刘新天, 冯杰, 朱明航, 等. 面向识别的长弯曲文本预处理算法[J]. 智能计算机与应用,2024,14(12):10-17. DOI:10.20169/j.issn.2095-2163.241202

面向识别的长弯曲文本预处理算法

刘新天[1], 冯 杰[1], 朱明航[1], 马汉杰[1], 郑雅羽[2]
(1 浙江理工大学 计算机科学与技术学院(人工智能学院), 杭州 310018; 2 浙江工业大学 信息工程学院, 杭州 310023)

摘 要: 光学字符识别(Optical Character Recognition,OCR)是对文本图片进行扫描,然后对图像进行分析处理,获取到其中的文字内容的过程。但是目前的 OCR 算法对于弯曲的长文本普遍识别效果不佳,为此,提出了一种面向识别的长弯曲文本预处理算法,即在文本行识别之前添加长弯曲文本处理模块(Long Curve Text Processing,LCTP),以提升图像中所有文本行识别的准确率。首先,在进行文本区域检测后,获取单条长弯曲文本行并清除干扰信息;其次,根据单条长弯曲文本行的特征计算每条弯曲文本行的关键拐点;进而,使用关键拐点对单条文本行进行切分和融合;最后,将经过切分与融合后的文本行输入文本行识别模型中得到最终识别结果。通过手动采集长弯曲文本图像形成的数据集 Long Curve Text 与目前主流 OCR 框架 PP-OCR 和 Tesseract OCR 进行对比实验可知, *LA*、*MED*、*NED* 指标均有提升,相比于 PP-OCR, *LA* 提升 49.5%, *MED* 和 *NED* 分别降低了 44 115 和 0.182;相比于 Tesseract OCR, *LA* 提升 3.2%, *MED* 和 *NED* 分别降低了 30 282 和 0.125。同时,也在 Long Curve Text 数据集中进行了消融实验以验证本文提出 LCTP 的有效性以及进行了 LCTP 各个结构的时间对比实验以验证本文提出 LCTP 的高效性。结果表明 LCTP 可以提高长弯曲文本识别准确率,总体上可以地获得更加准确、有效的识别结果。

关键词: 长弯曲文本; 干扰信息; 关键拐点; 切分; 融合
中图分类号: TP391.41 **文献标志码**: A **文章编号**: 2095-2163(2024)12-0010-08

Preprocessing algorithm for long curve text recognition

LIU Xintian[1], FENG Jie[1], ZHU Minghang[1], MA Hanjie[1], ZHENG Yayu[2]
(1 School of Computer Science and Technology(School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China; 2 College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: Optical Character Recognition (OCR) is the process of scanning text images, analyzing and processing the images to extract the textual content. However, current OCR algorithms generally have poor performance in recognizing long and curved texts. To address this issue, a pre-processing algorithm called Long Curve Text Processing (LCTP) is proposed, which aims to improve the accuracy of text line recognition in images. Firstly, after performing text region detection, a single long and curved text line is obtained and noise information is removed. Secondly, the key inflection points of each curved text line are calculated based on their features. Subsequently, the text lines are segmented and merged using the key inflection points. Finally, the segmented and merged text lines are fed into a text line recognition model to obtain the final recognition results. A comparative experiment is conducted between the manually collected dataset, Long Curve Text, and the state-of-the-art OCR frameworks, namely PP-OCR and Tesseract OCR. The experiments show improvements in the *LA* (Localization Accuracy), *MED* (Minimum Edit Distance), and *NED* (Normalized Edit Distance) metrics. Compared to PP-OCR, *LA* is improved by 49.5%, while *MED* and *NED* decrease by 44 115 and 0.182, respectively. Compared to Tesseract OCR, *LA* is improved by 3.2%, while *MED* and *NED* decrease by 30 282 and 0.125, respectively. Additionally, ablation experiments are performed on the Long Curve Text dataset to validate the effectiveness of LCTP, and time comparison experiments are conducted to demonstrate the efficiency of the proposed LCTP structures. The results indicate that LCTP can enhance the accuracy of long and curved text recognition, providing more precise recognition results in general.

Key words: long curve text; noise information; key inflection points; segmented; merged

基金项目: 浙江省科技计划项目(2021C01163)。

作者简介: 刘新天(1998—),男,硕士研究生,主要研究方向:文字检测与识别; 朱明航(1998—),男,硕士研究生,主要研究方向:语音合成部署与优化; 马汉杰(1982—),男,博士,副教授,主要研究方向:视频图像传输与处理,机器视觉,情感计算,数据挖掘,嵌入式系统; 郑雅羽(1979—),男,博士,副研究员,主要研究方向:图像处理,嵌入式应用系统。
通信作者: 冯 杰(1980—),男,博士,讲师,主要研究方向:文字检测与识别,视频分析与处理。Email:arlose@zstu.edu.cn。

