

基于结构特征分类 BP 网络的手写数字识别

耿西伟, 张 猛, 沈建京

(解放军信息工程大学, 河南 郑州 450001)

摘 要: 手写体数字识别有着重大的使用价值, 用多层 BP 网络来识别手写体数字是手写体数字识别的一大进步, 但是, 用单纯的 BP 网络来识别也存在识别精度不高等的问题。将 BP 网络技术和数字本身的结构特征结合起来, 提出了一种基于结构特征分类 BP 网络的手写体数字识别新方法。首先提取点、环等数字特征值, 并根据一些特征进行分类; 然后再运用 BP 神经网络识别, 以提高网络的识别能力; 最后, 选取了 500 个人的 0~9 的手写体数字, 运用以上算法进行 BP 神经网络识别, 用 3000 个手写体数字作为训练样本, 2000 个其他的样本进行测试, 网络收敛后, 识别率达到 96% 以上, 比原来有一定的提高。

关键词: 手写体数字识别; 结构特征; 神经网络

中图分类号: TP18

文献标识码: A

文章编号: 1673-629X(2007)01-0130-03

Recognition of Handwritten Numerals with Grouped BP Net Based on Structural Features

GENG Xi-wei, ZHANG Meng, SHEN Jian-jing

(PLA Information Engineering University, Zhengzhou 450001, China)

Abstract: Handwritten numeral recognition has important use value. It is a great advancement to use multi-layer BP network to identify numbers write by hand. But it has problems to use single BP network to identify. Combines BP network with the character of number structure, then a new method of grouped BP net based on structural features is proposed to classify handwritten numbers. Point and circle features cell are extracted and combined. Then, BP net is grouped based on some structural features. The system recognizes number by grouped BP neural network, therefore it obtains better effect. Finally, select the handwriting of 500 people from 0 to 9, using arithmetic to recognize(3000 swatches for training, 2000 for testing). After net constringed, the distinguishing rate is over 96%.

Key words: handwritten numeral recognition; structural features; neural network

0 引 言

手写数字识别 (Handwritten Numeral Recognition)^[1] 是光学字符识别技术 (Optical Character Recognition, 简称 OCR) 的一个分支, 它研究的对象是: 如何利用电子计算机自动辨认人手写在纸张上的阿拉伯数字。随着社会信息化的发展, 手写数字的识别研究有着重大的实用价值, 如在邮政编码、税务报表、统计报表、财务报表、银行票据、海关等需要处理大量字符信息录入的场合。

手写体数字识别方法大体可分为两类^[2]: 基于统计的识别方法和基于结构的识别方法。第一类方法包括模板匹配法、矩法、笔道的点密度测试、字符轨迹法及数字变换法等。第二类则是尽量抽取数字的骨架或轮廓特征, 如环路、端点、交叉点、弧状线、环及凹凸性等, 两类方法具有一定的互补性。因此文中使用了一个基于结构特征的分类识别器。

手写体数字用多层 BP 网络来识别可以采用两种输入网络的形式^[3]: 一种是点阵 (0,1 点阵) 直接输入网络, 利用网络来抽取特征并进行分类, 这也叫点阵输入网络; 另一种是通过一些算法, 抽取字符特征, 然后将一组特征值输入网络, 利用神经网络对特征分类, 达到识别字符的目的, 这也叫作特征输入网络, 它仅起分类作用。对于识别手写体数字, 特征输入网络要比点阵输入网络效果好。特征输入网络多数是直接输入提取的所有字符特征, 一般来说, 特征输入的越多识别才能越准确。但是太多的输入会使网络变的很大, 难于收敛或者收敛到局部极小点。可以先对待识别数字进行粗特征分类, 其作用是根据一些简单的特征对数字分类, 选择具有这些简单特征的数字准备进行进一步识别; 然后再提取其他特征, 输入粗分类中选中的数字判别网络进行判别。

