

基于OpenCV的车牌检测系统

【PDF报告-大学生版】

报告编号: ae0803f4f3747365

检测时间: 2018-05-25 09:59:03

检测字数: 13,028字

作者名称: 佚名

所属单位: 西华大学

检测范围:

- ◎ 中文科技期刊论文全文数据库
- ◎ 中文主要报纸全文数据库
- ◎ 中国专利特色数据库
- ◎ 博士/硕士学位论文全文数据库
- ◎ 中国主要会议论文特色数据库
- ◎ 港澳台文献资源
- ◎ 外文特色文献数据全库
- ◎ 维普优先出版论文全文数据库
- ◎ 互联网数据资源/互联网文档资源
- ◎ 高校自建资源库
- ◎ 图书资源
- ◎ 古籍文献资源
- ◎ 个人自建资源库
- ◎ 年鉴资源
- ◎ IPUB原创作品

时间范围: 1989-01-01至2018-05-25

检测结论:

- 全文总相似比: 20.02% (总相似比=复写率+他引率+自引率)
- 自写率: 79.98% (原创内容占全文的比重)
- 复写率: 19.43% (相似或疑似重复内容占全文的比重, 含专业用语)
- 他引率: 0.59% (引用他人的部分占全文的比重, 请正确标注引用)
- 自引率: 0% (引用自己已发表部分占全文的比重, 请正确标注引用)
- 专业用语: 0.00% (公式定理、法律条文、行业用语等占全文的比重)

总相似片段: 124

期刊: 25 博硕: 74 外文: 0 综合: 0 自建库: 0 互联网: 25

颜色标注说明:

- 自写片段
- 复写片段 (相似或疑似重复)
- 引用片段
- 引用片段(自引)
- 专业用语 (公式定理、法律条文、行业用语等)

目 录

目录

第1章 前言 3

- 1.1 车牌识别介绍 3
- 1.2 车牌识别应用领域 3
- 1.3 实现方法简介 4
- 1.4 OpenCV 4
- 1.4 开发环境 6

第2章 方案论证 7

- 2.1 车牌定位方案论证 7
 - 2.1.1 基于车牌颜色分布的定位方法 7
 - 2.1.2 基于Radon变换的车牌定位方法 8
 - 2.1.3 基于边缘检测的车牌定位方法 8
- 2.2 字符识别方案论证 9
 - 2.2.1 基于神经网络的字符识别 9
 - 2.2.2 基于KNN最邻近算法的字符识别 11
- 2.3 方案比较与选择 12

第3章 车牌位置检测 13

- 3.1 图像预处理 13
 - 3.1.1 RGB空间与HSV空间的转换 13
 - 3.1.2 图像滤波 14
 - 3.1.3 图像二值化 15
- 3.2 轮廓特征提取 16
 - 3.2.1 边缘检测 16
 - 3.2.2 轮廓筛选 17
 - 3.2.3 轮廓提取与分组 18
- 3.3 车牌切割 20

第4章 基于KNN的字符识别 21

- 4.1 KNN简介 21
- 4.2 字符预处理 22
 - 4.2.1 字符分割 22

4.2.2 图像归一化 23

4.3 样本训练 23

4.3.1 贪心算法 23

4.3.2 训练过程 24

4.4 KNN算法的字符识别 27

第5章 系统功能、指标参数 29

谢辞 32

参考文献 33

附录 34

摘要

车牌识别是指带有车牌的图片经过特定系统的处理以字符形式输出车牌信息，是计算机视觉领域的一部分。本文研究内容的便是实现这样一个系统，主要包括车牌位置检测，字符识别两个部分。OpenCV是一个广受欢迎的开源计算机视觉库，提供了完善的图像处理函数，使用OpenCV可以大大节省开发时间加快开发效率，本系统的便是在OpenCV的基础上进行开发。系统在位置检测与字符分割部分取得了极高的准确度，字符识别部分准确率还有待提高。

关键字：车牌识别，OpenCV，计算机视觉

Abstract

Vehicle license plate Recognition is a part of computer vision field, which is a kind of picture with license plate, which is processed by special system to output license plate information in character form. The content of this paper is to realize such a system, mainly including license plate location detection, character recognition two parts. OpenCV is a popular open source computer Vision Library, providing a perfect image processing function, using OPENCV can greatly save development time and speed up development efficiency, this system is based on the OpenCV development. The system obtains the extremely high accuracy in the position detection and the character segmentation part, the character recognition part accuracy rate still needs to improve.

