文章编号：1003-0077(2021)02-0133-08

基于加权贝叶斯的脱机手写阿文单词识别

许亚美，何继爱

（兰州理工大学 计算机与通信学院，甘肃 兰州 730050）

摘 要：针对手写阿拉伯单词书写连笔，且相似词较多的特点，该文提出一种新的脱机手写文字识别算法。该算法以固定组件为成分拆分阿拉伯单词，构建自组件特征至单词类别的加权贝叶斯推理模型。算法结合单词组件分割、多级混合式组件识别、组件加权系数估计等，计算单词类别的后验概率并得到单词识别结果。在 IFN/ENIT 库上的实验，获得了 90.03%的单词识别率，证实组件分解对笔画连写具有鲁棒性，组件识别能提高相似词的辨别能力，而且该算法所需训练类别少，易向大词汇量识别扩展。

关键词：手写文字识别；阿拉伯文；单词识别；加权贝叶斯

中图分类号：TP391 **文献标识码**：A

Offline Handwritten Arabic Word Recognition Based on Weighted Bayesian

XU Yamei, HE Ji'ai

(School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu 730050, China)

Abstract: A new offline handwritten Arabic word recognition algorithm is proposed to deal with its connected writing strokes and more similar words. The algorithm first establishes a structure model with fixed graphemes for each Arabic word category to be recognized, then segments the word samples into graphemes. Then a weighted Bayesian inference model is constructed from the grapheme features to word categories. The word recognition results are obtained by calculating the posterior probabilities of word categories. On the IFN/ENIT database, the proposed algorithm achieves as high as 90.03% accuracy.

Keywords: handwritten script recognition; Arabic; word recognition; weighted Bayesian

0 引言

在关于手写文字识别的文献中，阿拉伯文字识别的研究逐渐受到关注[1-4]。阿拉伯文，简称阿文，是西亚阿拉伯地区和伊斯兰教信仰者使用的文字，使用者来自不同的国家与民族。阿文字母不能独立运用，字母相连书写成单词后才有语义，因此单词识别具有实际意义[4]。

阿文共有 28 个辅音字母、1 个复合字母和 12 个元音符号，元音字母是在辅音字母上叠加元音符号而构成，每个字母根据在词中不同位置，有独立、前连、双连、后连中的 2～4 种字符格式，共演变成 100 个字符[5]。

阿文单词内部各字母粘连书写，但由于阿文有个别字母（如 د、ا 等）没有双连和后连形式，所以在单词中这些字母出现时会出现断开书写，从而构成阿文单词特有的结构规则[4]，如图 1 所示①，描述为[4]：①单词由多个字符沿着一条想象中的水平轴线（基线）自右至左书写而成；②其中沿着基线书写的笔画部分称为主要笔画，剩余的点、元音符号等笔画称为延迟笔画；③一个或多个字符相连书写形成连体段；④通常各字符既不等高也不等宽。

阿拉伯文字是典型的草体文字，即不论印刷体

① 图中单词样本来自 IFN/ENIT V2.0 库[6-7]

收稿日期：2020-01-12 **定稿日期**：2020-03-23

基金项目：国家自然科学基金(61562058,61561031)

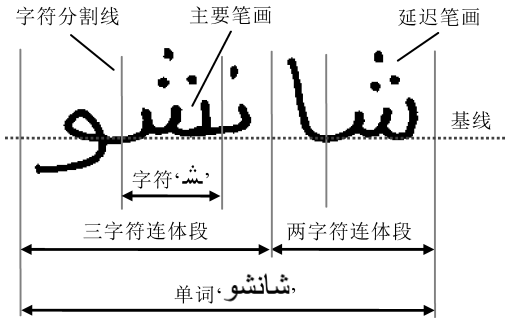

图1 手写阿拉伯单词结构规则示例

还是手写体字母都是相连书写[4]。手写草体文字识别是文字识别最困难的领域,草体文字的识别技术根据是否进行字符切分可以分为整词识别、切分识别和深度学习的策略[8-9]。基于整词的识别算法[10-12],将单词作为一个整体进行训练和识别,算法相对简单,但其对相似字微小差异的辨别能力较差,而且由于训练类别数随着单词数目增长,因而算法不易向大词汇识别扩展[12]。基于切分的识别算法[13-16],先把单词分割成字符,再识别字符,组成单词,现有字符切分算法主要基于图像分析的方法[15-16]。但是阿拉伯文字中有一些多段型字符,即字符在形态上近似于其他多个字符的组合,如س=ں+ـ+ـ等[8]。这种情况下,基于图像分析的分割方法容易产生较多的过分割错误,这使得字符分割的准确率不高,进而影响了单词识别的结果。

鉴于上述讨论,本文针对手写阿拉伯文字,提出在组件(即字符或字符的一部分)层面上分解和识别单词。算法首先建立阿拉伯单词组件库,过分割单词图像形成组件序列,再结合形态特征和位置信息识别各组件并估计其权重。然后构建组件特征至单词的加权贝叶斯推理模型,加权融合组件识别置信度和构词先验信息,得到最终的单词识别结果。