1 预处理

1.1 一般的处理方法

图像预处理是字符识别重要的一环, 它把原始的图像转换成识别器所能接受的二进制形式。要识别手写体数

收稿日期: 2006-03-26

作者简介: 耿西伟 (1981-), 男, 河南开封人, 硕士研究生, 研究方向为人工智能; 沈建京, 博士, 博导, 研究方向为人工智能。

字首先要对其字符图像进行预处理。预处理的主要目的是去除字符图像中的噪声、压缩冗余信息,得到规范化的点阵,为特征提取做好准备。特征输入网络的识别效果要取决于特征集的完备性,这就要求预处理在消除图像中与识别无关的因素时尽量保持原图像的字符特征。在预处理阶段,针对手写体字符的特点,对字符进行了较为完善的预处理,包括二值化、梯度锐化、平滑去噪、分割、倾斜调整、断笔填充、小归一化、细化。如图1所示。

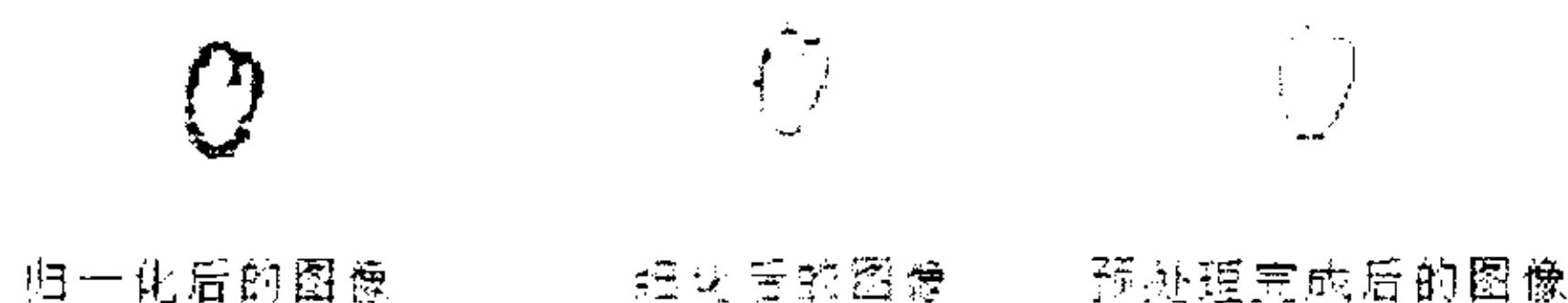


图1 归一化细化和预处理后的图像比较

1.2 特殊的处理

1.2.1 去除毛刺

经细化后的图像,由于噪声及笔道粗细的影响,骨架中仍会有不同程度的枝节无用信息,产生一些毛刺。因此要减少样本的异体字种类,使同一个数字的具有相似拓扑结构的样本尽可能地统一,还要简化骨架,仅保留能体现数字字形本身结构特点的线条,这样可以提高识别率和减小拒识率。由于骨架中的毛刺长短不一,不可能用一个固定问值一次性全部除去;而且有些毛刺与较短的笔划相近,可能造成短笔划被删除,而毛刺被保留的情况,所以删除毛刺的情况就比较复杂。

经细化后的图像,由于噪声及笔道粗细的影响,会产生一些毛刺。对于细化之后的图像,规定手写体对应的像素为黑,用“1”表示;背景对应的像素为白,用“0”表示。根据每一个黑像素点 (x, y) 周围相临8个位置为1的点的个数 c 来定义端点与节点。若 $c = 1$,则此点为端点;若 $c \geq 2$,则为节点^[3]。

$$c = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} f(i, j) - 1$$

检测端点与端点及端点与

节点之间线段的像素数目,若像素数目小于某一阈值,则将此线段视为毛刺,予以去除。

1.2.2 去除闭合环内的线段

对于数字而言,闭环内的线段对识别没有任何意义。在细化后图像的基础上,去除这些线段。对于细化后的数字图像,以区域生长法将其内环予以标记,然后检测各个端点,若某一端点的八邻域中有7点有同一内环标记,则以此端点为起始点的线段必为闭环内线段,可以将此线段去除。

