## 项目结构

|  |
| --- |
|  |

## 1.创建窗口：namedWindow(“窗口名字”，可选标记)

### 可选标记有很多如

|  |
| --- |
|  |

## 2.修改窗口大小：resizeWindow(窗口名字，宽，高)，创建窗口时必须使用cv2.WINDOW\_NORNAL标记

## 3.读取图片cv2.imread(图片链接，可选标记)，需要注意的是，opencv的颜色格式是bgr和普通图片的rgb模式是不一样的。这里的可选标记有很多，下面是比较常用的标记

|  |
| --- |
|  |

## 4.显示图片cv2.imshow(窗口名称，图片对象),注意如果创建了窗口，显示图片的时候最好使用这个窗口的名字

## 5.cv2.waitKey(num),num 表示等待的时间，0表示一直等待

# 实例代码

|  |
| --- |
| **import** cv2  img = cv2.imread(**'./jerry.png'**) cv2.namedWindow(**"pic"**,cv2.WINDOW\_NORMAL) cv2.resizeWindow(**'pic'**,300,200) cv2.imshow(**"pic"**,img) key = cv2.waitKey(0) **if** key == 113: *# 按下的是字母q* cv2.destroyAllWindows() |

## 6.保存图片：cv2.imwrite(文件名，图片对象)

## 扩展：用opencv python 显示保存网络图片

|  |
| --- |
| **import requests import cv2 import numpy as np  img\_url = "https://images.pexels.com/photos/1386604/pexels-photo-1386604.jpeg?auto=compress&cs=tinysrgb&w=600" file = requests.get(img\_url) img = cv2.imdecode(np.fromstring(file.content,np.int8),1) cv2.imshow('girl',img) cv2.imwrite('mygirl.jpg',img) *# 保存图片* key = cv2.waitKey(0) if key == 113: *# 按下的是字母q* cv2.destroyAllWindows()** |

### 效果：

|  |
| --- |
|  |

## 可以把cv2显示图片的方法封装成一个函数

|  |
| --- |
|  |
|  |

## 把上面的函数保存到一个叫做utils.py模块中，方便使用

|  |
| --- |
|  |

## 然后上面显示老鼠图片的代码就可以改为下面的样子

|  |
| --- |
|  |

## 7.视频采集

|  |
| --- |
|  |

### 一个比较简单的opencv驱动摄像头程序如下

|  |
| --- |
| **import** cv2  *# 创建视频窗口* cv2.namedWindow(**"video"**,cv2.WINDOW\_NORMAL) cv2.resizeWindow(**'video'**,600,500) cam = cv2.VideoCapture(0)  **while True**:  ret,frame = cam.read()  **if not** ret:  **break** cv2.imshow(**'video'**,frame)  key = cv2.waitKey(1) *#这里不能为0，否则程序永远无法退出*  **if** key == 113:  **break** cam.release() cv2.destroyAllWindows() |

### 注意代码层次，放错了，程序会卡死

## 一个能够保持视频的opencv程序

|  |
| --- |
| **import cv2  *#定义视频编码* four\_cc = cv2.VideoWriter\_fourcc(\*'mp4v') *# 注意：这里不支持divx  #创建VideoWriter实例* writer = cv2.VideoWriter("me.mp4",four\_cc,30,(640,480))  *#创建窗口* cv2.namedWindow('video cap',cv2.WINDOW\_NORMAL) cv2.resizeWindow('video cap',640,480)  cap = cv2.VideoCapture(0)  while cap.isOpened():  ret,frame = cap.read()  if not ret:  break  cv2.imshow('video cap',frame)  writer.write(frame)  key = cv2.waitKey(int(1000/30)) *#这里的数值不能为0最好int(1000/上面的帧数)，如果数字太小，会发现视频播放很快* if key == ord('q'):  break writer.release() cap.release() cv2.destroyAllWindows()** |

## 8.控制鼠标

|  |
| --- |
|  |
|  |

### 注意：flags表示的是一些组合按键，如：16 表示按下shift键同时使用鼠标

### 实例代码如下

|  |
| --- |
| **import cv2 import numpy as np  *#定义鼠标回调函数* def mouse\_callback(event, x, y, flags, userdata):  print(event, x, y, flags, userdata)  cv2.namedWindow('mouse event',cv2.WINDOW\_NORMAL) cv2.resizeWindow('mouse event',640,360)  *#设置鼠标回调* cv2.setMouseCallback('mouse event',mouse\_callback,'123')  *#用numpy生成黑图像* img = np.zeros((360,640,3),np.uint8) while True:  cv2.imshow('mouse event',img)  key = cv2.waitKey(0)  if key == ord('q'):  break cv2.destroyAllWindows()** |

