车牌识别大作业报告

周驿东 1900012909

December 17, 2019

目录

1 车牌分割

1.1 预处理

1.1.1 OTSU 算法

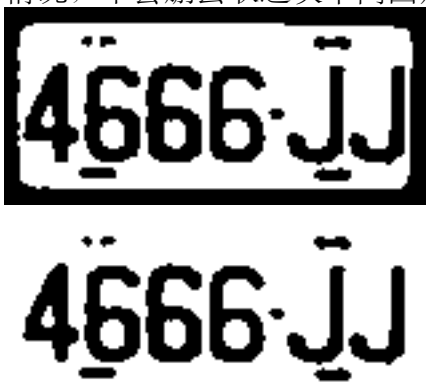
本文用 OTSU 算法进行图片的二值化。OTSU 算法会最大化类间方差。
设 a_i 表示图片中灰度为 i 的点的数量，x 表示二值化阀值。
背景像素占比 $\omega_1 = \frac{\sum_{i<x} a_i}{\sum_i a_i}$
前景像素占比 $\omega_2 = \frac{\sum_{i>=x} a_i}{\sum_i a_i}$
背景的平均灰度 $\mu_1 = \frac{\sum_{i<x} a_i i}{\sum_{i<x} a_i}$
前景的平均灰度 $\mu_2 = \frac{\sum_{i\geq x} a_i i}{\sum_{i\geq x} a_i}$
总平均灰度 $\mu = \frac{\sum_i a_i i}{\sum_i a_i}$
类间方差 $g = \omega_1(\mu - \mu_1)^2 + \omega_2(\mu - \mu_2)^2 = \omega_1\omega_2(\mu_1 - \mu_2)^2$
其中 x 是变量。最大化 g，将此时取到的 x 作为二值化阀值。
OTSU 算法可以避免图片明暗程度造成的影响。
实际代码中，因为分割时经常出现字符连在一起的情况，所以略微下调二值化阀值以减轻这种情况。

1.1.2 开运算

开运算是形态学操作，先腐蚀后膨胀。对图片进行开运算，同样是为了减轻字符连在一起的情况。

1.1.3 车牌边框的处理

给出的车牌的 bounding box 坐标基本上都包含车牌边框。如果一个在图片边界附近的联通块在宽度或高度上几乎跨越了整个图片，就认为这个联通块是边框，并把它去掉。为了避免边框和字符相连而把字符也一起去掉的情况，不会删去联通块中离图片边界过远的部分。



1.1.4 噪点的处理

图片中会出现一些较小的噪点，包括车牌中的“·”。考虑到字符的高度占整个图片高度的比例都较大，将所有高度占整个图片不到 40% 的联通块都认为是噪点，并把它去掉。

1.2 二值化后的分割方法

1.2.1 方法一

把每一列的黑色像素的比例称为前景比例。

如课件中所说，利用前景比例来判断一个字符的开始和结束。

由于会出现字符连在一起的情况，而所有图片的阈值只能设成一样，实际效果并不好。

1.2.2 方法二

经过预处理，理想的情况是图片上没有边框和噪点，只剩下 6 个字符，每个字符为一个联通块。方法二就是将图片上每个联通块看作一个字符。由于实际情况并不一定那么理想，可能会出现连续多个字符连在一起的情况。单个字符的宽高比接近 $1:2$，因此我借助宽高比来判断联通块中包含了几个字符，并把联通块相应等分。

方法二的问题有:

- 基于宽高比的判断不一定准确
- 即使联通块中的字符数判断准确，等分也未必能恰好分出每个字符
- 一个字符可能由多个联通块组成
- 极易受到噪点的影响

1.2.3 方法三

如果某一列的前景比例低于一定比例（代码中是 30%），就认为这一列可能是一条分割线。我们的目标是将整个图片分割成 6 个字符，因此需要找到 5 条分割线。为了防止 5 条分割线靠得过近，我们试图最大化 x，使得我们能找到 5 条互相之间距离至少为 x 的分割线。这可以通过二分完成。把 x_{max} 下找到的 5 条分割线作为方法三的结果。

方法三的问题有:

- 不能处理斜的车牌
- 所有前景比例低于 30% 的列都被认为是可以分割的，导致有时会把 J 这种字符分成两半，而不去分割空白列
- 如果出现字符连在一起的情况，或者有 J 这种字符时，某些字符相连的部分的前景比例甚至高于字符内部的前景比例，难以正确分割

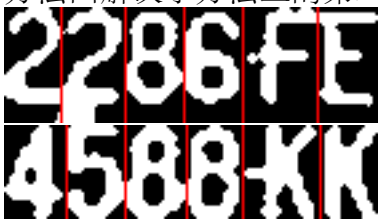
1.2.4 方法四

我们想要找到 5 条分割线，最小化它们的前景比例的平均值。为了不让它们靠得过近，设置一个 x，分割线之间距离至少为 x。代码中 x 取整张图片宽度的八分之一。

这可以通过动态规划解决。$a[i]$ 表示第 i 列的前景比例，$f[i][j]$ 表示前 i 列，已有 j 条分割线时前景比例的和的最小值。转移方程：

$$f[i][j] = min_{k<i-x}\{f[k][j-1]\} + a[i]$$

方法四解决了方法三的第二个问题，一定程度上减轻了第三个问题。



1.2.5 实际采用的方法

由于二值化效果较理想时，方法二比较准确，因此分割时先使用方法二，如果分割出的字符数不对，再使用方法三或方法四。

2 车牌识别

2.1 字符识别

2.1.1 训练数据

将车牌训练集中的车牌分割后作为训练数据，不使用 Chars_data 中的数据。

2.1.2 神经网络

由于需要分成 34 类（不包括 O 和 I），大大多于 LeNet5 原本的 10 类，因此把 LeNet5 扩大作为用于字符识别的网络。扩大后卷积层的 channel 数依次为 $1, 15, 100, 400$，全连接层的大小依次为 $400 \times 240, 240 \times 34$。

由于车牌的字体都比较标准，单个字符识别的准确率主要取决于字符分割的准确率，对网络进行其他优化或者进行数据增强的效果似乎不太明显。

2.2 几种分割方法的车牌识别准确率

方法二	56%
方法三	53%
方法四	73%
方法二 + 方法三	68%
方法二 + 方法四	84%

3 一些问题和可能的思路

1. 分割出错可能导致训练数据中混入错误数据，降低单个字符识别的准确率。而由于车牌字体比较标准，不需要太多的训练数据就能取得比较好的识别效果，因此也许可以人工筛选合适的训练数据。

2. 难以处理斜的车牌。方法二可以处理斜的车牌，但对二值化效果的要求比较高。Kmeans 和 Meanshift 等方法应当能处理这个问题。

4 参考文献

[1]. https://blog.csdn.net/u012198575/article/details/81128799