

静态与动态实时手势识别报告

基于视觉的手势识别按手势的动态特点可以分为静态的与动态的两种，静态手势识别针对单个手型，主要包括手势分割及手势识别两部分，前者是后者的基础。静态手势识别方法难以识别区分度较小的手势、实时性差，因此向动态手势识别过渡已经成为一种趋势。因此在实现对图像中手指个数的识别之后，另外完成了一个实时的识别手指个数的功能。

在静态手势识别中，没有使用复杂的神经网络，而是使用了计算机视觉中的方法，直接对输入图片进行分析：选定区域，在区域内以特定点为圆心，画不同半径的圆，得到圆与手指的交线，然后对交线条数进行判断，有交线的为手指，无交线的为背景。重点在于选定合适的圆心与半径，圆心的选定关系到圆能否与手指有相交线，而半径的选定则关系到判断的准确度。经过多次调整实验之后，圆心选在图片左上角顶点与上边界夹角 60° 的直线，与上边界中点向下延伸的直线的交点，半径大概为图片高度的三分之一。

使用此方法可以正确识别出手指数，而且正确率也较高，图 1—图 5 所示为识别结果。

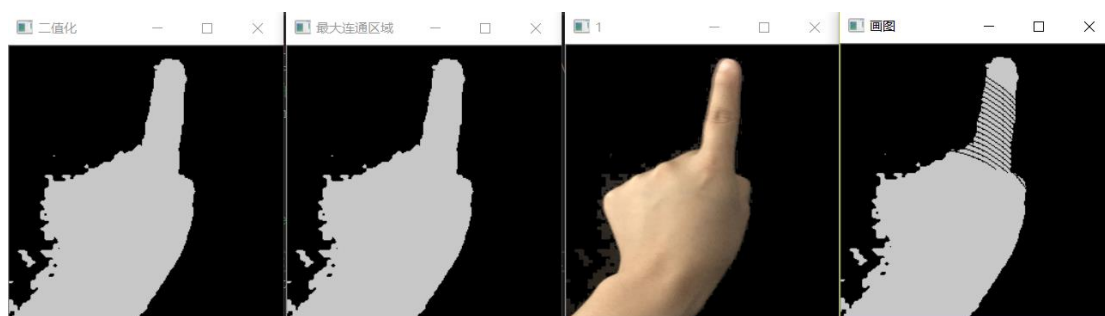


图 1

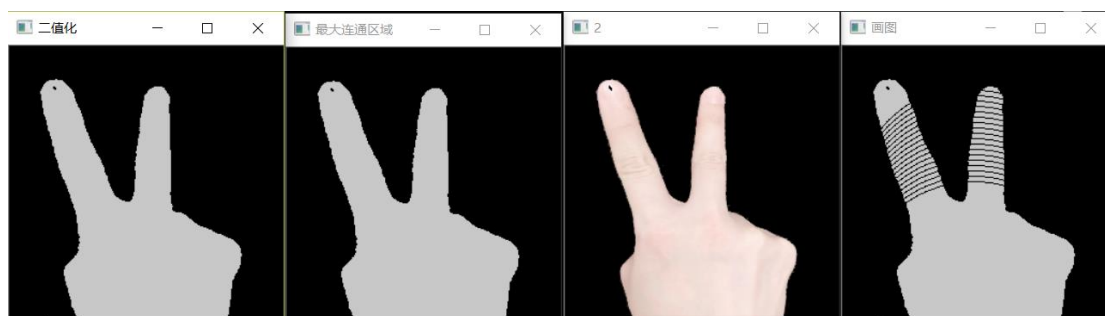


