

字符分割

顾秀烨

Table of Contents

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 一、概述..... | 1 |
| 1.1 常用颜色调整算法..... | 2 |
| 1.1.1 直方图均衡化 | 2 |
| 1.1.2 灰度拉伸 | 3 |
| 1.2 二值化算法 | 4 |
| 1.2.1 Sauvola 方法..... | 4 |
| 1.2.2 Bernsen 算法 | 4 |
| 1.3 一般车牌分割算法..... | 4 |
| 1.3.1 垂直投影法 | 5 |
| 1.3.2 连通区域法 | 5 |
| 1.3.3 静态边界法 | 5 |
| 1.4 我国车牌的基本情况（作为先验知识） | 5 |
| 二、一些论文与系统的字符分割方案..... | 6 |
| 2.1 于深洋 硕士论文《自然环境下的车牌定位与字符分割方法》 | 6 |
| 2.2 openALPR 系统中使用的字符分割方法 | 8 |
| 2.3 easyPR 系统中使用的字符分割方法..... | 9 |
| 2.4 Category-specific extremal region detection | 10 |
| 2.5 Yungang Zhang, Changshui Zhang, A New Algorithm for Character Segmentation of License Plate | 11 |
| 2.5.1 预处理 | 11 |
| 2.5.2 水平分割 | 12 |
| 2.5.3 垂直分割 | 13 |
| 2.6 Xiangjian He, Lihong Zheng, Qiang Wu, Wenjing Jia, Bijan Samali, Marimuthu Palaniswami. Segmentation of Characters on Car License Plates..... | 13 |
| 2.6.1 去除上下边界 | 13 |
| 2.6.2 二值化 | 13 |
| 2.6.3 垂直投影 | 13 |
| 部分参考文献: | 14 |

一、概述

目前字符分割一般由对车牌图像进行预处理（去除边框、铆钉，颜色调整）、二值化、分割算法、后期处理（post-processing, optional）4 个步骤组成，并且分割时一般会考虑先验知识（车牌的颜色、大小，字符的大小、位置等）。每个步骤都有一些常见方法。许多论文与开源系统都是这些常用方法的组合，并加上一些改进，或通过后期处理进行调整。所以在这里首先简述每个步骤的常用方法。

1.1 常用颜色调整算法

1.1.1 直方图均衡化

灰度直方图均衡化是数字图像处理中图像点变换的一种常用方法，其目的是变换灰度的统计直方图使得在输出的图像中每种灰度大致有相同的点数，这样输出的图像看上去会比较清晰。

计算方法为：

一幅图像中灰度级 r_k 出现的概率近似表示为：

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L-1 \quad (4-1)$$

其中， n 是图像中像素的总和， n_k 是灰度级 r_k 的像素个数， L 为图像中可能的灰度级总数，变换函数的表示形式为：

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}, \quad k = 0, 1, 2, \dots, L-1 \quad (4-2)$$

