安徽财经大学字

**本科毕业设计**

|  |  |
| --- | --- |
| **题 目** | **基于深度学习的垃圾分类器设计与实现** |
| **学 院** | **管理科学与工程学院** |
| **专 业** | **计算机科学与技术** |
| **班 级** | **20计科1班** |
| **学 号** | **20204133** |
| **姓 名** | **张雪** |
| **指导老师** | **张晓春** |

**2023 年 3 月**

安徽财经大学管理科学与工程学院

本科生毕业论文（设计）诚信承诺书

本人承诺：

1.所呈交的毕业论文（设计）《 毕业设计过程管理系统》，是在认真学习理解《安徽财经大学学位论文作假行为处理办法》和《管理科学与工程学院本科毕业论文（设计）工作管理办法》后，保质保量独立完成的，没有弄虚作假，没有抄袭别人的内容；

2.毕业论文（设计）所使用的相关资料、数据、观点等均真实可靠，文中所有引用的他人观点、材料、数据、图表均已注释说明来源；

3.毕业论文（设计）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，伪造、篡改数据的情况；

4.本人已被告知并清楚：学院对毕业论文（设计）中的抄袭、剽窃、弄虚作假等违反学术规范的行为将严肃处理，并可能导致毕业论文（设计）成绩不合格，无法正常毕业、取消学士学位资格或注销并追回已发放的毕业证书、学士学位证书等严重后果；

5.若在省教育厅、学校、学院组织的毕业论文（设计）检查中，被发现有抄袭、剽窃、弄虚作假等违反学术规范的行为，本人愿意接受学院按有关规定给予的处理，并承担相应责任。

学生（签名）：

年 月 日

指导老师（签名）：

年 月 日

**基于深度学习的垃圾分类器设计与实现**

**摘 要**

近年来，我国在各大城市通过推行政策以及相关法律法规来推进垃圾分类。本文在现有的垃圾分类倡导之下，基于深度学习对生活垃圾类别的检测进行研究，并结合研究成果设计了生活垃圾分类识别系统，辅助人们完成生活垃圾分类，推动垃圾分类政策的落实。本文的主要研究内容如下:

1. 本文针对深度学习模型中虽然网络层数越深理论上表达能力会更强，但CNN网络达到一定的深度后，再加深其分类性能不会提高，而是会导致网络收敛更缓慢，准确率也会随之降低的问题，引入基于残差学习方法的RestNet50深度学习模型。ResNet50使用了残差块来实现残差学习。每个残差块包含两个卷积层和一个跳跃连接。跳跃连接将输入直接传递到输出，从而避免了梯度消失的问题。同时，RestNet50使用了全局平均池化层，之歌层将每个特征图的所有像素的平均值作为该特征图的输出，能够减少模型的参数数量，从而减少过拟合的风险。最后，通过实验验证，本文采取的方法对于模型的准确率有着明显地提升。

（2）本文面向日常垃圾分类的实际使用场景，对于生活垃圾分类的相关需求进行了细致地调研、分析和细化，并根据系统需求分析对系统进行详细设计与实现。结合本文的研究成果，使用PyCharm和pytorch框架设计并实现了垃圾分类系统，将垃圾分为有害垃圾、厨余垃圾、可回收垃圾和其他垃圾四种。

**关键词：**垃圾分类，深度学习，ResNet50

**Design and implementation of rubbish classifier based on deep learning**

**Abstract**

In recent years, my country has promoted rubbish classification by implementing policiesand relevant laws and regulations in major cities. Under the current advocacy of rubbish classification,this paper studies the detection of domestic rubbish categories based on deeplearning, and designs a domestic rubbish classification and identification system based on theresearch results to assist people in completing domestic rubbish classification and promote theimplementation of rubbish classification policies. The main research contents of this paper areas follows:

(1) This article introduces the RestNet50 deep learning model based on residual learning method to address the problem that although the deeper the number of network layers, the stronger the theoretical expression ability in deep learning models. However, deepening the CNN network to a certain depth will not improve its classification performance, but will lead to slower network convergence and reduced accuracy. ResNet50 uses residual blocks to achieve residual learning. Each residual block contains two convolutional layers and one skip connection. Jump connections transfer input directly to output, thus avoiding the problem of gradient vanishing. At the same time, RestNet50 uses the global average pooling layer, and the Zhige layer takes the average value of all pixels of each feature map as the output of the feature map, which can reduce the number of parameters of the model, thus reducing the risk of overfitting. Finally, through experimental verification, the method adopted in this article has significantly improved the accuracy of the model.

(2) This article focuses on the practical usage scenarios of daily garbage classification, conducts detailed research, analysis, and refinement of the relevant requirements for household garbage classification, and designs and implements the system in detail based on the system requirements analysis.

**Keywords:Rubbish Classification,Deep Learning,ResNet50**

# 目 录

[摘 要 I](#_Toc130328637)

[Abstract II](#_Toc130328638)

[目 录 III](#_Toc130328639)

[1 引言 - 5 -](#_Toc130328640)

[2 可行性研究 - 5 -](#_Toc130328641)

[2.1 需求概述 - 5 -](#_Toc130328642)

**[2.1.1系统操作流程](#_Toc130328643)** [- 6 -](#_Toc130328643)

**[2.1.2功能和特色](#_Toc130328644)** [- 6 -](#_Toc130328644)

**[2.1.3系统流程图](#_Toc130328645)** [- 6 -](#_Toc130328645)

[2.2 系统可行性 - 7 -](#_Toc130328646)

**[2.2.1经济可行性](#_Toc130328647)** [- 7 -](#_Toc130328647)

**[2.2.2技术可行性](#_Toc130328648)** [- 10 -](#_Toc130328648)

**[2.2.3操作可行性](#_Toc130328649)** [- 10 -](#_Toc130328649)

**[2.2.4法律可行性](#_Toc130328650)** [- 10 -](#_Toc130328650)

[2.3 项目进度计划 - 10 -](#_Toc130328651)

[3 需求分析 - 12 -](#_Toc130328652)

[3.1 系统综合需求 - 12 -](#_Toc130328653)

**[3.1.1功能需求概述](#_Toc130328654)** [- 12 -](#_Toc130328654)

**[3.1.2系统性能需求概述](#_Toc130328655)** [- 12 -](#_Toc130328655)

**[3.1.3其他需求概述](#_Toc130328656)** [- 13 -](#_Toc130328656)

[3.2 数据流图分析 - 13 -](#_Toc130328657)

**[3.2.1顶层数据流图](#_Toc130328658)** [- 13 -](#_Toc130328658)

**[3.2.2功能级数据流图](#_Toc130328659)** [- 13 -](#_Toc130328659)

**[3.2.3数据流图细化与分解](#_Toc130328660)** [- 13 -](#_Toc130328660)

[3.3 数据字典 - 13 -](#_Toc130328661)

**[3.3.1数据流条目定义](#_Toc130328662)** [- 13 -](#_Toc130328662)

**[3.3.2数据存储条目定义](#_Toc130328663)** [- 14 -](#_Toc130328663)

**[3.3.3数据处理储条目定义](#_Toc130328664)** [- 14 -](#_Toc130328664)

**[3.3.4数据项条目定义](#_Toc130328665)** [- 15 -](#_Toc130328665)

[3.4实体联系分析 - 15 -](#_Toc130328666)

**[3.4.1实体提取及实体图](#_Toc130328667)** [- 15 -](#_Toc130328667)

**[3.4.2实体联系图](#_Toc130328668)** [- 16 -](#_Toc130328668)

[4系统设计 - 17 -](#_Toc130328669)

[4.1 总体设计 - 17 -](#_Toc130328670)

**[4.1.1系统功能结构](#_Toc130328671)** [- 17 -](#_Toc130328671)

**[4.1.2系统层次图](#_Toc130328672)** [- 18 -](#_Toc130328672)

[4.2 详细设计 - 18 -](#_Toc130328673)

**[4.2.1代表性模块设计](#_Toc130328674)** [- 18 -](#_Toc130328674)

**[4.2.2系统数据库设计](#_Toc130328675)** [- 19 -](#_Toc130328675)

[5 编码与测试 - 20 -](#_Toc130328676)

[5.1 编码 - 20 -](#_Toc130328677)

**[5.1.1编码规则简介](#_Toc130328678)** [- 20 -](#_Toc130328678)

**[5.1.2代表性模块示例](#_Toc130328679)** [- 20 -](#_Toc130328679)

[5.2 测试 - 20 -](#_Toc130328680)

**[5.2.1白盒测试](#_Toc130328681)** [- 20 -](#_Toc130328681)

**[5.2.2黑盒测试](#_Toc130328682)** [- 20 -](#_Toc130328682)

[6 系统使用说明 - 21 -](#_Toc130328683)

[6.1 系统运行环境和配置 - 21 -](#_Toc130328684)

[6.2 系统操作说明（按照结构图或层次图的框架依次介绍） - 21 -](#_Toc130328685)

**[6.2.1 XX1模块说明](#_Toc130328686)** [- 21 -](#_Toc130328686)

**[6.2.2 XX2模块说明](#_Toc130328687)** [- 21 -](#_Toc130328687)

**[6.2.3 XX3模块说明](#_Toc130328688)** [- 21 -](#_Toc130328688)

**[6.2.4 XX4模块说明](#_Toc130328689)** [- 21 -](#_Toc130328689)

[7 总结 - 22 -](#_Toc130328690)

[参考文献 - 23 -](#_Toc130328691)

[致谢 - 24 -](#_Toc130328692)

# 1 引言

近年来，随着国家的经济高速发展，工业化与城市化的进程不断向前推进，我国人民的衣、食、住、行等各个方面的生活水平提升显著，与此同时人们的日常生活的物质需要也在不断提高，我国生活垃圾的总量近些年来逐年递增。截止到2021年，我国的垃圾堆存量已经超过了65亿吨并保持上涨趋势，通过对全国 600多个城市进行调查发现一多半的大中城市都已被垃圾包围[1]。目前，我国对于大多数生活垃圾的处理方式还比较传统，主要有三种:卫生填埋、垃圾堆肥以及垃圾焚烧[2]。其中卫生填埋不仅对环境造成严重污染，而且垃圾中的有害物质还可能对人们的健康造成威胁，垃圾堆肥需要对垃圾进行分类再处理，难度较高，而垃圾焚烧不会对人们的健康产生威胁，并且易于实现，而且通过回收其中产生的热量，可以使得垃圾成为可利用资源，但是对于环境还是会造成严重污染。要想真正解决“垃圾围城”这一问题，将垃圾进行合理的分类处理才是关键[3]。我国对于垃圾分类的相关工作非常重视，2019年习总书记指出推行垃圾分类，关键是要加强科学管理、形成长效机制、推动习惯养成，随后上海率先通过并实施《上海市生活垃圾管理条例》，成为国内生活垃圾分类的示范与标杆，其他各大城市纷纷响应，垃圾分类开始进入法制强制时代[4]。

但是，当前城市生活垃圾分类不仅要靠法治强制，还需要提高垃圾分类源头分类参与率，提高生活垃圾分类政策的执行效率，推动城市垃圾分类长期有效地进行。目前主要采用两种方法来推动城市居民参与垃圾分类，其一为通过专业人员对垃圾的分类进行监督和督促;其二为通过发放垃圾分类手册给居民普及垃圾分类知识。然而推广这两种方法具有一定的困难性，一方面通过专业人员对每位居民监督成本太高，另一方面居民日常产生的生活垃圾种类繁杂，要求居民短期内快速掌握熟悉生活中大多数垃圾类别比较困难，而每次垃圾分类都需要查阅相关信息等方式会很大程度上增加居民进行垃圾分类的时间成本，从而影响到居民垃圾分类的积极性。基于以上考虑，需要一种成本较低并能够降低居民垃圾分类成本的方式来推动垃圾分类的工作。

近些年来，随着深度学习取得了突破性的进展，越来越多的技术开始基于深度学习向人们提供各种服务。例如:语音识别、人脸检测、汽车的自动驾驶等。而当前的垃圾分类问题也可以通过深度学习来解决，通过深度学习对于垃圾的类别进行快速而准确地识别判断，极大降低了垃圾分类的成本。当前已知的垃圾分类系统主要分为四种:投放智能垃圾桶，建立垃圾分类监督平台、智能垃圾分拣、垃圾分类识别系统。深度学习技术可以通过微信、支付宝小程序来实现辅助人们生活垃圾分类的模式创新[5]。以微信为例，目前微信上已经出现了很多通过应用图像分类技术实现的垃圾分类小程序来辅助居民进行生活垃圾分类，但是由于上述系统采用图像分类技术进行实现，使得每次只能对一种类别的垃圾进行识别，大大降低了垃圾分类的效率，本系统通过将基于残差学习方法的RestNet50深度学习模型引入到垃圾分类系统中，个人便可以利用本系统进行垃圾分类，对于人们生活垃圾分类的效率显著提升，增加人们对于垃圾分类的积极性。

本文旨在针对深度学习模型中虽然网络层数越深理论上表达能力会更强，但CNN网络达到一定的深度后，再加深其分类性能不会提高，而是会导致网络收敛更缓慢，准确率也会随之降低的问题，通过对ResNet50模型进行优化来解决该问题，再依据该方法建立一个垃圾分类系统，辅助用户对日常垃圾进行分类，将会使得用户在生活中垃圾分类的难度降低，提升用户对垃圾分类热情，有利于推动垃圾分类的具体实施，间接地减轻垃圾中转站二次分拣的工作成本，降低垃圾处理的成本，还可以与其他的方案进行配合，从而更好地推进垃圾分类，最终解决日渐严重“垃圾围城”和频发的垃圾冲突问题，改善生态，提高资源利用率。

# 2 可行性研究

随着垃圾数量的不断增加，垃圾分类成为一个越来越重要的问题。为了解决这个问题，基于深度学习的垃圾分类器已经成为了一个备受关注的研究领域。这个研究领域的目标是设计一种可以自动识别和分类垃圾的模型，使得人们可以更加方便地进行垃圾分类。本章节将综述基于深度学习的垃圾分类器的可行性研究。

