视频处理

电子信息工程系

袁羽

目录

CONTENTS

一 视频处理

OpenCV不仅能够处理图像，还能够处理视频。视频是由大量的图像构成的，这些图像以固定的时间间隔从视频中获取。这样，就能够使用图像处理的方法对这些图像进行处理，进而达到处理视频的目的。要处理视频，需要先对视频进行读取、显示和保存等相关操作。为此，OpenCV提供了VideoCapture类和VideoWriter类的相关方法。

二 读取并显示摄像头视频

VideoCapture类提供了构造方法VideoCapture()，用于完成摄像头的初始化工作。VideoCapture()的语法格式如下： capture = cv2.VideoCapture(index)

参数说明：

capture：要打开的摄像头。

index：摄像头的设备索引。

注意：

摄像头的数量及其设备索引的先后顺序由操作系统决定，因为OpenCV没有提供查询摄像头的数量及其设备索引的任何方法。

当index的值为0时，表示要打开的是第1个摄像头；对于64位的Windows 10笔记本，当index的值为0时，表示要打开的是笔记本内置摄像头，关键代码如下：capture = cv2.VideoCapture(0)

当index的值为1时，表示要打开的是第2个摄像头；对于64位的Windows 10笔记本，当index的值为1时，表示要打开的是一个连接笔记本的外置摄像头，关键代码如下：capture = cv2.VideoCapture(1)

二 读取并显示摄像头视频

为了检验摄像头初始化是否成功，VideoCapture类提供了isOpened()方法。isOpened()方法的语法格式如下：retval = cv2.VideoCapture.isOpened()

参数说明：

retval：isOpened()方法的返回值。如果摄像头初始化成功，retval的值为True；否则，retval的值为False。

说明

在VideoCapture()的语法格式基础上，isOpened()方法的语法格式可以简写为retval = capture.isOpened()

二 读取并显示摄像头视频

摄像头初始化后，可以从摄像头中读取帧，为此VideoCapture类提供了read()方法。read()方法的语法格式如下：

retval, image = cv2.VideoCapture.read() # 可以简写为retval, image = capture.read()

参数说明：

retval：是否读取到帧。如果读取到帧，retval的值为True；否则，retval的值为False。

image：读取到的帧。因为帧指的是构成视频的图像，所以可以把“读取到的帧”理解为“读取到的图像”。

注意：

在不需要摄像头时，要关闭摄像头。为此，VideoCapture类提供了release()方法。release()方法的语法格式如下：cv2.VideoCapture.release() # 可以简写为capture.release()

例 读取并显示摄像头视频：编写一个程序，打开笔记本内置摄像头实时读取并显示视频。当按下空格键时，关闭笔记本内置摄像头，销毁显示摄像头视频的窗口。

```
import cv2

capture = cv2.VideoCapture(0) # 打开笔记本内置摄像头

while (capture.isOpened()): # 笔记本内置摄像头被打开后

    retval, image = capture.read() # 从摄像头中实时读取视频

    cv2.imshow("Video", image) # 在窗口中显示读取到的视频

    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒

    if key == 32: # 如果按下空格键

        break

capture.release() # 关闭笔记本内置摄像头

cv2.destroyAllWindows() # 销毁显示摄像头视频的窗口
```

例 编写一个程序，使用图像处理的相关方法把读取并显示的彩色视频转换为灰度视频。当按下空格键时，关闭笔记本内置摄像头，销毁显示摄像头视频的窗口。

```
import cv2
capture = cv2.VideoCapture(0, cv2.CAP_DSHOW) # 打开笔记本内置摄像头
while (capture.isOpened()): # 笔记本内置摄像头被打开后
    retval, image = capture.read() # 从摄像头中实时读取视频
    # 把彩色视频转换为灰度视频
    image_Gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    if retval == True: # 读取到摄像头视频后
        cv2.imshow("Video", image) # 在窗口中显示彩色视频
        cv2.imshow("Video_Gray", image_Gray) # 在窗口中显示灰度视频
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    if key == 32: # 如果按下空格键
        break
capture.release() # 关闭笔记本内置摄像头
cv2.destroyAllWindows() # 销毁显示摄像头视频的窗口
```

