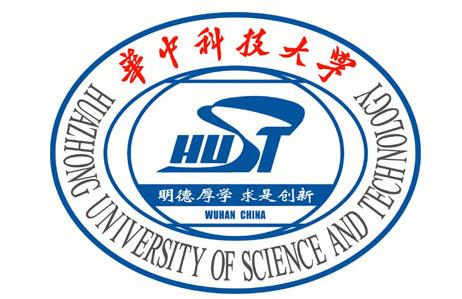
**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： 计算机1804班

学 号： U201814604

姓 名： 黄俊淇

成 绩：

指导教师： 邹复好

**完成日期： 2020年 6月19日**

# 课程Project

## 一、实验题目：人脸识别

## 二、实验要求

人脸识别功能包括人脸检测（从给定图片中用矩形框框住人脸位置，并裁剪出人脸照片）、人脸特征提取（输入人脸照片，输出是高维矢量特征）、人脸匹配（通常用K-NN算法，找出待查询人脸的最近邻居，在具体应用中，一般是1-NN）。可利用现有开源框架，完成一个人脸识别系统，能够在自己台式机或笔记本上实现实时人脸识别。测试时，人脸库中的照片可以是家庭成员，也可以是班级同学。该项目主要考核同学们对于K-NN算法和核函数的掌握，具体测试时实验设置如下：

1. 可以调整人脸库中单个人注册的人脸照片数；
2. 采用不同距离函数（L2-distance）或和核函数（余弦函数）的设置；测试不同参数设置情况下，参数变化对人脸识别系统影响。

## 三、算法设计

流程图如图1所示

其中，使用开源框架face\_recognition，编码函数使用：

手机屏幕截图

描述已自动生成

比较函数使用：

手机屏幕截图

描述已自动生成

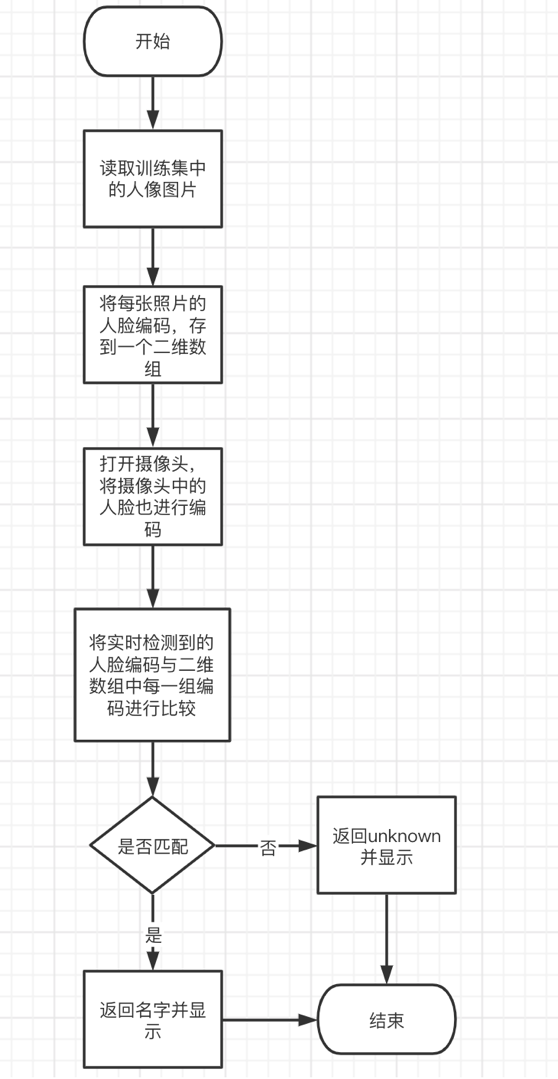


图1 流程图

## 四、实验环境与平台

macOS Catalina10.15.4

pycharmCE 2020.1.2版本，preference如图2所示

手机屏幕截图

描述已自动生成

图2 preference

## 五、程序实现

import face\_recognition

import cv2

import scipy.misc

import os

# 读取摄像头，并识别摄像头中的人脸，进行匹配。

video\_capture = cv2.VideoCapture(0)

def endwith(s,\*endstring):

resultArray = map(s.endswith,endstring)

if True in resultArray:

return True

else:

return False

def read\_file(path):

label\_list = []

dir\_counter = 0

#二维数组存储每个子文件夹下每张图片的face\_encoding.