## 2.1 需求概述

随着人们对环境保护意识的不断提高，垃圾分类已经成为一项全民行动。然而，人工分类效率低、成本高，垃圾分类的精度和效率都有待提高。基于深度学习的垃圾分类器可以有效地解决这一问题，实现快速、准确地垃圾分类。该垃圾分类器可以通过摄像头或者其他传感器获取垃圾图像或数据，利用深度学习算法对垃圾进行分类，从而实现高效、准确的垃圾分类。

### **2.1.1 系统操作流程**

开发垃圾图像分类系统的具体技术路线是:通过调取本地图片或线上图片获取待分类垃圾数据，然后切分出图片帧，通过图像算法确定图片中垃圾的类别，并将这些信息提供给搜索用户，使用户能够准确地进行垃圾分类。

### **2.1.2 功能和特色**

自动化识别和分类垃圾：垃圾分类器可以自动地对垃圾进行分类，无需人工干预。

高精度的分类准确率：由于使用深度学习模型进行训练，垃圾分类器可以达到较高的分类准确率。

实时性和高效性：垃圾分类器可以在实时场景中进行分类。

可扩展性：垃圾分类器可以针对不同类型的垃圾进行分类，可以扩展到更多的垃圾分类领域。

### **2.1.3 系统流程图**

本垃圾分类器的系统流程图如图2.1所示。

数据采集和预处理：收集垃圾图像数据并对其进行预处理，例如裁剪、缩放、灰度化等，以便于后续的模型训练。

模型训练：使用深度学习模型（例如卷积神经网络）对预处理后的垃圾图像数据进行训练，从而学习到垃圾分类的特征。

模型测试和验证：使用测试集来验证训练好的模型的分类准确率和泛化能力。



后端

前端

图2.1系统流程图

## 2.2 系统可行性

### **2.2.1 经济可行性**

基于深度学习的垃圾分类器可以减少人工分类的成本，并提高垃圾分类的准确率。随着技术的不断发展和成熟，该分类器的制造成本也逐渐降低。此外，随着全球环保意识的不断提高，垃圾分类器市场潜力巨大。

**（1）工作量估算**

在软件开发阶段需要使用到的人力工作量百分比如下表2-1所示。

表2-1毕业设计过程管理系统各个开发阶段的人力百分比

|  |  |
| --- | --- |
| 任务 | 人力（%） |
| 可行性研究 | 5 |
| 需求分析 | 10 |
| 概要设计和详细设计 | 25 |
| 编码和测试 | 60 |
| 总计 | 100 |

**（2）成本估算**

在软件开发阶段需要其他一次性支出如下表2-2所示。

表2-2软件开发过程中各个开发阶段的一次性支出

|  |  |
| --- | --- |
| 项目 | 费用（元） |
| 系统前期需求研究 | 500 |
| 开发计划与测试基准研究 | 500 |
| 数据库的建立与数据字典 | 1500 |
| 检查费用和管理性费用 | 1500 |
| 培训费及软件开发人员所需的一次性支出 | 1000 |
| 总计 | 5000 |

表2-3软件开发过程中成本估算

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 单价 | 数量 | 费用 |
| PC机 | 5000 | 2 | 10000 |
| 打印机 | 1000 | 1 | 1000 |
| 管理员工资 | 2000元/月 | 12 | 24000 |
| 不可预知费用 | 1000 | 10 | 10000 |
| 一次性支出 | 5000 | 1 | 5000 |
| 总计 | | 50000 | |

**（3）效益**

表2-4软件开发过程中各个开发阶段的效益

|  |  |
| --- | --- |
| 项目 | 收益（元/年） |
| 一次性收益 | 无 |
| 经常性收益（下载文献） | 80000 |
| 不可定量收益 | 无 |
| 企业定制服务 | 20000 |
| 总计 | 100000 |

**（4）收益/投资比**

一次性支出：26000元

经常性支出：24000元/年

收益：100000元/年

收益/投资比：100000\*5/(26000+24000\*5)=3.424657

**（5）货币的时间价值**

五年预计收益：100000\*5-（26000+24000\*5）=354000（元）

文献管理系统的前期软件开发阶段，需要耗费巨大的人力，投资的成本和未来获得的效益会随着软件每年的货币时间价值有所改变。文献管理系统五年收益为354000元。实际收益需要通过货币时间价值来进行核算。用利率的形式表示货币的时间价值。假设年利率为i，如果现在存入P元，则n年后可以得到的钱数为：

H:\temp\ksohtml14132\wps1.jpg (1)

这也就是P元钱在n年后的价值。反之，如果n年后能收入F元钱，那么这些钱的现在价值是：

H:\temp\ksohtml14132\wps2.jpg (2)

假定年利率为12%，利用上面计算货币现在价值的公式可以算出系统5年预计收益的现在价值，如下表2-5所示。

表2-5 将来收入折算成现在值

|  |  |  |
| --- | --- | --- |
| 将来值（元） | （1+i)^n | 现在值（元） |
| 354000 | 1.7623 | 200873 |

**（6）投资回收期估算**

文献管理系统基础建设成本约为26000元，第一年内收入为100000元，软件投资的回收期约为

 （3)

因此软件的投资回收期约为0.26年，本系统开发成本较低，维修成本适中，软件投资回报期较短可以较快获得利润，用户群体丰富且需求量大，经济利益客观值得投资。

所以，该项目在经济上是可行的。

### **2.2.2 技术可行性**

基于深度学习的垃圾分类器需要先进行图像识别、特征提取、分类等技术处理，然后再根据分类结果进行相应的处理。这需要相应的计算机技术支持，包括深度学习框架、计算机视觉、图像处理等技术。这些技术已经有了较为成熟的应用。深度学习算法在垃圾分类器中已经被广泛应用，并且不断有新的算法提出。深度学习算法需要使用高性能计算资源，如GPU集群等。这些硬件设备已经得到广泛应用，并且不断得到更新和升级。

因此，使用深度学习算法进行垃圾分类具有较高的技术可行性。

### **2.2.3 操作可行性**

该垃圾分类器可以通过手机App、小程序、智能家居等设备进行操作，使其具有较高的便携性和可操作性。此外，对于使用者而言，只需将垃圾放入相应的分类桶中即可，操作简单方便，无需太多的技术操作。

因此，该项目在操作上是可行的。

### **2.2.4 法律可行性**

垃圾分类器需要遵循相关的法律法规，包括环保法、消费者权益保护法等。垃圾分类器还需要符合电子产品相关的法规和标准，如CE认证等。同时，垃圾分类器需要保护使用者的隐私，遵循相关隐私保护法律规定。垃圾分类器的算法、模型、软件等可能涉及著作权问题，需要遵守相关的著作权法律法规。垃圾分类器需要保证数据的安全性，防止被黑客攻击和泄露，需要遵守相关的网络安全法律法规。

因此，在法律方面，该项目也是可行的。

总体来说，本项目基于深度学习的垃圾分类器设计与实现是可行的。但是，要实现这一目标，需要大量的数据集和计算资源，以及深度学习专业知识和技能。此外，垃圾的种类也可能会随时间而变化，因此需要定期更新数据集和模型以保持分类器的准确性。

## 2.3 项目进度计划

表2-5软件开发过程中各个开发阶段的一次性支出

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目阶段** | **时间** | **工作内容** | **成果** | **负责人** | **审核人** |
| 需求调研 | 2023.03.01-2023.03.10 | 对项目需求进行详细调研 | 系统需求规格说明书 | 张雪 |  |
| 系统设计 | 2023.03.11-2023.03.20 | 在需求调研的基础上对系统构架、安全体系、功能等进行系统设计 | 系统设计说明书 | 张雪 |  |
| 数据收集 | 2023.03.21-2023.04.10 | 收集不同类型的垃圾图像数据 | 数据收集 | 张雪 |  |
| 数据预处理 | 2023.04.11-2023.04.25 | 对收集到的数据进行处理，以便于模型训练和测试。 | 数据预处理 | 张雪 |  |
| 系统初验 | 2023.04.26-2023.05.10 | 选择和设计合适的深度学习模型并用训练数据对深度学习模型进行训练和调优。 | 选择最优模型 | 张雪 |  |
| 系统试运行 | 2023.05.11-2023.05.21 | 使用测试数据对训练好的深度学习模型进行验证和评估。 | 模型验证和评估 | 张雪 |  |
| 系统应用 | 2023.05.22-2023.06.15 | 将训练好的深度学习模型部署到实际场景中。 | 部署和应用 | 张雪 |  |
| 项目终验 | 2023.06.16-2013.06.30 | 根据项目结果，完善论文并项目终验 | 验收报告 | 张雪 |  |

# 3 需求分析

本论文的主要目的是基于深度学习技术，设计并实现一种垃圾分类器，旨在实现对垃圾的自动分类识别。该系统需要满足的需求如下所示。

## 3.1 系统综合需求

本系统应具有如下的系统综合需求：

1．系统具有良好的可扩展性和可维护性，能够方便地进行后续升级和维护。

2．系统应该易于使用，使普通用户能够方便地进行垃圾分类识别。

3．系统应该快速响应用户请求，对输入数据进行快速而准确的分类识别。

4．系统应该在不同的硬件平台和操作系统上具有很好的兼容性。

### **3.1.1 功能需求概述**

基于上述需求，本垃圾分类器需要具备以下功能：

1.垃圾分类：系统应能够自动将输入的图像进行分类，识别出不同种类的垃圾，并进行分类。

2.图像处理：系统应能够对输入的图像进行处理，包括调整大小、旋转、裁剪等操作，以提高分类的准确性。

3.模型训练：系统应支持深度学习模型的训练，以不断提高分类准确性和效率。

4.用户界面：系统应具有易于使用的用户界面，使用户能够上传图像、查看分类结果等。

### **3.1.2 系统性能需求概述**

为保证系统性能的稳定和优化，本论文的垃圾分类器需要满足以下性能需求：

1.准确性：系统应能够对不同种类的垃圾进行高准确率的分类，系统的分类准确率应该达到90%以上。

2.响应时间：系统应该支持多种输入格式，并在输入数据量较大时仍能保持良好的响应速度，使得用户能够在合理的时间内得到分类结果。

3.稳定性：系统应该在高并发的情况下保持稳定性和可靠性，能够同时处理多个用户的请求，能够在长时间运行过程中保持良好的性能。

4.可扩展性和可升级性：系统应具有良好的可扩展性，能够在不同规模的数据集上进行分类，以及支持不同的硬件和软件环境；系统能够根据需求进行适当的扩展和升级，满足未来的业务需求和用户需求。

5.系统需要考虑数据安全性和隐私保护，对用户上传的数据进行保护，防止数据泄露和滥用。

5.系统需要考虑资源利用效率，尽可能地减少系统的计算资源消耗和能源消耗，降低系统运行成本。

### **3.1.3 其他需求概述**

1.安全性：系统需要考虑数据安全性和隐私保护，对用户上传的数据进行保护，防止数据泄露和滥用。

2.可维护性：系统应易于维护和更新，以保持良好的性能和功能。

3.可移植性：系统应能够在不同的平台和设备上运行，以适应不同的用户需求。

## 3.2 数据流图分析

### **3.2.1 顶层数据流图**

在顶层数据流图中，用户输入原始垃圾图像；经过垃圾分类器的分类系统进行垃圾分类，最后输出分类结果给用户。

****

图3.1 顶层数据流图

### **3.2.2 功能级数据流图**

垃圾分类器的0层数据流图包括三个主要模块：数据预处理模块、模型训练模块和垃圾分类模块，如图3.2所示。

首先，数据预处理模块将原始的图像数据进行格式转换和预处理，以便于后续模型的训练和推断。然后，模型训练模块使用处理后的数据训练深度学习模型，得到可用于垃圾分类的模型。最后，垃圾分类模块使用训练好的模型对新的图像数据进行分类，并输出分类结果。



图3.2 0层数据流图

### **3.2.3 数据流图细化与分解**

数据预处理模块包括图像读取、图像格式转换、图像缩放和数据增强等子模块，如图3.3。图像读取子模块从本地文件系统或云存储中读取图像数据。图像格式转换子模块将原始图像转换为深度学习模型所需的输入格式，例如将彩色图像转换为灰度图像或将图像转换为张量格式。图像缩放子模块对图像进行缩放，以适应不同的模型输入大小。数据增强子模块通过随机旋转、裁剪和平移等方式，增强数据集的多样性和数量。



图3.3 数据预处理模块数据流图

模型训练模块包括数据集划分、模型构建、模型训练和模型评估等子模块，如图3.4。数据集划分子模块将原始数据集划分为训练集、验证集和测试集。模型构建子模块定义了深度学习模型的结构，包括卷积层、池化层、全连接层和激活函数等。模型训练子模块使用训练数据集训练模型，并使用验证数据集进行模型调优和模型选择。模型评估子模块使用测试数据集对模型进行评估，包括计算模型的准确率、召回率和F1值等指标。



图3.4 模型训练模块数据流图

垃圾分类模块包括图像读取、图像预处理和模型推断等子模块，如图3.5。图像读取子模块从本地文件系统或云存储中读取图像数据。图像预处理子模块将原始图像进行预处理，包括格式转换、缩放和数据增强等。模型推断子模块使用训练好的深度学习模型对预处理后的图像进行分类，并输出分类结果。



图3.5垃圾分类模块数据流图

## 3.3 数据字典

### **3.3.1 数据流条目定义**

根据3.2节数据流图分析情况可知，系统主要涉及的数据项包括： 据预处理, 模型训练和垃圾分类分别定义如表3-1~3-3所示。

表3-1 数据预处理信息数据字典定义

|  |  |
| --- | --- |
| 数据流 | |
| 系统名：基于深度学习的垃圾分类器 | 编号：D-001 |
| 条目名：数据预处理 | 别名： |
| 来源：用户输入的原始图片 | 去处：模型训练 |
| 数据流结构：  数据预处理：{图像读取+图像格式转换+图像缩放+数据增强}数据预处理 | |
| 简要说明：数据预处理数据流记录着预处理图片的基本信息，可用于图片的信息查看等处理 | |