Keywords: vehicle license plate recognition, OpenCV, computer vision

第1章 前言

1.1 车牌识别介绍

随着大数据与人工智能领域的发展，计算机视觉技术逐步得到应用，车牌识别作为其中一个炙手可热的应用项目，天生拥有广阔的市场与发展前景，从而吸引了大量企业从事相关研发。

车牌识别一般指车牌识别系统（vehicle license plate recognition, VLPR），即通过摄像头采集车辆信息，通过对视频图像的处理得到车牌信息，从而达到识别车辆的目的，是计算机视频图像识别技术在车辆牌照识别的一种应用。

1.2 车牌识别应用领域

作为使用最为广泛的交通工具，汽车的身影无处不在，车牌识别的应用同样无处不在，以下列举了几个常见的应用。

不停车电子收费系统（,Electronic Toll Collection, ETC），通过在高速公路出入口安装ETC系统大大加快了车辆通行速度，有效缓解高速公路收费处的拥堵问题，节省大量人力，并能通过系统记录行程信息，便于随时查证。

停车场出入管理系统，传统停车场出入由人工给卡并人工计算停车时间，不仅耗费人力，速度还很慢。应用停车出入管理系统自动识别车辆，配合缴费系统便可以实现无人操作。

超速违章处罚，通过在高速路安装超速监测系统，能够自动监测超速车辆并识别车辆信息，从而加大威慑力减少违法行为。通过遍布的摄像头能够迅速定位犯罪分子的车辆位置加快破案效率。

总之，车牌识别已经成为人类离不开的一项技术，使得车辆的监督管理变得极为便利，因此提高车牌识别的准确率，加快识别速度，

降低成本等变得极为重要，虽然这一领域的实际应用已经比较成熟，但依然存在较大的进步空间，依旧存在很大的研究价值，本设计便是本着这样的态度进行研究希望能够理清车牌识别的逻辑并寻求突破。

1.3 实现方法简介

车牌识别系统的实现主要是两个方面，车牌检测与字符识别。车牌检测即车牌图像定位，在一副完整图像中找到车牌位置并进行分割。车牌检测部分即对分割得到的车牌部分进行字符识别得到车牌号。

车牌图像定位是车牌识别中至关重要的一步，由于字符识别的输入正好是车牌图像定位的输出结果，所以车牌图像定位的准确与否直接关系到最后结果的正确与否，并且需要考虑到周围环境的影响。车牌图像定位主要用到图像变换与图像形态学处理等知识进行处理。对图像进行预处理实现灰度变换以及滤波除去噪声等操作，利用车牌部分明显的方形轮廓作为特征从而实现车牌位置的查找。

车牌识别部分使用K最邻近（K-NearestNeighbor）分类算法，为数据挖掘分类算法中的一种，通过先验知识提取车牌中各个字符特征记录与XML文件中，在识别时提取上一步车牌图像定位得到切割图像的特征进行特征对比，选取与模板特征最邻近的特征群所对应的字符为结果字符。

1.4 OpenCV

OpenCV (Open Source Computer Vision Library) 是一个广受欢迎的计算机视觉库，由C++编写并提供了Python, Ruby, matlab的接口支持，最开始有英特尔公司出资赞助开发，如今由Willow Garage提供支持。它为图像处理，模式识别，三维重建，物体跟踪，机器学习和线性代数提供了大量的算法。

OpenCV的基本结构主要包括5个模块，以下为其中的四个模块如图2.1所示。

图2.1 OpenCV基本结构

其中CV模块包含基本的图像处理函数与高级的计算机视觉算法，是图像处理的基础。MLL (Machine Learning Library) 机器学习库，包含一些机器学习算法以及基于统计的分类和聚类工具。HighGUI则是针对视频图像的读取显示部分的，能够方便地读取与显示视频图像。CXCORE包含了OpenCV基本的数据结构，动态数据库，绘图函数，数组操作相关函数，辅助功能与系统函数的宏。图中没有包含第五个模块CnAux模块，该模块一般存放一些即将被淘汰的算法。

OpenCV出现至今已经更新了多个版本，本实验采用OpenCV3.0进行操作，相对于OpenCV2.X大体保留了OpenCV2经典的C++和Python编程接口风格，添加了Python3.X的支持，优化了Java的支持，并新接入了MATLAB的支持。对架构进行了调整，添加了新的算法，新版本包括了TLD，鱼镜头模型等全新算法。还包括更高层次的高级封装，比如汽车检测等。优化了更多指令集，除了对Intel平台的cpu进行了优化，还添加了对ARM平台指令集的支持，加快了处理的速度。

