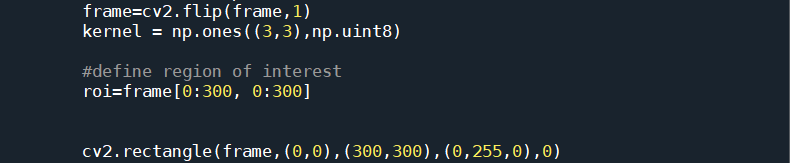
opencv-python手势识别实验

一.预备知识介绍：

①定义了检测区域roi，在[0:300,0:300]的区域内，并通过cv2.rectangle函数将此区域框出



②构建皮肤颜色的上下阈值（皮肤颜色大致在黄色-红色之间）



③arcLength 函数用于计算封闭轮廓的周长或曲线的长度，其第二个参数用于指示曲线是否封闭，这里计算出手轮廓的周长的0.0005倍，将作为其后多边形近似的精度参数

cv2.approxPolyDP(contour,epsilon,True) 则采用Douglas-Peucker算法，用于将轮廓近似为多边形（这里是将手的轮廓稍加近似，方便后续几何处理）

第一个参数是轮廓的点集（此处为手轮廓cnt）

第二个参数epsilon可以认为是近似多边形的精度，精度值越小，得到的多边形越接近原来图像

第三个参数指示新产生的轮廓是否闭合

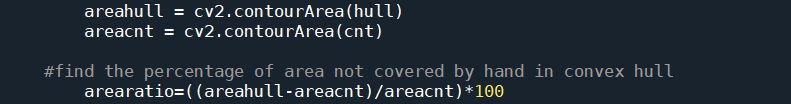
返回的是一些列点组成的多边形



④cv2.convexHull(points, clockwise, returnpoints)函数用于计算几何凸包，所谓几何凸包，即给定二维平面上的点集，将最外层的点连接起来构成的凸多边形，它能包含点集中所有的点