1 阿文单词的组件分析

阿文单词组件是根据29个阿拉伯字母的100个变体字符和12个元音符号的形态和结构[5]来定义的,指在阿文单词中相对独立且可被共享的笔画区域块。根据组件特征,可将阿文组件分为3类:①主体组件(main grapheme, MG):从字符的主要笔画中分割出的沿着基线书写的区域块,鉴于多段型组件易被过分割为其他组件,这里去掉多段型组件,并将这类组件分解后以不同于现有组件的新区域添加到主体组件;②点组件(dot grapheme, DG):延迟笔画中点笔画的组合;③附加组件(affix grapheme, AG):延迟笔画中除DG之外的区域块,其中12个元音符号中有些是两个AG的组合。阿拉伯文字组件库如表1所示,共包含50个MG、5个DG和9个AG,其中DG和AG的虚线表示其位置在基线的上方或下方。

表1 阿拉伯文字组件库

组件类型	组件									
MG	ں	ں	ـ	ـ	ح	ح	ـح	ـح	ع	ع
	ـع	ـع	ل	ل	ـل	ـل	م	م	ـم	ـم
	ه	ه	ـه	ـه	ف	ف	ـف	ـف	ل	ل
	ك	ك	ر	ر	ط	ط	ص	ص	د	د
	و	و	ا	ا	ى	ى	لا	لا	د	د
DG	◌ٔ	◌ٕ	◌ٖ	◌ٗ	◌ٜ					
AG	◌ٮ	◌ٔ	◌ٕ	◌ٙ	◌ٚ	◌ٛ	◌ٜ	◌ٝ	◌ٓ	

根据阿拉伯文字组件库,对阿文单词以组件为单元来拆分,获得阿拉伯单词的组件序列构成。以单词شانشو为例,其所包含的字符序列(自右至左)为:ش、ن、ا、ش、و,那么该单词的组件构成可表述为:MG序列:ـش、ـن、ـا、ـش、و;DG序列:◌ٜ、◌ٔ、◌ٜ,如图2所示。

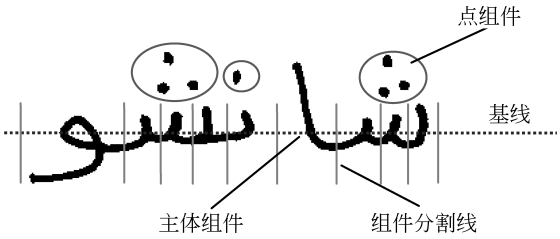

图2 手写阿文单词的组件构成示例

相比字符,在组件层面分解单词,不仅有效解决了多段型字符在分割时易产生过分割错误的问题,而且通过组件分析使相似字之间的微小差异被放大,从而易于检出和辨别。

2 阿文单词的贝叶斯推理模型

贝叶斯推理模型是一种以概率分析和图论为基础的数据模型,能有效地综合数据的先验信息和样本信息,近年来在模式识别领域上的应用逐渐被关注[17-18]。

本文依据对阿文单词的组件分析,构建自组件特征至单词类别的贝叶斯推理模型,该模型以单词的组件特征为起始状态节点,以组件为中间节点、以单词类别为终止节点,形成一个关系网络图,各状态

节点之间的有向弧表示节点状态发生的关系和概率，由父节点指向子节点，如图 3 所示。

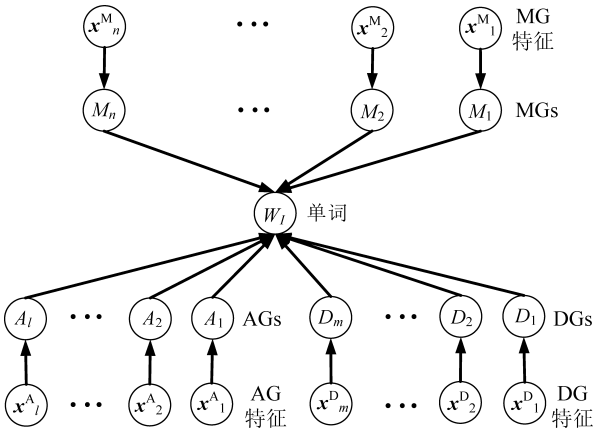

图 3　阿文单词的贝叶斯推理模型

该推理模型的具体解释如下：

(1) **状态节点**　图 3 中圆形框表示单词贝叶斯模型的状态节点，包括单词、MG、DG、AG 三类组件和组件特征，单词是组件的有序组合。如图 3 示例，单词 W_I 的父节点是所有组件序列 M_1，M_2，…，M_n；D_1，D_2，…，D_m；A_1，A_2，…，A_l。每个组件节点至少对应一个父节点，为其组件特征，即 $\boldsymbol{x}_1^{\mathrm{M}}$，…，$\boldsymbol{x}_n^{\mathrm{M}}$，$\boldsymbol{x}_1^{\mathrm{D}}$，…，$\boldsymbol{x}_m^{\mathrm{D}}$，$\boldsymbol{x}_1^{\mathrm{A}}$，…，$\boldsymbol{x}_l^{\mathrm{A}}$。

