

智能系统与控制

树莓派：OpenCV的安装 与简单应用

于泓

鲁东大学

信息与电气工程学院

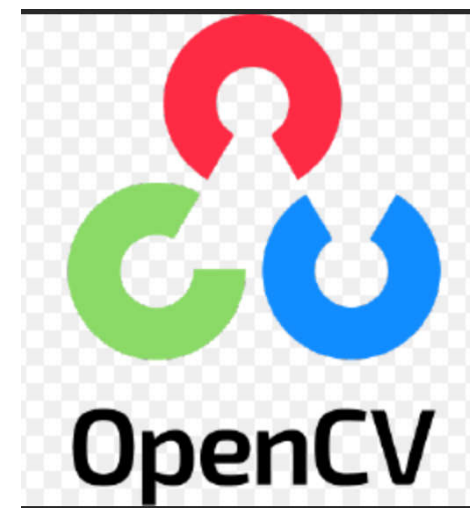
2022.1.2

OpenCV是Intel®开源计算机视觉库。它由一系列 C 函数和少量C++类构成，实现了图像处理和计算机视觉方面的很多通用算法。OpenCV 拥有包括 300 多个C函数的跨平台的中、高层 API。它不依赖于其它的外部库——尽管也可以使用某些外部库。OpenCV 对非商业应用和商业应用都是免费（FREE）的。

它有以下特点：

- 1) 开放的C/C++源码
- 2) 基于Intel处理器指令集开发的优化代码
- 3) 统一的结构和功能定义
- 4) 强大的图像和矩阵运算能力
- 5) 方便灵活的用户接口
- 6) 同时支持MS-WINDOWS、LINUX平台

作为一个基本的计算机视觉、图像处理和模式识别的开源项目，OPENCV可以直接应用于很多领域，作为第二次开发的理想工具。特别是新版的OpenCV中提供了大量的Python接口，极大的简化了Opencv函数的调用，使得使用者可以更为方便的利用OpenCV实现各种功能。



OpenCV安装

• 1 安装依赖包

```
sudo apt-get install -y libopencv-dev python3-opencv
```

```
sudo apt-get install libatlas-base-dev
```

```
sudo apt-get install libjasper-dev
```

```
sudo apt-get install libqtgui4
```

```
sudo apt-get install python3-pyqt5
```

```
sudo apt install libqt4-test
```

2 安装Python 版Opencv

```
pip3 install opencv-python
```

↓
下载有些慢
可以使用已经下载好的whl包

```
pip3 install opencv_python-4.5.5.62-cp37-cp37m-linux_armv7l.whl
```

注意：这里opencv 4.5.5 需要numpy 1.18以上

```
pip3 install numpy-1.21.5-cp37-cp37m-linux_armv7l.whl --upgrade --ignore-installed
```

安装成功后

```
pi@raspberrypi:~/EXP-Raspberry/Opencv_Base $ python3
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.5.5
>>> exit()
```

可能出现错误提示:

ImportError: /home/pi/.local/lib/python3.7/site-packages/cv2/cv2.cpython-37m-arm-linux-gnueabi.so:
undefined symbol: __atomic_fetch_add_8

修改方法：

```
pi@raspberrypi:~/EXP-Raspberry/0pencv_Base $ cd ~
pi@raspberrypi:~ $ ls -a
.                .gconf           .pki             Videos
..              .gnupg           .pp_backup       .vnc
2020-07-20-021040_1280x1024.png  GPIO             .profile         .wget-hsts
2020-09-16-190345_2032x1215_scrot.png  KEY_0.lircd.conf Public           .Xauthority
.bash_history    lesson-pi-opencv python3-lirc_1.2.1-1_armhf.deb .xorgxrdp.10.log
.bash_logout     lircd.conf       .python_history .xorgxrdp.11.log
.bashrc          lircrc           python-lirc      .xorgxrdp.11.log.old
Bookshelf        lirc-src         python-lirc_1.2.1-1_armhf.deb .xsession-errors
.cache           .local           Templates        .xsession-errors.old
.config          m_opencv        test2            yu2.txt
Desktop          Music           test_ir.py       yu.sh
Documents        onvif           testsite         yu.txt
Downloads        OpenCV          testyu
EXP_AD_DA        .pcsc11         tf
EXP-Raspberry   Pictures        thinclient_drives
pi@raspberrypi:~ $ sudo nano .bashrc
```

最后一行添加：

```
fi

export LD_PRELOAD=/usr/lib/arm-linux-gnueabihf/libatomic.so.1
```

退出后： 运行： **source .bashrc**

OpenCV的基本使用

(1) 打开并显示图像

```
import cv2

if __name__ == "__main__":
    # 读取图像并显示
    img = cv2.imread("face.bmp")
    print(img.shape)
    cv2.imshow("output", img)
    cv2.waitKey(0)
```