例 编写一个程序，打开笔记本内置摄像头实时读取并显示视频。当按下空格键时，关闭笔记本内置摄像头，保存并显示此时摄像头视频中的图像。

```
import cv2
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW) # 打开笔记本内置摄像头
while (cap.isOpened()): # 笔记本内置摄像头被打开后
    ret, frame = cap.read() # 从摄像头中实时读取视频
    cv2.imshow("Video", frame) # 在窗口中显示视频
    k = cv2.waitKey(1) # 图像的刷新时间为1毫秒
    if k == 32: # 按下空格键
        cap.release() # 关闭笔记本内置摄像头
        cv2.destroyWindow("Video") # 销毁名为Video的窗口
        cv2.imwrite("D:/copy.png", frame) # 保存按下空格键时摄像头视频中的图像
        cv2.imshow('img', frame) # 显示按下空格键时摄像头视频中的图像
        cv2.waitKey() # 刷新图像
        break
cv2.destroyAllWindows() # 销毁显示图像的窗口
```

例 读取并显示2个摄像头视频。

```
import cv2
cap_Inner = cv2.VideoCapture(0, cv2.CAP_DSHOW) # 打开笔记本内置摄像头
cap_Outer = cv2.VideoCapture(1, cv2.CAP_DSHOW) # 打开一个连接笔记本的外置摄像头
while (cap_Inner.isOpened() & cap_Outer.isOpened()): # 两个摄像头都被打开后
    retval, img_Inner = cap_Inner.read() # 从笔记本内置摄像头中实时读取视频
    ret, img_Outer = cap_Outer.read() # 从连接笔记本的外置摄像头中实时读取视频
    # 在窗口中显示笔记本内置摄像头读取到的视频
    cv2.imshow("Video_Inner", img_Inner)
    # 在窗口中显示连接笔记本的外置摄像头读取到的视频
    cv2.imshow("Video_Outer", img_Outer)
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    if key == 32: # 如果按下空格键
        break
cap_Inner.release() # 关闭笔记本内置摄像头
cap_Outer.release() # 关闭连接笔记本的外置摄像头
cv2.destroyAllWindows() # 销毁显示摄像头视频的窗口
```

三 播放视频文件

VideoCapture类及其方法除了能够读取并显示摄像头视频外，还能够读取并显示视频文件。当窗口根据视频文件的时长显示视频文件时，便实现了播放视频文件的效果。

VideoCapture类的构造方法VideoCapture()不仅能够完成摄像头的初始化工作，还能够完成视频文件的初始化工作。当VideoCapture()用于初始化视频文件时，其语法格式如下：

```
video = cv2.VideoCapture(filename)
```

参数说明：

video：要打开的视频。

filename：打开视频的文件名。

例 读取并显示视频文件。

```
import cv2
video = cv2.VideoCapture("python.mp4") # 打开视频文件
while (video.isOpened()): # 视频文件被打开后
    retval, image = video.read() # 读取视频文件
    # 设置“Video”窗口的宽为420，高为300
    cv2.namedWindow("Video", 0)
    cv2.resizeWindow("Video", 420, 300)
    if retval == True: # 读取到视频文件后
        cv2.imshow("Video", image) # 在窗口中显示读取到的视频文件
    else: # 没有读取到视频文件
        break
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    if key == 27: # 如果按下Esc键
        break
video.release() # 关闭视频文件
cv2.destroyAllWindows() # 销毁显示视频文件的窗口
```

调整waitKey()方法中的参数值可以控制视频文件的播播放速度。例如，当代码为cv2.waitKey(1)时，等待用户按下键盘的时间为1ms，视频文件的播放速度非常快；当代码为cv2.waitKey(50)时，等待用户按下键盘的时间为50ms，能够减缓视频文件的播放速度。

例 将视频文件由彩色视频转换为灰度视频。

```
import cv2
video = cv2.VideoCapture("机器视觉应用.mp4") # 打开视频文件
while (video.isOpened()): # 视频文件被打开后
    retval, img_Color = video.read() # 读取视频文件
    # 设置“Video”窗口的宽为420，高为300
    cv2.namedWindow("Gray", 0)
    cv2.resizeWindow("Gray", 420, 300)
    if retval == True: # 读取到视频文件后
        # 把“公司宣传.avi”由彩色视频转换为灰度视频
        img_Gray = cv2.cvtColor(img_Color, cv2.COLOR_BGR2GRAY)
        cv2.imshow("Gray", img_Gray) # 在窗口中显示读取到的视频文件
    else: # 没有读取到视频文件
        break
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    if key == 27: # 如果按下Esc键
        break
video.release() # 关闭视频文件
cv2.destroyAllWindows() # 销毁显示视频文件的窗口
```