图 2

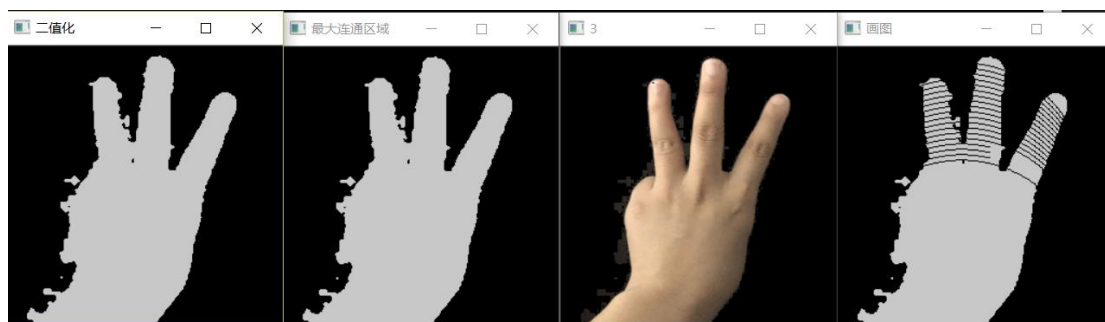


图 3

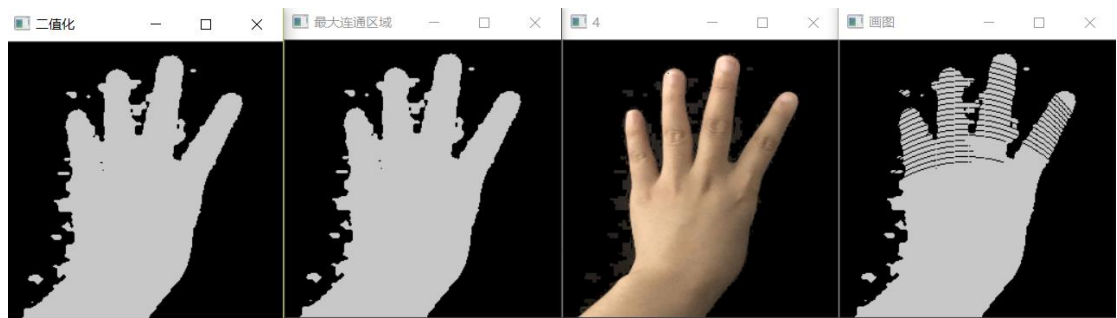


图 4

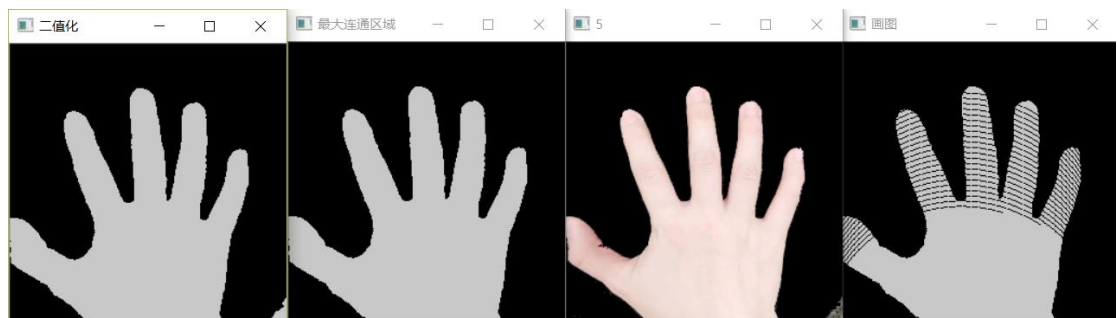


图 5

不过这种方法也存在自己的局限性，就是在输入图片如图 6 所示时，手势数字应该是三，但是识别出来是四，这是因为弯曲的关节也被程序识别为了一根手指。

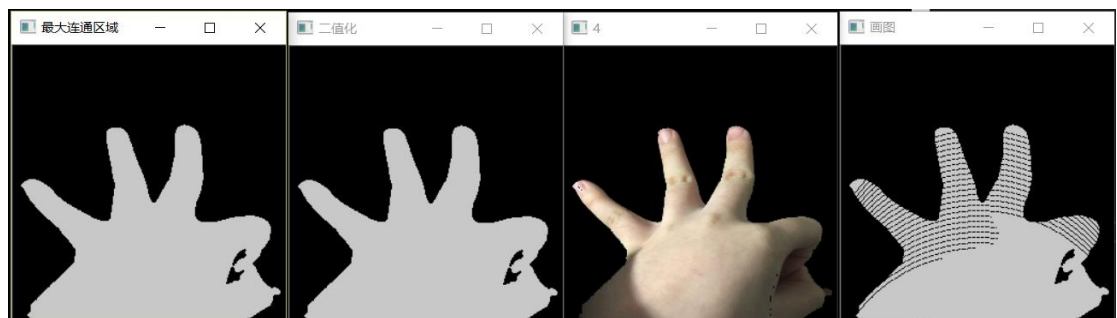


图 6

为了解决上述问题，特地在程序中加入了权值，对该图片中手指数进行赋权值，最后选取权值最大的一个数字作为结果输出。实验结果如图 7，取得了较好的效果。

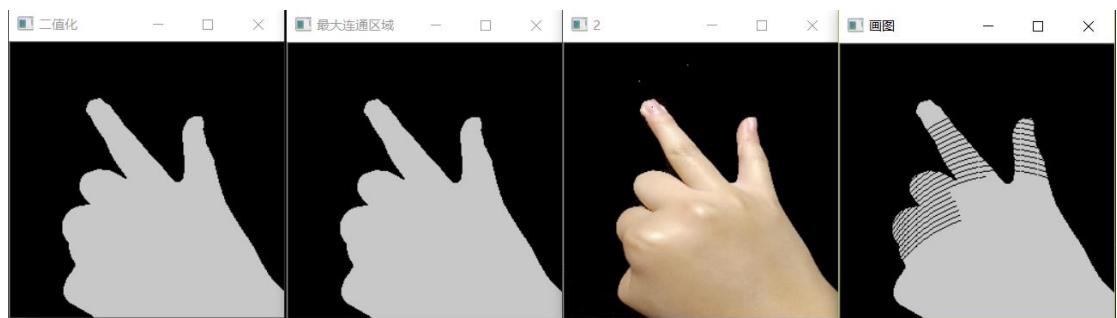


图 7

在完成静态图片手势识别之后，又开始考虑实时的手势识别。实时手势识别是利用摄像头实时采集手势图像，并进行 HSV 颜色空间转换及通道分离，之后进行肤色分离和中值滤波，最后进行轮廓检测及过滤，并且与模板库中的形状相匹配，找到适配的手势图片并输出结果。

