

文章编号:1001-9081(2009)05-1412-04

一种基于低维特征的高精度手写数字识别算法

高宏宾¹, 陈 军¹, 陈丽平²

(1. 五邑大学 信息学院, 广东 江门 529020; 2. 齐心商业设备有限公司 国际部, 广东 深圳 518110)

(womenzaishenzhen@126.com)

摘要:提出了数字字符的轮廓骨架特征,并将这一特征与粗网格特征相结合对脱机手写体数字进行识别。获取特征向量后,利用改进的基于两级级联结构的 AdaBoost 神经网络进行逐层淘汰识别。第一级首先使用基于粗网格特征的分类器进行粗分类,淘汰大部分负样本,而使几乎所有的正样本通过。第二级由基于轮廓骨架特征的分类器对通过第一级的样本进一步淘汰识别。仿真结果表明,该办法在识别速度与识别率方面都有较大幅度的改进。

关键词:数字识别;粗网格特征;轮廓骨架特征;AdaBoost;级联结构

中图分类号:TP391.4 **文献标志码:**A

Precise recognition algorithm for handwritten digit characters based on low-dimensional features

GAO Hong-bin¹, CHEN Jun¹, CHEN Li-ping²

(1. College of Information, Wuyi University, Jiangmen Guangdong 529020, China;

2. International Department, Comix Business Machine Company Limited, Shenzhen Guangdong 518110, China)

Abstract: The contour skeleton feature of digital character was proposed. A method based on this feature and the big gridding feature for the recognition of off-line handwritten digits was also developed. The feature vectors extracted were to be recognized and eliminated gradually by making use of the improved two-stage AdaBoost neural network. First stage, the categorizer based on big gridding feature conducted general assortment to eliminate most of negative samples and let almost all the positive samples pass. Second stage, the categorizer based on contour skeleton feature conducted further sorting for the positive samples from the 1st stage. Simulation result indicates that the proposed method has improvement in recognition speed and accuracy rate.

Key words: digit recognition; big gridding feature; contour skeleton feature; AdaBoost; cascade structure

0 引言

神经网络(Artificial Neural Network, ANN)的出现给脱机手写体字符识别(Off-line Handwritten Numeral Recognition)带来了新的希望。ANN是由大量简单的基本元件——神经元相互连接而成的具有自适应能力的非线性系统。从理论上讲,如果神经网络的分类能力足够强大,就可以将字符的图像点阵直接作为神经网络的输入(即将二值化后的图像逐像素取0或1作为神经网络的输入变量),利用神经网络来抽取特征并进行正确分类。如此一来,手写体数字识别将得到简单而完美的解决。然而,目前神经网络的研究还处于初级阶段,利用图像点阵这样的高维(通常100维以上)输入变量,神经网络的规模会很大,结构也很复杂,识别正确率往往不高。因此,在现阶段而言,进一步提高手写体数字识别正确率的有效途径有:1)获取低维而有效的特征集作为神经网络的输入变量。目前存在的低维特征提取法(如13点特征提取法^[1],整体特征提取法^[2]等)虽然能够反映大多数手写体数字字符拓扑结构中的关键信息,但实践表明,这些特征容易混淆一部分字符,很难进一步提高识别率。2)将多个神经网络进行组合,提高其分类精度^[3-4]。

本文提出了一种基于统计信息获取低维特征——轮廓骨架特征的新方法,并结合粗网格特征设计了一个两阶段识别

方案。在第一阶段使用基于粗网格特征的分类器进行粗分类,淘汰大部分负样本,而使几乎所有的正样本通过。第二阶段由基于轮廓骨架特征的分类器对通过第一阶段的样本进一步淘汰识别。实验使用从实际财务报表上提取的5000个手写体数字样本,经过训练后,测试识别率高达99.7%,并且单字的识别时间均小于3ms,取得了较好的效果。

1 特征提取

1.1 预处理

市场上现有的数字字符识别系统在预处理阶段通常包括:图像的灰度化、二值化、平滑去噪、断笔补偿、近距离连通域连接、字符分割、倾斜矫正、规范化(归一化)、细化、位置规整化等步骤。本文采取基于统计信息来获取特征向量的办法,只需要进行灰度化、二值化、平滑去噪、断笔补偿、近距离连通域连接、字符分割、规范化这些操作^[5],而无须进行细化、位置规整化等,从而简化了预处理的过程,加快了算法的速度。