将输入图像中的灰度级为 r_k 的各像素映射到输出图像中灰度级为 s_k 的对应像素即可。如下图所示：

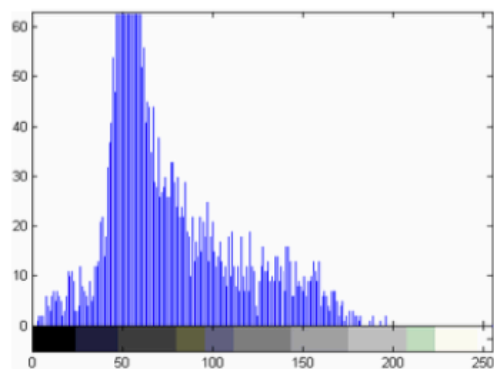


图 4-1 原灰度图像直方图

Figure 4-1 Gray histogram before translating

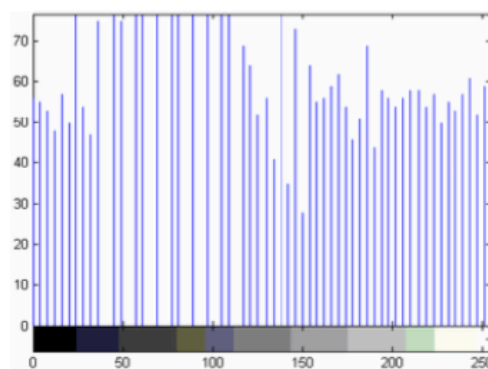
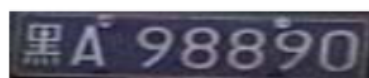


图 4-2 均衡化后的灰度直方图

Figure 4-2 Gray histogram after Linearization

车牌处理前后的效果图：



a) 原彩色图像



b) 原灰度图像



c) 直方图均衡化后

1.1.2 灰度拉伸

灰度拉伸是指根据灰度直方图的分布有选择的拉伸某段灰度区间以改善输出图像。如果一幅图像灰度集中在较暗的区域而导致图像偏暗，可以用灰度拉伸功能(斜率 $A>1$)来拉伸物体灰度区间以改善图像; 同样如果图像灰度集中在较亮的区域而导致图像偏亮，也可以用灰度拉伸功能(斜率 $A<1$) 来压缩物体灰度区。在 0 到 255 之间按顺序取两个点 x_1 和 x_2 ，它们将 0-255 之间的灰度值划分成 $[0, x_1]$ ， $[x_1, x_2]$ 和 $[x_2, 255]$ 的三个灰度区间。

假设车辆牌照区域的前景像素点和背景像素点的灰度值主要分布在 $[0, x_1]$ 和 $[x_2, 255]$ 两个区间内，采用分段式的线性拉伸方程强化车牌前景和背景的灰度差异，通过调整 y_1, y_2 点的大小来改变灰度拉伸方程在 $[0, x_1]$ 和 $[x_2, 255]$ 区间的斜率，加深车牌背景灰度，增亮车牌区域前景字符，从而增加车牌中前景和背景 的对比度，突出了字符区域，并且有效的抑制一部分其它的干扰信息。

灰度拉伸的变换函数表示如下：

$$f(x, y) = \begin{cases} \frac{y_1}{x_1} x & (x < x_1) \\ \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1 & (x_1 \leq x \leq x_2) \\ \frac{255 - y_2}{255 - x_2} (x - x_2) + y_2 & (x > x_2) \end{cases}$$

作用于灰度图像上的增强效果如下图所示：



论文【8】为做灰度拉伸时， x_1, x_2, y_1, y_2 的选择提供了一种方案：

(1) 计算车牌上每个灰度等级的密度，灰度等级的密度平均值为阈值。

阈值 = 车牌宽度 * 车牌高度 / 256

(2) 设 x_1 为密度小于阈值的最小的灰度等级（若不存在，则为 0），设 x_2 为密度小于阈值的最大的灰度等级（若不存在，则为 255）， $y_1 = 0$ ， $y_2 = 255$ 。

效果图：



Fig.3 Original Image



Fig.5 License Plate after Enhancement

1.2 二值化算法

二值化算法最常见的为大津法，以及经过改进的 **local adaptive** 的大津法。而局部自适应的大津法速度较慢，后来人们又提出了一些其他快速的局部二值化方法。Opencv 自适应的二值化方法采用的就是大津法，在这里就不做介绍了。

1.2.1 Sauvola 方法

其他 **local adaptive** 的二值化方法有 **Sauvola** 方法，后来 Shafait 使用 **Integral Image** 加速了这种方法，阈值的选择方式如下：