实验过程中的重点在于 HSV 颜色空间转换及通道分离、肤色过滤以及轮廓检测及过滤。

HSV 色彩空间 (Hue-色调、Saturation-饱和度、Value-值) 将亮度从色彩中分解出来, 以便更好地感知图像颜色, 利用 HSV 分量从图像中提取感兴趣的区域。而 opencv 中有完整的转换函数: `cvtColor(frame, frame, CV_BGR2HSV)`; 以及 HSV 通道分解的函数: `split(frame, channels)`; `frameH = channels[0]`; 转换示例如图 8 所示。

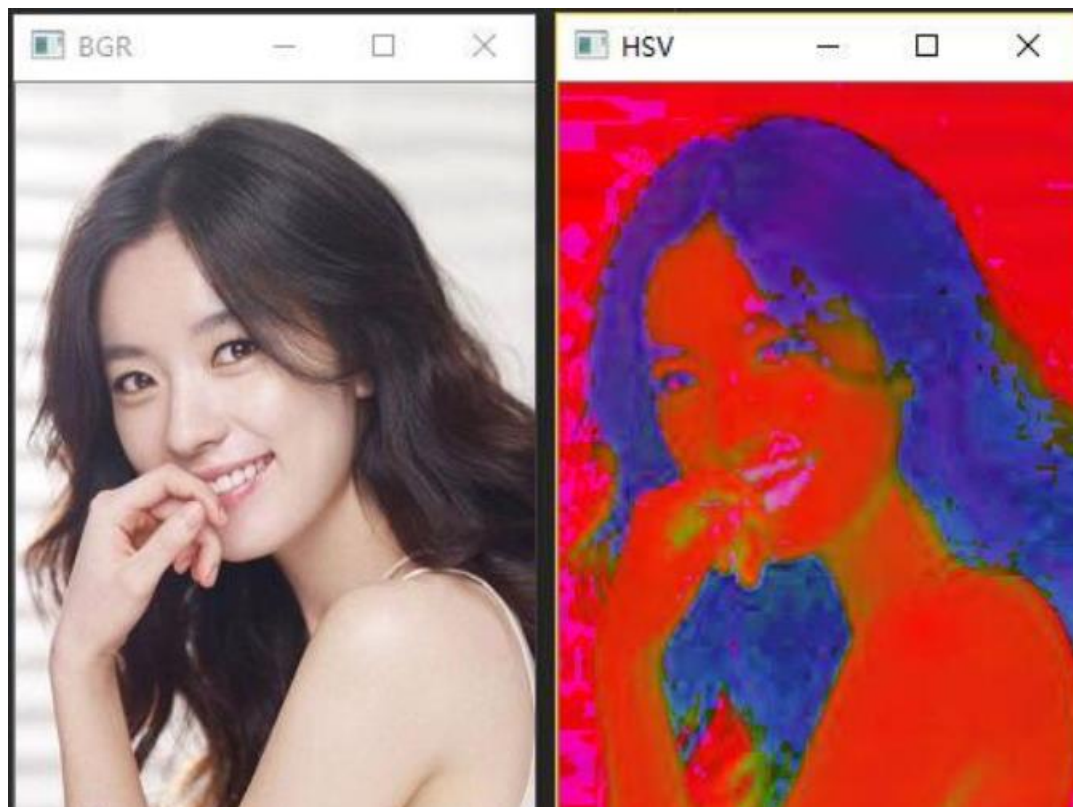


图 8

皮肤检测算法是在 HSV 空间进行的, 可以通过颜色阈值分割肤色部分。Opencv 中有完整的肤色分离函数: `inRange(frameH, Scalar(minVal), Scalar(maxVal), result)`。主要是把皮肤阈值分割和运动检测相结合, 优点是不需要了解算法原理, 直接使用, 缺点是抗噪差, 容易误检, 对同色背景边缘检出不准确。示例如图 9 所示。



