

线性合成的双粒度 RNN 集成系统

张亮¹ 黄曙光¹ 胡荣贵¹

摘要 针对脱机文字识别, 提出了一种基于线性合成的双粒度递归神经网络 (Recurrent neural network, RNN) 集成系统. 首先, 使用单词 RNN 对未知图像进行识别; 然后, 依据识别结果进行字符分割, 使用字符 RNN 对分割后的字符进行识别, 并利用查表法计算字符的后验概率; 最后, 综合两个 RNN 的识别结果决定最终单词输出. 在 CAPTCHA 识别和手写识别上的实验结果证明了该系统的有效性.

关键词 脱机文字识别, 递归神经网络, 集成系统, 字符分割

DOI 10.3724/SP.J.1004.2011.01402

Ensemble System of Double Granularity RNN by Linear Combination

ZHANG Liang¹ HUANG Shu-Guang¹ HU Rong-Gui¹

Abstract For offline text recognition, an ensemble system of double granularity RNN (recurrent neural network) by linear combination is proposed. Firstly, the unknown image is recognized by a word RNN. Secondly, based on the recognition result, connected characters are segmented, and then recognized by a character RNN. Characters' posterior probabilities are calculated by a table-looking method. Finally, the final output is decided by combining both RNNs' results. Experiment results based on CAPTCHA (completely automated public Turing test to tell computers and humans apart) recognition and handwritten recognition proved the effectiveness of this ensemble system.

Key words Offline text recognition, recurrent neural network (RNN), ensemble system, character segmentation

脱机文字识别是模式识别领域的一个重要研究方向. 针对粘连文字的识别, 目前常用的方法有 3 种: 显式分割法、隐式分割法和整体识别法^[1]. 显式分割法试图在字符级别上完成分割^[2], 思路清晰直观, 但是找到一种较好的分割准则往往不是那么容易, 目前存在大量的文献对此进行研究. 整体识别法基于人类是从单词图像的整体上进行识别的原理, 它不需要进行任何分割, 直接在图像上提取特征值, 然后使用某种分类器进行识别. 由于完全避免了分割环节, 整体识别法能够取得很高的识别率 (> 90%)^[3], 但是它只适合小字典情况下的文字识别, 而不适合大字典或者无字典的情况. 与显式分割法需要寻找某种分割准则不同, 隐式分割法只需要简单地将图像分为多个等宽细条, 然后在每个细条上提取特征形成特征值序列, 最后使用某种分类器对特征值序列进行识别即可, 是脱机粘连文字识别主要采取的一种方式. 目前在隐式分割法中广泛使用的一种分类器是隐马尔科夫模型 (Hidden Markov model, HMM)^[4], 但是 HMM 在理论上存在一些缺陷, 例如观测值的出现概率只取决于当前状态, 而与过去状态无关; HMM 属于“产生式”模型, 而这类模型在分类问题上通常逊色于“区分式”模型^[5]. 针对该情况, Graves 等提出了一种可用于隐式分割法的长短时记忆 (Long short-term memory, LSTM) 型递归神经网络 (Recurrent neural network, RNN) 模型^[5-6]. 在 ICDAR 2009 组织的几项识别

竞赛中, 基于 RNN 的识别系统均被测评为“Winner”^[7], 在识别率等方面较其他一些基于 HMM 的系统具有优势, 但是 RNN 的识别率仍然可以进一步提高. 通过实验发现, RNN 的 TOPN 识别率远高于 TOP1 (在本文的 CAPTCHA 识别实验中发现: TOP1, TOP3, TOP5 的识别率分别为 0.536, 0.661, 0.692). 因此, 如果能够找到一种较好的后处理方法, 从 TOPN 候选词中挑选出最可能的单词, 那么就能够有效提高识别率. 基于该原理, 本文提出了一种线性合成的双粒度 RNN 集成系统, 取得了较好的实验结果.

