

# 基于OpenCV的车牌识别

毕波,邵永谦,孙冬军,贾思超

(上海市地震局 上海 200062)

**摘要:** 针对车牌检测中关键的3个环节分别进行了改进和优化。利用数学形态学结合Canny算子实现车牌定位。在传统扫描字符算法中加入边限定,提高字符分割的准确度。在识别环节,采用计算效率较高的KNN算法进行字符的识别。最终,在OpenCV平台上实现车牌识别。

**关键词:** OpenCV; 车牌定位; 边缘检测; 字符分割; 图像识别; K阶最邻近

中图分类号: TN911.73

文献标识码: A

文章编号: 1674-6236(2019)01-0037-05

## License plate recognition based on OpenCV

BI Bo, SHAO Yong-qian, SUN Dong-jun, JIA Si-chao

(China Shanghai Earthquake Agency, Shanghai 200062, China)

**Abstract:** The three links are improved and optimized respectively in the field of license plate detection. Vehicle license plate location is realized by using mathematical morphology and Canny operator. Adding boundary restriction in traditional scanning character algorithm also improves accuracy of character segmentation. In the stage of character recognition, KNN is adopted due to high computational efficiency. All these processes are implemented on the OpenCV platform.

**Key words:** OpenCV; vehicle license plate location; edge detection; character segmentation; image recognition; K-Nearest Neighbor

车牌识别技术最早起源于国外,而国外所需要识别的字符与国内稍有不同,而且车牌的样式与格式也有一定的差别,因而如果直接将国外的技术应用过来,效果并不是很好<sup>[1-2]</sup>。随着图像处理技术的不断发展,越来越强大的工具支撑着科研问题的深入,OpenCV就是这样一款工具。它是开源、跨平台的计算机视觉库,可用于各种图像和视频处理操作<sup>[3-7]</sup>。虽然OpenCV是一个C++库,但是在C++代码基础上的Python接口也越来越完善。目前并不是所有的OpenCV组件都提供Python接口,但是其背后活跃的开发社区,使得python与OpenCV结合的前景越来越广阔。本文采用OpenCV与python相结合的开发技术,探讨车牌识别的算法和技术流程。

## 1 基于边缘检测-数学形态学方法的车牌定位实现

在我国,车牌的形状有440×140、440×220、

220×140、480×140 4种。在筛选图片中的区域时,可以用一定的长宽比例范围来选定。从颜色来看,有“黄底黑字,黑框线”、“蓝底白字,白框线”、“白底黑字,黑框线”、“黑底白字,白框线”、“渐变绿底黑字,黑框线”、“黄绿双拼底黑字、黑框线”这六类。由于车牌都有一定的边框,而且颜色区分度较高,能够很好地把车牌与背景分离,因而也就有了利用这一性质进行车牌定位的方法<sup>[8-9]</sup>。本次实验在高斯低通滤波的基础上采用canny算子进行车牌定位。

### 1.1 基于高斯滤波的平滑处理

滤波是将信号中特定波段频率滤出的操作,是抑制和防止干扰的一项重要措施。在计算机视觉中,常常利用滤波如高斯滤波来对图像进行处理,当然,为了提升运算速度,也会直接使用奇数阶的方阵以用于对图像进行卷积运算。具体的操作就是对于图像的每一个像素点,计算他的邻域像素和滤波器矩阵的对应元素的乘积,之后加起来即可,

收稿日期:2018-04-10 稿件编号:201804088

基金项目:国家地震局测震台网青年骨干培养专项(20150409);上海市科委项目(14231202602)

作者简介:毕波(1977—),男,上海人,工程师。研究方向:人工智能、数字图像识别及地震观测系统开发。

作为该像素点的值。通过这一操作,将灰度图像执行了平滑在操作,如图1所示。



图1 原始图像与平滑图像

## 1.2 基于Canny算子的边缘检测

边缘就是图像中局部区域颜色变化较大或者亮度突然变化的区域。这一处理是获取汽车轮廓信息的重要技术手段<sup>[10]</sup>。由于车牌的具体特征——车牌有较为明显的边框线,并且与汽车的背景有较大的颜色差异,如果提取效果较好的话,会在车牌附近得到较为密集的线条分布。

基于OpenCV的边缘检测有5个步骤,即:使用高斯滤波器对图像进行去噪;计算梯度;在边缘上使用非最大抑制(NMS);在检测到的边缘上使用双阈值以去除假阳性;最后分析所有的边缘及其之间的连接,以保证真正的边缘并消除不明显的边缘检测结果,如图2所示。



