

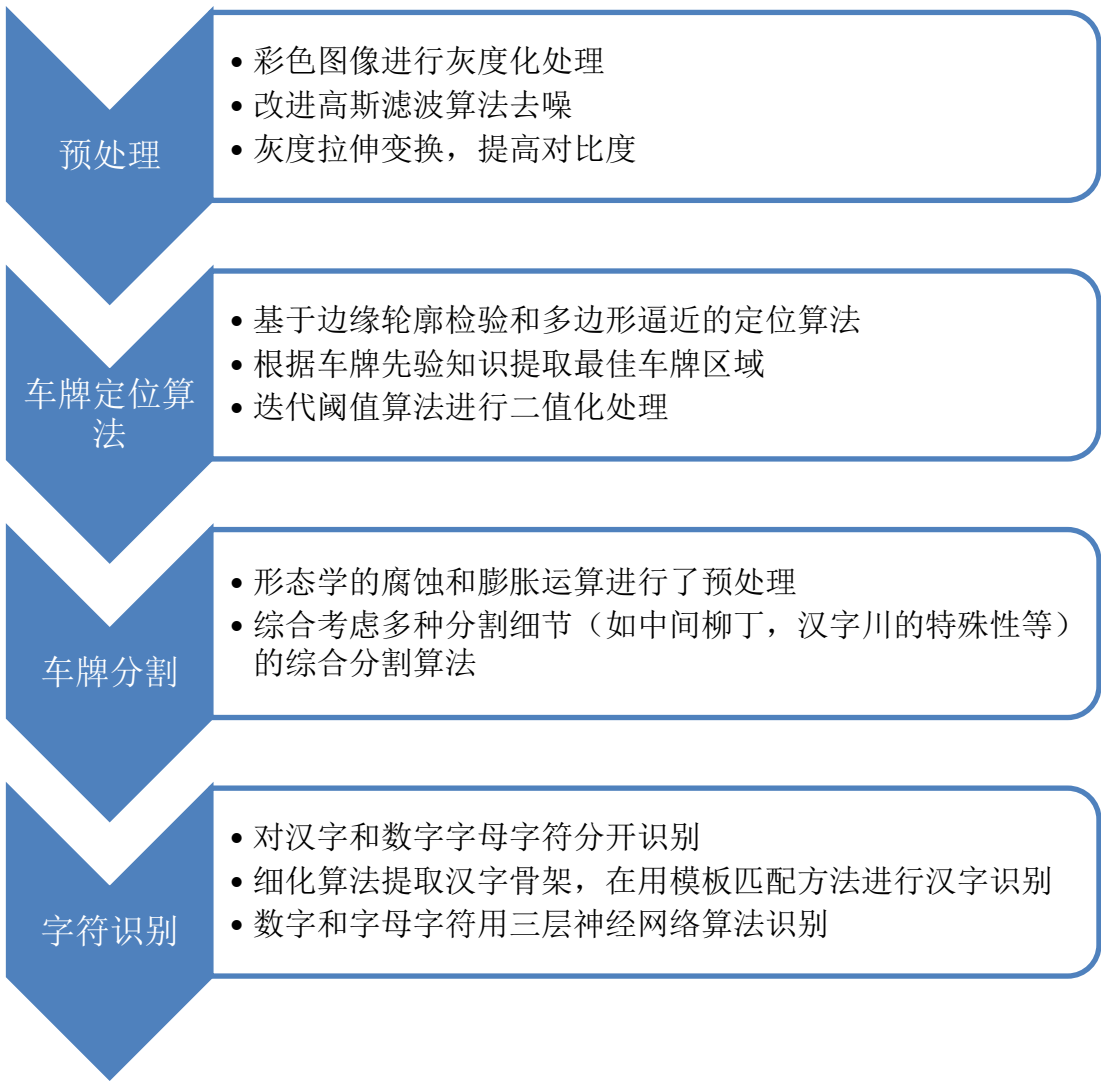
车牌识别算法的设计与实现

21121229 姜锦正

1 总体设计骨架

车牌识别总体分为如下四个过程：

1) 预处理 2) 车牌区域的定位 3) 车牌分割算法 4) 字符识别算法



操作的过程描述如下：

首先，对图像进行预处理，包括进行彩色图像灰度化处理以便减少计算量，用一种改进的高斯滤波策略进行去噪声处理，并在此基础上，进行灰度拉伸变化，提高对比度的同时，也提高了接下来的车牌定位成功率。

其次，在充分了解车牌的先验知识的基础上，结合已有的定位算法，提出了一种基于边缘轮廓检验和多边形逼近的定位算法。该算法利用 Canny 算法提取边缘轮廓，然后用边界跟踪算法提取出边界信息，利用多边形逼近算法来查找可能的候选多边形轮廓，并根据车牌的先验信息选出一个最佳的车牌区域。

