

SE-342

Final Report

分工

姓名	分工
马包威	图像采集、初步处理
王绍宇	CV 降噪处理、分块、优化
于喜千	GUI、结果标注
袁卓	SIFT 特征提取算法

工作总结

数据采集部分

拍摄工具

拍摄工具选用了单反相机，相比手机可以更好地记录细节。

拍摄环境

对于拍摄环境的选择，尝试并比较了在不同环境下拍摄图片的效果。

首先尝试在寝室内利用台灯光线照明，发现由于环境光线较暗、照明光线来源集中，会造成明显的阴影；

此外还尝试了在宿舍楼道内拍摄，发现光线的方向偏于一侧，也会造成比较明显的阴影。

故而决定在自然光下进行拍摄。

拍摄背景

为了尽量降低分块的难度和后续识别的复杂度，利用一张 A4 纸作为普通情形下的拍摄背景。

拍摄的结果如下图所示。



初步处理部分

在交由 CV 进行出去之前，先利用 Adobe Photoshop 软件对拍摄图像进行了处理，并对图片的亮度和高光阴影进行了微调。

在处理过程中有以下发现：

- 对于黄色瓶盖而言，若图片整体亮度较高（瓶盖颜色与背景颜色相近），在图片分割时会发现该瓶盖无法识别的情况。故而应对该图片进行降低亮度的操作。
- 对于两个瓶盖相连的情况，在图片分割时会把两个瓶盖分割到同一个小区域内，在后续识别时会发生错误。因此将此种情况从测试图像中排除。

分块裁剪部分

算法选择

首先，在网上查找资料，学习了 OpenCV 库中几个函数如 `threshold`、`findContours` 等函数的用法。

最初，试图通过边缘检测的方法识别瓶盖，但是同一个瓶盖很可能被识别两次，并且会识别到很多并不属于瓶盖的特征。

在注意到测试样例的背景颜色与瓶盖差异很大以后，选择用二值图识别连同区域的方法来识别瓶盖。

进一步处理

利用 `findContour` 找到的边界数组里有很多并不是瓶盖的区域。

经过观察发现这些区域面积普遍非常小，于是添加了一个通过面积阈值来筛选目标的判断语句，最终得到了几乎总是正确的结果。

特征提取部分

裁剪

为了提取一致的特征，首先对图片去除背景和裁剪，得到的被测瓶盖图像，如下图所示：



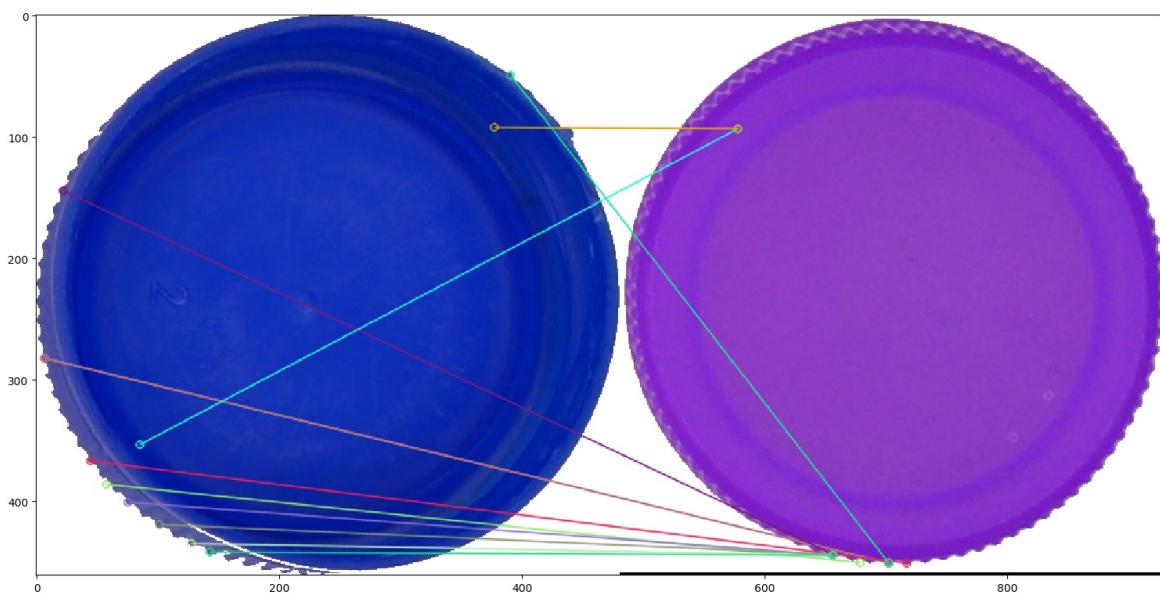
以上处理类似于分割，找到物体中面积大于 1500 的部分用 drawContours 画出图像的轮廓从而去除背景，并依据离中心点最远坐标裁剪无关的边角。

