课程名称：计算机图形图像技术试验

实验名称：图像分割——车牌识别

**姓 名： 应一凡**

**专 业： 智能科学与技术**

**学 号： 21312274**

**实验一 图像分割——车牌识别**

**一、实验内容及目的**

完成小车的车牌定位与识别

1. **实验相关原理描述**
2. 车牌定位
3. 边缘检测（可调用 edge()，method 自行选择，然后反色成白底黑边）
4. 开闭运算（对边缘检测图先开后闭）

图像形态学处理的一种手段，图像膨胀会扩大一幅图像的组成部分，而 图像腐蚀会缩小一幅图像的组成部分。

开操作：先腐蚀，后膨胀，效果为平滑物体轮廓，断开较窄的窄颈并消 除细的突出物

闭操作：先膨胀后腐蚀，有助于消除物体内部小黑点 用法及函数效果在实验所用函数处展示 完成后需要对过小的连通域进行删除（噪声）

1. 矩形化 对于连通区域进行矩形化，方便接下来选择车牌区域 从经过数学形态学运算得到的连通区域的图中可以看出，这些连通区域 是不规则的，它们不利于后续的处理 （从候选区域中提取车牌区域）。所 以，要使这些候选区域矩形化，即扩大这些候选区域的边缘，使其成为矩形 区域。 其具体步骤如 下：①若 Ｇ（ｉ，ｊ）为黑点，而且它的 4－邻 域中有至少 任意两像素为白点，则令其为白 点；②若 Ｇ（ｉ，ｊ） 为白点，而且它 的４－邻域中至少 2 个像素为黑点，则令其为黑点。 该步骤要从（0，0）（m,0）（0,n）（m,n）四个起始点对图像做迭代循 环，不用建新图，每次循环都是从上一次更新的图往下做
2. 选择区域 对上列矩形区域进行筛选，对于车牌，我们利用如下约束进行筛选： （1）矩形长宽比 （2）区域蓝色像素点所占比例（针对国内车牌为蓝底） （3）候选区域垂直投影函数与其 y=平均值的这条直线交点的个数要在 一定阈值内，count 在 15 到 45 之间。 其中条件 2，3 比较重要，3 用于排除颜色正确但样式不对的候选区域， 3 排除蓝色以外的干扰项，垂直投影函数可调用 Verticalprojection.m 函数
3. 图像分割 对图片进行二值化，突出车牌数字，分别尝试下列两种方法并分析 1）基于全局的阈值分割（大津法） 详见课本 P114 或附录 2）基于局部的阈值分割（bernsen） 详见课本 p115 或附录
4. 字符分离 分割字符，利用栅栏法（即寻找字符间的空隙）分离字符，这里我们可以利 用 y 方向上像素和为 0 来寻找分离点，也可以跳过此步做下一步识别
5. 车牌识别 本节自由发挥，可以使用第一次所做的模板匹配法，也可以用 python 机器学习来做
6. **实验过程**

