使用 OpenCV 的 python 接口处理视频

直接使用 python 而不借助与其他工具包来处理视频会比较繁琐。速度、解码器、摄像机、操作系统和文件格式都会有所不同。目前还没有针对 python 的视频库，我们通常使用 OpenCV 的 Python 接口来处理视频。

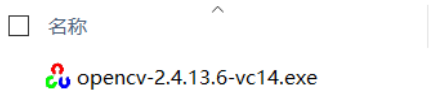
OpenCV 安装

1. 从官网下载 opencv 安装包：

https://opencv.org/releases.html

根据 python 版本进行选择，2.X 的 python 版本选择 2.X 的 opencv 版本，3.X 版本也是一样

2. 下载后双击.exe 文件进行安装

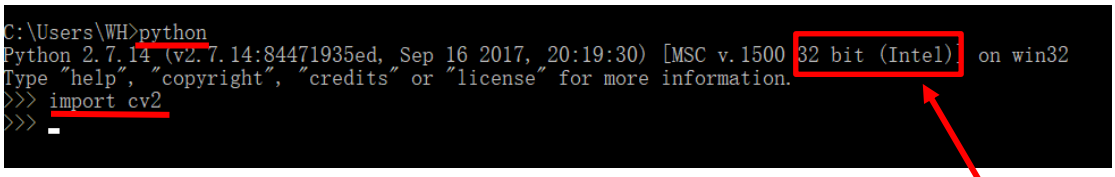


记住 opencv 安装路径，以我的安装路径 C:\opencv\为例

3. 在 opencv 安装文件夹中找到 C:\opencv\opencv\build\python\2.7\中找到 cv2.pyd 文件，并 copy 至 python 文件夹 C:\Python27\Lib\site-packages 中

此处需要注意的是 cv2.pyd 有 32 位和 64 位两个版本，根据 python 安装版本进行选择

4. 在 cmd 中输入 python，而后输入 import cv2，如果没有错误，则导入成功



使用 python 处理视频

练习 1. 从本地读取一段视频，并获取帧数，帧率以及时长

此处黑色代码是使用 Python2，蓝色代码为 python3

```
import cv2.cv as cv
#import cv2

capture = cv.CaptureFromFile('tree.avi')
#capture=cv2.VideoCapture('.\tree.avi')
nbFrames = int(cv.GetCaptureProperty(capture, cv.CV_CAP_PROP_FRAME_COUNT))
#nbFrames = intcap.get(cv2.CAP_PROP_FRAME_COUNT))
#获取其他一些视频帧相关参数
#CV_CAP_PROP_FRAME_WIDTH Width of the frames in the video stream
#CV_CAP_PROP_FRAME_HEIGHT Height of the frames in the video stream
```

读取本地视频

```
fps = cv.GetCaptureProperty(capture, cv.CV_CAP_PROP_FPS)
#fps = int(cap.get(cv2.CAP_PROP_FPS))
wait = int(1/fps * 1000/1)

duration = (nbFrames * fps) / 1000

print 'Num. Frames = ', nbFrames
#print(Num. Frames = ', nbFrames)
print 'Frame Rate = ', fps, 'fps'
#print(Frame Rate = ', fps, 'fps')
print 'Duration = ', duration, 'sec'
#print(Duration = ', duration, 'sec')
```

练习 2. 从摄像头读取视频并保存其中某一帧

```
import cv2
cap = cv2.VideoCapture(1)
while True:
    ret, im = cap.read()
    cv2.imshow('video test', im)
    key = cv2.waitKey(10)
    if key == 27:
        break
    if key == ord(' '):
        cv2.imwrite('vid_result.jpg',im)
cap.release()
cv2.destroyAllWindows()
```