再者，对提取出来的车牌图像用迭代阈值算法进行二值化处理。鉴于单纯的垂直投影分割方法，会存在没有分割开，和分割不到位的缺点，本文先是用形态学的腐蚀和膨胀运算进行了预处理，再综合考虑多种分割细节（如中间柳丁，汉字川的特殊性等）的情况下提出了一种改进的基于多细节的分割算法。

最后，对于汉字和一般数字字母字符分开进行识别。汉字的字符结构复杂，不易于提取特征值，用神经网络方法反而不易于识别。本文先是用细化算法提取汉字的骨架，再用模板匹配的方法进行汉字的识别。一般的数字字母字符相对比较简单，特征值易于提取，本文构建了三层的神经网络来识别，准确率较高，并且可以利用识别出来的图像进行网络权值的再更新。

2 模块设计与实现

§2.1 预处理

去噪的方法常见有下面几种：1) 均值滤波 2) 高斯平滑算法 3) 中值滤波



图 1 效果对比

从上图分析可以看出，均值滤波（用的是 3×3 模板）使得图像变得比较模糊，尤其车牌区域对比度减低，不易于后期的处理。高斯滤波（ $\sigma = 0.6$ ）减低了噪声的同时，也是图像的细节变得模糊了。中值滤波（用的是 3×3 模板）同样，模糊的更加厉害，车牌的“Q”字母都基本变为了数字“0”。

噪声一般是那些灰度跳变比较剧烈的地方，而一般的区域则没有如此大的灰度跳变。于是可以利用这一特性，选择性的进行滤波处理，即在灰度跳变比较大的点，就是认为的噪声点，进行高斯滤波，而一般的平常点，就是认为不是噪声的点，就不进行处理。

我这里设计的算法思想如下：

- 1) 如果局部区域的两个极值之差大于一定的阈值 T ，即：

$$\text{Max}(\text{Region}) - \text{Min}(\text{Region}) \geq T$$

- 2) 如果局部区域的方差大于某一个阈值 T ，即：

$$D(\text{Region}) \geq T$$

改进的高斯滤波算法描述如下：

```
for (所有的像素点){  
    以该像素为中心，计算领域内的最大值，最小值和反差；  
    if (符合噪声点的判断要素)  
        进行噪声的高斯滤波；  
    else  
        不处理该点；  
}
```

对比效果图：



1) 高斯滤波

2) 改进后的高斯滤波

图 2：改进后算法

利用灰度拉伸提高对比度可以将图像的灰度等级投影到全部的 255 个等级，
有较好的对比度之后，对于后续的识别是非常关键的。

效果图如下：



图 3 灰度拉伸效果图

§2.2 车牌定位

先分析一下车牌定位中会遇到的主要几个问题：

- 1) 图像的质量会对图像的定位起到很大的影响
- 2) 车牌的种类比较多，颜色，大小，位置都不尽相同。
- 3) 成像的角度很多，有时候车牌会倾斜甚至变形
- 4) 路边一些广告牌等相似物也会影响牌照的准确识别

根据这些主要牌照特征，目前的车牌定位方法概括起来可分为四大类：

- 1) 基于水平方向纹理特征的车牌定位方法：该类方法特点是速度快，漏检率低，但定位不够准确，容易将许多不相关区域当成车牌区域一并定位出来，因此误检率高。
- 2) 基于车牌色彩特征的车牌定位方法：该类方法因其包含颜色信息，所以定位较准确，但因为车牌颜色容易受到光线、背景颜色的干扰，因而适应性差，一般不单独使用，会与另外几种方法结合在一起定位车牌，效果较好。比较典型的是将图像从 RGB 颜色模型转换到 HLS 颜色模型^[6]，利用车牌颜色在该模型中的特征定位出车牌的可能区域，再利用边缘位置直方图来精确定位出车牌。
- 3) 基于边缘检测的车牌定位方法：该类方法特点是定位准确，定位出的不相关区域少，但定位速度慢，漏检率也较高。

4) 基于数学形态学的车牌定位方法^[7]：该类方法对于那些背景简单的特定场景下的车牌定位效果较好，且受牌照字符与大小限制较大，速度较慢。

鉴于时间和效果的考虑，首先去研究了基于形态学的方法，大致思想是：首先多次利用形态腐蚀和膨胀进行处理，分析出联通区域，再根据车牌的特征要求对车牌区域来定位。最后得出的结论就是，这个方法对参数的和图像的质量比较敏感，分析出来的联通区域也经常不正确，有时甚至会将车牌分割断裂，于是放弃了这个思路。

§2.2.1 二值化算法

基于纹理的方法同样也是受背景的影响较大，错误率很高。对于基于颜色的定位方案，比较的复杂，而且国内的车牌颜色种类很多，不好统一分析。最终决定以基于边缘检测的方法来研究分析。

