Hans Journal of Data Mining 数据挖掘, 2020, 10(2), 97-110
Published Online April 2020 in Hans. http://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2020.102010

Hans汉斯

Research on Identification of Printed Mathematical Formula

Yanran Lin, Lihong Yang

School of Mathematics, South China University of Technology, Guangzhou Guangdong
Email: 494402963@qq.com

Received: Feb. 25th, 2020; accepted: Apr. 1st, 2020; published: Apr. 8th, 2020

Abstract

With the development of the e-book industry, the application of OCR technology is becoming more and more widespread, but formula recognition has not been popularized. The main reason is the lack of regularity of formula. At the same time, the paper review has become more and more strict, and it has also proposed new formula recognition claim. This paper mainly studies the recognition of printed mathematical formulas in the form of pictures. For character segmentation, a combination of projection line segmentation and connected domain segmentation is used. For character recognition, template matching is adopted, and then the relative positions of characters are used for structural analysis. The segmentation method adopted by the system improves the segmentation accuracy while ensuring the operation speed, and has a good effect on the segmentable characters. The adopted character recognition method makes full use of the regularity of printed mathematical characters. The algorithm is simple with the low computational complexity, and the recognition accuracy is higher than 97%. The common formula structure is taken into account in structural analysis, and the classification discussion is without repetition or omission. The system can identify clear printed mathematical formulas, and computational complexity and character sticking problems need to be further optimized.

Keywords

Connected Domain Segmentation, Template Matching, Structural Analysis

印刷体数学公式识别研究

林妍然，杨立洪

华南理工大学数学学院，广东 广州
Email: 494402963@qq.com

收稿日期：2020年2月25日；录用日期：2020年4月1日；发布日期：2020年4月8日

文章引用: 林妍然, 杨立洪. 印刷体数学公式识别研究[J]. 数据挖掘, 2020, 10(2): 97-110.
DOI: 10.12677/hjdm.2020.102010

摘　要

随着电子书产业的发展，OCR技术(光学字符识别)的应用越来越广泛，但公式识别还未普及，主要原因是公式本身缺乏规律，同时，论文查重越来越严，也给公式识别提出了新的要求。本文主要研究以图片形式呈现的印刷体数学公式识别，字符分割方面采用了投影行分割和连通域分割相结合的方法，字符识别采用了模板匹配法，然后利用字符的相对位置进行结构分析。系统采用的分割方法在保证运算速度的同时提高了分割精度，对可分割的字符效果好。采用的字符识别方法充分利用了印刷体数学字符的规律性，算法简单，运算复杂度较低，识别精度高于97%。结构分析上考虑了常见的公式结构，分类讨论不重不漏。系统能识别清晰的印刷体数学公式，运算复杂度和字符粘连问题还需要进一步优化。

关键词

连通域分割，模板匹配法，结构分析

1. 引言

信息技术的进步，促进了电子书产业的蓬勃发展，电子书内容的识别与应用的需求也越来越高。OCR技术，即对文本资料进行扫描后对图像文件进行分析处理，获取文字及版面信息的过程，其提出至今已有近 90 年，在中文材料和英文文章中均已得到广泛的应用。然而，在对数学公式的识别上还未普及，论文查重中还未能对数学公式进行识别，而这正是一些理工科论文的核心内容，同时，当前国内对论文查重的要求越来越高。因此，数学公式的识别与检索对于信息快速共享和预防学术不端现象具有非常重要的现实意义[1]。

与普通文本识别不同的是，数学公式的分布呈二维状态，其分布无序，而普通文本的分布一般呈现一维状态，同时，对于英文单词的识别，还可通过语义与单词规律进行校正。这使得数学公式的识别更加困难。一般而言，数学公式的识别分为图像预处理、字符分割、字符识别、结构分析、错误纠正、输出几个步骤。因此，本文将主要从前面五个部分进行介绍。