img\_encoding = [[] for i in range(6)]

for child\_dir in os.listdir(path):

child\_path = os.path.join(path, child\_dir)

for dir\_image in os.listdir(child\_path):

if endwith(dir\_image,'JPG'):

img = scipy.misc.imread(os.path.join(child\_path, dir\_image))

img\_encoding[dir\_counter].append(face\_recognition.face\_encodings(img)[0])

label\_list.append(dir\_counter)

dir\_counter += 1

return img\_encoding,label\_list,dir\_counter

#读取训练数据集的文件夹，把他们的名字返回给一个list

def read\_name\_list(path):

name\_list = []

for child\_dir in os.listdir(path):

name\_list.append(child\_dir)

return name\_list

#main

all\_encoding, lable\_list, counter = read\_file("train")

name\_list = read\_name\_list("train")

face\_locations = []

face\_encodings = []

face\_names = []

process\_this\_frame = True

while True:

ret, frame = video\_capture.read()

small\_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

rgb\_small\_frame = small\_frame[:, :, ::-1]

if process\_this\_frame:

face\_locations = face\_recognition.face\_locations(small\_frame)

face\_encodings = face\_recognition.face\_encodings(small\_frame, face\_locations)

face\_names = []

# 匹配，并赋值

for face\_encoding in face\_encodings:

i = 0

j = 0

for t in all\_encoding:

for k in t:

match = face\_recognition.compare\_faces([k], face\_encoding,tolerance=0.48)

if True in match:

name = name\_list[i]

j = 1

i = i + 1

if j == 0:

name = "unknown"

face\_names.append(name)

process\_this\_frame = not process\_this\_frame

for (top, right, bottom, left), name in zip(face\_locations, face\_names):

top \*= 4

right \*= 4

bottom \*= 4

left \*= 4

cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), 2)

font = cv2.FONT\_HERSHEY\_DUPLEX

cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

cv2.imshow('Video', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

video\_capture.release()

cv2.destroyAllWindows()

## 六、实验结果

文字框显示意义：HJQ-我自己，ba-我爸，unknown-未识别成功

识别单张人脸，如图3所示

图片包含 人, 室内, 男人, 前

描述已自动生成

图3 单张人脸

识别多张人脸，如图4所示

图片包含 人, 室内, 男人, 照片

描述已自动生成

图4 多张人脸

遮挡脸部部分，如图5所示

图片包含 人, 室内, 男人, 眼镜

描述已自动生成

图5 识别不出

## 七、结果分析

（1）比较函数中的参数tolerance的取值对于结果的影响。

手机屏幕截图

描述已自动生成

由此可以知道这里的tolerance是容许两张脸不同的误差，超过该误差值则识别两张脸属于两个人。

按官网的说法，最佳是0.6，我按照0.6进行测试，并保证每个人注册的照片数量是一定的，且人脸清晰可见。如图6所示。

男人戴着眼镜

描述已自动生成

图6 tolerance=0.6

可以看到，这个时候是识别错误的，将我识别成了我爸，由于误差值过大导致两张脸分辨不清。

那么我将tolerance调到0.3，再次尝试，如图7所示。

男人戴着眼镜

描述已自动生成

图7 tolerance=0.3

可以看到，这个时候识别也是错误的，已经识别不出人脸属于谁了。因为光线不同，每个人的状态不同，导致库里照片的特征向量与实测时的特征向量的差值超过了0.3，因此无法识别。

因此，我选取了我拍的十张照片，脸的角度各不相同，来进行测试，得到表一所示数据，并绘制了图8所示曲线。

表一

手机屏幕的截图

描述已自动生成

图片包含 游戏机, 文字

描述已自动生成

图8 正确率曲线

在tolerance等于0.45的时候，十张照片都识别出我，但不用照片，进行实测，有那么一些瞬间会将我识别成我爸，但可以忽略不计。

因此，实验结果中的展示都是我取值为0.45时得到的结果。

# 参考文献

[1] https://face-recognition.readthedocs.io/en/latest/face\_recognition.html

[2]XXXXXX

[3]XXXXX