- 1) 选择一个 T 作为初始值。
- 2) 利用阈值 T 把图像分割成两个区域 R_1 和 R_2 。
- 3) 对区域 R_1 和 R_2 中的所有像素计算平均的灰度值 μ_1 和 μ_2 。
- 4) 计算新的阈值：

$$T = 1/2(\mu_1 + \mu_2)$$

- 5) 重复 2~4 的过程，直到迭代到指定的最大次数，或者计算的阈值不再变化为止。

用迭代算法计算出来的阈值是由图像的结构信息决定的，不是主观定的值，自适应性较好，也就会得到更好的效果。

然后使用 Canny 算子提取边缘：



二值化效果

Canny 算法效果

图 4 效果图

§2.2.2 边界提取算法

Canny 提取出来的边缘是针对整个图片的，而我们分析的 是针对具体的某一个轮廓来分析（分析是否是多边形）。于是还要设计出提取边界的算法，将整个图像的轮廓提取为一个一个的具体轮廓。

边界提取算法就可以解决这个问题，其大致算法思想如下：

为了依次记录下边界上的各个像素，边界跟踪首先按照某种扫描规则找到目标物体边界上的一个相似物，而后就以该点作为起始点，再根据某种顺序（如顺时针）依次找出物体边界上的其余像素，直到又回到了起始点，完成整条边界的跟踪。

例如，我们可以按照从左到右，从上到下的顺序扫描图像，这样首先会找到目标物体的最左上角的边界点 P_0 ，显然，这个点的左侧和上侧都不可能存在边界点（否则左侧或者上侧的边界点就会成为第一个被扫描到的边界点），因此不妨从左下方向逆时针开始探查，如左下方的点就是黑点（有用的边界），则直接跟踪至边界点，否则探查方向逆时针旋转 45 度，直到找到第一个黑点为止。找到边界点后，在当探查方向的基础上顺时针回转 90 度（如果超过了左下角方向的起始方向，就

直接从左下角方向开始探查起)。不停用上述的探查方向，直到探查又回到初始的边界点 P_0 为止。

可以参照图 5 的示意图来分析：

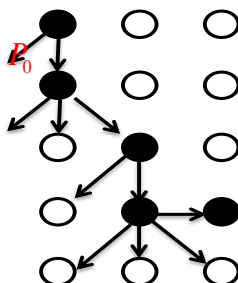


图 5 边界跟踪算法示意图

§2.2.3 多边形逼近算法

在获取目标物体的轮廓之后，在一定的误差准则下，通过对原始轮廓的采样，去除轮廓线的冗余点，以减少构成轮廓的点的数目，减少后续处理的计算工作量，同时要求能够准确表征目标轮廓的形状。

常用的轮廓采样方法有：等角度法、等步长法、曲率采样方法、Douglas-Peucker 方法。本文采用的是 Douglas-Peucker 方法^[8]，该方法是由 D.Douglas 和 T.Peucker 于 1973 年提出，简称 DP 算法，是公认的线状要素化简经典算法。它的优点是具有平移和旋转不变性，并且对于给定的曲线和阈值，采样结果一定。

下面就重点介绍一下 DP 算法：

DP 算法的几何原理是：曲线上任意一条弧都可用弧上的一条弦来做任意精度的逼近，并且弧上任意点到弧上弦的最大距离能够描述这种逼近的精确程。其基本思想是：预先给定距离容差，将曲线首、末两点连线，依次计算首尾两点之间的所有中间点到该线段的距离，找出距离最大的点，并判断该距离是否小于给定的距离容差，若是，则舍去所有中间点；若否，则保留该点，并以该点为界，将曲线分为两部分，然后对这两部分重复以上操作。当曲线被新找到的中间点分为两部分后，将这两部分分别作为独立曲线重复以前的步骤。

下图 6 描述了 DP 算法的思想：

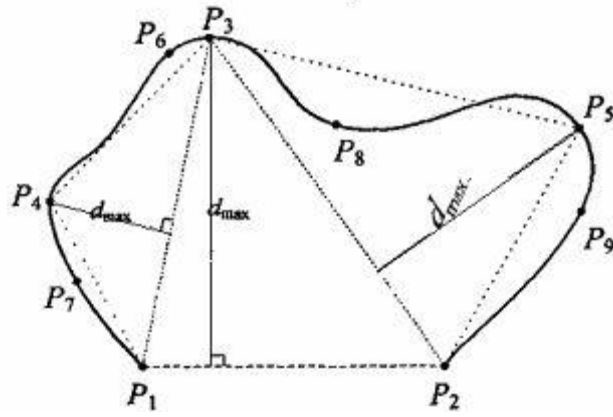


图 6 DP 算法示意图

设曲线的首末两点 P_1 和 P_2 构成直线段 $\overline{P_1P_2}$ ，计算曲线上这两点之间所有点到此直线段的距离，并找到距离最大值 $dist$ 及其对应点 P_3 ，将 $dist$ 与阈值 ε 作比较；若 $dist > \varepsilon$ 则保留 P_3 ，将该段曲线以 P_3 为界分为两段独立的曲线，重复上述步骤；若 $dist \leq \varepsilon$ ，则将首末两点之间曲线上所有的点全部舍去，用 $\overline{P_1P_2}$ 取代该段曲线。当 ε 越小，采样得到的点数越多，连接采样点构成的多边形曲线更加逼近原始曲线。