在下文中如无特殊说明,均假定已对字符图像进行必要的预处理,尺寸规范化后其外框大小为20×20(以像素为单位),顶行与底行均有黑色像素。

1.2 粗网格特征^[6]

粗网格特征提取的基本思想是:先把字符图像划分为 n

收稿日期:2008-11-07;修回日期:2009-01-15。

作者简介:高宏宾(1960-),男,陕西耀县人,副教授,博士,主要研究方向:数据库、数据挖掘;陈军(1979-),男,广东湛江人,硕士研究生,主要研究方向:数据库、数据挖掘;陈丽平(1981-),女,广东湛江人,实习研究员,硕士,主要研究方向:人工神经网络。

个小区域,然后统计每个小区域中黑色像素占总黑色像素的百分比作为该区域的特征值, n 个特征值按照一定的次序组成的特征向量就是所求的粗网格特征。此种特征能很好地反映字符的整体分布情况,具有极强的抗噪声能力。对于笔画简单而风格多样的手写体数字来说,是一种难得的粗分类特征。

针对 20×20 的点阵,将其平均划分成 16 块,每块的大小为 5×5 ,可得到一个 16 维的粗网格特征。

1.3 轮廓骨架特征

为了便于描述,下面先给出四个相关的定义,然后再给出轮廓骨架特征(Contour-Skeleton, CS)的提取思想及算法。

定义 1 对数字二值化点阵图像中的某一行,以该行中最左的黑色像素为起点,水平向左扫描到字符边框,所经过的像素数,称为该行的左边缘特征值。 n 个左边缘特征值从上到下按顺序排列所构成的特征向量,称为左边缘(left-side)特征。与左边缘特征相类似,可定义数字的右边缘(right-side)特征。

设 $Ls(k)$ 、 $Rs(k)$ 分别为第 k 行的左、右边缘特征值:

$Ls(k) = j_0, 0 \leq k < 20, j_0$ 满足 $f(k, j_0) = 1$ 且当 $j < j_0$ 时, $f(k, j) = 0$

$Rs(k) = j_0, 0 \leq k < 20, j_0$ 满足 $f(k, j_0) = 1$ 且当 $j > j_0$ 时, $f(k, j) = 0$

其中 $f(k, j)$ 代表字符图像第 k 行,第 j 列的像素颜色,1 代表黑色,0 代表白色。下文 $f(k, j)$ 均一样。

定义 2 数字字符在第 k 行的宽度定义为:

$$W(k) = 20 - Ls(k) - Rs(k) + 2; 0 \leq k < 20$$

定义 3 左、右边缘的一阶差分分别定义为:

$$Ldif(k) = \begin{cases} 0, & k = 0 \\ Ls(k) - Ls(k-1), & 0 < k < 20 \end{cases}$$

$$Rdif(k) = \begin{cases} 0, & k = 0; \\ Rs(k) - Rs(k-1), & 0 < k < 20 \end{cases}$$

定义 4 对数字二值化点阵图像中的某一行,以字符外框在该行的左边缘为起点,水平向右扫描,到右边框为止,所经过的黑像素段(有效笔画数),称为该行的行分段特征值。 n 个行分段特征值从上到下按顺序排列所构成的特征向量,称为行分段特征。

设 $Rw(k)$ 为第 k 行的行分段特征值:

$$Rw(k) = j_0; 0 \leq k < 20$$

其中:若 $j_0 < 1$,则取 j_0 为 1,若 $j_0 > 3$,则取 j_0 为 3。

所谓轮廓骨架特征是指根据字符的轮廓、骨架特性对行分段特征进行修改及压缩而得到的特征向量。其基本思想是:当行分段特征值连续为 1 时,用方向向量来代替原来的行分段特征值;当行分段特征值发生变化时(1→2,2→1,2→3,3→2),判断字符在该处是否开口,若开口,则进一步区分所增加(或减少)的笔画在左边,还是在右边。对于不同的情况分别用不同的数值来代替原来的行分段特征值。最后删除无关紧要的特征值即可得到轮廓骨架特征。提取算法如下。