图 9

轮廓检测是基于 canny 边缘检测算法之上进行的, 先使用 canny 算法标记出图像边缘, 然后使用 opencv 自带函数: `findContours(srcImage, mContours, mHierarchy, RETR_CCOMP, CHAIN_APPROX_NONE, Point(0, 0))`; 来进行轮廓标记, 然后使用函数: `drawContours(draw, mContours, -1, Scalar(0, 0, 255), 1, 8, mHierarchy)`; 来进行绘制轮廓。之后过滤小面积的轮廓, 以及与窗口贴近或相连的轮廓边界。示例如图 10 所示。

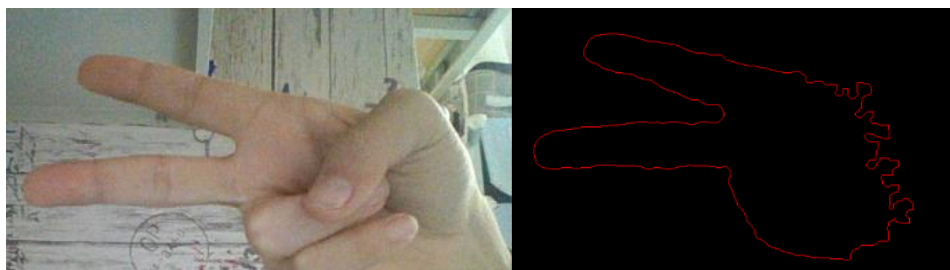
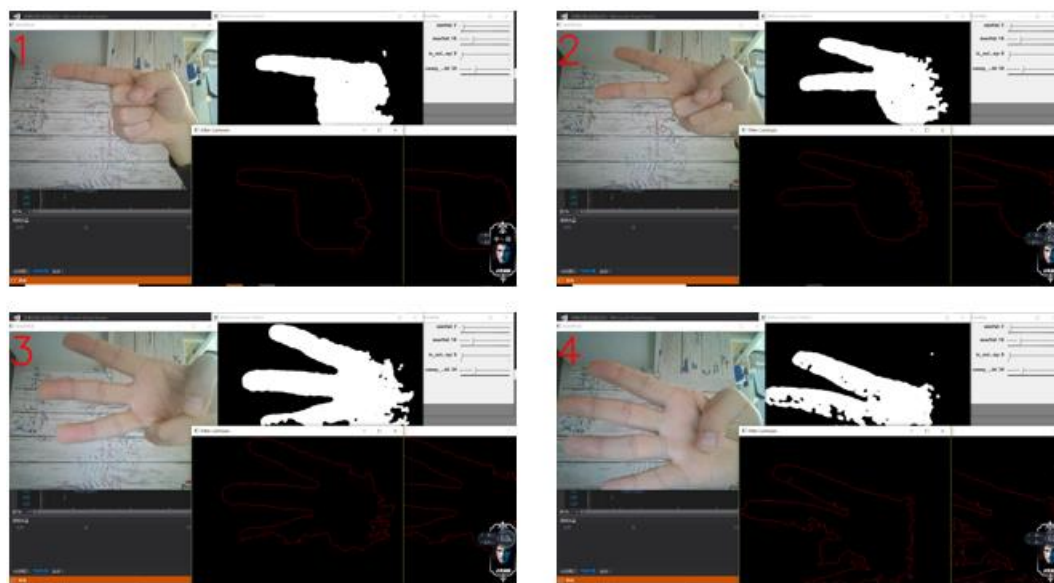
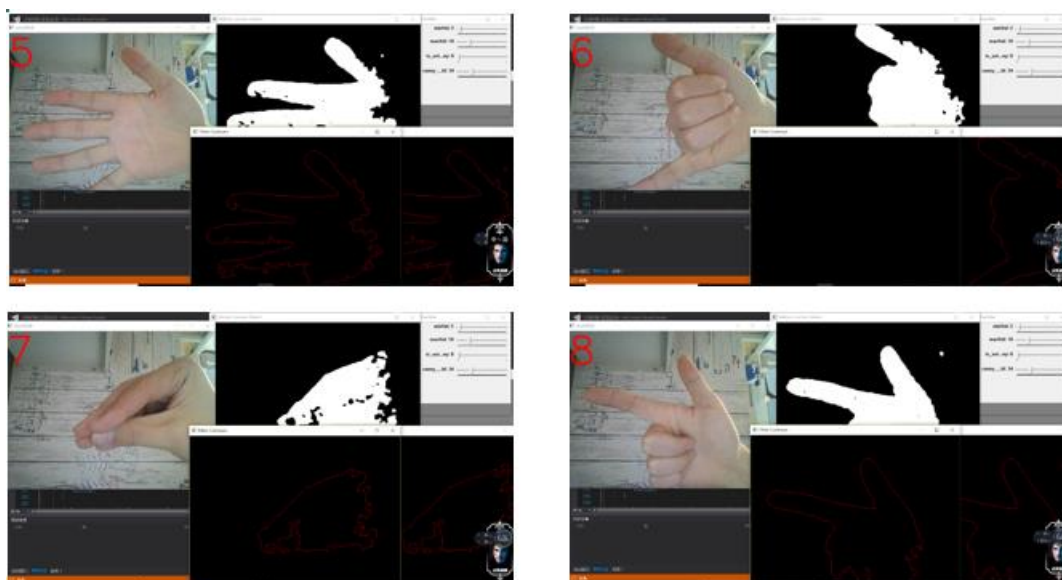