收稿日期: 2023-06-27

0 引　言

近年来，光学字符识别（OCR）取得了巨大的发展（Wang 等学者[1]，2020），但也面临一些问题。对于书籍扫描场景，由于书籍无法完整展平，因此会出现大量长的弯曲文本，而在长的弯曲文本中干扰信息和文本的弯曲均会影响对该文本识别的准确性。因此，在保证非弯曲文本识别精度不受影响的前提下，提升长的弯曲文本的准确性成为了一项重大挑战。处理长的弯曲文本是一种旨在从文本图像中获取更准确、更全面文本信息的 OCR 算法增强技术，在医疗、教育等领域具有广泛应用（贾智彬等学者[2]，2022；徐倩等学者[3]，2022）。

目前主流的文本检测和识别算法包括 PP-OCR（Du 等学者[4]，2020）和 Tesseract OCR（Hegghammer[5]，2022）等。PP-OCR 是百度开源的一套面向产业应用的 OCR 系统，通过在基础检测和识别模型的基础上推出一系列优化策略，实现了在通用领域的产业级 SOTA 模型，同时支持多种预测部署方案，帮助企业快速应用 OCR 技术。其中，文本检测算法采用了 DBNet（Liao 等学者[6]，2020），文本识别算法采用了 CRNN（Shi 等学者[7]，2016）。PP-OCR 已经发布了 3 个版本，最新的 PP-OCRv3（Li 等学者[8]，2022）在与 PP-OCRv2（Du 等学者[9]，2021）相当速度下，在中文场景下的效果相比 PP-OCRv2 提升了 5%，在英文场景下提升了 11%，并且多语言模型平均识别准确率提升了 5%以上。Tesseract OCR 是由惠普实验室开发，目前由谷歌维护的一款开源 OCR 库。能够识别图片中的文字，并将其转化为可编辑的文本。Tesseract OCR 可以通过 API 接口来使用，无需在本地安装 OCR 软件。该库提供了多种 OCR 功能，包括文字检测和文字识别等。其中，文字检测功能可以检测出图片中的文字位置和边界框，而文字识别功能可将图片中的文字识别出来，并以文本的形式返回。

现如今，对于处理弯曲文本问题，主要采用基于回归的方法和基于分割的方法。最初采用的是基于回归的方法，其中多数方法使用多点坐标表示弯曲文本的边界多边形，并直接预测这些顶点的坐标。例如，CTD、ContourNet、LOMO、TextBoxes、TextBoxes++、PCR 等方法（Dai 等学者[10]，2021；Wang 等学者[1]，2020；Zhang 等学者[11]，2019；Liao 等学者[12]，2017；Liao 等学者[13]，2018）。CTD 提出了一种通过将弯曲文本的边界框定义为具有 14 个顶点的多边形，并预测这 14 个顶点坐标的方法。以此为基础，还使用 Bi-LSTM 来细化预测的 14 个顶点坐标，从而实现基于回归的弯曲文本检测。ContourNet 通过对文本轮廓点进行建模来获取弯曲文本的检测框。由 Adaptive-RPN 模块、LOTM 模块和 Point Rescoring Algorithm 模块组成，用于提取候选框特征、学习水平和竖直方向的文字特征，并使用轮廓点表示文本区域，同时滤除预测中的强单向或弱正交激活，最终以一组高精度的坐标点表示文本轮廓。LOMO 则通过 3 个部分解决弯曲文本问题，包括直接回归器（DR）、迭代优化模块（IRM）和形状表示模块（SEM）。首先，LOMO 使用 DR 分支获取文字的大致区域，然后 IRM 利用 DR 得到的区域通过迭代优化使得边界框更接近真实框（GT）。最后，SEM 模块学习文本实例中的几何属性，包括文本区域、文本中心线和边界偏移量，以获取弯曲文本的检测结果。TextBoxes 是根据目标检测方法 SSD（Liu 等学者[14]，2016）进行修改的，将默认的文本框更改为适应文本方向和宽高比的矩形框，从而更适应长文本检测。TextBoxes++在 TextBoxes 的基础上进行了改进，可以检测多角度的文本，但这 2 种方法在弯曲文本的检测效果上并不理想。尽管基于回归的方法在弯曲文本检测方面取得了一定成效，但方法获得的文本包围曲线通常不够平滑，而且这些模型多数比较复杂，在速度等方面也并不占据优势。

近年来，基于图像分割的方法在文本检测中得到广泛应用。这种方法借鉴了全卷积神经网络的思想，通过对图像中的每个元素进行分类和判断，将文本图像分割为文字区域和非文字区域，并生成文本图像的概率图。通过前后处理的方法，可以得到文本分割区域的矩形框。一些代表性的方法包括 FCENet、DBNet、DBNet++和 RBox 等（Zhu 等学者[15]，2021；Liao 等学者[6]，2020；Liao 等学者[16]，2022；Tang 等学者[17]，2022）。FCENet 通过预测基于傅里叶变换的任意形状的文本包围框，提高了对高度弯曲文本实例的检测精度。然而，对于非弯曲文本的检测，DBNet 表现更加优秀。DBNet 基于分割思想，通常需要使用自定义阈值将获取到的分割概率图转换为二值图像。因此，阈值的确定变得非常重要。DBNet 引入了可微分二值化概念，即对每个像素点进行自适应二值化，二值化阈值由网络学习得到。这种方法完全将二值化步骤与网络一起训练，使得最终输出图像对阈值非常鲁棒。DBNet++在 DBNet 的基础上增加了一个 ASF 模块，用于处理不同尺度的特征，从而获得更好的特征融合效果。