1 基于 RNN 的脱机文字识别系统

用于脱机文字识别的 RNN 一般为 3 层结构: 1 个输入层, 1 个隐层, 1 个输出层. 输入层节点数为输入数据的维度, 隐层包含多个 LSTM 单元, 输出层的节点数为 $M+1$, 其中 M 为待识别的文字中包含的字符类别数, “+1”节点对应空格字符 (“-”). 得到一张单词图片后, 直接使用滑动窗口在图片上提取特征值序列, 并与图片对应的字符序列 (也即单词) 构成一个训练样本, 即可使用这个样本对 RNN 进行训练. 设某个图片的特征值序列为 $O_1O_2 \cdots O_T$ (T 为时间序列长度), 它对应的字符序列为 $z = z_1z_2 \cdots z_L$ (例如“ABCD”, L 为字符序列长度), 那么训练的目标是使所有训练样本 (设为 Tra) 的字符序列在对应的特征值序列之上的概率之积最大. 为了防止溢出, 可以使用 \ln 函数将其描述为 (加“-”变为求最小):

$$O = -\ln \left(\prod_{(x,z) \in Tra} p(z|x) \right) = -\sum_{(x,z) \in Tra} \ln p(z|x) \quad (1)$$

目标函数确定以后, 即可使用随时间反向传播 (Back propagation through time, BPTT) 算法进行训练, 其中 $p(z|x)$ 的计算需要使用前后向算法^[6]. RNN 训练好以后, 即可用于识别. 对于某个未知图片 img , 首先使用滑动窗口机制提取它的特征值序列, 然后将这个特征值序列输入到 RNN 中, RNN 即输出 img 对应的 TOPN 列表: $TOPN(img) = \{\langle word_1, prob_1 \rangle, \dots, \langle word_N, prob_N \rangle\}$, 其中 $word_i$ 表示第 i 个候选词, $prob_i$ 是其对应的后验概率, 并且 $prob_1 \geq prob_2 \geq \dots \geq prob_N$.

2 线性合成的双粒度 RNN 集成系统

2.1 基本思想和识别流程

双粒度 RNN 集成系统的基本思想是: 首先使用单词 RNN 对未知单词图像进行识别, 产生前 N 个单词粒度下的候选词, 然后依据候选词进行字符分割, 并使用字符 RNN 对分割后的字符进行识别, 最后综合两个 RNN 的识别结果决定未知图像的最终类别. 集成系统的理论依据主要来自于多分类器集成^[8]、商空间原理^[9]等. 使用该系统进行脱机文字识别的主要流程为:

步骤 1. 对未知图像 img 进行预处理, 主要包括灰度化、倾斜校正和高度归一化等.

步骤 2. 使用滑动窗口进行特征提取, 窗口的宽度为 1 个像素, 高度为图像的高度 (H), 第 i 个窗口的特征值为 O_i . 窗口上的特征可使用灰度值、矩和结构特征等. 在使用灰度值的情况下, $O_i = (O_{i1}, O_{i2}, \dots, O_{iH})^T$, 其中 O_{ij} 为图像中 (i, j) 处的灰度值. 特征提取后使用均值方差法进行归一化^[6].

步骤 3. 使用训练好的 RNN₁ (单词 RNN) 对特征值

序列 $O_1O_2 \cdots O_T$ 进行识别, 产生单词粒度下的识别结果: $\{(word_1, prob_1), \cdots, (word_N, prob_N)\}$.

步骤 4.

for $i = 1: N$ //对于每个候选词依据 $word_i$ 将 img 划分为 $length(word_i)$ 个图像片段, 第 j 个片段对应的字符类别为 $word_i(j)$;

for $j = 1: length(word_i)$

使用训练好的 RNN_2 (字符 RNN) 求取 $word_i(j)$ 在第 j 个片段上的后验概率 $pc(j)$;

end

合并 $pc(j)$ ($1 \leq j \leq length(word_i)$), 构成 $word_i$ 在字符 RNN 上的后验概率 $pword_i$;

合并 $word_i$ 在单词 RNN、字符 RNN 上的概率 $prob_i$ 和 $pword_i$ 作为它的新后验概率, 并将其赋值给 $prob_i$;

end

步骤 5. 依据 $prob_i$ 对 TOPN 列表中的候选词重新排序, 并输出概率最大候选词.

