**导入模块**

import cv2

import sys

import os.path

from glob import glob

from matplotlib import pyplot as plt

**构建主函数**

def detect(filename, cascade\_file="lbpcascade\_animeface.xml"):

if not os.path.isfile(cascade\_file):

raise RuntimeError("%s: not found" % cascade\_file)

**导入haar特征识别器**

cascade = cv2.CascadeClassifier(cascade\_file)

**读入图片**

image = cv2.imread(filename)

**色彩通道转化**

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

**灰度处理**

gray = cv2.equalizeHist(gray)

**调用detectMultiScale()函数检测**

faces = cascade.detectMultiScale(gray,

# detector options

scaleFactor=1.1,

minNeighbors=5,

minSize=(48, 48)**#识别区大小**

)

for i, (x, y, w, h) in enumerate(faces):

face = image[y: y + h, x:x + w, :]

**#此处输出裁剪后大小**

face = cv2.resize(face, (256,256))

**#可视化**

cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 10)

if opt['opencv-val']==1:

**OpenCV显示识别结果**

cv2.namedWindow('AnimeFaceDetect', cv2.WINDOW\_KEEPRATIO)

cv2.imshow("AnimeFaceDetect", image)

cv2.waitKey(0)

if opt["PLT-val"]==1:

**plt显示识别结果**

cvimg = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

plt.title(filename)

plt.xlabel('X-Px')

plt.ylabel('Y-Px')

plt.imshow(cvimg)

plt.show()

if opt["save-val-image"]==1:

**保存识别结果**

save\_filename = '%s-%d.jpg' % (os.path.basename(filename).split('.')[0], i)

cv2.imwrite("out/" + save\_filename, image)

if opt["show-fin-image"]==1:

**使用OpenCV展示裁剪结果**

cv2.imshow("img",face)

cv2.waitKey(0)

if opt["save-fin-image"]==1:

**保存裁剪结果**

save\_filename = '%s-%d.jpg' % (os.path.basename(filename).split('.')[0], i)

cv2.imwrite("Data/" + save\_filename, face)

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

if os.path.exists('Data') is False:

os.makedirs('Data')

if os.path.exists('out') is False:

os.makedirs('out')

if os.path.exists('IMG') is False:

print('file not exists!')

exit()

opt = {"opencv-val": 0,

"PLT-val": 1,

"show-fin-image":0,

"save-fin-image":0,

"save-val-image":0}

file\_list = glob('IMG/\*.jpg')

for filename in file\_list:

try:

detect(filename)

except:

print('A Error happy ...')

else:

print(str(filename) + ' done !')