(2) **基于单词类别的模型规整**　在本文单词识别算法中，需要利用单词贝叶斯推理模型分别计算该单词样本至 946 个单词类别的后验概率。由于单词所包含的各类组件数目不定，为计算待测样本至单词类别的识别概率，设定一个空组件 Φ，代表该处没有组件，利用空组件 Φ 规整样本特征和单词类别的模型结构。规整方法是：以单词类别的节点数为标准，调整单词样本的节点，如果样本的组件节点个数较大，则去掉后面多余的组件；反之，则以空组件 Φ 补全。包含空组件后，组件数目变更为 51 个 MG、6 个 DG 和 10 个 AG。

(3) **状态转移概率**　图 3 中状态节点间的实线箭头表示状态转移概率，方向是父节点指向子节点。状态转移概率可分为两类：①表示单词和组件构成关系的转移概率，记作 $p(W_I|M_i)$，$i=1$，…，N_{M}(MG 至单词)；$p(W_I|D_j)$，$j=1$，…，N_{D}(DG 至单词)；$p(W_I|A_k)$，$k=1$，…，N_{A}(AG 至单词)，$I=1$，…，N_{W}。②表示组件特征至组件节点的转移概率，即组件识别置信度，记作 $p(M_i|\boldsymbol{x}_i^{\mathrm{M}})$，$i=1$，…，$N_{\mathrm{M}}$；$p(D_j|\boldsymbol{x}_j^{\mathrm{D}})$，$j=1$，…，$N_{\mathrm{D}}$；$p(A_k|\boldsymbol{x}_k^{\mathrm{A}})$，$k=1$，…，$N_{\mathrm{A}}$。其中，$N_{\mathrm{W}}=946$，$N_{\mathrm{M}}=51$，$N_{\mathrm{D}}=6$，$N_{\mathrm{A}}=10$。

3　单词识别算法整体流程

本文基于加权贝叶斯的阿文单词识别算法，首先将阿文单词分割为组件序列，再进行组件识别和组件加权系数估计，最后通过加权贝叶斯推理，计算单词后验概率并获得单词识别结果。具体算法描述如下。

3.1　组件分割

本文采用我们在以前工作[19]中所提出的主体分割和附加聚类算法(main segmentation and additional clustering，MSAC)，对脱机阿文单词进行组件分割，组件分割的流程如图 4 所示，主要包括以下几个步骤。

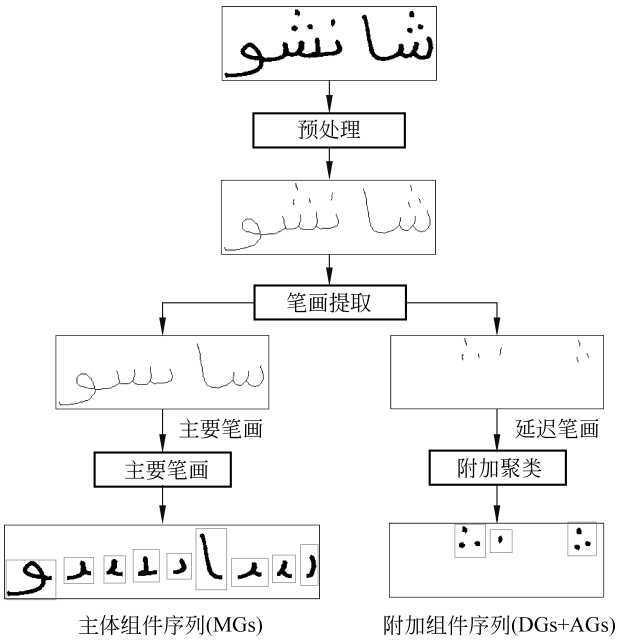

图 4　MSAC 组件分割流程

(1) **预处理**　对脱机阿文单词进行预处理，包括二值化、归一化、断笔修复、倾斜校正和细化，其中二值化是灰度 255 置 1，其余置 0；归一化为宽度 512，高度等比例缩放；断笔连接是两笔画间距小于笔画宽度的 3/2 时进行修复；倾斜校正范围是 ±30°；细化采用 Z-S+Holt 算法[19]。

(2) **笔画提取**　首先通过连通域检测提取单词笔画，根据点阈值判定点笔画，再对除去点笔画后的剩余笔画进行 Hough 变换，并根据其峰值点找到基线位置，获取基线域，然后将与基线域连通的笔画确定为主要笔画，其他笔画为延迟笔画。