### 效果：在窗口里面移动鼠标，控制台会输出信息

|  |
| --- |
|  |

### 扩展1，实现opencv python双击鼠标关闭窗口

|  |
| --- |
| *""" 双击左键关闭窗口： """* **import** cv2  **def** mousecb(event,x,y,flags,param):  **global** open  **if** event == cv2.EVENT\_LBUTTONDBLCLK:  open = **False** cv2.namedWindow(**'mouse'**,cv2.WINDOW\_NORMAL) cv2.setMouseCallback(**'mouse'**,mousecb) img = cv2.imread(**'jerry.png'**) open = **True while** open:  cv2.imshow(**'mouse'**, img)  **if** cv2.waitKey(10) == 27:  **break** cv2.destroyAllWindows() |

### 扩展2：实现双击鼠标在屏幕上面绘制一个绿色实心圆

|  |
| --- |
| **import numpy as np import cv2  def draw\_circle(event,x,y,flags,param):  if event == cv2.EVENT\_LBUTTONDBLCLK:  cv2.circle(img,(x,y),50,(0,255,0),-1)  img = np.zeros((512,512,3),np.uint8) cv2.namedWindow('img',cv2.WINDOW\_NORMAL) cv2.setMouseCallback('img',draw\_circle) while True:  cv2.imshow('img',img)  if cv2.waitKey(20) == 27: *# 按esc退出* break cv2.destroyAllWindows()** |

### 扩展3. 实现按下鼠标左键拖动绘制圆，按下鼠标右键拖动绘制矩形

|  |
| --- |
| *""" 实现按下鼠标左键拖动绘制圆，按下鼠标右键拖动绘制矩形 """* **import** cv2 **import** numpy **as** np  drawCircle = **False** drawRect = **False** ix,iy = -1,-1  **def** draw\_shape(event,x,y,flags,param):  **global** drawCircle,drawRect,ix,iy  **if** event == cv2.EVENT\_LBUTTONDOWN:  drawCircle = **True** drawRect = **False** ix,iy = x,y  **elif** event == cv2.EVENT\_RBUTTONDOWN:  drawCircle = **False** drawRect = **True** ix,iy = x,y  **elif** event == cv2.EVENT\_MOUSEMOVE:  *# if drawCircle:  # cv2.circle(img,(x,y),50,(0,255,255),-1)* **if** drawRect:  cv2.rectangle(img,(ix,iy),(x,y),(255,255,0),-1)   **elif** event == cv2.EVENT\_LBUTTONUP:  cv2.circle(img, (x, y), 50, (0, 255, 255), -1)  drawCircle = **False  elif** event == cv2.EVENT\_RBUTTONUP:  cv2.rectangle(img, (ix, iy), (x, y), (255, 255, 0), -1)  drawRect = **False** img = np.zeros((512,512,3),np.uint8) cv2.namedWindow(**'img'**,cv2.WINDOW\_NORMAL) cv2.setMouseCallback(**'img'**,draw\_shape) **while True**:  cv2.imshow(**'img'**,img)  **if** cv2.waitKey(20) == 27: *# 按esc退出* **break** cv2.destroyAllWindows() |

### 扩展4：opencv python实现点击鼠标左键视频翻转图片，点击右键垂直翻转图片

|  |
| --- |
| *""" opencv python实现点击鼠标左键视频翻转图片，点击右键垂直翻转图片 """* **import** cv2  **def** flip(event,x,y,flags,param):  **global** img  **if** event == cv2.EVENT\_LBUTTONDOWN:  img = cv2.flip(img,1)  **elif** event == cv2.EVENT\_RBUTTONDOWN:  img = cv2.flip(img,0)  img = cv2.imread(**'mygirl.jpg'**) cv2.namedWindow(**'img'**,cv2.WINDOW\_NORMAL) cv2.setMouseCallback(**'img'**,flip) **while True**:  cv2.imshow(**'img'**,img)  **if** cv2.waitKey(20) == 27: *# 按esc退出* **break** cv2.destroyAllWindows() |