图2 边缘检测结果

## 1.3 基于OTSU的二值化处理

OTSU算法常用于分割图片的前景与背景。由于一般的图片会有前景与背景之分,一般前景会比较明亮,背景较为昏暗,要想将其前景与背景分离,

就需要找到合适的灰度值即阈值,把两部分图像通过二值化分割出来,即背景为黑色,前景为白色。

有一点需要提出,OTSU算法对于灰度统计直方图为双峰的图像效果会好些,计算所得阈值能够更好的地分割前景与背景。

实际试验时,采用均值滤波对边缘检测后的结果进行滤波处理。通过多次试验分析,选择3x9的内核来处理效果最好。二值化处理之后,边缘线条会有加粗的效果,利于腐蚀膨胀操作,如图3所示。



图3 二值化的轮廓图

## 1.4 数学形态学算法定位

数学形态学是一种基于集合的滤波器。这一环节的关键是找好结构元素。在本次实验中选取22x7的矩形作为结构元素,其长宽比例与要识别的车牌是相同的。相应的通过腐蚀之后再膨胀运算能够使得最终的图形更加接近于实际的车牌形状。为了通过数学形态学运算构造连通区域,需要先对二值化之后的图像进行闭运算,填补类似于车牌中字符之间的凹槽,如图4所示。



图4 形态学处理结果

图4中明显还留有一些多余的线条。通过多次的腐蚀操作(具体操作次数需要依实验而定),可以去除白色的点噪声,如图5所示。

根据图5抓取的车牌位置,在原始图像上用矩形框圈出,如图6所示。

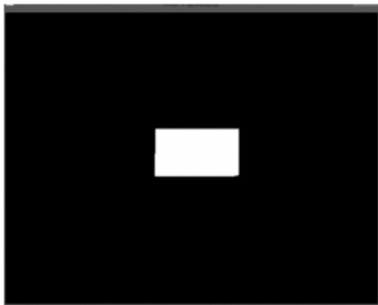


图5 车牌位置



图6 车牌范围

## 2 车牌字符分割

经过车牌定位之后,如果成功的话,我们会得到含有车牌的小范围图片。直接对其进行分割处理,往往会出现分割失败的情况。由于诸多不可抗的因素类似于车牌有磨损,车牌表面有泥点,由于光照因素,车牌表面会有反光现象以及最重要的一点,由于拍摄角度问题,车牌会呈现平行四边形或者车牌整体呈现一定的倾斜角度,导致分割时会出现问题<sup>[11-14]</sup>。当然,车牌本身由于有分隔符以及铆钉等影响,也会出现分割不准确的结果。如果不好好处理,将会影响下一步的识别正确率。因而,在分割之前需要对车牌进行一定的预处理。

### 2.1 字符分割预处理之图像旋转矫正

为了避免车牌图片倾斜对之后的扫描以及识别操作造成影响,需要对车牌图片进行角度矫正。由于车牌边框会对字符分割产生或多或少的影响,去掉车牌边框,进一步缩小分割的范围,可以提高分割的成功率。

由于定位之后,车牌图片中还有车牌边框的存在,所以可以利用这一直线的倾斜度作为车牌的倾斜度。运用 Hough 变换进行直线检测,之后计算检测到的前3个最长直线与 $x$ 轴夹角的平均值,得到整张图片的倾角。最后,只需要利用 OpenCV 提供的仿

射变换函数进行平移,旋转放大即可,本次实验计算的倾斜度为 $-1.6667$ ,如图7所示。



图7 车牌校正

### 2.2 字符分割预处理之车牌去边框处理

去边框处理的思路比较简单,就是统计并比较这一行(列)中白色像素点数,再将其余下一行(列)中白色像素点数进行比较。由于是直接基于车牌像素点的值进行定界,车牌反光以及磨损等因素会对结果造成影响,因而比例值的确定至关重要。对于有些车牌,分割情况较好,有些则不是太好,之后会进一步进行研究。去边框后,如图8所示。



图8 去边框的车牌

### 2.3 基于改进的扫描字符分割算法实现

与传统的扫描算法相比,本次实验加了一个判定条件来确定字符左右边界,具体算法如下(其中 $src[]$ 为经过去边框处理后的车牌):

1)经过去边框处理之后,计算去边框之后的车牌尺寸,宽高分别为 $w, h$ ,设定一个标志位 $flag$ 并初始化为 $False$ 。定义两个列表, $Lborder[]$ 、 $Rborder[]$ ,分别用于存储各字符的左右边界。定义两个阈值 $threshlod1=5$ 、 $threshlod2=15$ (分别对应于字符开始结束和字符宽度的阈值)。进入步骤2)。