(3) **主体分割和MG序列获取** 在基线域内计算主要笔画的垂直差分投影[20],取其极小值点为MG切分点,自MG切分点,垂直分割主要笔画得到主体组件,按位置自右至左记作 $\boldsymbol{M}=(M_1, M_2, \cdots, M_n)$。

(4) **附加笔画聚类** 手写点笔画存在连写情况,通常有3种点连写形式,分别是:"ـــ"("ـ ـ"连写)、"ـٰـ"、"ـٔـ"("ـ ـ"连写)。首先,从除去点之外的延迟笔画中判断出点连写笔画,其中,基线上的点连笔通过识别"ـٰـ""ـٔـ""ـــ"和附加组件"ـٗ""ـٕ""ـٖ""ـٓ""ـٙ""ـٚ""ـٛ""ـٜ""ـٝ"来获得,基线下的点连笔通过识别"ـــ"和附加组件"ـٕ""ـٖ"来获得;然后,根据顺序聚类的最大最小算法[21],对点笔画和点连写笔画进行聚类。

(5) **DG序列和AG序列的获取** 聚类后的点群作为点组件,按位置自右至左,记作 $\boldsymbol{D}=(D_1, D_2, \cdots, D_m)$,除去MG和DG以外的单个笔画构成附加组件,按位置自右至左记作 $\boldsymbol{A}=(A_1, A_2, \cdots, A_l)$。

3.2 组件子分类和识别

本文结合组件分割时获得的位置信息,提出一种新的多级混合式组件识别算法,首先根据位置信息和组件类型将所有组件预分类为8个子类,然后针对MG、AG、DG三类组件,根据各自的结构特点,设计不同的特征提取和分类器,在各自所属的子类范围内再进一步分类。组件子类的描述如表2所示。

表2 阿拉伯文组件的8个子类

组件子类	组件									
MG-S	ں و	ح ا	ع ى	ل لا	م د	ه	ف	ل	ر	د
MG-FC	ں و	ح ا	ع ى	ل لا	م د	ه	ف	ل	ر	د
MG-M	ـں	ـحـ	ـعـ	ـلـ	ـمـ	ـهـ	ـفـ	ـكـ	ـط	ـص
MG-BC	ـں	ـحـ	ـعـ	ـلـ	ـمـ	ـهـ	ـفـ	ـكـ	ـط	ـص
DG-U	٭	٭٭	∴							
DG-D	٭	٭٭								
AG-U	ـٗ	ـٕ	ـٖ	ـٓ	ـٙ	ـٚ	ـٛ			
AG-D	ـٕ	ـٖ								

表2中,MG根据其在连体段中所处的位置可分为独立(S, stand alone)、前连(FC, front-connection)、双连(m, middle)、后连(BC, behind-connection)[5]等4个子类,DG和AG各自根据其位于基线的上方或下方可分为上方(up-diacritics)和下方(down-diacritics)两个子类,于是共有MG-S、MG-FC、MG-M、MG-BC、DG-U、DG-D、AG-U、AG-D等8个组件子类。

对于DG,由于点的数目特征确切直观,于是直接根据点数目 n_d($n_d=1, 2, 3$)计算点组件的识别距离,并对点数目的差值加1以避免距离为0情况。设组件特征向量为 $\boldsymbol{x}$,那么点组件识别距离的计算如式(1)所示。

$$d_i(\boldsymbol{x})=[|\ n_d-(n_d)_i\ |+1],\quad i=1,\cdots,N_S \tag{1}$$

其中,$d_i(\boldsymbol{x})$代表组件 $\boldsymbol{x}$ 对第 i 类候选的识别距离,N_S是组件子类别数,对于DG-U,$N_S=3$;对于DG-D,$N_S=2$。

对于MG和AG,采用轮廓Freeman上、下、左、右4方向链码结合弹性网格特征提取(elastic mesh directional features, EMDF)[22]。考虑到MG和AG的面积比例,网格大小对MG取8×8,对AG取4×4。采用MQDF分类器[23]计算MG和AG的识别距离,如式(2)所示。

$$\begin{aligned} d_i(\boldsymbol{x})=\frac{1}{\sigma}\Bigg\{ & \|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2 \\ & -\sum_{k=1}^{r}\left(1-\frac{\sigma}{\lambda_{i,k}}\right)[\boldsymbol{\varphi}_{i,k}^T(\boldsymbol{x}-\boldsymbol{\mu}_i)]^2\Bigg\} \\ & +\sum_{k=1}^{r}\log\lambda_{i,k}+(n-r)\log\sigma, \\ & i=1,\cdots,N_S \end{aligned} \tag{2}$$

其中,$\boldsymbol{\mu}_i$ 是均值向量,$\lambda_{i,k}$代表协方差矩阵的第 k 个特征值,$\boldsymbol{\varphi}_{i,k}$是其对应的特征向量,$r$ 是主轴个数,$r<n$,常量 σ 是补偿因子,上述参数可在训练中利用最大似然估计得到。另外,i 代表第 i 类候选,N_S是组件子类别数,对于MG-S和MG-FC,$N_S=15$;对于MG-M和MG-BC,$N_S=10$;对于AG-U,$N_S=7$;对于AG-D,$N_S=2$。