从以上流程可以看出, 该算法需要解决如下问题: 1) 如何从单词图像中, 分割出 $word_i(j)$ 对应的图片; 2) 如何训练字符 RNN 并用其计算 $word_i(j)$ 在对应图像片段上的概率; 3) 如何对 $word_i$ 在两个不同粒度 RNN 上的概率进行合并, 从而对 TOPN 列表重排序. 下面分别对它们进行介绍.

2.2 基于 RNN 的字符分割算法

为了介绍基于 RNN 的字符分割算法, 首先需要了解 RNN 的输出特性.

2.2.1 RNN 的输出特性分析

RNN 的输入为特征值序列 $O_1O_2 \cdots O_T$ (T 为时间步数). $O_1O_2 \cdots O_T$ 依次输入到网络中, 在每个时刻 t , 每个输出节点上都会有输出值. 每个时刻可以选择一个输出节点作为该时刻的输出字符, 将 T 个时刻的输出字符连接起来, 则构成一条路径 π , π 的概率等于 T 个时刻的输出端输出值之积. 但是路径并不是最终输出字符, 其中一个原因是由于 LSTM 单元的记忆性, 会导致 $t-1$ 时刻输出 c 字符, 在 t 时刻也很有可能输出 c 字符, 因此对于路径需要使用函数 B 进行处理, 才能形成候选词 l , 即:

$$l = B(\pi) \quad (2)$$

B 函数的主要功能是首先合并相邻的相同字符, 然后移除空格 “-”. 例如 B 函数对路径 “AA-B-B-CC” 的处理结果为 “ABBC”. 输入特征值序列 $O_1O_2 \cdots O_T$ 后, 单词 RNN 的输出结果为 TOPN 列表 $\{(word_1, prob_1), \cdots, (word_N, prob_N)\}$, 以及各个候选词对应的概率最大路径 $\{path_1, path_2, \cdots, path_N\}$ ^[5]. 由于使用宽度为 1 的滑动窗口进行特征提取, 因此单词图像中 x 坐标值与观测序列中的 t 值一一对应. 可以将输出路径绘制到单词图像中, 图 1 即是 RNN 的几条输出路径示例. 虚线表示 RNN 在此时刻有输出 (但图中没有指出输出字符是什么), 没有虚线的时刻表示 RNN 输出空格字符 “-”. 由于在各个时刻的相邻时刻是否存在输出并不一致, 因此虚线的粗细不同. 图 1(a) 中 “7” 上面只有一条细虚线, 表示在该时刻输出了字符 “7”, 并且相邻时刻没有输出字符. 图 1(a) 中 “m” 上有一条粗虚线, 这是因为 3 个相邻时刻都有输出字符造成的. 在这 3 个时刻中, 前两个时刻输出 “m”, 最后时刻输出的是 “3”. 同样在图 1(b) 中 “w” 上也存在两个相邻时刻都输出 “w” 的现象.

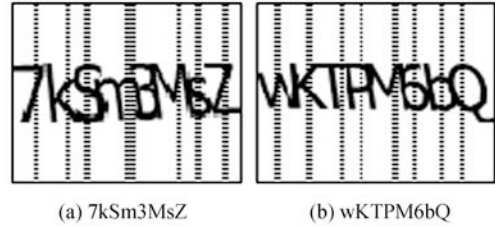


图 1 RNN 输出示例

Fig. 1 Examples of RNN's output

2.2.2 使用 RNN 进行字符分割

文献 [6] 指出: “在每个候选词对应的输出路径中, RNN 输出某个字符的时刻基本上位于该字符的图像上”, 例如图 1(a) 中, 输出 “7”、“m” 的时刻分别位于 “7”、“m” 的图像上. 依据该现象, 可以证明命题 1 成立 (证明参见附录).

命题 1. 假如 img 的真实类别是 $word_i$, img 中字符重叠程度较小, 并且在 $word_i$ 对应的输出路径 $path_i$ 中, 设输出第 j 个字符 $word_i(j)$ 的时刻为 $sp(j)$ (如果有多个相邻时刻输出 $word_i(j)$, 则 $sp(j)$ 为中间位置), 输出 $word_i(j-1)$ 的时刻为 $sp(j-1)$, 输出 $word_i(j+1)$ 的时刻为 $sp(j+1)$, 那么在图像 img 中, 从第 $sp(j-1)$ 列到第 $sp(j+1)$ 列的图像 $img(:, sp(j-1) : sp(j+1))$ 包含字符 $word_i(j)$ 图像的大部分.

