OpenCV

[toc]

图像与视频处理与播放

图像

```
import cv2 as cv
import numpy as np

#读取图像，解决imread不能读取中文路径路径的问题
def cv_imread(file_path):
    #imdedcode读取的是RGB图像
    cv_img = cv.imdecode(np.fromfile(file_path,dtype=np.uint8),-1)
    return cv_img

img=cv_imread("D:\桌面\我的小车\OpenCV\屏幕截图 2022-09-15 095726.jpg")
cv.namedWindow("blue",cv.WINDOW_NORMAL)
#cv.resizeWindow("blue",472,676)
cv.imshow("blue",img)
cv.waitKey(0)
cv.destroyAllWindows()
```

视频

```
cv.namedWindow("window",cv.WINDOW_NORMAL)
window=cv.resizeWindow("window",(480,720))
vc=cv.VideoCapture("D:\qq聊天记录\MobileFile\VID_20220915_104559.mp4")        #VideoCapture是个类，vc是对象

while vc.isOpened():
    ret,frame=vc.read()
    if not ret:
        break

    cv.imshow("window",frame)
    #如果视频帧率为30，可设置为 1000 // 30,要取整数
    if cv.waitKey(10)==ord("q"):
        break
vc.release()
cv.destroyAllWindows()
```

图像叠加

```
#图像叠加
cat=cv_imread("D:\桌面\我的小车\cat.jpg")
dog=cv_imread("D:\桌面\我的小车\dog.jpg")
dog=cv.resize(dog,(cat.shape[1],cat.shape[0]))
new_img=cv.addWeighted(cat,0.5,dog,0.5,0)
cv_imshow("catAnddog",new_img)
```

图像阈值

ret, dst = cv2.threshold(src, thresh, maxval, type)

- src:输入图，只能输入单通道图像，通常来说为灰度图. dst:输出图

- thresh:阈值
- maxval:当像素值超过了阈值_(或者小于阈值，根据type来决定)，所赋予的值
- type:二值化操作的类型，包含以下5种类型: cv2.THRESH_BINARY; cv2.THRESH_BINARY_INV; c2.THRESH_TRUNC; cv2.THRESH_ TOZERO; cv2.THRESH_TOZERO_INV

```
cv2.THRESH_BINARY        #超过阈值部分取maxval(最大值)，否则取0
cv2.THRESH_BINARY_INV    #THRESH_BINARY的反转
cv2.THRESH_TRUNC         #大于阈值部分设为阈值，否则不变
cv2.THRESH_TOZERO        #大于阈值部分不改变，否则设为0
cv2.THRESH_TOZERO_INV    #THRESH_TOZERO的反转
```

颜色空间转换

```
cat=cv_imread("D:\桌面\cat.jpg")
cv.namedWindow("con_window")

def callback(value):
    print("当前颜色空间为",cv.getTrackbarPos("traceBar","con_window"),sep="")

color_space=[cv.COLOR_BGR2BGRA,cv.COLOR_BGR2RGB,cv.COLOR_BGR2GRAY,cv.COLOR_BGR2HSV,cv.COLOR_BGR2YUV]
cv.createTrackbar("traceBar","con_window",0,3,callback)
while 1:
    pos=cv.getTrackbarPos("traceBar","con_window")
    cvt_img=cv.cvtColor(cat,color_space[pos] )
    cv.imshow("con_window",cvt_img)
    if cv.waitKey(10)==ord("q"):
        break
cv.destroyAllWindows()
```

鼠标回调函数

OpenCV允许我们对窗口上的鼠标动作做出响应。

- **setMouseCallback(winname, callback, userdata) **

  winname 是窗口的名字, callback是回调函数, userdata是给回调函数的参数.

- **callback(event,x, y, flags, userdata)**

  回调函数必须包含这5个参数. event是事件，x、y是点鼠标的坐标点, flags主要用于组合键, userdata就是上面的setMouseCallback的userdata)

- 鼠标事件:

```
EVENT_MOUSEMOVE      0     鼠标移动
EVENT_LBUTTONDOWN     1     按下鼠标左键
EVENT_RBUTTONDOWN     2     按下鼠标右键
EVENT_MBUTTONDOWN     3    按下鼠标中键
EVENT_LBUTTONUP       4    左键释放
EVENT_RBUTTONUP       5    右键释放
EVENT_MBUTTONUP       6    中键释放
EVENT_LBUTTONDBLCLK     7      左键双击
EVENT_RBUTTONDBLCLK   8       右键双击
EVENT_MBUTTONDBLCLK   9       中键双击
EVENT_MOUSEWHEEL      10    鼠标滚轮上下滚动
EVENT_MOUSEHWHEEL      11     鼠标左右滚动
```

代码如下：

```
#鼠标事件监控
def mouse_callback(event,x,y,flags,userdata):
    print(event,x,y,flags,userdata)
    if event==cv.EVENT_LBUTTONDOWN:
        cv.destroyAllWindows()
        exit(0)
img=cv_imread("D:\桌面\cat.jpg")
#要先创建窗口
cv.namedWindow("mouse",cv.WINDOW_AUTOSIZE)

cv.setMouseCallback("mouse",mouse_callback,"data")
cv_imshow("mouse",img)
```

TrackBar

- **createTrackbar(trackbarname, winname, value, count, onChange)**

  创建TrackBar控件, value为trackbar的默认值, count为bar的最大值,最小为0,onChange为回调函数

- **getTrackbarPos(trackbarname, winname)**

  获取TrackBar当前值

  代码如下：

```
#TrackBar
cv.namedWindow("TrackBar",cv.WINDOW_AUTOSIZE)
track_img=np.zeros((480,480,3),np.uint8)

def trackbar_callback(value):
    print(value)

cv.createTrackbar("R","TrackBar",0,255,trackbar_callback)
cv.createTrackbar("G","TrackBar",0,255,trackbar_callback)
cv.createTrackbar("B","TrackBar",0,255,trackbar_callback)

while 1:
    r=cv.getTrackbarPos("R","TrackBar")
    g=cv.getTrackbarPos("G","TrackBar")
    b=cv.getTrackbarPos("B","TrackBar")

    track_img[:]=[b,g,r]
    cv.imshow("TrackBar",track_img)
    if cv.waitKey(1)==ord("q"):
        break
cv.destroyAllWindows()
```

模板函数

图片读取

```
#读取图像，解决imread不能读取中文路径路径的问题
def cv_imread(file_path):
    #imdedcode读取的是RGB图像
    cv_img = cv.imdecode(np.fromfile(file_path,dtype=np.uint8),-1)
    return cv_img
```

图片显示

```
def cv_imshow(name,img):
    cv.imshow(name,img)
    cv.waitKey(0)
    cv.destroyAllWindows()
```