opencv-python颜色轨迹描绘实验

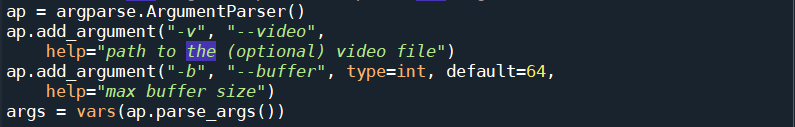
一.预备知识介绍：

①这段代码是在构建命令行参数，命令行参数是一些需要在程序运行时给出的标志，包含了让程序可以顺利执行的一些额外的信息，此处我们使用命令行参数是为了在运行程序处理不同的视频流时通过命令行给它传入一些运行所必需的参数，而不需要到.py文件中去一个一个地改相应的参数

ap=argparse.ArgumentParser()：实例化一个ArgumentParser类的对象，将其命名为ap

ap.add\_argument()：添加那些你需要为其赋值的参数，其中如"-v", "--video"是给这个参数的命名，两个命名可看作是简写/全拼，意义一致，都可以正常使用

args=vars(ap.parse\_args())：解析命令行参数，转化为程序执行时所必需的参数的类型



②cv2.erode()腐蚀：将前景物体变小，理解成将图像断开裂缝变大（在图片上画上黑色印记，印记越来越大）

cv2.dilate()膨胀：将前景物体变大，理解成将图像断开裂缝变小（在图片上画上黑色印记，印记越来越小）

此处先腐蚀再膨胀是去掉毛刺再恢复原状的操作



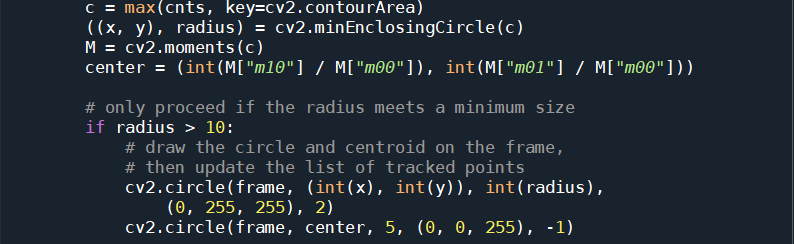
③cv2.findContours()函数可得到图像轮廓，但其接受的参数为二值图，即黑白的（不是灰度图），所以读取的图像要先转成灰度的，再转成二值图



