

**Linux系统与开源软件设计**

**学 院 ：遥感信息工程学院**

**专 业 名 称 ：软件工程**

**姓 名 ： 常川**

**学 号 ：2018282110371**

**签名：**

**目录**

[一、项目说明 1](#_Toc532818484)

[1、项目背景 1](#_Toc532818485)

[2、项目业务范围 1](#_Toc532818486)

[3、项目影响 2](#_Toc532818487)

[二、ReadMe文件 3](#_Toc532818488)

[1、 运行环境及依赖库 3](#_Toc532818489)

[2、 训练数据 3](#_Toc532818490)

[3、源代码说明 3](#_Toc532818491)

[4、 使用指南 3](#_Toc532818492)

[5、测试结果示例： 4](#_Toc532818493)

[三、源代码文档 6](#_Toc532818494)

[1、train.py 6](#_Toc532818495)

[2、prepare\_image.py 8](#_Toc532818496)

[4、plot.py 10](#_Toc532818497)

[5、main.py 11](#_Toc532818498)

[四、测试文档 12](#_Toc532818499)

[1、测试图片与测试结果 12](#_Toc532818500)

[2、测试说明 13](#_Toc532818501)

# 一、项目说明

**项目github链接：https://github.com/MakeSomethingDifferent/Trash-Classification**

## 1、项目背景

中国高度重视生态环境保护，一直在大力推进垃圾分类，但许多人垃圾分类意识不高，或者缺乏对垃圾分类的了解，这些问题，在一定程度上阻碍了垃圾分类的发展。

对垃圾进行分类，可以提高资源的回收利用率。

随着人工智能的不断发展，人们生活的智能化程度不断提高，采用智能垃圾分类回收机器人，进行对垃圾的分类与回收，将会满足人们在垃圾的智能化处理方面的需求

## 2、项目业务范围

本项目最终将要实现智能垃圾分类回收，主要面向：

（一）、环卫部门：通过智能垃圾分类回收机器人对道路等公共场所进行垃圾分类回收，可以降低人力成本。

（二）、危险场所：如对风景区山脚进行垃圾分类回收，以降低人工垃圾分类回收的危险性。

（三）、游乐场所：可增加语音聊天等模块，起到吸引游客的作用。

（四）、普通家庭的垃圾分类处理。

## 3、项目影响

智能垃圾分类可以对垃圾进行分类，可以提高垃圾的回收利用率，从而达到节约社会资源的作用；

其次，采用智能垃圾，可以节约人力资源，同时降低一些场景进行垃圾分类回收带来的危险性；

最后，智能垃圾可以带动产业革命，加快中国人工智能的发展进程。

# 二、ReadMe文件

**武汉大学开源软件工程**

**trash-classification**

基于TensorFlow和Keras的智能垃圾分类系统。

## 1、 运行环境及依赖库

项目使用的python版本为python 3.6，所需的python模块包括: Tensorflow Numpy Keras cv2 h5py

## 2、 训练数据

由于训练数据比较庞大，因此不上传到github，可在下列链接进行下载： [https://www.kaggle.com/asdasdasasdas/datasets。](https://www.kaggle.com/asdasdasasdas/datasets%E3%80%82)

**3、源代码说明**

train.py：用于训练垃圾分类模型 predict.py:用于加载垃圾分类模型 prepare\_image.py:对需要进行预测的图片进行预处理 plot.py:对预测结果图片进行显示 main.py:调用各个模块对图片进行测试和显示结果

**4、 使用指南**

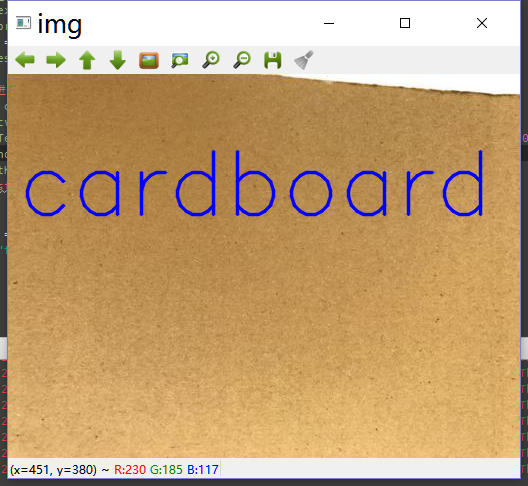
在main.py的main方法内，调用detect方法，并输入所需要进行预测的图片的路径，即可完成对图片的预测。