1) 初始化,令 $CS(k) = 0, 0 \leq k < 20$ 。

2) 当字符第 k 行的行分段特征值为 1,即 $Rw(k) = 1$ 时, $CS(k)$ 取值可分为三种情况:

a) $k = 0$ (即第 $k - 1$ 行是边界) 或 $Rw(k - 1) = 3$, 令 $CS(k) = 0.50$ 。

b) $k > 0$ 且 $Rw(k - 1) = 1$, 根据相邻两段笔画的方向矢量,使 $CS(k)$ 取 7 种不同值,设方向矢量:

$$Y = ((Ls(k) + 1/2W(k)) - (Ls(k - 1) + 1/2W(k -$$

$$1))) / 1 = 1/2(Ldif(k) - Rdif(k)); 0 < k < 20$$

令:

$$CS(k) = \begin{cases} 0.35, & Y < -2.5 \\ 0.40, & -2.5 \leq Y < -1 \\ 0.45, & -1 \leq Y < 0 \\ 0.50, & Y = 0 \\ 0.55, & 0 < Y \leq 1 \\ 0.60, & 1 < Y \leq 2.5 \\ 0.65, & Y > 2.5 \end{cases}$$

c) $k > 0$ 并且 $Rw(k - 1) = 2$, 根据字符笔画数从 2 到 1 过度时是否有开口,以及开口方向,使 $CS(k)$ 取 3 种不同的值,设 X_1 、 X_2 分别代表左、右开口大小(即开口处白色像素的数目):

$$X_1 = \begin{cases} 0, & Ldif(k) \leq 0 \\ \sum_{i=Ls(k-1)}^{i=Ls(k)} (1 - f(k-1, i)), & Ldif(k) > 0 \end{cases}$$

$$X_2 = \begin{cases} 0, & Rdif(k) \leq 0 \\ \sum_{i=20-Rs(k)}^{i=20-Rs(k-1)} (1 - f(k-1, i)), & Rdif(k) > 0 \end{cases}$$

令:

$$CS(k) = \begin{cases} 0.50, & X_1 \leq \beta \text{ 且 } X_2 \leq \beta \\ 0.07, & X_1 > \beta \text{ 且 } X_1 \geq X_2 \\ 0.80, & X_2 > \beta \text{ 且 } X_1 < X_2 \end{cases}$$

其中 β 为阈值,本实验取 $\beta = 3$,下文 β 均一样。

3) 当字符第 k 行的行分段特征值为 2,即 $Rw(k) = 2$ 时, $CS(k)$ 取值可分为四种情况:

a) $k = 0$ (即第 $k - 1$ 行是边界),令 $CS(k) = 0.02$ 。

b) $k > 0$ 且 $Rw(k - 1) = 1$, 根据字符笔画数从 1 到 2 过度时是否有开口,以及开口方向,使 $CS(k)$ 取 3 种不同的值,设 X_3 、 X_4 分别代表左、右开口大小(即开口处白色像素的数目):

$$X_3 = \begin{cases} 0, & Ldif(k) \geq 0 \\ \sum_{i=Ls(k)}^{i=Ls(k-1)} (1 - f(k, i)), & Ldif(k) < 0 \end{cases}$$

$$X_4 = \begin{cases} 0, & Rdif(k) \geq 0 \\ \sum_{i=20-Rs(k-1)}^{i=20-Rs(k)} (1 - f(k, i)), & Rdif(k) < 0 \end{cases}$$