RBox 提出了一种基于 Transformer 的场景文本检测网络,可以在较低语义信息中获取有效的信息,并且无需复杂的后处理步骤,从而提高了检测性能。

上述模型算法都是对检测模型进行优化,以实现更准确的文本行检测和定位。当下,主流的文本检测算法通常基于最小外接矩形或者多边形获取文本行。然而,对于较密集的长弯曲文本来说,仅仅通过文本行的检测和定位,在获取长弯曲文本行的最小外接矩形时会带入其他干扰信息。目前,主流 OCR 对于长弯曲文本的检测与识别结果如图 1 所示。图 1 左侧黑色框内的文本行代表识别正确的文本行,白色框内的文本行代表识别错误的文本行;图 1 右侧文字为识别文本行的内容和置信度。可以观察到白色框内的文本行识别效果差,置信度较低。此外,通过多边形获取长弯曲文本行时文本行本身的弯曲性质也会导致对其进行识别时效果不佳。

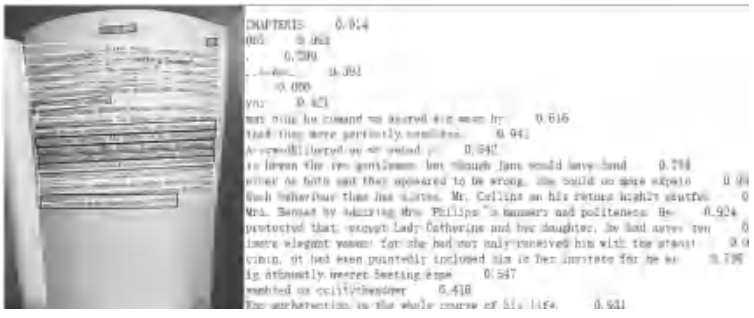


图 1 目前主流 OCR 对于长弯曲文本的检测与识别结果

Fig. 1 Detection and recognition results of current mainstream OCR for long curve text

本文针对长弯曲文本检测与识别面临的挑战,参考了当前主流的 OCR 框架,并提出了一种面向识别的长弯曲文本预处理算法,即文本行识别之前添加长弯曲文本处理模块(Long Curve Text Processing, LCTP)。本文在 Long Curve Text 数据集上进行了实验,以验证所提算法的有效性和可行性。LCTP 通过单独文本行获取和弯曲矫正模块两个方面对现有文本检测与识别算法进行优化。其中,单独文本行获取包含获取单独文本行方法以及清除单独文本行的干扰信息方法,弯曲矫正模块包含文本行关键拐点获取方法以及文本行切分和融合方法。整体流程如图 2 所示。图 2 中虚线框内为本文提出的算法模块,同时本文公开了针对于长条弯曲文本的数据 Long Curve Text。(数据集地址为: https://drive. google. com/drive/folders/1rjydkja4UOaqFdWejsa8VTDclYXsGSSp)

本文第 1 节详细的介绍了所提方法;第 2 节展示了所提出的方法与现有公开数据集上的实验结果;最后总结全文。

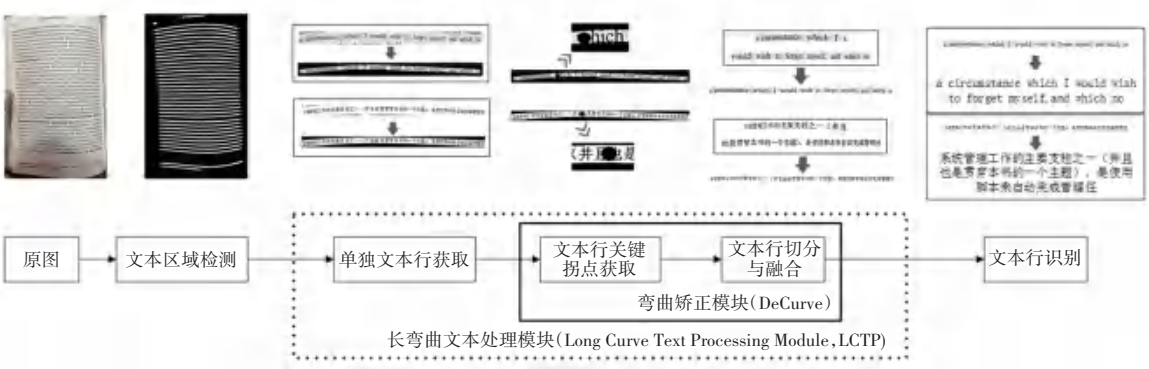


图 2 嵌入 LCTP 模块的文本行检测与识别流程

Fig. 2 Text line detection and recognition process embedded in LCTP module

1 长弯曲文本预处理算法

1.1 单独文本行获取方法

单独文本行获取方法是通过将文本区域检测获取的结果使用基于最小外接矩形的方法来获取单独文本行,并对单独文本行进行干扰信息的清除(Clear Up)。该方法旨在处理长弯曲文本时,去除干扰信息对于文本行识别的影响,为后续的处理和