2. 印刷体数学公式识别研究

本文的研究对象是以图像形式存储的印刷体数学公式，一般由 PDF 文件或对公式截图获得。因此，在进行字符分割与识别之前，需要进行图像预处理。

2.1. 图像预处理

首先将 RGB 图像进行灰度化处理，常见的灰度化处理有分量法、最大化法、平均法及加权平均法。系统采用 MATLAB 的内置函数 rgb2gray，加权平均法：

$$X = 0.2989R + 0.5870G + 0.1140B$$

其中 X 为灰度图像的像素值，R、G、B 分别是红绿蓝三种颜色的亮度值。

对得到的灰度图像二值化处理，图像的二值化处理将直接关系到识别的成功与否，同时还关系到字

符分割，字符的粘连与分离。常见的二值化阈值选取方法可分为静态和动态，局部和全局，静态常见的有均值法，直方图双峰法等。通过对几种方法的效果对比，最终选取直方图双峰法，并通过多次测试，将二值化阈值设为 190，为方便计算，将 0~255 转换为 0~1。

2.2. 字符分割

常见的字符分割方法有投影分割法和连通域分割法。

2.2.1. 投影分割法

投影分割法分为竖直投影分割和水平投影分割，以水平投影分割为例，其通过计算每行没有像素点的数量，对一行中没有像素点的位置进行切割，通过水平投影，可以计算一幅图像的行数，及每一行的高度。水平投影分割的结果见图 1。

$$\frac{\sin v\cos v}{\sqrt{1+\cos^2 v}}$$

$$\frac{\sin v\cos v}{\sqrt{1+\cos^2 v}}$$

Figure 1. Horizontal projection segmentation
图 1. 水平投影分割示例

竖直投影分割同理，其通过计算每一列没有像素点的数量，对一列中没有像素点的位置进行切割，通过竖直投影分割，可以计算一幅图像的列数，及每一列的宽度。竖直投影分割的结果见图 2。

$$=\lim_{x\to 1}\frac{F(x)-F(1)}{x-1}$$

$$=\lim_{x\to 1}\frac{F(x)-F(1)}{x-1}$$

Figure 2. Vertical projection segmentation
图 2. 竖直投影分割示例

可见，投影分割法操作简单，计算复杂度低。然而，其无法分割嵌套结构，如图 3：

$$\sqrt{1+\cos^2 v}$$

Figure 3. Projection segmentation cannot segment nested structure
图 3. 投影分割无法分割嵌套结构例子

而连通域分割法可以处理这种字符。

连通域分割法[2]，指自上而下搜索第一个有像素的点(x_0, y_0)，集合$X = \{(x_0, y_0)\}$。继而搜索它附近的八连通邻域，若其八连通邻域的某个点(x_1, y_1)处有像素，则将这个点纳入集合X中，即$X = \{(x_0, y_0), (x_1, y_1)\}$。继续搜索集合X内其他点的八连通邻域(如图4)，一轮结束后，将八连通邻域有像素的点的坐标存储在X中，将有像素的点清除，再继续迭代搜索集合X中其他点的八连通邻域，将有像素的点的坐标存储在X中，直到X不再更新。

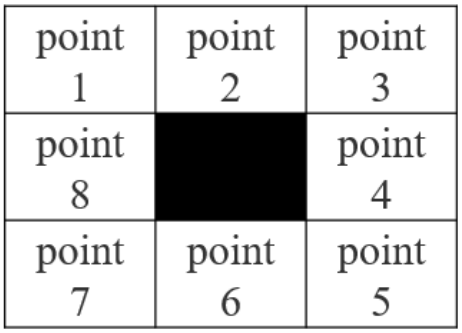

Figure 4. Eight connected neighborhoods
图 4. 八连通邻域

算法过程见表1。

Table 1. Connected domain segmentation algorithm
表 1. 连通域分割算法

Input: 公式图片 F
Iteration:
1. 扫描 F 的第一个有像素的点 $\text{point0} = (x_0, y_0)$， $X = \{(x_0, y_0)\}$；
2. 寻找 point0 的八连通领域 point1~point8；
3. 若其八连通领域中存在点有像素，则将该点纳入集合 X 中。假设 point2，point5 有像素，如 $\text{point2}(x_2, y_2) == 0$ (1 代表白色，0 代表黑色，即为像素点)，则更新 $X = \{(x_0, y_0), (x_2, y_2), (x_5, y_5)\}$；
4. 对图片 F，继续循环迭代 X 中的其他像素点，执行 2~3，直至 X 不再更新；
5. 将 X 中的像素点置为 1，如 $\text{point2}(x_2, y_2) = 1$。清空 X，重新扫描图片 F，寻找 point0，执行 1~4，直至 F 全为 1 (即空白)。