1.2.3 去除花饰环

闭合环的提取即检测数字中出现的圆环或椭圆。有许多种检测闭合圆,即提取检测数字中出现的圆环或椭圆环的方法,这里采用在环中建立一个像素成员表的方法得到检测环,建立的链表的个数即为环的个数;每一条链表

的长度即为环内像素的个数,也就是环的质量。在许多字符图像中,发现数字中含有一种很小的环,容易产生干扰称之为花饰环。通过设立一个阈值,来区分真正的环和花饰环。如果环的质量低于阈值,此为一个花饰环,去除环上除节点以外的点,这样花饰环中的四叉点(节点中 $c = 4$ 的点)就没有了,变成了拐点。如图2所示。

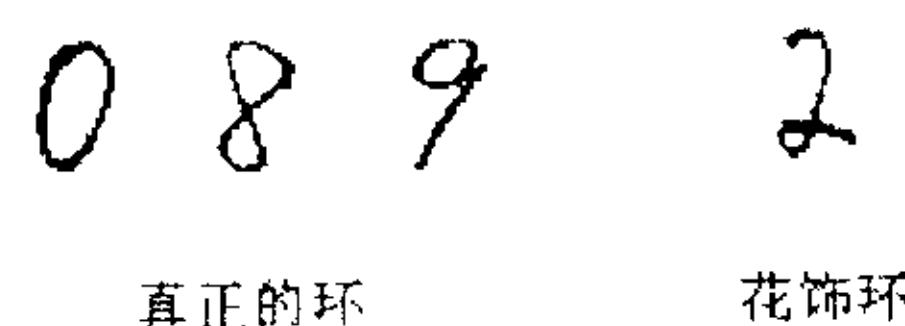


图2 真正的环与花饰环对比效果图

2 手写体数字的分类识别

2.1 特征的选择

特征的选择^[4]是很重要的,其好坏直接关系到字符的分类,决定着网络识别的能力。因此,选择的特征必须能充分反映字符的本质特征,对于不同的手写体数字,有较强的分辨能力。首先,考虑点特征。对一个字符图像,有无四叉点的属性的不确定性最好,可以作为粗分类的特征,对字符图像进行一级分类。端点的特征包括端点的位置、方向。把图像分为四等份,端点的位置分为左上、左下、右上、右下。端点的方向指这点与沿轮廓点扫描到下一个特征点的连线方向,分为四个方向,考虑手写体数字的特点和书写习惯,优先次序分别向右、向下、向左、向上。拐点特征包括图像中拐点的个数、位置以及左、右拐点。端点特征、拐点特征可以作为细分类的特征,进行子网络分类识别。如图3所示。