文字行识别提供更准确的文本行。

清除干扰信息算法运用原始单独文本行和其四点坐标以及文本区域检测得到的分割图 S，根据本文提出的清除干扰信息算法获得清除干扰信息后的单独文本行。算法步骤如下。

算法 1　清除干扰信息算法

输入　原始单独文本行 A，A 中 4 个顶点坐标 $P4$，文本区域检测得到的分割图 S

输出　清除干扰信息后的单独文本行 CA 以及经过处理后的掩膜图像 $M2$

Step 1　使用 $P4$ 在 S 中截取，获得截取后的分割图 S，并保存 S 中每个文字区域的多个轮廓点。

Step 2　对 $S1$ 进行细化操作，拟合 $S1$ 中多条曲线。

Step 2.1　对 $S1$ 中每条曲线，获取每条曲线上的多个点 P。

Step 2.2　对 $S1$ 中的每条曲线，使用 P 进行曲线拟合。

Step 3　确定最长弧长的曲线对应文字区域的多个轮廓点 $P1$。

Step 4　根据 $P1$ 得到算法输出。

Step 4.1　创建与 A 相同形状的全黑掩膜图像 M。

Step 4.2　在 M 中画出 $P1$，并将 $P1$ 内部填充为白色，得到 $M1$。

Step 4.3　对 $M1$ 进行膨胀操作，得到 $M2$。

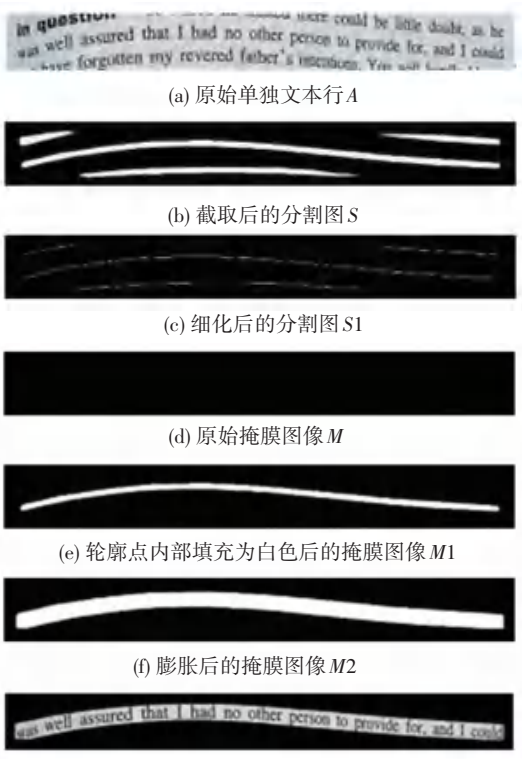
Step 4.4　将 A 与 $M2$ 进行位运算，得到算法输出。

算法中间过程可视化如图 3 所示。经过单独文本行获取方法获取到的单独文本行清除了文本行周围的干扰信息，消除了干扰信息对文本行识别的影响。

1.2　文本行关键拐点获取方法

文本行关键拐点获取是将经过单独文本行获取方法处理后得到的消除干扰信息的单独文字行使用获取关键拐点核心算法获取该文字行的关键拐点，即为拐点。该方法旨在处理长弯曲文本时，提高文字区域的定位精度，为后续的处理和文字行识别提供更准确的文本行。

