

基于 Tesseract_OCR 文字识别的研究

曾悦¹, 马明栋²

(1. 南京邮电大学 通信与信息工程学院, 江苏 南京 210003;
2. 南京邮电大学 地理与生物信息学院, 江苏 南京 210003)

摘要: 光学字符识别(optical character recognition, OCR), 简单来说, 主要是利用光学技术和计算机技术将目前所使用的印刷体字符通过检测每个像素的亮、暗模式转换成一个黑白图像的文件, 然后再使用识别的手段将这个黑白图像的文件转换成计算机可以识别的文字。该文主要分为四个模块: 文字信息提取、字符识别、系统实现、实验结果与分析。文字信息提取模块包括图像预处理、文字信息区域的截取和修正、字符分割, 对输入的图片进行处理, 以降低随机噪声, 确保文字信息区域包含完整的文字信息, 提高识别的准确性。使用 Tesseract 的 OCR 引擎对处理后的文字信息区域部分进行识别, 提取出图片中的文字信息。微软基础类库(Microsoft foundation classes, MFC), 是微软公司实现的一个 C++ 类库, 主要封装了一部分的 API 函数, 灵活性大。最后, 在 VS2015 环境下使用微软基础类库实现了一个文字识别系统, 并对样本图片库进行系统的测试。测试结果表明, 该系统具有更高的识别率。

关键词: 光学字符识别; 文字识别; Tesseract 框架; 微软基础类库; C++

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2021)11-0076-05

doi:10.3969/j.issn.1673-629X.2021.11.013

Research on Text Recognition Based on Tesseract_OCR

ZENG Yue¹, MA Ming-dong²

(1. School of Telecommunications & Information Engineering, Nanjing University of
Posts and Telecommunications, Nanjing 210003, China;

2. School of Geographical and Biological Information, Nanjing University of Posts and
Telecommunications, Nanjing 210003, China)

Abstract: Optical character recognition (OCR), in simple terms, mainly uses optical technology and computer technology to convert the currently used printed characters into a black and white image by detecting the light and dark patterns of each pixel the file, and then uses the means of recognition to convert this black and white image file into text that can be recognized by the computer. This article is mainly divided into four modules: text information extraction, character recognition, system implementation, experimental results and analysis. The text information extraction module also includes image preprocessing, text information area interception and correction, and character segmentation. The input image is processed to reduce random noise, ensure that the text information area contains complete text information, and improve the accuracy of recognition. The Tesseract's OCR engine is used to recognize the processed text information area and extract the text information in the picture. Microsoft foundation classes (MFC) is a C++ class library implemented by Microsoft Corporation, which mainly encapsulates a part of API functions with great flexibility. Finally, a text recognition system is implemented using MFC in the VS2015 environment, and the sample picture library is systematically tested. The test shows that this system has a higher recognition rate.

Key words: optical character recognition; text recognition; Tesseract framework; Microsoft foundation classes; C++

0 引言

在二十世纪三十年代末, 德国科学家 Tausheck 提出了 OCR 的概念^[1], 并获得了 OCR 专利。之后美国

科学家 Handel 也提出了利用技术进行文字识别^[2]的想法。二十世纪六十年代, OCR 技术首次被使用于生产实践中, 至此第一批 OCR 系统诞生, 其中

收稿日期: 2020-12-28

修回日期: 2021-04-29

基金项目: 江苏省自然科学基金-青年基金项目(BK20140868)

作者简介: 曾悦(1996-), 女, 硕士研究生, CCF 会员(F3864G), 研究方向为图像处理; 马明栋, 博士, 教授, 研究方向为地理信息系统平台软件设计与开发等。

Farrington3010 最具有代表性^[3],但是运行速度慢且只能识别简单的字符。1996 年,IBM 公司的 Casey 和 Nagy 完成了中文字符识别系统的研发,采用模板匹配法能识别 1 000 个印刷体汉字^[4]。

中国在二十世纪七十年代才开始在字符识别方面的研究,主要是常见的字符,比如数字、英文和汉字。1986 年进入到一个实质性的阶段,虽然很多研究单位推出了一些中文 OCR 的产品,但由于识别率低,硬件设备成本高,运行速度慢等,未能达到实际要求。只有信息部门、新闻出版社等个别单位使用汉字识别软件。

到二十世纪九十年代,随着平台式扫描仪的普及,以及信息科学技术的发展,大大推动了该技术的发展,使得识别的速度、效率大大满足了用户的使用要求,并在学校、医院、企业等地方得到了广泛的应用。