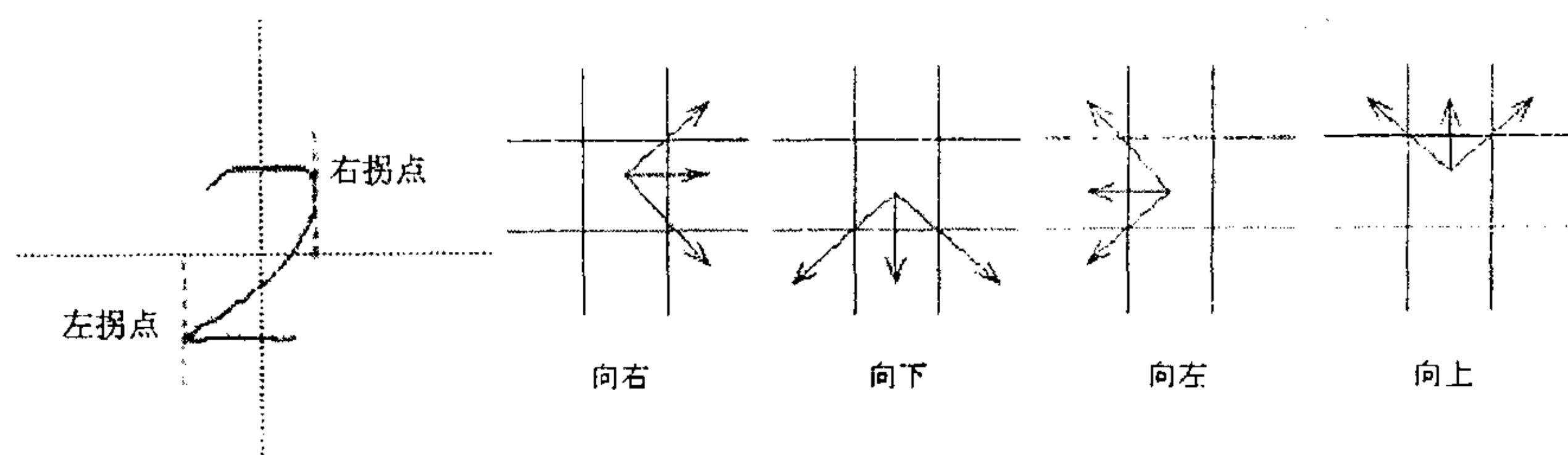


图3 点特征的位置和端点的方向

然后,考虑字符图像中是否存在环,作为二级分类的特征。

最后,还要考虑字符宽度特征。字符宽度特征是一个相对值。它表示的是字符最大宽度与最小宽度的比值,即 $\eta = W_{\max}/W_{\min}$ 。该特征对于将1与其他字符分开是很有效的,可以作为二级分类的特征。值得注意的是,由于起笔和落笔处的顿笔和笔锋的关系,最小宽度值一般不应取最顶端和最底端的值。在扫描字符时应从某个合适的常数 α 开始到 $y_{\max} - \alpha$ 为止。

2.2 识别算法

一般来说,字符识别系统分为预处理、特征提取、判别、后处理等模块。本算法不是把提取到的特征直接输入BP网络进行判别,而是先根据一些特征进行简单的分类,然后再分组识别来提高网络识别的性能^[5]。

首先进行粗分类,利用主要结构特征判别数字属于哪个特征类;第二阶段,在所属特征类中检查是否能再次分组,如不能再分组则进行细分类,输入 BP 网络进行识别,系统图如图 4、图 5 所示。因为采用了对数字粗分类的方法,所以可以减少查找特征次数,降低网络的规模,提高识别速度。

