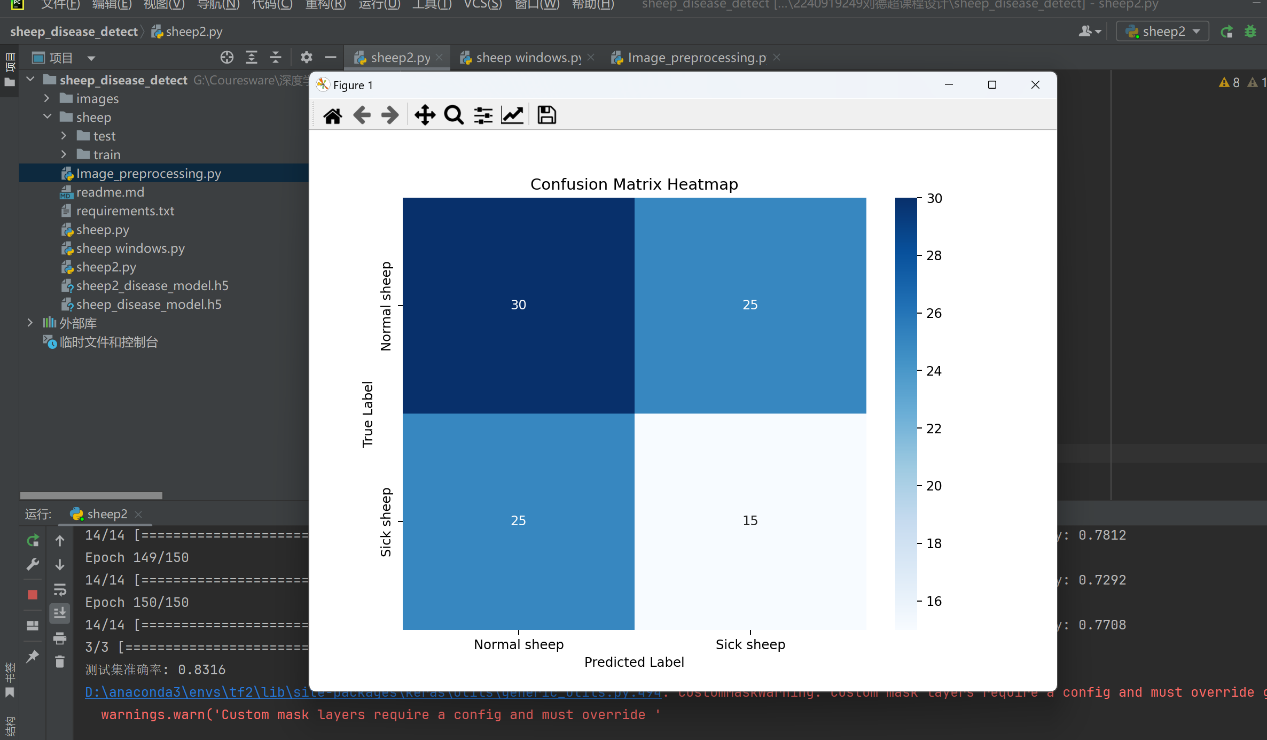
文本

描述已自动生成

下图是训练了100轮和500轮的模型训练过程中的准确率和损失的变化曲线。

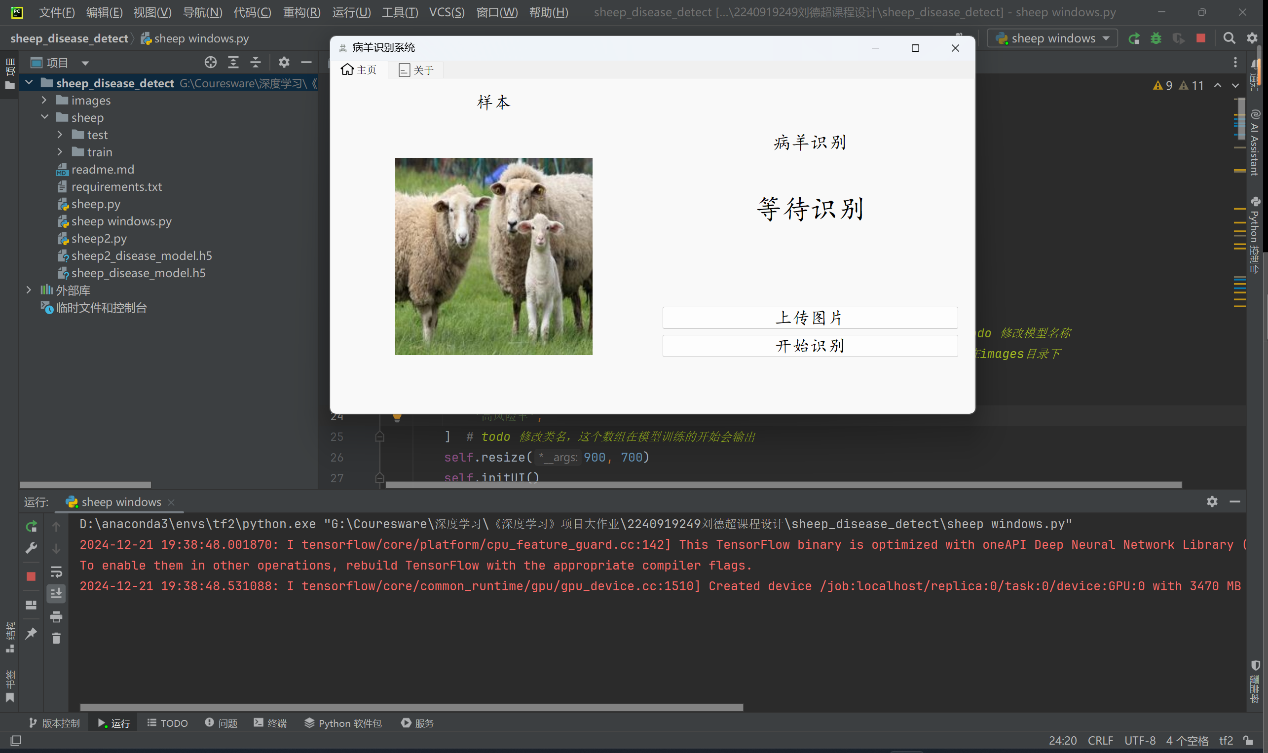
图形用户界面, 应用程序

描述已自动生成

图为matplotlib绘制测试准确率的热力图  


1. 模型运用及可视化展示

sheep windows.py是使用pyqt5编写的一个窗口程序，把输入的图片文件转化为numpy数组，再输入到模型中进行分类。运行sheep windows.py文件