分析可知，DP 算法可以用递归的思想来设计程序，其算法描述如下：

```
void DPAlgorithm(Line){
    if (首末两点之间没有点)
        返回;
    else{
        for (Line 之间的所有点){
            计算各自的垂直距离;
            保存最大值的信息;
        }
        if (最大值小于阈值){
            舍弃中间所有点;
            返回;
        }
        else{
            最大距离点保存到结果数组中;
            以该点分界为两个部分, L1 和 L2;
```

```
        DPAlgorithm (L1);
        DPAlgorithm (L2);
    }
}
}
```

再回到我们要解决的问题上：车牌。

利用 DP 算法是要来寻找多边形的区域轮廓拟合，本文在计算时将 ε 定义为每个轮廓的外围周长的 $1/50$ 。然后再根据车牌的先验信息来判断车牌的最佳区域或者没有车牌的结论。

在用 DP 算法处理后，有如下几个原则来判断车牌的具体区域：

- 1) 对返回的结果中顶点的个数是 4 个的区域进行保存，否则不保存。
- 2) 对所有保存下的是四边形区域进行分析，计算其长宽比例（水平方向两个最值的差为长），符合长宽比在 2~5 之间的保留，否则去除。

- 3) 对二值化图像中车牌区域内的灰度跳变次数统计，在 5~15 之间的保留，否则去掉。具体做法如下：选择候选区域的中间三行进行水平扫描，计算通过这三行的灰度跳变平均值。经过研究，牌照区域可能出现的几十个汉字中，字符中间灰度跳变最大的是“渝”，可以达到 5，最小的是“宁”，为 1，第二、三个字符为英文字母或数字，其中字母“M”的灰度跳变次数最大为 3，字母“C”的灰度跳变次数最少为 1，数字“0”的灰度跳变次数最大为 2，数字“1”的灰度跳变次数最小为 1，所以得到车牌区域中间三行的灰度跳变次数最大为 14，最小为 7，同样，考虑到噪声的影响，噪声有可能增加灰度跳变次数，将值定在 5~15 之间。

- 4) 边缘检测后的图像中，将候选车牌区域图像的方差最小的作为最佳的区域。

在边缘检测后的图像中，由于车牌区域有一系列的排列均匀整齐的字符边缘，即字符骨架。所以，在理论上车牌区域图像的方差相对较小。

图 6 描述的是定位判别的流程图：

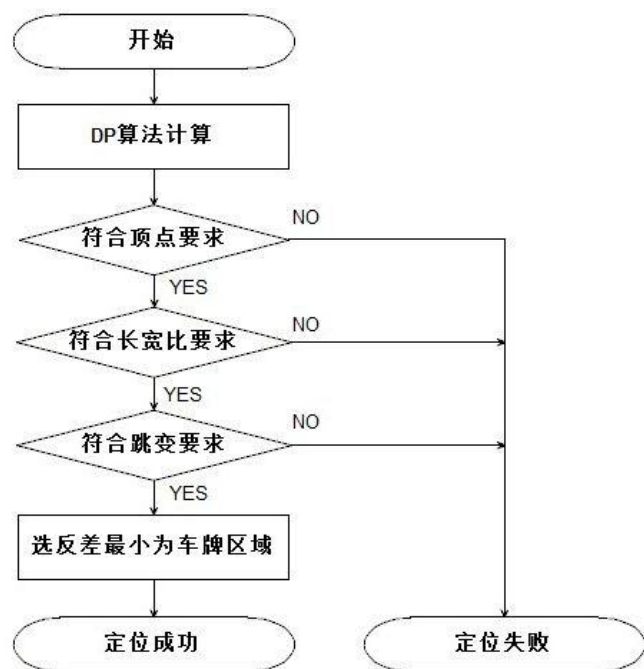


图 6 定位车牌判别的流程图

到此为止，整个定位车牌的算法就介绍完了。其整体的思想还是基于轮廓检测的。总结一下具体过程为：首先用 Canny 算子检测边缘，用边界提取算法提取具体的轮廓，用 DP 算法来多边形逼近，再根据车牌的特征，进行一下条件判断（图 3.5），最后给出结果。评定的效果如图 7 所示：



图 7 定位效果图

从识别出的图分析可知道，该算法对于光照和车牌的位置不一的情况都可以较好的识别，识别率较高，效果也不错，缺点时间复杂度较高，对一些算法中的（如 Canny 的阈值选择）参数还比较敏感。

§3 字符分割

§3.1 倾斜度校正

首先，将车牌区域以中点位置分为左右两个半边，计算左边的平均高度和右边的平均高度，计算公式可以表示为：

$$\theta = \arctan\left(\frac{h_{right} - h_{left}}{width / 2}\right) \quad (0.1)$$