## 9.控制trackBar

|  |
| --- |
| **import numpy as np import cv2  def nothing(x):  pass  *#创建image* img = np.zeros((300,512,3),np.uint8) cv2.namedWindow("image")  cv2.createTrackbar("R","image",0,255,nothing) cv2.createTrackbar("G","image",0,255,nothing) cv2.createTrackbar("B","image",0,255,nothing)  switch = '0:OFF \n 1:ON' cv2.createTrackbar(switch,"image",0,1,nothing)  while True:  cv2.imshow('image',img)  key = cv2.waitKey(1)  if key == 27:  break  r = cv2.getTrackbarPos("R",'image')  g = cv2.getTrackbarPos("G",'image')  b = cv2.getTrackbarPos("B",'image')  s = cv2.getTrackbarPos(switch,'image')   if s == 0:  img[:] = 0  else:  img[:] = [b,g,r]  cv2.destroyAllWindows()** |

### 注意：创建trackBar用createTrackbar，获取滑块当前位置使用getTrackbarPos，img[:] = [b,g,r]表示吧黑色图片的每一个像素三个通道的颜色用这三个滑块的值来赋值，实现移动滑块关闭颜色的功能

## 10.RGB和BGR颜色空间（opencv使用BGR颜色空间）

### 颜色空间的转换可以使用image=cvtColor(image,flag)，flag有很多值，其中最常用的是下面几种

|  |
| --- |
| **COLOR\_BGR2BGRA  COLOR\_BGR2GRAY  COLOR\_BGR2HLS  COLOR\_BGR2HLS\_FULL  COLOR\_BGR2HSV  COLOR\_BGR2HSV\_FULL COLOR\_BGR2RGB** |

## 11.HSV(也叫做HSB),HSL和YUV颜色空间

|  |
| --- |
|  |
|  |

### 从图中可以看到，饱和度最低是在柱体的中心，越往外，饱和度越高，明度从下往上逐渐增加

### 为什么学习HSV？是为了更好的学习opencv，只需要根据hue就可以判断背景色

|  |
| --- |
|  |

### HSL和HSV差不多

|  |
| --- |
|  |

### 这两种颜色空间只有H是完全一样的

|  |
| --- |
|  |
|  |

### HSL和HSV的区别

|  |
| --- |
|  |
|  |

### YUV

|  |
| --- |
|  |

### 扩展：pandas库里面的DataFrame类创建的对象可以查看颜色是否重复

### 实例，把图片转化为hsv颜色空间然后提取蓝色

|  |
| --- |
| **import cv2 import numpy as np  img = cv2.imread('./test.jpeg') hsv = cv2.cvtColor(img,cv2.COLOR\_BGR2HSV) *# define range of blue color in HSV* lower\_blue = np.array([110,50,50]) upper\_blue = np.array([130,255,255])  *# Threshold the HSV image to get only blue colors* mask = cv2.inRange(hsv,lower\_blue,upper\_blue) *#将掩膜和用来图片进行位与操作* ret = cv2.bitwise\_and(img,img,mask=mask)  cv2.imshow('img',img) cv2.imshow('mask',mask) cv2.imshow('ret',ret) cv2.waitKey(0)  cv2.destroyAllWindows()** |

## 还有一个是摄像头版本

|  |
| --- |
| **import** cv2 **as** cv  **import** numpy **as** np  cap = cv.VideoCapture(0)  **while**(1):  *# Take each frame*  \_, frame = cap.read()  *# Convert BGR to HSV*  hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)  *# define range of blue color in HSV*  lower\_blue = np.array([110,50,50])  upper\_blue = np.array([130,255,255])  *# Threshold the HSV image to get only blue colors*  mask = cv.inRange(hsv, lower\_blue, upper\_blue)  *# Bitwise-AND mask and original image*  res = cv.bitwise\_and(frame,frame, mask= mask)  cv.imshow('frame',frame)  cv.imshow('mask',mask)  cv.imshow('res',res)  k = cv.waitKey(5) & 0xFF  **if** k == 27:  **break**  cv.destroyAllWindows() |

## 如何去查找确定HSV值

|  |
| --- |
| **>>>** green = np.uint8([[[0,255,0 ]]])  **>>>** hsv\_green = cv.cvtColor(green,cv.COLOR\_BGR2HSV)  **>>>** print( hsv\_green )  [[[ 60 255 255]]] |