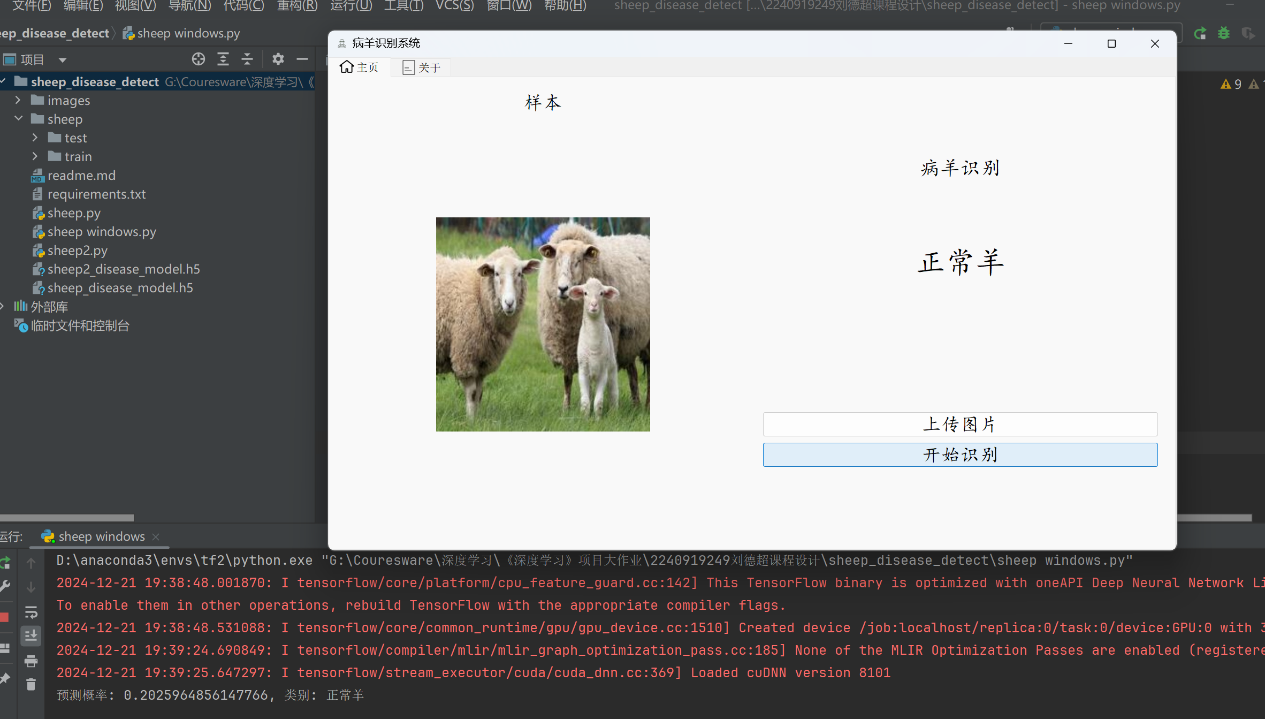


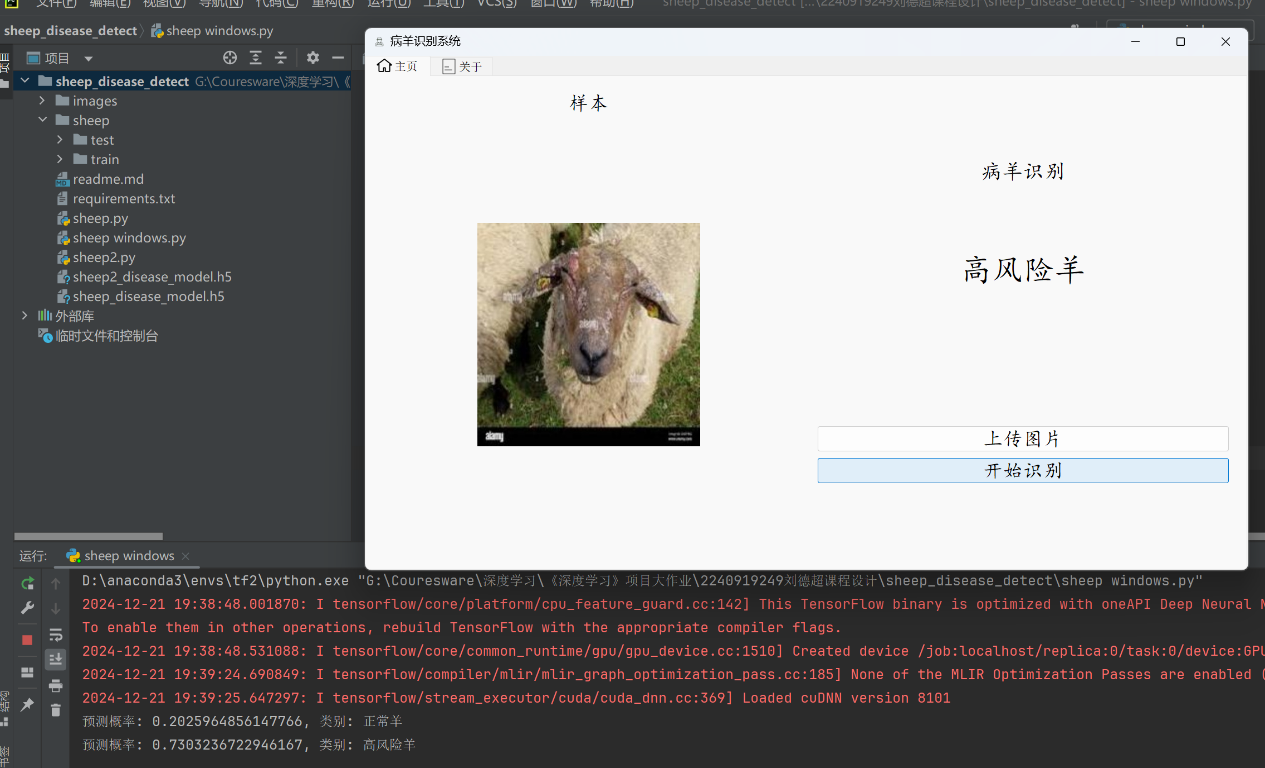
成功打开窗口

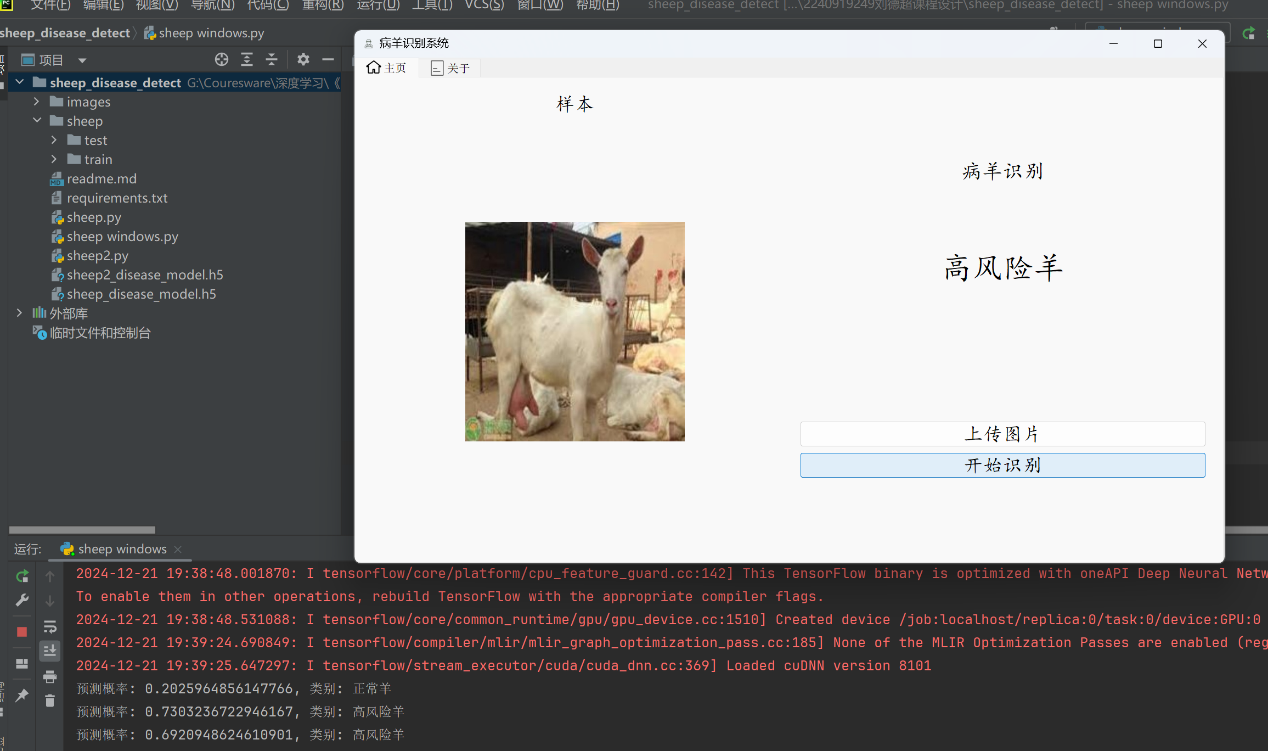
图形用户界面, 应用程序, 网站

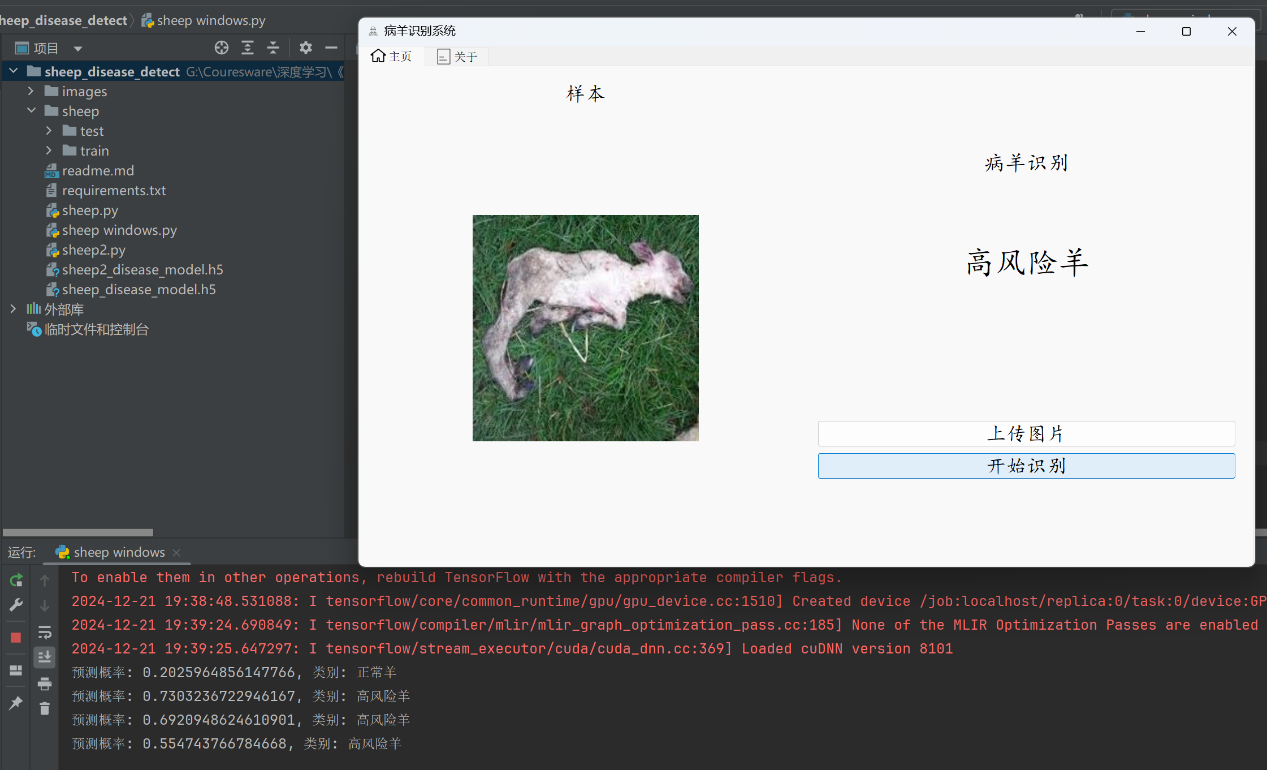
描述已自动生成

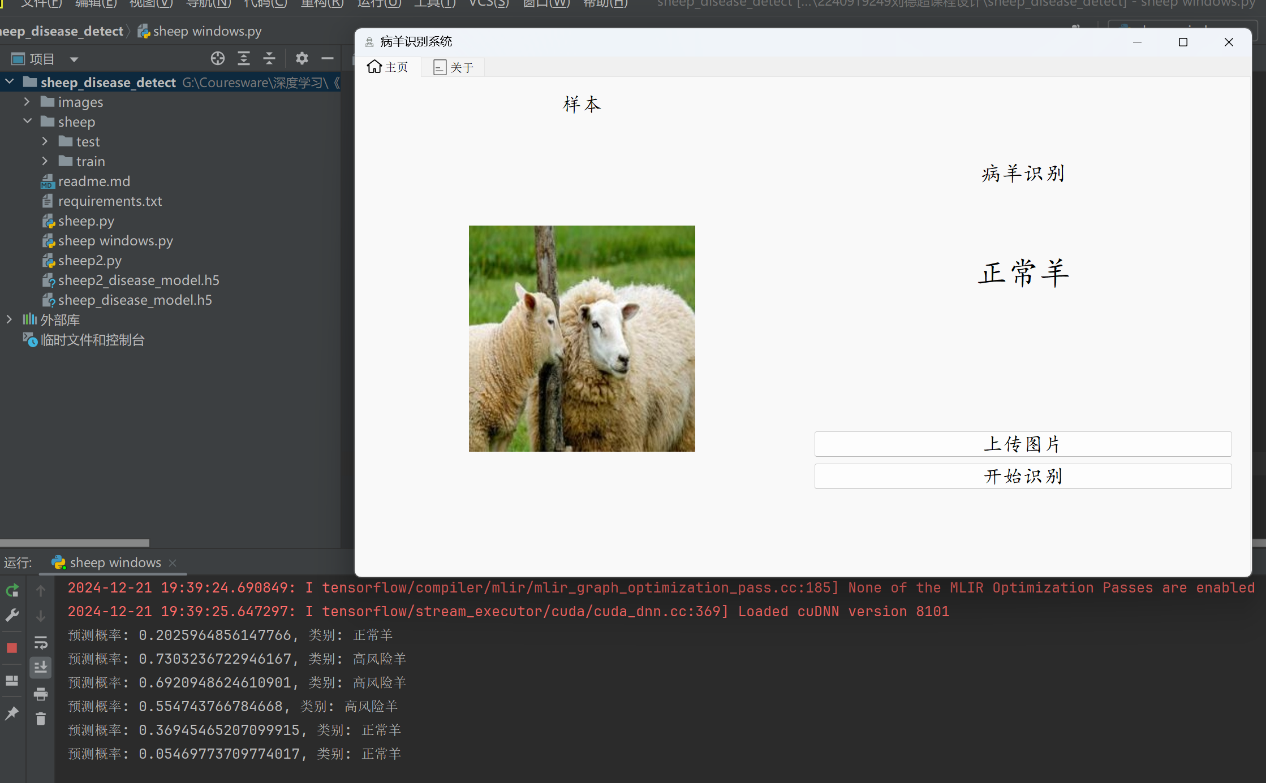
选择图片上传到并返回识别结果