2)进入循环,循环变量 $x$ 最初从0开始,每循环一次加一,循环截止条件为 $x=w$ ,初始化统计变量 $count=0$ ,进入步骤3);循环结束时,进入步骤7)。

3)纵向扫描第 $x$ 列白色像素点的数目,即如果 $src[y, x]=255$ , $count$ 加1。进入步骤4);

4)如果满足条件 $not\ flag=true$ 并且 $count>threshlod1$ ,则字符边界确定下来,令 $xx1=x$ ,并将左边界值加入到列表 $Lborder[]$ 中,置 $flag=true$ ,进入步骤5),否则直接进入步骤5)。

5)如果满足条件 $flag=true$ 并且 $count<threshlod1$ ,则进入步骤6),否则返回步骤2)。



6)如果满足条件  $x-xx1>threshlod2$ ,则表明找到了合适的右边界,令  $xx2=x$ ,并将  $xx2$  的值加入到右边界列表  $Rborder[]$ 中,置标志位  $flag=False$ ,回到步骤2)。

7)如果  $len(Lborder[])>len(Rborder[])$ ,就把  $w-1$  加入到右边界列表中(防止没有检测到最后一个字符的右边界)。进入步骤8)。

8)分割字符,每个字符的为  $src[:, Lborder[i]:Rborder[i]]$ 归一化字符大小为宽40,高20。为防止归一化过程中一些像素值的改变,对每一个字符分割后的图片进行二值化处理。分割结束。分割结果,如图9所示。



图9 字符分割结果

### 3 车牌号码识别

车牌字符一般就3类,分别为阿拉伯数字、汉字、英文字母。其中,具体来讲,数字取0~9,英文字母取A~Z(其中因为1和字母I、0和字母O难以区分,所以国家规定O和I不能使用)。汉字的取值范围较广,除专用号牌外,一般字符为发牌机关所在省、自治区、直辖市的简称,如“京”,“豫”等一共31个汉字。对于专用号牌有特殊汉字如警用号牌简称为“警”,具体为{“领”“使”“警”“学”“挂”“港”“澳”“试”“超”}9个汉字。总的来说,就是需要识别74个字符。所以车牌字符识别的目的就是利用算法对于分割后的字符图片,能够将图片中的字符正确的对应到这74个字符<sup>[15-17]</sup>。本文采用KNN算法实现。

KNN全称为K-NearestNeighbor,正如其字面意思,就是K阶最近邻匹配。实质上是待检测对象的类别可以由与他最接近的 $k$ 个对象中最多归属的那一类决定。OpenCV函数库已经包含有KNN实现的函数,只需要直接调用即可。

具体操作如下:

首先,需要对训练数据进行处理。我们收集到的训练图片有0~9,A~Z(O和I除外),每一个字符都有25张图片。

1)定义一个列表labels作为标签,定义一个列表samples用于存储标准化后的训练数据。定义一个计数变量  $i=0$ ,定义一个存取字符范围的列表alphabet。定义一个列表img\_path用于存储所有图

片的路径。进入步骤2);

2)进入循环。逐条读取存在列表img\_path的路径,循环的结束条件为所有路径都读取完毕。循环中的操作为:读取当前循环是路径下的图片,用img存储。依次对img进行灰度转化,高斯平滑化处理,阈值分割,对大小进行归一化处理,归一化为40行,20列,最后把归一化的结果转化为1行,800列的向量,把它加入到列表samples中,相应的,把  $i/25$  的整数部分加入到标签列表labels中。最后做  $i=i+1$ ;循环结束后进入步骤3);

3)把列表labels和samples标准化为二维数组,再将列表labels转化为一个列向量。并将转化后的数组labels和samples存储为两个文件。自此,训练数据准备完毕。进入步骤4);

4)同样的,对待识别的图片我们也进行一样的操作。即定义一个列表samples[],用img\_path用于存储待识别图片的所有路径。接下来执行步骤2),3)类似的操作,最后得到存

有测试数据的文件samples\_test.npy文件。进入步骤5);

5)定义  $train\_label=labels[:850]$  ( $850=25*34$ ),  $train\_input[:850]$ ,  $test\_input=samples\_test$ ,加载之前生成的3个文件,构造KNN训练模型,以train\_input作为模型的训练样本输入,train\_label作为对应标签。将test\_input作为输入,设定  $k=10$  ( $k$ 值依据实验而定,这是多次比较后的结果。通常  $k$  的值是运用交叉检验方法确定(以  $k=1$  为标准),但是为避免麻烦,一般会选取低于总训练量的平方根的值作为  $k$  值)。最后得到对应于输入图片的经运算后的标签数组。进入步骤6);

