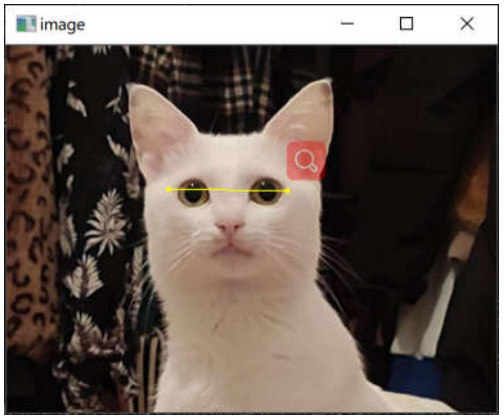
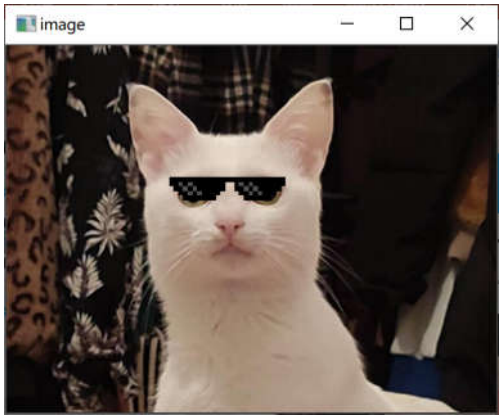




Python编程与人工智能实践

应用篇：OpenCV基础

于泓
鲁东大学
信息与电气工程学院
2021.4.11

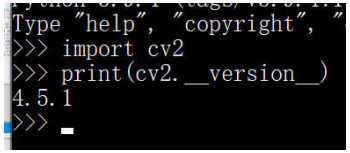
OpenCV

- OpenCV 是 Intel® 开源计算机视觉库。它由一系列 C 函数和少量 C++ 类构成，实现了图像处理和计算机视觉方面的很多通用算法。 OpenCV 拥有包括 300 多个 C 函数的跨平台的中、高层 API。它不依赖于其它的外部库——尽管也可以使用某些外部库。 OpenCV 对非商业应用和商业应用都是免费的