其中 h_{right} 和 h_{left} 分别表示右边和左边的平均高度， $width$ 表示图像的宽度。按照计算出的角度，以中心点为原点进行旋转。



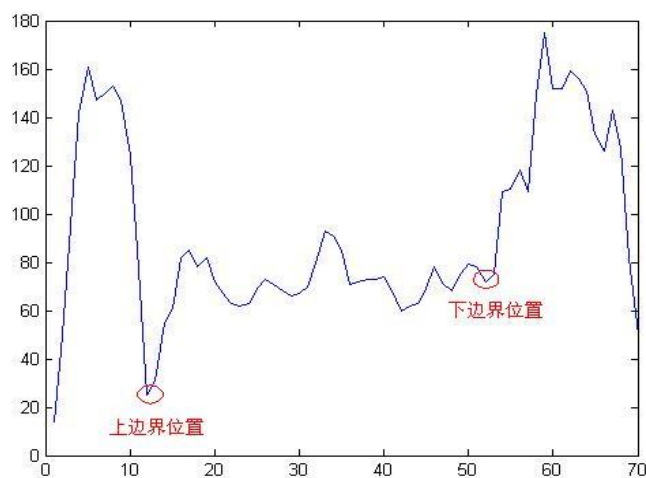
图 8 字符倾斜度调整

§3.2 考虑多种细节的改进投影分割算法

基于投影的字符分割算法，原理上比较简单，就是将车牌区域做垂直的投影，分析出白点比较少的位置作为分割的边界位置。但是在实际操作的时候发现并不是那么的简单。存在以下几个问题：

- 1) 车牌分割不到位会直接的影响到分割算法的效果
- 2) 车牌定位的时候，识别出来的车牌区域是带有车牌边框的。分割字符是不能有边框的，所以首先需要去除边框的影响。
- 3) 二值化图像后，车牌的第二和第三个字符之间有时会有“.”，做投影分析的时候就会导致分割受到影响。问题在于二值化之后，有时候该点存在，而有的时候是没有的，或者比较不明显，要对其做特别的处理。
- 4) 不论如何，得到的二值化的车牌区域图像不可能是很完美的图。肯定会存在一些干扰的点，噪声之类的。这些都增加了我们分割的难度。
- 5) 字符的大小长度不一。比较多的特殊的情况，如遇到如汉字“川”之类的情况，就会导致将“川”分割成三个部分。所以还要对分割出来的字符宽度加以限制，但是这个样子又会在分割数字“1”的时候出现问题，因为“1”的宽度肯定是不符合长度要求的，所以单一的限制宽度也不是好的办法。这些具体的细节问题都要在设计算法的时候去加以考虑。

首先解决的是如何去除边框的影响，先以一幅图的水平投影开始分析：



图中横坐标表示图像的高度坐标，纵坐标是对应的白点个数（白色为前景色，黑色为背景色）。分析可知，可以以点的个数小于给定的阈值来判定边界，问题是车边框有的是黑色有的是白色的，这个统一的阈值不好定。如图 4.2 是一个白色边框的车牌，则阈值就要给的比较高，但这样就会导致黑色车牌的分割不准确。所以不应该用一个给定的值作为判定准则，应该用局部最小值的思想来判断。

令 x_1 和 x_2 为边框的边界位置， $f(x)$ 为 x 位置的白点个数，则评定准则为：

$$\begin{cases} f(x_1) < f(x_1 + 1) \\ f(x_1) \leq f(x_1 - 1) \\ f(x_2) \leq f(x_2 + 1) \\ f(x_2) < f(x_2 - 1) \end{cases} \quad (0.2)$$

从上面开始扫描的第一个符合要求的 x_1 就是上面的边界位置，同理下面的第一个符合要求的 x_2 就是下边界位置。注意其中的 $<$ 和 \leq 的区别，因为一个边界之后可能还有一些的完全没有用的区域（边框线离实际字符最上边区域），通过不同的比较符合来区分。