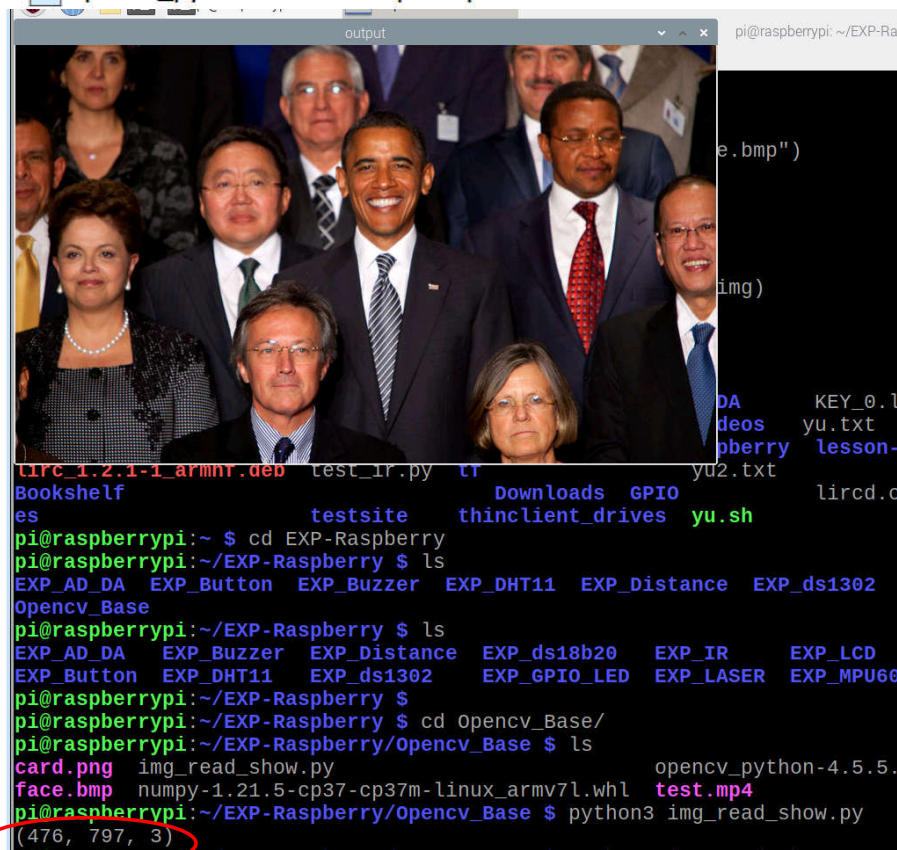
按任意键退出

输出图像是numpy
矩阵

2022/1/2

Name

img_read_show.py
test.mp4
face.bmp
card.png
numpy-1.21.5-cp37-cp37m-linux_armv7l.w
opencv_python-4.5.5.62-cp37-cp37m-linux



打开视频:

```
import cv2

if __name__ == "__main__":
    # 读取视频并显示
    # 读取视频文件
    # cap = cv2.VideoCapture('test.mp4')

    # 读取摄像头
    # cap = cv2.VideoCapture(0)

    # 读取视频流
    video = "http://admin:123456@192.168.1.17:8081/"
    cap = cv2.VideoCapture(video)

    while True:
        success, img = cap.read()
        if success:
            cv2.imshow("video", img)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    cap.release()
```