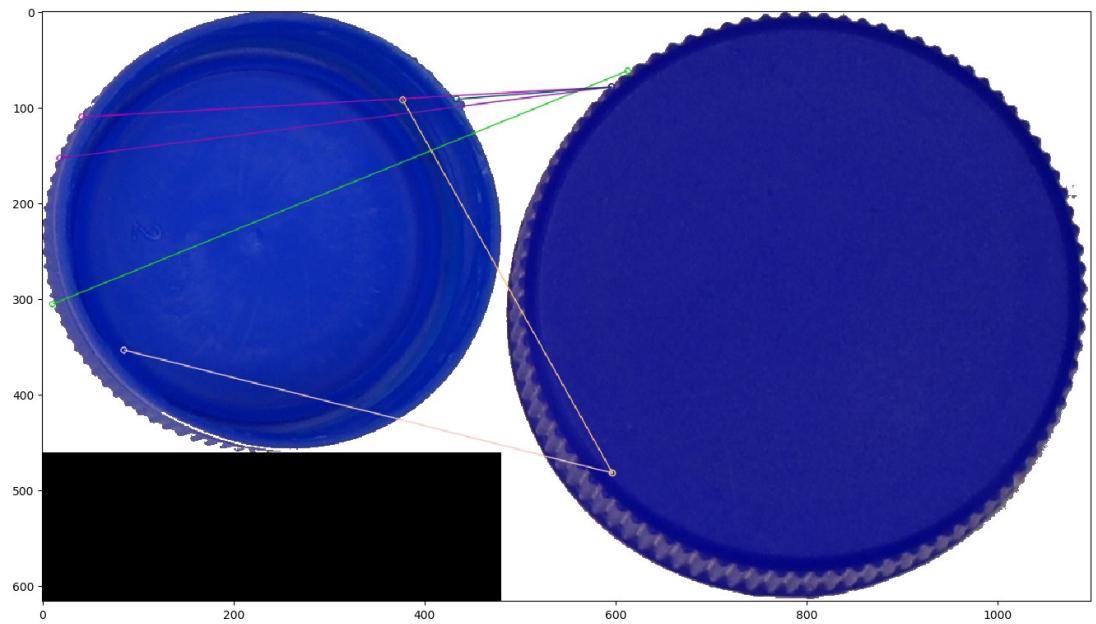
另外，发现立着的瓶盖在裁剪后长宽比超过 1.3 以上，而正放或反放的瓶盖在裁剪后长款比接近 1 : 1，所以根据这个特征已经可以得出立着的瓶盖。

匹配结果分析

考虑到瓶盖正放和反放的关键区别在瓶盖边缘的棱角，所以用SIFT算法对被测瓶盖和不同形状标准正放做特征匹配，对匹配结果做分析：

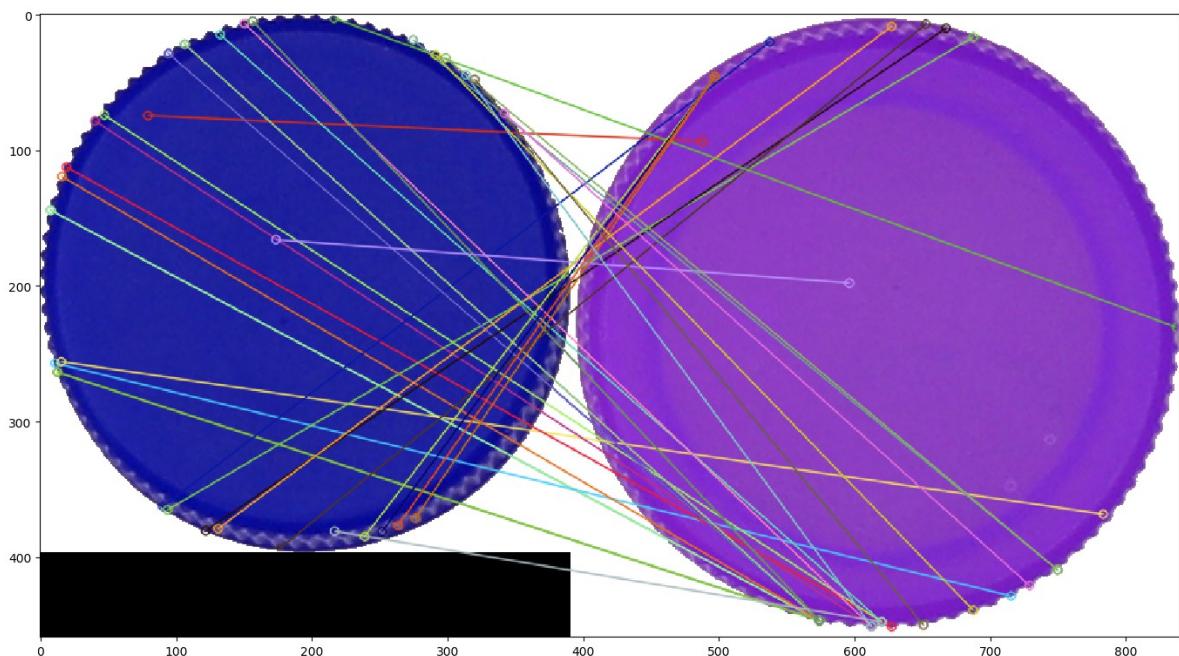
1. 反向放置的瓶盖有两种匹配情况，如下图所示：





这表明，一种是反向瓶盖和正放瓶盖匹配点极少，另一种是匹配点集中分布在一侧；

2. 正向放置瓶盖只有一种匹配情况，如下图所示：



这表明正向瓶盖与标准瓶盖匹配点多，且均匀地沿瓶盖边缘分布。

整体流程

根据以上分析，最终确定了流程：

1. 裁剪图片，根据长宽比判断立着的瓶盖；
2. 对裁剪得到的图片与多张标准图片做特征匹配，选取匹配最大的情况；
3. 判断匹配点的个数和分布情况，如果集中分布在一侧或数量很少则判断为反向瓶盖，否则为正向。