In Sauvola's binarization method, the threshold $t(x, y)$ is computed using the mean $m(x, y)$ and standard deviation $s(x, y)$ of the pixel intensities in a $w \times w$ window centered around the pixel (x, y) :

$$t(x, y) = m(x, y) \left[1 + k \left(\frac{s(x, y)}{R} - 1 \right) \right] \quad (2)$$

where R is the maximum value of the standard deviation ($R = 128$ for a greyscale document), and k is a parameter which takes positive values in the range $[0.2, 0.5]$. The local mean $m(x, y)$ and standard deviation

1.2.2 Bernsen 算法

考虑以 (x, y) 为中心的 $(2W + 1) \times (2W + 1)$ 模板，则根据 **Bernsen** 算法有：

1. 计算每个像素点的阈值

$$T(x, y) = 0.5 \times \left(\max_{\substack{-W < k \leq W \\ -W < l \leq W}} f(x + k, y + l) + \min_{\substack{-W < k \leq W \\ -W < l \leq W}} f(x + k, y + l) \right) \quad (4-4)$$

2. 如果 $f(x, y) > T(x, y)$ ，则 $b(x, y) = 1$ ，否则 $b(x, y) = 0$ 。

1.3 一般车牌分割算法

目前常用的车牌字符分割算法主要有以下三种：垂直投影法、连通区域法和静态边界法。

垂直投影法可以很容易地确定字符边界，但字符宽度阈值的选取在这个方法中十分关键；连通区域法可以分割清晰的车牌图像，但对图像质量要求过高；静态边界法不受噪声影响，但它依赖于车牌图像的正确提取。【1】

1.3.1 垂直投影法

垂直投影法利用了字符块在垂直方向上的投影必然在正确的分割位置上 (即字符或者字符内的间隙处)取得了局部最小值, 且这个位置要满足车牌照字符规则和字符尺寸限制。对车牌照区域作垂直方向上投影, 首先在水平方向上从左至右检测各坐标的垂直投影数值, 当找到第一个局部最小值点的时候, 认为这个点是最左面字符的边界。然后在水平方向上从右至左检测坐标的垂直投影数值, 当找到第一个局部最小值点的时候, 认为这个点是最右面字符的边界。得到两边字符的边界之后, 用同样方法可以找到每个字符的边界。

如果单纯采用垂直投影法进行车牌字符分割, 很容易产生字符切分过度的现象, 即将一个字符的切分成了两块。因为有的字符的中间部分有可能会出局部极小点, 如汉字中的“川”, 英文字符中的“M”, 这些字符都很可能被过度切分, 因此采用垂直投影法一定要首先确定好切分宽度阈值和投影最小值的标准, 而它们的取值对于实际情况的依赖较强, 并不容易确定。

1.3.2 连通区域法

采用基于连通域的切分方法是利用了每个字符或字符的一部分构成连通域的特性, 把这些连通域切分出来, 也就切分出了车牌字符。因为通常情况下车牌上的字符除汉字外, 其它的都是字母或数字, 它们在理想状态下是全连通的, 这样就可以使用连通域的方法来对车牌字符进行分割, 而第一个字符可以利用其与后续字符的相对关系, 排列规则等信息实现切分。

但在实际情况中, 车牌区域内往往存在着很多的噪声区域, 这些区域与车牌中的字符线条粘连在了一起, 使字符的连通域特性变得很不规范, 甚至有的时候车牌中的很多字符与车牌上下边框粘连成一个很大的连通区域, 如果仍然使用连通区域法进行车牌字符分割将很难得到令人满意的结果。