Tesseract 是一个 OCR 库,目前由 Google 赞助^[5]。Tesseract 是目前公认最优秀、最精确的开源 OCR 系统。除了极高的精确度,Tesseract also 具有很高的灵活性。它可以通过训练识别出任何字体,也可以识别出任何 Unicode 字符^[6]。

文中在分析了文字识别系统的需求后,结合相关技术和方法设计了一个基于 Tesseract 文字识别框架的文字识别系统,主要工作如下:

(1) 文字信息提取:在识别输入图片之前对图片进行处理,避免原图的质量、亮暗程度、倾斜程度对实验结果的影响,使得提取出来的文字信息区域平滑、规范。截取修正包含完整的文字信息区域,对原图上的所有文字进行识别,以提高有效性。

(2) 文字识别:通过 Tesseract 识别框架,对提取出的文字信息进行模型训练和优化字库,提取输入图片中的文字信息,并重构原图版面和矫正识别信息。

(3) 系统架构设计:根据项目需求分析,对各个模块进行实现并优化系统。

1 文字信息提取

1.1 图像预处理

图像处理主要是对图像成像所出现的问题进行修正,可以降低随机噪声,为后续识别做准备。预处理一般包括二值化,灰度化,畸变校正,几何变换(扭曲、旋转、透视等等),去除模糊、图像增强、光线校正,规范化等等^[7]。

二值化(image binarization)^[8]是一种基于灰度直方图的图像分割算法,将图像中的像素分为两类,通过适当的阈值选取获得仍然可以反映图像整体和局部特征的二值化图像,减少图像中的数据量,凸显目标轮廓。适用于物体与背景的灰度值差别比较大的情况。二值化算法的初始值被设置为整个图像灰度值的平均

值,求取最优二值化的值是这个算法的关键,也就是尽量求取灰度直方图中两个双峰间的最低点。

然而,如果直接用 OTSU 大律法^[9]进行二值化,实际效果并不是很好,因为这是全局阈值,比较好的二值化方法应该用局部阈值,毕竟图像上每个地方的灰度值差别是比较大的。局部自适应二值化的基本思想:首先针对某个像素点,确定它的邻域大小,然后根据邻域内像素值的分布情况来决定该像素点处的阈值,进行二值化。该方法有较强的自适应能力,能够根据局部亮度值设定阈值,解决一定程度的明暗情况。可以有效地划分亮度或者对比度差异较大的区域,获得效果较好的二值化图像。

Bernsen 方法主要是一种动态选择阈值的自适应方法,主要步骤为:首先读取图像并按照要求填充为指定形式的图像,然后根据当前位置像素点邻域灰度情况计算该点二值化阈值,比较当前像素点灰度值和阈值大小,如果阈值较小则将该点视为白点,反之将该点视为黑点,最后构建二值图像并显示。

假设在一幅 $M \times N$ 大小的图像中,针对 (i, j) 处的像素点,它的灰度值为 $f(i, j)$,将大小为 $(2\omega + 1) \times (2\omega + 1)$ 的正方形区域视为邻域窗口,则图像中 (i, j) 位置像素点的二值化阈值 $T(i, j)$ 可以表示为:

$$T(i, j) = 0.5 \times \left[\max_{\substack{-\omega \leq m \leq \omega \\ -\omega \leq n \leq \omega}} f(i + m, j + n) + \min_{\substack{-\omega \leq m \leq \omega \\ -\omega \leq n \leq \omega}} f(i + m, j + n) \right] \quad (1)$$

将该像素点的灰度值 $f(i, j)$ 和二值化阈值 $T(i, j)$ 进行比较,从而对图像进行二值化。比较的规则如下:

$$b(i, j) = \begin{cases} 0 & f(i, j) < T(i, j) \\ 1 & f(i, j) \geq T(i, j) \end{cases} \quad (2)$$

由于 $(2\omega + 1) \times (2\omega + 1)$ 的窗口并不能在 $M \times N$ 尺寸的图像中移动,所以要对图像进行填充。

1.2 文字信息区域的截取和修正

由于输入图片上的字体位置不固定、文字倾斜等情况,很难确保后续包含完整的文字信息,所以要对原图上的文字信息进行精确截取和修正^[10],确保待识别的文字信息区域含有较少的干扰信息和完整的文字信息。

