

华南师范大学

《人工智能导论》课程项目

课 程 项 目 报 告

项 目 题 目：垃圾图像分类

所 在 学 院：计算机学院

项 目 组 长：李骅航

小 组 成 员：张荣杰，肖灿彬

开 题 时 间：2020 年 4 月 1 日

目录

一、 引言	3
二、 国内外研究现状	4
三、 模型与算法	5
3.1 工具介绍	5
3.2 数据预处理	5
四、 实验结果分析	7
五、 结论	8
六、 参考文献	8

一、引言

垃圾分类，一般是指按一定规划或标准将垃圾分类存储、分类投放和分类搬运，从而转变成为公共资源的一系列活动的总称。分类的目的是提高垃圾的资源价值和经济价值，力争物尽其用。其中，垃圾可分为：可回收物、其他垃圾（干垃圾）、厨余垃圾（湿垃圾）、有害垃圾四种。虽然垃圾分类的种类不多，但因为我们刚实行垃圾分类政策，我国人民对某种垃圾的归属类往往分不清楚。

人工智能技术能够替代垃圾分类中的部分劳动力，从而大大降低垃圾分类所需要的人力成本，同时提升垃圾分类的效率。在这样的大背景下，垃圾分类过程中的辅助性人工智能产品大量涌现。

基于这种社会背景下，本项目希望利用深度学习方法对垃圾的图像进行分类。因硬件、时间，数据集的局限性，本项目只对六种垃圾进行分类，分别是：硬纸壳、玻璃、铁、纸张、塑料、垃圾。本项目使用基于 python 的 tensorflow2.0 技术实验，使用 Inception-ResNet 残差网络，最后取十二个图片进行可视化。

大致步骤如下：

1. 对数据进行预处理，设置每个图片中垃圾的数据类别。
2. 取出 20%的数据，将不同数据保存在不同的文件下，将数据读出后做 seed、shuffle。
3. 将 20%的数据放入测试集中，剩下的数据放入训练集中。
4. 加载每个图片并预处理数据，利用预先在 keras 中训练好的模型(分类别为 6，激活方式为 softmax，outputs)输出结果。
5. 取前十二个图片，利用可视化输出结果，大致查询准确率。

本项目训练输出结果损失值在 0.12 上下，准确率大致为 0.96。但测试实际结果只有 0.4。本项目还有很大优化空间：优化算法、数据增强等，因为时间，硬件原因不再过多尝试。

二、 国内外研究现状

2.1.1 国内研究现状

目前，已投入市场与在研发过程的产品大致可分为三大类，分别服务于垃圾分类的三个不同环节：宣传教育环节、收集环节与处理环节。

1. 服务于宣传教育环节的智能软件

2019 年 7 月上海作为第一试点城市打响了全国城市垃圾强制分类第一枪，相较于欧美、日韩等垃圾分类政策已推行了几十年的国家，我国还处于最初的起步阶段，有关垃圾分类的知识仍旧处于极为困难的普及阶段。在此环境之下，一些企业推出了基于计算机视觉技术与大数据的人工智能软件，主要都为辅助分类型。

2. 服务于收集环节的智能终端

在垃圾收集环节，智能垃圾桶是目前市场上投放最多、研发最热的人工智能产品，主要有监督分类与自动分类两种类型。

监督分类型智能垃圾桶的技术难度相对较低。通过机器视觉判断垃圾分类正确性，用人脸识别等功能将居民信用与垃圾分类行为挂钩，以此督促居民正确投放。同时智能垃圾桶的设计也囊括了通过传感器、机械结构与物联网等来解决垃圾满仓、清洁与运输便捷性等问题。国内的德澜仕智能垃圾桶便是此中的佼佼者，其配套的奖惩系统也使居民更有动力，目前已在全国投放了近 1000 套设备。

3. 服务于处理环节的智能分拣设备

目前中国垃圾处理要求做到减量化、资源化、无害化，而垃圾分类处理则是最终也是最关键的环节，垃圾分类收集环节就是为了最终分类处理而服务的。对这一环节的机械化、智能化研究目前还处于初级阶段，其研发投资与难度较大。

2.1.2 国外研究现状

2017 年，Gringgo 发布了几款与垃圾处理相关的软件。其中一款软件允许环卫工人追踪回收物的类型和数量，还能向他们推荐更加有规划的路线来节省时间，实现了人工量化回收物品，带来了潜在收入。普拉塔玛介绍，发布这些软件一年内，Gringgo 在印尼首个试点村庄将垃圾回收率提升了 35%。