例 编写一个程序，在播放视频文件的过程中，当按空格键时，暂停播放视频；当再次按空格键时，继续播放视频；当按Esc键时，关闭视频文件并销毁显示视频文件的窗口

```
import cv2
video = cv2.VideoCapture("机器视觉应用.mp4") # 打开视频文件
while (video.isOpened()): # 视频文件被打开后
    retval, image = video.read() # 读取视频文件
    cv2.namedWindow("Video", 0)     # 设置“Video”窗口的宽为420，高为300
    cv2.resizeWindow("Video", 420, 300)
    if retval == True: # 读取到视频文件后
        cv2.imshow("Video", image) # 在窗口中显示读取到的视频文件
    else: # 没有读取到视频文件
        break
    key = cv2.waitKey(50) # 窗口的图像刷新时间为50毫秒
    if key == 32: # 如果按下空格键
        cv2.waitKey(0) # 不刷新图像，实现暂停效果
        continue # 再按一次空格键，继续播放
    if key == 27: # 如果按下Esc键
        break
video.release() # 关闭视频文件
cv2.destroyAllWindows() # 销毁显示视频文件的窗口
```

三 播放视频文件

在实际开发中，有时需要获取视频文件的属性。为此，VideoCapture类提供了get()方法。get()方法的语法格式如下：

```
retval = cv2.VideoCapture.get(propId)
```

参数说明：

retval：获取与propId对应的属性值。

propId：视频文件的属性值。

三 播放视频文件

视频文件的属性值	含 义
cv2.CAP_PROP_POS_MSEC	视频文件播放时的当前位置（单位：ms）
cv2.CAP_PROP_POS_FRAMES	帧的索引，从0开始
cv2.CAP_PROP_POS_AVI_RATIO	视频文件的相对位置（0表示开始播放，1表示结束播放）
cv2.CAP_PROP_FRAME_WIDTH	视频文件的帧宽度
cv2.CAP_PROP_FRAME_HEIGHT	视频文件的帧高度
cv2.CAP_PROP_FPS	帧速率
cv2.CAP_PROP_FOURCC	用4个字符表示的视频编码格式
cv2.CAP_PROP_FRAME_COUNT	视频文件的帧数
cv2.CAP_PROP_FORMAT	retrieve()方法返回的Mat对象的格式
cv2.CAP_PROP_MODE	指示当前捕获模式的后端专用的值
cv2.CAP_PROP_CONVERT_RGB	指示是否应将图像转换为RGB

三　播放视频文件

说明

（1）视频是由大量的、连续的图像构成的，把其中的每一幅图像称作一帧。

（2）帧数指的是视频文件中含有的图像总数，帧数越多，视频播放时越流畅。

（3）在播放视频的过程中，把每秒显示图像的数量称作帧速率（FPS，单位：帧／s）。

（4）帧宽度指的是图像在水平方向上含有的像素总数。

（5）帧高度指的是图像在垂直方向上含有的像素总数。

例 编写一个程序，使用VideoCapture类get()方法，先获取“公司宣传.avi”的帧速率、帧数、帧宽度和帧高度，再把上述4个属性值输出在PyCharm的控制台上。

```
import cv2
video = cv2.VideoCapture("机器视觉应用.mp4") # 打开视频文件b
fps = video.get(cv2.CAP_PROP_FPS) # 获取视频文件的帧速率
frame_Count = video.get(cv2.CAP_PROP_FRAME_COUNT) # 获取视频文件的帧数
frame_Width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH)) # 获取视频文件的帧宽度
frame_Height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT)) # 获取视频文件的帧高度
# 输出获取到的属性值
print("帧速率:", fps)
print("帧数:", frame_Count)
print("帧宽度:", frame_Width)
print("帧高度:", frame_Height)
```