⑤通过面积计算arearatio参数：几何凸包中没有被手覆盖部分的百分比

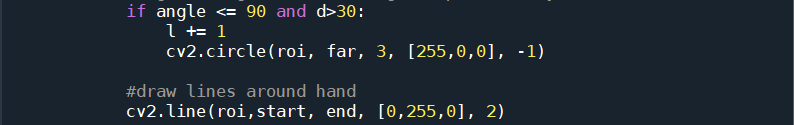


⑥cv2.convexityDefects函数将手的近似轮廓approx和其凸包hull进行比较，得到凸缺陷（此处即指缝凹陷处）

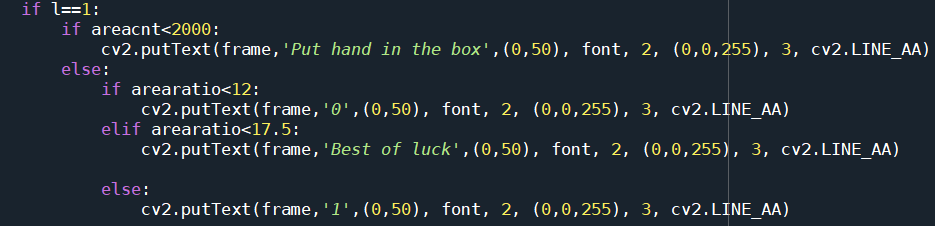


⑦cv2.circle函数绘制出了凸缺陷点（一个凸缺陷点即代表一道指缝）

cv2.line函数则连点成线，绘制出了包裹手的多边形



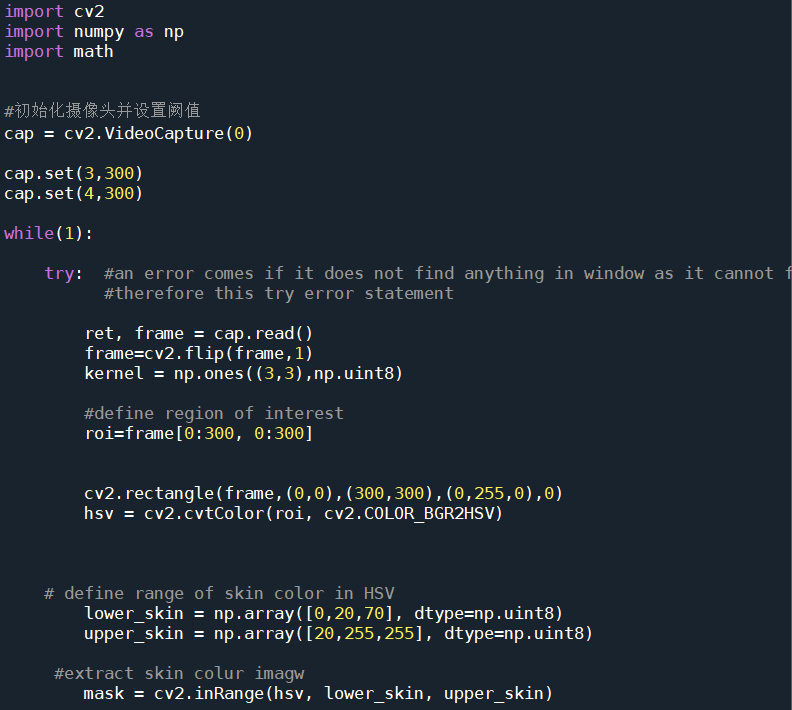
⑧l = 1说明没有检测到凸缺陷点（即没有指缝），则可能是0，1或者Best of luck手势，需要通过arearatio参数（几何凸包中没有被手覆盖部分的百分比）来区分