执行算法的结果见图5：

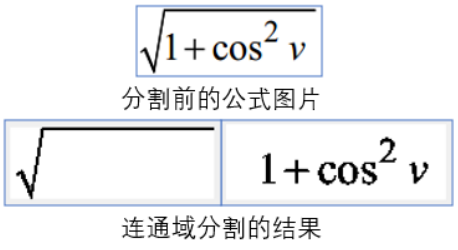

Figure 5. Connected domain segmentation
图 5. 连通域分割示例

2.2.2. 系统字符分割法

由上可以得知，投影分割法操作方便，计算复杂度低，节约运算资源，但不能处理嵌套结构，而连通域分割法处理灵活，可处理多种复杂公式结构，缺点是迭代循环造成计算复杂度高。因此，结合两种算法，系统决定先对输入的结构利用水平投影分割法分割成多行，再分别对每一行利用连通域分割法分割。使得在处理多种复杂结构的同时降低计算复杂度。

2.2.3. 系统字符分割结果

通过两种方法相结合的分割方法，能够结合两者的优点，分割效果较好，且节约运算资源，能够分割所有理论上能分割的字符。但不能分割粘贴字符，造成粘贴字符不能分割的原因有两种：

一种是组合字符，如积分符合与积分上下限，如图 6 所示：

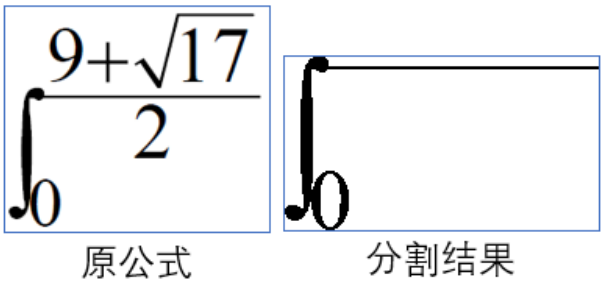

Figure 6. Combining characters cannot be segmented
图 6. 组合字符无法分割示例

由于原公式中积分符号与上下限为组合符号，故视为不可分割，对于这种类型的字符，采用添加相关字符进入模板库，主要有：$\int_0^\infty$ 型(主要是 0~9，−，∞的有序组合)，dx 等。

或平行字符，如图 7。

Figure 7. Conglutination caused by parallel character
图 7. 平行字符粘连示例

另外一种是由于图像预处理造成的字符粘连，如图 8：

Figure 8. Conglutination caused by image processing
图 8. 图像预处理造成的字符粘连示例

而最常见的字符粘连，是图 9 这种情况：

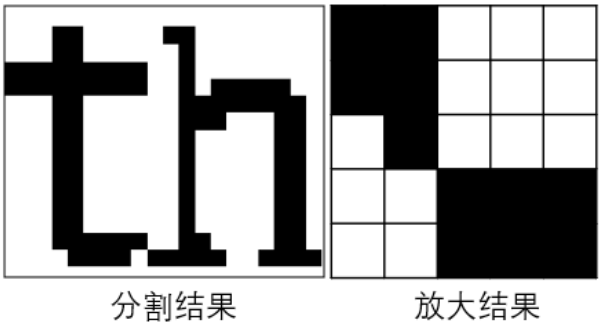

Figure 9. Commonconglutination
图 9. 常见的字符粘连示例

2.3. 字符识别

由于研究对象是印刷体数学公式，其具有固定结构，不似手写字符般多变。因此，在识别之前，结合实际，对字符建立模板库。

2.3.1. 模板库建立

印刷体数学公式具有固定的形式，所以，在建立模板库时，以精准、全面为目标，考虑常见的字体等形式，通过以下规则建立模板库，见表 2。

Table 2. Requestof drawing template library
表 2. 模板库构建要求

类型	具体内容
种类	大写字母、小写字母、数字、希腊字母、特殊(其他)字符(含粗体、斜体、斜体粗体)
字体	宋体(中文正文)、仿宋、Times New Roman、Calibri (西文正文)、Calibri Light (西文标题)、Cambria Math、mathtype 字体、latex 字体
字号	六号、五号、小四、四号、三号、二号、小初