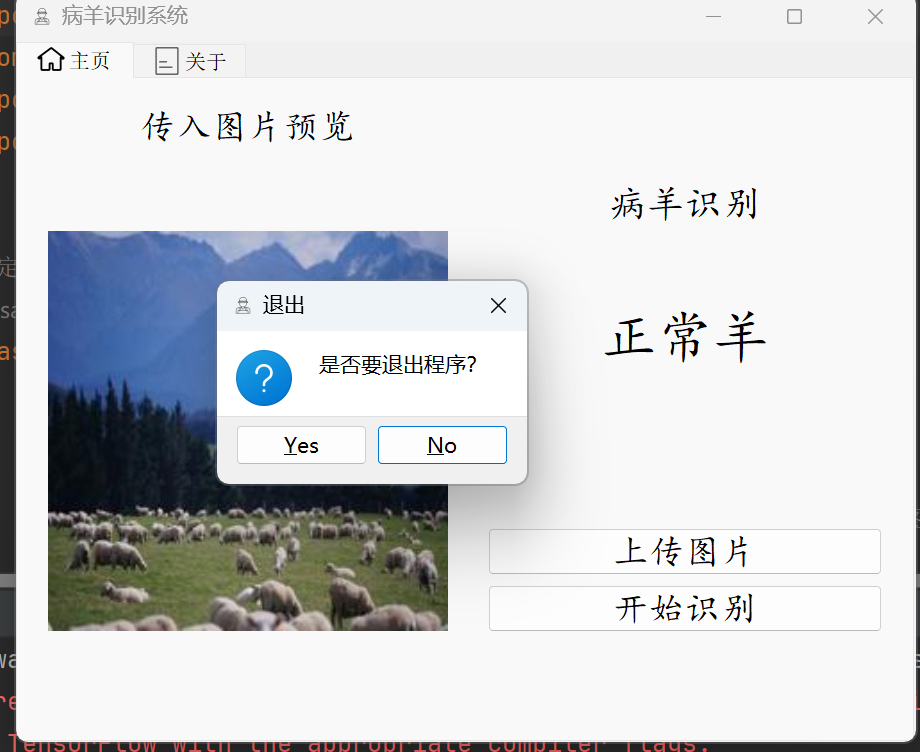
成功识别



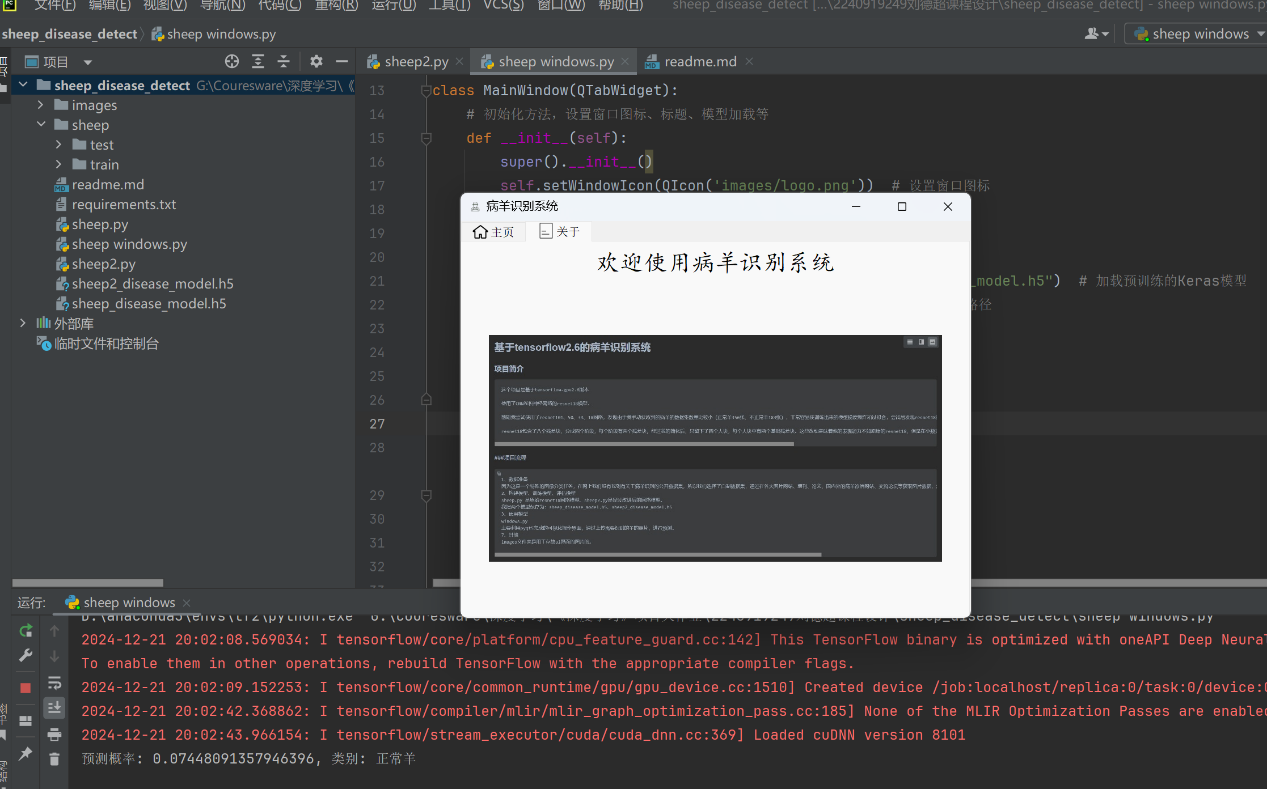








退出程序



介绍页

#### 3.3结果讨论

通过修改模型和清理数据集中的重复图片和不明显的图片，我们的模型准确率最高能达到90%的准确率，但是无论是加深网络的层数或者是对数据集进行数据增强还是无法使模型达到更高的准确率，我觉得主要的原因还是数据集数量不够，且有些数据图片的分辨率不够高。