另外,对3.2节中所描述的空组件 Φ,规定空组件的特征为全0向量。

3.3 置信度转换和权重估计

3.3.1 识别置信度转换

根据组件分类器输出的识别距离 $d_j(\boldsymbol{x})$,$j=1, \cdots, N_S$,采用softmax函数对识别距离进行置信度转换,得到组件识别置信度,计算如式(3)所示。

$$p(q_j\mid\boldsymbol{x})=\frac{\exp[-d_j(\boldsymbol{x})]}{\sum_{k=1}^{N_S}\exp[-d_k(\boldsymbol{x})]},\quad j=1,\cdots,N_S \tag{3}$$

其中,$p(q_j\mid\boldsymbol{x})$代表组件 $\boldsymbol{x}$ 至第 j 类候选的识

别置信度，N_S是组件子类别数。

若上述估计出的识别置信度在不同子类范围，则需要将其扩张到统一的 MG、DG 或 AG 组件空间，扩张方法如式(4)所示。

$$p(\omega_i \mid \boldsymbol{x}) = \begin{cases} p(q_j \mid \boldsymbol{x}), & \omega_i = q_j \\ 0, & \omega_i \neq q_j \end{cases} \tag{4}$$

其中，$p(\omega_i \mid \boldsymbol{x})$为扩张后的组件识别置信度，$i = 1, \cdots, N$，对于 MG，$N = N_M = 51$；对于 DG，$N = N_D = 6$；对于 AG，$N = N_A = 10$。

3.3.2　组件权重估计

实验发现，单词中各组件的识别可靠度不同。当某组件的候选类别里具有相似字符时，相似字符对应的识别置信度往往较为相近，这时该组件的识别结果不太可靠；反之，当某一候选识别置信度相较其他候选显著地高，则说明该组件前几候选中没有相似字符的情况，识别结果较为可靠。本文算法试图在单词识别中对结果较可靠的组件给予较大的权重，以提高最终的单词识别率，考虑以组件识别置信度的熵值的分布来表述可靠性，组件权重估计的计算如式(5)所示。

$$\lambda_k = \frac{\dfrac{1}{H(\boldsymbol{x}^k)}}{\displaystyle\sum_{j=1}^{N_G} \frac{1}{H(\boldsymbol{x}^j)}}, \quad k = 1, \cdots, N_G,$$

$$H(\boldsymbol{x}^k) = -\sum_{i=1}^{N} p(\omega_i \mid \boldsymbol{x}^k) \log [p(\omega_i \mid \boldsymbol{x}^k)] \tag{5}$$

其中，λ_k是第 k 个组件的权重系数，$p(\omega_i \mid \boldsymbol{x}^k)$是第 k 个组件的第 i 类候选识别置信度，N 是组件类别数，N_G 是该组件所在单词中的组件总数。

3.4　单词识别

单词识别的原理是计算待测样本至单词类别的识别置信度(即后验概率)，然后按照置信度自大至小的顺序输出候选单词类别，整个识别过程包括训练阶段和识别阶段。

3.4.1　训练阶段

在训练阶段完成对单词贝叶斯模型推理中状态转移概率的获取。其中：

(1) 对于表示单词和组件构成关系的转移概率 $p(W_I \mid M_i)$，$I = 1, \cdots, N_W$，$i = 1, \cdots, N_M$；$p(W_I \mid D_j)$，$j = 1, \cdots, N_D$，$p(W_I \mid A_k)$，$k = 1, \cdots, N_A$。采用最大似然估计进行参数学习，统计数据来自阿拉伯文语料库，先由阿文字母使用频率[24]转化得到各组件的先验概率，再对各单词类别的组件构成统计得到单词内各组件和单词的联合概率，然后用条件概率公式计算得到组件至单词的条件概率，即转移概率。

(2) 对于表示组件特征至组件节点的转移概率，即组件识别置信度，$p(M_i \mid \boldsymbol{x}_i^M)$，$i = 1, \cdots, N_M$；$p(D_j \mid \boldsymbol{x}_j^D)$，$j = 1, \cdots, N_D$，$p(A_k \mid \boldsymbol{x}_k^A)$，$k = 1, \cdots, N_A$。依照前述识别算法，通过组件分类器输出识别距离，再进行置信度识别，通过估计来获得转移概率。

3.4.2　识别阶段

计算待测样本至阿文单词类别的识别置信度 $p(W_I \mid \boldsymbol{x})$，其中 $\boldsymbol{x} = \{ \boldsymbol{x}_1^M, \cdots, \boldsymbol{x}_n^M, \boldsymbol{x}_1^D, \cdots, \boldsymbol{x}_m^D, \boldsymbol{x}_1^A, \cdots, \boldsymbol{x}_l^A, \}$是待测样本的组件特征，根据概率乘法公式以及贝叶斯推理的条件独立性，如式(6)所示。

$$p(W_I \mid \boldsymbol{x}) = \prod_{i=1}^{N_G} p(V_i \mid \mathrm{pa}(V_i), S^h) \tag{6}$$

