

ET 手写王

详细设计说明书

1 引言	3
1.1 编写目的.....	3
1.2 背景.....	3
2 程序系统的结构	4
3 图像处理模块设计说明	4
3.1 程序描述.....	4
3.2 功能.....	4
3.3 算法.....	5
3.4 输出项.....	13
4 分析模块设计说明	14
4.1 程序描述.....	14
4.2 功能.....	14
4.3 输入项.....	14
4.4 输出项.....	14
4.5 算法.....	14

1 引言

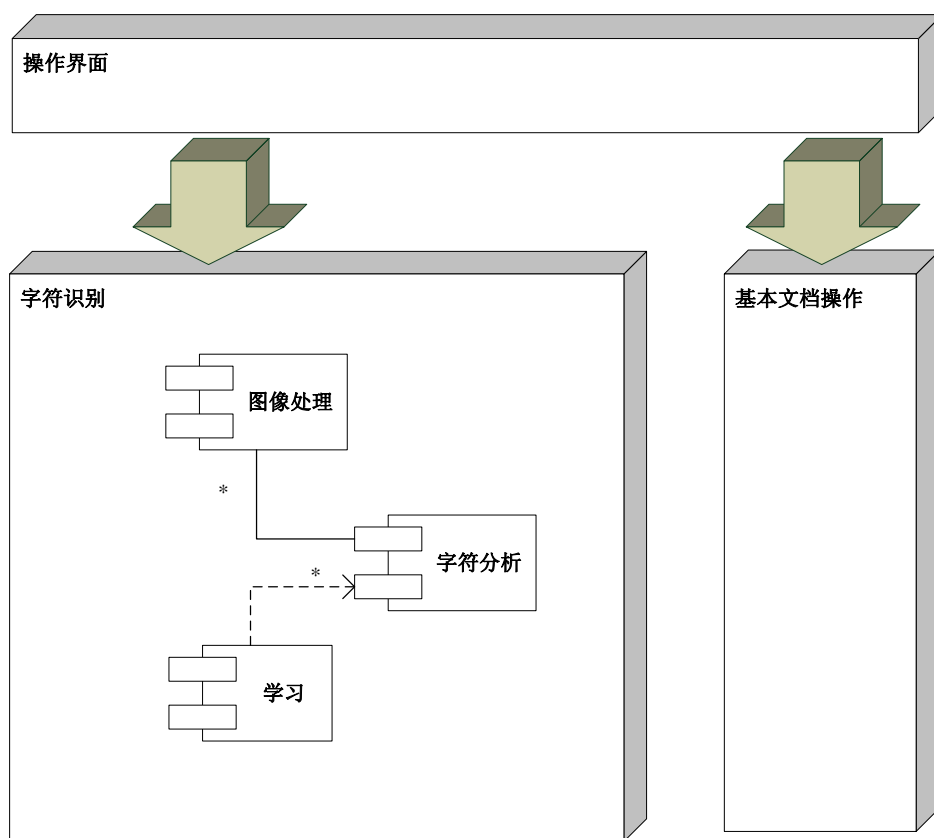
1.1 编写目的

编写这一份详细设计说明书是为了让用户能更加深入地了解ET手写王的内部结构和各个功能模块的原理。

1.2 背景

在普及计算机的时代浪潮之中，计算机走入寻常老百姓家，成为许多家庭的必备电器之一。然而键盘上密密麻麻的按键却使许多未受过计算机训练的人望而却步。人们正在寻求一种能够像过去使用铅笔一样简单的手写输入工具，ET 手写王正是为了解决这个问题而诞生的。