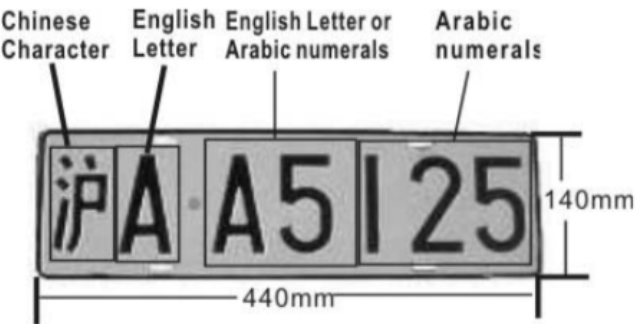
1.3.3 静态边界法

静态边界法认为定位阶段提取的车牌区域就是车牌的精确位置, 即车牌的边界已经准确的定位。利用车牌本身存在的模板特征来分割车牌, 但是事实上这对于车牌定位阶段的要求过高, 因为定位阶段所能得到的区域边界很难恰好是真正的车牌边界, 这就使得应用其进行字符分割的误差较大。有时分割结果可能产生偏移, 严重的影响了分割的准确性, 而且一旦出现偏移, 分割得到的所有字符图像都将是错误的, 这时识别结果的可信性将难以保障。但是如果在此基础上加之动态的边界匹配规则, 分割准确率会大大的提高。

1.4 我国车牌的基本情况 (作为先验知识)

我国车牌有黄底黑字、蓝底白字、白底黑字、黑底白字四种情况。

车牌大小为 440*140mm，每个字符大小为 45*90mm，字符间距为 12mm，前两个字符与后五个字符之间有一 34mm 的大间距。



二、一些论文与系统的字符分割方案

2.1 于深洋 硕士论文《自然环境下的车牌定位与字符分割方法》

(1) 使用了 1.2.2 节中介绍的 Bernsen 算法做了局部阈值二值化。由于车牌图像的局部灰度差异性很多时候表现在垂直方向，本文采用的局部阈值法是将车牌图像分成上下四段区域，以每段区域中间部分的迭代阈值作为该段区域的分割阈值。该方法有效克服了车牌区域纵向光照不均造成的线条 不规则现象，且运算速度快。效果图如下：

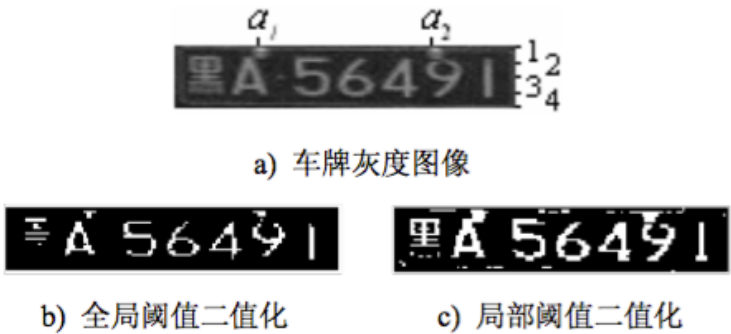


图 4-6 全局和局部阈值法比较

(2) 多尺度模板匹配方法应用于车牌字符分割

a. 车牌原始模板

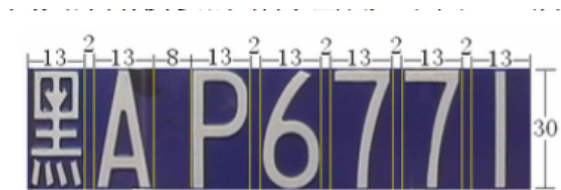


图 4-8 车牌原始模板图

Figure 4-8 License plate source template

b. 字符分割过程中判优函数的选取

定义模板 M' 作用于 I 的覆盖率 $c(M', I)$:

在车牌图像中符合模板 M' 大小的窗口区域 I 处套用 M' ， I 中七个字符区域内的白点总数与 I 中所有白点总数的比值称为模板 M' 作用于 I 的覆盖率，记为 $c(M', I)$ 。

定义模板 M' 作用于 I 的差异率:

设车牌中窗口区域 I 套用模板 M' 后七个字符图像块中的白点数分别为 N_1, N_2, \dots, N_7 ,

令 $N_{\max} = \max\{N_1, N_2, \dots, N_7\}$,

$N_{\min} = \min\{N_1, N_2, \dots, N_7\}$,

$N_{\text{mid}} = \text{中位数}\{N_1, N_2, \dots, N_7\}$,

则此模板 M' 作用于 I 的差异率定义为:

$$\sigma(M', I) = (N_{\max} - N_{\min}) / N_{\text{mid}}$$

模板 $M' = \theta M$ ，这里 θ 为线性变换函数，简化为只含有位置和伸缩比例的三参数变化向量定义模板 M' 作用于 I 的判优函数如下式所示:

$$F(\bar{\theta}) = \begin{cases} c(\bar{\theta}M, I), & \sigma(\bar{\theta}M, I) \leq \Delta\sigma \\ 0, & \sigma(\bar{\theta}M, I) > \Delta\sigma \end{cases}$$

此时，取得最优模板匹配信息的方法为:

$$\bar{\theta}^* = \arg \max_{\bar{\theta} \in R^3} \{F(\bar{\theta})\} \text{ subject to } \sigma(\bar{\theta}M, I) \leq \Delta\sigma$$

c. 具体实现算法

算法首先利用尺度变换的方法找到车牌区域的全局最优模板匹配信息，当覆盖率超过一定阈值时，则认为该区域是真车牌，并可通过最优 θ 对应的参数集合计算出每个字符图像的位置；否则，认为该区域是伪车牌，将其删除从而达到去伪的作用。

算法:

1. 车牌模板总长度初始化为 $W/2$ ，车牌模板首行和首列匹配位置都初始化为 0，匹配尺度标记 $i = 0$ 。
2. 按照此模板尺度对整个车牌二值图像进行滑动匹配，根据式 4-7 计算此尺度下的最优模板匹配信息 $\bar{\theta}_i^*$ ，并置尺度标记 $i = i + 1$ 。
3. 逐渐扩大模板尺度，如果扩大后的模板总长度小于 W 并且模板高度小于 H ，则转回 2，否则置尺度总数 $n = i$ 。
4. 计算得到全局最优模板匹配信息 $\bar{\theta}_{\max}^* = \arg \max_{i=0}^{n-1} \{F(\bar{\theta}_i^*)\}$ 。若判优函数值 $F(\bar{\theta}_{\max}^*) \geq \Delta c$ ，则返回 $\bar{\theta}_{\max}^*$ ，否则返回 0。

采用此算法对图 4-9 所示的车牌图像进行分割，结果如图 4-10 所示:

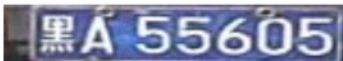


图 4-9 车牌图像

Figure 4-9 Image of vehicle plate



图 4-10 字符分割结果

Figure 4-10 Result of character segmentation

(3) 字符分割后处理

实验中我们发现车牌字符分割结果中往往存有若干的噪声区域。例如在字符分割结果中包含了车牌的上下边框等噪声区域，这些噪声区域会严重影响字符识别阶段的识别正确率。为了有效地剔除字符图像中的这些噪声区域，本文采用了基于字符子图像最大连通域的边界均值方法。

由于我国车牌的第一个字符为汉字，而汉字中含有的连通域较多，不对汉字提取连通区域。

具体做法简述如下：对于字符分割结果中后面的六个字符，分别得到它们每个图像中所有的连通区域，把每个图像中外接矩形面积最大的连通区域 c_1, c_2, \dots, c_6 认为是后六个字符的纹理。由于部分字符的纹理可能会与上下边框相粘连，所以在取得后面六个字符的上下边界值之后，分别去掉上边界和下边界最高和最低两对极值，再对剩余四个字符图像的上下边界值取平均值，最后将得到上、下边界均值作为包括汉字在内的所有字符图像的统一上下边界。而每个字符图像的左右边界则分别依据 c_1, c_2, \dots, c_6 的左右边界而定。根据上述算法，取得所有字符图像上下边界的方法如下式所示：

$$Top = \frac{\sum_{i=1}^6 c_{上}^i - \max_{i=1}^6 \{c_{上}^i\} - \min_{i=1}^6 \{c_{上}^i\}}{4}, \quad Bottom = \frac{\sum_{i=1}^6 c_{下}^i - \max_{i=1}^6 \{c_{下}^i\} - \min_{i=1}^6 \{c_{下}^i\}}{4}$$