OpenCV安装：

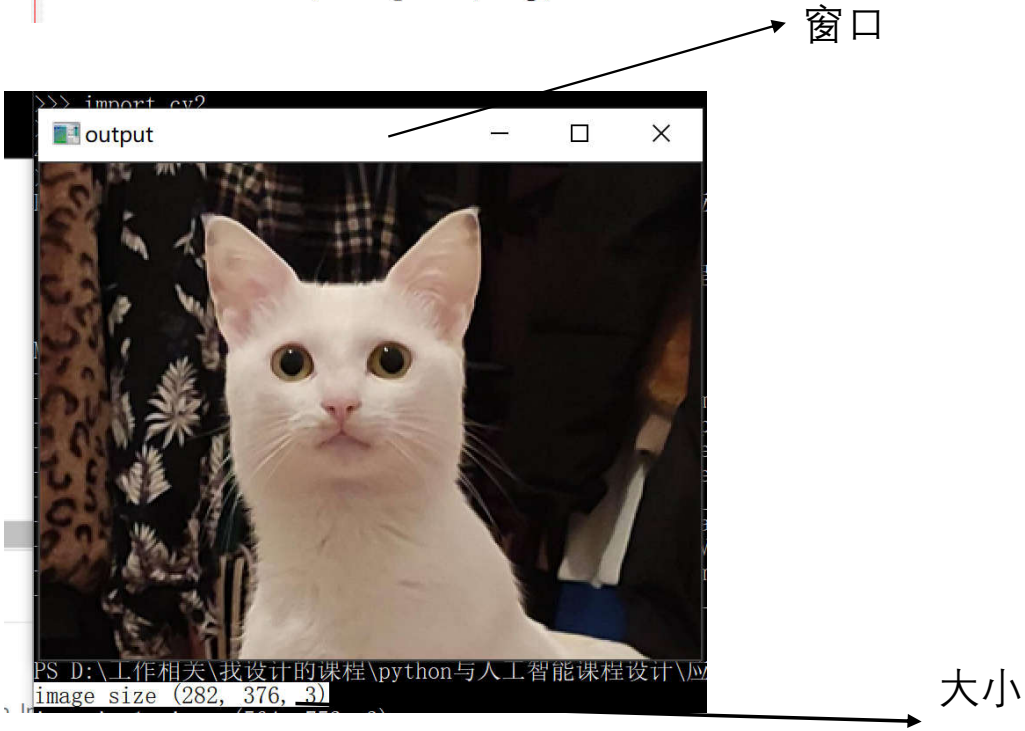
pip install opencv-python

安装成功后打印版本进行测试

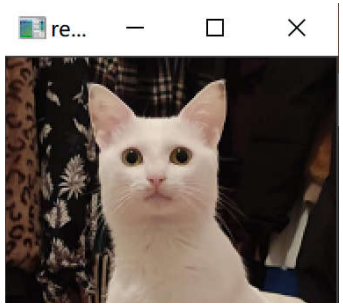


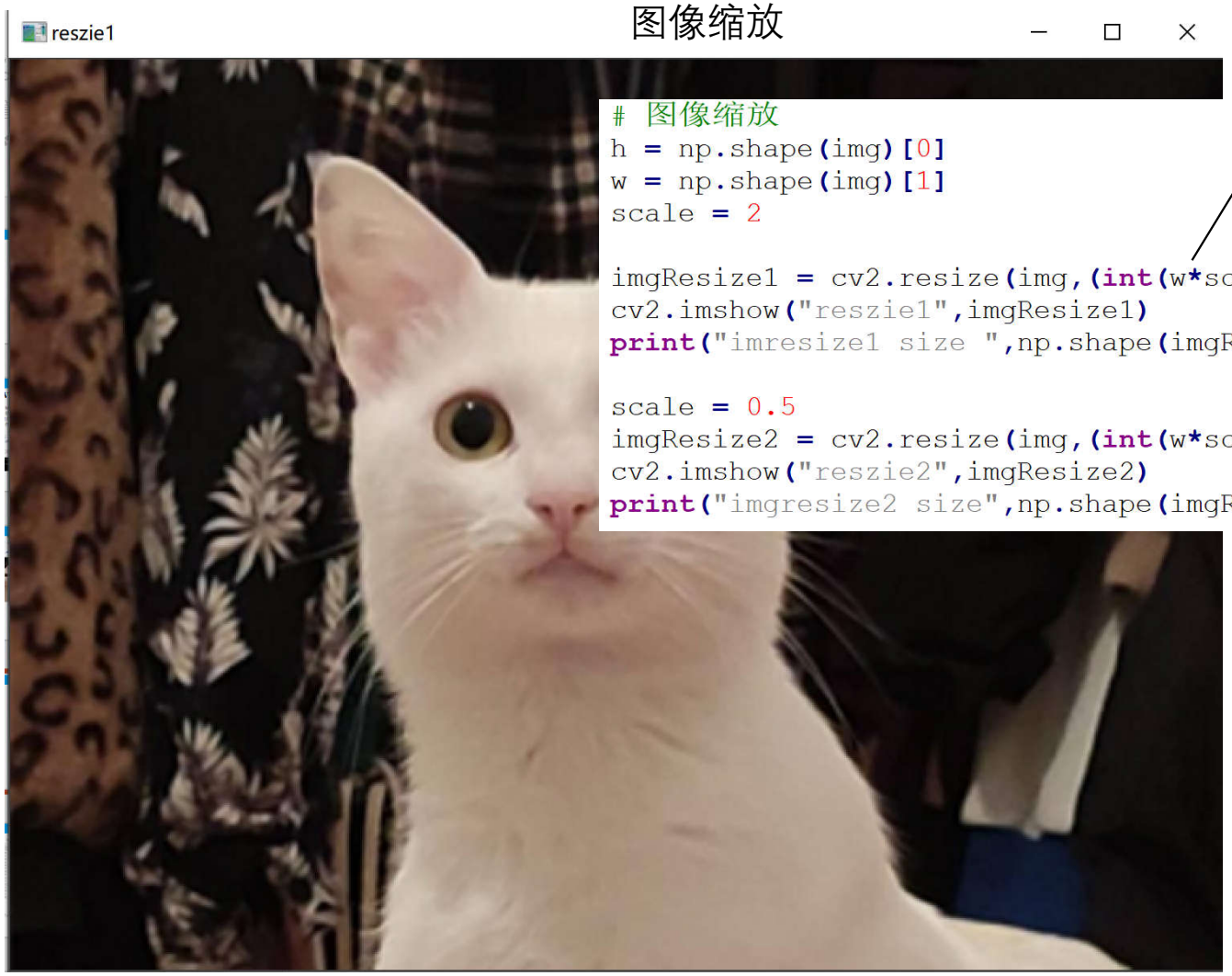
图像读取 cv2.imread

```
import cv2
import numpy as np
if __name__ == "__main__":
    # 读取图像并显示
    img= cv2.imread("cat.bmp")
    print("image size",np.shape(img))
    cv2.imshow("output",img)
```