**5、测试结果示例：**

**测试图片：**

[](https://github.com/cchangcs/trash-classification/blob/master/%E6%B5%8B%E8%AF%95/test_pics/3.jpg)

**测试结果：**

[](https://github.com/cchangcs/trash-classification/blob/master/%E6%B5%8B%E8%AF%95/%E6%B5%8B%E8%AF%95%E7%BB%93%E6%9E%9C%E7%A4%BA%E4%BE%8B/cardboard_test_result2.png)

# 三、源代码文档

## 1、train.py

用于对垃圾分类模型进行训练。

# encoding:utf-8

'''

用于训练垃圾分类模型

'''

from keras.preprocessing.image import ImageDataGenerator

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D

from keras.layers import Activation, Dropout, Flatten, Dense

from keras.callbacks import ModelCheckpoint

from keras import backend as K

# dimensions of our images.

img\_width, img\_height = 512, 384

train\_data\_dir = 'data/train'

validation\_data\_dir = 'data/validation'

nb\_train\_samples = 2357

nb\_validation\_samples = 170

epochs = 30

batch\_size = 20

if K.image\_data\_format() == 'channels\_first':

input\_shape = (3, img\_width, img\_height)

else:

input\_shape = (img\_width, img\_height, 3)

model = Sequential()

model.add(Conv2D(32, (3, 3), input\_shape=input\_shape))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(64))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(3)) # 3分类

model.add(Activation('softmax')) # 采用Softmax

model.compile(loss='categorical\_crossentropy', # 多分类

optimizer='rmsprop',

metrics=['accuracy'])

# this is the augmentation configuration we will use for training

train\_datagen = ImageDataGenerator(

rescale=1. / 255,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True)

# this is the augmentation configuration we will use for testing:

# only rescaling

test\_datagen = ImageDataGenerator(rescale=1. / 255)

train\_generator = train\_datagen.flow\_from\_directory(

train\_data\_dir,

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

batch\_size=batch\_size,

class\_mode='categorical') # 多分类

validation\_generator = test\_datagen.flow\_from\_directory(

validation\_data\_dir,

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

batch\_size=batch\_size,

class\_mode='categorical') # 多分类

filepath="weights-improvement-{epoch:02d}-{val\_acc:.2f}.h5"

checkpoint = ModelCheckpoint(filepath, monitor='val\_acc', verbose=0, save\_best\_only=False, save\_weights\_only=False, mode='auto', period=1)

callbacks\_list = [checkpoint]

model.fit\_generator(

train\_generator,

steps\_per\_epoch=nb\_train\_samples // batch\_size,

epochs=epochs,

callbacks=callbacks\_list,

validation\_data=validation\_generator,

validation\_steps=nb\_validation\_samples // batch\_size)

## 2、prepare\_image.py

**对预测的图片进行预处理。**

# encoding:utf-8

'''

对需要进行预测的图片进行预处理

'''

from keras.preprocessing.image import ImageDataGenerator

from keras.preprocessing.image import ImageDataGenerator, load\_img, img\_to\_array

from keras.applications.imagenet\_utils import preprocess\_input

import numpy as np

def prepare\_image(img\_path, model):

# 加载图像

img = load\_img(img\_path, target\_size=(512, 384)) # x = np.array(img, dtype='float32')test

# 图像预处理

x = img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

x = preprocess\_input(x)

results = model.predict(x)

print(results)

return results

**3、predict.py**

加载训练好的垃圾分类模型权值。

# encoding:utf-8

'''

加载垃圾分类模型

'''

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D

from keras.layers import Activation, Dropout, Flatten, Dense

from keras import backend as K

def model():