表3-2 模型训练信息数据字典定义

|  |  |
| --- | --- |
| 数据流 | |
| 系统名：基于深度学习的垃圾分类器 | 编号：D-002 |
| 条目名：模型训练 | 别名： |
| 来源：数据预处理 | 去处：垃圾分类 |
| 数据流结构：  用户：{数据集划分+模型构建+模型训练+模型评估}模型训练 | |
| 简要说明：模型训练数据流记录着数据训练的基本信息，可用于查看数据训练情况和模型调优等处理 | |

表3-3 垃圾分类信息数据字典定义

|  |  |
| --- | --- |
| 数据流 | |
| 系统名：基于深度学习的垃圾分类器 | 编号：D-003 |
| 条目名：垃圾分类 | 别名： |
| 来源：模型训练 | 去处：用户 |
| 数据流结构：  用户：{图像读取+图像预处理+模型推断}垃圾分类 | |
| 简要说明：垃圾分类数据流记录着垃圾分类后的基本信息，可用于用户查看分类结果等处理 | |

### **3.3.2 数据存储条目定义**

表3-4 垃圾分类数据字典定义

|  |  |  |
| --- | --- | --- |
| 数据存储 | | |
| 系统名：基于深度学习的垃圾分类器 | 编号：**D-004** | |
| 条目名：垃圾分类 | 别名： | |
| 存储组织：每次垃圾分类一条记录 | 记录数：10^8 | 主关键字：垃圾分类 |
| 简要说明：存储每次垃圾分类的基本信息 | | |

### **3.3.3 数据处理储条目定义**

表3-6 垃圾分类数据字典定义

|  |  |
| --- | --- |
| 数据处理 | |
| 系统名：基于深度学习的垃圾分类器 | 编号：**D-005** |
| 条目名：垃圾分类 | 别名： |
| 输入数据流：用户搜索垃圾图片 | 输出数据流：垃圾分类结果 |
| 加工逻辑：   1. 输入信息合法性检查； 2. 图片匹配； 3. 根据识别图片类别禁用或开放相应功能。 | |
| 简要说明：对垃圾分类信息进行管理 | |

## 3.4 实体联系分析

### **3.4.1 实体提取及实体图**

根据需求分析的结果，在垃圾分类系统中主要有6个实体：用户，垃圾信息，垃圾类别，广告，积分记录和商品，实体提取如表3-7所示。

表3-7 实体提取结果

|  |  |  |
| --- | --- | --- |
| 实体 | 主码 | 其他属性 |
| 用户 | 用户ID | 密码，姓名，性别 |
| 垃圾信息 | 垃圾ID | 垃圾名称，垃圾分类ID |
| 垃圾类别 | 垃圾类别ID | 类别名称 |

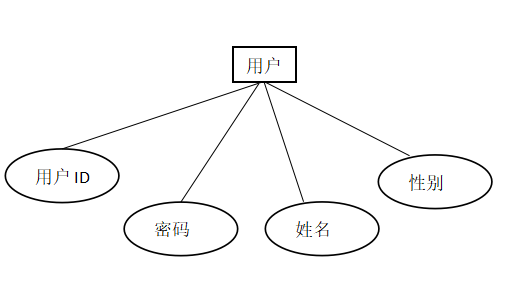


图3.6 用户属性图

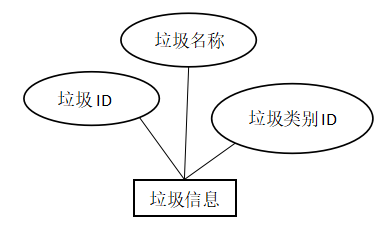


图3.7 垃圾信息属性图

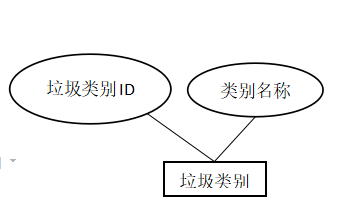


图3.8 垃圾类别属性图

### **3.4.2 实体联系图**

垃圾分类器的实体联系图如图3.9所示。

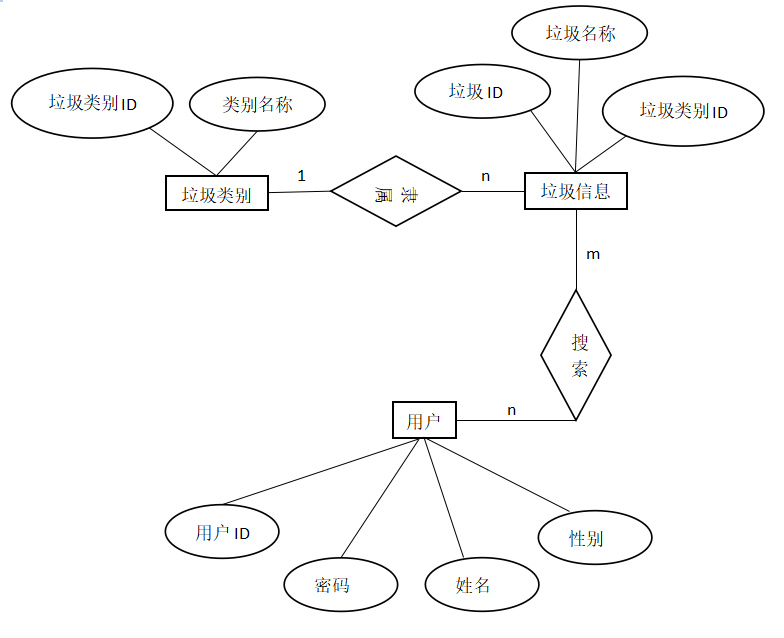


图3.9 垃圾分类器的实体联系图

# 4 系统设计

根据需求分析，垃圾分类系统层次图如图4.3所示，以此来直观分析系统的层次结构，这样既能使开发任务清晰，又便于系统的设计与实现。

## 4.1 总体设计

总体设计阶段即使用比较抽象概括的方式确定本垃圾分类器如何完成预定的任务，确定本系统的物理配置方案，进而确定组成本系统的每个程序的结构。

### **4.1.1 系统功能结构**

需求分析阶段得出的数据流图是总体设计的极好的出发点，该垃圾分类器的变换型数据流图如图4.1所示。



图4.1 变换型数据流图示例

由变换型数据流图可知，其对应的软件结构图如图4.2所示。

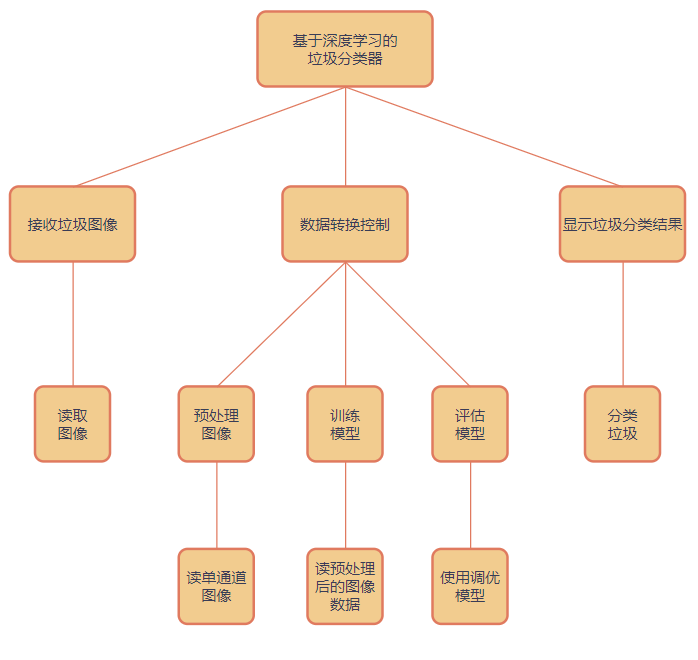


图4.2 变换型数据流图对应的软件结构图

### **4.1.2 系统层次图**

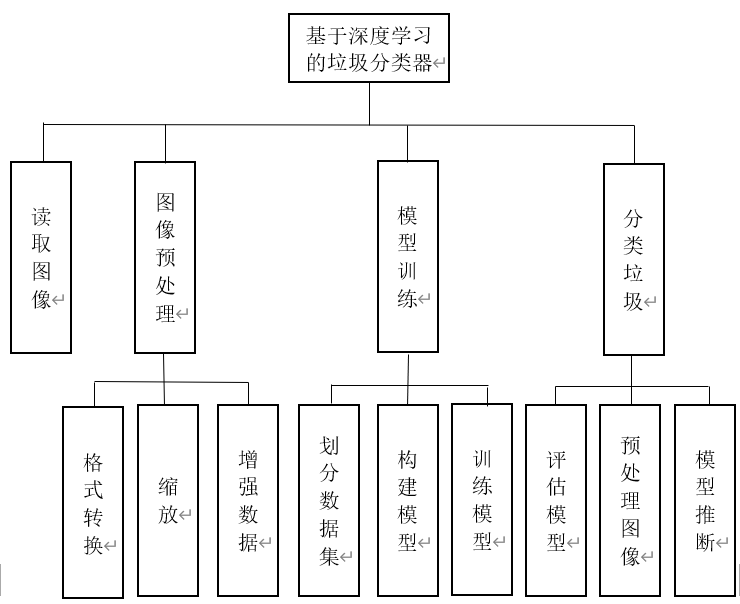


图4.3 系统层次图

## 4.2 详细设计

本章根据需求分析对垃圾分类器作出详细的设计，分别介绍了系统代表性模块设计以及系统数据库设计，为系统的具体实现打下了基础。

### **4.2.1 代表性模块设计**

#### 4.2.1.1 图像预处理模块设计简介

**（1）图像预处理模块简介**

图像预处理模块首先将读取的图像格式转换，将原始图像转换为深度学习模型所需的输入格式，例如将彩色图像转换为灰度图像或将图像转换为张量格式。然后，对图像进行缩放，以适应不同的模型输入大小。最后进行数据增强，通过随机旋转、裁剪和平移等方式，增强数据集的多样性和数量。图像预处理模块流程图如图4.4所示。

**（2）图像预处理模块流程图**

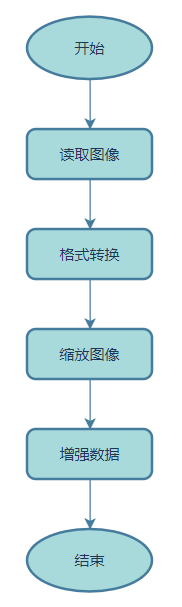


图4.4 图像预处理模块流程图

#### 4.2.1.2 模型训练模块设计简介

**（1）模型训练模块简介**

模型训练模块包括数据集划分、模型构建、模型训练和模型评估。数据集划分子模块将原始数据集划分为训练集、验证集和测试集。模型构建子模块定义了深度学习模型的结构。模型训练子模块使用训练数据集训练模型，并使用验证数据集进行模型调优和模型选择。模型评估子模块使用测试数据集对模型进行评估。

**（2）模型训练模块流程图**

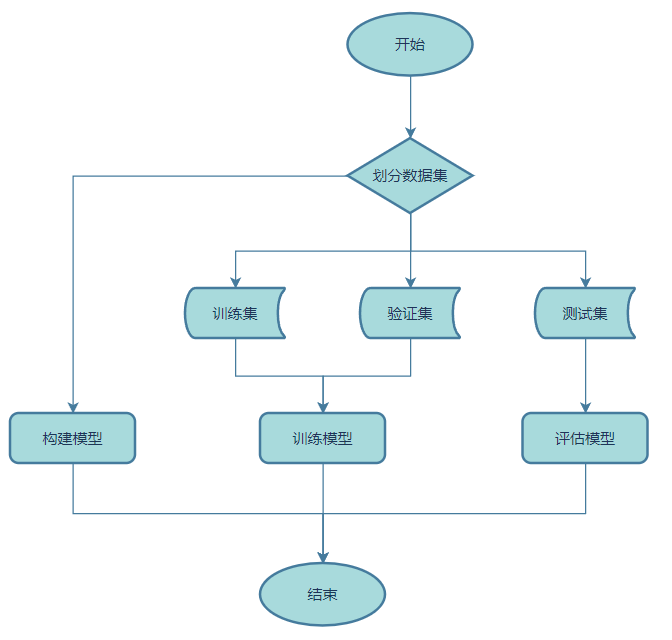


图4.5 模型训练模块流程图

**（3）针对复杂组合逻辑的判定表或判定树分析**

模型训练模块判定树分析如图4.6所示。

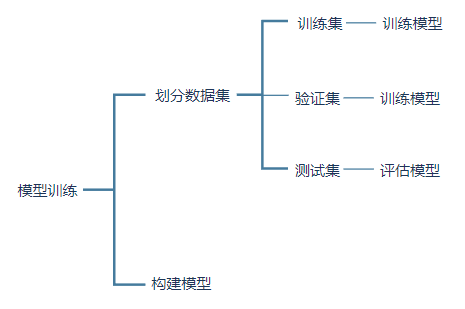


图4.6 模型训练模块判定树分析

#### 4.2.1.3 垃圾分类查询模块设计简介

**（1）垃圾分类模块简介**

用户进行垃圾分类时，上传本地图片后，图像预处理模块主要是将图像进行缩放、裁剪成固定大小的图像；完成图像的预处理后，加载模型训练出的深度学习模型，并将预处理后的图像上传到基于深度学习的垃圾分类模型中进行识别，返回查询垃圾的分类情况，以文字的形式展示，垃圾分类流程图如图4.5所示。

