中山大學

图像理论作业4

题　　目：	图像分割-车牌识别
上课时间：	第13-15周
授课教师：	王萍
姓　　名：	周德峰
学　　号：	21312210
日　　期：	2023-6-2

实验内容及目的

完成小车的蓝色车牌定位与识别

实验原理

1　图像开运算（**Opening**）

一种基本的形态学图像处理操作，常用于去除图像中的噪声、平滑边缘并保持图像的整体形状。它由两个基本的形态学操作组成：腐蚀（Erosion）和膨胀（Dilation）。

1. 腐蚀（Erosion）：腐蚀是一种将图像中的对象边界腐蚀掉的操作。它通过将一个结构元素（通常是一个小的矩形或圆形区域）与图像进行逐像素的比较，如果结构元素完全包含在图像中的某个位置，则该位置的像素值保持不变，否则将其置为0。腐蚀操作可以将对象的边界向内部收缩，并消除小的物体或细小的图像特征。
2. 膨胀（Dilation）：膨胀是一种将图像中的对象边界扩展的操作。它与腐蚀相反，通过将结构元素与图像进行逐像素的比较，如果结构元素与图像中的某个位置有重叠，则该位置的像素值保持不变，否则将其置为最大像素值（通常为255）。膨胀操作可以填充对象内部的空洞，并增加图像中的物体大小。
3. 开运算（Opening）：开运算是将腐蚀和膨胀操作按照特定顺序进行组合的操作。首先，对输入图像进行腐蚀操作，然后再对腐蚀结果进行膨胀操作。这个过程可以消除图像中的小的噪声点、细小的边缘以及细节，同时保持图像的整体形状和结构。开运算可以在不明显改变物体大小和形状的情况下平滑图像，并提取出物体的整体特征。

2　图像闭运算

常用于填充图像中的小孔、平滑物体边界并保持物体的整体形状。

同样，闭运算也由腐蚀和膨胀组成，但是开运算是先腐蚀后膨胀，而闭运算是先膨胀后腐蚀

3　基于全局的阈值分割

该方法基于一个全局阈值，将图像中像素的灰度值与该阈值进行比较，从而确定像素属于目标物体还是背景。

1. 灰度化：首先，将彩色图像转换为灰度图像。这可以通过将彩色图像的RGB通道进行加权平均来实现，也可以通过将彩色图像转换为其他灰度表示（如YUV或HSV）并选择相应通道来实现。灰度化是为了方便后续的阈值处理，将图像转换为单通道的灰度级表示。
2. 直方图计算：对灰度图像进行直方图计算。直方图是一种统计图，表示图像中每个灰度级的像素数量。通过计算图像的直方图，我们可以了解图像中灰度级的分布情况。
3. 阈值选择：根据图像的直方图，选择一个合适的全局阈值。阈值可以手动选择，也可以通过自动阈值选择算法（如大津算法）来确定。阈值的选择会影响分割的准确性和效果，因此需要根据具体情况进行调整。

4. 分割处理：将图像中的每个像素的灰度值与选定的阈值进行比较。如果像素的灰度值大于阈值，则将其标记为目标物体，否则标记为背景。根据阈值的选择和比较结果，可以得到一个二值化的图像，其中目标物体的像素值为前景（通常为255），背景的像素值为背景（通常为0）。
5. 后处理（可选）：根据具体需求，可以对分割结果进行后处理。例如，可以应用形态学操作（如腐蚀和膨胀）来平滑和修复分割边界，去除小的噪声点等。