噪声后的字符分割结果如下图所示：



(4) 评价

该方法有效地克服了实际应用中存在的车牌模糊、噪声区域多等问题，具有很好的鲁棒性和切分准确性。（作者自评，实际结果如何有待考量）

2.2 openALPR 系统中使用的字符分割方法

作者在 design document 里是这么写的：

“The character segmentation phase tries to isolate all the characters that make up the plate image. It uses a vertical histogram to find gaps in the plate characters. This phase also cleans up the character boxes by removing small, disconnected speckles and disqualifying character regions that are not tall enough. It also tries to remove "edge" regions so that the edge of the license plate doesn't inappropriately get classified as a '1' or an 'l'”

可以看到，这里主要使用的是垂直分割方法，另外去除了一些干扰的污点、不符合高度要求的字符区域与左右边界。

2.3 easyPR 系统中使用的字符分割方法

总体处理流程：



1. 灰度化，因为算法是针对灰度化图片的，也能提高效率。
2. 颜色判断，详见车牌定位综述中，他们使用颜色定位中的模板匹配法。
3. 二值化，蓝色车牌直接使用 `CV_THRESH_BINARY`，黄色车牌使用 `CV_THRESH_BINARY_INV`（全局大津法）。
4. 取轮廓（`findContours` 函数）。

而对于一些中文字符的轮廓会断裂的现象（如“苏”字会被分为上下两个连通区域），则可以先不处理中文字符，先获取后面六个字符。



汉字后面的字符，如‘E’，在中文车牌里代表城市的代码，在这里简称它为“城市字符”或者“特殊字符”。

这个字符有一个特征，就是与它后面的字符存在一定的间隔。但是与前面的中文字符靠的较紧。首先获取这个特殊字符的外接矩形，再把这个外接矩形向左做一些的偏移（偏移的大小可以通过经验指定，例如设置为字符宽度的 1.15 倍），这样这个外接矩形就成了包含中文字符的一个矩形了。下面就可以截取中文字符的图块。

下图就是“特殊字符”与被反推得到的“中文字符”的矩形，在图中用红色矩形表示。



下面的问题就是如何获取“特殊字符”的位置？

依次判断所有取轮廓操作得到的矩形的位置，设矩形的中点恰好在整个车牌的 $1/7$ 到 $2/7$ 之间的矩形为特殊矩形。这样操作的前提是我们的车牌定位的非常准确，恰到把整个车牌截取的正正好。在这种情况下，只要外接矩形满足这些条件，就可以判断为特殊字符的矩形。



5. 找外接矩形，成功获取所有字符的外接矩形。
6. 截取图块并归一化到统一的格式。

这个方法思路很简单，实际中应用效果也不错，因此也是 EasyPR 目前采用的方法。

其中一些细节处理如下：

（1）颜色判断时，可以将四周截去一小部分再进行颜色判断，这样可以消除车牌定位不准确带来的杂色背景的干扰。

（2）获得中文字符图块后，将其他图块从左到右排序，一次选择 6 个即可，可以避免将最后的边框选入：



（3）去除铆钉。

中国车牌有时会有铆钉。倘若对一副含有铆钉的图进行二值化，极有可能会出现下图的结果。一些字符图块（下图的"9"和"1"）通过铆钉的原因联系到了一体，那样的话就无法通过取轮廓操作来分割了。



去除铆钉的思想为：依次扫描每行，判断跳变次数。车牌字符所在的行的跳变次数是很多的，而铆钉所在的行就会偏少。因此当发现某行跳变次数较少，则可以把该行的所有像素值赋值为 0，这样就会大幅度消除铆钉的影响了。效果图：