获取关键拐点核心算法借用最小二乘法的思想并运用在清除干扰信息算法中获取到的每个单独文本行的分割图来获取关键拐点算法。算法步骤如下。



(a) 原始单独文本行 A

(b) 截取后的分割图 S

(c) 细化后的分割图 $S1$

(d) 原始掩膜图像 M

(e) 轮廓点内部填充为白色后的掩膜图像 $M1$

(f) 膨胀后的掩膜图像 $M2$

(g) 清除干扰信息后的单独文本行 CA

图 3　单独文本行获取方法中的可视化结果

Fig. 3　Visualization results from a separate text line acquisition method

算法 2　获取关键拐点算法

输入　清除干扰信息后的单独文本行 CA 以及经过处理后的掩膜图像 $M2$

输出　关键拐点 K

Step 1　对 $M2$ 进行细化操作，得到 $S2$ 并获取细化后曲线上的多个点 P。

Step 2　对 P 进行去重和排序操作，并记录曲线上最左端点 $P1$ 和最右端点 $P2$。

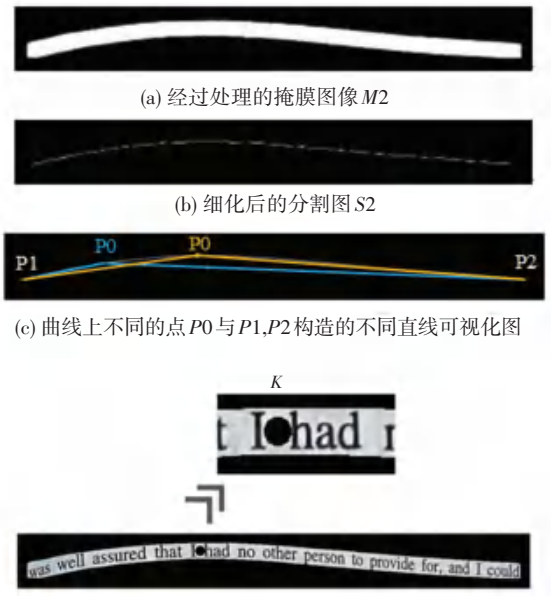
Step 3　根据 P，$P1$，$P2$ 获取算法输出。

Step 3.1　每次以曲线上除了 $P1$，$P2$ 之外的 P 中某点为端点 $P0$ 构造 $P1P0$ 和 $P0P2$ 两条直线。

Step 3.2　计算曲线上 $P0$ 和 $P1$ 之间的点到直线 $P1P0$ 的距离以及曲线上 $P0$ 和 $P2$ 之间的点到直线 $P0P2$ 的距离的总和。

Step 3.3　将总和最小时的 $P0$ 作为 K，若总和小于 $P1$ 到 $P2$ 距离的四分之一，则将 $P1$ 作为 K。

经过大量测试表明，总和阈值取 $P1$ 到 $P2$ 距离的四分之一时，单独文本行本身弯曲程度较小，无需进行文本行切分和融合。算法中间过程可视化如图 4 所示。经过关键点获取方法，单独文本行获取到关键点，为消除文本本身弯曲提供切分文本行时的位置信息。



(a) 经过处理的掩膜图像 $M2$

(b) 细化后的分割图 $S2$

(c) 曲线上不同的点 $P0$ 与 $P1$,$P2$ 构造的不同直线可视化图

(d) 关键拐点 K 可视化图

图 4 关键拐点获取方法中的可视化结果

Fig. 4 Visualization results from the key inflection point acquisition method

1.3 文本行切分和融合方法

文本行切分和融合方法是通过关键拐点使用文本行切分和融合核心算法实现长弯曲文本的弯曲矫正,最终获取经过 LCTP 处理后的单独文本行。该方法旨在处理长弯曲文本时,去除文本行本身在弯曲上对文本行识别的影响,为文字行识别提供更准确的文本行。

文本行切分和融合核心算法利用关键拐点在去除干扰信息后的文本行上进行切分,矫正和融合。

算法 3 文本行切分和融合核心算法

输入 关键拐点 K 以及清除干扰信息后的单独文本行 CA

输出 经过 LCTP 处理后的单独文本行

Step 1 对 CA 使用 K 进行文本行切分

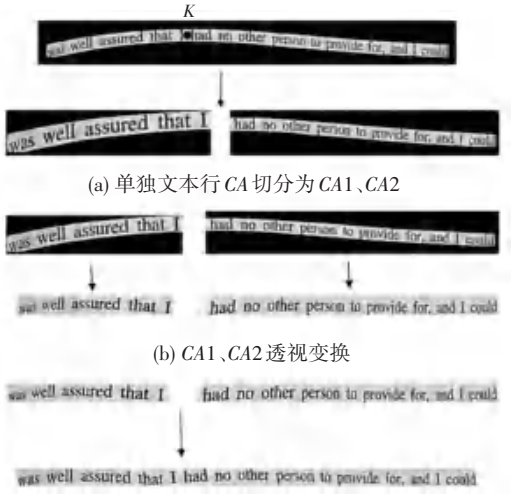
Step 1.1 判断 K 是否为 CA 左端点。

Step 1.2 若是,则将该单独文本行输入文本行识别;若不是,则利用 K 对 CA 进行切分,此时,CA 被切分为左右两个文本行 $CA1$ 和 $CA2$。

Step 2 对 $CA1$ 和 $CA2$ 分别进行透视变换操作。

Step 3 对 $CA1$ 和 $CA2$ 进行左右融合操作,得到算法输出。

算法中间过程可视化具体结果如图 5 所示。经过文本行切分和融合方法,单独文本行清除了由于文本行本身存在弯曲对文本行识别时的影响。



(a) 单独文本行 CA 切分为 $CA1$、$CA2$

(b) $CA1$、$CA2$ 透视变换

(c) $CA1$、$CA2$ 融合为单独文本行

图 5 文本行切分和融合方法中的可视化结果

Fig. 5 Visualization results from text line splitting and fusion methods

2 实验与结果分析

2.1 实验数据及评价指标

为了验证本文提出算法的可行性,使用手动采集的长条弯曲文本数据集 Long Curve Text 进行对比实验,该数据集包括英文书籍图片和中文书籍图片,如图 6 所示。英文书籍图片包含 2 684 条文本,共有 152 155 个字符,中文书籍图片包含 1 448 条文本,共有 42 984 个字符,中英文字符数为 195 139 个字符,平均图片大小为 2 560×1 920 像素,每张图片均包含有需要弯曲处理的文本行和无需弯曲处理的文本行。该数据集中包含 3 个文件夹:Original images 文件夹中存放的是收集的原图;Preprocessing images 文件夹中存放的是经过图像预处理后的图片;GT 文件夹中存放的是每张图片中需要弯曲处理的文本行的真实标签。