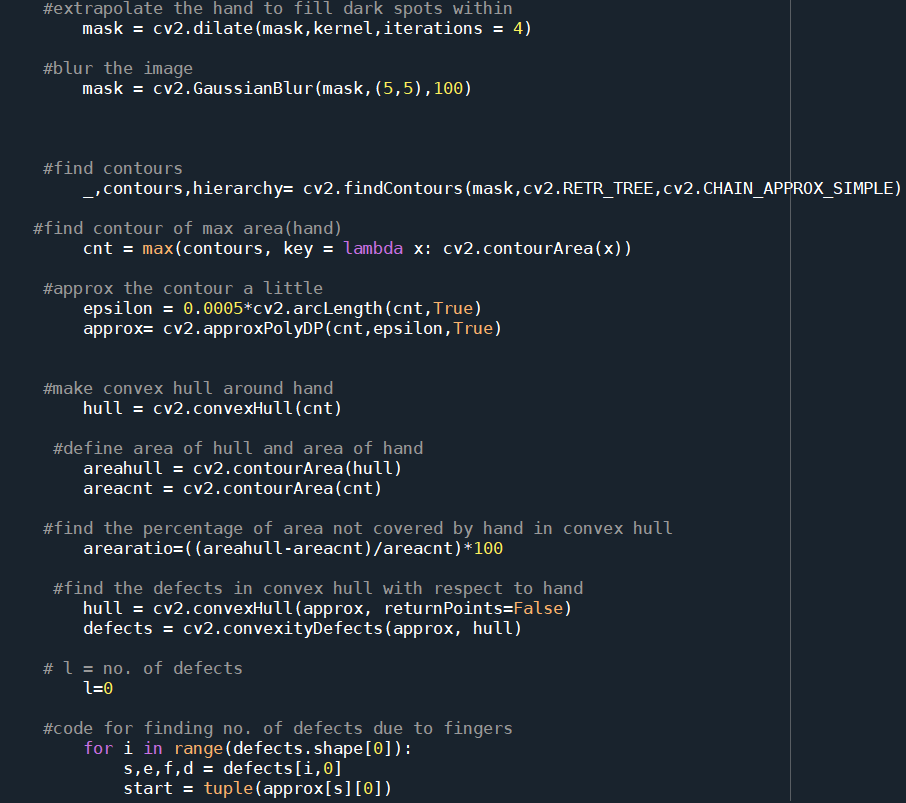


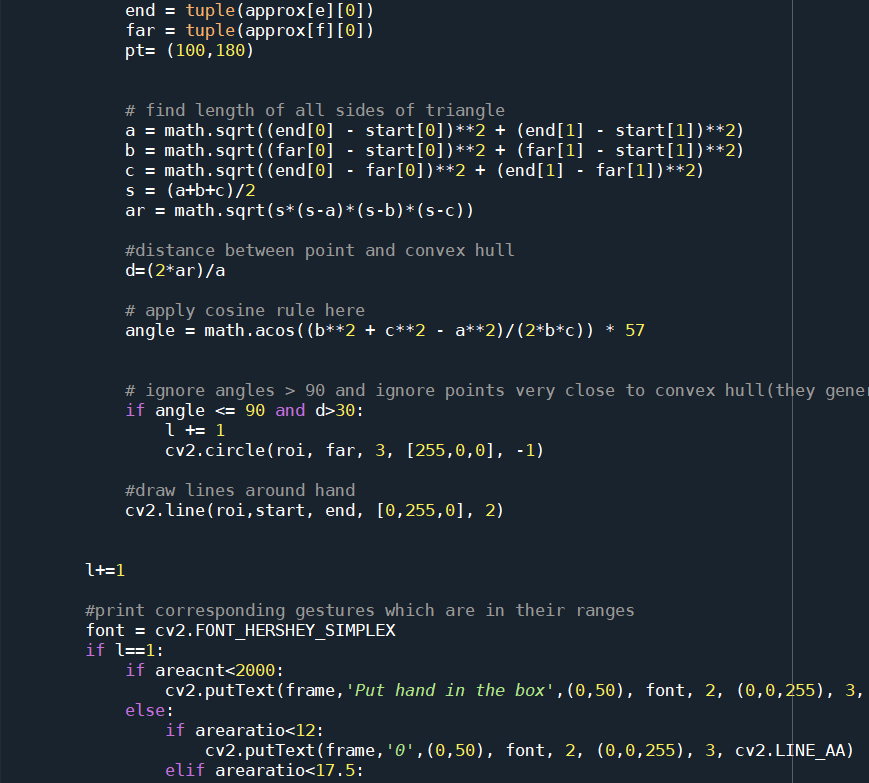
⑨l = 2说明检测到一个凸缺陷点（有一道指缝），只可能是2手势

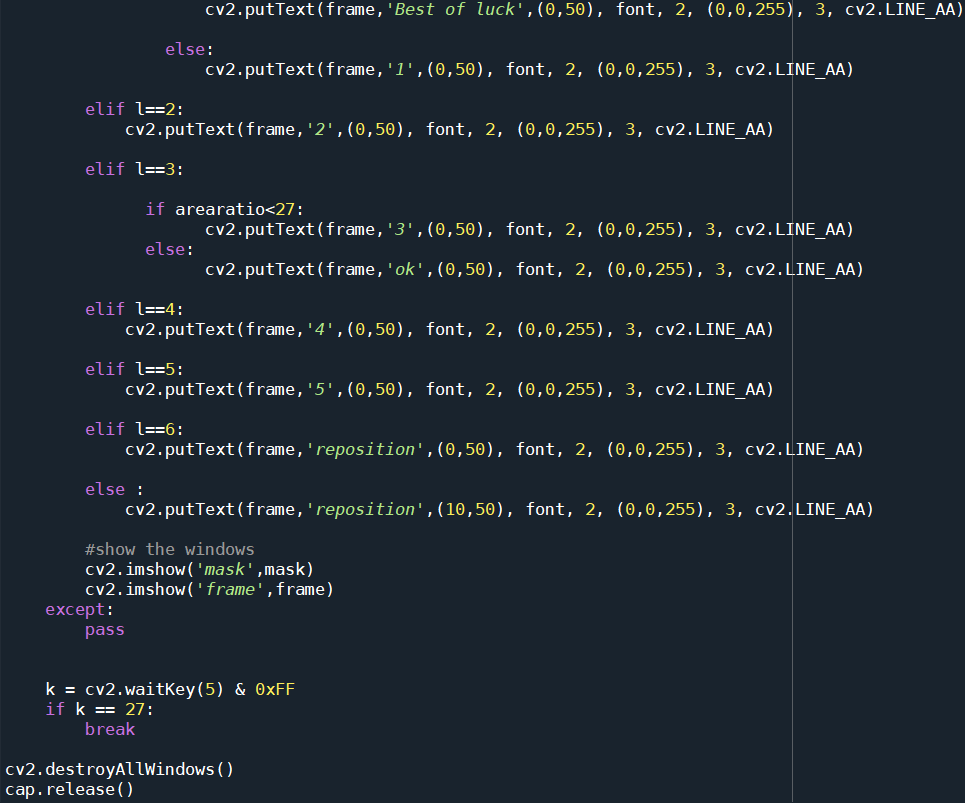


二.实验代码：









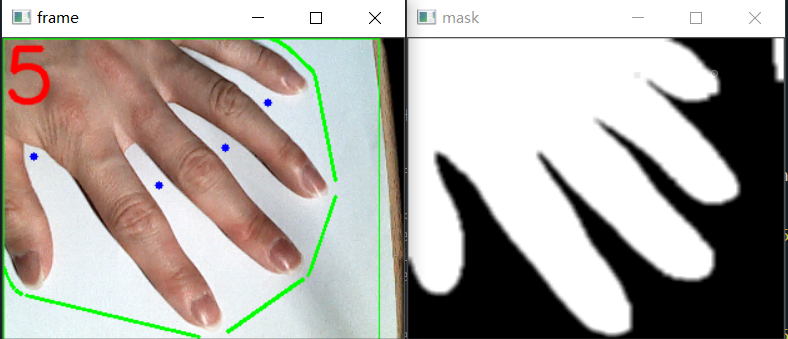
三.实验步骤：

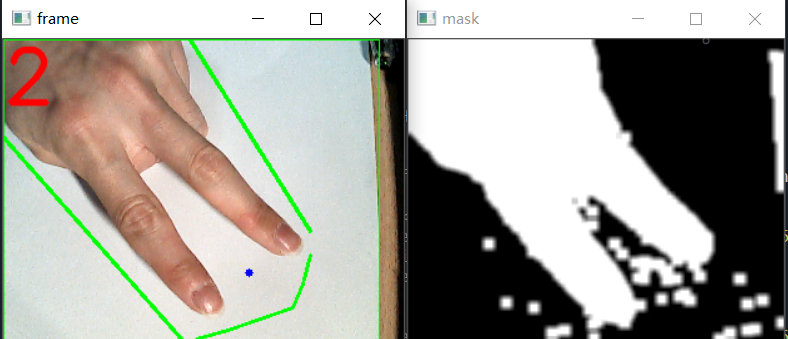
