# 垃圾分类系统项目结题报告

## 研究背景和目标

随着人们的环境保护意识的提高以及政策的推广，垃圾分类已然成为了一种潮流。但能够正确地对垃圾进行分类确不是一件容易的事，许多人都不能准确地将常见的垃圾进行正确分类，因为这需要人们有足够丰富的相关知识。垃圾分类难，易出错等问题已经成为了亟待解决的重大课题，此时若是有一种通过图像识别来帮助人们进行垃圾分类的应用程序，想必能够给人们的生活带来很大的便利，也能对国家推行垃圾分类政策起到促进作用，有利于共同创建绿色健康的美好生活环境。

### 1.1国内外发展状况

1. 国内发展情况

当前市面上已经有这样功能的应用，如支付宝APP有一个垃圾分类查询的小程序，可以拍照识别垃圾种类并对其做出垃圾分类经过简单测试，该小程序可以识别常见的物品，并且一定程度上正确分类，如塑料瓶、纸巾、搪瓷杯和椅子等，但是也有一些常见的物品不能被准确识别，如眼镜盒、哑铃、计算器等，可见，该小程序虽然拥有一定的垃圾分辨能力，但是仍有许多不足，特别是不能在离线情况下工作，这无疑给该小程序的使用带来了许多不便。微信同样有一个垃圾分类小程序，但是查询结果同样不稳定，同样不能在离线情况下实现识别功能。而且无论是支付宝APP还是微信APP的垃圾分类程序，都需要用户登录后才可以使用，在这个互联网高度发达的年代，这还可能存在着信息泄露的安全隐患。

1. 国外发展状况

利用图像识别进行垃圾分类在国外已有先例，美国佛罗里达州萨拉索塔的Single Stream Recyclers LLC公司研发并采用机器人可通过视觉传感器来对不同类型的垃圾进行分类，机器人内置的算法通过海量的图像进行训练，包括水瓶、啤酒罐、牛奶罐、食品包装盒等各种各样的可回收废弃物。图像数据库包含了这些回收物完好无损、有凹痕以及被压碎等各种状态。视觉传感器利用计算机视觉扫描快速移动的物体，通过颜色、质地、形状、大小和材料来区分物体。这是更进一步的应用，在混合的垃圾内区分不同的垃圾，并通过机器人来进行不同垃圾的分类，但这是科研端的前沿研究成果，还未能给普通民众带来便利。

### 1.2本课程的设计目标

随着深度学习技术在视觉领域的应用和发展，我们看到了利用AI来自动进行垃圾分类的可能：通过摄像头拍摄垃圾图片，检测图片中垃圾的类别。

本项目的设计目标为：通过使用大量垃圾外观特征数据来训练卷积神经网络，实现在用户端上传垃圾图片后，识别上传的图片中含有的垃圾外观数据与特征，从而识别并反馈垃圾种类，给出该种垃圾的所在分类。

在家庭中，垃圾分类存在着分不清，理不明，分不净的缺点，为垃圾处理厂工作增加了负担。本系统可以照拍照快速识别垃圾种类，使用户可以快速且准确地得出分类结果，完成垃圾分类在家庭中的落实，免去了用户在传统应用中需要检索大量数据的冗余操作。

目前国内的垃圾处理厂基本都是采用人工流水线分拣的方式进行垃圾分拣，存在工作环境恶劣、劳动强度大、分拣效率低等缺点。在海量垃圾面前，人工分拣只能分拣出极有限的一部分可回收垃圾和有害垃圾，绝大多数垃圾只能进行填埋，带来了极大的资源浪费和环境污染危险。本项目同样可以协助人工进行高准确率垃圾分类，并且也可以与垃圾分拣机器人等相结合，减少垃圾处理的工作强度和成本，大大提升垃圾处理效率。