其中，$V_i (i = 1, \cdots, N_G)$表示贝叶斯推理模型中与单词 W_I 相关联的状态节点，有 $N_G = n + m + 1$，$\mathrm{pa}(\cdot)$表示节点 V_i 的父节点集，S^h表示该父节点集的路径分布。

结合图 3 中所述的模型结构，如式(7)所示。

$$\begin{aligned} p(W_I \mid \boldsymbol{x}) &= \prod_{i=1}^{n} p(W_I \mid M_i) p(M_i \mid \mathrm{Pa}(M_i)) \times \\ &\quad \prod_{j=1}^{m} p(W_I \mid D_j) p(D_j \mid \mathrm{Pa}(D_j)) \times \\ &\quad \prod_{k=1}^{l} p(W_I \mid A_k) p(A_k \mid \mathrm{Pa}(A_k)) \\ &= \prod_{i=1}^{n} p(W_I \mid M_i) p(M_i \mid \boldsymbol{x}_n^M) \times \\ &\quad \prod_{j=1}^{m} p(W_I \mid D_j) p(D_j \mid \boldsymbol{x}_j^D) \times \\ &\quad \prod_{k=1}^{l} p(W_I \mid A_k) p(A_k \mid \boldsymbol{x}_k^A) \end{aligned} \tag{7}$$

用组件权重系数 $\lambda_k (k = 1, \cdots, n + m + 1)$对式(7)进行修正，得到式(8)：

$$\begin{aligned} p(W_I \mid \boldsymbol{X}) &= \prod_{i=1}^{n} p(W_I \mid M_i) [p(M_i \mid \boldsymbol{x}_n^M)]^{\lambda_i} \times \\ &\quad \prod_{j=1}^{m} p(W_I \mid D_j) [p(D_j \mid \boldsymbol{x}_j^D)]^{\lambda_{n+j}} \times \\ &\quad \prod_{k=1}^{l} p(W_I \mid A_k) [p(A_k \mid \boldsymbol{x}_k^A)]^{\lambda_{n+m+k}} \end{aligned} \tag{8}$$

于是，组件特征为 $\boldsymbol{x}$ 的待测样本，其单词首选识别结果为最大后验概率对应的单词类别，如式(9)所示。

$$I=\text{argmax}\{p(W_I \mid \boldsymbol{x}), I\} \tag{9}$$

4 实验

算法性能在 IFN/ENIT v2.0 手写阿拉伯文字数据库[6-7]上验证，该数据库包含 946 个突尼斯城市/村庄名，共 32 492 个脱机阿文单词样本，分为编号为 a、b、c、d 和 e 的五个组[6-7]。以下各实验均使用 a～d 组数据训练，使用 e 组数据测试，算法用 VC++ 6.0 编程，运行环境是 2.6G Intel i5-4300M CPU、4.0 GB 内存的 PC 机。

4.1 组件分割性能分析

为评估分割结果，使用三个度量标准：准确率、召回率和误检率。准确率是算法所获得分割点中正确的比率；召回率指真值分割位置中能被算法正确检出的比率；误检率＝1－准确率，包括过分割和错分割两种错误，其中过分割是将一个组件分割成多个组件，而错分割则指分割边界不正确。

本实验在 IFN/ENIT v2.0 数据库[6-7]上测试三种过分割算法的性能，使用 a～d 组数据进行训练，测试数据是 e 组 6 033 个单词样本所包含的 65 884 个组件分割点。算法 1 即本文过分割 MSAC 算法。算法 2 是采用文献[15]提出的最少像素定位结合最优拓扑结构筛选的手写阿文过分割算法。算法 3 是采用文献[16]提出的基于改进垂直投影和模板匹配的启发式手写阿文过分割算法。

表 3 给出了三种过分割算法的组件分割性能比较，可以看出，本文组件分割算法(算法 1)性能良好，获得 97.78%准确率和 98.05%召回率。算法 1 针对过分割的误检率仅有 0.96%，对于错分割的误检率为 1.26%，均远低于另外两种算法。良好的组件分割性能是本文基于分割策略的单词识别算法实施的基础。

表 3 组件分割性能比较

	准确率/%	召回率/%	误检率/%			耗时/(ms/词)
			过分割	错分割	共计	
算法 1	97.78	98.05	0.96	1.26	2.22	98
算法 2	83.66	97.89	13.02	3.32	16.34	148
算法 3	92.57	95.13	4.25	3.18	7.43	86