1.4 开发环境

本设计采用的开发工具为Visual Studio2012，以下简称VS。VS是微软公司推出的开发工具集，所编写的代码适用于Windows所有平台，具有极其完善的功能，包括了整个软件生命周期所需要的大部分工具。

第2章 方案论证

2.1 车牌定位方案论证

2.1.1 基于车牌颜色分布的定位方法

由于汽车车牌颜色对比度很高且颜色固定，所以可以通过检测车牌颜色的方法定位车牌。基于车牌颜色的方法定位车牌的方法采用区域生长法对彩色图像进行分割，根据车牌的外部特性初步确定车牌的候选区域，再利用车牌背景以及字符颜色的差异实现对位置的确定。

获取的汽车图像大多是RGB格式，处理起来比较麻烦，所以需要将图片从RGB空间转换到HSV空间。在HSV空间中只需要色度（Hue）和饱和度（Saturation）两个分量便可以对车牌颜色分布进行计算。基于车牌颜色分布的定位流程如图2.1所示。

图2.1 基于车牌颜色分布定位流程

此方法减少了车牌的漏检情况，在车牌倾斜情况下也有很高的检测准确度，提高了检测的准确性与稳定性。但是不适合极端天气下的检测，在车牌与背景对比度较低的情况下检测的准确性与速度都有所下降，而且由于车牌的颜色不一致，比如小轿车是蓝色背景，货

车是黄色背景，除此之外还有很多其它不同的车牌，所以情况较多加大了检测的难度。

2.1.2 基于Radon变换的车牌定位方法

Radon变换在数学上指一种积分变换，在图像处理领域主要用于对倾斜的图象进行复原。现实生活中车牌都是矩形的，拥有固定不变的形状，边缘检测后车牌的长宽相对集中，其中长宽之间的距离相对固定，边缘都由一定比例的线段组成。但由于拍摄角度的原因，导致成像后的车牌发生变形扭曲，与固定的特征不符，基于Radon的变换算法使将图像在各个方向进行投影计算，通过对投影结果的分析确定图像的偏移角度，从而对图像进行旋转，拉伸，变形，使之成为矩形或者近似矩形。基于Radon变换的车牌定位方法流程如图2.2所示。

图2.2 基于Radon变换的车牌定位流程图

2.1.3 基于边缘检测的车牌定位方法

边缘即为灰度图像中灰度值发生急剧变化的区域边缘，是图像的基本特征，车牌区域边缘特征信息丰富，所以通过边缘检测能够准确地检测到车牌区域，是车牌图像定位的有效方法和重要手段。边缘检测指通过采用某种算子来提取图像中的物体与边缘交界部分的方法，常见的算子有sobel算子，拉普拉斯边缘检测算子，prewitt算子，roberts算子等。

边缘检测的基本方法是利用边缘部分与背景具有较大的对比度，即差分值较大，通过检测发现当前点在临近区域为局部最大值且差值大于所规定的阈值，则判定当前点为边界点，众多点连在一起便形成了物体边缘的轮廓。车牌边缘检测便是通过这样的原理找到车牌位置，实现定位。其实现的流程如图2.3所示。

图2.3 基于边缘检测的车牌定位流程图

此方法检测速度快，精准度与效率高，且对噪声有很好的适应能力，所以适用范围广，对于一张图片中有多个车牌的情况有也有很好的效果。但是对于车牌破损，边缘有污渍的车辆检测效果大大下降，对于背景复杂的情况，可能存在有类似于车牌的边缘，错把一些具有类似性质的其他标志物当做车牌导致定位错误。

2.2 字符识别方案论证

2.2.1 基于神经网络的字符识别

作为机器学习中的热门算法，神经网络模拟人类大脑的工作原理，以神经元为基本单位建立复杂的神经网络系统。人工神经网络结构如图2.4所示，由细胞体，树突，轴突构成的细胞体组成。

图2.4 神经网络结构

抽象后的神经元数学模型如图2.5所示，输入信号P经过w的加权后经过累加器与b相加，经过函数f的处理最终输出a。图2.5表示的为单层神经网络模型，实际运用中常常使用单层神经网络的级联形式即多层神经网络模型。