同理利用垂直方向的投影来分割左右的边框线，原理同上。

分割之后的效果如图 9 所示：

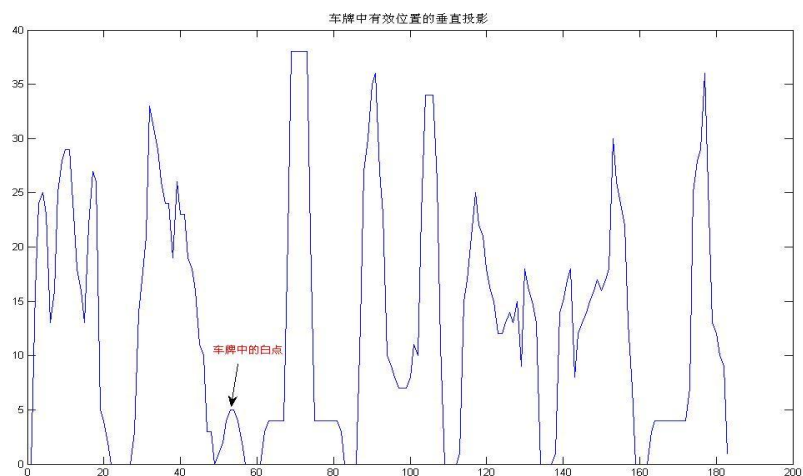


图 9 分割效果图

下面就来讨论如何具体的分割字符。

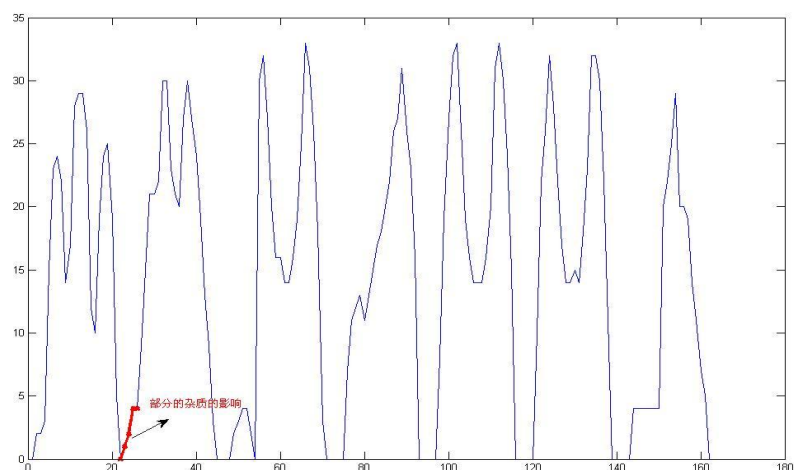
针对上文提出的几个难点，本文用以下的解决方法来解决：

1) 针对第二个和第三个字符之间出现的“.”，在分割完第二个字符之后，直接加上一个大致距离长度，这样子就可以跳过对这个点的分析。下图说明了点的影响：



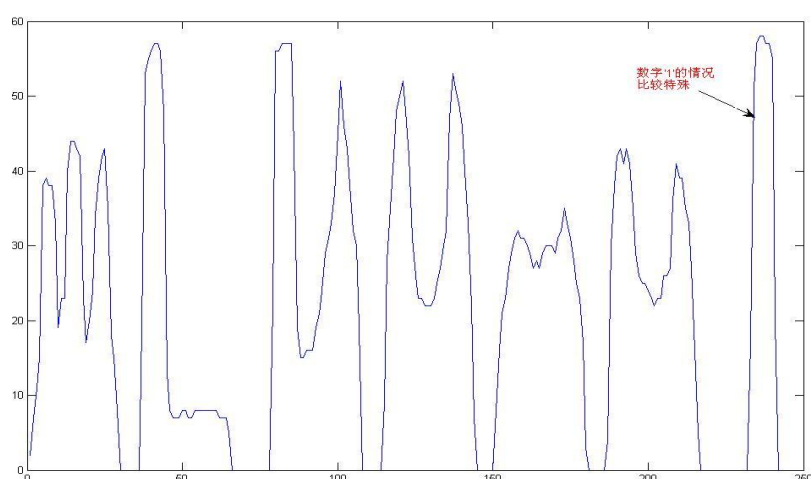
根据上面的图片和先验知识，给出加的距离为 $w/24$ （下文中用到的 w 和 h 分别代表车牌有效区域的宽和高）。

2) 分割的时候，由于噪声点的存在，如下图所示：



边界的评定条件也是用局部最小值来判断，这里不再重复。同时由于杂质的存在，还要加上对于 x 位置处 $\frac{h}{12}$ 的限制，本文给的参数是 $h/12$ 。

3) 在处理如汉字“川”的时候,就会导致分割失败,所以还需要加上分割的宽度限制。通过实际的测试,本文取 $w/8$ 作为分割宽度阈值。但是遇到数字“1”的时候又会失败(宽度肯定不符合要求),于是在检测出右边界同时宽度没有达到要求的同时,再加上对数字“1”的判断。本文的判断思想大致可以概述为:在分割的区域内,如果大部分的区域的 $f(x)$ 值都比较高,就认为是数字“1”。数字“1”的影响可以见下图:



总体的算法描述如下:

//分割车牌有效区域算法描述

```
leftBoundry = false;           //是否找到左边界
addFlag = false;               //是否加个点的位置长度
for (从左到右对投影的个数分析) {
    if (!leftBoundry && 符合边界要求) {
        leftBounday = true;
        记录左边界位置;
        continue;
    }
    if (leftBoundry && 符合边界要求){
        if (分割宽度 > w/8){
            leftBoundry = false;
            记录右边界位置;
            continue;
        }
    }
}
```

```

    }
    else if (满足为数字“1”的条件) {
        leftBoundry = false;
        记录右边界位置;
        continue;
    }else{                                     //就是可能出现汉字“川”的情况

        continue;
        //什么也不做，继续向前分析，直到找到符合宽度的右边界
    }
    if (!addFlag && 分割完了两个字符) { //处理二三字符间的点
        addFlag = true;                //只加一次
        将分析的位置 x 加上 w/24;      //x 表示目前循环到的坐标位置
    }
} //end if
} //end for