打开视频文件

打开USB 摄像头

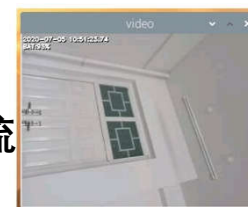
打开视频流



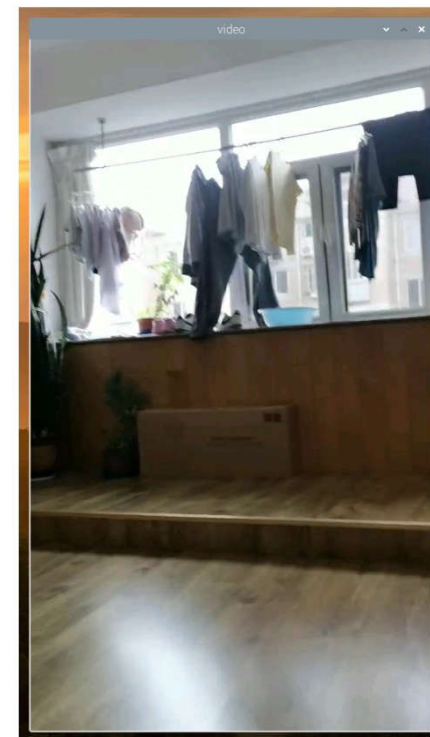
网络视频服务器



IP摄像头



网络视频



视频文件

图像的缩放和剪裁

```
import cv2
import numpy as np

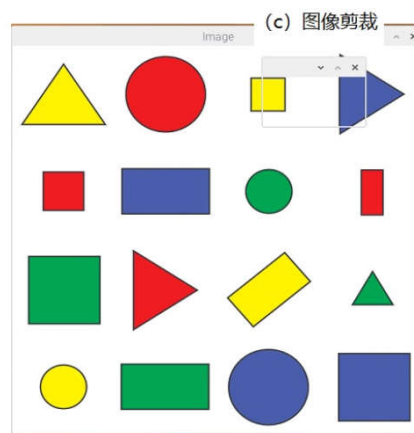
if __name__ == "__main__":
    # 读取图像
    img = cv2.imread("shapes.png")
    # 打印尺寸
    print(img.shape)

    # 图像缩放
    imgResize = cv2.resize(img, (1000, 500))
    # 打印尺寸
    print(imgResize.shape)

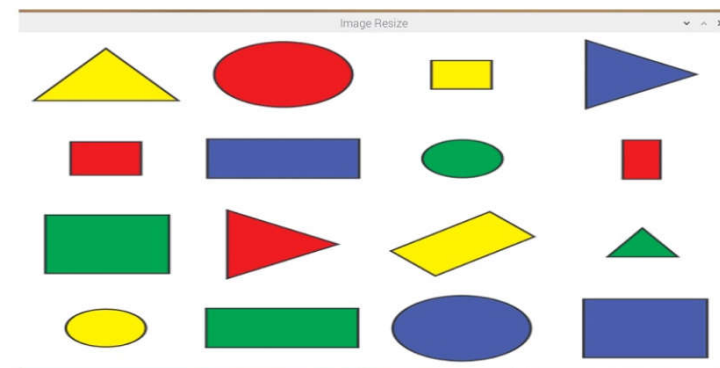
    # 图像剪裁
    imgCropped = img[46:119, 352:495]

    # 图像显示
    cv2.imshow("Image", img)
    cv2.imshow("Image Resize", imgResize)
    cv2.imshow("Image Cropped", imgCropped)
    cv2.waitKey(0)
```

Diagram illustrating the cropping operation: The original image is a 4x4 grid of shapes. The cropped region is defined by the coordinates [46:119, 352:495], which corresponds to the bottom-right 2x2 quadrant of the grid. Arrows labeled 'X' and 'Y' indicate the coordinate system for the crop operation.



(a) 原始图



(b) 图像缩放




```
import cv2
import numpy as np
from PIL import Image, ImageDraw, ImageFont
```

```
if __name__ == "__main__":
```

```
# 创建一个纯黑的图像用来进行绘图展示
img = np.zeros((512,512,3),np.uint8)
```

```
# 画直线
```

```
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),3)
```

```
# 画矩形 空心
```

```
cv2.rectangle(img,(0,0),(250,350),(0,0,255),2)
```

```
# 画矩形 实心
```

```
cv2.rectangle(img,(100,100),(200,200),(255,0,0),cv2.FILLED )
```

```
# 画圆形 空心
```

```
cv2.circle(img,(400,50),30,(255,255,0),5)
```

```
# 画圆形 实心
```

```
cv2.circle(img,(450,80),30,(0,255,255),cv2.FILLED)
```

```
#英文文字输出
```

```
#          图像    文字    位置(左下)    字体    字号    颜色    线宽
cv2.putText(img," OPENCV ",(300,200),cv2.FONT_HERSHEY_COMPLEX,1,(0,150,0),3)
```

```
# 中文文字输出
```

```
img = paint_chinese_opencv(img, "这是中文", (350,200), (0, 255,255), 20)
```

```
cv2.imshow("Image",img)
```

```
cv2.waitKey(0)
```

2022/1/4

```
def paint_chinese_opencv(im,chinese,pos,color,font_size=20):
    img_PIL = Image.fromarray(cv2.cvtColor(im,cv2.COLOR_BGR2RGB))
    font = ImageFont.truetype('NotoSansCJK-Bold.ttc',font_size,encoding="utf-8")
    fillColor = color #(255,0,0)
    position = pos #(100,100)
    draw = ImageDraw.Draw(img_PIL)
    draw.text(position,chinese,fillColor,font)
    img = cv2.cvtColor(np.asarray(img_PIL),cv2.COLOR_RGB2BGR)
    return img
```

Name	Size (B)
..	
opencv_draw.py	1
NotoSansCJK-Bold.ttc	20 441
img_resize_crop.py	1
shapes.png	101
video_read_show.py	1
img_read_show.py	1
test.mp4	775
face.bmp	1 111
card.png	1 416

字体文件
和源代码
在同一
文件夹