图 10

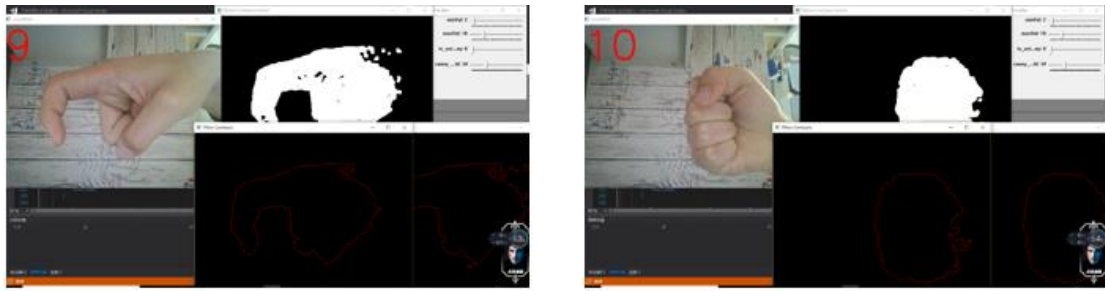
实验结果如图 11-图 13 所示，所用模板库如图 14 所示。



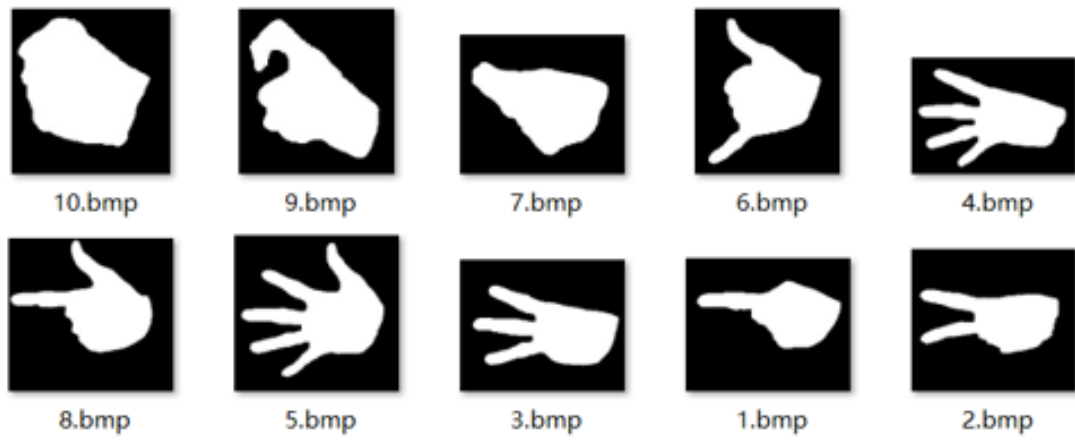
图片 11



图片 12



图片 13



图片 14

此外，考虑到图片的旋转和翻转，也为了让方法具有更高的准确度，将实验数据集中的图片经过旋转和翻转后全部进行训练，再随机抽取不同的图片进行测试，得到测试结果基本全部正确。