**（2）垃圾分类模块流程图**

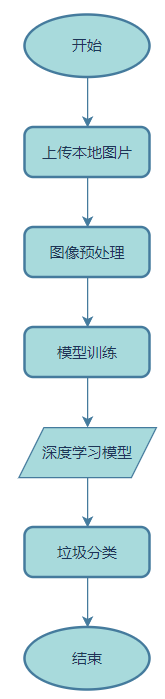


图4.7 垃圾分类流程图

**（3）针对复杂组合逻辑的判定表或判定树分析**

垃圾分类模块判定树分析如图4.8所示。

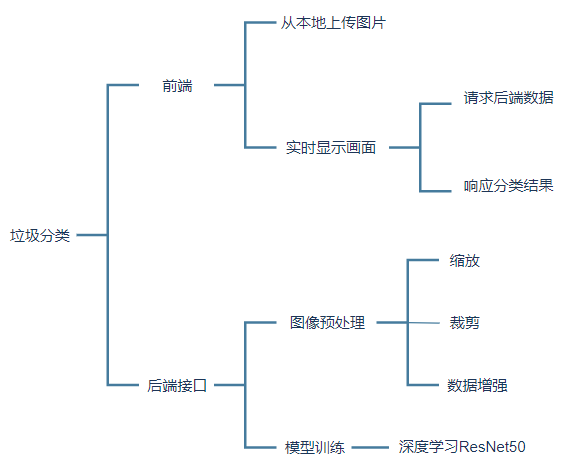


图4.8 垃圾分类模块判定树分析

### **4.2.2 系统数据库设计**

#### 4.2.2.1 数据库的逻辑设计

垃圾分类器的数据库架构包括以下三个表，关系模式如下：

用户（#用户ID，密码，姓名，性别）

垃圾信息（垃圾ID，垃圾名称，#垃圾类别ID，搜索次数，垃圾描述）

垃圾类别（垃圾类别ID，类别名称，类别描述，投放提示）

上述关系模式中，以下“\_\_\_\_\_”标注的属性名称为主键，如：用户ID、垃圾ID、垃圾类别ID等；以“#”标注的属性名称为外键，如：#用户ID、#垃圾类别ID等。

#### 4.2.2.2 数据库的表设计及表与表之间的关联

表4-1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用途 | 记录用户的基本信息 | | | |
| 字段名 | 类型 | 长度 | 键 | 备注 |
| 用户ID | Varchar | 10 | 主键 | Not Null |
| 密码 | Char | 6 |  | Not Null |
| 姓名 | Varchar | 10 |  | Not Null |
| 性别 | char | 1 |  |  |
| 相关表 | 垃圾信息表 | | | |

表4-2 垃圾信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用途 | 记录垃圾的基本信息 | | | |
| 字段名 | 类型 | 长度 | 键 | 备注 |
| 垃圾ID | Varchar | 8 | 主键 | Not Null |
| 垃圾名称 | Varchar | 10 |  | Not Null |
| 垃圾类别ID | Varchar | 10 | 外键 | Not Null |
| 相关表 | 垃圾类别表 | | | |

表4-3 垃圾类别信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用途 | 记录垃圾的基本信息 | | | |
| 字段名 | 类型 | 长度 | 键 | 备注 |
| 垃圾类别ID | Varchar | 8 | 主键 | Not Null |
| 类别名称 | Varchar | 10 |  | Not Null |
| 相关表 | 垃圾信息表 | | | |

# 5 编码与测试

本章节将从编码与测试两部分对基于深度学习的垃圾分类系统进行分析。其中，编码部分例出编码规则简介以及示例代表性模块；测试部分将从黑盒测试和白盒测试详细进行测试用例分析。

## 5.1 编码

### **5.1.1 编码规则简介**

源程序代码的逻辑简明清晰、易读易懂是好程序的一个重要标准，为了做到这一点，应该遵循下述规则：

（1）程序内部的文档

名字命名要鲜明，使它能正确地提示程序对象所代表的实体，这对于帮助阅读者理解程序是很重要的。如果使用缩写，那么缩写规则应该一致，并且应该给每个名字加注解。

注解：通常在每个模块开始处有一段序言性的注解，简要描述模块的功能、主要算法、接口特点、重要数据以及开发简史。插在程序中间与一段程序代码有关的注解，主要解释包含这段代码的必要性。对于用高级语言书写的源程序，不需要用注解的形式把每个语句翻译成自然语言，应该利用注解提供一些额外的信息。应该用空格或空行清楚地区分注解和程序。注解的内容一定要正确，错误的注解不仅对理解程序毫无帮助，反而会妨碍对程序的理解。

布局：程序清单的布局对于程序的可读性也有很大影响，应该利用适当的阶梯形式使程序的层次结构清晰明显。

（2）数据说明

数据说明的次序应该标准化，有次序就容易查阅，因此能够加速测试、调试和维护的过程。当多个变量名在一个语句中说明时，应该按字母顺序排列这些变量。如果设计时使用了一个复杂的数据结构，则应该用注解说明用程序设计语言实现这个数据结构的方法和特点。

（3）语句构造

设计期间确定了软件的逻辑结构，然而个别语句的构造却是编写程序的一个主要任务。构造语句时应该遵循的原则是，每个语句都应该简单而直接，不能为了提高效率而使程序变得过分复杂。下述规则有助于使语句简单明了。

①不要为了节省空间而把多个语句写在同一行。

②尽量避免复杂的条件测试。

③尽量减少对“非”条件的测试。

④避免大量使用循环嵌套和条件嵌套。

⑤利用括号使逻辑表达式或算术表达式的运算次序清晰直观。

（4）输入输出

在设计和编写程序时应该考虑下述有关输入输出风格的规则。

①对所有输入数据都进行检验。

②检查输入项重要组合的合法性。

③保持输入格式简单。

④使用数据结束标记，不要要求用户指定数据的数目。

⑤明确提示交互式输入的请求，详细说明可用的选择或边界数值。

⑥当程序设计语言对格式有严格要求时，应保持输入格式一致。

⑦设计良好的输出报表。

⑧给所有输出数据加标志。

（5）格式

在书写语句时，应通过采用递缩式格式使程序的层次更加清晰，同时一行上最好只书写一条语句。

### **5.1.2 代表性模块示例**

划分数据集（data-split）模块代码：

# @File : data\_split.py  
# @Software: PyCharm  
# @Brief : 将数据集划分为训练集、验证集和测试集  
import os  
import random  
import shutil  
from shutil import copy2  
import os.path as osp  
  
def data\_set\_split(src\_data\_folder, target\_data\_folder, train\_scale=0.6, val\_scale=0.2, test\_scale=0.2):  
 *'''  
 读取源数据文件夹，生成划分好的文件夹，分为train、val、test三个文件夹进行  
 :param src\_data\_folder: 源文件夹  
 :param target\_data\_folder: 目标文件夹  
 :param train\_scale: 训练集比例  
 :param val\_scale: 验证集比例  
 :param test\_scale: 测试集比例  
 :return:  
 '''* print("开始数据集划分")  
 class\_names = os.listdir(src\_data\_folder)  
 # 在目标目录下创建文件夹  
 split\_names = ['train', 'val', 'test']  
 for split\_name in split\_names:  
 split\_path = os.path.join(target\_data\_folder, split\_name)  
 if os.path.isdir(split\_path):  
 pass  
 else:  
 os.mkdir(split\_path)  
 # 然后在split\_path的目录下创建类别文件夹  
 for class\_name in class\_names:  
 class\_split\_path = os.path.join(split\_path, class\_name)  
 if os.path.isdir(class\_split\_path):  
 pass  
 else:  
 os.mkdir(class\_split\_path)  
  
 # 按照比例划分数据集，并进行数据图片的复制  
 # 首先进行分类遍历  
 for class\_name in class\_names:  
 current\_class\_data\_path = os.path.join(src\_data\_folder, class\_name)  
 current\_all\_data = os.listdir(current\_class\_data\_path)  
 current\_data\_length = len(current\_all\_data)  
 current\_data\_index\_list = list(range(current\_data\_length))  
 random.shuffle(current\_data\_index\_list)  
  
 train\_folder = os.path.join(os.path.join(target\_data\_folder, 'train'), class\_name)  
 val\_folder = os.path.join(os.path.join(target\_data\_folder, 'val'), class\_name)  
 test\_folder = os.path.join(os.path.join(target\_data\_folder, 'test'), class\_name)  
 train\_stop\_flag = current\_data\_length \* train\_scale  
 val\_stop\_flag = current\_data\_length \* (train\_scale + val\_scale)  
 current\_idx = 0  
 train\_num = 0  
 val\_num = 0  
 test\_num = 0  
 for i in current\_data\_index\_list:  
 src\_img\_path = os.path.join(current\_class\_data\_path, current\_all\_data[i])  
 if current\_idx <= train\_stop\_flag:  
 copy2(src\_img\_path, train\_folder)  
 # print("{}复制到了{}".format(src\_img\_path, train\_folder))  
 train\_num = train\_num + 1  
 elif (current\_idx > train\_stop\_flag) and (current\_idx <= val\_stop\_flag):  
 copy2(src\_img\_path, val\_folder)  
 # print("{}复制到了{}".format(src\_img\_path, val\_folder))  
 val\_num = val\_num + 1  
 else:  
 copy2(src\_img\_path, test\_folder)  
 # print("{}复制到了{}".format(src\_img\_path, test\_folder))  
 test\_num = test\_num + 1  
  
 current\_idx = current\_idx + 1  
  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*{}\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*".format(class\_name))  
 print(  
 "{}类按照{}：{}：{}的比例划分完成，一共{}张图片".format(class\_name, train\_scale, val\_scale, test\_scale,  
 current\_data\_length))  
 print("训练集{}：{}张".format(train\_folder, train\_num))  
 print("验证集{}：{}张".format(val\_folder, val\_num))  
 print("测试集{}：{}张".format(test\_folder, test\_num))  
  
  
# 数据集划分  
if \_\_name\_\_ == '\_\_main\_\_':  
 src\_data\_folder = "/home/l/data\_2/lzy/Workspace/rub\_classification-master/dataset" # *todo 修改你的原始数据集路径* target\_data\_folder = src\_data\_folder + "\_" + "split"  
 if osp.isdir(target\_data\_folder):  
 print("target folder 已存在， 正在删除...")  
 shutil.rmtree(target\_data\_folder)  
 os.mkdir(target\_data\_folder)  
 print("Target folder 创建成功")  
  
 data\_set\_split(src\_data\_folder, target\_data\_folder)  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("数据集划分完成，请在{}目录下查看".format(target\_data\_folder))

## 5.2 测试

在软件初步成型时应该进行软件测试。软件测试是为了发现错误而执行程序的过程。或者说，软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现程序错误的过程。

总的来说以最少的时间和人力，系统地找出软件中潜在的各种错误和缺陷，以考虑用户是否能接受该产品。

（1）测试规范

1）所有测试都应该能追溯到用户需求。

2）应该远在测试开始之前就制定出测试计划。实际上，一旦完成了需求模型就可以着手制定测试计划，在建立了设计模型之后就可以立即开始设计详细的测试方案。因此，在编码之前就可以对所有测试工作进行计划和设计。

3）把Pareto原理应用到软件测试中。Pareto原理说明，测试发现的错误中的80%很可能是由程序中20%的模块造成的。

4）应该从“小规模”测试开始，并逐步进行“大规模”测试。通常，首先重点测试单个程序模块，然后把测试重点转向在集成的模块簇中寻找错误，最后在整个系统中寻找错误。

5）穷举测试是不可能的。所谓穷举测试就是把程序所有可能的执行路径都检查一遍的测试。即使是一个中等规模的程序，其执行路径的排列数也十分庞大，由于受时间、人力口发其他资源的限制，在测试过程中不可能执行每个可能的路径，因此，测试只能证明程序中有错误，不能证明程中没有错误。但是，精心地设计测试方案，有可能充分覆盖程序逻辑并使程序达到所要求的可靠性。

6）为了达到最佳的测试效果，应该由独立的第三方从事测试工作。所谓“最佳效是指有最大可能性发现错误的测试。

（2）测试方法

对于本系统的测试采用两种方法：一种方法称为黑盒测试，已知产品应该具有的功能，通过测试来检验是否每个功能都能正常使用；另一种方法称为白盒测试，已知产品的内部工作过程，通过测试来检验产品内部动作是否按照规格说明书的规定正常进行。

对于软件测试而言，黑盒测试法把程序看作一个黑盒子，完全不考虑程序的内部结构和处理过程。也就是说，黑盒测试是在程序接口进行的测试，它只检查程序功能是否能按照规格说明书的规定正常使用，程序是否能适当地接收输入数据并产生正确的输出信息，程序运行过程中能否保持外部信息(例如数据库或文件)的完整性。黑盒测试又称为功能测试。白盒测试法与黑盒测试法相反，它的前提是可以把程序看成装在一个透明的白盒子里，测试者完全知道程序的结构和处理算法。这种方法按照程序内部的逻辑测试程序，检测程序中的主要执行通路是否都能按预定要求正确工作。白盒测试又称为结构测试。

### **5.2.1 白盒测试**

设计测试方案是测试阶段的关键技术问题。白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试，发现内部代码在算法，溢出，路径，条件等等中的缺点或者错误，进而加以修正。下面主要介绍学生住宿信息查询模块的白盒测试流程，其他模块功能均以此功能测试过程为模板进行测试。

1.测试目标：从软件设计者角度检测模型训练模块功能是否存在错误;

2.测试数据：数据预处理后的图片

3.测试对象：被测试模块程序流程图如图5.1所示：

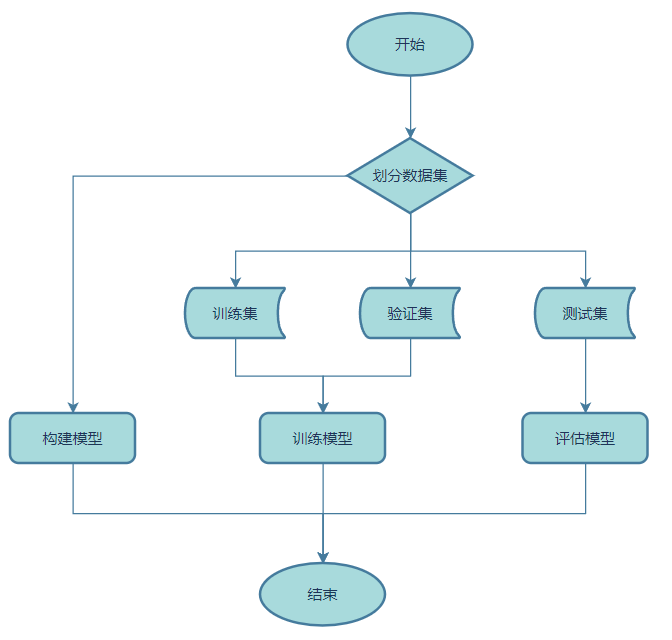


图5.1 程序流程图

#### 5.2.1.1 模型训练模块独立路径分析

基本路径测试是Tom McCabe提出的一种白盒测试技术。使用这种技术设计测试用例，首先计算程序的环形复杂度，并用该复杂度为指南定义执行路径的基本集合，从该基本集合到处的测试用例可以保证程序中的每条语句至少执行一次，而且每个条件在执行时都将分别取真假两种值。