结果标注部分

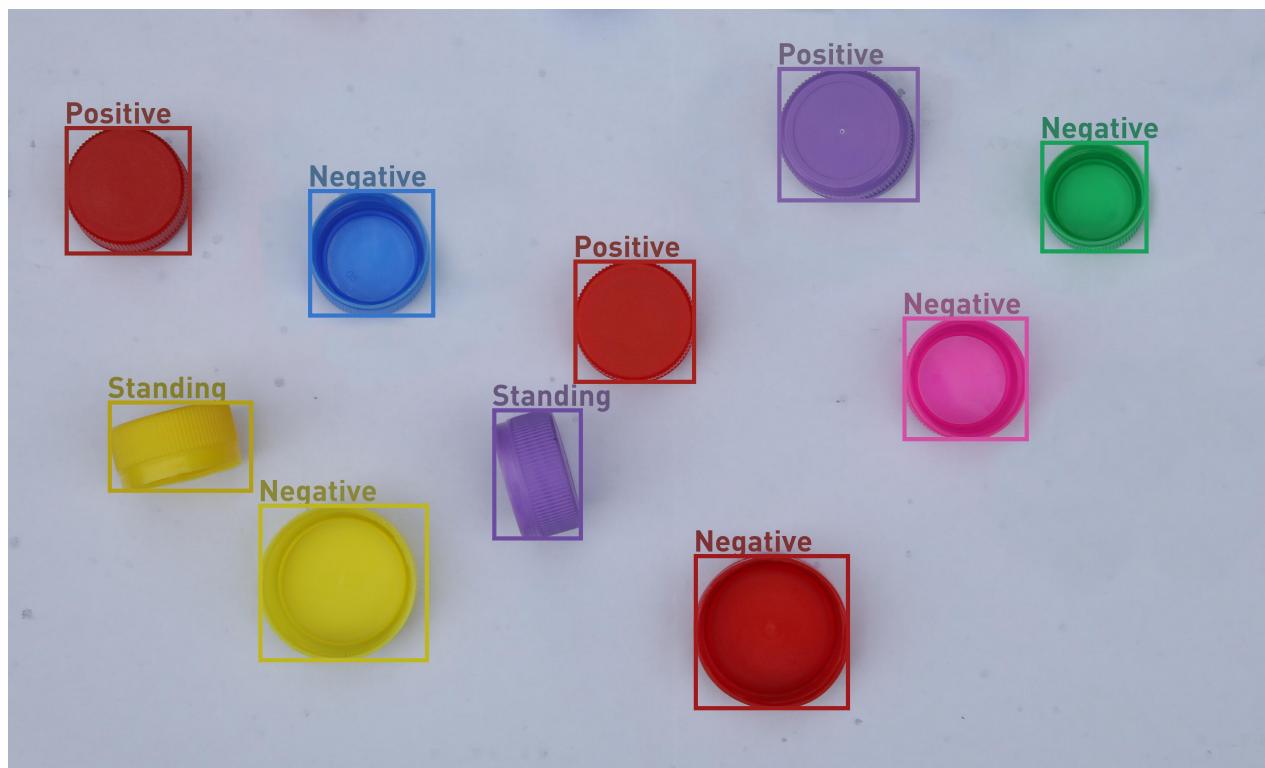
特征提取返回的结果是一个外切包围瓶盖物体的四元组 `BoundRect`、一个表示瓶盖正反立状态的 `Tag`、和一个表示瓶盖颜色的三元组 `Color`。

采用 `Pillow` 库提供的 `ImageDraw` 子库中提供的 `drawRect` 和 `drawText` 方法即可对原图像进行标注。

为了保证对不同尺寸的图像都能给出相对合适的标注结果，采用 $\text{sqrt}(\text{width} * \text{height}) / 100$ 作为框线的厚度、采用 $\text{sqrt}(\text{width} * \text{height}) / 33$ 作为文本的磅数。

另外，为了表现不同瓶盖的差异，采取将粗匣框线和文本上色的策略来直观表示颜色差异。

标注的结果如下图所示。



GUI 部分

GUI 采用 Python 3 结合 PyQt5 开发，提供了良好的跨平台能力。

为了方便观察和调试，提供了对图像进行缩放、调节算法的阈值、开启或关闭颜色区分等功能。

同时，还支持一次开启多张图片并在其间进行切换。

GUI

图形界面