4 基于局部的阈值分割

与基于全局阈值分割不同，它在**不同区域内使用不同的阈值**，以适应图像中的局部变化和不均匀性。这种方法能够有效处理光照不均匀、纹理复杂等情况下的图像分割问题。

1. 灰度化：将彩色图像转换为灰度图像，以便进行后续的阈值处理。可以使用RGB通道加权平均或其他灰度表示方法将彩色图像转换为灰度级表示。
2. 图像块划分：将图像分成若干个大小相等的局部块。块的大小可以根据实际需求进行选择，通常选取的大小为16x16或32x32像素。
3. 块内阈值计算：对每个局部块内的像素计算一个适应性阈值。可以使用不同的自适应阈值计算方法，如局部平均、局部中值、局部高斯等。这些方法根据块内像素的灰度统计特征来计算阈值，以适应局部区域的变化。
4. 分割处理：对每个块内的像素，将其灰度值与对应的局部阈值进行比较。如果像素的灰度值大于阈值，则将其标记为目标物体，否则标记为背景。根据阈值的选择和比较结果，可以得到一个二值化的图像，其中目标物体的像素值为前景（通常为255），背景的像素值为背景（通常为0）。
5. 后处理（可选）：根据具体需求，可以对分割结果进行后处理。例如，可以应用形态学操作（如腐蚀和膨胀）来平滑和修复分割边界，去除小的噪声点等。

5 矩形化

对于连通区域进行矩形化，方便后续选择车牌区域

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

6 OCR

OCR（Optical Character Recognition，光学字符识别）是一种图像处理技术，用于将印刷体文本或手写体文本转换为可编辑和可搜索的电子文本。OCR技术在许多领域都有广泛应用，如文档数字化、自动化数据录入、文字识别等。OCR的准确性和性能受到多种因素的影响，如图像质量、字体、文字大小和风格、背景复杂度等。

报告中运用了已开源的OCR技术`tesseract`

开源代码地址

实验步骤

1 车牌定位

1.1 边缘检测

```
img=imread("test1.jpeg");
img_2=rgb2gray(img);
im_edge=edge(img_2);
im_edge=~im_edge;        %取反!
```

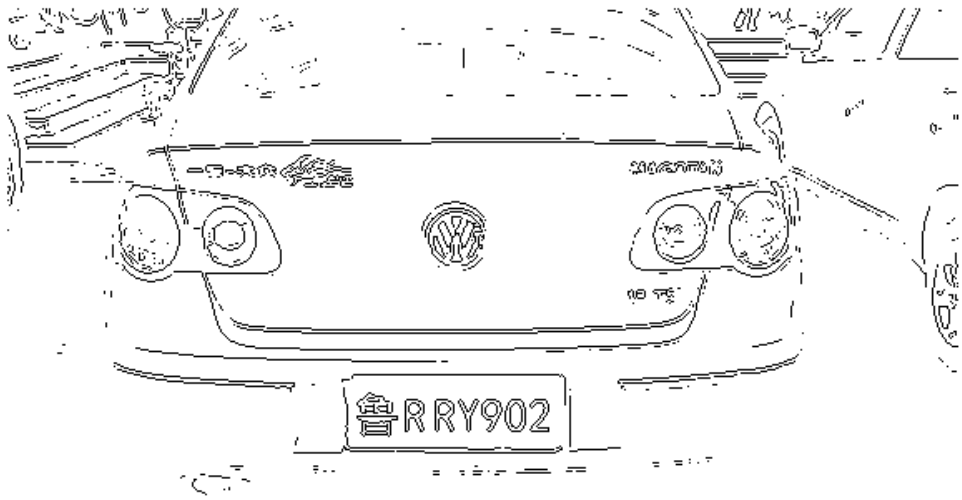


图 1 边缘检测结果

1.2 开闭运算

```
se=strel("rectangle",[10,20]);
im_open=imopen(im_edge,se);
imshow(im_open);
im_close=imclose(im_open,se);
```