## 2. 团队任务完成情况

### 2.1团队人员组成

组长：莫开寅

组员：杨秀林

万小榕

韩凌辉

杨佳霖

### 2.2任务分工

莫开寅：项目架构设计与具体任务分发，以及开源代码学习，结题报告文档撰写

杨秀林：UI设计与实现，并负责UI界面与功能函数串联，结题报告文档撰写

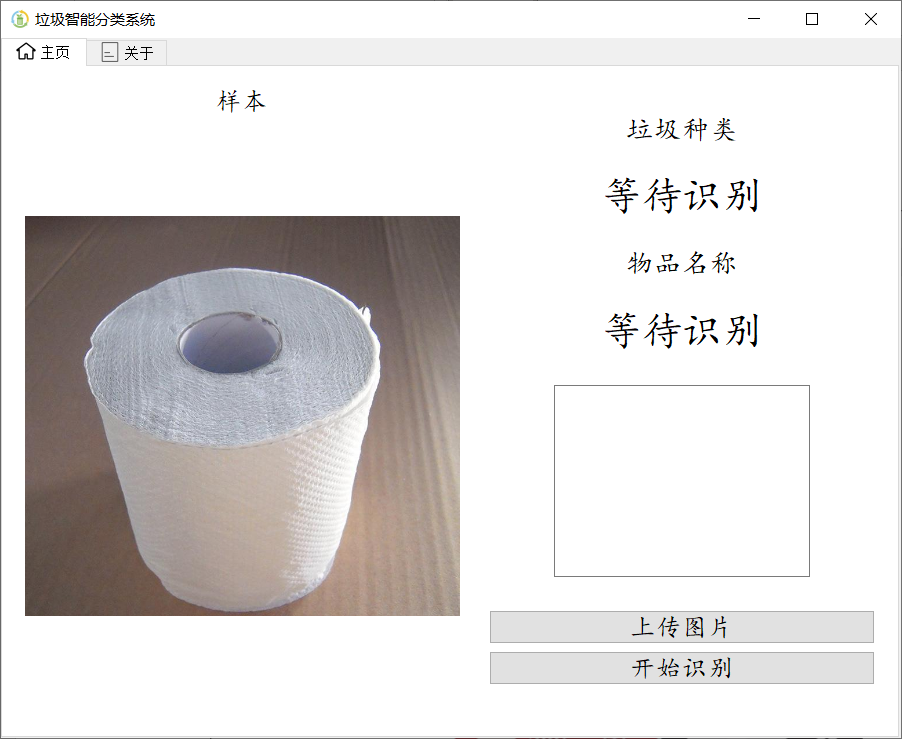
万小榕：开源代码学习，包括训练集与验证集的构建、训练模型的保存与导出，结题报告文档撰写

韩凌辉：开源代码学习，包括训练集与验证集的构建、训练模型的保存与导出，结题报告文档撰写

杨佳霖：数据库寻找与导入，包括图片与对应标签的提供，以及开源代码学习，结题PPT制作

### 2.3项目进度里程碑

* Ananaconda下载与安装
* 相关conda虚拟环境配置
* 训练集函数、验证集函数的书写完成
* 数据集导入与训练验证
* UI界面搭建成功

* 网上图片导入进行系统测试



## 3. 总体设计

### 3.1多因素影响分析

* 训练模型用时，gpu较cpu 训练用时可以达到1:10的效果，本项目限于技术与硬件，采用cpu 训练模型，对于GB级别的数据集用时较长（几乎是一个晚上）
* 数据集与系统应用时传进的图片可能差距过大，以至于不能很好地识别。比如说沙滩上的已经残破的塑料瓶与餐桌上相对完整的塑料瓶，因为数据集采集时没有采集足够大的数据，对于这两种塑料瓶，想同时辨别为塑料瓶是存在难度的。

### 3.2多技术比较

* Pytorch 与 Tensorflow
  + 相对来说tensorflow出现时间更早，网上的教程更多
  + 同时tensorflow的数据可视化处理起来更方便
* 激活函数relu 与 sigmoid的选择
  + 采用Sigmoid等函数，算激活函数时（指数运算），计算量⼤，反向传播求误差梯度时，求导涉及除法和指数运算，计算量相对大，而采用ReLU激活函数，整个过程的计算量节省很多。
  + 第⼆，对于深层网络，Sigmoid函数反向传播时，很容易就会出现梯度消失的情况（在Sigmoid接近饱和区时，变换太缓慢，导数趋于0，这种情况会造成信息丢失），这种现象称为饱和，从而无法完成深层网络的训练。⽽ReLU就不会有饱和倾向，不会有特别⼩的梯度出现。
  + 第三，ReLU会使⼀部分神经元的输出为0，这样就造成了网络的稀疏性，并且减少了参数的相互依存关系，缓解了过拟合问题的发生。当然现在也有⼀些对ReLU的改进，⽐如PReLU，random ReLU等，在不同的数据集上会有⼀些训练速度上或者准确率上的改进。
* Sequential序贯模型 与 mobilenet模型
  + Sequential序贯模型训练用时更长，效果更差
  + mobilenet模型属于小巧类型，分辨效果更好，训练用时更少