```

按照上面的算法分割出来的效果如图 10:



图 10 分割效果图

说明的是：由于是将二值化的图像进行倾斜校正，原来的彩色图并没有校正，所以看起来有些比较歪的车牌分割不是很好，彩色图片只是为了示意效果。

3 字符识别

§3.1 汉字的识别

由于汉字的字符结构复杂，不易于提取特征值，所以利用神经网络方法反而不易于识别。而若是从识别出来的汉字的整体来分析，由于识别出来的汉字形状和宽度各不相同，这些都会影响识别的效果。所以可以首先提取出汉字的骨架结构，它描述的是物体的几何形状和拓扑结构，用到的算法是细化算法。

§3.1.1 字符细化算法

所谓细化，就是从原来的图中去掉一些点，但仍要保持原来的形状。实际上，是保持原图的骨架。所谓骨架，可以理解为图象的中轴，例如一个长方形的骨架是它的长方向上的中轴线；正方形的骨架是它的中心点；圆的骨架是它的圆心，直线的骨架是它自身，孤立点的骨架也是自身。

经典的细化算法有：Hilditch 算法，Zhang 快速并行算法，和 Deutsch 算法。Hilditch 算法是串行的处理方式，效果明显，但是速度较慢。后面的两种算法都是并行的算法，提高了速度。

下面就主要介绍一下 Zhang 快速并行细化算法的实现思想^[8]：

Zhang 快速并行细化算法采用两层子循环，第一个子循环中，目标像素删除的条件是：

1) $2 \leq B(P) \leq 6$

2) $A(P) = 1$

3) $P_0 * P_2 * P_6 = 0$

4) $P_0 * P_4 * P_6 = 0$

P_3	P_2	P_1
P_4	P	P_0
P_5	P_6	P_7

其中 $B(P)$ 是 $(P_0 \cdots P_7)$ 中的非 0 像素数， $A(P)$ 是有序集 $(P_0 \cdots P_7)$ 中的 01 模式数。

在第二个子循环中，把上述条件中的(3)，(4)改为：

$$3) \quad P_0 * P_2 * P_4 = 0$$

$$4) \quad P_2 * P_4 * P_6 = 0$$

若条件成立，则目标像素 P 为可删除像素。该算法得到的骨架形态是 8 连接的。：



图 11 细化算法效果

可以看出细化算法在去除了图像很多不需要的信息的同时，也保留了字符的主要结构特征。

§3.1.2 基于模板匹配的算法

模板匹配是图像识别中最具代表性的方法之一。它是将待识别的图像中提取信息和模板进行匹配，计算图像和模板特征量之间的距离，用最小距离法来判定所属类。

当然要首先产生各个汉字字符细化后的模板库，本文将所有的分割字符都调整到统一的大小（ 32×64 ），这样子便于分析和处理。将待识别的图像和所有的模板相比较，计算其距离 $dist(i, j)$ ，其中 i 表示待识别图像的编号， j 表示模板的编号。

$dist(i, j)$ 的计算方法可以描述为：

- 1) 将待识别图像从上到下，从左到右，逐个扫描像素，直到找到为灰度值为 255 的字符（前景色，下文都用白点代替），记录下该点的位置 loc_i 。

- 2) 在模板图像中，在以 loc_i 位置为中心的 5×5 领域中，搜索距离 loc_i 最近位置的白点，并且记录下两点之间的距离 $d(m, j)$ ，其中 m 为待识别图像中白点的编号。如果领域内不存在白点，则用一个较大值代替 $d(m, j)$ 。

-
- 3) 返回步骤一，直到待识别图像中所有白点分析完为止。
 - 4) 计算每个待识别图像 i 与模板图像 j 的距离 $dist(i, j)$ ：

$$dist(i, j) = \sum_{m=1}^{N_i} d(m, j)$$

其中， N_i 表示图像 i 的白点总数。

- 5) 对于每个图像 i ，匹配的结果就是最小的距离：

$$result_i = \min_j (dist(i, j))$$

其中， $result_i$ 表示的值就是图像 i 的匹配结果。

- 6) 同时定义置信度 $confidence$ 参数，用来表述匹配的准确程度，令 m 和 mm 分别表示 $dist(i, j)$ 中最小和次小的两个值，则：

$$dconfidence = (mm - m) / m \times 100\%$$

§3.2 数字和字母的识别

基于模板匹配的算法比较的简单，速度较快，但是识别的效果一般。而相比于汉字，一般的数字和字母则结构简单，特征值易于提取，所以可以考虑用神经网络的方法来识别。

§3.2.1 基于神经网络的识别算法

本算法通过下面的一些改进方法，可以有效的降低 BP 搜索停留在局部极小值的概率。

- 1) 将网络权值初始化为接近于 0 的小随机值。

注意到 Sigmoid 函数在输入接近 0 的时候接近线性。如果把网络权值初始化为接近于 0 的值，则作为 Sigmoid 单元的净输入的 net 也必然接近于 0，因此在早期的梯度下降步骤中，网络布线为一个平滑的函数，近似为输入的线性函数，基本不存在局部极值的问题。当训练进行一定时间后，随着权值的增长，网络演变