基于命题 1, 可以使用如下算法依据 $path_i$ 将粘连单词图像 img 分割为单字符:

步骤 1. 对 $path_i$ 中相邻的输出字符进行预处理. 如果 $path_i(j) = path_i(j+1) \neq \text{"-"} (1 \leq j \leq T)$, 也即它们输出同一非空格字符, 则将 $path_i$ 中 $j+1$ 时刻的输出字符置为空格; 如果 $j-1, j, j+1$ 都输出同一非空格字符, 则将 $j-1$ 和 $j+1$ 时刻的输出置为空格. 4 个以上按照 3 个进行处理 (不常见). 例如路径 “AA-BBB-C” 经过预处理以后得到 “A-B-C”.

步骤 2. 找出 $path_i$ 中所有的非空格位置, 即:

$sp = find(path_i \neq \text{"-"});$ // sp 中存储所有非空格字符的出现时刻

步骤 3. 将单词图像的宽度 (也即 T) 加入到 sp 数组中最后一个元素之后:

$sp = [sp, T];$

步骤 4. 依据 sp 分割字符:

$d_0 = 1; d_1 = 1; id = 1;$ // d_0, d_1 分别是图像片段的开始位置和结束位置

for $j = 1: length(word_i)$

$d_1 = sp(j+1);$ //将 d_0 到 d_1 之间的图像片段作为字符 $word_i(j)$ 的图像

$tempImg = img(:, d_0 : d_1);$

$tempName = id + word_i(j);$

$imwrite(tempImg, tempName);$ //保存图像

$d_0 = sp(j);$ //移动图像片段的开始位置 d_0

$id = id + 1;$

end.

该算法的执行示例如图 2 所示. 从图中可以看出: 通过本文算法分割形成的字符图像基本上能够完整地包含 $word_i$ 中相应的字符, 但是存在较大的噪声干扰. 为了避免分割某个字符的同时破坏与之粘连的字符, 算法中还使

用了复用技术, 主要体现在: 图像片段 $img(:, sp(j-1) : sp(j+1))$, $img(:, sp(j) : sp(j+2))$ 分别被用作字符 $word_i(j)$, $word_i(j+1)$ 的图像, 由于 $sp(j-1) < sp(j) < sp(j+1)$, 因此 $img(:, sp(j) : sp(j+1))$ 被字符 $word_i(j)$, $word_i(j+1)$ 所复用. 复用技术保证了字符图像的完整性, 但它也是噪声的主要来源.

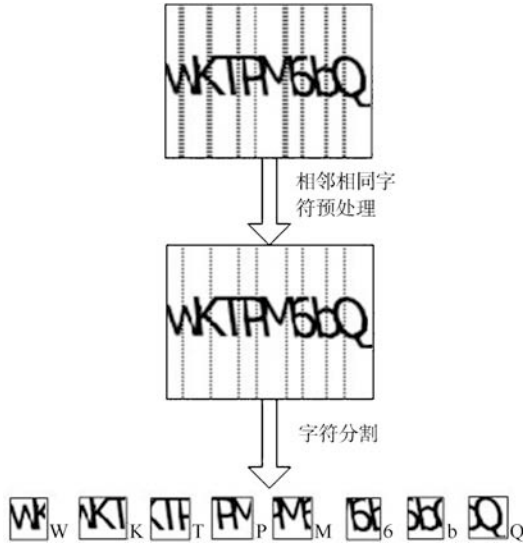


图 2 基于 RNN 的字符分割算法执行示例

Fig. 2 Processing steps of character segmentation algorithm based on RNN

2.3 字符 RNN 的训练和识别

本文使用 RNN 对字符进行识别, 设其为 RNN_2 , 将其称为字符 RNN. 使用本文提出的字符分割算法对 RNN_1 的训练集和验证集中 RNN_1 识别正确的样本进行分割, 将分割后图片分别作为 RNN_2 的训练集和验证集, 这样避免了手动对图像进行分割, 并且能够拟合到 RNN_1 识别正确的特点. RNN_2 与 RNN_1 的特征提取策略略完全相同, 因此无论在训练或测试阶段, RNN_2 都可以直接使用 RNN_1 的特征, 从而避免了重新提取特征的问题, 降低了处理时间. 使用 RNN_2 对单字符图片进行训练和识别的细节与使用 RNN_1 对脱机单词图片进行训练和识别的细节完全相同, 通过 RNN_2 计算字符 c 在图片上的后验概率可以使用文献 [6] 提出的前后向算法, 限于篇幅本文不再详述. 但是可以使用查表法加快概率的计算速度, 下文将对此进行描述.