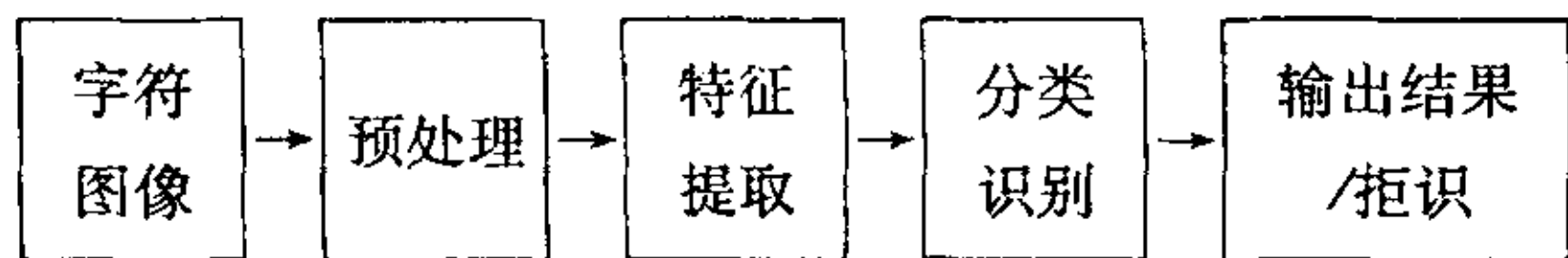


图 4 字符识别流程图

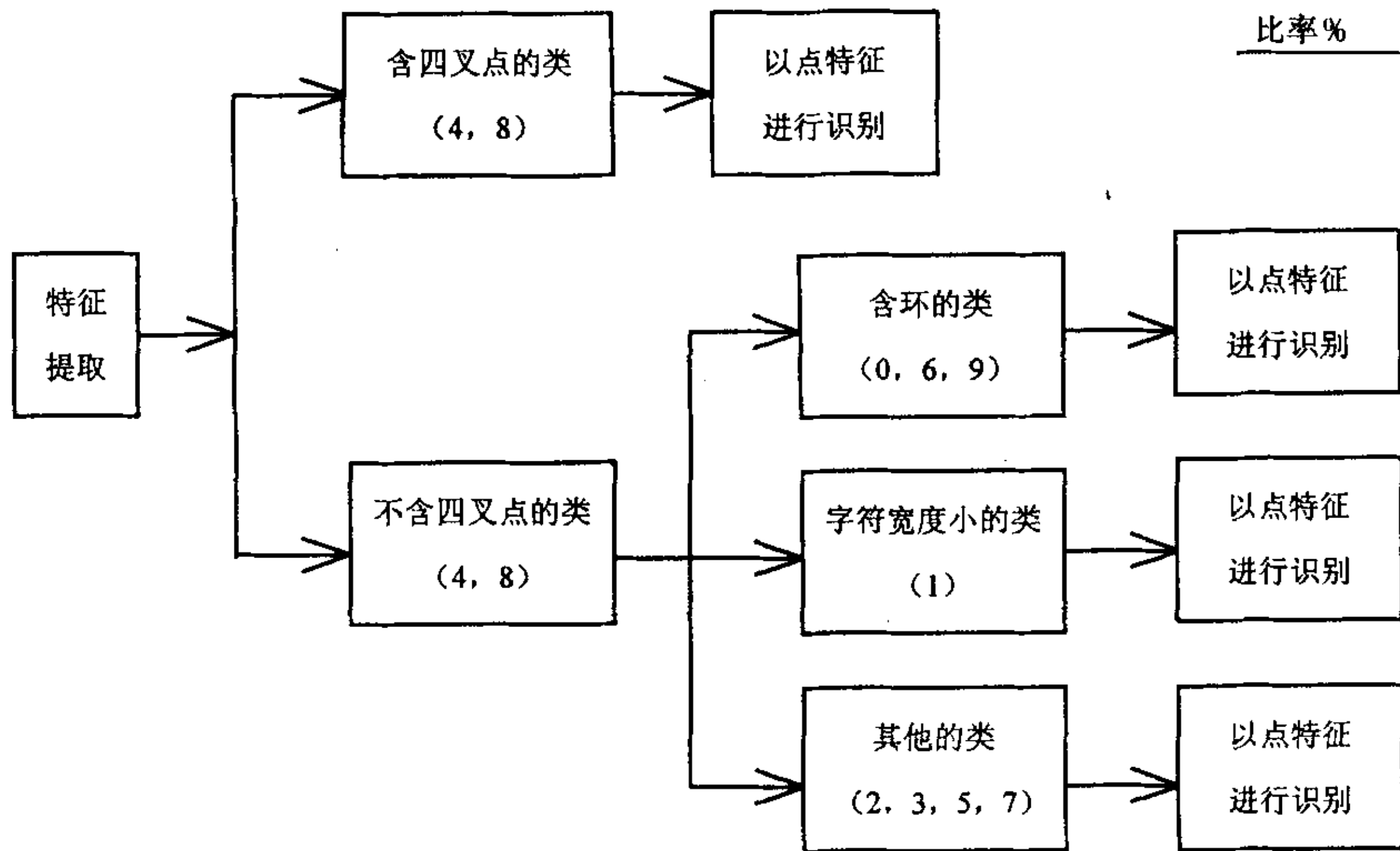


图 5 分类识别流程图

3 结 论

选取了 500 个人的 0~9 的手写体数字,运用以上算法进行 BP 神经网络识别,用 3000 个手写体数字作为训练样本,2000 个其他的样本进行测试,网络收敛后,识别率达到 96% 以上(见表 1)。