为尽量减少信息损失，建立模板库的过程见图 10：

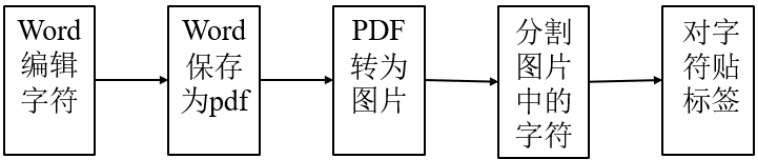

Figure 10. Process of drawing template library
图 10. 模板库建立过程

2.3.2.字符识别算法

常见的 OCR 识别算法有模板匹配法，支持向量机法，人工神经网络法，但人工神经网络要求模板数据量较大，适用于手写字符识别[4]。而识别对象是有固定规律的印刷体，所以系统使用模板匹配法。

1) 特征提取

特征提取指，预处理后，提取公式字符的特征，并用数值进行表示，要求该特征应尽量体现该字符区别于其他字符的不太特点，且体现的特征尽量不交叉。对公式字符提取的特征常见的有：宽高比、孔洞数、网格特征、穿越线等。由于本系统的字符库建立力求精准全面，所以，在考虑特征提取时，考虑

对字符大小归一化到 16 × 16 的 256 维数据作为特征。

字符大小归一化即图片的缩放。常见的放大缩小算法有最近邻插值算法，双线性插值算法，双三次插值算法。为提高缩放精度，采用双三次插值算法。

双三次插值[3]是二维空间中最常用的插值方法。假设源图像 A 大小为 $m\times n$，缩放后的目标图像 B 的大小为 $M\times N$。根据缩放比例可以得到 $B(X,Y)=A(x\times(M/m),y\times(N/n))$。其可能为浮点数，考虑任意像素点的坐标 $P(x+u,y+v)$。其中 x,y 分别表示整数部分，u,v 表示小数部分。考虑 P 点附近的 16 个像素点的位置(如图 11)，用 $a(i,j),(i,j=0,1,2,3)$ 表示。

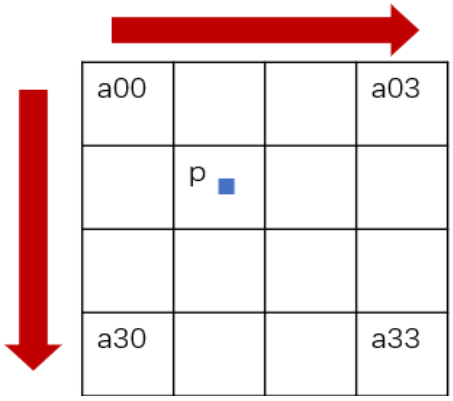

Figure 11. Locationof 16 neighborhoods
图 11. 16 邻域位置

计算 a_{ij} 的权值 k_{ij}。以 $P_{mn}(x+u,y+v)$ 为例，首先计算

$$k_{m0}=W(1+u),k_{m1}=W(u),k_{m2}=W(1-u),k_{m3}=W(2-u);$$

$$k_{n0}=W(1+v),k_{n1}=W(v),k_{n2}=W(1-v),k_{n3}=W(2-v);$$

则 $k_{ij}=k_{mi}*k_{nj}$。其中 $W(x)$为 BICUBIC 函数，如下：

$$W(x)=\begin{cases}(a+2)|x|^3-(a+3)|x|^2+1, & |x|\le 1\\ a|x|^3-5a|x|^2+8a|x|-4a, & 1<|x|<2\\ 0, & \text{其他}\end{cases}$$