2.4 查表法计算字符概率

分离出图像片段 $img(:, d_0 : d_1)$ 之后, 计算字符 c 在该片段上的概率可以采用直接法, 即直接将这个片段交给 RNN_2 进行识别, 然后使用前后向算法计算 c 的后验概率. 但是通过分析 TOPN 列表可以发现, 各个候选词的大部分字符是相同的, 它们只在某些字符上存在差异. 例如 “ABCDE”, “ACCDE”, “ABCD”, 因此可以使用查表法加快字符概率的计算速度. 设哈希表 $probMap$ 用于存储某个字符在某个片段上的概率, 并设目前需要求解 c 在图像片段 $img(:, d_0 : d_1)$ 上的概率, 则查表法的计算流程为:

- 步骤 1. $key = num2str(d_0) + \text{"_"} + num2str(d_1) + \text{"_"} + c$; //构造关键字
- 步骤 2.

```

if probMap.Contain(key) //说明以前已经计算过
    probability = probMap[key]; //取出以前的值
else //使用 RNN2 计算 c 的概率
    probability = RNN2(img(:, d0 : d1), c);
    将键值对 (key, probability) 加入到 probMap 中;
end

```

步骤 3. return probability.

通过该方法, 可以避免概率的重复计算, 有助于提高系统的识别速度.

2.5 候选词在字符 RNN 上的后验概率

由于 $word_i = word_i(1)word_i(2) \cdots word_i(L)$ ($L = \text{length}(word_i)$), 因此求出 $word_i(j)$ 在第 j 个图像片段上的概率 $pc(j)$ 以后, 按照均值准则^[10-11], $word_i$ 在字符 RNN 上的后验概率 $pword_i$ 为

$$pword_i = \frac{1}{L} \sum_{j=1}^L pc(j) \quad (3)$$

2.6 双粒度 RNN 的结果合成

分别使用单词 RNN 和字符 RNN 求出候选词 $word_i$ 的两个概率 $prob_i$ 和 $pword_i$ 以后, 使用如下加权线性平均法重新计算 $word_i$ 的后验概率:

$$prob_i \leftarrow (1 - \alpha) \cdot prob_i + \alpha \cdot pword_i \quad (0 \leq \alpha \leq 1) \quad (4)$$

其中 α 值可以通过双粒度 RNN 系统在验证集上的最佳性能进行确定.

3 实验和结果分析

本文实验的主要目的并不是要在相应的数据集上取得最高的识别率, 而是证明通过使用字符 RNN 对单词 RNN 的识别结果进行处理后, 能够有效地提高识别率, 也即双粒度 RNN 系统的有效性.

3.1 实验数据和实验设置

本文分别进行了全自动的区分计算机和人的图灵测试 (Completely automated public Turing test to tell computers and humans apart, CAPTCHA)^[12] 识别和脱机手写识别两个实验. 在这两个实验中, 单词 RNN 和字符 RNN 均使用灰度值特征, 并且均为 3 层结构, 隐层节点数均为 100. 使用 $\mu = 0, \sigma = 0.1$ 的高斯分布对单词 RNN 和字符 RNN 的权值进行初始化, 使用 BPTT^[6] 算法对它们进行训练, 学习速率为 $1E-4$, 冲量值为 0.9. 训练时对输入数据加入 $\mu = 0, \sigma = 1$ 的高斯噪声^[5]. 每 10 epoch 检测一次 RNN 在验证集上的识别率, 10 次检测均没有出现更优值则停止训练.

程序使用 C++ 语言编写, 由 VS2003 进行编译. 测试平台是 Windows XP SP3, 内存 1 G, CPU P4 2.6 G.