2 程序系统的结构



部署图

3 图像处理模块设计说明

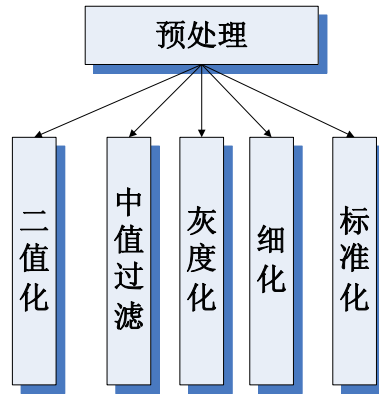
3.1 程序描述

处理用户通过手写输入的字符图像，将图像上字符笔画的特征数值化，所得结果分析将作为分析模块的输入。

3.2 功能

1. 联机识别模式：将用户输入的字符划分成小笔画，并求出每一小笔画的方向、长度、重心等。

2. 脱机识别模式：脱机识别模式下，我们一般将此过程称为预处理。预处理过程包括：灰度化、中值过滤、二值化、规范化、细化等。如下图所示：



3.3 算法

3.3.1 联机识别模式

根据用户输入时产生的鼠标事件，可以将字符的笔画分成多个小笔画。再根据小笔画的首末节点的坐标值可以求得该小笔画的角度，长度和重心。但是这些数据并不能直接给分析模块处理，仍需要进一步处理这些数据。

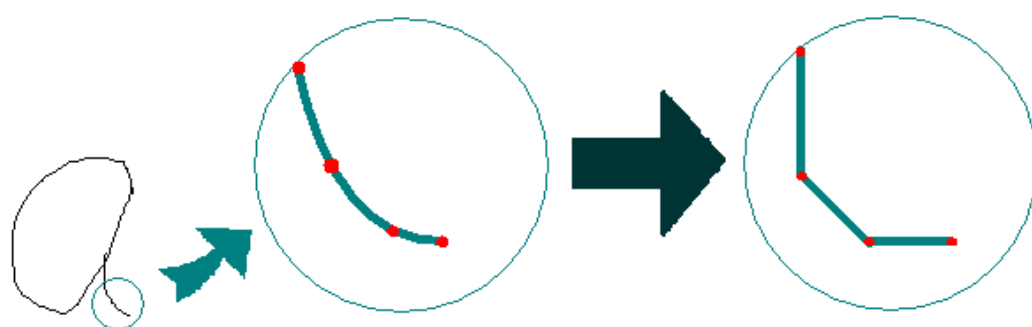
取得字符的宽、高：为了后面处理的需要，必须取得字符的宽、高。通过比较鼠标在绘图区绘出线段的节点坐标，取出最左、最右、最高、最低的坐标，通过相应坐标值相减即可得到字符的宽、高。

方向的处理：这些小笔画的角度变化范围为 0-180 度。由于手写时人手画出的线条不可能是单纯的直线，比如画横时，笔划的角度也不可能完全为 0 度，可能会稍微偏大于 0 度，也可能会稍小于 0 度。为了消除这种人手的偏差造成的判断误差，在设计时，以中间值为界

线，将各个笔画的方向归为 0 度，45 度，90 度，135 度四个方向，分别赋以 0，1，2，3 四个数值。

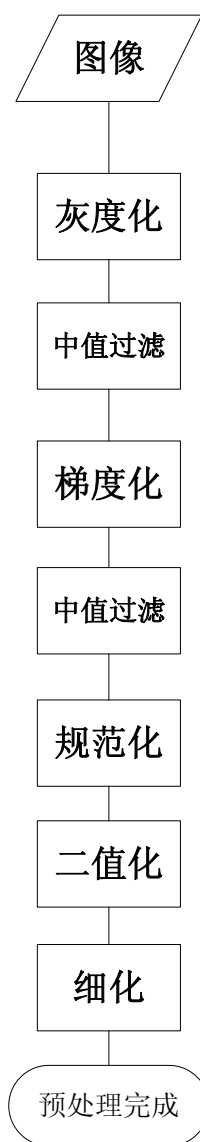
长度的处理：用户在手写区写出字符时可能有时会把字符写得比较大，笔划的长度数值就会大一点，有时写得比较小，笔划的长度数值就小。因此必须将长度数值归一化。方法是先求出所有小笔画的长度，累加之后得到整个字符的笔划长度。再把单个小笔画的长度除以整个字符的笔划总长度，得到的值其实是笔画长度的比值。