6)输出  $alphabet[int(string[n])]$ ,其中  $string[n]$  为标签(要转化为整形值)。

识别的结果,如图10所示。

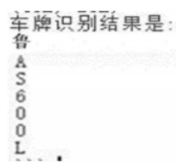


图10 车牌识别结果

利用KNN识别的好处是它不需要训练,每次进行识别操作时,都是直接调用数据文件,然后进行运算即可。不同于神经网络,神经网络需要先对大量的图片进行处理,得到合适的权值数据才行,而且想

让结果收敛是一个漫长的过程,计算量会比较大。相比之下,KNN的执行效率则高了很多,满足了识别的实时性的要求。但是,与之相对应的,k值的选择是一个比较麻烦的问题。

## 4 结束语

在OpenCV和python的协同作用下,车牌识别研究更加便捷,每个技术环节也越来越简洁与规范,这为今后的深入研究提供强大的平台与技术。在整个研究过程中有如下问题值得今后进一步探讨。

在车牌定位时,当遇到整张图片较为昏暗或者转化为灰度图像后区分度不高等情况时,使用OTSU算法,就会出现二值化后车牌区域值为0,即为黑色,导致车牌定位失败。此外,由于车牌定位主要基于数学形态学运算,运用腐蚀膨胀操作,如果图片中也有二值化后白色像素点比较密集的地方,可能会被误判为车牌区域。

当字符分割这一环节,如果车牌出现反光现象,即二值化后车牌字符处被大量噪点覆盖,就会出现字符分割失败。如果图片中噪点过多,将会对分割甚至是识别造成很大的影响。利用KNN算法进行车牌识别时,k值的选择会影响识别的结果。同时KNN的结果也会受到每个样本容量的影响。所以要尽可能的保持每个分类的样本容量要比较接近。当然,相对于传统的KNN算法,也可以考虑把距离作为权值进行计算,即距离近的权值大,最后求每个分类的总和来确定。

### 参考文献:

- [1] 宁彬.图像处理技术在机动车车牌自动识别技术中的应用[J].科学技术与工程,2013,13(2):366-371.
- [2] 蒋玲玲,杨冬,张友能.改进特征的车牌识别算法[J].安庆师范大学学报:自然科学版,2017,23(2):63-65,75.
- [3] 周宇.基于OpenCV-Python的图像分割技术的设计和应用[J].电子世界,2018(3):116-117.
- [4] 徐欢,李振壁,姜媛媛,等.基于OpenCV和改进Canny算子的路面裂缝检测[J].计算机工程与设计,2014,35(12):4254-4258.
- [5] 雷建锋,汪伟.基于OpenCV的图像阈值分割研究与实现[J].现代电子技术,2013,36(24):73-76.
- [6] 苏荻,刘晓东.基于OpenCV的智能视频监控告警系统的研究[J].电子设计工程,2016,24(3):26-28,36.
- [7] César Domínguez, Jónathan Heras, Vico Pascual. IJ-Open CV:Combining image J and Open CV for processing images in biomedicine. Computers in Biology and Medicine [J]. 2017(84):189-194.
- [8] 薛倩.基于图像预处理和纹理特征的车牌定位算法[J].电子设计工程,2014,22(16):159-162.
- [9] 马飞.一种基于小波变换和Tophat变换的车牌定位算法[J].电子设计工程,2016,24(22):118-121.
- [10] 薛岚燕,刘杜鹃,陈艺慧.基于Canny边缘检测和外观特征的微血管瘤检测方法[J].现代电子技术,2017,40(6):103-108.
- [11] 封雪,李东新.基于数字图像处理的车牌字符分割方法[J].电子设计工程,2014,22(22):185-187.
- [12] Tejendra Panchal, Hetal Patel, Ami Panchal. License plate detection using harris corner and character segmentation by integrated approach from an image[J]. Procedia Computer Science, 2016, 79, 419-425.
- [13] Jorge Eduardo Ramírez Flores, Antonio Susín Sánchez. Segmentation-based skinning[J]. Computer Animation and Virtual Worlds, 2017, 28(1): 123-126.
- [14] Antoine Bossard. Automatic segmentation of chinese characters as wire-frame models[J]. Procedia Computer Science, 2017(108):91-98.
- [15] 楚岩,邵严,陈亮,巨永锋.一种车牌识别软件系统研究[J].电子设计工程,2013,21(4):149-151, 155.
- [16] 穆日磊.差分进化算法和神经网络的车牌自动识别模型[J].现代电子技术,2017,40(15):183-186.
- [17] 栾众楷,徐文涛,刘冰.基于安卓平台的车牌识别系统的设计与实现[J].电子设计工程,2016,24(9):176-179.