令:

$$CS(k) = \begin{cases} 0.30, & X_3 \leq \beta \text{ 且 } X_4 \leq \beta \\ 0.02, & X_3 > \beta \text{ 且 } X_3 \geq X_4 \\ 0.75, & X_4 > \beta \text{ 且 } X_3 < X_4 \end{cases}$$

c) $k > 0$ 且 $Rw(k - 1) = 2$, 令 $CS(k) = 0.30$ 。d) $k > 0$ 且 $Rw(k - 1) = 3$, 根据开口方向(即减少的笔画在左边还是右边)及开口大小,使 $CS(k)$ 取 3 种不同的值。计算开口大小的办法与第二步中计算 X_1 、 X_2 的办法一样,令:

$$CS(k) = \begin{cases} 0.30, & X_1 \leq \beta - 1 \text{ 且 } X_2 \leq \beta - 1 \\ 0.20, & X_1 > \beta - 1 \text{ 且 } X_1 \geq X_2 \\ 0.95, & X_2 > \beta - 1 \text{ 且 } X_1 < X_2 \end{cases}$$

4) 当字符第 k 行的行分段特征值为 3,即 $Rw(k) = 3$ 时, $CS(k)$ 取值可分为三种情况:

a) $k = 0$, 或 ($k > 0$ 且 $Rw(k - 1) = 3$), 令 $CS(k) = 0.70$ 。

b) $k > 0$ 且 $Rw(k - 1) = 1$, 令:

$$CS(k) = \begin{cases} 0.70, & Ldif(k) \geq -\alpha \text{ 且 } Rdif(k) \geq -\alpha \\ 0.15, & Ldif(k) < -\alpha \text{ 且 } Ldif(k) \leq Rdif(k) \\ 0.90, & Rdif(k) < -\alpha \text{ 且 } Ldif(k) > Rdif(k) \end{cases}$$

其中 α 是阈值,本实验中取 $\alpha = 4$ 。

c) $k > 0$ 且 $Rw(k-1) = 2$, 根据开口方向(即增加的笔画在左边还是右边)及开口大小,使 $CS(k)$ 取 3 种不同的值。计算开口大小的办法与第三步中计算 $X3$ 、 $X4$ 的办法一样,令:

$$CS(k) = \begin{cases} 0.70, & X_3 \leq \beta - 2 \text{ 且 } X_4 \leq \beta - 2 \\ 0.15, & X_3 > \beta - 2 \text{ 且 } X_3 \geq X_4 \\ 0.90, & X_4 > \beta - 2 \text{ 且 } X_3 < X_4 \end{cases}$$

5) 去掉四个重要性较轻的 $CS(k)$ 值,从而减少特征维度。原则如下:

① 若字符有连续两行较宽的单笔画,且错位较小时,可将其中一行删除,如图 1(b) 所示。即 $CS(k) = 0$ (将第 k 维特征删除),若满足:

$$k > 0 \text{ 且 } Rw(k) = 1 \text{ 且 } Rw(k-1) = 1 \text{ 且 } W(k) > \delta \text{ 且 } |Ldif(k)| + |Rdif(k)| < \varepsilon$$

其中 δ, ε 是阈值,本实验分别令 $\delta = 8, \varepsilon = 3$ 。

② 若首尾两行是单笔画且笔画较窄时,均可删除,如图 1(c) 所示。即 $CS(k) = 0$ (将第 k 维特征删除),若满足:

$$W(k) < \delta - 3 \text{ 且 } Rw(k) = 1 \text{ 且 } (k = 0 \parallel k = 19)$$

③ 找出字符中连续取相同行分段特征值最多的部位,将其中关键度最小的一行删除,如图 1(d) 所示。

设整个字符被从上到下水平划分为 N 块 ($1 \leq N < 20$), 每块中各行的行分段特征值相同,任何相邻两块的行分段特征值相异。即字符可表示为:

$$C = \{C_0, C_1, \dots, C_{(N-1)}\}$$

其中 C_i 是若干行相邻字符像素的集合,可按以下形式表示:

$$C_i = \{l_{d+1}, l_{d+2}, \dots, l_{d+|C_i} \}$$

其中 l_w 代表一行像素, w 是行号 ($0 \leq w < 20$), $|C_i|$ 代表集合 C_i 的元素个数。对于任何 $l_a, l_b \in C_i$, 均有 $Rw(a) = Rw(b)$, 且对于任何 ($l_p \in C_i \wedge l_q \in C_{(i+1)}$), 均有 $Rw(a) \neq Rw(b)$ 。