（1）根据过程设计结果画出相应的流图：

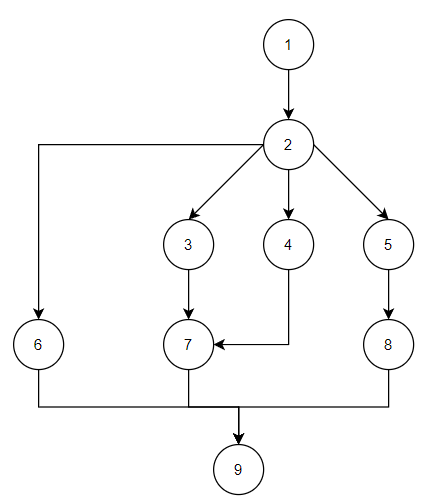


图5.2 流图

（2）计算流图的环形复杂度：

根据：V（G）= E-N+2 （E:流图的边数11 N：节点数9）

可得该流图的环形复杂度为4，即有4条独立路径条数。

（3）确定线性独立路径的基本集合：

由于环形复杂度为4，因此共有4条独立路径，如下列出具体路径：

1. 1-2-6-9
2. 1-2-3-7-9
3. 1-2-4-7-9
4. 1-2-5-8-9

#### 5.2.1.2 模型训练模块测试用例设计

模型训练路径 模型 数据集 评估 独立路径

模型构建 Resnet50 ①

模型训练 Resnet50 训练集 ②

模型训练 Resnet50 验证集 ③

模型评估 Resnet50 测试集 0.99994 ④

#### 5.2.1.3 模型训练模块测试情况分析

测试步骤依次为语句覆盖测试、判定覆盖测试、条件覆盖测试、判定/条件覆盖测试、条件组合覆盖测试、路径覆盖。测试用例以路径覆盖为原则。在每个步骤中需要设计相应的测试用例，使其达到不同的覆盖标准。

1. 语句覆盖测试

模型训练路径 模型 数据集 评估 覆盖路径 预期输出

模型构建 Resnet50 ① 模型Resnet50

模型训练 Resnet50 训练集 ② 训练结果

模型训练 Resnet50 验证集 ③ 验证结果

模型评估 Resnet50 测试集 0.99994 ④ 测试结果

1. 判定覆盖测试

判定覆盖需要将程序中每个判断的取真分支和取假分支至少经历一次，应执行路径：

1. 1-2-6-9
2. 1-2-3-7-9
3. 1-2-4-7-9

④ 1-2-5-8-9

模型训练路径 模型 数据集 评估 覆盖路径 预期输出

模型构建 Resnet50 ① 模型Resnet50

模型训练 Resnet50 训练集 ② 训练结果

模型训练 Resnet50 验证集 ③ 验证结果

模型评估 Resnet50 测试集 0.99994 ④ 测试结果

1. 路径覆盖测试

路径覆盖需要使得每一条可能的路径至少执行一次，所有可能路径：

1. 1-2-6-9
2. 1-2-3-7-9
3. 1-2-4-7-9
4. 1-2-5-8-9

模型训练路径 模型 数据集 评估 覆盖路径 预期输出

模型构建 Resnet50 ① 模型Resnet50

模型训练 Resnet50 训练集 ② 训练结果

模型训练 Resnet50 验证集 ③ 验证结果

模型评估 Resnet50 测试集 0.99994 ④ 测试结果

### **5.2.2 黑盒测试**

黑盒测试又名为功能测试，主要目的是发现软件设计的需求或者是软件设计规格说明书中的错误缺陷。软件的开发具有特定性，一般都是为了某种具体特定功能开发的，软件功能的描述依赖于在软件需求阶段的需求规格说明书的分析，软件在设计过程中被分成了一个或者是多个功能，保证这些功能能够正常运行的就是软件测试，从而进一步满足用户的需求与需要。黑盒测试与白盒测试相反，软件设计程序被看做一个打不开的盒子，盒子里的程序代码测试人员不能看到，只能看到软件或者是某些模块的简单功能描述，这种测试方法主要是验证软件或者是功能的实现度。

#### 5.2.2.1 垃圾分类模块的等价类分析

等价类划分法是一种典型的、重要的黑盒测试方法，它将程序所有可能的输入数据划分为若干个等价类。然后从每个部分中选取具有代表性的数据当做测试用例。测试用例由有效等价类和[无效等价类](https://baike.baidu.com/item/%E6%97%A0%E6%95%88%E7%AD%89%E4%BB%B7%E7%B1%BB/7847821?fromModule=lemma_inlink)的代表数据组成，从而保证测试用例具有完整性和代表性。

首先要先划分出有效等价类和无效等价类：

一、图片格式

1. 有效等价类：

① 图片格式存在

② 图片格式为jpg,ipeg,png

1. 无效等价类：

① 图片格式不存在

② 除以上三种的其他图片格式

③ 其他非图片格式

1. 图片内容
2. 有效等价类：

① 图片内容存在

② 图片内容为有害垃圾，厨余垃圾，可回收垃圾和其他垃圾中的一种

（2）无效等价类：

① 图片内容不存在

② 非垃圾图片

三、机器学习模型

（1）有效等价类：

① 模型存在

② 模型为ResNet50d

（2）无效等价类：

① 模型不存在

② 除此模型外其他模型

表5.1 垃圾分类模块黑盒测试-等价类的划分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入条件** | **有效等价类** | **编号** | **无效等价类** | **编号** |
| 图片格式 | 格式为jpg  格式为jpeg  格式为png | （1）  （2）  （3） | 格式为bmp  非图片格式 | （5）  （6） |
| 存在 | （4） | 不存在 | （7） |
| 图片内容 | 有害垃圾图片  厨余垃圾图片  可回收垃圾图片其余垃圾图片 | （8）  （9）  （10）  （11） | 非垃圾图片 | （13） |
| 存在 | （12） | 不存在 | （14） |
| 模型 | ResNet50d | （15） | 其他模型 | （16） |

#### 5.2.2.2 垃圾分类模块测试用例设计

表6.2 学生信息查询模块黑盒测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **输入数据** | **覆盖等价类** | **预计输出** |
| 1 | 格式为jpeg的有害垃圾图片 | （2）（8） | 有害垃圾 |
| 2 | 格式为jpeg的厨余垃圾图片 | （2）（9） | 厨余垃圾 |
| 3 | 格式为jpeg的可回收垃圾图片 | （2）（10） | 可回收垃圾 |
| 4 | 格式为jpeg的其他垃圾图片 | （2）（11） | 其他垃圾 |
| 5 | 格式为bmp的厨余垃圾图片 | （5）（9） | 无法打开 |
| 6 | 格式为jpg的有害垃圾图片 | （1）（8） | 有害垃圾 |
| 7 | 格式为jpg的厨余垃圾图片 | （1）（9） | 厨余垃圾 |
| 8 | 格式为jpg的可回收垃圾图片 | （1）（10） | 可回收垃圾 |
| 9 | 格式为jpg的其他垃圾图片 | （1）（11） | 其他垃圾 |
| 10 | 视频类型文件 | （6） | 无法打开 |
| 11 | 格式为png的有害垃圾图片 | （3）（8） | 有害垃圾 |
| 12 | 格式为png的厨余垃圾图片 | （3）（9） | 厨余垃圾 |
| 13 | 格式为png的可回收垃圾图片 | （3）（10） | 可回收垃圾 |
| 14 | 格式为png的其他垃圾图片 | （3）（11） | 其他垃圾 |
| 15 | 使用ResNet50d模型进行训练 | （15） | 训练结果 |
| 16 | 使用其他模型进行训练 | （16） | 虽有训练结果，但不符合 |

#### 5.2.2.3 垃圾分类模块测试情况分析

[边界值分析法](https://baike.baidu.com/item/%E8%BE%B9%E7%95%8C%E5%80%BC%E5%88%86%E6%9E%90%E6%B3%95/4137943?fromModule=lemma_inlink)是对程序输入或输出的边界值进行测试的一种黑盒测试方法。实际的测试工作证明，考虑了边界条件的测试用例比那些没有考虑边界条件的测试用例具有更高的测试回报率。这里所说的边界条件，是指输入和输入等价类中那些恰好处于边界、或超过边界、或在边界以下的状态。本节使用边界值分析法对垃圾分类模块测试情况进行分析。

（1）输入的图片的格式为bmp，且刚好为垃圾图片

输入：.bmp格式的垃圾图片

预期输出：“无法打

输入非图片格式的文件，且刚好与垃圾有关

输入:.mov形式的垃圾视频

预期输出：“无法打开”

（2）输入非垃圾内容的图片，且图片格式刚好为jpg

输入：jpg格式的良好水果图片

预期输出：“厨余垃圾”