今年早些时候，Gringgo 被评为“谷歌 AI 影响力挑战赛”的 20 个受资助者之一。

“我们想到可以创建一个图像识别工具，对不同垃圾进行分类，并判断其价值，进而帮助提升塑料垃圾回收率。” 普拉塔玛说。

在谷歌的帮助下，Gringgo 正和另一家创业公司合作，使用谷歌的机器学习平台 TensorFlow 研发图像识别工具。目标是让环卫工人更好地对垃圾进行分析和分类，并量化它们的价值。有了人工智能的加持，环卫工人将可以为垃圾拍照，通过图像识别判断相关物品及其价值。

三、 模型与算法

3.1 工具介绍

本项目使用基于 python 的 tensorflow2.0 技术实验，使用 Inception-ResNet 网络。

ResNet 是 Kaiming He 等人开发的残差网络，是 ILSVRC 在 2015 年的获胜者。它大量使用特殊的跳过连接和批量归一化。该架构在网络末端也没有使用全连接层。随着网络的加深，出现了训练集准确率下降的现象，因为梯度反向传播到前面的层，重复相乘可能使梯度无穷小。随着网络层数的加深，性能趋于饱和，甚至开始迅速下降。故 ResNet 针对这一问题引入了新的思想：identity shortcut connection 恒等快速链接。

Google 团队发布的 Inception-ResNet-v2，它在 ILSVRC 图像分类基准测试中实现了最好的成绩。该网络使早期 Inception V3 模型变化而来，从微软的 ResNet 论文中得到的灵感。

3.2 数据预处理

本项目利用深度学习对垃圾的图像进行分类，采用 2527 个样本的数据集，利用百分之八十的图像数据进行模型训练，利用百分之二十的图像数据用于测试模型准确率。

首先对数据进行预处理，依据设定的规则（如图一）将每个图片标记数据类后，写入文件 list.txt 中(如图二)。

```
class_names_to_ids = {'cardboard': 0, 'glass': 1, 'metal': 2, 'paper': 3, 'plastic': 4, 'trash': 5}
```

图一：设置数据类别规则

```
glass/glass263.jpg 1
paper/paper496.jpg 3
paper/paper20.jpg 3
metal/metal16.jpg 2
cardboard/cardboard210.jpg 0
plastic/plastic344.jpg 4
cardboard/cardboard53.jpg 0
glass/glass337.jpg 1
```

图二：例：标记数据类别

取出百分之二十的数据，即 505 张图片，设不同的数据保存在不同的文件下（如图三）。利用 python 的文件读取代码，将文件中图片取出，做 seed 和 shuffle 随机操作。将随机取出的 505 个图像数据集放入测试集的文件中，将剩下的图像数据集放入训练集的文件中。

```
list_path = 'list.txt'
train_list_path = 'list_train.txt'
val_list_path = 'list_val.txt'
```

图三：测试集训练集文件名

从训练集和测试集的文件中，加载每条数据的路径和标签（如图四）。循环进入每个数据的路径利用 load_img() 来加载对应的图片。同时对加载数据利用 preprocess_input() 进行预处理。

```
x_train, y_train = get_train_test_data('list_train.txt')
x_test, y_test = get_train_test_data('list_val.txt')
```

图四：解析训练集测试集文件名

3.2.1 构造模型

本项目使用预先在 keras 中训练好的 Inception-ResNetV2 模型，其中不使用该模型的最后一层全连接层，而是按照本项目需要的类别进行设置（如图五）。模型设置分类别为 6，利用 softmax 激活，利用上一层 base_model 的输出作为下一层输入。以

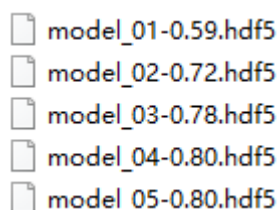
Base_model.inputs 为模型输入，outputs 为模型输出建立模型。

```
base_model = InceptionResNetV2(include_top=False, pooling='avg')
outputs = Dense(6, activation='softmax')(base_model.output)
model = Model(base_model.inputs, outputs)
```

图五：加载模型

3.2.2 模型训练

为确保模型准确率一直上升，防止数据过拟合而导致准确率下降，本项目使用 ModelCheckpoint 保存验证集中准确率较大的模型（如图六）。保存的文件中，文件名包含训练代数和验证集的准确率。