图2.5 神经元数学模型

多层神经网络分为三层，输入层，隐含层，输出层。通常来说，隐含层层数越多，神经网络的分析能力越强，训练的数据也更多，多层神经网络模型如图2.6所示。神经网络的训练即为对模型中各个参数的不断修正的过程，首先批量输入学习样本并对输入量进行归一化处理，并对最大训练次数，学习精度，隐含层节点数，初始权值，阈值，学习速率进行初始化，然后计算各层已经最后的输出情况，计算输出层误差从而作为反馈对参数进行调整直到误差减小到允许的范围。

图2.6 多层神经网络模型

2.2.2 基于KNN最邻近算法的字符识别

KNN (K-Nearest Neighbor) 属于非参数方法，在非参数估计 (nonparametric estimation) 中，我们只假设相似的输入具有相似的输出，这是一种合理的假设：世界是平稳的，并且无论密度，判别式还是回归函数都是缓慢地变化[2]。KNN算法则是这种思想的具体体现，通过计算判定样本与训练样本集的距离，在指定距离内涵盖最多样本所属的类别即为需要判定样本所属类别，如图2.7所示选取k值，计算以欧氏距离k为半径的圆内其他类别的个数，图中中心小红点以k为半径的圆内三角形个数最多，则判定中心小红点为三角形。

图2.7 KNN示意图

KNN算法为惰性算法，不需要和神经网络一样训练参数得到判别函数，而是直接将训练集记录下来在判别的时候依次比较，所以此算法的训练成本较低，但是在数据量很大的时候使用knn算法有很大的计算量，所以一般用在训练量不大的地方。KNN算法可以归纳为以下几步：

选择邻近的数量K和距离度量方法[1]。

找到待分类样本的K个最近邻居[1]。

根据最邻近的类标进行多数投票[1]。

2.3 方案比较与选择

通过对以上主流车牌方法识别的研究学习，发现各种方法均有相应的优点与缺点，考虑到车牌普遍拥有一样的方形结构并且车牌内有比较规律的字符，所以最终决定采用基于轮廓提取的定位方法查找车牌的位置。本次车牌识别主要对象限于只有字母与数字的车牌，所以仅有26个字母与10个数字，也就是在36个类别中进行匹配，所以类别并不是很多，若采用神经网络的方法进行识别会加大训练难度，所以最终选择了KNN算法实现车牌字符识别。综上，最终选择的方案为轮廓提取+K-邻近算法。

第3章 车牌位置检测

3.1 图像预处理

图像预处理是车牌提取中的重要环节，通过输入设备输入的通常为多通道的彩色图像，数据量过大不利于图像的处理，图像的预处理包括图像格式空间转换，图像的二值化，图像的滤波。

RGB空间与HSV空间的转换

RGB色彩模式主要面向硬件描述，是基于三原色原理的一种描述方式，通过红，黄，蓝三种颜色按不同比例混合可以合成各式各样的颜色，实际应用中输入设备采集的图像正是这种格式，但不便于人感官的辨认，比如我们看一种颜色，并不知道它由什么什么比例的三原色混合得到。HSV(hue, saturation, value)色彩空间则是面向用户的一种颜色空间，也称作六角锥体模型，由色调H，饱和度S，前度V组成。颜色模型如图3.1所示。

图3.1 HSV颜色模型

当图像处于RGB空间时数据量太大且不利于对图像的处理，转化为HSV空间后直接取V(Value)通道便得到对应的灰度图像，图像处理时主要对灰度图像进行处理，转换后的灰度图像分别如图3.2所示。

图3.2 灰度变换后的汽车图像

图像滤波

图像滤波指在尽量保留图片特征信息的条件下对目标的噪声进行滤除，是图像处理中重要的部分。由于灰度图像中存在很多细小的颗粒噪声，如果直接进行二值化处理不仅会增加计算量，还会提取到很多噪声信息不利于后面的轮廓提取，进行滤波处理后可以滤除细小的颗粒物与复杂背景中的细微信息，利于后续对图像的处理。

数学形态学是图像处理中所用到的基本方法，其基本思想为用具有一定形态的结构元素去度量和提取图像中的对应形状，从而达到对图片分析识别与处理的目的，数学形态学的基本运算包括：膨胀与腐蚀，骨架抽取，开闭运算，击中与击不中变换，灰值形态学梯度，top-hat变换，颗粒分析，流域变换等。形态学运算的主要功能为：（1）图像的预处理（去除噪声，简化形状）；（2）增强物体结构（抽取骨骼，粗化细化，物体标记）；（3）从背景中分割兴趣区域；（4）物体量化描述（周长，面积，投影）。