### 3.3总体设计方案

#### 3.3.1 UI界面设计方案

将页面分为主页和关于两个窗口

**主页：**主页窗口由左边子窗口和右边子窗口组成，左边子窗口用于显示需要识别的图片（即图片样本），右边子窗口用于操作系统和显示该垃圾的分类信息，设有“上传图片”和“开始识别”两个按键，以及“垃圾种类”和“物品名称”两个显示子窗口，左右子窗口采用水平布局，各类标题均对自己的子窗口居中对齐；

**关于：**关于窗口有“欢迎使用垃圾分类系统”标题，及四类垃圾类型图标，对“关于”窗口居中对齐。

#### 3.3.2 训练模块设计方案

采用训练集：验证集=8:2的比例，预计通过归一化图片数据集之后，通过卷积层（采用relu激活函数）、池化层将特征图大小减半，导出128个神经元的全连接层，最后用softmax导出输出层，对应数据集具体的类别数目

#### 3.3.3 测试模块设计方案

采用训练集：验证集=8:2的比例，调用keras库的函数

loss, accuracy = model.evaluate(val\_ds)显示其损失率、准确率。

## 4.功能与算法设计

### 4.1功能设计

#### 4.1.1 UI功能设计

**主页面标题：**“垃圾智能分类系统”。将主页面分为主页和关于两个窗口，在主页窗口操作系统，在关于窗口放置其他相关信息

**主页：**将主页划分为左右两个部分，在左子窗口显示需进行识别的图片，右子窗口显示垃圾分类辨识结果及按键按钮，需要两个功能按钮：“上传图片”和“开始查询”；按下“上传图片”按钮可以跳转到本地，在本地文件夹中选择需要识别的图片，并左子窗口显示该图片；

上传好图片后，可按下“开始查询”按键，将识别及分类结果显示出来，显示内容包括：垃圾种类、物品名称以及该类垃圾处理方法规范，显示为文本。

**关于：**关于窗口为欢迎语句“欢迎使用垃圾分类系统”及四种垃圾类型图片：厨余垃圾、可回收物、有害垃圾和其他垃圾。

#### 4.1.2 模型训练功能设计

在对应文件夹存放图片数据集之后，在代码中导入数据集文件地址与模型保存地址，即可开始训练模型，并预计模型可以用于预测垃圾图片的具体分类。

#### 4.1.3 图片相似度匹配功能设计

寻址到UI界面导入的待查询图片之后，导出训练好的模型，通过特征比对得到预测的标签——即输出的垃圾种类，导出结果到UI界面。

### 4.2核心算法

#### 4.2.1 模型训练

# 数据加载，将会按照8比2的比例分割数据集，其中8份作为训练集，2份作为测试集‘

def data\_load(data\_dir, img\_height, img\_width, batch\_size):

train\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

data\_dir,

label\_mode='categorical',

validation\_split=0.2, # 划分比例

subset="training", # 训练集

seed=123,

image\_size=(img\_height, img\_width),

batch\_size=batch\_size)

val\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

data\_dir,

label\_mode='categorical',

validation\_split=0.2, # 划分比例

subset="validation", # 验证集

seed=123,

image\_size=(img\_height, img\_width),

batch\_size=batch\_size)

class\_names = train\_ds.class\_names # 获取数据集的类名

return train\_ds, val\_ds, class\_names # 返回训练集、验证集和类名

# 模型加载

def model\_load(IMG\_SHAPE=(224, 224, 3), class\_num=245):

# 通过keras构建模型

model = tf.keras.models.Sequential([

tf.keras.layers.experimental.preprocessing.Rescaling(1. / 255, input\_shape=IMG\_SHAPE), # 归一化，将像素值处理成0到1之间的值

tf.keras.layers.Conv2D(32, (3, 3), activation='relu'), # 卷积层，32个输出通道，3\*3的卷积核，激活函数为relu

tf.keras.layers.MaxPooling2D(2, 2), # 池化层，特征图大小减半

# Add another convolution

tf.keras.layers.Conv2D(64, (3, 3), activation='relu'), # 卷积层，32个输出通道，3\*3的卷积核，激活函数为relu

tf.keras.layers.MaxPooling2D(2, 2), # 池化层，特征图大小减半

tf.keras.layers.Flatten(), # 将二维的特征图拉直

# The same 128 dense layers, and 10 output layers as in the pre-convolution example:

tf.keras.layers.Dense(128, activation='relu'), # 128个神经元的全连接层

tf.keras.layers.Dense(class\_num, activation='softmax') # 输出层，对应数据集具体的类别数目

])

model.summary() # 输出模型信息

# 模型训练

model.compile(optimizer='adam', loss='categorical\_crossentropy',

metrics=['accuracy']) # 编译模型，指定模型的优化器是adam，模型使用的损失函数的交叉熵损失函数

return model # 返回模型

#### 4.2.2 图片匹配

# 预测图片

def predict\_img(self):

img = Image.open('images/target.png')

img = np.asarray(img)

outputs = self.model.predict(img.reshape(1, 224, 224, 3))

result\_index = int(np.argmax(outputs))

result = self.class\_names[result\_index]

names = result.split("\_")