4.2 组件识别性能分析

本实验所使用的组件样本通过对 IFN/ENIT v2.0 数据库[6-7]样本进行手动分割得到，训练数据是来自该数据库 a～d 组的共 305 042 个组件，测试数据是来自 e 组的 71 917 个组件。实验对比两种识别算法的性能。算法 1 即本文多级混合式的手写阿文组件识别算法。算法 2 是文献[25]提出的脱机阿文字符识别算法，该算法基于神经网络分类器，并结合了统计和结构特征。

表 4 列出了两种识别算法分别对 MG、DG 和 AG 的识别结果比较，可以看出，本文组件识别算法(算法 1)相较算法 2 性能较好，这是因为本文多级混合式的手写阿文组件识别算法根据组件分割时的位置信息预分类组件，又为 MG、DG 和 AG 设计不同的特征提取和分类器，能获得较好的识别效果。而且，算法 1 使用距离分类器，因而相较算法 2 神经网络分类器的耗时少。对 DG 组件，算法 1 获得的识别率较算法 2 高 2.11%，因为本文算法考虑了三种点连笔的情况，因而对书写连笔较多的样本组识别率高。

表 4 组件识别性能比较

	MG		AG		DG	
	识别率/%	耗时/(ms/组件)	识别率/%	耗时/(ms/组件)	识别率/%	耗时/(ms/组件)
算法 1	97.89	124	98.21	123	99.99	36
算法 2	97.75	732	98.17	731	97.88	730

4.3 单词识别性能分析

本实验使用 IFN/ENIT v2.0 数据库[6-7]的 a～d 组进行训练，训练数据包括单词样本 26 459 个，字符样本 212 211 个，组件样本 305 042 个，测试数据是 e 组的 6 033 个单词样本。实验对比了四种算法的性能，算法 1、2 和 3 基于切分识别，算法 1 是本文手写阿文单词识别算法；算法 2 是文献[13]提出的基于多边形近似描述结构特征和多边形模糊匹配的阿文单词识别算法；算法 3 采用文献[14]提出的结合纵、横向扫描模板和支持向量机(support vector machine，SVM)分类器的手写阿文单词识别算法；算法 4 基于整词识别，由文献[10]提出，采用滑动窗统计特征结合多流隐马尔可夫模型(hidden Markov models，HMM)分类器。表 5 总结了四种算法的单

词识别性能。

表 5　单词识别性能比较

	平均识别率/%		训练类别数	识别策略	运行时间/(毫秒/单词)
	首选	前五			
算法 1	90.03	94.17	62	切分识别	625
算法 2	79.22	85.56	100	切分识别	924
算法 3	85.54	91.32	100	切分识别	785
算法 4	89.65	93.79	946	整词识别	307

可以看出，本文算法(算法 1)性能良好，单词首选识别率为 90.03%，证实了该算法的有效性。分析来说，首先，在分割单元方面，对比算法 1 和算法 2、3 可知，本文基于组件的分解和建模可以减少过分割错误，在组件层面识别单词，能将相似词间的微小差异定位至不同组件，并且在分割时考虑到点笔画的三种连写形式，有效解决了手写文字笔画粘连的识别难点，进而有效提高了单词识别率。其次，对比识别策略可知，本文基于切分识别的算法 1 获得的识别率稍高于基于整词识别的算法 4，而识别所需的训练基元是 50 个 MG、9 个 AG 和 3 个点连笔，共 62 个组件，训练所需类别数目小且固定，算法向大规模词汇识别的可扩展性较强。最后，在耗时方面，由于分割模块会部分增加算法复杂度，切分识别策略相比整词识别策略，算法的运行时间较长。

5　结束语

脱机手写阿文单词书写粘连，笔画形态复杂，文字特征很难准确提取。本文将阿文单词分解为组件，并设计多级混合式分类器来识别组件，再通过单词加权贝叶斯模型的构建和推理来获取单词识别结果。算法不但能检测和辨识到相似单词间的微小差异，而且对书写连笔、笔画漂移等手写复杂情况具鲁棒性。另外，算法训练所需组件类别有限，易于向大词汇量识别任务扩展。

算法目前的识别错误主要出现在书写潦草、点笔画连写不规整和点笔画丢失的情况。下一步研究期望通过提高组件识别率和改进单词结构模型来获得更好的单词识别性能。

参考文献

[1] Impedovo S. More than twenty years of advancements on frontiers in handwriting recognition[J]. Pattern Recognition, 2014, 47 (1): 916-928.

[2] Harmandeep Kaur, Munish Kumar. A comprehensive survey on word recognition for non-Indic and Indic scripts[J]. Pattern Analysis and Applications, 2018, 21: 897-929.

[3] 许亚美，卢朝阳，李静. 多部件自适应融合的手写体阿拉伯字符识别[J]. 西安电子科技大学学报，2012，39(6)：16-21.

[4] 谢旭东，李宁，彭良瑞 等. 与基线信息无关的手写阿拉伯文字特征提取[J]. 清华大学学报(自然科学版)，2012，52(12)：1682-1686.