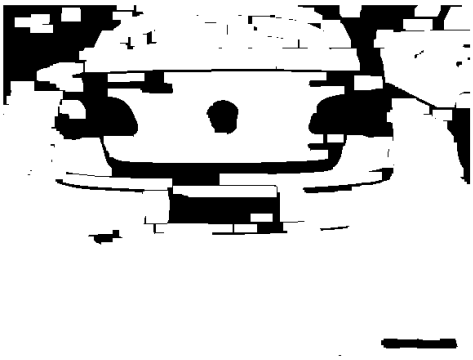


图 2 开运算后结果

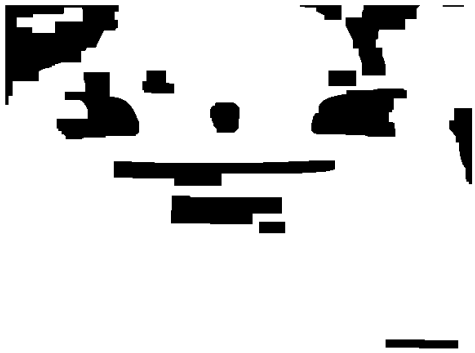


图 3 闭运算后结果

1.3 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，实际操作时，若矩形化效果不好，可以适当改变实验扩散条件，比如当黑点领域中有3个以上白点，对应的点才变成白色，这样更改条件后，会使得扩散的矩阵更大

此处我根据矩形化规则定义了两个函数`iswhite`和`isblack`，分别判断对应的像素点是否满足扩散条件

`retangle`函数则为四次迭代矩形化

图 4 矩形化结果

1.4 筛选区域

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

详细筛选见源码

```
[L, num] = bwlabel(~im_close,8);
stats = regionprops(L, 'Area','BoundingBox');
% 参数设定
ratio_range = [2, 8];
count_range = [15, 45];
blue_threshold = 0.7;
```

2 图形分割

对图片进行二值化，突出车牌数字，分别用下列两种方法并分析

2.1 基于全局的阈值分割（大津法）

```
% 提取候选框内的图像区域
region_image = imcrop(img_2, selected_regions);

% 计算图像区域的全局阈值
threshold = graythresh(region_image);

% 对图像区域进行二值化分割
binary_image_1 = imbinarize(region_image, threshold);
```

2.2 基于局部的阈值分割（**bernsen**）

```
% 设置局部阈值分割参数
window_size = 15; % 窗口大小
contrast_threshold = 15; % 对比度阈值

% 对图像区域进行基于局部的阈值分割
binary_image_2 = local_threshold_segmentation(region_image, window_size,
contrast_threshold);
```