(上接第 129 页)

3 函数调用与异常处理的异同

最后对比函数调用和异常处理之间的异同。throw 表达式的行为有点像函数的调用,而 catch 子句有点像函数定义。函数调用和异常处理的主要区别是:建立函数调用所需要的全部信息在编译时已经获得,而异常处理机制要求运行时的支持。对于普通函数调用,通过函数重载解析过程,编译器知道在调用点上哪个函数会真正被调用。但对于异常处理,编译器不知道特定的 throw 表达式的 catch 子句在哪个函数中,以及在处理异常之后执行权被转移到哪里。这些都在运行时刻决定,异常是随机发生的,处理异常的 catch 子句是沿调用链逆向进行查找,这与运行时的多态——虚函数也是不一样的。当一个异常不存在处理代码时,编译器无法通知用户,所以要有 terminate() 函数,它是一种运行机制,当没有处理代码(catch 子句)能够匹配被抛出的异常时由它通知用户^[5]。

表 1 试验结果对比表

类别	正确识别	误识	拒识	识别率%	可靠性%
0	492	3	5	98.40	99.39
1	497	0	3	99.40	100
2	485	7	8	97.00	98.38
3	478	3	19	95.60	99.38
4	480	4	16	96.00	99.17
5	489	2	9	97.80	99.59
6	482	3	15	96.40	99.38
7	490	3	7	98.00	99.39
8	479	3	18	95.80	99.38
9	472	5	23	94.40	99.95
总计	4844	33	123	96.88	99.39
比率%	96.88	0.66	2.46	96.88	99.39

本算法通过与直接用 0,1 点阵输入 BP 网络来识别的算法以及利用某些字符特征直接输入 BP 网络的算法相比,识别率有一定的提高,而且易于实现。

参考文献:

- [1] 王伟,盛立东.基于级连分组 BP 网络的高精度手写数字识别[J].中文信息学报,2000,14(2):61-62.
- [2] 付庆铃,韩力群.基于人工神经网络的手写数字识别[J].北京工商大学学报:自然科学版,2004,122(13):44-45.
- [3] Hagan M T, Demuth H B, Beale M H. Neural Network Design[M]. Beijing: China Machine Press, 2002.
- [4] 杜彦蕊.基于特征编码的手写字符识别技术.计算机工程,2004,30(5):156-157.
- [5] 朱江,宣国荣.一种基于骨架特征顺序编码的脱机手写体数字识别方法[J].小型微型计算机系统,2001,22(8):958-959.

4 结束语

程序在编写及运行时不可避免地会出现异常,使程序不能正常运行甚至崩溃,而 C++ 异常处理机制很好地解决了这一问题,使得程序更加简洁、健壮,同时也增强了程序的清晰性和可维护性。

参考文献:

- [1] 裘宗燕.C++ 语言异常处理机制的研究[J].计算机科学,2003,30(11):155-156.
- [2] Deitel H M, Deitel P J. Visual C++ .NET 高级编程[M].郭凯,等译.北京:清华大学出版社,2004.
- [3] 谭浩强.C++ 程序设计[M].北京:清华大学出版社,2004.
- [4] 吴乃陵.C++ 程序设计[M].北京:高等教育出版社,2003.
- [5] 海燕,李晓玲.异常处理技术在 C++ 中的编程实现[J].计算机技术与自动化,2005,24(3):72-74.