④找到mask中最大的轮廓c：cv2.contourArea是计算轮廓面积

然后计算最小外接圆及其圆心

再用cv2.circle函数将外接圆和圆心都绘制出来



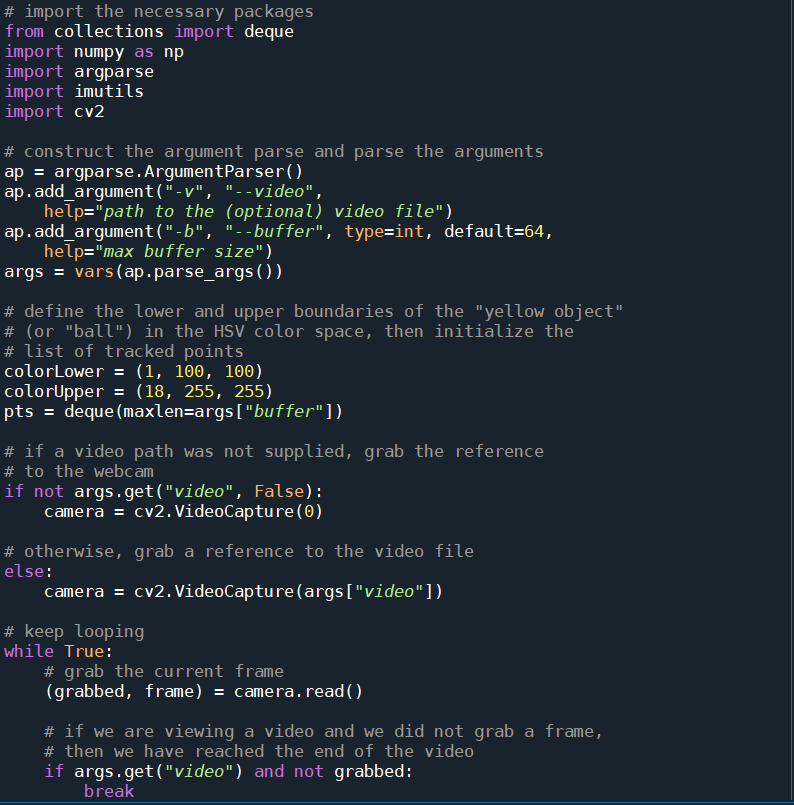
⑤将新的中心点center更新到pts队列中

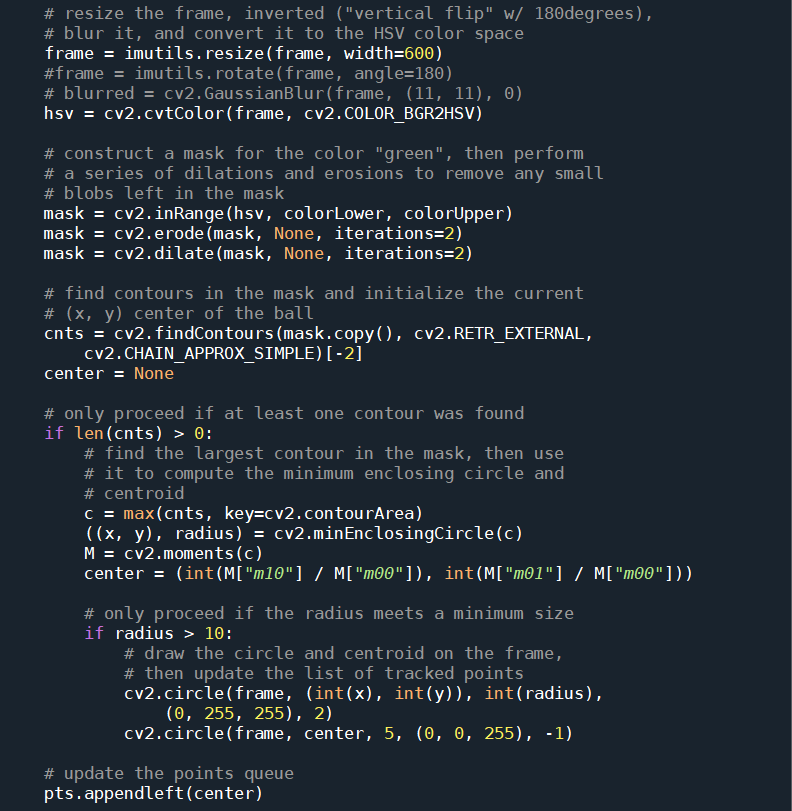


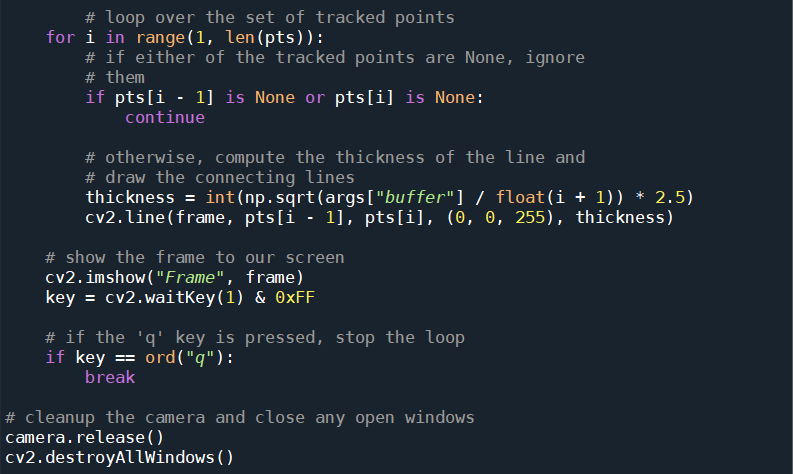
⑥定义宽度并绘制出颜色轨迹（即pts中的点）



二.实验代码(Color\_Tracking文件下color\_tracking.py程序)：





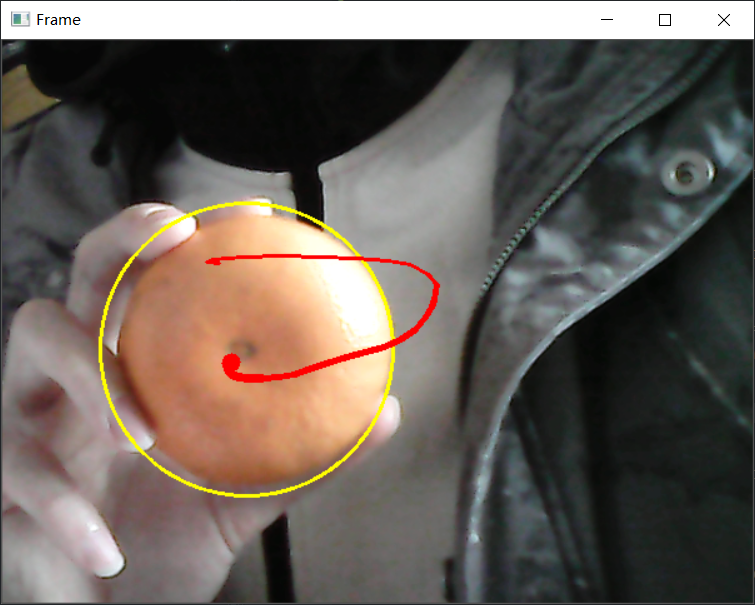


三.实验步骤：

①通过“opencv-python颜色检测实验”中的方法获得所要追踪物体（该物体尽量为球形）的颜色的上下界值，并修改color\_tracking.py中的相应代码



②运行color\_tracking.py，手持要追踪的物体在摄像头前运动，即可描绘该物体的运动轨迹



注：opencv对颜色检测并非特别灵敏，在此实验中，经测试，相同的参数下，红色，橘色，黄色物体都能被识别