3　字符分离

由于此处图像分割时会有少数白边进入图像中，所以此处如果利用栅栏法进行字符分割效果并不好

下面为栅栏法画的对应的逻辑图

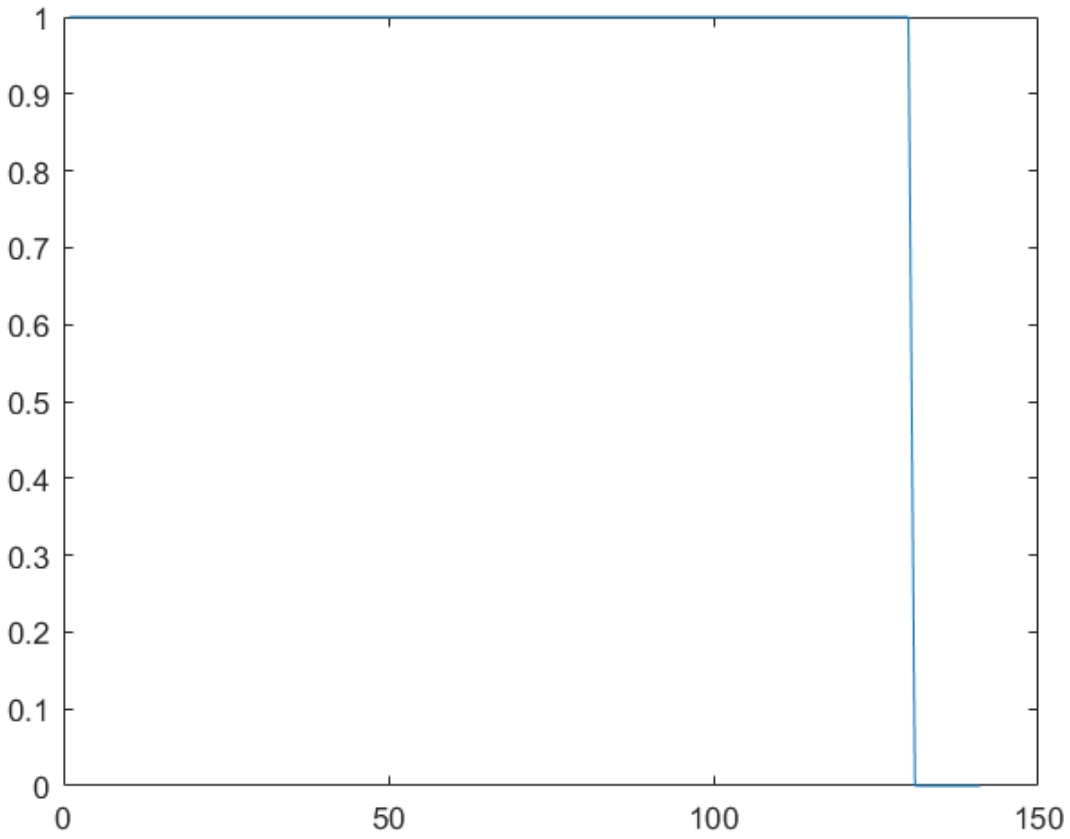


图 5 矩形化结果

可以看出间隔不均匀，说明栅栏法效果并不好

4　OCR识别文字

由于第一次实验中没有字母模板，因此此处选择OCR进行识别

1. 在官方下载地址安装OCR

tesseract-ocr/tessdoc: Tesseract documentation (github.com)

2. 在官方网址下载对应的中文语料训练数据到对应的`tessdata`文件夹下
3. 将上一步分割得到的图片保存到`image`文件夹中，然后在命令行中运行命令进行OCR识别

实验结果

1 候选框选择

图 6 候选框结果

2 图像分割

图 7 全局阈值分割

图 8 局域阈值分割

实验结果分析:

基于全局的阈值分割可以将图像分割成目标物体和背景两部分，有助于后续的图像分析和处理。然而，全局阈值分割方法对于具有复杂背景或光照变化的图像可能效果不佳

基于局部的阈值分割方法能够适应图像中的局部变化和不均匀性，提高分割的准确性和鲁棒性。它在处理光照不均匀、纹理复杂、目标物体大小变化等问题时具有优势。然而，该方法需要选择适当的块大小和自适应阈值计算方法，以获得满意的分割效果。