(a) 英文书籍图片 (b) 中文书籍图片

(c) 英文文本行

(d) 中文文本行

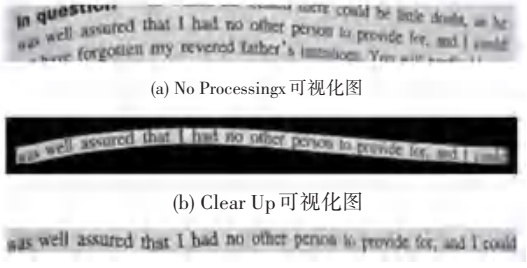
图 6 长弯曲文本数据集中文和英文样例

Fig. 6 Chinese and English examples of long curve text dataset

为了验证本文提出算法的有效性，在长弯曲文本数据集 Long Curve Text 中分别挑选出 473 条，及 758 条存在干扰信息以及本身存在弯曲的中文文本和英文文本来进行消融实验。在这些挑选出的中文文本和英文文本中，中文文本共有 18 142 个字符，英文文本共有 50 692 个字符。将这些数据分别进行如下处理：

（1）去除每条数据中的干扰信息而不消除其本身存在的弯曲，如图 7（b）所示。

（2）对每条数据进行弯曲矫正处理，如图 7（c）所示。



(a) No Processingx 可视化图

(b) Clear Up 可视化图

(c) DeCurve 可视化图

图 7　单条文本处理可视化结果

Fig. 7　Visualization results of single text processing

文本采用的评价指标为准确率 *LA*（Line Accuracy），计算公式为：

$$LA = \frac{C}{A} \tag{1}$$

其中，C 表示识别正确的行数，A 表示所有的行数。*LA* 的计算方式为识别正确的行数占总识别出来所有的行数的比例，该指标是一个简单的度量方法，当文本行中存在一个字符错误，就将该行视为错误的文本行。

字符编辑距离 *MED*（Minimum Edit Distance），计算公式为：

$$MED_{a,b}(i,j) = \begin{cases} \max(i,j), & \text{if } \min(i,j) = 0 \\ \min\begin{cases} MED_{a,b}(i-1,j)+1 \\ MED_{a,b}(i,j-1)+1 \\ MED_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases}, & \text{otherwise} \end{cases} \tag{2}$$

其中，a，b 表示 2 个字符串；i，j 分别表示 a 中前 i 个字符和 b 中前 j 个字符；$1_{(a_i \neq b_j)}$ 表示一个指示函数，表示当 $a_i = b_j$ 时、取 1，当 $a_i \neq b_j$ 时、取 0。*MED* 表示将一个字符串转换为另一个字符串所需的最小编辑操作数，操作包含插入、删除和替换，该指标用来度量 2 个字符串之间的相似度，值越小表示 2 个字符串之间的相似度越大。该指标可以看作为识别错误的字符数量。

归一化字符编辑距离 *NED*（Normalized Edit Distance），计算公式为：

$$NED_{a,b} = \frac{MED_{a,b}}{\max(a,b)} \tag{3}$$

其中，$MED_{a,b}$ 表示字符串 a，b 的字符编辑距离，$\max(a,b)$ 表示字符串 a，b 中最大长度。*NED* 是在 *MED* 基础上，将其除以 2 个字符串的最大长度，从而得到一个归一化的距离，这个距离范围是 [0，1]，值越小表示 2 个字符串越相似。*NED* 可以避免字符串的长度对距离的影响。

2.2　实验设计

本文运用目前 2 个不同主流 OCR 框架：PP-OCR 和 Tesseract OCR，分别使用本文提出的 LCTP，设置相同的参数并在同一环境下进行实验，依次来验证本文提出算法的有效性和可行性。为了排除图片本身的噪声和阴影问题，使用高斯自适应阈值算法来对图片进行预处理。

2.3　实验结果与分析

在消融实验中，验证长弯曲文本干扰信息对文本识别影响的实验结果对比见表 1 和表 2。分析可知，清除文本干扰信息（Clear Up）与未做任何处理的单独文本行（No Processing）比较，前者在 3 个指标中均有不同程度的提升。另外分析可知，清除文本干扰信息（Clear Up）与消除长弯曲文本弯曲程度（DeCurve）比较，后者在 3 个指标中均有大幅度的提升，因此证明了文本的弯曲会影响该文本文本识别的效果。

表 1　清除干扰信息和弯曲矫正前后中文长弯曲文本结果对比

Table 1　Comparison of Chinese long curve text results before and after clearing up noise information and DeCurve

方法	*NED* ↓	*MED* ↓	*LA*/% ↑
NP	0.445	8 187	3.6
CU	0.433	6 524	6.9
DC	**0.118**	**323**	**67.7**

表 2　清除干扰信息和弯曲矫正前后英文弯曲文本结果对比

Table 2　Comparison of English long curve text results before and after clearing up noise information and DeCurve