因为我们尝试了很多方法，不管是数据增强，交叉验证，在网络中加入CBAM注意力模块，把卷积快换成深度可分离卷积，但是这些方法出来的效果都不好，所以我们只能用更简单的网络模型去训练，不然的话就很容易出现梯度爆炸的情况，模型无法收敛。

### 四、项目总结与收获

#### 4.1 项目总结

本项目可对目前常见的绵羊身上发生的疾病进行风险预测，准确率为0.8726，损失率为0.2741。后续我们要增加数据集的数量优化数据集，这样才可以用更深层的卷积神经网络，当数据集达到一定量级我们就可以增加分类的数目，以实现不同疾病的预测，给人们更精确的疾病预警。

#### 4.2 心得体会

在完成病羊风险预测系统的过程中，我不仅深化了对深度学习技术的理解，也积累了很多经验。课堂上学到的卷积神经网络（CNN）和残差网络（ResNet）等知识，在实际项目中得到了应用。通过绵羊疾病识别，我意识到理论和实践相结合的重要性。理论为项目的顺利进行提供了基础，而实践则让我发现了许多书本上没有提到的问题，如数据集的局限性和模型过拟合。数据的质量和数量对模型性能的影响非常大。我们的病羊数据集较小（正常羊356张、不正常羊183张），导致模型容易过拟合。尽管我们尝试了多种数据增强方法，如翻转、旋转、亮度调整等，但数据量不足。初期我尝试了多个不同深度的ResNet模型（如ResNet-101、ResNet-50、ResNet-34、ResNet-18），发现对于小数据集，过于复杂的模型容易导致梯度爆炸和过拟合。最终，我选择了较为简单的ResNet-18，并进一步简化，保留了四个大块，每个大块包含两个基础残差块。这一简化提高了模型的收敛速度，避免了梯度爆炸。通过对学习率、批量大小（batch size）、损失函数等参数的调整，我找到了适合该项目的最佳配置。特别是引入了学习率衰减策略，当验证集上的损失值在连续5个epoch内没有显著下降时，学习率会以0.1的比例降低，防止了模型过早收敛或陷入局部最优解。使用Adam优化器和二元交叉熵损失函数也显著提升了训练效率和准确性。最后我使用PyQt5开发了一个图形用户界面（GUI），用户可以通过该界面上传图片并获得识别结果。这不仅增强了项目的实用性和交互性。通过这个项目，我掌握了深度学习的基本原理和应用技巧，培养了独立思考和解决问题的能力。我深刻体会到，深度学习不仅仅是一项技术，更是一种思维方式。它要求我们在面对复杂问题时，既要具备扎实的理论基础，又要灵活运用各种工具和方法，不断探索和创新。