Opencv人脸识别测试教程

由于opencv安装的教程比较繁琐，我这里就不介绍了，我们提供的是可以已经安装好的镜像。

需要的材料有：显示屏一个，树莓派一个，广角摄像头一个

1. 接线图如下



1. 测试代码

### Imports ###################################################################

from picamera.array import PiRGBArray

from picamera import PiCamera

from functools import partial

import multiprocessing as mp

import cv2

import os

import time

### Setup #####################################################################

os.putenv( 'SDL\_FBDEV', '/dev/fb0' )

resX = 320

resY = 240

cx = resX / 2

cy = resY / 2

os.system( "echo 0=150 > /dev/servoblaster" )

os.system( "echo 1=150 > /dev/servoblaster" )

xdeg = 150

ydeg = 150

# Setup the camera

camera = PiCamera()

camera.resolution = ( resX, resY )

camera.framerate = 60

# Use this as our output

rawCapture = PiRGBArray( camera, size=( resX, resY ) )

# The face cascade file to be used

face\_cascade = cv2.CascadeClassifier('/home/pi/opencv-3.4.1/data/lbpcascades/lbpcascade\_frontalface.xml')

t\_start = time.time()

fps = 0

### Helper Functions ##########################################################

def get\_faces( img ):

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

faces = face\_cascade.detectMultiScale( gray )

return faces, img

def draw\_frame( img, faces ):

global xdeg

global ydeg

global fps

global time\_t

# Draw a rectangle around every face

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

cv2.rectangle( img, ( x, y ),( x + w, y + h ), ( 200, 255, 0 ), 2 )

cv2.putText(img, "Face No." + str( len( faces ) ), ( x, y ), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, ( 0, 0, 255 ), 2 )

tx = x + w/2

ty = y + h/2

if ( cx - tx > 15 and xdeg <= 190 ):

xdeg += 1

os.system( "echo 0=" + str( xdeg ) + " > /dev/servoblaster" )

elif ( cx - tx < -15 and xdeg >= 110 ):

xdeg -= 1

os.system( "echo 0=" + str( xdeg ) + " > /dev/servoblaster" )

if ( cy - ty > 15 and ydeg >= 110 ):

ydeg -= 1

os.system( "echo 1=" + str( ydeg ) + " > /dev/servoblaster" )

elif ( cy - ty < -15 and ydeg <= 190 ):

ydeg += 1

os.system( "echo 1=" + str( ydeg ) + " > /dev/servoblaster" )

# Calculate and show the FPS

fps = fps + 1

sfps = fps / (time.time() - t\_start)

cv2.putText(img, "FPS : " + str( int( sfps ) ), ( 10, 10 ), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, ( 0, 0, 255 ), 2 )

cv2.imshow( "Frame", img )

cv2.waitKey( 1 )

### Main ######################################################################

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

pool = mp.Pool( processes=4 )

fcount = 0

camera.capture( rawCapture, format="bgr" )

r1 = pool.apply\_async( get\_faces, [ rawCapture.array ] )

r2 = pool.apply\_async( get\_faces, [ rawCapture.array ] )

r3 = pool.apply\_async( get\_faces, [ rawCapture.array ] )

r4 = pool.apply\_async( get\_faces, [ rawCapture.array ] )

f1, i1 = r1.get()

f2, i2 = r2.get()

f3, i3 = r3.get()

f4, i4 = r4.get()

rawCapture.truncate( 0 )

for frame in camera.capture\_continuous( rawCapture, format="bgr", use\_video\_port=True ):

image = frame.array

if fcount == 1:

r1 = pool.apply\_async( get\_faces, [ image ] )

f2, i2 = r2.get()

draw\_frame( i2, f2 )

elif fcount == 2:

r2 = pool.apply\_async( get\_faces, [ image ] )

f3, i3 = r3.get()

draw\_frame( i3, f3 )

elif fcount == 3:

r3 = pool.apply\_async( get\_faces, [ image ] )

f4, i4 = r4.get()

draw\_frame( i4, f4 )

elif fcount == 4:

r4 = pool.apply\_async( get\_faces, [ image ] )

f1, i1 = r1.get()

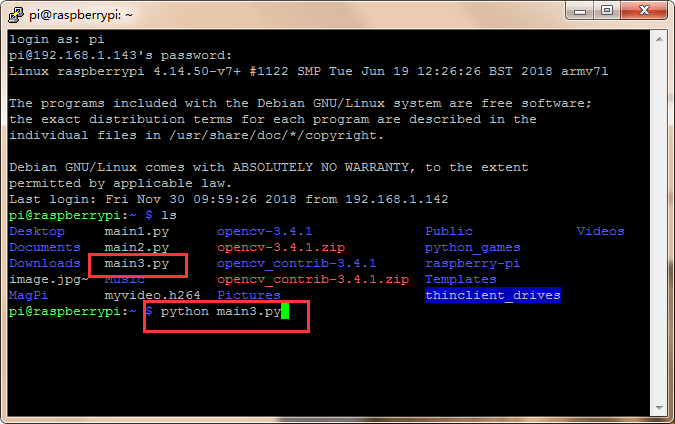
draw\_frame( i1, f1 )

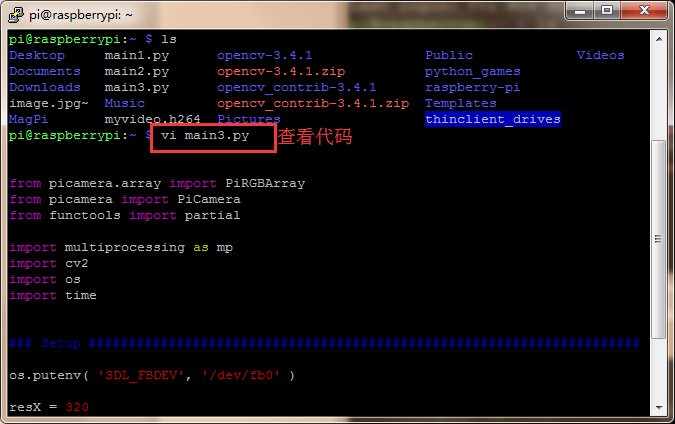
fcount = 0

fcount += 1

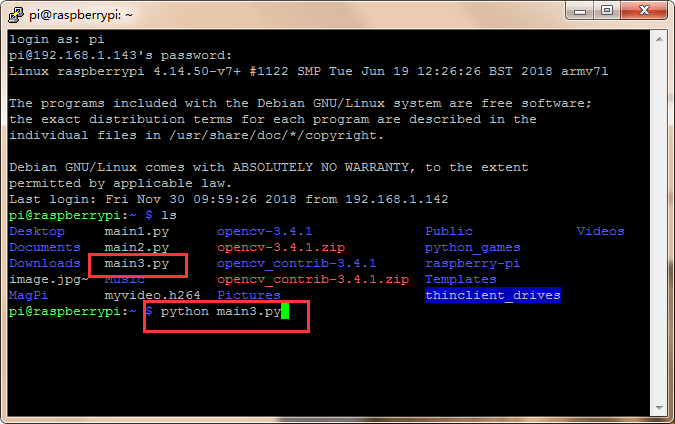
rawCapture.truncate( 0 )

1. 代码上传到树莓派上





1. 运行代码（远程运行会失败）只能在树莓派上运行



1. 识别效果



这只是测试opencv是否安装成功的例子其他的需要你们自己去扩展呢