（3）模型训练时使用的模型非ResNet50d

输入：使用其他模型训练

预期输出：不符合预期训练目的

# 6 系统使用说明

## 6.1 系统运行环境和配置

Windows 11 家庭中文版

PyCharm Community Edition 2023.1.1

## 6.2 系统操作说明

### **6.2.1 读取图像模块说明**

从网络上爬取图片作为数据集。

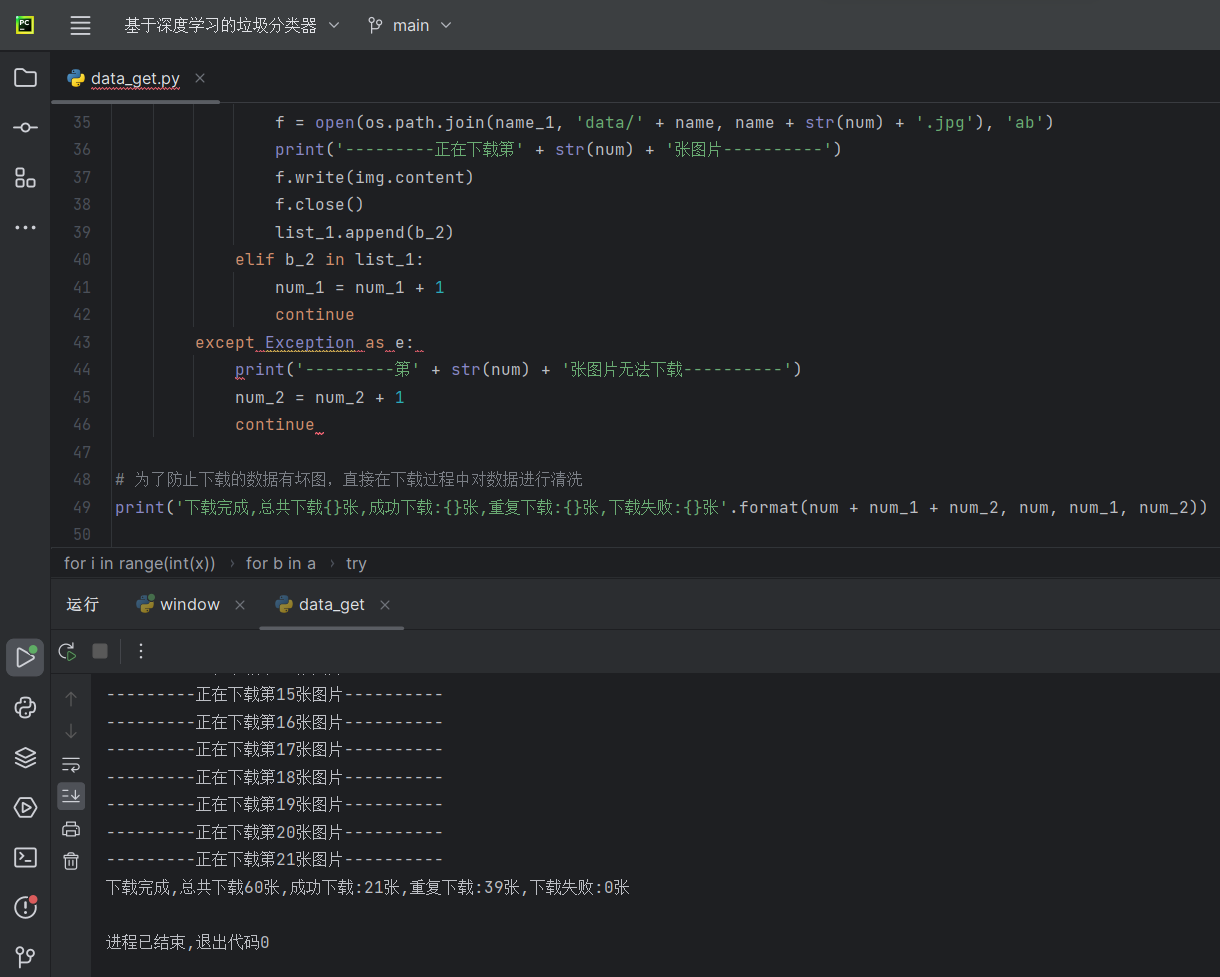


图6.1 图片获取代码界面

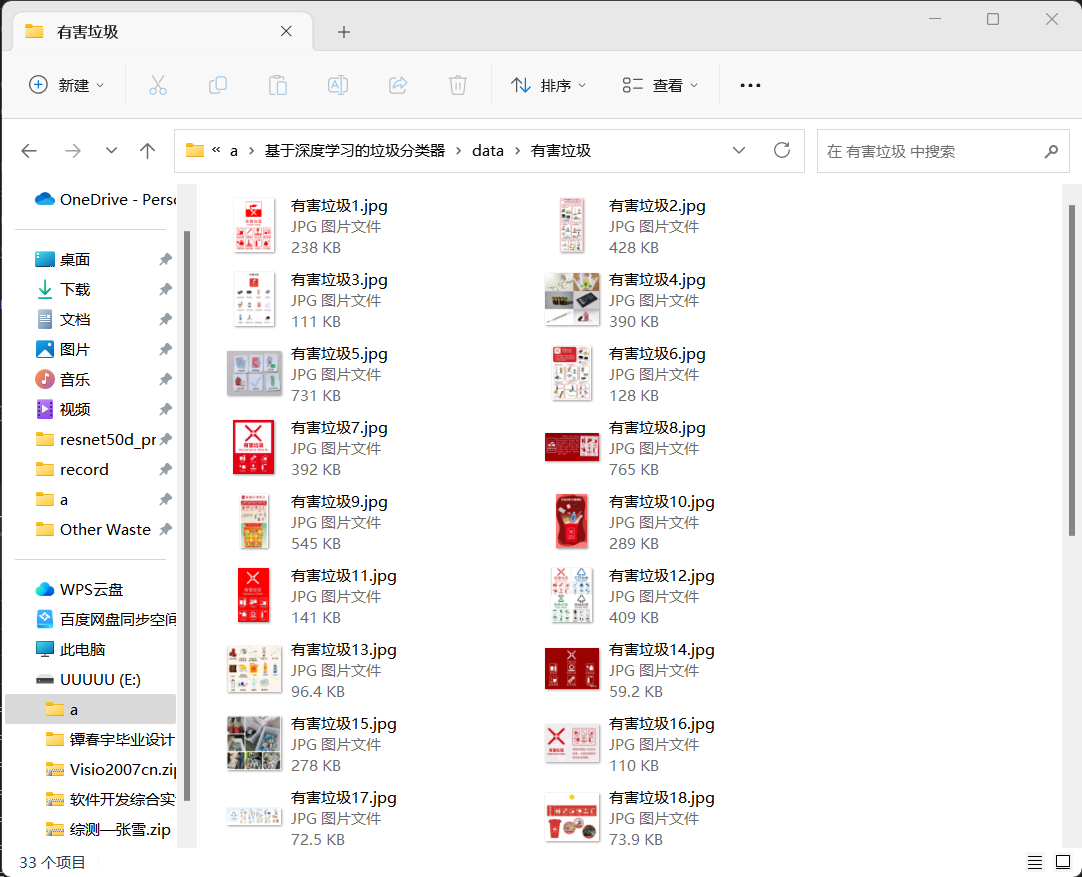


图6.2 图片获取成功

### **6.2.2 图像预处理模块说明**

图像预处理模块首先将读取的图像格式转换，将原始图像转换为深度学习模型所需的输入格式，例如将彩色图像转换为灰度图像。然后，对图像进行缩放或清晰，以适应不同的模型输入大小。最后进行数据增强。

1. **将彩色图像转换为灰度图像**

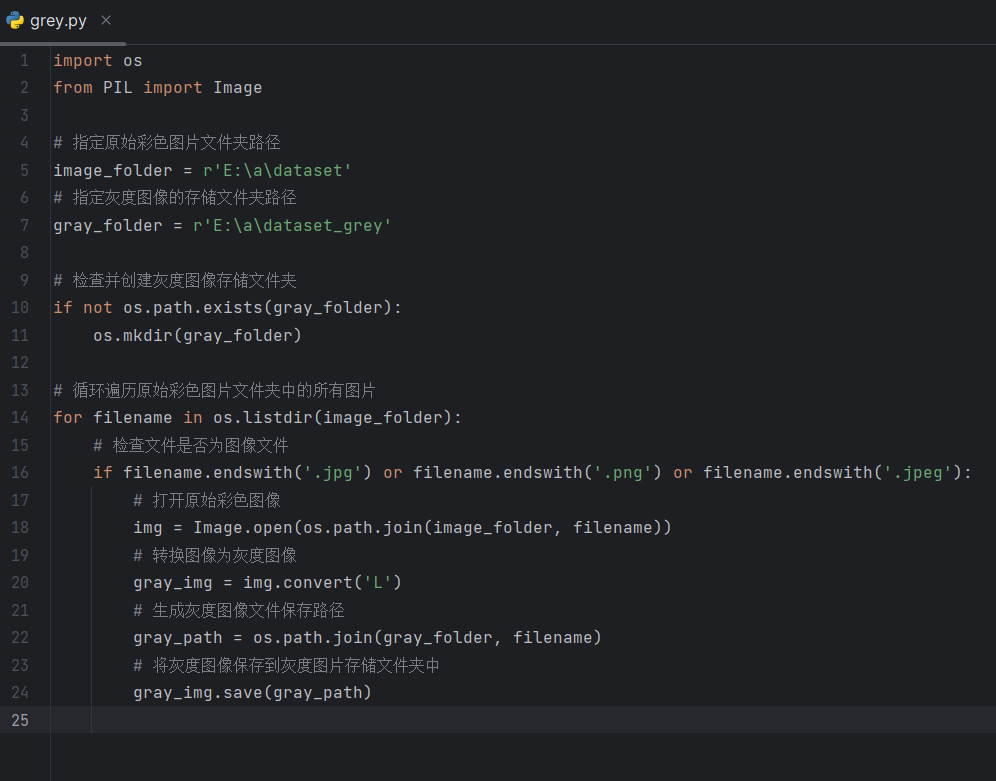


图6.3 图像转换代码界面

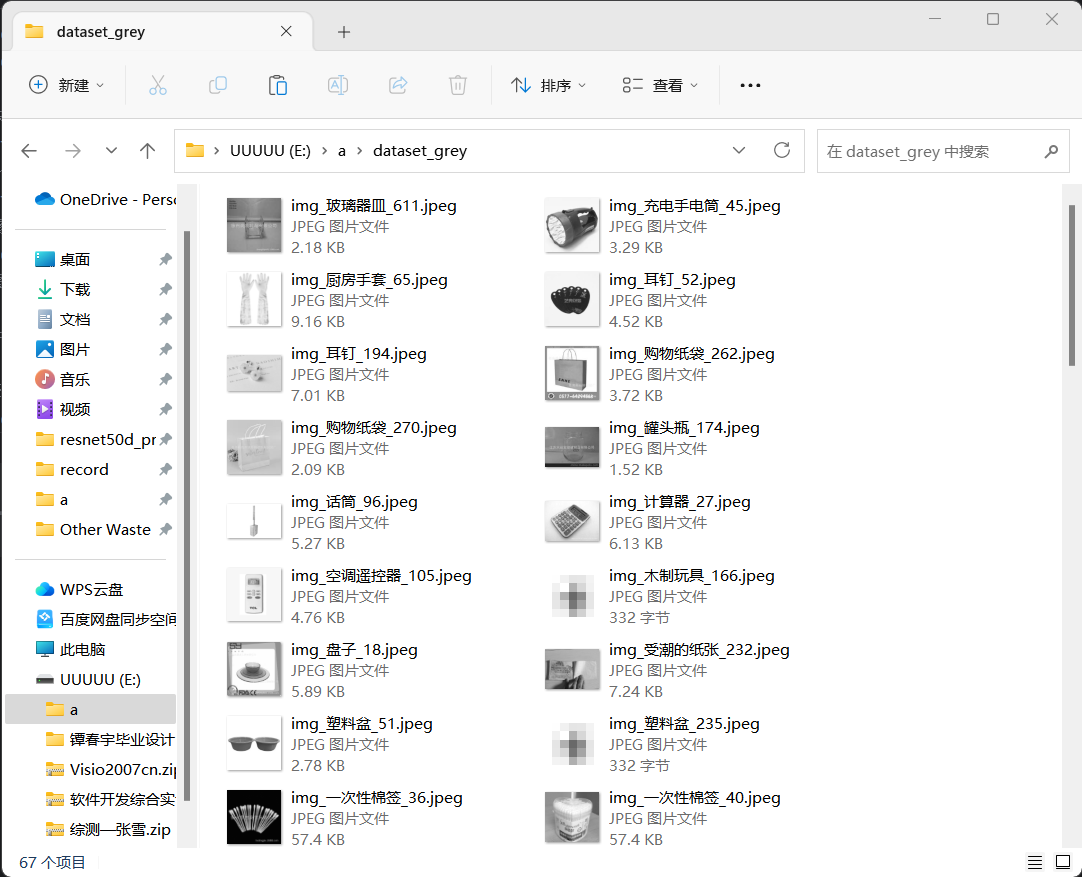


图6.4 转换为灰度图像

1. **图像清晰**

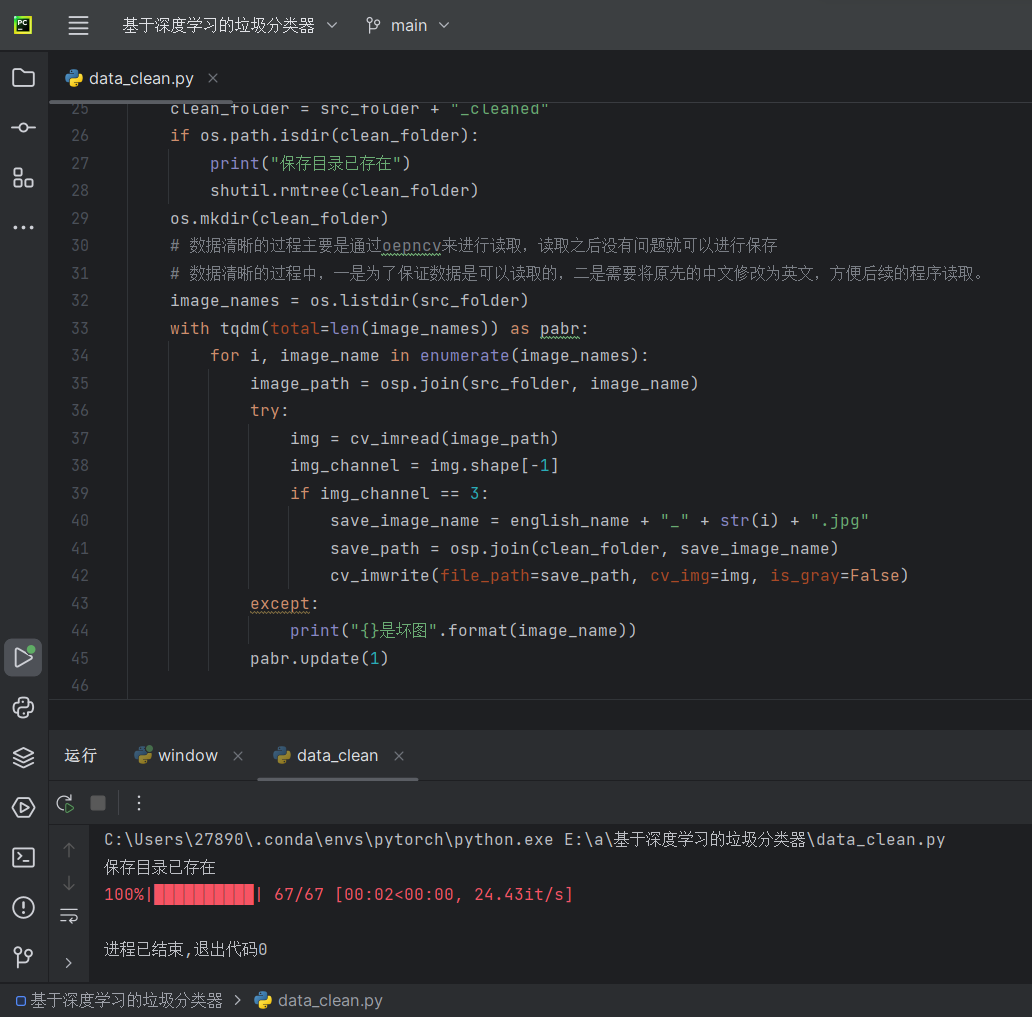


图6.5 图像清晰代码界面

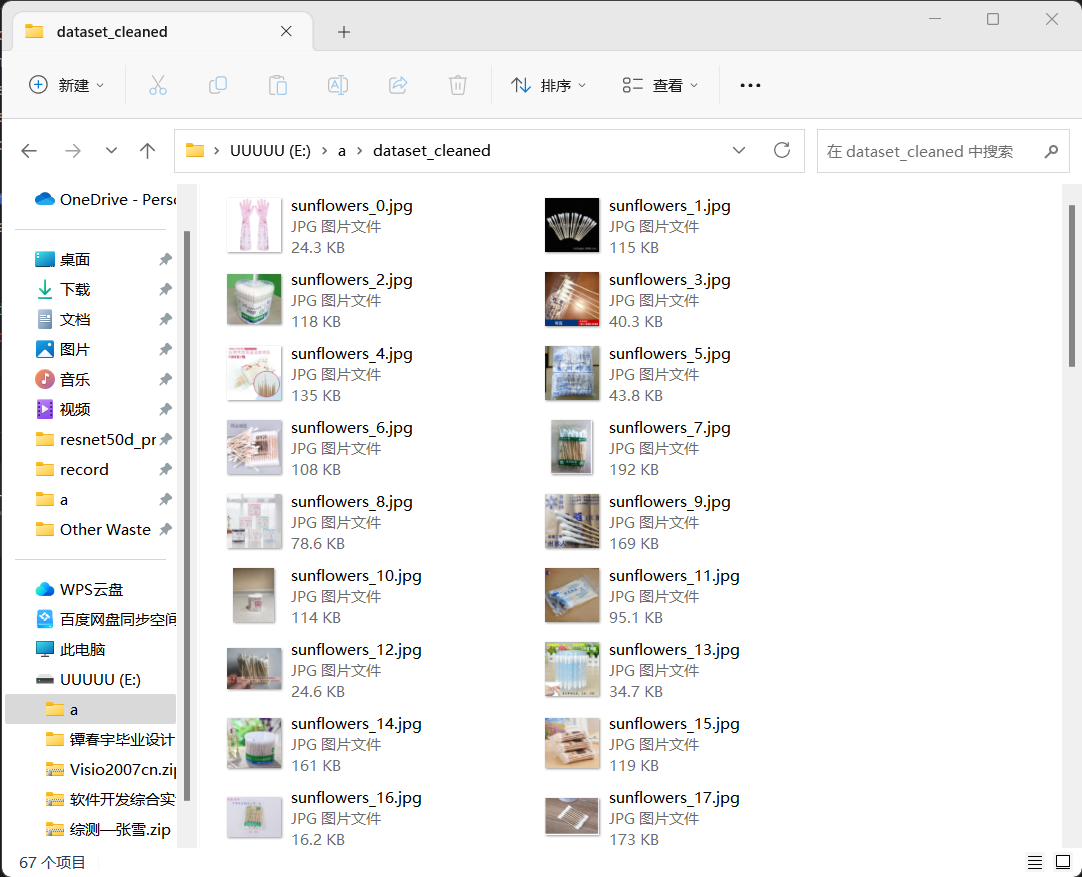


图6.6 清晰后的图像

### **6.2.3 模型训练模块说明**

模型训练模块包括数据集划分、模型构建、模型训练和模型评估等子模块。数据集划分子模块将原始数据集划分为训练集、验证集和测试集。模型构建子模块定义了深度学习模型的结构——ResNet50d。模型训练子模块使用训练数据集训练模型，并使用验证数据集进行模型调优和模型选择。模型评估子模块使用测试数据集对模型进行评估。