为可表示高度非线性的函数，从而开始出现更多的局部最小值，但一般情况下此时搜索已经足够接近全局最小值，即便是这个区域的最小值也是可以接受的。

2) 增加冲量项。

就是在计算权值更新的时候，一部分更新值依赖于上一次迭代时候更新的值。一般选定一个常数 $\alpha (0 < \alpha < 1)$ 与上一次更新值相乘加到本次更新的权值上，这个系数 α 叫做冲量常数。

冲量项有时可以带动梯度下降搜索冲过狭窄的局部极小值而不是陷入其中。设想一个球沿误差曲面向下滚， α 的作用就是使这个球从一次迭代到下一次迭代时以同样的方向滚动。冲量有时会使这个球滚动过误差曲面的局部最小值或平坦区域。同时，冲量项还具有在梯度不变的区域逐渐增大搜索步长，从而可以加快收敛的速度。

3) 使用随机的梯度下降代替真正的梯度下降。

采用梯度下降的随机近似对于训练样本沿一个不同的误差曲面有效的下降，用 这些梯度的平均来近似为整个训练集合的梯度。这些不同的误差曲面通常有不同的局部极小值，这使得下降过程不太可能陷入某一个局部极小值

在通过上面的步骤优化 BP 算法后，可以较快较准确的收敛。

最后，再利用模板图像更新权值后，就可以用分割出来的样本来进行识别了。在这里说明一下置信度 *confidence* 的定义：

$$confidence = (\max_1 - \max_2) / \max_1 \times 100\%$$

其中， \max_1 和 \max_2 分别表示的是输出值最大的两个值，注意这里是越大说明越相似，当然值最大的为给出的类别结果，参数 *confidence* 只是为了说明可靠的程度，如果 \max_1 和 \max_2 值很相近，那么 *confidence* 就较小，也就是说明识别结果不是很可靠。最终的识别效果见图 12：



图 12 识别最终效果图

结果分析:

汉字的识别都比较的准确，一般的字符也大部分都可以准确识别。而且利用了神经网络的算法，可以说对于一些模糊的评定也是比较有效的。同时可以看出，本文中强调的置信度参数可以有效的说明识别效果的准确度，较高的置信度则比较准确，如果出现较低的置信度可以作为是否需要后期处理的评定依据。

同时也存在的一些问题，有的字母还是识别效果不好，如字母“Q”和数字“0”确实比较的相似，不容易识别准确，置信度仅为11%。在程序中，可以利用置信度来进行后期的修改，如果置信度都是大于40%，则可以直接将结果保存，如果有存在置信度小于40%的情况，就提醒用户来后期处理，以便修改错误，保证保存结果的准确性，就是人工控制，原则就是保证保存下来进入数据库的结果一定是尽量正确的。

参考文献

- [1] R. Mullot, C. Oliver, J. L. Bourdon. Automatic extraction methods of container identity number and registration plates of cars[C]. Proc IECON' 91, International Conference on Industrial Electronics, Control and Instrumentation, Japan, 1991, 3: 1739-1745.
- [2] J. Bulas-Cruz. Number plate reading using computer vision[R]. ISIE' 97, Proceedings of the IEEE International Symposium on Industrial Electronics, 1997.
- [3] D.W. Tindall. Deployment of automatic licence plate recognition systems in multinational environments[J]. European Conference on Security and detection, 1997, 10: 42-46.
- [4] 郭大波, 陈礼民, 卢朝阳等. 基于车牌底色识别的车牌定位方法[J]. 计算机工程与设计, 2003, 24(5): 81-87.
- [5] 陈寅鹏, 丁晓青. 复杂车辆图像中的车牌定位与字符分割方法[J]. 红外与激光工程, 2004, 33(1): 29-33.
- [6] Deb.k, Kang-Hyun Jo. HSI color based vehicle license plate detection[C]. ICCAS 2008, International Conference on Automation Control and Systems 2008, 2008: 687-691.
- [7] 卢雅琴, 乌仔凌超. 基于数学形态学的车牌定位方法[J]. 计算机工程, 2005, 31(3): 224-226.
- [8] 米江, 纪国宜. 改进的 BP 神经网络在风机故障诊断中的应用, 噪声与振动控制期刊, 2011 年 4 月 18 日.
- [9] "Introduction to artificial neural networks", Electronic Technology Directions to the Year 2000, 1995. Proceedings, vol., no., pp. 36-62, 23-25 May 1995
- [10] Conci, A.; de Carvalho, J.E.R.; Rauber, T.W.; , "A complete system for vehicle plate localization, segmentation and recognition in real life scene.," Latin America Transactions, IEEE (Revista IEEE America Latina) , vol. 7, no. 5, pp. 497-506, Sept. 2009