①选择一块白色背景版（白色A4纸即可）

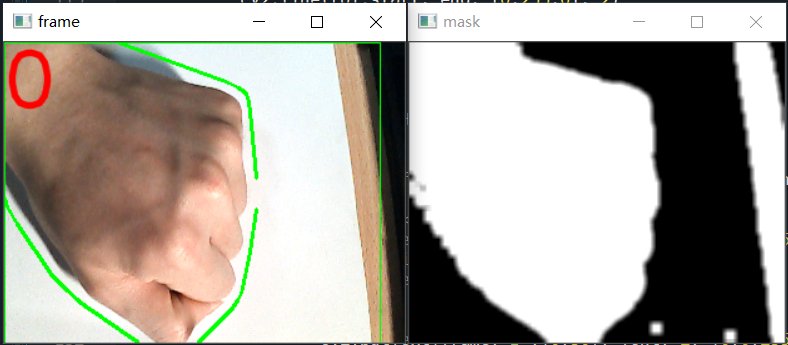
②运行Hand\_gestures文件夹下的Hand\_gestures.py文件，将手放于摄像头下，即可识别手势

支持识别的手势：

1，2，3，4，5及ok，Best of luck手势







注：opencv对手势的识别并不足够准确，不使用白色背景时周围物体很容易干扰手势识别，白色背景下mask中白色的噪点是干扰手势识别的主要因素，可通过调整摄像头位置或调节光源来减少噪点