对已处理后的二值化区域部分进行适当的调整,截取相对的文字信息区域。在获得区域截图之后,使用间隙法来确定截取的图片是否包括完整的文字信息。间隙法的主要思想是用于检查图区的图片是否存在两条间隙,若不存在则根据对应的情况进行调整,直到截取成功位置,如果在最大循环次数结束后,依旧没有检测到两条间隙,则返回错误代码。

1.3 字符分割

字符分割是将二值化后的图像分割成只包含单个

字符的图片,也是关键步骤。分割的效果如何对后续的文字识别结果有直接的影响,避免分割的字符出现重叠、分裂的情况,尽量取出完整清晰的文字。

基于投影的图像分割算法^[11-12]就是依据图像在水平和垂直两个方向的投影密度来确定行切分和字符切分。一般是先进行行切分,然后对每一文本行进行垂直投影进行字符切分。以行切分为例,首先判断图片是否存在倾斜,若存在则先进行倾斜校正(计算倾斜角度然后进行坐标变换),保证文本行都是水平以后,逐行扫描图像,记录每一行的前景像素点(即组成文字的像素点)个数,得到一个图像像素在水平方向上的统计结果。

然后根据这个结果来决定每一个文本行的起始行和结束行,最一般的做法是阈值法,即设定一个阈值,根据每一行的前景像素个数决定该行是否属于“文本行”的一部分。

(1)通过空白区域宽度判断当前空白区域是字与字之间的空隙,还是同一个字内结构与结构之间的空隙。比如和前一个空白区域、后一个空白区域的宽度进行比较,或者与空白区域宽度的均值进行比较,一般来说,同一个字内部结构之间的空隙宽度是要比字与字之间的空隙小的。

(2)通过字的宽度/宽高比来判断当前切割出来的“字”是一个完整的字还是一个完整的字的一部分。不同的字体,会有不同的字体宽高比,所以利用字宽是个更好的方法,原理与通过空白区域宽度进行判断的方法类似。

基于区域的图像分割算法可以确保字符重叠时能够将包含多个字符的单个区域分割开来。以直接寻找区域为基础的分割技术,基于区域提取方法有两种基本形式:一种是区域生长,从单个像素出发,逐步合并以形成所需要的分割区域;另一种是从全局出发,逐步切割至所需的分割区域。

文中提出了一种垂直投影结合区域判定算法,算法步骤如下:

(1)使用基于投影的图像算法统计各个位置上的投影点的个数;

(2)将字符大致分割,并计算分割位置的宽度众数;

(3)设置图像中字符的宽度为字符宽度的宽度众数;

(4)比较宽度众数和字符宽度,如果二者相差较小,就说明字符没有重叠,可以直接返回,反之则认为字符重叠,需要进一步分割字符;

(5)计算宽度众数和字符宽度的比值并向上取整数,寻找两个字符中间领域最小的投影点位置作为重

叠字符的分割位置进行分割。

2 字符识别

Tesseract 是 1985 年 Ray Smith 在 HP 实验室研发的一个 OCR 引擎^[13],在 1995 年成为了最准确的三款 OCR 识别引擎之一。曾经在 UNLV 精确度测试中名列前茅。但 1996 年后基本停止了开发。2006 年,Google 邀请 Smith 加盟,重启该项目。Tesseract 目前支持 Windows、Linux 和 Mac OS 等平台^[14],在各个领域应用广泛。

2.1 前期准备

配置所需测试环境组件如表 1 所示。

表 1 设备环境

组件	版本	说明
CMake	3.16.0	
Jdk	1.8	Tesseract 的编译有诸多依赖, Tesseract 依赖于 leptonica, 而 leptonica 又依赖于 png, tiff, jpeg 等基础库
jTessBoxEditor		
Libtiff	4.09	
tesseract	3.5.01	
leptonica	1.76.0	

2.2 训练流程

(1)将训练数据打包成 tif 格式,如果有多个图片可以用 jTessBoxEditors 合并成单个,图片可以利用 Windows 自带的画图工具另存为 tif 格式;

(2)将训练数据生成 box 格式。生成 box 文件的语法格式如下:

```
Tesseract [ lang ]. [ fontname ]. exp [ num ]. tif
[ lang ]. [ fontname ]. exp[ num ] batch. nohop makebox
```

(3)用 jTessBoxEditor 打开 tif 文件进行矫正错误并训练,根据实际情况进行修正,可能会分页需要逐页调整;

(4)显示分析修正的字,生成 .tr 文件。tr 文件格式如下:

```
tesseract [ lang ]. [ fontname ]. exp [ num ]. tif
[ lang ]. [ fontname ]. exp[ num ] nobatch box. train
```

