

# OpenCV-python 学习笔记 OpenCV视频操作

## 1. 使用摄像头捕获视频

`cv2.VideoCapture()`

输入参数为0为“计算机默认摄像头”，1可以更换来源。

例如：

```
1 import numpy as np
2 import cv2
3
4 cap = cv2.VideoCapture(0) # 计算机摄像头
5 while (True):
6     # 逐帧捕捉图像
7     ret, frame = cap.read()
8
9     # our operation on the frame come here
10    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # 转换为灰
    度图像
11
12    # display the resulting frame
13    cv2.imshow('frame', gray) # 展示当前帧
14    if cv2.waitKey(1) & 0xFF == ord('q'): # 按q键退出
15        break
16 # when everything done , release the capture
17 cap.release()
18 cv2.destroyAllWindows() # 关闭所有窗口
```

当代码报错时，可以使用 `cap.isOpened()` 来检查是否成功初始化，若返回值为 `True`，说明摄像头正常打开，否则需要使用 `cap.open()`。

获取视频的参数信息，使用 `cap.get(propId)` 来获取视频的一些参数信息，`propId` 可以是0~18的任何数，每个数代表一个属性。

修改某些值，可以使用 `cap.set(propId, value)` 来修改，例如 `cap.get(3)` 和 `cap.get(4)` 来查看每一帧的宽和高，默认是 640x480。我们可以使用 `ret=cap.set(3,320)` 和 `ret = cap.set(4,240)` 来把宽和高改成 320x240。

## 2. 从文件中播放视频

`cv2.VideoCapture()`

将原先填入的“设备索引号”改成“文件名”即可。

在播放每一帧时，使用 `cv2.waitKey()` 适当持续一段时间，一般可以设置为 `25ms`（帧率）。

例如：

```
1  import cv2
2
3  cap = cv2.VideoCapture('test vedio.mov') #视频文件
4  while (True):
5      # 逐帧捕捉图像
6      ret, frame = cap.read()
7
8      # our operation on the frame come here
9      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # 转换为灰
    度图像
10
11     # display the resulting frame
12     cv2.imshow('frame', gray) # 展示当前帧
13     cv2.waitKey(25)
14     if cv2.waitKey(1) & 0xFF == ord('q'): # 按q键退出
15         break
16 # when everything done , release the capture
17 cap.release()
18 cv2.destroyAllWindows() # 关闭所有窗口
```

修改视频流属性：

- **CV\_CAP\_PROP\_POS\_MSEC** Current position of the video file in milliseconds.
  - **CV\_CAP\_PROP\_POS\_FRAMES** 0-based index of the frame to be decoded/captured next.
  - **CV\_CAP\_PROP\_POS\_AVI\_RATIO** Relative position of the video file: 0 - start of the film, 1 - end of the film.
  - **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.
  - **CV\_CAP\_PROP\_FPS** Frame rate.
  - **CV\_CAP\_PROP\_FOURCC** 4-character code of codec.
  - **CV\_CAP\_PROP\_FRAME\_COUNT** Number of frames in the video file.
  - **CV\_CAP\_PROP\_FORMAT** Format of the Mat objects returned by retrieve() .
  - **CV\_CAP\_PROP\_MODE** Backend-specific value indicating the current capture mode.
  - **CV\_CAP\_PROP\_BRIGHTNESS** Brightness of the image (only for cameras).
  - **CV\_CAP\_PROP\_CONTRAST** Contrast of the image (only for cameras).
  - **CV\_CAP\_PROP\_SATURATION** Saturation of the image (only for cameras).
  - **CV\_CAP\_PROP\_HUE** Hue of the image (only for cameras).
  - **CV\_CAP\_PROP\_GAIN** Gain of the image (only for cameras).
  - **CV\_CAP\_PROP\_EXPOSURE** Exposure (only for cameras).
  - **CV\_CAP\_PROP\_CONVERT\_RGB** Boolean flags indicating whether images should be converted to RGB.
  - **CV\_CAP\_PROP\_WHITE\_BALANCE** Currently unsupported
  - **CV\_CAP\_PROP\_RECTIFICATION** Rectification flag for stereo cameras (note: only supported by DC1394 v 2.x backend currently)
- 
- int propID – 这个参数指定你要更改的属性。这个参数有很多可选项，这里列出一部分。
    - CV\_CAP\_PROP\_POS\_MSEC – 视频的当前位置（毫秒）
    - CV\_CAP\_PROP\_POS\_FRAMES – 视频的当前位置（帧）
    - CV\_CAP\_PROP\_FRAME\_WIDTH – 视频流的宽度
    - CV\_CAP\_PROP\_FRAME\_HEIGHT – 视频流的高度
    - CV\_CAP\_PROP\_FPS – 帧速率（帧/秒）
    - CV\_CAP\_PROP\_FOURCC – 编码器的四字符编码
  - double value – 给propID分配的值。

### 3. 保存视频

创建一个 `VideoWriter` 对象，确定输出文件名，执行 `FourCC` 编码，播放帧率和帧大小，最后设置 `isColor` 标签为 `True` 彩色。

`FourCC` 是一个4字节码，用来确定视频的编码格式

1.In Fedora : DIVX , XVID , MJPG , X264 , WMV1 , WMV2 , XVID是最好的, MJPG是高尺寸视频, X264得到小尺寸视频

2.In Windows : DIVX

设置 `FourCC` 格式时，原文里采用了 `cv2.VideoWriter_fourcc()` 这个函数，若运行程序的时候显示这个函数不存在，可以改用了 `cv2.cv.CV_FOURCC` 这个函数。

例如：

```
1 cap = cv2.VideoCapture(0)
2
3 # Define the codec and create VideoWriter object
4 fourcc = cv2.VideoWriter_fourcc(*'XVID')
5 out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
6
7 while(cap.isOpened()):
8     ret, frame = cap.read()
9     if ret==True:
10         frame = cv2.flip(frame,0)
11
12         # write the flipped frame
13         out.write(frame)
14
15         cv2.imshow('frame',frame)
16         if cv2.waitKey(1) & 0xFF == ord('q'):
17             break
18     else:
19         break
20
21 # Release everything if job is finished
22 cap.release()
23 out.release()
24 cv2.destroyAllWindows()
```