3.2 CAPTCHA 识别实验

CAPTCHA 是一种在互联网上广泛应用的安全机制. 目前安全性较高的 CAPTCHA 普遍采用任意字母组合的粘连单词形式. 我们使用一个 CAPTCHA 生成程序生成了 3000 张 CAPTCHA 图片, 将其分成 3 个不相交的数据集: 训练集、验证集、测试集的大小分别为 1000, 500, 1500. 图 3 是一些 CAPTCHA 样本示例. 特征提取时, 将所有的图像归一化成训练集上的平均高度 (27 个像素).

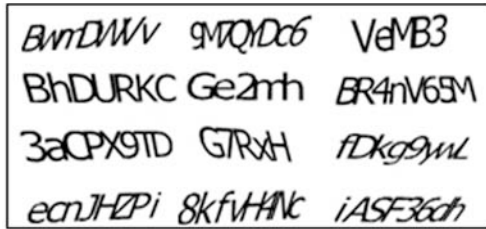


图3 CAPTCHA 样本示例

Fig.3 Samples of CAPTCHA used in experiments

通过使用基于 RNN 的分割算法, 本文在单词 RNN 的训练集上获得了 6863 个单字符训练样本, 在其验证集获得了 1825 个单字符验证样本 (样本中不存在 0, o, O, 1, I 等字符, 因为自然人在没有语言信息的情况下很难区分它们, 它们在字母与数字混合的 CAPTCHA 中使用较少). 通过字符分割实验发现: 输出每个字符的时刻基本上位于该字符的图像之上, 即文献 [6] 指出现象是成立的. 我们同时使用 SVM 和字符 RNN 对分割后的字符进行了识别实验. 实验结果表明: 字符 RNN 在测试集上取得了 90.7% 的识别率, 而 SVM^[13] (使用网格特征, 并进行 PCA + LDA^[14] 降维) 仅取得 65.1% 的识别率, 说明使用 RNN 进行字符识别是有效的. 依据验证集上的最佳性能, 确定式 (4) 中 α 为 0.85, 并且只考虑前 7 个候选词. 表 1 是双粒度 RNN 系统与 RNN 系统在测试集上的识别率对比. 从表中可以看出: 双粒度 RNN 系统的识别率明显高于 RNN, 具有一定的优势.

表 1 双粒度 RNN 系统和 RNN 系统在 CAPTCHA 识别上的对比
Table 1 Comparison between the double granularity RNN system and RNN system on CAPTCHA recognition

识别系统	识别率 (%)
双粒度 RNN	60.5
RNN	53.6

图 4 是测试集上不同的字符概率计算方法所需的平均识别时间与考虑的候选词数目 (N) 之间的关系图, 其中识别时间 = 单词 RNN 识别时间 + 字符分割时间 + 字符 RNN 识别时间 + 双 RNN 结果集成时间.

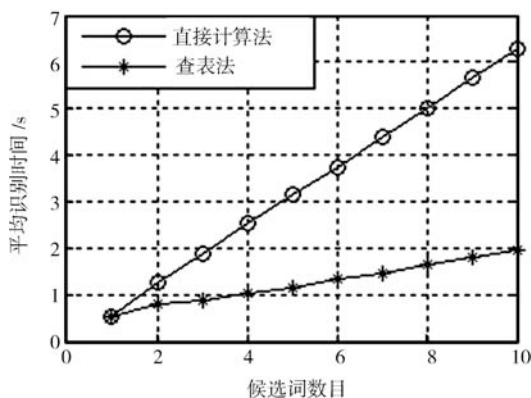


图 4 平均识别时间和候选词数目的关系

Fig.4 Relationship between the average recognition time and the number of candidates

从图 4 中可以看出直接法需要的识别时间随 N 的增大而显著变长, 当 $N = 10$ 时, 识别时间已经超过 6s. 而查表法所需的识别时间随 N 的变化相对平缓一些, 当 $N = 10$ 时, 识别时间仅为 2s, 而两者的概率计算结果并无差异. 因此在实际使用中, 应该使用查表法计算字符概率.