(5)计算字符集,生成一个 unicharset 文件。unicharset 文件格式如下:

```
unicharset _ extractor [ lang ]. [ fontname ]. exp
[ num ]. box
```

(6)定义字体特征文件,新建一个文件 font_properties;

(7)聚集字符特征;

(8)合并 5 个文件。

Tesseract 中主要的数据结构如表 2 所示。

表 2 Tesseract 中的主要数据结构

名称	说明
Page_RES	页面分析结果
BLOCK_RES_LIST	页面分析结果包含块分析结果字段的列表
BLOCK_RES	块分析结果
ROW_RES_LIST	块分析结果包含行分析结果字段的列表
ROW_RES	行分析结果
WERD_RES_LIST	行分析结果包含单词分析结果字段的列表
WERD_RES	是一个可公开访问的成员的集合,用于收集有关单词结果的信息

Init 接口函数对内部的变量进行初始化,patapath 为字符库的路径,language 使用默认的字库,默认的英文字库为“eng”,默认的中文字库为“chi_sim”,在这里也可以添加自己训练的字库。

SetImage 接口函数输入待识别的图片,以及宽度、高度、像素等信息,为 Tesseract 提供最后识别的图片。

3 系统实现

MFC 是微软公司提供的—个 C++ 类库,用于在 C++ 环境下编写应用程序的框架和引擎。MFC 是 WinAPI 与 C++ 的结合^[15]。API,即微软提供的 Windows 下应用程序的编程语言接口,是一种软件编程的规范,但不是一种程序开发语言本身,可以允许用户使用各种各样的第三方的编程语言对 Windows 下应用程序进行开发,使这些被开发出来的应用程序能在 Windows 下运行,比如 VB、VC++、Java、Delphi。编程语言函数本质上全部源于 API,因此用它们开发出来的应用程序都能工作在 Windows 的消息机制和绘图里,遵守 Windows 作为一个操作系统的内部实现,这其实也是一种必要。MFC 不只是一个功能单纯的界面开发系统,它提供的类绝大部分用来进行界面开发,关联一个窗口的动作,但它提供的类中有好多类不与一个窗口关联,即类的作用不是一个界面类,不实现对一个窗口对象的控制(如创建、销毁),而是一些在 Windows(用 MFC 编写的程序绝大部分都在 Windows 中运行)中实现内部处理的类,如数据库的管理类等^[16-17]。

在 MFC 应用中常规的应用程序主要分为 SDI(single document interface,单文档界面)、MDI(multiple document interface,多文档界面)、MTI(multiple top-level windows interface,多顶级窗口界面)和 Dialog 对话框程序几类。

文中主要使用 Dialog 对话框程序来实现文字识

别系统。对话框是图形化用户界面中—种组件,用于对用户输入、选择的信息进行接收,用户可对各种控件进行操作,也可向用户显示响应的操作。对话框中包括编辑框、文本框、列表框、组合框、单选按钮和复选按钮等多种控件,来实现用户的需求,完成用户指定的操作和响应。

对话框的组成:(1)对话框资源:在程序执行过程中,用户可以动态地创建对话框资源,可以使用对话框资源对控件位置和类型、对话框位置和大小进行编辑来配置对话框界面。通过创建对话框资源来添加所需的各种控件,以设置控件的 ID 和内容;(2)对话框类:—般这个类由 CDialog 派生出,然而 CDialog 这个类又是由 CWnd 类派生。

CDialog 成员变量如表 3 所示。

表 3 CDialog 成员变量

名称	类型
m_nIDHelp	UINT
m_lpszTemplateName	LPCTSTR
m_hDialogTemplate	HGLOBAL
m_lpDialogTemplate	LPCDLGTEMPLATE
m_lpDialogInit	void *
m_pParentWnd	CWnd *
m_hWndTop	HWND
m_bClosedByEndDialog	BOOL

系统主要的控制按钮及其说明如表 4 所示。

表 4 控制按钮与说明

ID	成员变量	类型	说明
IDC_PIC			输入原图框
IDC_RES	m_res	CString	输出识别的文字框
IDC_SELMOD	m_selmod	CComboBox	选择识别的字符库
IDC_PATH	m_path	CEdit	选择文件所在的目录
IDC_SEL			打开图片按钮
IDC_OCR			识别图片按钮