# dimensions of our images.

img\_width, img\_height = 512, 384

if K.image\_data\_format() == 'channels\_first':

input\_shape = (3, img\_width, img\_height)

else:

input\_shape = (img\_width, img\_height, 3)

model = Sequential()

model.add(Conv2D(32, (3, 3), input\_shape=input\_shape))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(64))

model.add(Activation('relu'))

model.add(Dropout(0.5))

model.add(Dense(3)) # 3分类

model.add(Activation('softmax')) # 采用Softmax

model.compile(loss='categorical\_crossentropy', # 多分类

optimizer='rmsprop',

metrics=['accuracy'])

model.load\_weights("weights.h5")

return model

## 4、plot.py

显示预测结果。

# encoding:utf-8

'''

用于对预测结果图片进行显示

'''

import cv2

import numpy as np

# 根据预测结果显示对应的文字label

classes\_types = ['cardboard', 'glass', 'trash']

def generate\_result(result):

for i in range(3):

if(result[0][i] == 1):

print(i)

return classes\_types[i]

def show(img\_path, results):

# 对结果进行显示

frame = cv2.imread(img\_path)

font = cv2.FONT\_HERSHEY\_SIMPLEX

cv2.putText(frame, generate\_result(results), (10, 140), font, 3, (255, 0, 0), 2, cv2.LINE\_AA)

cv2.imshow('img', frame)

cv2.waitKey(0)

cv2.destroyAllWindows()

## 5、main.py

**调用各个模块实现对图片的测试并显示结果。**

# encoding:utf-8

'''

调用各个模块实现对图片的测试和显示结果

'''

from plot import show

import predict

from prepare\_image import prepare\_image

def detect(img\_path):

model = predict.model()

results = prepare\_image(img\_path=img\_path, model=model)

show(img\_path=img\_path, results=results)

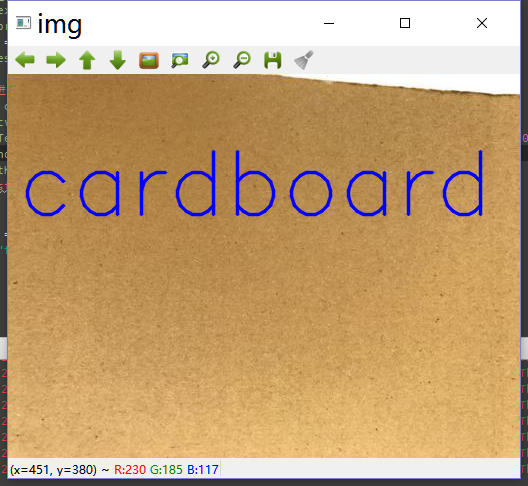
if \_\_name\_\_ == '\_\_main\_\_':

detect("test/15.jpg")

# 四、测试文档

## 1、测试图片与测试结果

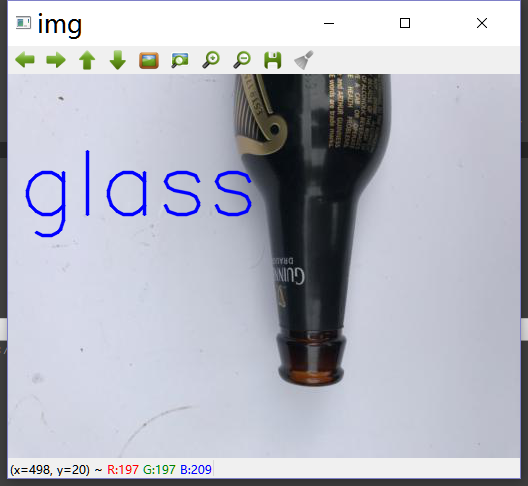
（1）、cardboard：

1. 测试图片 (b)测试结果

图1

（2）glass

1. 测试图片 (b)测试结果

图2

(3)trash:

1. 测试图片 (b)测试结果

图3

## 2、测试说明

图1:cardborad

图2:glass

图3:trash