其中，a 取−0.5。

则 P 点像素值为 $B(X,Y)=\sum_{i=0}^{3}\sum_{j=0}^{3}A(i,j)k_{ij}$。

为了提高识别精度，在归一化前后进行以下优化/处理：

- 归一化之前，将字符通过加白边的形式扩展为方阵，如图 12：

Figure 12. Extendto square matrix
图 12. 扩展为方阵

- 对于类似于“,”等大小小于等于5*5的字符，归一化到5*5，这样能防止小字符放大后信息损失太多；
- 以灰度值存储归一化后的数据。

2.3.2. 字符识别

采用模板匹配法进行字符识别，即计算待识别字符 X 的16*16维数据与模板库中的字符 Y 的欧式距离，选取距离最小的模板库字符的类别作为 X 的类别。即：

$$d\min\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{\left|X(:,:)-Y_{ij}(:,:)\right|^2}\right)_{\min}$$

$$\text{label}_X = 1_{\left(d_{ij}=d_{\min}\right)}\times i$$

其中 i 指第 i 种字符，共 m 种字符；j 指字符 i 的第 j 种形式，共 n 种。

2.3.3. 字符识别结果

通过随机挑选一部分公式数据(公式量大，覆盖字符多)，进行字符识别，得到结果见表3：

Table 3. Character recognition experiment
表 3. 模板库构建要求

图片	总数	识别正确数	识别错误数	正确率	备注
1	223	208	15	93.27%	|错误识别为l (66.7%)，1错误识别成l，导数符号识别成l
2	307	294	13	95.77%	|错误识别为l，i下半部分识别成导数符号，t,1错误识别为l,n识别成Π。
3	126	122	4	96.83%	1错误识别为l (100%)
4	161	153	8	95.03%	1错误识别为l (100%)
5	145	140	5	96.55%	1错误识别为l (100%)
6	347	333	14	95.97%	1错误识别为l (100%)
7	264	255	9	96.59%	1错误识别成l,0错误识别成O
8	239	233	6	97.49%	1错误识别为l,j识别为3
9	221	212	9	95.93%	1错误识别为l (100%)
10	175	172	3	98.29%	1错误识别为l (100%)
11	431	431	0	100.00%	/

从以下实验中可以发现，识别错误的情况主要出现在字符l上，一方面，数字1与小写字母l相似度太高，另一方面Calibri字体下l与|雷同。

2.4. 结构分析

结构分析，即将识别后的字符，利用字符的相对位置，建立公式规则，重组为公式，并用latex语言进行表示。

2.4.1. 位置信息

由于结构分析主要利用的信息是字符的类别和字符的二维结构，因此，在进行结构分析之前，需要对字符的位置进行标记，由于公式的分割是先行投影分割后连通域分割，因此对字符的储存方面采用树结构，每一行为结点，行内分割后的单独字符作为子节点，如图13所示：

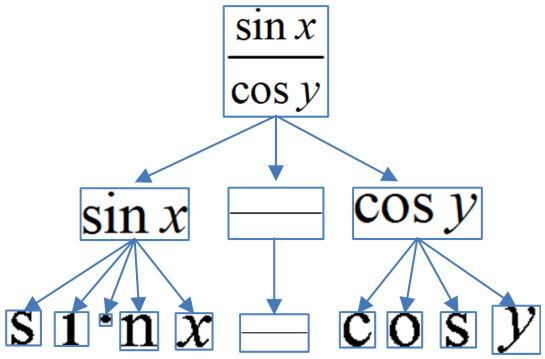

Figure 13. Row formula tree
图 13. 行公式的树结构

首先记录行节点的行信息：该行所在的上下界，如图 14 所示：

$$\begin{matrix} m_1 \\ \\ m_2 \end{matrix} = \prod_{i=1}^{n}\left(1+\frac{b_m(x_i,\theta')-F(x_i+\epsilon|\theta')}{F(x_i+\epsilon|\theta')-F(x_i-\epsilon|\theta')}\right) \quad (m_1,m_2)$$

$$\begin{matrix} m_3 \\ \\ m_4 \end{matrix} + \frac{F(x_i-\epsilon|\theta')-a_m(x_i,\theta')}{F(x_i+\epsilon|\theta')-F(x_i-\epsilon|\theta')}\Bigg)^{-1} \quad (m_3,m_4)$$