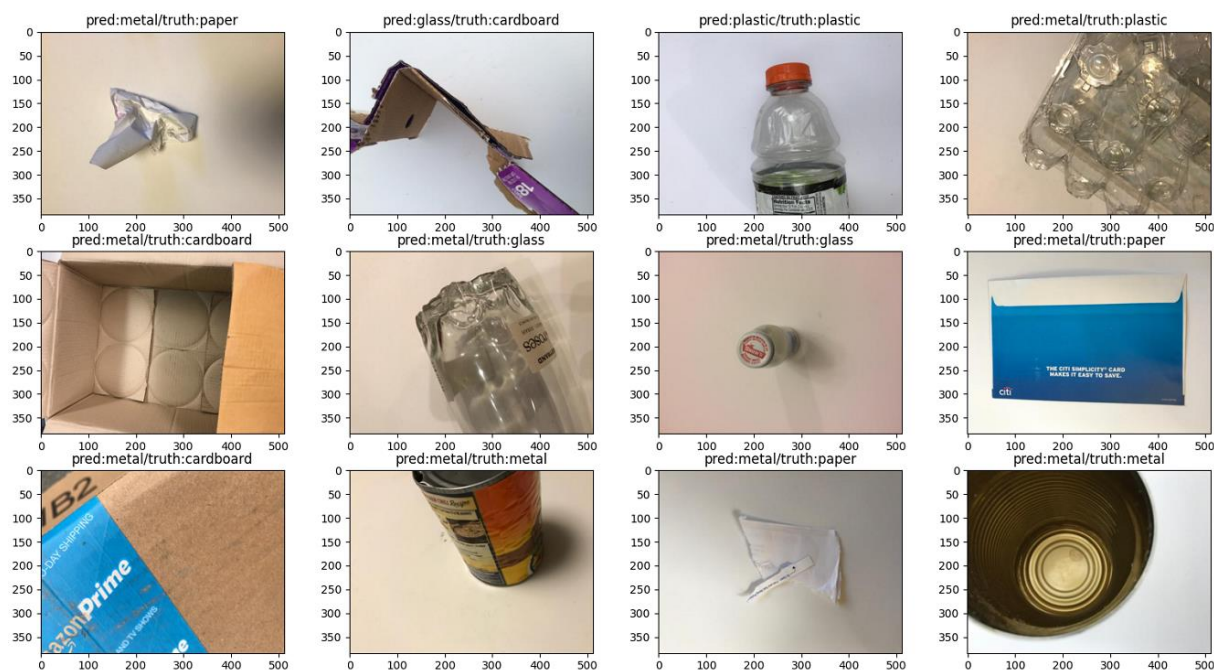
model_01-0.59.hdf5
model_02-0.72.hdf5
model_03-0.78.hdf5
model_04-0.80.hdf5
model_05-0.80.hdf5

图六：保存验证集准确率

本项目使用 adam 优化器，交叉熵损失函数，采用 top3 的准确率和 top5 的准确率。以此对 model.compile() 设置参数。在模型训练中，因硬件 GPU 问题，batch_size 为 8。训练五代次数。训练结果输出如图六所示：在第四第五代时，准确率达到百分之八十。

四、实验结果分析

将测试集数据进行预测，最终结果为百分之三十。随机取出 12 个垃圾图像数据，进行可视化处理，发现预测成功三个。小组成员认为可能是样本数太少，没有尝试数据增强，导致训练集和测试集难度不平衡，从而使训练集和测试集准确率差很远。



图七：可视化预测结果

五、 结论

在当今社会电子计算机飞速发展的整体环境下，人工智能的应用在我们的生活中已不少见，例如刷脸支付、疫情期间的红外自动测温。这是由多方面的共同发展呈现出的智慧结晶，依靠诸如大数据、云计算等先进技术为人们的生活带来便利。目前正是人工智能发展的关键阶段，解决网络模型中所遇的困难只是时间问题。

六、 参考文献

- [1]. scikit-learn (sklearn) 官方文档中文版 (cntofu.com)
- [2]. 吕思敏. 以史为鉴, 开启垃圾分类新时代. 城乡建设, 2020, (3): 30–32
- [3]. He KM, Zhang XY, Ren SQ, et al. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.
- [4]. 吴健, 陈豪, 方武. 基于计算机视觉的废物垃圾分析与识别研究. 信息技术与信息化, 2016, (10): 81–83. [doi: 10.3969/j.issn.1672-9528.2016.10.020]