[5] ISO/IEC JTC 1. ISO/IEC 8859-6-1999, Information Technology-8-Bit Single-byte Coded Graphic Character Sets-Part 6: Latin/Arabic Alphabet [S]. IX-ISO, 1999.

[6] Pechwitz M, Snoussi Maddouri S, Märgner V, et al, IFN/ENIT-database of handwritten Arabic words [C]//Proceedings of the 7th Colloque International Francophone sur l'Ecrit et le Document. Hammamet: Citeseer, 2002: 127-136.

[7] El Abed H, Märgner V. The IFN/ENIT-database-a tool to develop Arabic handwriting recognition systems [C]//Proceedings of the 9th International Symposium on Signal Processing and Its Applications. Sharjah: IEEE, 2007: 1-4.

[8] Tanzila S, Amjad R, Mohamed E B. Methods and strategies on off-line cursive touched characters segmentation: A directional review[J]. Artificial Intelligence Review, 2014, 42 (4): 1047-1066.

[9] Wu Y C, Yin F, Liu C L. Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models [J]. Pattern Recognition, 2017, 65: 251-264.

[10] Khaoula Jayech, Mohamed Ali Mahjoub, Najoua Essoukri Ben Amara. Synchronous multi stream hidden Markov model for offline Arabic handwriting recognition without explicit segmentation[J]. Neurocomputing, 2016, 214(19): 958-971.

[11] Reza Tavoli, Mohammadreza Keyvanpour, Saeed Mozaffari. Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic Persian handwritten words recognition [J]. IET Image Processing, 2018, 12 (9): 1606-1616.

[12] Zahia Tamen, Habiba Drias, Dalila Boughaci. An efficient multiple classifier system for Arabic handwritten words recognition[J]. Pattern Recognition Letters, 2017, 93(7): 123-132.

[13] Mohammad Tanvir Parvez, Sabri A Mahmoud. Arabic handwriting recognition using structural and syn-

tactic pattern attributes[J]. Pattern Recognition, 2013, 46(1): 141-154.

[14] Faouzi Zaiz, Mohamed Chaouki Babahenini, Abdelhamid Djeffal. Puzzle based system for improving Arabic handwriting recognition[J]. Engineering Applications of Artificial Intelligence, 2016, 56(11): 222-229.

[15] Elzobi M, Al Hamadi A, Al Aghbari Z, et al. IESK-ArDB: A database for handwritten Arabic and an optimized topological segmentation approach[J]. International Journal on Document Analysis and Recognition, 2013, 16(3): 295-308.

[16] Abdelhay Zoizou, Arsalane Zarghili, Ilham Chaker. A new hybrid method for Arabic multi-font text segmentation, and a reference corpus construction[J]. Journal of King Saud University-Computer and Information Sciences, 2018. doi: https://doi.org/10.1016/j.jksuci.2018.07.003.

[17] 眭萍, 郭英, 李红光 等. 半监督条件下的贝叶斯估计辐射源指纹特征识别[J]. 华中科技大学学报(自然科学版), 2018, 46(8): 71-76.

[18] Liu Li, Wang Shu, Su Guoxin, et al. Towards complex activity recognition using a Bayesian network-based probabilistic generative framework[J]. Pattern Recognition, 2017, 68 (8): 295-309.

[19] 许亚美. 手写维吾尔文字识别若干关键技术研究[D]. 西安: 西安电子科技大学博士学位论文, 2013.

[20] Al Hamad H A, Zitar R A. Development of an efficient neural-based segmentation technique for Arabic handwriting recognition[J]. Pattern Recognition, 2010, 43(8): 2773-2798.

[21] Juan A, Vidal E. Comparison of four initialization techniques for the K-medians clustering algorithm[C]//Proceedings of the Advances in Pattern Recognition, Lecture Notes in Computer Science. Berlin: Springer, 2000: 842-852.

[22] Jin Lianwen, Wei Gang. Handwritten Chinese character recognition with directional decomposition cellular features[J]. Circuits, Systems and Computers, 1998, 8(4): 517-524.

[23] Wei Xiaohua, Lu Shujing, Lu Yue. Compact MQDF classifiers using sparse coding for handwritten Chinese character recognition[J]. Pattern Recognition, 2018, 76(4): 679-690.

[24] Frequency statistics of Arabic letters[EB/OL]. https://en.m.wikipedia.org/wiki/Arabic_letter_frequency.

[25] Lamghari N, Charaf M E H, Raghay S. Hybrid feature vector for the recognition of Arabic handwritten characters using feed-forward neural network[J]. Arabian Journal for Science & Engineering, 2018, 43: 7031-7039.

许亚美(1978—),通信作者,博士,副教授,主要研究领域为模式识别、手写文字识别。
E-mail: yameixu@126.com

何继爱(1969—),硕士,副教授,主要研究领域为信号处理、信息控制。
E-mail: Hejiai@lut.cn