2.4 Category-specific extremal region detection

该方法在车牌检测章节中已经叙述过了。

得到 ROI 中所有的 extremal region，（车牌上的字符们会被检测为 extremal region）。通过对每一个 extremal region 提取出一些特征：normalized central algebraic moments, compactness, Euler number, Entropy of cumulative histogram (这些特征都是 incrementally

computable 的，所以可以在线性时间得到)。将特征放到 SVM 中，进行训练（车牌与非车牌）与检测。

该方法的好处在于 plate detection 与 segmentation 一起完成了，效率高。

2.5 Yungang Zhang, Changshui Zhang, A New Algorithm for Character Segmentation of License Plate

这篇文章提出的算法主要分为三个阶段：预处理，水平分割，垂直分割。

2.5.1 预处理

(1) 统一大小。

将所有车牌图片统一为 160*40 (pixel) 的大小。

(2) 决定车牌种类。

将车牌图片转化为灰度图，根据白色像素点与黑色像素点的比例，将所有的车牌灰度图转化为黑底白字。（【4】中提到，一张车牌上，字符的像素比例平均为 20%）

(3) 物体增强。

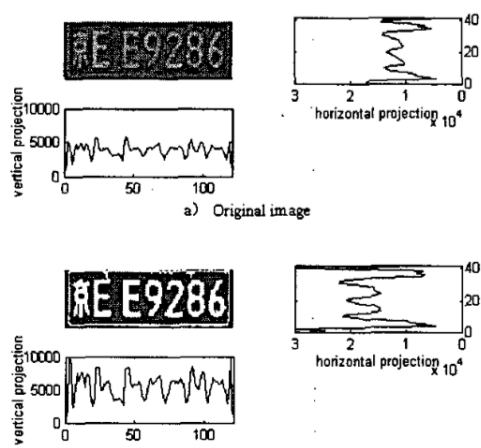
由于车牌图像的质量有很大不同，由于照明、噪音干扰等原因。在这里，作者没有简单地使用直方图均衡化，或灰度拉伸，因为这两种方式会增强噪音。

作者认为，只有字符部分应该得到增强，而背景部分应该被抑制。所以这种方法被称为物体增强（object enhancement）。

首先，将所有像素的灰度等级正规化到 0 – 100 的范围里。这样就将整张图片都减弱了。

然后，根据灰度等级的大小降序排列。选择灰度等级排在前 20% 的相似，将它们的灰度值乘以 2.55。这样，最有可能是字符的像素点被增强了。

这样做以后，可以发现，水平投影与垂直投影的峰与谷之间的对比度更强烈了。如下图所示：



2.5.2 水平分割

这篇论文相较于其他论文比较特殊，他不但在垂直方向将字符切开，还在水平方向切除了字符以外的部分。这样一来，去除了边框、铆钉的干扰，提高了字符识别部分的准确性。

对于字符倾斜程度较大的车牌来说，很难使用水平投影获得水平分割线。但如果首先将车牌使用垂直投影将字符分割开（并不一定分割得完全准确），那么这样以后，再使用水平投影，就能获得比较准确的水平分割线了。具体算法如下：

- (1) 找到垂直投影的波谷，将车牌垂直分割为几块，此分割不一定很准确。
- (2) 对于(1)中得到的块，使用水平投影，获得水平分割线。
- (3) 对于获得的水平分割线的中点使用霍夫变换，以去掉不准确的分割线，将准确的水平分割线连成一条直线。