令 $CS(k) = 0$ (将第 k 维特征删除), 若满足:

$$\begin{aligned} & l_k \in C_{\max} \text{ 且 } Rw(k) = 2 \text{ 且 } CS(k) = 0.30 \\ & \text{或 } l_k \in C_{\max} \text{ 且 } Rw(k) = 3 \text{ 且 } CS(k) = 0.70 \\ & \text{或 } l_k \in C_{\max} \text{ 且 } Rw(k) = 1 \text{ 且 } (|Ldif(k)| + |Rdif(k)|) = \end{aligned}$$

$$\min_{i \in C_{\max}} \{ |Ldif(i)| + |Rdif(i)| \}, \text{ 其中 } C_{\max} \in C \text{ 且 } |C_{\max}| = \max_{C_i \in C} \{ |C_i| \}$$

以上三条原则,优先考虑 ①,再到 ②,最后考虑 ③,其中

③ 可反复执行,结束条件是特征维度降到 16。



(a) 压缩前 (b) 原则① (c) 原则② (d) 原则③ (e) 压缩后
图 1 降维过程

容易证明,提取轮廓骨架特征的时间复杂度为 $O(n)$, n 为字符图像的规模(像素总数)。

2 神经分类网络

2.1 AdaBoost 算法

AdaBoost 算法的基本思想是:在给定的训练集上,反复选择特征构建两值弱分类机,通过加权的办法组合构成一个强分类机。具体算法描述如下:

1) 弱学习器数量 $1, 2, \dots, T$;

2) 所有学习器共用一个样本集合: $X = [x_1, x_2, \dots, x_N]$;

3) 学习器训练之前首先对每个样本进行加权,不同权重样本在训练中被“关注”的程度不同,权矢量为: $W_t = [w_{t1}, w_{t2}, \dots, w_{tN}]$;

4) 不同学习器采用不同权矢量, W_t 是学习器 t 所采用的权矢量,它由学习器 ($t-1$) 的学习误差而定, w_{tn} 是学习器 t 对样本 x_n 的加权系数。

两类学习样本集合: $X = [x_1, x_2, \dots, x_N]$ 和 $D = [d_1, d_2, \dots, d_N]$, 其中 X 是样本矩阵, D 是理想输出矢量, $d_n \in \{1, 0\}$ 。

学习器 1 对每个样本都“一视同仁”,即,取 W_1 中元素: $w_{11} = w_{12} = \dots = w_{1N} = 1/N$ 。

AdaBoost 对弱学习器 $t = 1, 2, \dots, T$ 执行以下算法:

1) 在加权矢量 W_t 下训练学习器 t 。

2) 得到弱假设 $H_t = X \rightarrow V$ 。

3) 计算 H_t 的错误率: $\varepsilon_t = \sum_{i: H_t(x_i) \neq d_i} w_{ti}$ 。

4) 令 $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$ 。

5) 更新权值:

$$\bar{w}_{(t+1)i} = w_{ti} \times \begin{cases} \beta_t, & H_t(x_i) = d_i \\ 1, & H_t(x_i) \neq d_i \end{cases}$$

$$w_{(t+1)i} = \frac{\bar{w}_{(t+1)i}}{\sum_{n=1}^N \bar{w}_{(t+1)n}}$$

$$W_{t+1} = [w_{(t+1)1}, w_{(t+1)2}, \dots, w_{(t+1)N}]$$

工作时假设输入矢量为 x , 各弱学习器的输出分别为 $h_1(x), h_2(x), \dots, h_N(x)$, 则总输出即对矢量 x 的最终判断为:

$$H(x) = \begin{cases} 1, & \sum_{i=1}^T a_i h_i(x) \geq \frac{1}{2} \sum_{i=1}^T a_i \\ 0, & \text{其他} \end{cases}$$