系统实现代码如下:

```
CString file;
m_path.GetWindowText(file);

char * path = T2A(file);
tesseract::TessBaseAPI * ap = new tesseract::TessBaseAPI();

string language = "eng_my+chi_my";

if (ap->Init(NULL, language.c_str())) /
{
```



```

cerr<<" Could not initialize. " <<endl;
exit(1);
}

Pix * pic = pixRead( path );
ap->SetImage( pic );

char * put = api->GetUTF8Text();

ifstream in( "../Debug/tmp. txt", ios::binary );
while ( getline( in, t ) ) {
r += UTF8ToGB( t. c_str() ). c_str();
}

MessageBox( _T( " 识别成功" ) );
m_res = A2W( r. c_str() );
UpdateData( false );
ap->End();
pixDestroy( &pic );

```

4 实验结果

识别结果如图 1 所示。



图 1 识别结果

5 结束语

文中介绍了 OCR 识别的过程和相应的模块,主要从文字信息提取和字符识别两大模块进行研究。实现了容错性强、易扩展等特点的文字识别系统。由于汉字字符集庞大导致汉字识别比英文字母的识别难度大,而传统的基于字形结构的方法不能满足当下实际的需求。文中主要结合了国内外关于文字识别领域的文献,针对目前技术现状的需求特点,基于 Tesseract 框架实现了一种文字识别系统。

由于该识别模型的识别率主要与训练的数据量有关,中文字符集较多且字符的结构大致相识,系统依靠 Tesseract 引擎进行识别,实验结果表明识别率并不高。目前文字信息提取的识别样本有限,训练集也远远不够,所以在以后的处理中会逐步加入更多的数据集,以

提高系统的识别率。

参考文献:

- [1] 杨丽娟,李 利. 基于双线性插值的内容感知图像缩放算法仿真[J]. 计算机仿真,2019,36(12):244-248.
- [2] TANG X,GAO X,LIU J,et al. A spatial-temporal approach for video caption detection and recognition[J]. IEEE Transactions on Neural Networks,2002,13(4):961-971.
- [3] 弓耀辉. 图像文字识别中的预处理技术研究综述[J]. 信息通信,2017(9):291-292.
- [4] GLLAVATA J, EWERTH R, STEFI T, et al. Unsupervised text segmentation using color and wavelet features[C]//Image and video retrieval. Dublin, Ireland; Springer,2004:216-224.
- [5] 郭宪军,赵海旭,姚 新,等. 声呐图像分割中的改进 Otsu 算法[J]. 声学电子工程,2018(2):1-4.
- [6] JIANG G, JIE Y. An adaptive algorithm for text detection from natural scenes [C]//Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. Kauai, HI, USA; IEEE,2001:84-89.
- [7] 张婷婷,马明栋,王得玉. OCR 文字识别技术的研究[J]. 计算机技术与发展,2020,30(4):85-88.
- [8] LI Jiang, WAN Heyang, SHANG Songhao. Comparison of interpolation methods for mapping layered soil particle-size fractions and texture in an arid oasis [J]. CATENA,2020,190:104514.
- [9] 李霄霄. 基于 OCR 的字符识别的研究与实现[J]. 科技视界,2017(14):98.
- [10] 刘 琪,李 鑫. 关于 Android 平台的 OCR 文字识别[J]. 数字技术与应用,2017(7):229.
- [11] ZHANG D Q, CHANG S F. A Bayesian framework for fusing multiple work knowledge models in video text recognition[C]//2003 IEEE computer society conference on computer vision and pattern recognition. Madison, WI, USA; IEEE,2003:528-533.
- [12] ZHANG Z, JIN L, KAI D, et al. Character-SIFT: a novel feature for offline handwritten Chinese character recognition [C]//2009 10th international conference on document analysis and recognition. Barcelona, Spain; IEEE, 2009: 763 - 767.
- [13] CHELLAPPA R, BAGDAZIAN R. Fourier coding of image boundaries[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1984,6(1):102-105.
- [14] 韩 萍,刘则徐. 基于灰度级分组的 X 光行李图像增强改进方法[J]. 中国民航大学学报,2011,29(4):23-26.
- [15] 王 芳. 模式识别技术及其在文字识别领域的应用与研究[D]. 西安:西北工业大学,2002.
- [16] 梁 涌. 印刷体汉字识别系统的研究与实现[D]. 西安:西北工业大学,2006.
- [17] 武 桐. 基于图像匹配的汉字识别系统研究与实现[D]. 上海:上海交通大学,2010.