这个方法的好处在于：

- (a) 霍夫变换使用了投票的策略，不准确的水平分割线会少数服从多数而被筛选出来，而直线拟合的算法对不准确的分割线很敏感。例如，可以纠正由车牌上的铆钉造成的错误的水平分割线。
- (b) 这是一个局部的投影方法，光照差异、旋转都来的影响都将大大减小。
- (c) 这避免了对图片进行仿射变换，事实上，仿射变换对使得图片扭曲，给字符识别带来一定困难。（这一点我认为不成立，原文甚至写的是 **rotation**，**rotation** 并不能使图片扭曲）

实验表明，该方法能够在有背景噪音、光照差异、旋转、铆钉、车牌边框的情况下获得不错的效果，效果图如下：



(a) Images with rotation



(b) Images with noise



(c) Images with illumination variance



(d) Images with rivet and plate frame

2.5.3 垂直分割

- (1) 在垂直投影的波谷中寻找候选的垂直分割线。
- (2) 使用先验知识与水平分割线估计车牌与每个字符的大小。
- (3) 使用先验知识估计前两个字符与后五个字符之间的大间隔的位置。

在一条垂直分割线上，它的灰度变化应该比较小，因为他贯穿的是灰度相近的背景区域。相反，当他穿过字符时，灰度变化会比较大、比较多。基于这些，在候选的垂直分割线中、及他们的周围，确定大间隔的位置。

- (4) 其他垂直分割线都可以使用相似的方法定位。

2.6 Xiangjian He, Lihong Zheng, Qiang Wu, Wenjing Jia, Bijan Samali, Marimuthu Palaniswami. Segmentation of Characters on Car License Plates.

此方法主要涉及三个主要步骤：使用水平投影去除上下边界，二值化，垂直投影。

2.6.1 去除上下边界

- (1) 垂直边缘检测。

计算水平梯度，使用【-3 0 3; -10 0 10; -3 0 3】的 Sobel 算子获得水平梯度值。

使用 Otsu 二值化方法对检测到的边缘做二值化。

- (2) 水平投影。

水平投影直方图的平均值作为阈值来确定上下边界。

为了更好地做字符分割，他们将上边界降低了 2 个像素，将下边界提高了 2 个像素。

- (3) 去除上下边界。

2.6.2 二值化

二值化使用的是在概述中提到的全局大津方法，在这里就不赘述了。

2.6.3 垂直投影

- (1) 在去除了上下边界的图像上做垂直投影，并做初步分割。

(2) 由于字符可能受车牌午饭的影响而粘连在一起，或者由于上下边框去除得不到位而将字符分割开了，例如“J”，需要精细化字符分割。

任意两个字符，若宽度小于平均字符宽度的 $\frac{2}{3}$ ，需要合并为一个字符。

任意字符，宽度超过平均字符宽度的 $\frac{1}{3}$ ，则需要进一步分割。

循环这一过程，直到字符数目符合先验知识，或字符宽度都符合要求为止。

在另一篇论文中，有类似的精细化过程，个人认为更加合理一些（因为一些字符的宽度本来就很小，例如“1”），它在合并字符时，选择的判断准则是相邻两个投影块中线之间的距离。【7】

部分参考文献：

- 【1】于深洋，《自然环境下的车牌定位与字符分割方法》。
- 【2】easyPR 开发详解。
- 【3】Jiri Matas, Karel Zimmermann. Unconstrained Licence Plate and Text Localization and Recognition.
- 【4】J.A.Hegt, R.J.Haye, N.A.Dela&Khan. A high performance license plate recognition system. Proc. IEEE International Conference on System, Man and Cybernetics, pp. 4357-4362, 1998.
- 【5】Yungang Zhang, Changshui Zhang. A new algorithm for character segmentation of license plate.
- 【6】Xiangjian He, Lihong Zheng, Qiang Wu, Wenjing Jia, Bijan Samali, Marimuthu Palaniswami. Segmentation of Characters on Car License Plates.
- 【7】冉令峰。基于垂直投影的车牌字符分割方法。
- 【8】Feng Yang, Zheng Ma, Mei Xie. A Novel Approach for License Plate Character Segmentation.