膨胀操作使得物体轮廓向外扩张，若物体中存在细小空洞或者某些边缘有细小的断口，可以使用膨胀操作对断口进行填补。腐蚀操作是膨胀的反操作，会去掉物体的边缘点，细小物体会直接被认定为边缘点而被消除。腐蚀操作将图像轮廓边缘缩小，从而达到除去孤立的细小噪声点，使得轮廓平滑。

对图像进行先膨胀再腐蚀的操作称为闭运算，相反，对图像进行先腐蚀再膨胀的操作称为开运算。开运算用于消除孤立的小物体，在纤细点处分离物体，平滑物体轮廓并不明显改变其面积。顶帽操作是原图像与开运算结果之差，黑帽操作为闭运算结果与开运算之差。通常来说，由于存在噪声的影响，图像在二值化操作后的边界往往不平滑，存在一些噪声孔，背景区域也散布着一些小的噪声物体，使用开闭操作通常可以有效改善此类情况。本文通过顶帽与黑帽操作去除孤立噪声和平滑轮廓曲线，填补细小空洞和链接邻域。

高斯滤波是一种线性平滑滤波，用于消除高斯噪声，本实验中采取的滤波方式正是高斯滤波，其一般有两种实现方式，一种是通过傅里叶变换去除高频分量，另外一种是用离散化窗口滑窗卷积，最常见的是用滑窗实现，本文使用高斯函数对灰度图像进行处理从而达到滤除高频噪声的目的，对于零均值的高斯函数为：

(3-1)

3.1.3 图像二值化

进行轮廓提取前需要找到合适的阈值对图像二值化，二值化后图像中就只有黑色与白色两种颜色，便于进行轮廓提取，所以找到合适的阈值便显得及其重要，一般采取试探法尝试不同阈值观察效果，当车牌颜色与车牌内文字颜色能清晰分开时，便认定为合格。为了加大图像中黑白区域对比度使用图像处理形态学方法（灰度+闭运算-开运算）处理，二值化后的结果如图3.3所示。观察图像易知经过二值化处理后图像中的关键信息（车牌与字符）并未损失反而更加突出，接下来的处理便基于此二值图像，由于仅有两种颜色，所以处理起来的信息量大大降低。

图3.3 二值化后图像

3.2 轮廓特征提取

轮廓提取指在包含目标与背景的数字图像中，忽略图像背景，噪声等不需要的信息，对目标进行提取检测的技术。目前轮廓提取主要有两种方式，第一种利用传统的形态学方法使用边缘检测算子对边缘进行检测，另外一种方法是使用从人类视觉中提取出来的数学模型提取边缘信息，这儿使用传统的形态学方法进行边缘检测。

3.2.1 边缘检测

边缘检测属于计算机视觉与图像处理中的基本问题，其目的是检测图像中亮度变化明显的点，在二值化图像中即是提取白色与黑色交界处的点。通过边缘检测大幅度地剔除了与目标无关的信息，利用OpenCV中的cv::findContours函数可以轻易得获取图像的边缘信息，二值化图像经过边缘检测函数后输出图像的轮廓信息，每个轮廓为一个整体。提取轮廓后的结果如图3.4所示。

图3.4 提取轮廓信息后的图像

对比二值化图像与灰度图像容易发现，提取轮廓后的图像由线条构成，以黑色为背景，白色即为提取之后的轮廓。

3.2.2 轮廓筛选

由轮廓提取结果可知，轮廓提取后拥有大量的背景信息，所以需要轮廓进行筛选去除大部分没必要的轮廓信息。通过计算轮廓的半径，周长等信息对轮廓进行筛选除去太大或者太小的轮廓从而达到筛选轮廓的目的。使用OpenCV中possibleChar.boundingRect.area()函数得到轮廓的面积，使用OpenCV中内置函数possibleChar.boundingRect.width得到轮廓的最大宽度，使用内置函数possibleChar.boundingRect.height获取轮廓最大高度，设定最小area为80，最小width为2，最小height为8，筛选后的轮廓如图3.5所示。