3.3 手写识别实验

该实验在 Cambridge 的 LOB 数据集¹上进行. LOB 数据集中包含 4617 个脱机手写单词, 将其划分为不相交的 3 个数据集: 训练集大小为 1846、验证集为 925、测试集为 1846. 特征提取时, 将所有图像的高度归一化成 30 个像素. 依据验证集上的最佳性能, 确定式 (4) 中 α 为 0.80, 并且只考虑前 7 个候选词. 表 2 是在测试集上, 双粒度 RNN 与 RNN 的识别率对比. 从该表中看出: 在手写识别上, 双粒度 RNN 同样也能取得较 RNN 更高的识别率.

表 2 双粒度 RNN 系统和 RNN 系统在脱机手写识别上的对比
Table 2 Comparison between the double granularity RNN system and RNN system on handwritten recognition

识别系统	识别率 (%)
双粒度 RNN	65.1
RNN	61.9

4 结论

针对脱机文字识别问题, 在 RNN 这种较好的分类器的基础上, 本文提出了一种基于线性合成的双粒度 RNN 集成系统. 在该系统中, 使用查表法计算字符的后验概率, 有效地避免了概率的重复计算, 缩短了识别时间. 相对于 RNN, 该系统能够取得更高的识别率. 本文还提出了一种基于 RNN 进行粘连字符分割的算法, 该算法充分利用了 RNN 的输出特性, 在分离出单字符的同时能够避免破坏与之粘连的相邻字符, 保证了字符图像的完整性. 并且该分割算法不需要刻意寻找分割准则, 适应于各种粘连字符, 具有广阔的应用前景. 在后续工作中, 我们将对该分割算法做进一步的改进和实验.

附录. 对命题 1 的证明

首先给出字符重叠程度的概念, 设 A, B 依次为图像中两个相邻的字符, 设 $startPos(C)$ 用于计算字符 C 的开始纵坐标, $endPos(C)$ 用于计算字符 C 的结束纵坐标, AB 之间的字符重叠程度 d 的计算公式为

$$d = \begin{cases} endPos(A) - startPos(B), & \text{若 } endPos(A) - startPos(B) > 0 \\ 0, & \text{否则} \end{cases} \quad (5)$$

设图像中字符的最大重叠程度为 n 个像素, 下面对命题 1 进行证明.

证明. 如图 5 所示, p, q, r 分别是单词图像中输出字符 A, B, C 的时刻 (p, q, r 分别对应 $sp(j-1), sp(j), sp(j+1)$), A, B, C 分别对应 $word_i(j-1), word_i(j), word_i(j+1)$, n 为字符之间的最大重叠程度. 在最坏的情况下, p 位置 (含) 之前和 r 位置 (含) 之后还有宽度为 n 的子图像属于字符 B , 此时图像片段 $img(:, p:r)$ 包含 B 的像素至少占 B 全部像

¹可以从 ftp.eng.cam.ac.uk/pub/data 免费下载

素的比例 pst 为

$$pst = \frac{r-p+1}{r-p+1+2(n-1)} = 1 - \frac{2(n-1)}{r-p+1+2(n-1)} \quad (6)$$

其中, $r-p+1+2(n-1)$ 即为 B 的总宽度, 设其为 w , 并设 $n = tw$ ($0 \leq t \leq 1$), 此时:

$$pst = 1 - 2t + \frac{2}{w} \quad (7)$$

当 $t \leq 15\%$ 时, $pst \geq 0.7 + 2/w$. 由于 w 非负, 因此 $pst > 0.7$. 这反映出当字符的重叠程度不超过它本身宽度的 15% 时, 基于 RNN 的字符分割算法至少能够获得字符图像的 70% . 并且字符的宽度 w 越小, 分割得到的图像越完整 (pst 越大). \square