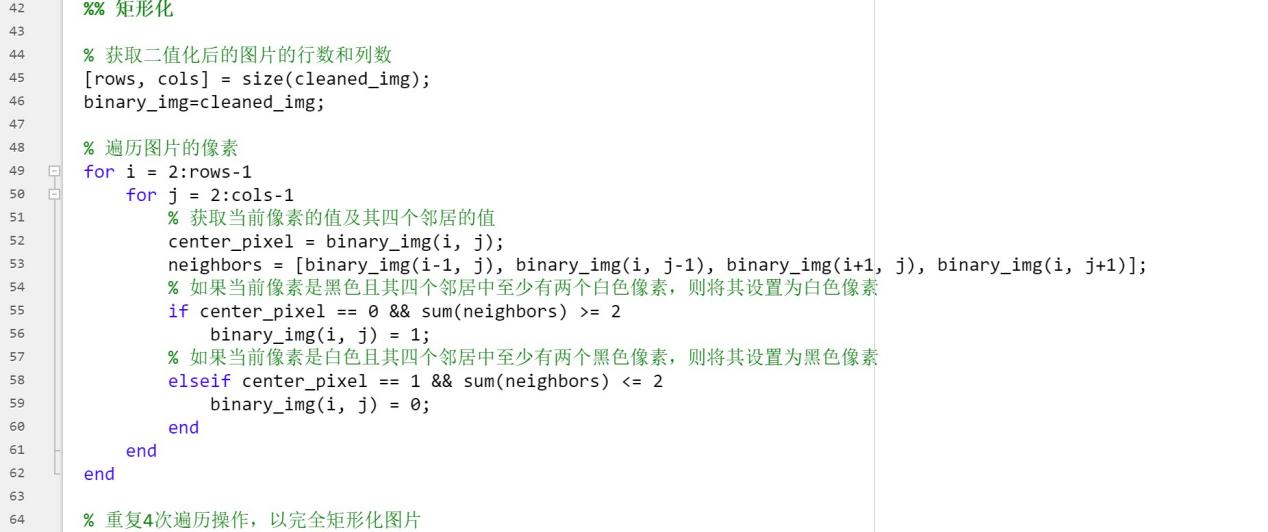
3.1 车牌定位



首先，使用Canny边缘检测算法对灰度车辆图像进行边缘检测，得到一个二值图像。然后，对边缘图像进行取反处理，使边缘区域变为黑色，背景变为白色。

接下来，构造两个矩形模板，分别用于开运算和闭运算操作。开运算通过腐蚀和膨胀操作来平滑边缘，去除小的细节。闭运算则通过先膨胀再腐蚀的操作来闭合边缘，填充空隙。然后，对取反后的边缘图像分别进行开运算和闭运算操作，得到经过处理后的图像。随后，再次对闭运算后的图像进行取反处理，使车牌区域变为白色，背景变为黑色。最后，使用连通成分分析的方法去除小的连通区域，即去除噪点，保留较大的区域，这里设置了一个最小面积阈值。

3.2 矩形化



此处只列出四分之一代码，其余部分原理一致。

这段代码用于将二值化图像进行矩形化处理。首先，获取二值化图像的行数和列数。然后，通过遍历图像的每个像素，对每个像素及其四个邻居的像素进行判断和调整。对于每个像素，如果其为黑色且其四个邻居中至少有两个是白色像素，则将其设置为白色像素；如果其为白色且其四个邻居中至少有两个是黑色像素，则将其设置为黑色像素。这样的处理过程会使图像中接近矩形形状的区域更加突出，其他区域可能会发生调整。通过循环遍历所有像素，实现了图像的矩形化处理。

3.3 区域选择



首先，将车辆图像转换为HSV格式，并进行连通性处理。然后，使用4连通算法对二值化图像进行标记，得到每个候选区域的标签以及总候选区域数量。接下来，通过 regionprops 函数获取每个候选区域的边界框信息。使用循环遍历每个候选区域，并根据矩形的长宽比例要求来提取矩形区域。对于满足长宽比例要求的候选区域，首先提取该区域在HSV图像中的像素信息。然后，计算该区域中蓝色像素点的数量，并计算蓝色像素点的占比。如果蓝色像素点的占比超过等于50%，则将该候选区域在原始车辆图像中提取出来，并显示处理后的候选区域图像。

3.4 图像分割与字符保存



这段代码用于图像分割预处理，旨在进一步提取车牌信息。首先，将候选区域图像转换为灰度图像。然后，使用Otsu's方法确定合适的阈值进行二值化处理，将灰度图像转换为二值图像。接下来，对二值化后的图像进行连通性处理和面积过滤。通过定义一个线性结构元素 se，使用闭运算操作 imclose 对图像进行闭合，以填充图像中的空洞。然后，使用 bwareaopen 函数对图像进行面积过滤，保留面积大于等于20的连通区域。最后，在图像的子图中显示二值化后的图像和经过面积过滤后的图像，以便观察图像分割预处理的效果。



这段代码用于获取车牌图像中每个字符的位置信息，并将每个字符提取出来保存为单独的图片。通过使用 regionprops 函数，可以获取处理后的二值图像中每个连通区域的边界框和质心信息。接下来，通过遍历 stats 结构中的每个元素，对于第二到第四个元素，将它们竖直合并成一个图像，这是为了避免不连通的汉字内部被分割。对于其他元素，将每个字符/数字的位置信息提取出来，根据边界框的位置将其在二值图像中切割出来。然后，将切割出来的字符/数字图像进行大小调整，使其统一为相同的尺寸。之后，对图像进行反转处理，将黑白颜色翻转。最后，将每个字符/数字的图像保存为单独的文件，并存入名为 "test" 的文件夹中。

3.5 模板的处理与保存



此代码为对数字图片的预处理，对英文、汉字图片的预处理原理一致，故只放此部分。首先，通过使用 dir 函数获取数字文件夹中图片。接下来，使用 length 函数获取文件列表的长度。然后，通过循环遍历文件列表中的每个文件。对于每个文件，首先获取文件的完整路径。然后，使用 imread 函数读取图像文件并将其转换为灰度图像。接下来，设定一个阈值，用于二值化灰度图像。进一步，通过 imbinarize 函数对灰度图像进行二值化处理，得到二值图像。通过寻找二值图像中黑色像素的坐标，确定黑色字母所在的区域，以进行裁剪操作，从而提高识别的准确率。根据裁剪得到的区域，将其调整为统一的大小（40x30）。最后，将处理后的模板图片保存到指定的输出文件夹中，命名为数字对应的序号。

3.6 字符识别



此代码为对数字/字符识别，对汉字图片的识别原理一致，故只放此部分。首先，通过使用一个循环变量 i 来遍历测试图片。每次迭代，读取当前测试图片并进行二值化处理，将其转换为黑白图像以便于后续的模板匹配。然后初始化一个误差变量为一个较大的值，并开始匹配模板。在匹配模板的过程中，循环遍历数字和英文字母的模板图片，计算测试图片和模板之间的误差。通过比较误差值，选择最小误差的模板，并记录下相应的数字或字母的索引。最后，在图像上显示识别结果，使用 subplot 函数将测试图片和识别结果显示在一起，并设置标题为识别的数字或字母。