# print(result)

if names[0] == "厨余垃圾":

self.label\_info.setText(

"厨余垃圾是指居民日常生活及食品加工、饮食服务、单位供餐等活动中产生的垃圾，包括丢弃不用的菜叶、剩菜、剩饭、果皮、蛋壳、茶渣、骨头等。由于厨余垃圾含有极高的水分与有机物，很容易腐坏，产生恶臭。经过妥善处理和加工，可转化为新的资源，高有机物含量的特点使其经过严格处理后可作为肥料、饲料，也可产生沼气用作燃料或发电，油脂部分则可用于制备生物燃料。")

if names[0] == "有害垃圾":

self.label\_info.setText(

"有害垃圾指对人体健康或者自然环境造成直接或者潜在危害的生活废弃物。常见的有害垃圾包括废灯管、废油漆、杀虫剂、废弃化妆品、过期药品、废电池、废灯泡、废水银温度计等，有害垃圾需按照特殊正确的方法安全处理，一般需要经过特殊的处理之后才可以进行焚烧，堆肥，填埋处理")

if names[0] == "可回收物":

self.label\_info.setText(

" 根据《城市生活垃圾分类及其评价标准》行业标准以及参考德国垃圾分类法，可回收物是指适宜回收循环使用和资源利用的废物。主要包括：纸类，塑料，金属，玻璃，织物等。主要的处理方式有：1.垃圾再生法；2.垃圾焚烧法；3.垃圾堆肥法；4.垃圾生物降解法。")

if names[0] == "其他垃圾":

self.label\_info.setText(

"其他垃圾指危害比较小，没有再次利用的价值的垃圾，其他垃圾包括砖瓦陶瓷、渣土、卫生间废纸、瓷器碎片、动物排泄物、一次性用品等难以回收的废弃物。一般都采取填埋、焚烧、卫生分解等方法处理，部分还可以使用生物分解的方法解决")

self.result.setText(names[0])

self.result\_f.setText(names[1])

## 5.实施与运行

### 5.1实施过程

#### 5.1.1 模型训练

过程准备:

利用pycharm编写有关机器训练的核心算法与功能结构;

在Windows系统上搭载程序运行所需的环境,安装所需的模块;

在网上查找有关垃圾分类的训练集并对原始训练图片进行筛查与选择;

模型训练:

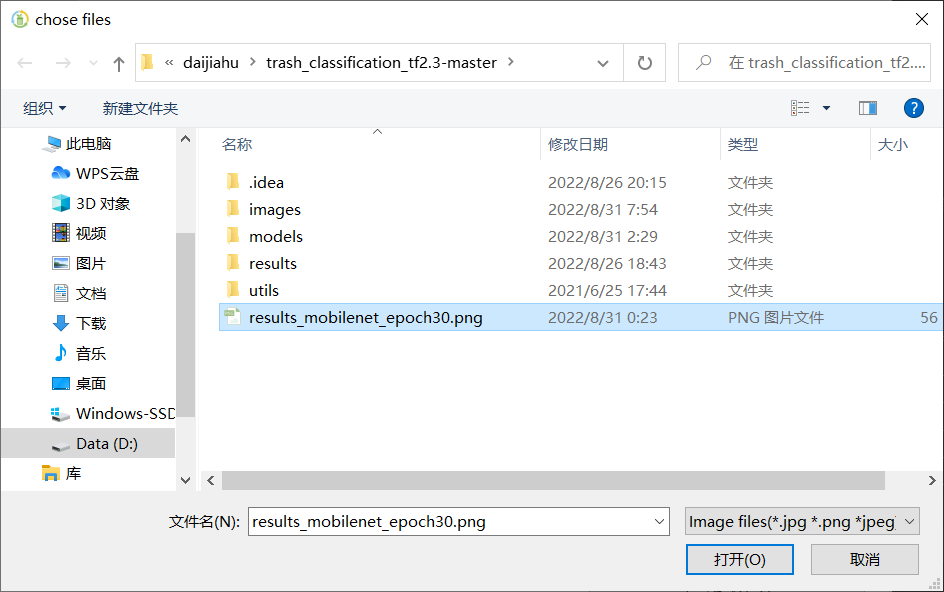
设置训练模型的数据来源路径,对训练集进行moblienet型机器学习训练,并将训练的结果以h5文件的格式保存。