1. **数据集划分**

数据集划分将原始数据集划分为训练集、验证集和测试集。

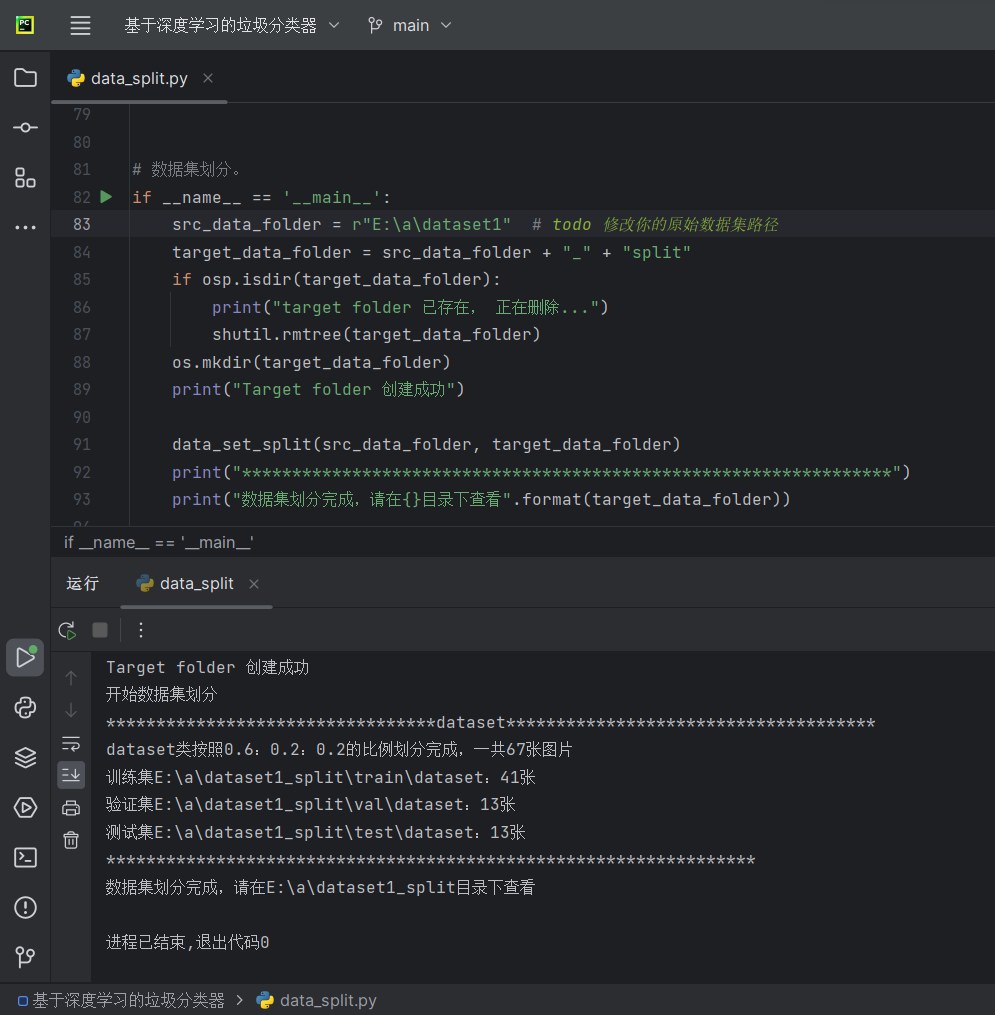


图6.7 数据集划分代码界面

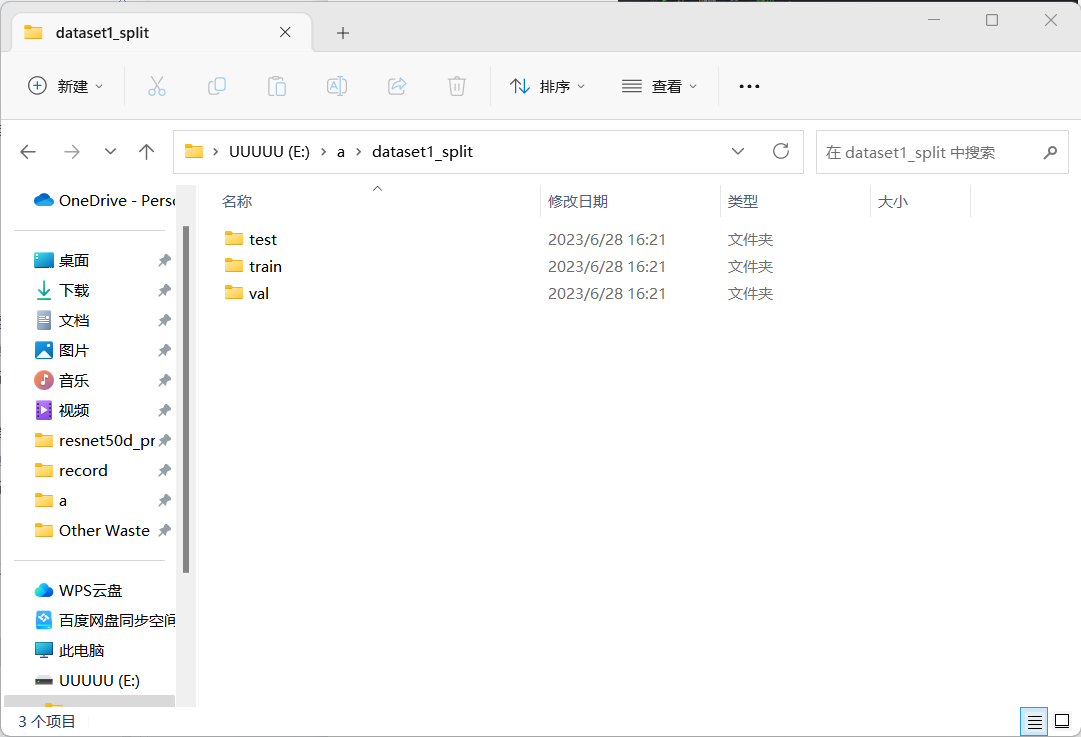


图6.8 数据集划分成功

1. **模型训练**

模型训练子模块使用训练数据集训练模型，并使用验证数据集进行模型调优和模型选择。

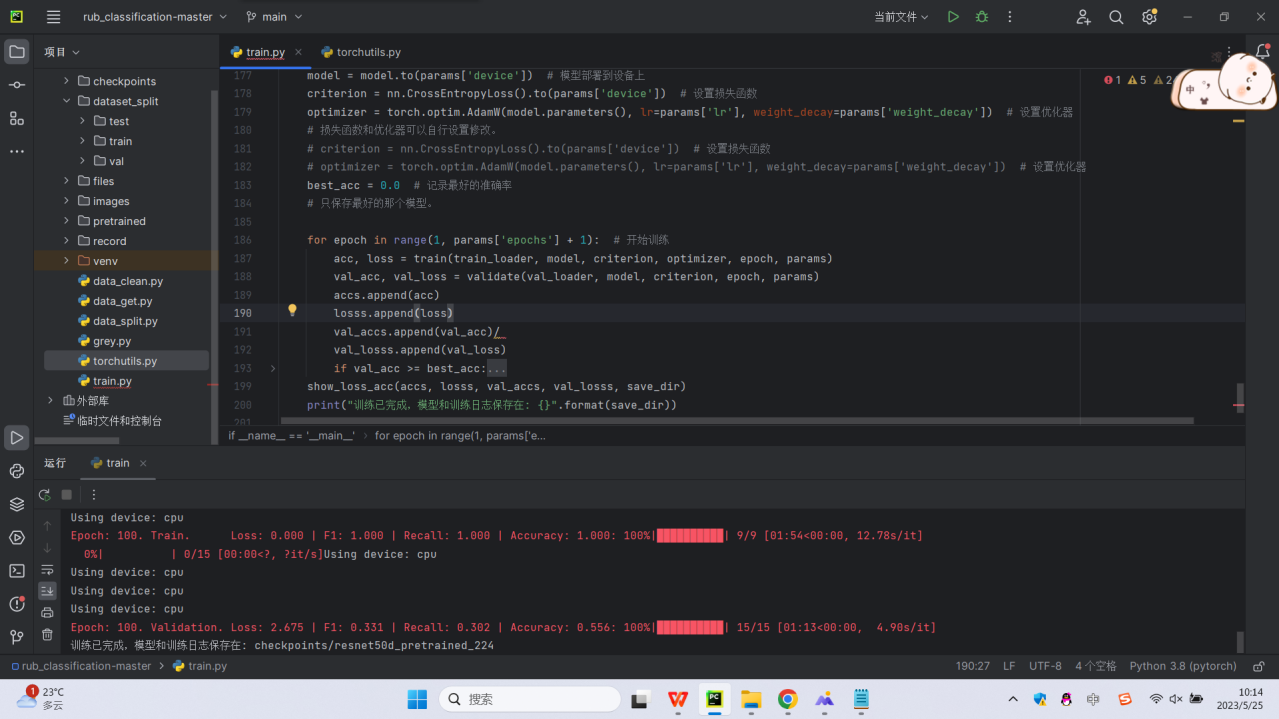


图6.9 模型训练代码界面

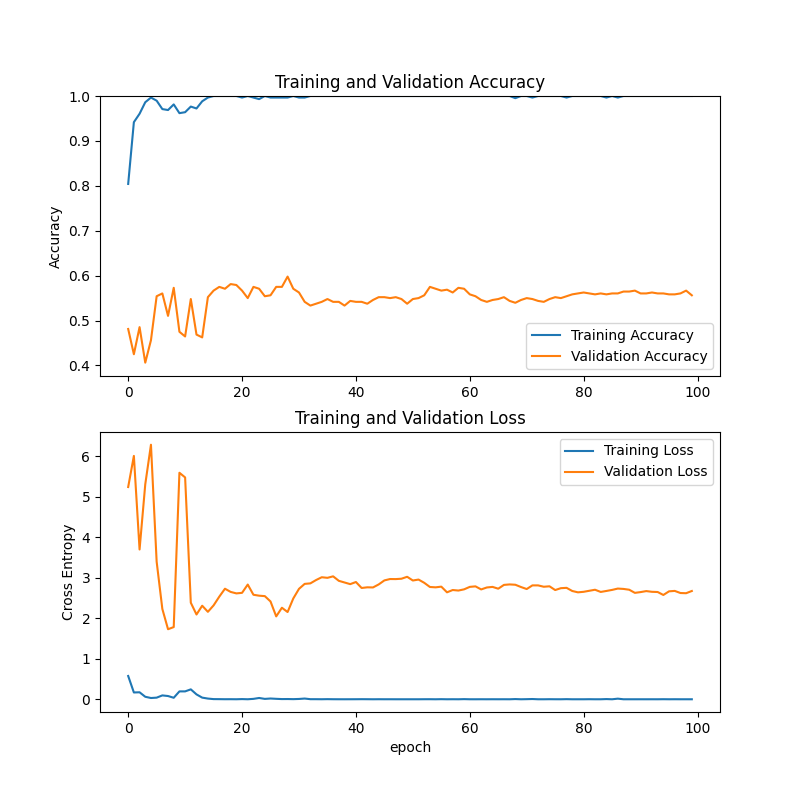


图6.10 模型训练结果

1. **模型评估**

模型评估使用测试数据集对模型进行评估。

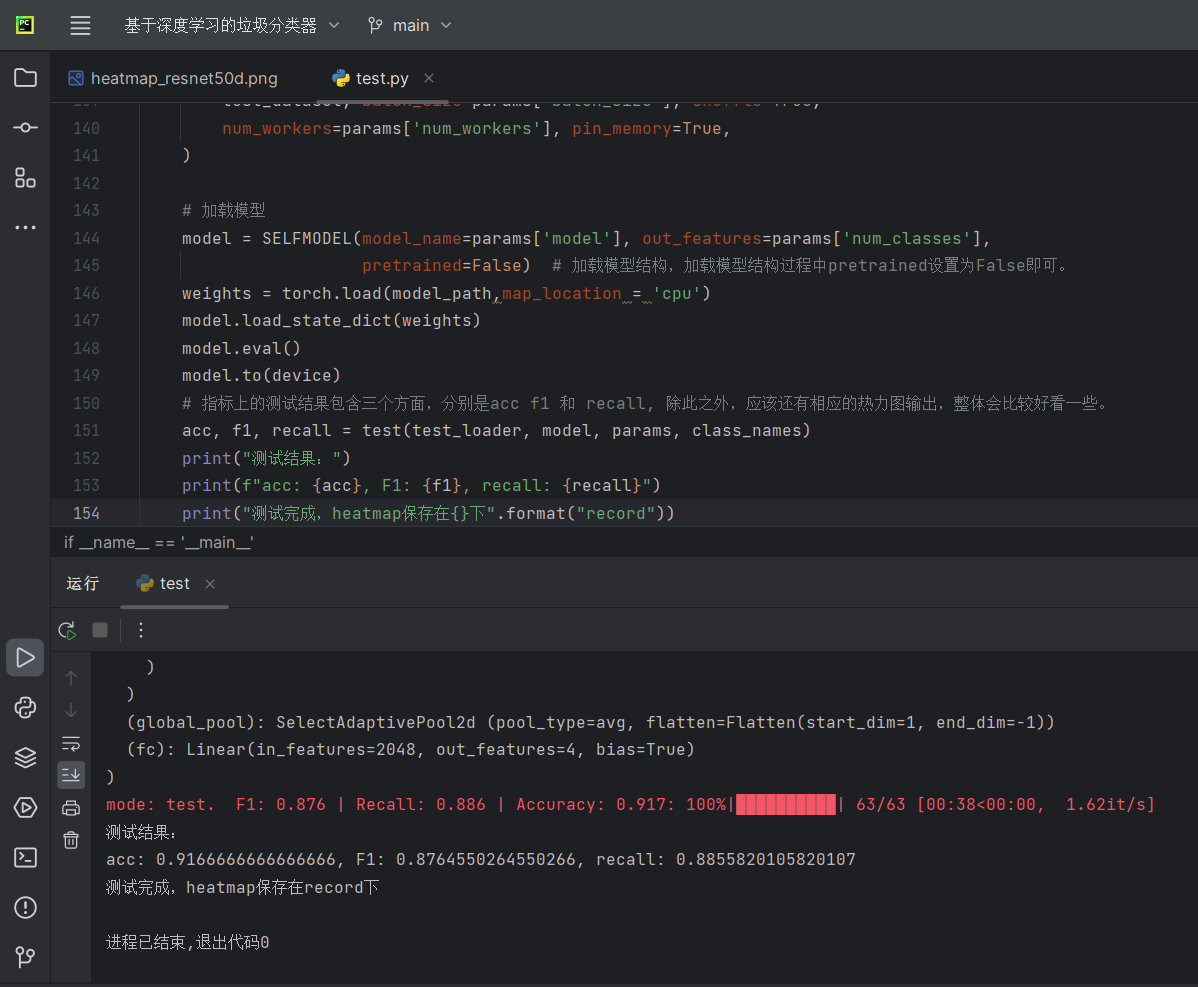


图6.11 模型评估代码界面

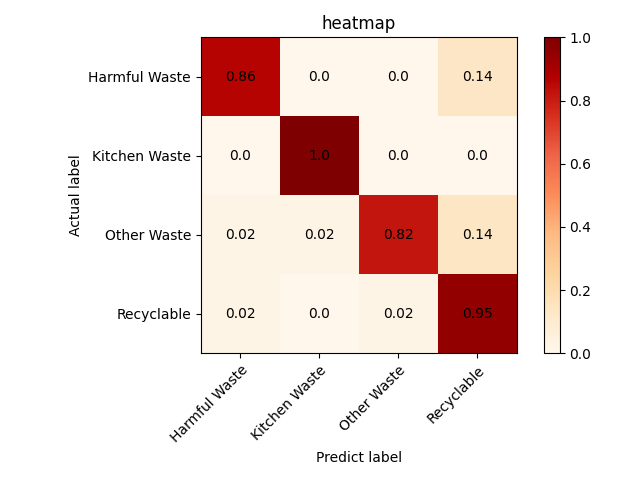


图6.12 模型评估结果

### **6.2.4 分类垃圾模块说明**

用户进行垃圾分类时，上传本地图片后，图像预处理模块主要是将图像进行缩放、裁剪成固定大小的图像；完成图像的预处理后，加载模型训练出的深度学习模型，并将预处理后 的图像上传到基于深度学习的垃圾分类模型中进行识别，返回查询垃圾的分类情况，以文字的形式展示。

**（1）分类界面**



图6.13 分类界面

**（2）等待识别界面**

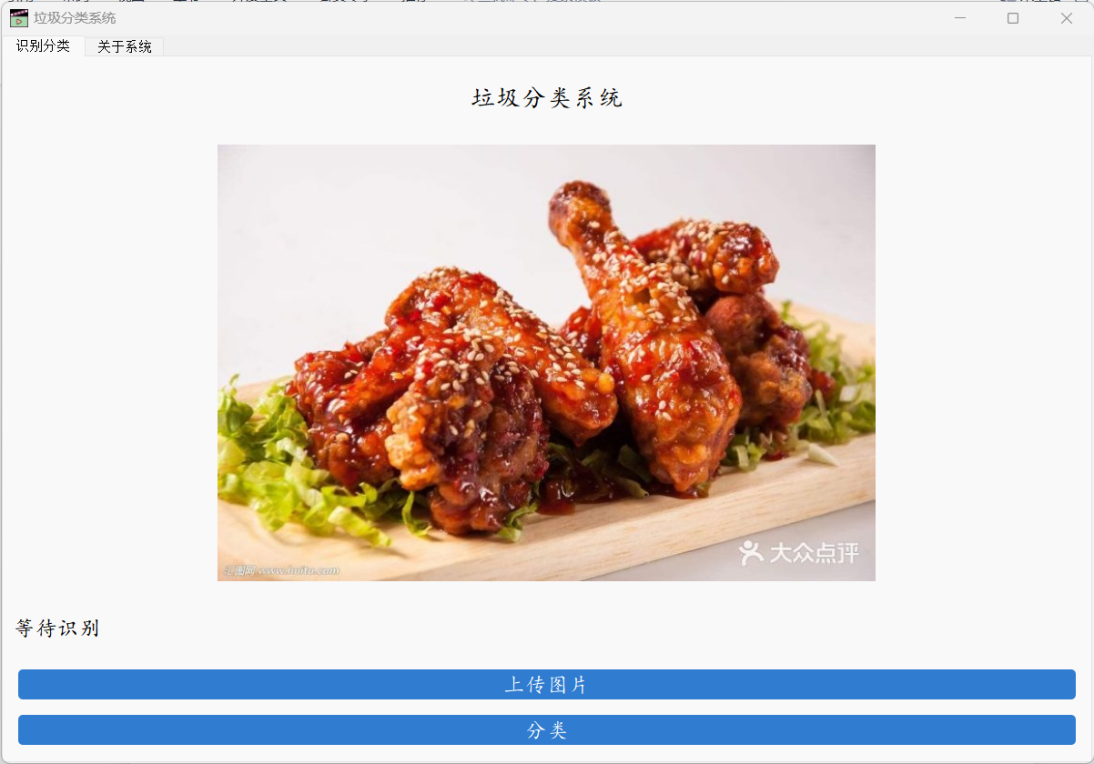


图6.14 等待识别界面

**（3）显示分类结果界面**

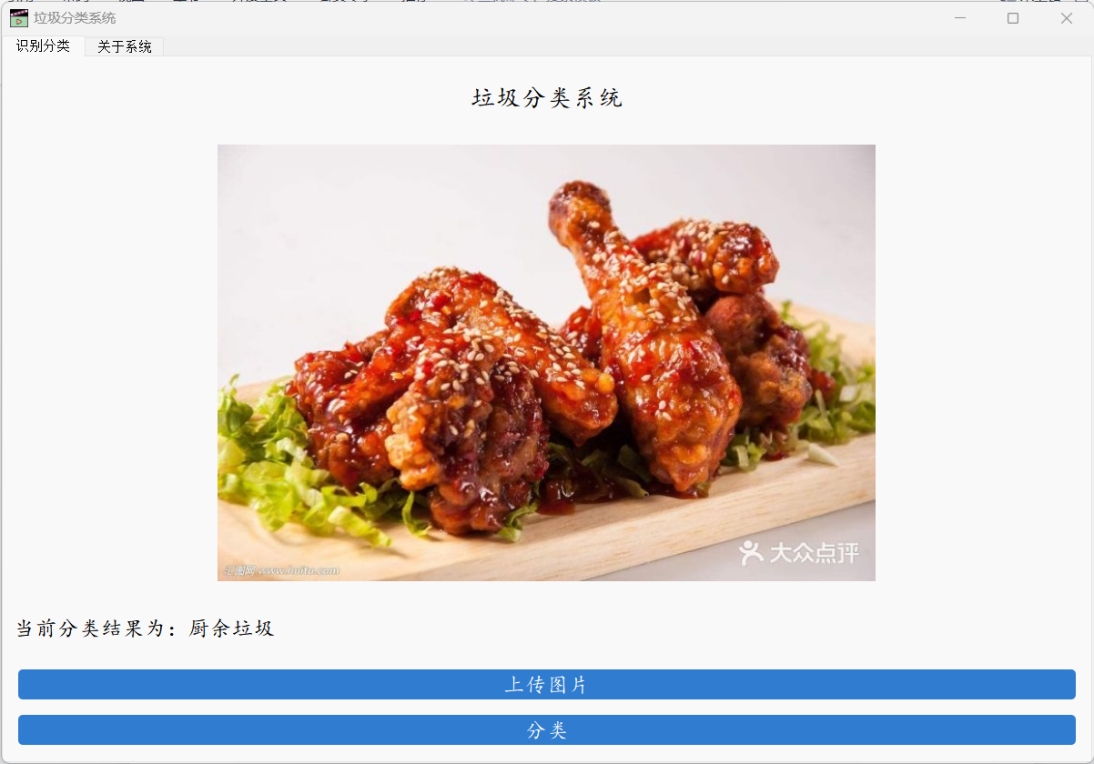


图6.15 显示分类结果

# 7 总结

垃圾分类是一项重要的环保任务，可以有效减少对环境的污染并促进可持续发展。基于深度学习的垃圾分类器是一种自动化解决方案，利用计算机视觉和深度学习技术，能够识别和分类不同类型的垃圾。

首先，设计垃圾分类器需要准备一个标注好的垃圾图像数据集。这个数据集应包含各种垃圾类别的图像样本，并确保每个样本都正确标注了其所属的类别。数据集的规模越大越好，以提高分类器的准确性和泛化能力。

其次，选择适当的深度学习模型架构。本项目选择基于ResNet50d深度学习的模型，可以根据数据集的规模和复杂性进行评估，确保模型能够有效地提取图像特征并进行准确的分类。

在模型选择之后，需要对模型进行训练。训练过程包括将数据集划分为训练集和验证集，并利用训练集对模型进行参数优化，以使模型能够更好地拟合数据集。

训练完成后，需要对模型进行评估。可以使用验证集来评估模型的性能，计算分类的准确率、精确率、召回率和F1得分等指标。通过评估指标可以了解模型的分类能力和泛化能力，进而进行必要的调整和改进。

最后，对于一个成功的垃圾分类器，还可以进行部署和应用。可以将训练好的模型集成到实际系统中，用户可以通过上传本地垃圾图像，然后通过模型进行分类，实现自动化的垃圾分类过程。

设计基于深度学习的垃圾分类器需要综合考虑数据集的准备、模型选择、训练过程和评估指标。通过合理的设计和优化，可以实现准确、高效的垃圾分类系统，提高垃圾分类的效率和可行性，为环保事业作出贡献。未来，可以进一步研究和改进深度学习模型，以提高垃圾分类器的性能和适应性，促进垃圾分类技术的发展和应用。

# 参考文献

1. 蔡兆彬. 基于深度学习的生活垃圾分类识别系统的设计与实现[D].东北林业大学,2022.DOI:10.27009/d.cnki.gdblu.2022.001684.