Figure 14. Boundary line location of row formula
图 14. 行公式的上下界位置

然后记录单个字符的位置信息，以水平向右和水平向下建立坐标系，记录字符的左上角和右下角坐标(x_1,y_1,x_2,y_2)。如图 15 所示：

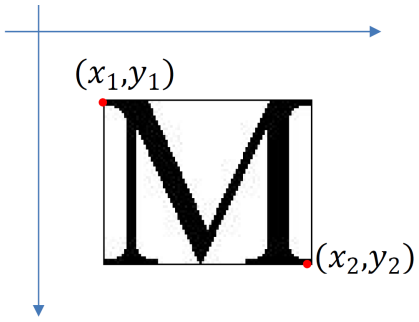

Figure 15. Location imformation of single character
图 15. 单个字符的位置信息

结构分析时分两种情况讨论：组合字符和独立字符。首先讨论组合字符。

2.4.1. 组合字符的结构分析

组合字符指由两个或两个以上字符组成的字符，如：“=”，“≈”，“÷”，“±”，“≤”，“，”，“I”，“j”，“!”等。对于这种字符，需要根据字符之间的相对位置确定实际字符。以分析“=”为例，若遇到字符“–”或“ – ”，则判断下一个字符是否是“–”或“ – ”，且下一个字符处于上一个字符的正下方，代码见图 16：

```
%解决等号问题=================================================================
if ((data{1,i}{6,j} == "-" && j < n && data{1,i}{6,j+1} == "-") || ...
   (data{1,i}{6,j} == "\frac" && j < n && data{1,i}{6,j+1} == "-")|| ...
   (data{1,i}{6,j} == "-" && j < n && data{1,i}{6,j+1} == "\frac") ...
    ||(data{1,i}{6,j} == "\frac" && j < n && data{1,i}{6,j+1} == "\frac")) ...
    &&(data{1,i}{2,j}(2) == data{1,i}{2,j+1}(2)) &&(data{1,i}{2,j}(4) == ...
    data{1,i}{2,j+1}(4)) && (data{1,i}{2,j}(3) < data{1,i}{2,j+1}(1))
    data{1,i}{6,j} = '=';data{1,i}{6,j+1} = '';
end
```

Figure 16. The code of judging "="
图 16. "="判断代码

2.4.2. 独立字符的结构分析

对于独立字符，主要考虑字符之间的相对位置，及特定公式的结构，常见的有分数结构，根号结构，积分结构，乘方/下标(上下限)结构，极限结构。如图 17：

$$\frac{x}{2y},\sqrt{2xy},x^3,y_2,\left[2x-3\right]_0^2$$

$$\int_1^2 2xdx,\log_2,\lim_{x\to\infty},\bar{x},\prod_{i=1}^{10}$$

Figure 17. Familiarformulastructer
图 17. 常见公式结构

考虑分数结构。

1) 分数公式分析

对于分数结构，首先检测是否有"–"或"–"，(下记"–")若存在，记录"–"的左右边界 y_1,y_2，接着判断分数线上下位置是否存在字符，值得注意的是这里要按分数线与分子分母是否处于同一行分情况讨论(这里的同一行是指投影行分割的行)，按照系统的分割法，投影行内字符将按自上而下顺序存储，如投影行数据 $\frac{x}{2y}$ 的字符储存顺序是：$-,2,x,y$。因此可以按顺序检索。判断依据参考图 18：

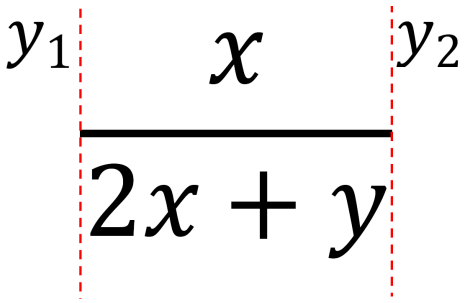

Figure 18. Familiarformula structer
图 18. 常见公式结构

若分数线上下位置存在字符，则将字符组合为分数，用 latex 语言表示为'\frac{x}{$2x+y$}。原来分子分母的字符清空，并对原来分子分母内的字符迭代公式规则判断。

2) 根号公式分析

根号公式相对比较简单，因为根号公式一定处于同一行，且字符之间的相对顺序有序，遇到根号时，首先记录根号的右边界位置 y，判断其后的字符的右边界是否小于 y，若是，则为被开方数，直至大于右边界 y，即图 19 所示：