附录 1

本项目垃圾分类实现代码

```
import os

import random

from idlelib import history

import matplotlib.pyplot as plt # plt 用于显示图片
import PIL

from tensorflow.python.keras.preprocessing import image
from PIL import Image

import matplotlib.image as mpimg


import numpy as np

from keras.layers import Dense
from keras.models import Model
from keras.metrics import top_k_categorical_accuracy
from keras.preprocessing import image
from keras.applications.inception_resnet_v2 import preprocess_input
from keras.preprocessing.image import load_img
from keras.callbacks import ModelCheckpoint
from sklearn.metrics import classification_report


# from keras.utils import np_utils
# from keras.utils.np_utils import *

from keras.utils.np_utils import to_categorical # 做one_hot 编码
from keras.applications.inception_resnet_v2 import InceptionResNetV2


class_names_to_ids = {'cardboard': 0, 'glass': 1, 'metal': 2, 'paper': 3,
                       'plastic': 4, 'trash': 5}
```

```
data_dir = 'dataset-resized/'
output_path = 'list.txt'
fd = open(output_path, 'w')
for class_name in class_names_to_ids.keys():
    images_list = os.listdir(data_dir + class_name)
    for image_name in images_list:
        fd.write('{} / {} {} \n'.format(class_name, image_name,
class_names_to_ids[class_name]))
fd.close

_NUM_VALIDATION = 505
_RANDOM_SEED = 0
list_path = 'list.txt'
train_list_path = 'list_train.txt'
val_list_path = 'list_val.txt'
fd = open(list_path)
lines = fd.readlines()
fd.close()
random.seed(_RANDOM_SEED)
random.shuffle(lines)
fd = open(train_list_path, 'w')
for line in lines[_NUM_VALIDATION:]:
    fd.write(line)
fd.close()
fd = open(val_list_path, 'w')
for line in lines[:_NUM_VALIDATION]:
    fd.write(line)
fd.close()
```

```
def get_train_test_data(list_file):
    list_train = open(list_file)
    x_train = [] # 获取的路径
    y_train = [] # 图像的标签
    for line in list_train.readlines():
        x_train.append(line.strip()[:-2])
        y_train.append(int(line.strip()[-1]))
        # print(line.strip())
    return x_train, y_train

x_train, y_train = get_train_test_data('list_train.txt')
x_test, y_test = get_train_test_data('list_val.txt')

def process_train_test_data(x_path):
    images = []
    for image_path in x_path:
        img_load = load_img('dataset-resized/' + image_path)
        img = image.img_to_array(img_load)
        img = preprocess_input(img)
        images.append(img)
    return images

train_images = process_train_test_data(x_train)
test_images = process_train_test_data(x_test)
```

```
base_model = InceptionResNetV2(include_top=False, pooling='avg')
outputs = Dense(6, activation='softmax')(base_model.output)
model = Model(base_model.inputs, outputs)

# 设置ModelCheckpoint, 按照验证集的准确率进行保存准确率比较大的模型 from
sklearn.metrics import classification_report

save_dir = 'train_model'

filepath = "model_{epoch:02d}-{accuracy:.2f}.hdf5" # 保留测试时第几代, 验
证准确率 (保留两位小数)

checkpoint = ModelCheckpoint(os.path.join(save_dir, filepath),
monitor='accuracy', verbose=1,

                                save_best_only=True) # 监控测试集准确率, 将结果
输出

# 模型设置

def acc_top3(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=3)

def acc_top5(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=5)

model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy', acc_top3, acc_top5])

# 使用adam 优化器, categorical_crossentropy 交叉熵损失, 监控top3 和top5 的准
确率

# 模型训练
```

```
# model.fit(np.array(train_images), to_categorical(y_train),
batch_size=8, epochs=5, shuffle=True,
#           validation_data=(np.array(test_images),
to_categorical(y_test)), callbacks=[checkpoint])

model.load_weights('train_model/model_05-0.80.hdf5') # 添加路径加载图片进
行预测

y_pred = model.predict(np.array(test_images))
ans = np.argmax(y_pred, axis=-1)

labels={0:'cardboard',1:'glass',2:'metal',3:'paper',4:'plastic',5:'trash'
}
# plt.imshow(pic) # 显示图片
plt.figure(figsize=(20, 16))
for i in range(12):
    plt.subplot(3, 4, i + 1)
    plt.title('pred:%s/truth:%s' % (labels[np.argmax(y_pred[i])],
labels[(y_test[i])]))
    pic = mpimg.imread('dataset-resized/' + x_test[i])
    plt.imshow(pic)
```