四 保存视频文件

在实际开发过程中，很多时候希望保存一段视频。为此，OpenCV提供了VideoWriter类。下面先来熟悉一下VideoWriter类中的常用方法。

VideoWriter类中的常用方法包括VideoWriter类的构造方法、write()方法和release()方法。其中，VideoWriter类的构造方法用于创建VideoWriter类对象，其语法格式如下：

<VideoWriter object> = cv2.VideoWriter(filename, fourcc, fps, frameSize)

参数说明：

VideoWriter object：VideoWriter类对象。

filename：保存视频时的路径（含有文件名）。

fourcc：用4个字符表示的视频编码格式。

fps：帧速率。

frameSize：每一帧的大小。

在OpenCV中，使用cv2.VideoWriter_fourcc()来确定视频编码格式。

四 保存视频文件

在OpenCV中，使用cv2.VideoWriter_fourcc()来确定视频编码格式。

fourcc值	含义
cv2.VideoWriter_fourcc('M','P','4','V')	MPEG-4编码 .mp4 可指定结果视频的大小
cv2.VideoWriter_fourcc('I', '4', '2', '0')	未压缩YUV编码类型，文件名后缀为.avi ，广泛兼容，但会产生大文件
cv2.VideoWriter_fourcc('P', 'I', 'M', 'I')	MPEG-1编码类型，文件名后缀为.avi
cv2.VideoWriter_fourcc('X', 'V', 'I', 'D')	MPEG-4编码类型，文件名后缀为.avi
cv2.VideoWriter_fourcc('T', 'H', 'E', 'O')	Ogg Vorbis编码类型,文件名后缀为.ogv
cv2.VideoWriter_fourcc('F', 'L', 'V', '1')	Flash视频，文件名后缀为.flv

四 保存视频文件

为了保存一段视频，除需要使用VideoWriter类的构造方法外，还需要使用VideoWriter类提供的write()方法。write()方法的作用是在创建好的VideoWriter类对象中写入读取到的帧，其语法格式如下：

```
cv2.VideoWriter.write(frame)
```

参数说明：

frame：读取到的帧。

注意

使用write()方法时，需要由VideoWriter类对象进行调用。例如，在创建好的VideoWriter类对象output中写入读取到的帧frame，关键代码如下：

```
output.write(frame)
```

当不需要使用VideoWriter类对象时，需要将其释放掉。为此，VideoWriter类提供了release()方法，其语法格式如下：cv2.VideoWriter.release()

例 编写一个程序，首先打开笔记本内置摄像头，实时读取并显示视频；然后按Esc键，关闭笔记本内置摄像头，销毁显示摄像头视频的窗口，并且把从打开摄像头到关闭摄像头的这段视频保存为PyCharm当前项目路径下的output.avi。

```
import cv2
capture = cv2.VideoCapture(0, cv2.CAP_DSHOW) # 打开笔记本内置摄像头
fourcc = cv2.VideoWriter_fourcc('X', 'V', 'I', 'D') # 确定视频被保存后的编码格式
output = cv2.VideoWriter("output.avi", fourcc, 20, (640, 480)) # 创建VideoWriter类对象
while (capture.isOpened()): # 笔记本内置摄像头被打开后
    retval, frame = capture.read() # 从摄像头中实时读取视频
    if retval == True: # 读取到摄像头视频后
        output.write(frame) # 在VideoWriter类对象中写入读取到的帧
        cv2.imshow("frame", frame) # 在窗口中显示摄像头视频
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    if key == 27: # 如果按下Esc键
        break
capture.release() # 关闭笔记本内置摄像头
output.release() # 释放VideoWriter类对象
cv2.destroyAllWindows() # 销毁显示摄像头视频的窗口
```