图3.5 轮廓筛选结果

经过筛选，去除了大量不相关的轮廓，并且完整保留了关键信息，大大地减少了后续计算量。

3.2.3 轮廓提取与分组

由图3.5可知，筛选后的轮廓依旧包含与车牌无关的部分，利用车牌字符轮廓之间的关联性可以有效提取有用轮廓与多余轮廓，车牌部分字符中心间距较窄，并且分布规律，对轮廓规律的提取流程如图3.6所示。将满足相关性的轮廓分到同一个组别，相关性的条件正是为了满足车牌的字符而量身定制，也就是说，车牌字符的轮廓都会被分到同一个类别并且成为一个整体，通过对现存轮廓的分组使得所要提取的所有信息都包含于此，后续处理便只需要针对组别进行便可。处理至此便完成了车牌检测的大部分工作，用红色的框将每个组括起来便能清楚地观察到处理后的结果，如图3.7所示。

图3.6 轮廓分组流程

图3.7 轮廓分组筛选结果

对分组后的轮廓用红色的方框包含，计算方框的旋转角度，对不满足条件的组别直接剔除。

3.3 车牌切割

之前的所有处理均是在原图像上进行，目的都是为了提取图像中的车牌部分信息以便后续识别部分的处理，经过预处理与轮廓提取操作后已经成功提取出车牌部分信息，现在只需要对兴趣区域定位裁剪即可，即裁剪轮廓提取部分的红色区域。裁剪结果如图3.8所示。

图3.8 车牌切割后结果

第4章 基于KNN的字符识别

4.1 KNN简介

所谓最邻近算法，即使给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的K个实例，这K个实例的多数属于某个类，就把该输入实例分类到此类中。

KNN算法虽然为最简单的机器学习方法之一，但是应用得能得到极好的识别效果，本实验使用精简的最邻近算法。非参数方法的时间和空间复杂度正比于训练集的大小，目前已经提出了一些精简算法，以减少存放的实例数而不降低其性能。其基本思想是选择X的最小子集Z使得用Z代替X时，误差不增加（Dasarathy 1991）。最著名的方法是精简的最邻近（condensed nearest neighbor），其使用1-nn作为分类的非参数估计（Hart 1968）。1-nn以分段线性的方式近似判别式，并且只需要保存定义判别式的实例。类区域内部的实例不必作为它的同一类的最邻近存放，并且它的缺失不会导致（训练集上的）任何错误。这样的子集称为相容子集，并且我们希望找出最小的相容子集。

早在二十世纪Hart提出一种找到Z的贪心算法。即找到当前最佳是配置，训练集充空集开始，以空集去识别整个训练实例，每当出现一个错误时，便将错误实例添加至训练集中，直到没有错误出现为止，这样便将训练集有效地压缩并且实现误差不会增加，此方法大大减小了训练样本的冗余度，尤其是在大样本的情况下大大降低了计算量提高了计算速度，训练过程如图4.1所示。

图4.1 压缩的最邻近算法

4.2 字符预处理

在对图片进行训练之前有必要对图像进行预处理使之满足统一的规则，从网上得到的图片各式各样，经过预处理之后的图像才能用于计算相互之间的距离。首先经过与3.1章相同的预处理得到车牌字符的二值化图像，之后进行字符分割与归一化处理。

4.2.1 字符分割

字符分割是车牌识别系统中的重要部分，车牌识别的时候也是对单个字符进行识别，所以字符识别的准确与否直接影响了最终的识别结果。

本实验采取计算字符轮廓大小的方法实现字符分割，直接计算字符轮廓大小然后以方形对图形进行分割。这样的方法分割的结果十分准确并且具有自动适配性，不会因为车牌的型号不同而受影响，处理之后的结果如图4.2所示。

图4.2 字符分割结果

计算出字符的轮廓之后，按照轮廓对车牌进行分割并将分割结果存入新的数组中，以便后续训练。

4.2.2 图像归一化

归一化是一种简化的计算方式，用以简化量纲以便统一计算，使物体系统绝对值的计算方式变为相对值的计算方式。

截取之后的字符图片大小大小各异，没法直接进行欧式距离的计算，所以需要对图像进行归一化处理，在综合考虑归一化后的速度与效率之后选择了200×300作为归一化的结果，将所有切割后的字符图像统一重新缩放至200×300的分辨率。