图像缩放

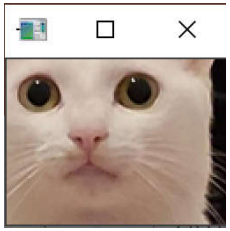


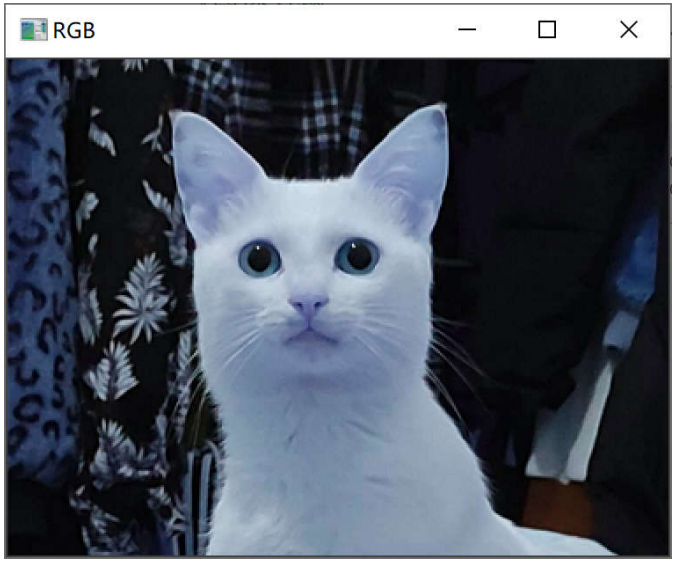


```python
# 图像缩放
h = np.shape(img)[0]
w = np.shape(img)[1]
scale = 2

imgResize1 = cv2.resize(img,(int(w*scale),int(h*scale)))
cv2.imshow("reszie1",imgResize1)
print("imresize1 size ",np.shape(imgResize1))

scale = 0.5
imgResize2 = cv2.resize(img,(int(w*scale),int(h*scale)))
cv2.imshow("reszie2",imgResize2)
print("imgresize2 size",np.shape(imgResize2))
```

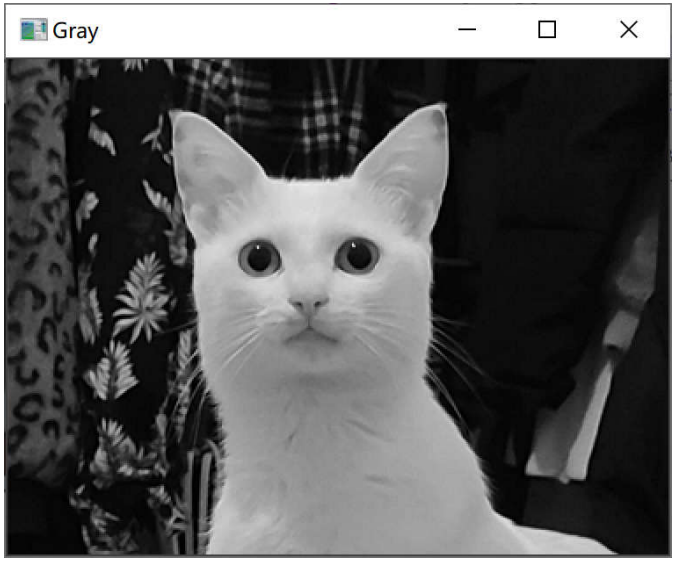
openCV中的参数一般顺序 （w,h）
(x,y)
注意与numpy维度之间的区别

```
image size (282, 376, 3)
imresize1 size  (564, 752, 3)
imgresize2 size (141, 188, 3)
```