2. 陈伟. 基于深度学习的垃圾分类算法研究[D].天津职业技术师范大学,2021.DOI:10.27711/d.cnki.gtjgc.2021.000065.
3. 董子源. 基于深度学习的垃圾分类系统设计与实现[D].中国科学院大学(中国科学院沈阳计算技术研究所),2020.DOI:10.27587/d.cnki.gksjs.2020.000005.
4. 周鸿利. 基于深度学习的垃圾分类识别算法研究[D].华中科技大学,2020.DOI:10.27157/d.cnki.ghzku.2020.006408.
5. 潘唯一. 基于深度学习的垃圾分类识别方法研究与实现[D].成都理工大学,2020.DOI:10.26986/d.cnki.gcdlc.2020.000336.
6. 战秋成,季龙华,赵际云,修艳琪,戴婷婷.基于深度学习的智能垃圾分类系统研究[J].机械工程师,2022(08):100-103.
7. 陈牧图,谭睿,石垒垒,冯月芹.基于深度学习的智能垃圾分类系统设计[J].电子测试,2022,36(17):12-14+18.DOI:10.16520/j.cnki.1000-8519.2022.17.003.
8. 陶航,江学焕,张金亮,陈波.基于深度学习的垃圾分类系统[J].湖北汽车工业学院学报,2022,36(02):36-39+44.
9. 李丕兵. 基于深度学习的垃圾分类系统研究与应用[D].青岛大学,2021.DOI:10.27262/d.cnki.gqdau.2021.002259.

[10]徐丽. 基于深度学习的垃圾分类系统的设计与开发[D].浙江大学,2021.DOI:10.27461/d.cnki.gzjdx.2021.002525.

[11]田震. 基于深度学习的垃圾分类算法研究[D].哈尔滨工业大学,2021.DOI:10.27061/d.cnki.ghgdu.2021.004427.

[12]梁旭东. 基于深度学习的智能垃圾分类系统研究[D].西安建筑科技大学,2021.DOI:10.27393/d.cnki.gxazu.2021.001038.

[13]莫卓亚,彭创权.基于深度学习的垃圾分类识别技术[J].现代工业经济和信息化,2020,10(10):60-61.DOI:10.16525/j.cnki.14-1362/n.2020.10.26.

[14]武嘉年. 基于深度学习的生活垃圾分类检测及定位的研究[D].合肥学院,2022.DOI:10.27876/d.cnki.ghfxy.2022.000018.

[15]王小燕,谢文昊,杨艺芳,胡瑞.基于深度学习的垃圾分类检测方法[J].现代电子技术,2021,44(21):110-113.DOI:10.16652/j.issn.1004-373x.2021.21.023.

# 致谢

首先，我要对指导老师张晓春老师表示衷心的感谢，感谢她一直以来对我进行精心的指导与帮助。由于张晓春老师的专业知识，严谨的工作态度，以及对我的耐心、细心的辅导，我才能圆满地完成了我的设计。在这里，我要向张晓春老师致以最真诚的谢意。

同时，我要感谢安财的同学们，感谢你们在我学术和个人发展中的陪伴和帮助。学校为我提供的良好学习和生活环境。在这个美丽的校园里，我能够专注于学业，拓宽自己的知识领域，并参与各种丰富多彩的活动。这段宝贵的经历将成为我人生中的宝贵回忆。

我也要感谢其他所有的教师和导师，他们在我大学期间给予了我专业知识和人生指导。他们的教诲和帮助使我能够不断成长和进步。