4.3 样本训练

为了减少样本个数，提高识别速度，训练过程中采取了贪心算法的思想，并将训练过后的结果数字化之后保存于xml文件中。

4.3.1 贪心算法

贪心算法又称贪婪算法，指在对某一问题进行求解时，只考虑适配与当前样本的最佳结果而不从整体最优解上考虑取值，所得的结果为局部最优解。精简的最邻近算法就是一种贪心算法，其目的是最小化训练误差与复杂度。可以用一个增广误差函数表示如式4-1：

$$E(Z|X)=E(X|Z)+\lambda|Z| \quad (4-1)$$

其中 $E(X|Z)$ 用于存放Z在X上的误差。 $|Z|$ 是z的基数，第二项为对复杂度的惩罚。与其他的正则化方案相同， λ 体现了误差与复杂度的

折中，使对于较小的 λ ，误差变得更为重要，随着 λ 的逐渐增大，对模型的惩罚则会更大。

贪心算法的一般训练流程如图4.3所示。

图4.3 贪心算法一般流程

4.3.2 训练过程

因为KNN是非参数化模型，所以其训练过程与神经网络，SVM（支持向量机），核机等模型的训练方法不同，不用对未知参数进行训练。简单来说，作为惰性学习算法，KNN的训练过程就是对样本的存储过程，同时因为采用了贪心算法，样本的存储与筛选便成了训练过程中的首要问题。

首先从训练样本集中读出第一个样本，此时没有任何数据对样本进行识别，所以识别结果自然错误，将此样本值保存到用于识别未筛选样本的识别文件中。如此不断往复，使用识别文件中的数据对未筛选内容进行识别，若识别成功则舍弃此样本，若识别失败，将识别失败的图片数据存入识别文件中，直到达到要求的正确率为止，训练流程如图4.4所示。

图4.4 KNN训练流程

训练用的样本来自互联网，训练的时候只需要对图片就行归一化后便可以直接使用。样本如图4.5所示。

图4.5 训练样本

训练之后的数据以XML的格式保存起来，两个XML文件分别保存图片数据与数据对应的分类。本次设计以classification.xml文件保存对应的分类，images.xml与classification.xml数据一一对应，用于保存每个训练样本数据所处的分类。本次车牌识别仅仅针对只有字母与数字的车牌，所以车牌字符类别共有26个字母与10个数字一共36个类别，每个类别有5组不同的数据与之对应。判断类别的数据如图4.6所示。

图4.6 KNN车牌字符类别数据

由图可以看出，数据均为数字，这是因为所有的类别均用相应的ASCII码表示，字符0-9对应着ASCII码的48-57，字符A-Z对应着ASCII码的65-90。这就是上图中数字所表示的意义。上面说到，每一个分类都有对应的数据，数据即是用作与待识别字符进行欧式距离计算的，距离最近的数据所对应的类别即被判定为待识别字符的类别。其部分数据如图4.7所示（数据太大只截取了开头与结尾处数据）。由图可以看出，样本图片的数据为180600的矩阵，180与classification.xml中的180行对应，对应着365=180个类别，600则是字符图像归一化后的分辨率2030=600，即对应的图片数据，目前为止，KNN模型的训工作练便完成了。

图4.7 部分字符图片数据

4.4 KNN算法的字符识别

关于KNN算法的字符识别过程前面有过简要的介绍，即通过计算待识别字符图片与训练样本中的已分类数据之间的距离得到相似度，根据相似度来对未分类数据进行判定。OpenCV中内置的机器学习算法中直接包含了KNN算法，利用内置的KNearest->setDefaultK(1)设置K的初始值为1；使用kNearest->train()读取4.3节中训练好的样本数据；使用kNearest->findNearest()读取待识别数据并且与样本数据进行对比并返回对应类别的ASCII编码。为了将识别后的结果更清晰地显示出来，直接将字符写入原图像车牌下面进行对比。识别后的结果如图4.8所示

图4.8 识别结果

第5章 系统功能、指标参数

系统的主要任务大致分为车牌位置检测与字符识别两个部分，车牌位置检测的结果直接影响字符识别的准确度。所以以下指标分为两个部分，车牌位置检测以随机的带有车辆的图片进行识别，字符识别的指标建立在车牌位置识别正确的前提下进行指标测试。

识别特征：车牌号码

输出结果：车牌号码

车牌倾斜角度范围：-10-- +10

车牌位置检测准确率：100%

字符识别准确率：98.7%

识别速度：20.55ms

第6章 总结与展望

本次设计针对目前计算机视觉领域的热门问题车牌识别给出了边缘检测+KNN最邻近算法的识别方案，旨在提出一种快捷简单的方法实现车牌识别的工作。整个项目的开发环境为Visual Studio2012，使用开源的计算机视觉库OpenCV。结合前人经验与个人的实践成功实现了车牌的识别工作。