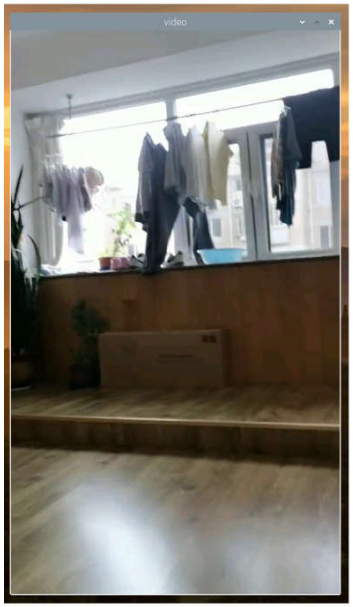



```
image size (282, 376, 3)
imresize1 size  (564, 752, 3)
imgresize2 size (141, 188, 3)
imgcropped size (94, 125, 3)
image_Gray size (282, 376)
```




```
#图像剪裁
imgCropped = img[int(h/3):int(2*h/3),int(w/3):int(w*2/3)]
cv2.imshow("cropped",imgCropped)
print("imgcropped size",np.shape(imgCropped))

# 颜色变换
img_RGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_Gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
print("image_Gray size",np.shape(img_Gray))
cv2.imshow("RGB",img_RGB)
cv2.imshow("Gray",img_Gray)

# 等待按键
cv2.waitKey(0)
```

等待任意按键
0 表示一直等待
加任意数字（ms）
表示等待一段时间
后继续执行



视频文件

网络视频服务器

视频读取

读取视频文件

读取摄像头

读取视频流

循环

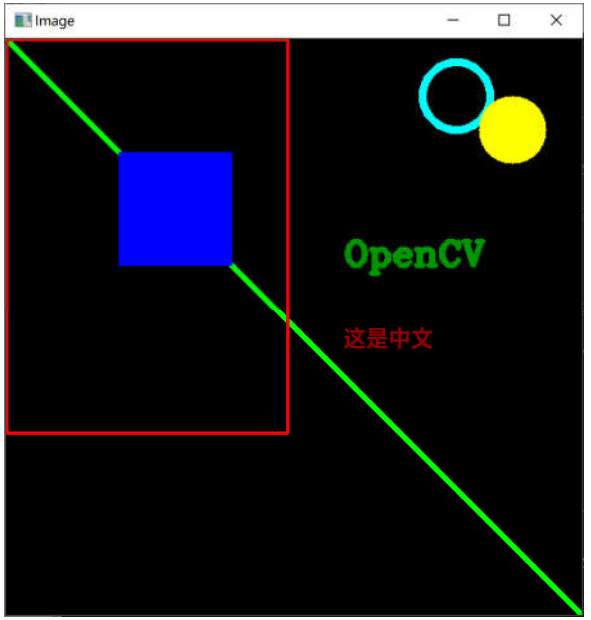
按q退出

```python
import cv2
# 读取视频并显示
# 读取视频文件
cap = cv2.VideoCapture('test.mp4')

# 读取摄像头
cap = cv2.VideoCapture(0)

# 读取视频流
video = "http://admin:admin@192.168.1.3:8081/"
cap = cv2.VideoCapture(video)

while True:
    success, img = cap.read()
    if success:
        cv2.imshow("video",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
```

