

OpenCV 视频操作

OpenCV 连接 webcam 或 USB 摄像头

OpenCV 读取摄像头并显示：导入必要库→调用摄像头→读取一帧帧图像→镜像→（灰度显示）→显示画面→退出条件（Q 键退出）→释放窗口

```
import cv2

import numpy as np

# 使用 VideoCapture，读取默认摄像头，后面的数字表示摄像头的编号，如果有多个摄像头可以换成其他数字
video = cv2.VideoCapture(0)

# 再使用 cap.read() 读取视频流，类似照片，他会以一帧帧的图片返回，所以我们需要用一个循环语句来一直获取

while True:
    # 返回的是元组
    ret, frame = video.read()

    # 镜像
    frame = cv2.flip(frame, 1)

    # 灰度
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # 显示图像
    cv2.imshow("Mac camera", frame)

    # 退出条件：q
```

```
if cv2.waitKey(10) & 0xFF == ord("q"):  
    break  
  
video.release()  
cv2.destroyAllWindows()
```

OpenCV 操作视频文件

关于视频流保存函数 `VideoWriter` 方法的使用 [🔗](#) :

https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html

- OpenCV 保存视频流：导入必要库→调用摄像头→配置保存信息（位置，名称，格式，帧率，画面宽高）→读取一帧帧图像→镜像→保存画面→（灰度显示）→显示画面→退出条件（Q 键退出）→释放窗口、`释放写`

```
import cv2  
  
import numpy as np  
  
  
# 使用 VideoCapture, 读取默认摄像头, 后面的数字表示摄像头的编号, 如果有多个摄像头可以换成其他数字  
video = cv2.VideoCapture(0)  
  
  
#帧率、宽高  
fps = 20  
  
width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH))  
height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT))  
  
  
#配置视频保存信息  
  
'''
```

这里使用 OpenCV 的 `VideoWriter` 方法来，我们看一下官网他是如何使用的

可以看到第一个参数是文件名，然后是 `fourcc` 编码，然后是 `FPS` 帧率，再是画面大小

这里需要注意的是 `Fourcc` 编码，我们再看一下文档，可以看到

Windows 系统建议用 DIVX 编码

macOS 系统建议用 MJPG、DIVX、X264

推荐用 X264、DIVX，一般 macOS 和 Windows 都适用

写法需要注意 'X264'

FPS 帧率一般根据摄像头的帧率来填写，比如我的是 20

高度、宽度可以自定义，不过我们也可以直接和原画面一样，使用 cap.get 方法获取

...

```
writer =
```

```
cv2.VideoWriter('./capture.mp4', cv2.VideoWriter_fourcc(*'MJPG'), fps, (width, height))
```

```
# 再使用 cap.read() 读取视频流，类似照片，他会以一帧帧的图片返回，所以我们需要用一个循环语句来一直获取
```

```
while True:
```

```
# 返回的是元组
```

```
ret, frame = video.read()
```

```
# 镜像
```

```
frame = cv2.flip(frame, 1)
```

```
# 灰度
```

```
# frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
# 保存视频
```

```
writer.write(frame)
```

```
# 显示图像
```

```
cv2.imshow("Mac camera", frame)
```

```
# 退出条件: q

if cv2.waitKey(10) & 0xFF == ord("q"):
    break

#释放句柄

writer.release()

video.release()

cv2.destroyAllWindows()
```

- OpenCV 读取视频文件：导入必要库→调用视频路径→异常提醒→读取一帧帧图像（可以通过 sleep 方法延缓视频播放速度）→播放视频→播放完视频后关闭窗口 & 退出条件（Q 键退出）→释放窗口

```
from logging import exception

import cv2

import numpy as np

import time

# 使用 VideoCapture，读取默认摄像头，后面的数字表示摄像头的编号，如果有多个摄像头可以换成其他数字

video = cv2.VideoCapture("./capture.mp4")

try:

    while True:

        # 返回的是元组

        ret, frame = video.read()

        #延缓视频播放速度

        time.sleep(0.05)

        #镜像
```

```

frame = cv2.flip(frame, 1)

#灰度

frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# 显示图像

cv2.imshow("Local camera", frame)

# # 退出条件: q

if cv2.waitKey(10) & 0xFF == ord("q"):

    break

except:

    print('No such file!')


#释放句柄

video.release()

cv2.destroyAllWindows()

```

OpenCV 在视频上添加文字、图形

- OpenCV 在摄像头视频流上添加文字和图形：导入必要库→调用摄像头→读取一帧图像→将 cv2 默认的 BGR 格式转换成 RGB 格式→对图像进行操作（作画）→镜像→（灰度显示）→恢复图片色彩（将 RGB 转换回 BGR）→显示画面→退出条件（Q 键退出）→释放窗口

可以考虑显示帧率在左上角，帧率计算方式：导入时间库，在程序开始设置一个起始时间，在每个画面处理函数中再获得一个时间，两个时间相减得到每个画面的处理时间。用 1 除以每个画面的处理时间即为帧率。

```

from logging import exception

import cv2

```

```
import numpy as np

import time

# 使用 VideoCapture，读取默认摄像头，后面的数字表示摄像头的编号，如果有多个摄像头可以换成其他数字
video = cv2.VideoCapture(0)

#系统启动时间
start_time = time.time()

try:
    while True:
        # 返回的是元组
        ret, frame = video.read()

        #将 cv2 的 BGR 格式转换成 RGB
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        #镜像
        frame = cv2.flip(frame, 1)

        #灰度
        #frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        #计算帧率
        current_time = time.time()
        fps = int(1/(current_time - start_time))
        start_time = current_time
```

```
#设置显示帧率
font = cv2.FONT_HERSHEY_PLAIN

frame = cv2.putText(img = frame, text = str(fps), org = (20, 50), fontFace = font,
fontScale = 3, color = (0, 255, 0), thickness = 3, lineType = cv2.LINE_AA)

#恢复颜色
frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)

# 显示图像
cv2.imshow("Local camera", frame)

## 退出条件: q
if cv2.waitKey(10) & 0xFF == ord("q"):
    break

except:
    print('No such file!')

#释放句柄
video.release()

cv2.destroyAllWindows()
```