其中 $a_i = \log \frac{1}{\beta_i}$ 。

显然,工作时在对一个具体的输入矢量 x 所做的最终判断结果中不仅有各子网模块的实际输出,也考虑了学习过程中它们的表现情况。

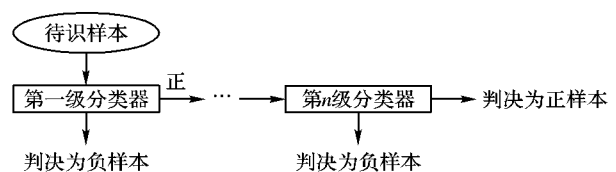


图 2 级联结构分类器模型

2.2 AD AdaBoost 算法

AdaBoost 算法在实际应用中已经有很多研究了,其中最著名的是基于级联结构的 AdaBoost 算法,结构如图 2 所示,该算法在人脸检测的研究中取得了很好的效果。在级联结构中,强分类器的复杂程度随着级数的增加而增加。大多数的负样本都在前面比较简单的级次中被淘汰,而使几乎全部的正样本顺利通过。尽管级数越高,分类器越复杂,但是越往后需要识别的候选样本越少,因此使系统具有很好的实时性。

在级联结构的基础上,文献[8]进一步对 AdaBoost 算法作了改进,提出 AD AdaBoost 算法。该算法主要的改进是:在级联结构的各个层次中,在保证正样本高通过率的同时,有效

地降低了负样本的错误率。

新的算法仅采用了与传统 AdaBoost 算法不同的加权参数的求解公式,也就是只需要将式(4)改为令:

$$\beta_i = \frac{1}{2} \ln \left[\frac{1 - \varepsilon_i}{\varepsilon_i} \right] + \kappa \cdot e^{p_i}$$

其中, κ 为常数, $p_i = \sum_{i: y_i=1, H(x_i)=1} w_{i0}$

2.3 手写体数字模糊分类系统的设计

AdaBoost 算法是二分类算法,而手写体数字识别问题是多分类问题,因此必须对 AdaBoost 分类器进行组合变形,使其满足多分类要求。在本系统中,AdaBoost 分类器采用了两级级联结构,第一级以 16 个粗网格特征值作为输入变量;第二级以 16 个轮廓骨架特征值作为输入变量。分类系统框架如图 3 所示。其中 C_i 是第一级分类器; D_i 是第二级分类器。系统中任一分类器都只有二分类功能,则只输出“是这类字符”和“不是这类字符”。

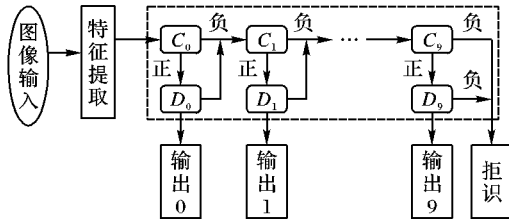


图 3 分类系统框架

在图 3 所示的系统中,提取字符特征后,首先将粗网格特征输入 C_0 , 让 C_0 识别, C_0 的输出结果有两种可能:“是 0”或“非 0”。如果 C_0 认为“是 0”,则将轮廓骨架特征输入 D_0 , 让 D_0 进一步识别,这时如果 D_0 也认为“是 0”,则输出最终结果:“0”,识别过程结束;如果其中 C_0 或 D_0 认为“非 0”,则让 C_1 及 D_1 用相同的办法判断是否为“1”。依此类推,一直到字符被识别或被拒识。

3 仿真实验与结果分析

3.1 实验系统的描述

本文采用 Visual C++ 6.0 编制了手写体数字模糊分类系统的运行程序,其中第一级以 BP 人工神经网络作为弱学习器,一个分类器包含 3 个弱学习器;第二级以支持向量机作为弱学习器,一个分类器包含 5 个弱学习器,对所有弱学习器的训练都不要求严格达到全局收敛。程序的实现包括样本训练和网络识别两大模块,其中网络识别又进一步分为扫描、预处理和识别三个步骤。系统的主界面如图 4 所示。