#### 5.1.2 模型测试

通过keras自带的函数进行loss和accuracy的可视化，判断训练出来的模型是否可以较好地识别图片。

#### 5.1.3 外部图片上传与比对

识别检测时，将所训练出的模型文件导入到UI文件中,实现对机器学习成果的调用;同时通过手动添加上传图片路径：如图选中要判断的png文件



#### 5.1.4 比对结果输出



### 5.2运行效果评价

* 识别成功率
  + 目前比较规整的图片，如垃圾保存相对完整、周边环境相对干扰不大的情况下识别正确率较高
* 识别速度
  + 刨除人工提供上传查询图片的存储地址用时之后，识别总用时不超过1s，相对来说是比较灵敏的

## 6.总结与反思

### 6.1系统方案技术经济性评估

本项目采用cnn卷积神经网络技术，这是一类包含卷积计算且具有深度结构的前馈神经网络（Feedforward Neural Networks），是深度学习（deep learning）的代表算法之一。卷积神经网络具有表征学习（representation learning）能力，能够按其阶层结构对输入信息进行平移不变分类（shift-invariant classification），因此也被称为“平移不变人工神经网络（Shift-Invariant Artificial Neural Networks, SIANN）”，多被应用于并被应用于计算机视觉、自然语言处理等领域，在图像识别（image classification）、物体识别（object recognition）等领域上具有较大优势，正好与本项目有非常高的契合度；且该技术在网络上有较多的教程及学习资料，核心代码公开，可自行使用，成本低，易实现，效果好，具有很高的经济性。

### 6.2项目研究主要结论

在给系统划分训练集时，将垃圾分为塑料 金属 硬纸板 纸张 玻璃五类（由于缺少足够多的数据集以及训练时间，只能将训练复杂程度降低），将数据集以8：2的比例划分为训练集与测试集，在完成20轮训练后，最后该系统的准确度为87%，能够较为准确地分辨出常见的垃圾并对其进行分类，完成度较好。

### 6.3系统适用场景分析与未来改进方向

适用场景：本系统在运行后可以从本地选择jpg、jpeg、png这三种常见的图片格式来进行识别，并且程序不需要进行联网运行，在满足运行环境的情况下就能稳定运行，具有较高的便利性，功能界面简洁易懂，操作简单，拥有良好的交互体验。

未来改进方向：目前收到数据集、时间条件以及算法单一的限制，系统的准确性较低，且可识别的垃圾种类较少，在后续的工作中，要增加识别的垃圾种类数，结合其他算法，做一个更加复杂准确以及兼容性更强的系统，做成兼容各个平台及系统的应用程序是必须的，还可以增加程序的功能，不局限于图片识别垃圾种类，还可以做垃圾分类知识学习板块，帮助用户学习垃圾分类知识，养成垃圾分类的好习惯。

### 6.4课程学习过程反思

* **知识的学习上：**越是对项目的功能进行深度开发，越是对项目的问题进行改良，我们越感觉到自身所学的浅薄。在开发的过程中，我们通过各类渠道，如各类专业书籍；bilibili，知乎，CSDN等平台等自主学习了相关知识，技术水平得到显著的提升。
* **团队协作上：**良好的团队合作是项目达到预期目标的基石，团队在项目前期存在着效率不高的问题。在项目后期，我们总结了前期任务分配不明确的问题，并制定了切实可行，精确到人的每日目标。
* **项目目标制定上：**在总结前期经验教训后，团队系统学习了对项目进行可行性评估的方法，并对项目目标进行了重新审视。我们抛弃了项目中不切实际和冗余的部分，并选择了更加贴近实际，结合国家政策与科技前沿的方案，并更加注重用户的反馈和体验。
* **知识的学习上：**越是对项目的功能进行深度开发，越是对项目的问题进行改良，我们越感觉到自身所学的浅薄。在开发的过程中，我们通过各类渠道，如各类专业书籍；bilibili，知乎，CSDN等平台等自主学习了相关知识，技术水平得到显著的提升。