方法	*NED* ↓	*MED* ↓	*LA*/% ↑
NP	0.629	32 410	0.3
CU	0.400	21 082	4.1
DC	**0.177**	**9 550**	**25.9**

注：NP 表示未做处理；CU 表示单独文本行获取方法；DC 表示弯曲矫正模块；“↓”表示值越小越优；“↑”表示值越大越优；黑体表示最优值。

在本文提出的结构时间对比中,结果对比见表3。单独文本行获取占用时间为整体时间的11.6%,弯曲矫正模块占用时间为整体时间的2.7%,本文提出的 LCTP 占用时间为整体时间的14.3%。因此,证明本文提出的 LCTP 算法的高效性。

表 3 在 Baseline 中添加结构后时间对比

Table 3 Time comparison after adding structures in Baseline

结构	CU	DC	*Speed*/ ms
Baseline			1 660
Baseline+CU	√		1 887
Baseline+DC		√	1 713
Baseline+LCTP	√	√	1 940

注: CU 表示单独文本行获取方法;DC 表示弯曲矫正模块;LCTP 表示 CU + DC

在对比实验中,本文所提出的算法在不同 OCR 框架内的实验结果对比见表4。由表4分析可知,在 Long Curve Text 数据集中, *LA*、*MED* 以及 *NED* 指标均有明显提升。其中,在 PP-OCRv3 中 *LA* 提升了 49.5%, *MED* 下降 44115, *NED* 下降 0.18;由于 Tesseract OCR 中对于图片质量有较高的要求,本文中所使用的文本行多数带有一些噪点,因此在 Tesseract OCR 中 *LA* 仅提升了 3.2%,而 *MED* 下降 30 282、*NED* 下降 0.125。因此可以说明本文所提出的算法在不同的 OCR 框架下,均能提升文本识别的识别效果,证明了本文提出的 LCTP 的有效性。

中英文长弯曲文本经过 LCTP 前后可视化结果对比如图8所示。图8中,Original 表示未经过 LCTP 处理的单独文本行,Result of Original 表示 Original 文本识别的结果,LCTP 表示经过本文提出算法处理后的单独文本行,Result with LCTP 表示经过本文提出算法处理后的单独文本行文本识别的结果。

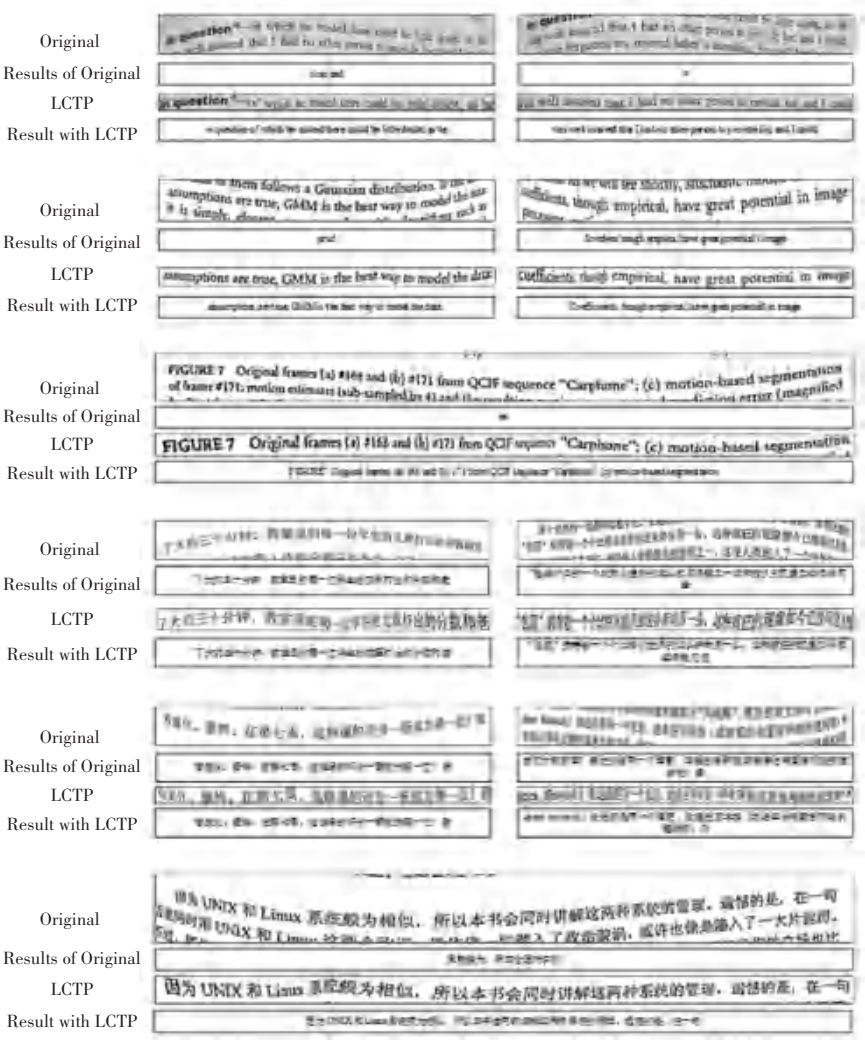


图 8 LCTP 前后可视化结果

Fig. 8 Visualization results before and after LCTP

表 4　长弯曲文本处理前后不同 OCR 框架结果对比

Table 4　Comparison of results between different OCR frameworks before and after long curve text processing