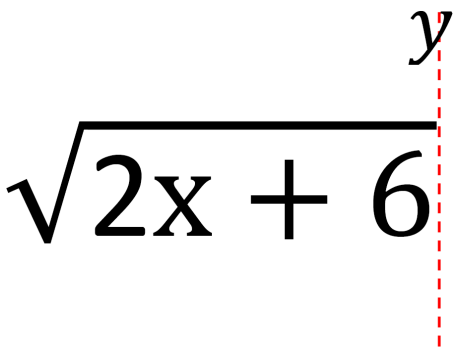

Figure 19. Right boundary of radical
图 19. 根号右边界

3) 上下标公式分析

乘方，下标，积分上下限属于同一种结构，可以类比分析，这里以乘方为例，进行分析。以 x^3 为例(见图 20)，当检索到字符“3”(记为 B，其前一个字符为 A)时，判断字符“3”是否满足以下条件：

$$\begin{cases} B\cdot y_1 > A\cdot y_2 \\ B\cdot x_1 < A\cdot x_1 \\ B\cdot x_2 < A\cdot x_1 + \left(A\cdot x_2 - A\cdot x_1\right)/2 \end{cases}$$

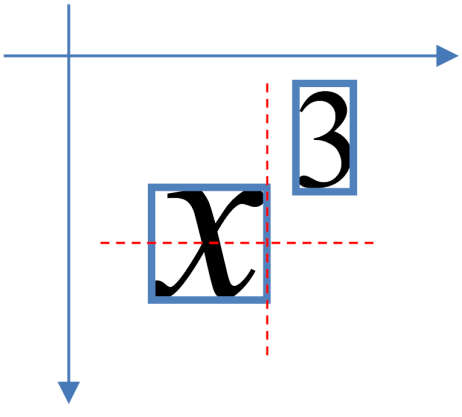

Figure 20. Locationanalyse of superscript
图 20. 上标位置分析

值得注意的是 B 的下一个字符也可能在上标位置，因此，在满足上述条件下，要对 B 的下一个字符，继续判断上述条件。

4) 标注符号结构分析

在概率统计中，可能会出现类似于平均数$\bar{x}$这样的标注符号，下面以最常见的平均数结构为例，在遇到“–”时(在排除“=”和分数情况后)只需判断其正下方的位置是否有字符，且二者高的比例小于 1:3 即可。见图 21：

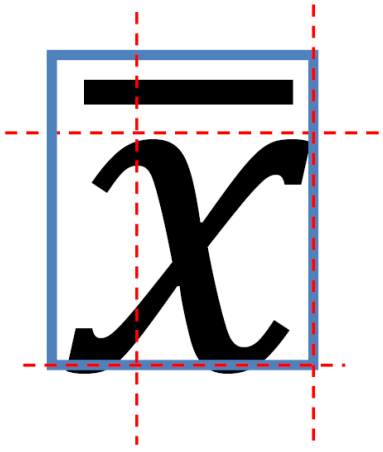

Figure 21. Locationanalyse of markedsymbol
图 21. 标注符号位置分析

5) 上下结构特殊公式分析

上下结构的特殊公式指形如图 22 中所示的上下结构的公式：

$$\sum_{i=0}^{10} , \lim_{x\to\infty} , \prod_{i=0}^{n} , \bigcup_{i}$$

Figure 22. Formulaof up-down structer
图 22. 上下结构公式

以极限符号为例进行分析。若遇到连续的三个字符“lim”时，判断其正下方是否有字符“→”，且判断两者之间的相对位置，一般有以下两种情况，见图 23：

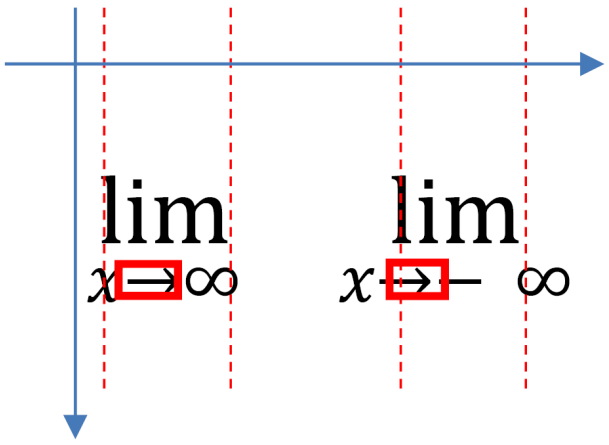

Figure 23. Relative location analyse of “→” in “lim”
图 23. “lim”中“→”的相对位置分析

即：

$$when \quad to.x_1 < \max(l.x_2, i.x_2, m.x_2),$$
$$count\ if$$
$$l.y_1 < to.y_1 < m.y_2 \ or \ l.y_1 < to.y_2 < m.y_2$$