首先，使用图像形态学领域的知识结合OpenCV便利的接口函数，通过图像空间变换，灰度变换，滤波，阈值化，轮廓提取等操作定位车牌位置，确定位置后切割车牌，则车牌位置检测部分工作基本完成；字符识别使用KNN算法，作为惰性算法，KNN的训练过程即使对训练样本的加工与分类存储，整个过程包括字符预处理，字符分割，图像归一化，计算欧式距离得到待分类字符所属类别，基本完成了预定的目标，实现的车牌字符的提取。

该系统在车牌识别中通用性强，全程使用C++进行开发，可以方便地移植到移动设备与嵌入式系统上工作。本次设计虽然完成了车牌识别的工作，但是整个过程中依然存在很多不足的地方有待提高。

对于车牌角度纠正的问题没有给出合适的方案，所以对于倾斜的车辆图片不能很好地进行识别，此处还有较大的改进空间。

系统对于复杂背景与复杂环境的抗干扰能力不足，当图像背景复杂（比如有类似车牌的标志，多个车牌的情况）时，识别效果并不理想。

在夜晚或是光线不均匀的环境中，阈值化的效果不佳，造成车牌定位错误的情况，或是由于反光导致识别过程中分类错误，不能有效地提取到车牌字符。

识别的准确率与实时性有待进一步地提高，车牌识别的实时性至关重要，正确识别车牌只是最基本的要求，当输入设备的分辨率较高，输入图像的实时性要求高的时候，系统的识别能力下降，有卡顿的现象发生，算法依然需要完善。

谢辞

在熬过了漫长的两个月后，毕业论文终于接近尾声，毕业设计能够正常完工离不开老师同学们的默默支持，在这里首先要对我的指导老师卿曹进教授表示诚挚的感谢，卿老师为毕业设计的方向与实施方案给出了指导性的意见。以及CSDN论坛上的博主们，你们的博客解决了我遇到的很多难题。

同时，我还要感谢微软公司，感谢你们开发出visual studio这么好用的IDE以及Microsoft Word，没有这些我是完成不了代码的调试以及论文的编写工作；感谢Willow Garage一直以来对OpenCV的开发维护，使得我们能免费用上如此好用的计算机视觉库，没有你们的支持，此次毕业设计就要更换题目了；感谢Image-Net网站维护人员十多年来的搜集素材努力，是你们为机器学习提供了大量的训练样本，极大地减少了此次论文的工作量；感谢学校及时地装上了空调，使得同学们能在炎炎夏日睡个好觉，否则是不会有如此精力持续地接收毕业论文的摧残；感谢学校二食堂年复一年坚持不懈地做出难以下咽的饭菜，让我时刻体验到生活的艰辛与不易，给了我们不断奋斗的强大动力，让我们明白吃得苦中苦方为人上人的祖传真理；感谢少年JUMP一直以来推出的优秀漫画，让我能够在论文的重压下找到一处喘息的净土，重新发现生活的美好之处；感谢篮球，好的身体是完成毕业设计不可缺少的重要支撑，每周两次的篮球运动使得身体一直保持着不错的状态。感谢维普论文检测系统，你精准地检测使我清晰地了解到论文的重复内容，论文查重能过你功不可没。一路走来需要感谢的人和事太多太多，怀有一颗感恩的心，才能更好地向前，小草因为感谢土地使它有了栖身之所，因此献给人间万顷碧绿；鲜花由于感恩大地对它的滋养之恩，所以呈现给人间万紫千红。感恩是一束金色的阳光，它能融化冰雪，温暖万物，传播大爱。再次感谢大家的帮助与支持。

参考文献

Sebastian Raschka. Python机器学习[M]. 机械工业出版社，2017.3

Ethem Alpaydin. 机器学习导论[M]. 机械工业出版社，2015.11

王志明. 数字图像处理与分析[M]. 清华大学出版社，2012.2

郭延辉、王小权、陈叶芳. C++程序设计教程基于案例与试验驱动[M]，2010.6

附录

- 声明:

报告编号系送检论文检测报告在本系统中的唯一编号。

本报告为维普论文检测系统算法自动生成，仅对您所选择比对资源范围内检验结果负责，仅供参考。

客服热线: 400-607-5550 | 客服QQ: 4006075550 | 客服邮箱: vpcs@cqvip.com

唯一官方网站: <http://vpcs.cqvip.com>



关注微信公众号