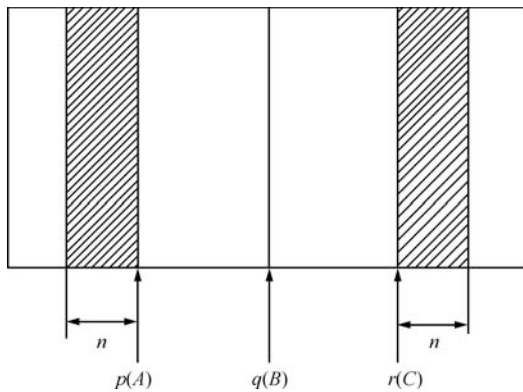


图 5 分割算法的合理性证明

Fig. 5 Proof of segmentation algorithm's reasonableness

References

- 1 Cheriet M, Kharram N, Liu C L, Suen C Y. *Character Recognition Systems: A Guide for Students and Practitioners*. Hoboken: John Wiley and Sons, 2007. 204–207
- 2 Sadri J, Suen C Y, Bui T D. A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. *Pattern Recognition*, 2007, **40**(3): 898–919
- 3 Sagheer M W, He C L, Nobile N, Suen C Y. Holistic Urdu handwritten word recognition using support vector machine. In: *Proceedings of the 20th International Conference on Pattern Recognition*. Istanbul, Turkey: IEEE, 2010. 1900–1903
- 4 Kessentini Y, Paquet T, Hamadou A B. Off-line handwritten word recognition using multi-stream hidden Markov models. *Pattern Recognition Letters*, 2010, **31**(1): 60–70
- 5 Graves A, Liwicki M, Fernandez S, Bertolami R, Bunke H, Schmidhuber J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, **31**(5): 855–868
- 6 Graves A. *Supervised Sequence Labelling with Recurrent Neural Networks* [Ph.D. dissertation], Technical University of Munich, Germany, 2008
- 7 ICDAR 2009 [Online], available: <http://www.cvc.uab.es/icdar2009/index.html>, April 28, 2011

- 8 Su Tong-Hua. *Off-line Recognition of Chinese Handwriting: From Isolated Character to Realistic Text* [Ph.D. dissertation], Harbin Institute of Technology, China, 2008 (苏统华. 脱机中文手写识别 — 从孤立汉字到真实文本 [博士学位论文], 哈尔滨工业大学, 中国, 2008)
- 9 Zhang Yan-Ping, Luo Bin, Yao Yi-Yu, Miao Duo-Qian, Wang Guo-Yin, Zhang Ling, Zhang Bo. *Quotient Space and Granular Computing: Principium and Application of Structural Problem Solving*. Beijing: Science Press, 2010. 2–5 (张燕平, 罗斌, 姚一豫, 苗夺谦, 王国胤, 张铃, 张钺. 商空间与粒计算 — 结构化问题求解理论与方法. 北京: 科学出版社, 2010. 2–5)
- 10 Kittler J, Hatef M, Duin R P W, Matas J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, **20**(3): 226–239
- 11 Koerich A L, Sabourin R, Suen C Y. Recognition and verification of unconstrained handwritten words. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, **27**(10): 1509–1522
- 12 Ahn L V, Maurer B, Mcmillen C, Abraham D, Blum M. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 2008, **321**(5895): 1465–1468
- 13 Chang C C, Lin C J. LIBSVM: a library for support vector machines [Online], available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, April 28, 2011
- 14 Cheng Zheng-Dong, Zhang Yu-Jin, Fan Xiang, Zhu Bin. Study on discriminant matrices of commonly-used Fisher discriminant functions. *Acta Automatica Sinica*, 2010, **36**(10): 1361–1370 (程正东, 章毓晋, 樊祥, 朱斌. 常用 Fisher 判别函数的判别矩阵研究. 自动化学报, 2010, **36**(10): 1361–1370)

张 亮 电子工程学院博士. 主要研究方向为模式识别, 人工智能, 信息安全技术. 本文通信作者. E-mail: mathfun@163.com (ZHANG Liang Ph.D. at Electronic Engineering Institute. His research interest covers pattern recognition, artificial intelligence, and information security. Corresponding author of this paper.)

黄曙光 电子工程学院教授. 主要研究方向为计算机应用技术, 信息安全技术. E-mail: sg Huang20020505@sina.com (HUANG Shu-Guang Professor at Electronic Engineering Institute. His research interest covers computer application and information security.)

胡荣贵 电子工程学院教授. 主要研究方向为信息安全技术. E-mail: rghu2000@126.com (HU Rong-Gui Professor at Electronic Engineering Institute. His main research interest is information security.)