需要注意“lim”与“→”是否在同一个投影行，与极限数的个数。

2.4.3. 公式识别结构

通过以上过程，对部分公式样本进行实验，得到的识别结构较好，举例如下(见图 24 和图 25)：

$$\lim_{x\to\infty}\frac{\ln(1^x+2^x)}{x}=\lim_{x\to\infty}\frac{\frac{1}{1^x+2^x}(1^x\ln 1+2^x\ln 2)}{1}$$
$$=\lim_{x\to\infty}\frac{2^x\ln 2}{1^x+2^x}=\lim_{x\to\infty}\frac{2^x(\ln 2)^2}{1^x\ln 1+2^x\ln 2}=\ln 2$$
$$\lim_{x\to\infty}(1^x+2^x)^{1/x}=\lim_{x\to\infty}e^{\ln y}=e^{\ln 2}=2$$

原公式

$$\lim_{x\to\infty}\frac{m(1^x+2^x)}{x}=\lim_{x\to\infty}\frac{\frac{1}{1^x+2^x}(1^x ln1+2^x ln2)}{1}$$
$$=\lim_{x\to\infty}\frac{2^x ln2}{1^x+2^x}=\lim_{x\to\infty}\frac{2^x(ln2)^2}{1^x ln1+2^x ln2}=ln2$$
$$\lim_{x\to\infty}(1^x+2^x)^{1/x}=\lim_{x\to\infty}e^{lny}=e^{ln2}=2$$

系统公式识别结果

Figure 24. Eg.1 Formula recognization
图 24. 公式识别示例 1

$$A=\int_{-3}^{-2}(-x^2-5x-6)dx$$
$$=\left[-\frac{1}{3}x^3-\frac{5}{2}x^2-6x\right]_{-3}^{-2}$$
$$=\left(\frac{8}{3}-10+12\right)-\left(9-\frac{45}{2}+18\right)=\frac{1}{6}\approx 0.17$$

原公式

$$A=\int_{-3}^{-2}(-x^2-5x-6)\,dx$$
$$=\left[-\tfrac{1}{3}x^3-\tfrac{5}{2}x^2-6x\right]_{-3}^{-2}$$
$$=\left(\tfrac{8}{3}-10+12\right)-\left(9-\tfrac{45}{2}+18\right)=\tfrac{1}{6}\approx 0.17$$

系统公式识别结果

Figure 25. Eg.2 formula recognization
图 25. 公式识别示例 2

3. 评价与展望

本系统采用的分割方法在运算速度和分割精度之间取得平衡，利用印刷体字符的规律性，以精准为模板建立字符模板库，并采取简单高效的模板匹配法进行识别，识别精度高，对于识别错误的字符追根溯源，找到识别错误的原因(字符高度相似如 1 和 l，模板库字符形态相似)，并针对性处理，采用的公式规则逻辑性强，覆盖性全，识别效果较好。但是由于字符粘连的情况无法分割导致识别错误的情况发生，因此对识别效果产生一定影响。同时对印刷体书写原因造成的字符粘连采用的扩充模板库方法，还未涵盖所有情况。因此，本系统对字符间距非特别小的情况识别较好，同时识别时间还需要进一步优化。

参考文献

[1] 李丹, 秦玉平. 数学公式识别与检索研究综述[J]. 渤海大学学报(自然科学版), 2015, 36(1): 44-48.

[2] 张自强. 印刷体文档中的数学公式识别算法的研究[D]: [硕士学位论文]. 马鞍山: 安徽工业大学, 2016.

[3] CSDN. 图像缩放之双三次插值算法[EB/OL]. https://blog.csdn.net/qq_29058565/article/details/52769497, 2016-10-09.

[4] 余圣新, 夏成蹊, 唐泽恬, 丁召, 杨晨. 基于改进 Inception 卷积神经网络的手写体数字识别[J]. 计算机应用与软件, 2019, 36(12): 143-149.