重心的处理：与上面相似，笔划的重心也必须进行归一化处理。将笔画重心的横竖坐标值分别与字宽、高相比。



字符处理图示

3.3.2 脱机识别模式



脱机识别模式下预处理流程图

3.3.2.1 灰度化

通过扫描或摄像头输入的文字图像中除了灰度色以外还有其它杂色，为了方便下一步处理，必须先将图像转化为灰度图。

3.3.2.2 中值过滤

通过光学仪器输入的文字图像可能会在某些空白处出现一些噪点，这些噪点将影响到后面的判断，应该去除这些噪点。

使用一个 3*3 的辅助矩阵扫描点阵图上的点。取出辅助矩阵中所有点灰度值的中间值 X_{mid} ,再将这个中间值赋给矩阵中的所有点。

X4	X3	X2
X5	X0	X1
X6	X7	X8

$$X_{mid} = \text{Mid}(X_0, X_1, \dots, X_8)$$

3*3矩阵

3.3.2.3 二值化

梯度处理

使用一个 3*3 的矩阵扫描图像，计算每一个点的灰度值变化度 N 。将点阵中每一个点的梯度值相加之后计算整个点阵图的梯度变化平均值 n 。再将每个点的灰度值与 N/n 相乘。以此加强图片的对比度。

X4	X3	X2
X5	X0	X1
X6	X7	X8

$$N = (X_0 - X_1) + (X_0 - X_2) + (X_0 - X_3) + (X_0 - X_4) \\ + (X_0 - X_5) + (X_0 - X_6) + (X_0 - X_7) + (X_0 - X_8)$$

3*3矩阵

二值化

为了方便处理，需要将图片转化成只有黑白两色。给定一个阈值 T ，扫描整幅图像，将灰度值小于 T 的点设成黑色，将灰度值大于 T 的点设成白色。这样就能将中间的过渡色去除。最后，再用一个 2*2

的矩阵扫描一遍点阵，每遇到黑点，则将矩阵中的所有点都填成黑色。

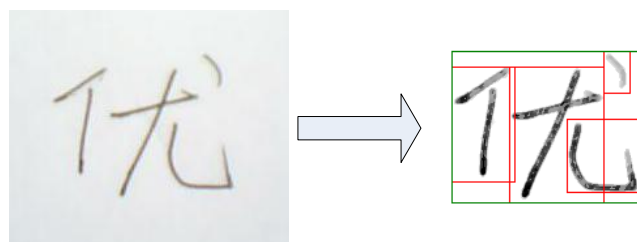
3.3.2.4 规范化

字符捕捉

由于手写的文字大小、位置可能会各有所不同，这些差异可能会成为判断失误的根源。因此有必要对输入的图像进行标准化。

先对图像进行扫描，找到黑点后，以该黑点为起点进行深度搜索，寻找下一个相邻的黑点，并记录找到的黑点的坐标。当深度搜索遍历完所有相连的黑点后，将能找到一个能容纳将所有这些相连的点阵的最小的矩形。

如此循环将能找到很多个矩形来捕获图上所有相连点。噪音产生的黑点也会被矩形捕获。因此必须设置一定的条件来筛选这些矩形。这些条件包括黑色区域面积占整个矩形面积的比重、矩形的长宽比、矩形的面积等等。符合条件的矩形被选中后，将形成一个更大的矩形来容纳所有选中的矩形。