OpenCV 绘图

```
if __name__=="__main__":
    # 创建一个纯黑的图像用来进行绘图展示
    img = np.zeros((512,512,3),np.uint8)

    '''画直线'''
    # 点1 坐标
    p1 = (0,0)
    # 点2 坐标
    p2 = (img.shape[1],img.shape[0])
    # 线的颜色
    color = (0,255,0)
    # 线的宽度
    size_line = 3
    cv2.line(img,p1,p2,color,size_line)

    '''画矩形 空心'''
    # 左上角坐标
    pos_rect = (0,0)
    # 矩形的 宽和高(w,h)
    size_rect  = (250,350)
    # 矩形颜色
    color = (0,0,255)
    # 线宽
    size_line = 2
    cv2.rectangle(img,pos_rect,size_rect,color,size_line)

    '''画矩形 实心'''
    cv2.rectangle(img,(100,100),(200,200),(255,0,0),cv2.FILLED )
```

```
'''画圆形 空心'''
# 圆心
p_center = (400,50)
# 半径
len_R = 30
cv2.circle(img,p_center,len_R,(255,255,0),5)

'''画圆形 实心'''
cv2.circle(img,(450,80),30,(0,255,255),cv2.FILLED)
```



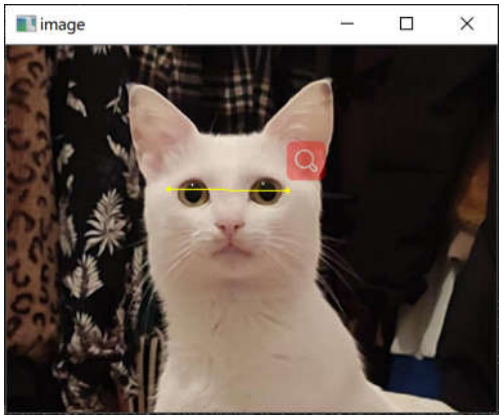
文字输出

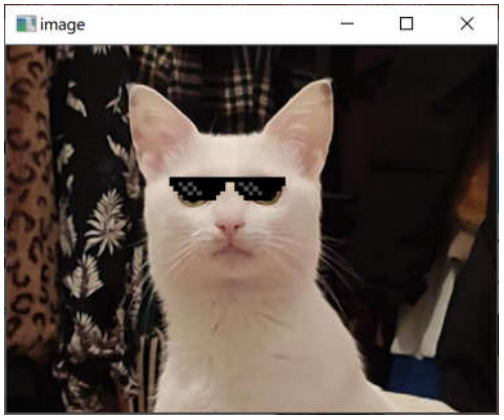
```
'''文字输出'''
# 输出的文字
str_txt = "OpenCV"

# 文字输出区域的左上角坐标
pos_txt = (300,200)

# 字体
font = cv2.FONT_HERSHEY_COMPLEX

# 字号
font_size =1

# 颜色
color = (0,150,0)

# 绘制文字的线宽
line_size = 3

cv2.putText(img,str_txt,pos_txt,font,font_size,color,line_size)
```

绘制中文 pip install Pillow

```
def paint_chinese_opencv(im,chinese,pos,color,font_size = 20):
    img_PIL = Image.fromarray(cv2.cvtColor(im,cv2.COLOR_BGR2RGB))
    font = ImageFont.truetype('NotoSansCJK-Bold.ttc',font_size,encoding="utf-8")
    fillColor = color
    position = pos

    draw = ImageDraw.Draw(img_PIL)
    draw.text(position,chinese,fillColor,font)
    img = cv2.cvtColor(np.asarray(img_PIL),cv2.COLOR_RGB2BGR)
    return img


'''中文输出'''
color_rgb = (150,0,0)
str_txt = "这是中文"
pos_text = (300,250)
img = paint_chinese_opencv(img,"这是中文",pos_text,color_rgb)

cv2.imshow("Image",img)
cv2.waitKey(0)
```