变量 ret 判断视频是否正确读入，im 变量为实际读入视频帧

函数 waitKey()等待用户按键：如果按下 esc 键（ascii 码 27），则退出应用；如果按下空格键，则保存改视频帧

结束释放视频流，销毁窗口

练习 3. 在 opencv 窗口实时显示高斯模糊后的（彩色）图像

在练习 2 的例子中加入

```
blur = cv2.GaussianBlur(im,(0,0),5)
```

思考这些参数是什么意思

练习 4. 将视频读取到 NumPy 数组中

使用 OpenCV 可以从一个文件读取视频帧，并将其转换成 NumPy 数组。

```
import numpy as np
import cv2

cap = cv2.VideoCapture("tree.mp4")
wid = int(cap.get(3))
hei = int(cap.get(4))
framerate = int(cap.get(5))
framenum = int(cap.get(7))
```

```
video = np.zeros((framenum,hei,wid,3),dtype='float16')

cnt = 0
while(cap.isOpened()):
    a,b=cap.read()
    cv2.imshow('%d'%cnt, b)
    cv2.waitKey(20)
    b = b.astype('float16')/255
    video[cnt]=b
    #print(cnt)
    cnt+=1
```

练习 5.光流（选做）

光流（optical flow）是空间运动物体在观察成像平面上的像素运动的瞬时速度。

卢卡斯- Kanade 光流

```
import numpy as np
import cv2

cap=cv2.VideoCapture(".\basketball.flv")

# Shi-Tomasi 算法检测拐角的参数
feature_params = dict( maxCorners = 100,
                       qualityLevel = 0.3,
                       minDistance = 7,
                       blockSize = 7 )

# Parameters for lucas kanade optical flow
lk_params = dict( winSize  = (15,15),
                  maxLevel = 2,
                  criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03))

# Create some random colors
color = np.random.randint(0,255,(100,3))

# Take first frame and find corners in it
ret, old_frame = cap.read()
old_gray = cv2.cvtColor(old_frame, cv2.COLOR_BGR2GRAY)
p0 = cv2.goodFeaturesToTrack(old_gray, mask = None, **feature_params)

# Create a mask image for drawing purposes
mask = np.zeros_like(old_frame)

while(1):
```

```
    ret,frame = cap.read()
    frame_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # calculate optical flow
    p1, st, err = cv2.calcOpticalFlowPyrLK(old_gray, frame_gray, p0, None, **lk_params)

    # Select good points
    good_new = p1[st==1]
    good_old = p0[st==1]

    # draw the tracks
    for i,(new,old) in enumerate(zip(good_new,good_old)):
        a,b = new.ravel()
        c,d = old.ravel()
        mask = cv2.line(mask, (a,b),(c,d), color[i].tolist(), 2)
        frame = cv2.circle(frame,(a,b),5,color[i].tolist(),-1)
    img = cv2.add(frame,mask)

    cv2.imshow('frame',img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

    # Now update the previous frame and previous points
    old_gray = frame_gray.copy()
    p0 = good_new.reshape(-1,1,2)

cv2.destroyAllWindows()
cap.release()
```

OpenCV 中的密集光流

```
import cv2
```

```
import numpy as np
cap = cv2.VideoCapture("vtest.avi")

ret, frame1 = cap.read()
prvs = cv2.cvtColor(frame1,cv2.COLOR_BGR2GRAY)
hsv = np.zeros_like(frame1)
hsv[...,1] = 255

while(1):
    ret, frame2 = cap.read()
    next = cv2.cvtColor(frame2,cv2.COLOR_BGR2GRAY)

    flow = cv2.calcOpticalFlowFarneback(prvs,next, None, 0.5, 3, 15, 3, 5, 1.2, 0)

    mag, ang = cv2.cartToPolar(flow[...,0], flow[...,1])
    hsv[...,0] = ang*180/np.pi/2
    hsv[...,2] = cv2.normalize(mag,None,0,255,cv2.NORM_MINMAX)
    rgb = cv2.cvtColor(hsv,cv2.COLOR_HSV2BGR)

    cv2.imshow('frame2',rgb)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
    elif k == ord('s'):
        cv2.imwrite('opticalfb.png',frame2)
        cv2.imwrite('opticalhsv.png',rgb)
    prvs = next

cap.release()
cv2.destroyAllWindows()
```