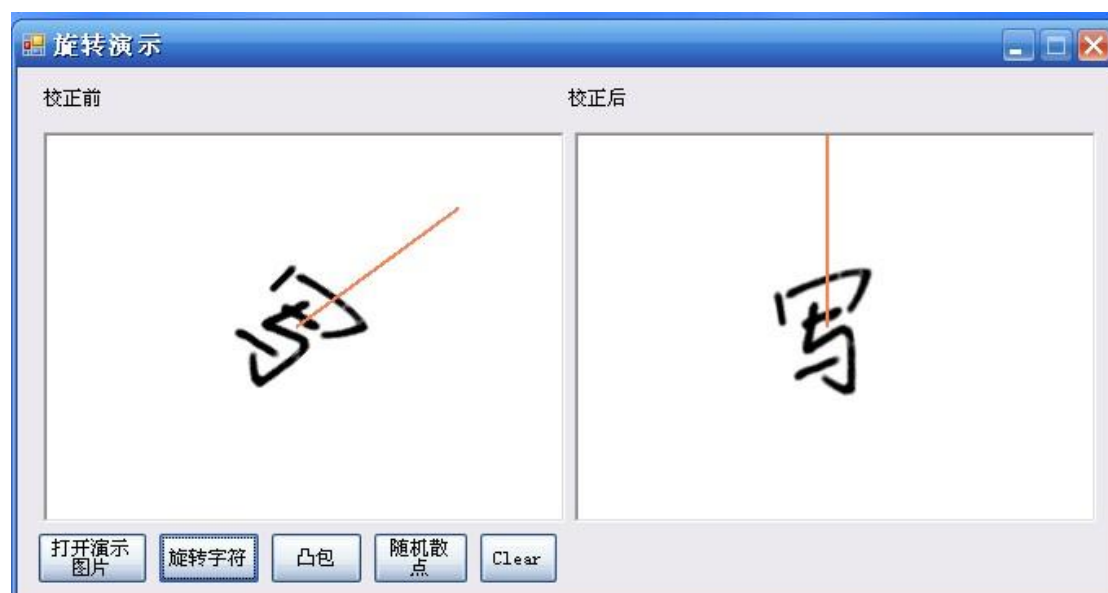
字符捕捉示例

标准化

成功捕捉字符后，需对捕捉到的字符图像进行标准化。本软件中统一将字符拉伸为 30*40（像素）。

旋转不变性

本系统使用基于协方差矩阵变换的方法，力求在最大范围内校正原输入图像的角度。我们对字符的点集进行 **K-L** 展开，所谓的 **K-L** 展开，可以理解为把原点移动到点阵的重心上，然后求出通过原点的这些点阵的一条最佳拟合直线。可以证明，最优拟合直线经过点集的均值点，并会落在点集协方差矩阵的主特征向量的方向上。



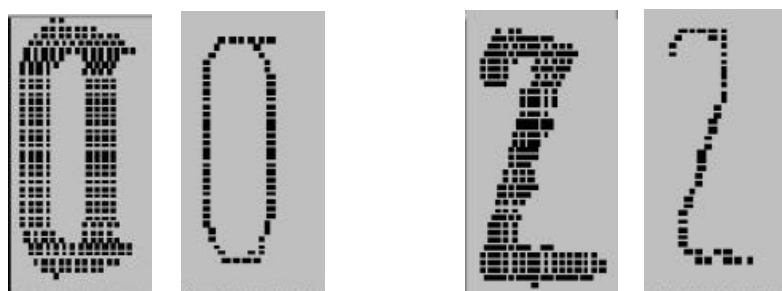
红线表示图像主特征向量的方向

3.3.2.5 细化

细化是将二值化文字点阵逐层剥去轮廓边缘上的点变成只有一个比特宽度的文字骨架图形在二值化点阵图形中对识别有价值的文字特征信息主要集中在文字骨架上细化后的文字骨架既保留了原文字绝大部分特征又利于特征提取图像细化大大压缩了原始图像的数