3 字符识别

以下是两种截图方式的识别结果

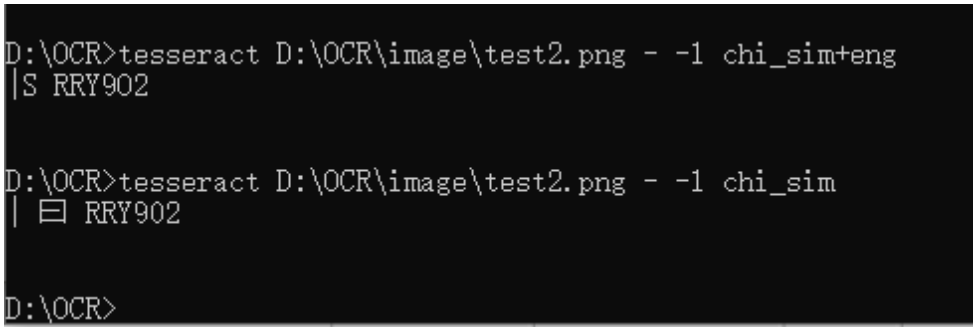


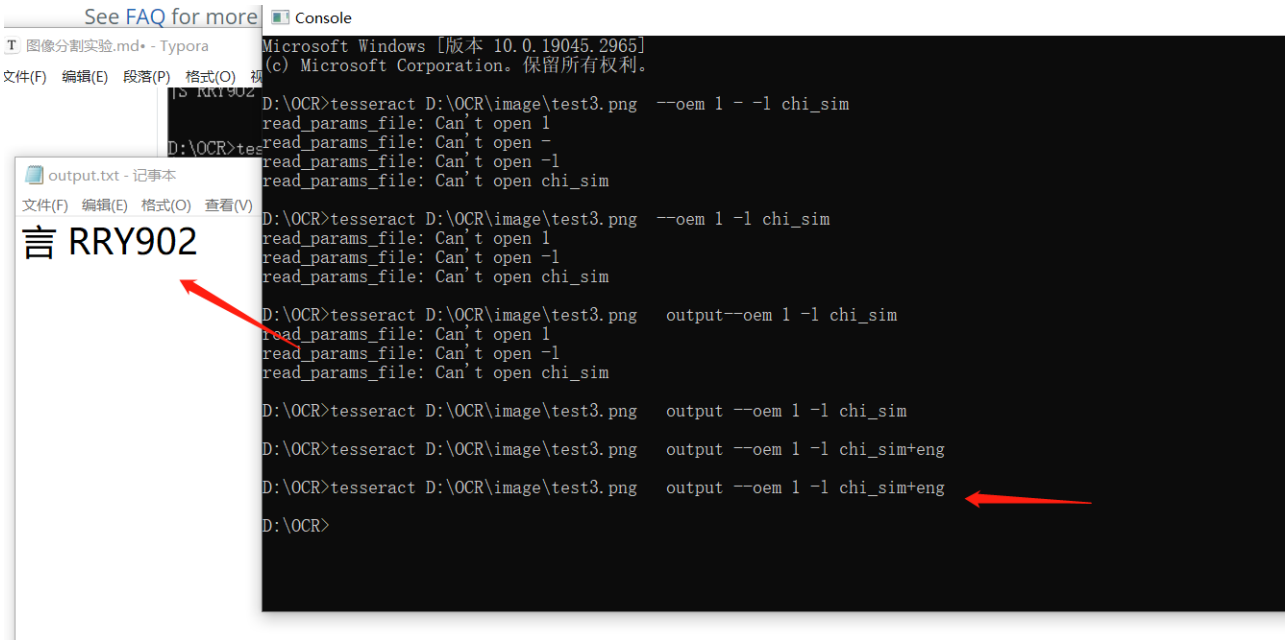
图 1 OCR识别结果

利用`LSTM`model进行预测得到结果

在命令行运行

```
tesseract D:\OCR\image\test3.png output --oem 1 -l chi_sim+eng
```

会在当前路径文件夹下生成`output.txt`，里面存放着对应的识别结果



由于OCR本身问题，只有一个`鲁`被识别成了`言`

总结

实验过程分为**车牌定位和车牌识别**两个主要步骤。

在车牌定位阶段，我们首先进行了边缘检测，通过边缘检测算法（如Sobel、Canny等）提取出图像中的边缘信息。然后，进行了开闭运算操作，使用形态学开闭运算对边缘检测图像进行处理，以平滑物体轮廓并消除噪声。接着，对处理后的图像进行连通区域分析，并将连通区域进行矩形化处理，以便选择车牌区域。最后，根据一些约束条件（如矩形长宽比、蓝色像素比例、垂直投影函数等），筛选出候选车牌区域。

在车牌识别阶段，我们使用了基于全局的阈值分割算法和基于局部的阈值分割（Bernsen算法）对候选车牌区域进行图像分割，然后，使用OCR方法进行字符识别。

实验结果表明，我们成功实现了小车蓝色车牌的定位与识别。