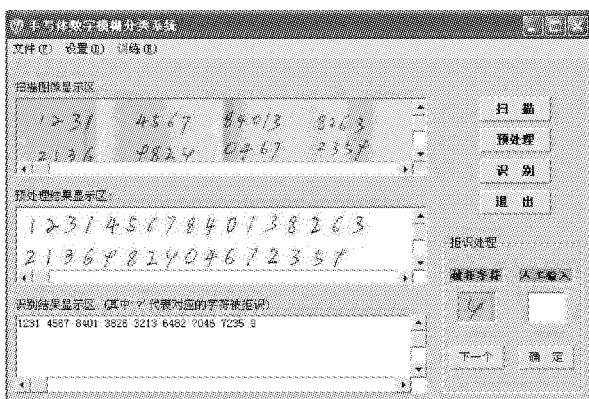


图 4 系统主界面

3.2 仿真结果分析

实验数字是从实际财务报表上提取的,0~9 每类 500

个,共 5000 个手写体数字图像。预处理后,每幅图像用 20×20 的向量表示。从中随机抽取 3000 个作为神经网络的训练样本,用文中所述的方法逐层训练 AdaBoost 分类器。待学习训练结束后,再用余下的 2000 数字作为测试样本,单个数字的识别时间均小于 3 ms。测试误差与拒识情况如表 1 所示。

将实验样本集分别应用于 13 点特征提取法^[1]、动态特征提取法(骨架特征法+笔画延续最大优势方向法)^[9]、演化算法(基于模板匹配的 AdaBoost 演化算法)^[10]和本文方法作比较分析,结果如表 2 所示(样本数目为 2000)。

由表 2 可见,本文所述的特征提取法结合 AdaBoost 神经网络所构成的识别系统取得了更好的识别效果,识别正确率有所提高,误差率和拒识率也明显下降。

表 1 测试误差与拒识情况

识别记录	字符									
	0	1	2	3	4	5	6	7	8	9
误差个数	0	0	1	0	1	1	0	0	0	1
拒识个数	0	0	0	0	1	0	0	1	0	0

表 2 识别效果比较

算法	%		
	误差率	拒识率	正确率
13 点特征法	8.50	无	91.5
动态特征法	5.60	1.20	93.2
演化算法	4.20	无	95.8
本文方法	0.02	0.01	99.7

4 结语

脱机手写体数字识别是一项相当有研究价值而极具挑战的课题,尽管人们在长期的研究中提出了大量算法,但至今为止机器的识别水平尚无法与人类相比。实验表明,尽量提取简洁而完备的特征,通过多个神经网络组合来进一步提高手写体数字识别率,是一种行之有效的方法。文中所设计的手写体数字识别系统不管在识别正确率,还是在识别速度上,都取得了比较理想的结果。

参考文献:

- [1] 宋曰聪,胡伟. 手写体数字识别系统中一种新的特征提取方案[J]. 计算机科学, 2007, 34(9): 236-239.
- [2] 叶飞,黎峰. 基于整体特征的快速手写体数字字符识别[J]. 计算机工程与设计, 2006, 27(22): 4347-4352.
- [3] 吴丽芸,王文伟,张平,等. 手写混合字符集识别的多特征多级分类器设计[J]. 计算机应用, 2005, 25(12): 2948-2950.
- [4] 赵锋,薛惠锋,王伟,等. 采用多级分类器的手写数字识别技术研究[J]. 微电子学与计算机, 2006, 23(12): 30-37.
- [5] 张猛,余仲秋,姚绍文. 手写体数字识别中图像预处理的研究[J]. 微计算机信息, 2006, 22(61): 256-258.
- [6] YU HU, LI MING-JING, ZHANG HONG-JING, et al. Color texture moments for content-based image retrieval [C]// Proceedings of 2002 International Conference on Image Processing: ICIP 2002. Washington, DC: IEEE Press, 2002, 3: 929-932.
- [7] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of online learning and an application to Boosting[J]. Journal of Computer and System Sciences, 1997, 55(1): 119.
- [8] 李闯,丁晓青,吴佑寿. 一种改进的 AdaBoost 算法——AD AdaBoost 算法[J]. 计算机学报, 2007, 30(1): 103-109.
- [9] 范艳峰,肖乐,甄彤. 自由手写体数字识别技术研究[J]. 计算机工程, 2005, 31(10): 168-170.
- [10] 陈家辉,付忠良,陈腊梅. 基于模板匹配的 AdaBoost 演化算法[J]. 计算机应用, 2007, 27(12): 3072-3074.