**四、实验结果**

4.1 识别结果



由于在切割过程中做了一些预处理，所以观察到正确率较高，事实上模板过少的情况下正确率会偏低

4.2 车牌定位过程中的处理

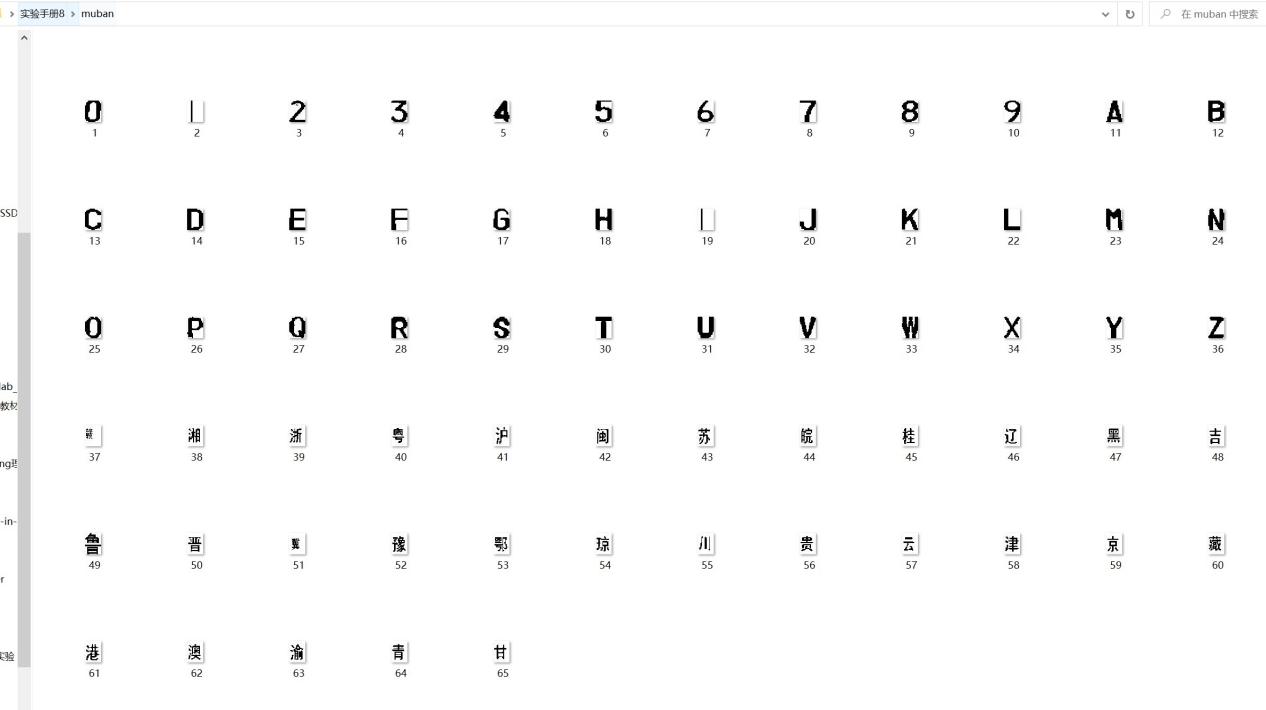


可以观察到开闭操作、矩形化操作都符合预期

4.3 候选区域处理，用于分割



4.4 文件操作结果





上图为处理完的模板图片的保存，下图为分割后的字符的保存，均符合预期

**五、总结**

在这个实验中，实现了车牌定位和字符识别的算法。通过实验，我对图像处理和模式识别领域的一些基本概念和技术有了更深入的了解。

车牌定位阶段是通过一系列图像处理操作，如边缘检测、形态学运算和连通性分析，来提取出车牌区域。通过调整参数和运算操作，可以对不同类型的车牌进行有效的定位。

字符识别阶段是针对提取出的车牌区域，进行字符的分割和识别。利用二值化、连通性分析和模板匹配等技术，对字符进行准确的分割和识别。这个过程中，需要根据字符的特点进行不同操作，其准确性很大程度上取决于模板的质量。

通过完成这个实验，我学到了许多关于图像处理和模式识别的基本原理和常用技术。认识到图像处理在实际应用中的重要性，例如在车牌识别、人脸识别、物体检测等领域的广泛应用。同时，我也意识到图像处理算法的设计和调优对结果的影响巨大，需要细致的参数调整和算法优化才能获得满意的识别效果。此外，这个实验还让我更加熟悉了MATLAB编程环境和图像处理工具箱的使用。通过编写和调试代码，我提升了自己的编程能力和问题解决能力。