算法	With/Without LCTP(ours)	*NED* ↓	*MED* ↓	*LA*/% ↑
PP-OCRv3	×/√	0.198	47 763	36.9
PP-OCRv3	√/×	**0.016**	**3 648**	**86.4**
Tesseract OCR	×/√	0.296	54 520	12.0
Tesseract OCR	√/×	**0.171**	**24 238**	**15.2**

注：“↓”表示值越小越优；“↑”表示值越大越优；粗体表示最优值

3　结束语

在特定场景下、例如书籍扫描时，长文本的弯曲性对于整个文本图像的识别准确率影响较大，因此提升长弯曲文本识别的准确率具有重要的意义。本文提出长弯曲文本处理模块(LCTP)，分别从消除干扰信息和降低弯曲程度两个方面来降低对文本识别准确率的影响。将 LCTP 应用在不同的主流 OCR 框架，在手动采集的 Long Curve Text 数据集的验证中，均取得了不错的识别效果。

但由于图片存在的噪点等问题，导致在一些主流 OCR 框架中的行准确率 *LA* 提升较低，因此下一步工作尝试解决图片存在的噪点等问题，进一步提升目前主流 OCR 框架的识别效果。

参考文献

[1] WANG Yuxin, XIE Hongtao, ZHA Zhengjun, et al. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 11753-11762.

[2] 贾智彬, 吕学强, 何健, 等. 基于两点法的医疗化验单倾斜校正算法[J]. 计算机与数字工程, 2022, 50(10): 2280-2284.

[3] 徐倩, 郭必然, 贾泓波. 面向票据的 OCR 识别算法研究与实现[J]. 计算机科学与应用, 2022, 12(12): 2778-2787.

[4] DU Yuning, LI Chenxia, GUO Ruoyu, et al. PP-OCR: A practical ultra lightweight OCR system[EB/OL]. (2020-10-15). https://arxiv.org/abs/2009.09941.

[5] HEGGHAMMER T. OCR with Tesseract, Amazon Textract, and Google Document AI: A benchmarking experiment[J]. Journal of Computational Social Science, 2022, 5(1): 861-882.

[6] LIAO Minghui, WAN Zhaoyi, YAO Cong, et al. Real-time scene text detection with differentiable binarization[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Washington DC, USA: AAAI, 2020: 11474-1148.

[7] SHI Baoguang, BAI Xiang, YAO Cong. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(11): 2298-2304.

[8] LI Chenxia, LIU Weiwei, GUO Ruoyu, et al. PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system[EB/OL]. (2022-06-14). https://arxiv.org/abs/2206.03001.

[9] DU Yuning, LI Chenxia, GUO Ruoyu, et al. PP-OCRv2: Bag of tricks for ultra lightweight OCR system[EB/OL]. [2021-10-12]. https://arxiv.org/abs/2109.03144.

[10] DAI Pengwen, ZHANG Sanyi, ZHANG Hua, et al. Progressive contour regression for arbitrary-shape scene text detection[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 7393-7402.

[11] ZHANG Chengquan, LIANG Borong, HUANG Zuming, et al. Look more than once: An accurate detector for text of arbitrary shapes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 10552-10561.

[12] LIAO Minghui, SHI Baoguang, BAI Xiang, et al. Textboxes: A fast text detector with a single deep neural network[C]// Proceedings of the AAAI Conference on Artificial Intelligence. San Francis, USA: AAAI, 2017: 4161-4167.

[13] LIAO Minghui, SHI Baoguang, BAI Xiang. TextBoxes++: A single-shot oriented scene text detector[J]. IEEE Transactions on Image Processing, 2018, 27: 3676-3690.

[14] LIU Wei, ANGUELOV D, ERHAN D, SZEGEDY C, et al. SSD: single shot multibox detector[C] //Proceedings of the European Conference on Computer Vision. Cham: Springer, 2016: 21-37.

[15] ZHU Yiqin, CHEN Jianyong, LIANG Lingyu, et al. Fourier contour embedding for arbitrary-shaped text detection[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 3123-3131.

[16] LIAO Minghui, ZOU Zhisheng, WAN Zhaoyi, et al. Real-time scene text detection with differentiable binarization and adaptive scale fusion[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1): 919-931.

[17] LIU Yuliang, JIN Lianwen, ZHANG Shuaitao, et al. Detecting curve text in the wild: new dataset and new solution[EB/OL]. (2017-12-06). https://arxiv.org/abs/1712.02170.

[18] TANG Jingqun, ZHANG Wenqing, LIU Hongye, et al. Few could be better than all: Feature sampling and grouping for scene text detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 4563-4572.

[19] 王建新, 王子亚, 田萱. 基于深度学习的自然场景文本检测与识别综述[J]. 软件学报, 2020, 31(5): 1465-1496.