绘图加鼠标操作给小猫加眼镜





鼠标回调函数

鼠标动作

鼠标坐标

传递参数

```
def points_collect(event,x,y,flags,param):
    dic_points = param
    if event == cv2.EVENT_LBUTTONDBLCLK:
        if len(dic_points['p2'])>0:
            dic_points['p2']=()
            dic_points['p1']=(x,y)
        elif len(dic_points['p1'])==0:
            dic_points['p1']=(x,y)
        elif len(dic_points['p1'])>0:
            dic_points['p2']=(x,y)

    if event == cv2.EVENT_MOUSEMOVE:
        dic_points['p_move']=(x,y)
```

左键双击

已经有记录了则重新记录

没有记录记录P1

P1已经记录了记录P2

记录移动坐标

```
if __name__=="__main__":

    # 记录坐标的字典
    dic_points = {}
    dic_points["p1"]=()
    dic_points["p2"]=()
    dic_points["p_move"]=()

    # 设置回调函数
    cv2.namedWindow('image')
    cv2.setMouseCallback('image', points_collect,param=dic_points)
```

响应鼠标动作的窗口

P1 P2都有
连接 P1 P2

```
def drawline(img,dic_points):
    color = (0,255,255)
    if len(dic_points['p2'])>0:
        cv2.circle(img,dic_points['p1'],2,color,cv2.FILLED)
        cv2.circle(img,dic_points['p2'],2,color,cv2.FILLED)
        cv2.line(img,dic_points['p1'],dic_points['p2'],color,1)
    elif len(dic_points['p1'])>0 and len(dic_points['p_move'])>0:
        cv2.circle(img,dic_points['p1'],2,color,cv2.FILLED)
        cv2.line(img,dic_points['p1'],dic_points['p_move'],color,1)
```

有P1 无P2
连接P1 与动点


```python
# 图像叠加
def img_overlayer(img,img_fg,pos_fg,bk_fg):

    #把前景图变换为灰度
    fg_gray = cv2.cvtColor(img_fg,cv2.COLOR_BGR2GRAY)
    h_gf,w_fg = np.shape(fg_gray)


    # 获取前景图的mask 有图部分为 1 背景部分为 0
    if bk_fg == 255:
        mask_fg = fg_gray<250
    elif bk_fg == 0:
        mask_fg = fg_gray>5

    mask_fg = mask_fg[:,:,np.newaxis]
    not_mask_fg = ~mask_fg

    # 截取背景图
    bk = img[pos_fg[1]:pos_fg[1]+h_gf,pos_fg[0]:pos_fg[0]+w_fg]

    img_overlayer = bk*not_mask_fg + img_fg*mask_fg
    img[pos_fg[1]:pos_fg[1]+h_gf,pos_fg[0]:pos_fg[0]+w_fg] = img_overlayer
    return img
```

```python
def add_glass(img,img_glass,dic_points,bk_fg=255):

    if len(dic_points['p2'])==0:
        return img

    # 获取眼镜图像大小
    w_glass = np.shape(img_glass)[1]
    h_glass = np.shape(img_glass)[0]

    # 计算缩放尺度
    scale = np.abs(dic_points['p1'][0]-dic_points['p2'][0])/w_glass

    # 眼镜图像缩放
    resize_glass = cv2.resize(img_glass,(int(w_glass*scale),int(h_glass*scale)))

    # 计算眼镜图像的起始位置(左上坐标)
    pos_glass = (dic_points['p1'][0],dic_points['p1'][1]-int(h_glass*scale/2.0))

    # 图像叠加
    img_out = img_overlayer(img,resize_glass,pos_glass,bk_fg)

    return img_out
```

```
# 不加眼镜
flag_add_glass = 0

# 读取眼镜图像
img_glass = cv2.imread("glasses.bmp")

while True:
    img = cv2.imread("cat.bmp")   #加载图片

    if flag_add_glass ==0:
        drawline(img,dic_points)

    if flag_add_glass ==1:
        img = add_glass(img,img_glass,dic_points)

    key = cv2.waitKey(1) & 0xFF

    if key == ord('q'):
        break
    if key == ord('a'):
        flag_add_glass = 1
        key =0
    if key == ord('r'):
        flag_add_glass = 0
        key =0

    cv2.imshow("image",img)
```

按 q 退出

按a 加眼镜

按r 去掉眼镜