练 保存一段时长为10s的摄像头视频：编写一个程序，首先打开笔记本内置摄像头，实时读取并显示视频；然后录制一段时长为10s的摄像头视频；10s后，自动关闭笔记本内置摄像头，同时销毁显示摄像头视频的窗口，并且把这段时长为10s的摄像头视频保存为PyCharm当前项目路径下的ten_Seconds.avi。

```
import cv2
capture = cv2.VideoCapture(0, cv2.CAP_DSHOW) # 打开笔记本内置摄像头
fourcc = cv2.VideoWriter_fourcc('X', 'V', 'I', 'D') # 确定视频被保存后的编码格式
fps = 20 # 帧速率
# 创建VideoWriter类对象
output = cv2.VideoWriter("ten_Seconds.avi", fourcc, fps, (640, 480))
frame_Num = 10 * fps # 时长为10秒的摄像头视频含有的帧数
# 笔记本内置摄像头被打开且时长为10秒的摄像头视频含有的帧数大于0
while (capture.isOpened() and frame_Num > 0):
    retval, frame = capture.read() # 从摄像头中实时读取视频
    if retval == True: # 读取到摄像头视频后
        output.write(frame) # 在VideoWriter类对象中写入读取到的帧
        cv2.imshow("frame", frame) # 在窗口中显示摄像头视频
    key = cv2.waitKey(1) # 窗口的图像刷新时间为1毫秒
    frame_Num -= 1 # 时长为10秒的摄像头视频含有的帧数减少一帧
capture.release() # 关闭笔记本内置摄像头
output.release() # 释放VideoWriter类对象
cv2.destroyAllWindows() # 销毁显示摄像头视频的窗口
```

例 编写一个程序，首先读取PyCharm当前项目路径下名为“机器视觉应用.mp4”的视频文件，然后将视频文件保存为PyCharm当前项目路径下的copy.avi

```
import cv2
video = cv2.VideoCapture("机器视觉应用.mp4") # 打开视频文件
fps = video.get(cv2.CAP_PROP_FPS) # 获取视频文件的帧速率
size = (int(video.get(cv2.CAP_PROP_FRAME_WIDTH)),
        int(video.get(cv2.CAP_PROP_FRAME_HEIGHT)))# 获取视频文件的帧大小
fourcc = cv2.VideoWriter_fourcc('X', 'V', 'I', 'D') # 确定视频被保存后的编码格式
output = cv2.VideoWriter("copy.avi", fourcc, fps, size) # 创建VideoWriter类对象
while (video.isOpened()): # 视频文件被打开后
    retval, frame = video.read() # 读取视频文件
    if retval == True: # 读取到视频文件后
        output.write(frame) # 在VideoWriter类对象中写入读取到的帧
    else:
        break
print("已经保存为PyCharm当前项目路径下的copy.avi。") # 控制台输出提示信息
video.release() # 关闭视频文件
output.release() # 释放VideoWriter类对象
```

练 保存视频文件中的前10s视频：编写一个程序，首先读取PyCharm当前项目路径下的视频文件，然后将视频文件中的前10s视频保存为PyCharm当前项目路径下的ten_Seconds.avi

```
import cv2
video = cv2.VideoCapture("机器视觉应用.mp4") # 打开视频文件
fps = video.get(cv2.CAP_PROP_FPS) # 获取视频文件的帧速率
# 获取视频文件的帧大小
size = (int(video.get(cv2.CAP_PROP_FRAME_WIDTH)),
        int(video.get(cv2.CAP_PROP_FRAME_HEIGHT)))
fourcc = cv2.VideoWriter_fourcc('X', 'V', 'I', 'D') # 确定视频被保存后的编码格式
output = cv2.VideoWriter("ten_Seconds.avi", fourcc, fps, size) # 创建VideoWriter类对象
frame_Num = 10 * fps # 视频文件的前10秒视频含有的帧数
# 视频文件被打开后且视频文件的前10秒视频含有的帧数大于0
while (video.isOpened() and frame_Num > 0):
    retval, frame = video.read() # 读取视频文件
    if retval == True: # 读取到视频文件后
        output.write(frame) # 在VideoWriter类对象中写入读取到的帧
    frame_Num -= 1 # 视频文件的前10秒视频含有的帧数减少一帧
# 控制台输出提示信息
print("前10s视频已经保存为PyCharm当前项目路径下的ten_Seconds.avi。")
video.release() # 关闭视频文件
output.release() # 释放VideoWriter类对象
```

小结

视频是由一系列连续的图像构成的，这一系列连续的图像被称作帧，帧是以固定的时间间隔从视频中获取的。因为视频播放的速度就是获取帧的速度，所以把视频播放的速度称作帧速率，其单位是帧 / s（即1s内出现的图像数）。所谓视频处理，处理的对象就是从视频中获取的帧，而后使用图像处理的方法对获取的帧进行处理。OpenCV提供了VideoCapture类和VideoWriter类处理视频，

THANKS