据量,并保持其形状的基本拓扑结构不变,从而为文字识别中的特征抽取等应用奠定了基础。

ET 手写王使用了以像素迭代删除为基础的方法,逐层细化文字图像的笔画。

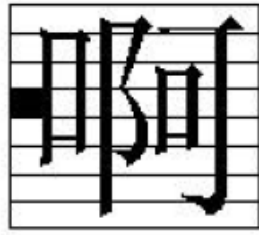


字符‘0’、‘2’细化示意图

3.3.2.5 特征向量提取

外围特征提取

ET 手写王采用基于统计量的外围特征与网格特征的方法提取字符图像的特征向量。将 30×40 的点阵图等分为 10×10 的网格。计算每一行左边缘从白变黑的像素点的面积 s_1 ,再计算每一行点阵图第二次从白变黑的像素点的面积 s_2 。由此得出 s_1 、 s_2 与整个大行的总面积 S 之比。再用上述方法提出其他三个边缘的特征。由此得出 $10 \times 2 \times 4 = 80$ 个数据作为特征向量的前 80 维。



(c)



(d)

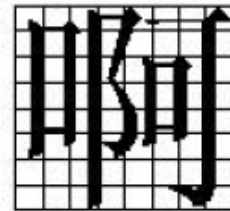
提取外围特征示意图

网格特征提取

将字符点阵划分为每个小格 $5*5$ 像素的网格，计算每个网格中黑色像素点的面积与小格面积的比。由此得到 $6*8=48$ 个数据作为特征向量的后 48 维。由此得到一个 128 维的字符特征向量。



(a)

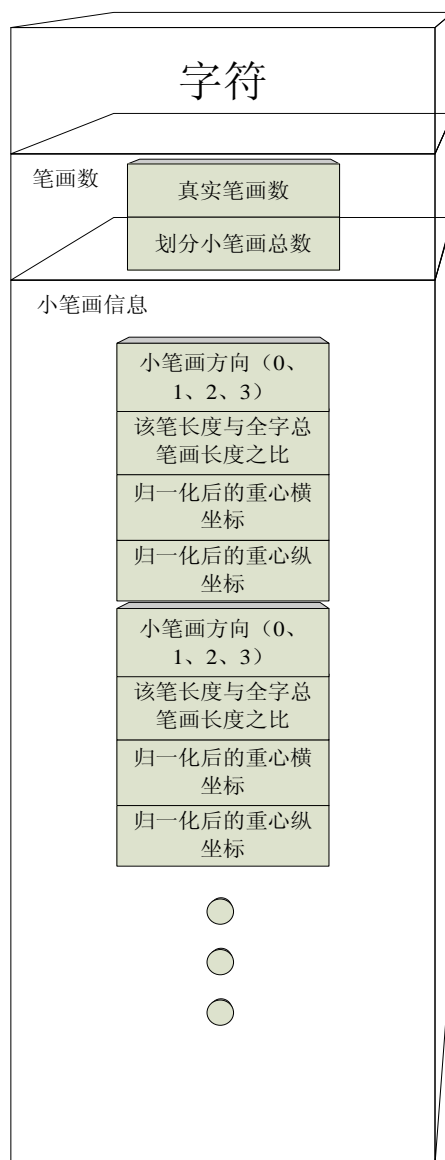


(b)

提取网格特征示意图

3.4 输出项

图像处理后将输出数值化的字符笔画特征。在内存中存储的结构如图所示：



联机模式数据存储结构图

脱机识别的输出项是一个 128 维的数组。其中前八十维是字符图像的外围特征，后 48 维是网格特征。

4 分析模块设计说明

4.1 程序描述

学习分析模块将用统计的方法，对用户手写输入的内容进行分析。

4.2 功能

分析用户手写输入的图像特征，得到分析结果，输出与分析结果匹配的项。

4.3 输入项

联机识别的输入项包括以下几项：

1. 笔画数
2. 对字符笔画的描述（详见图像处理模块设计说明）

脱机识别的输入项为一些 128 维的数据。

4.4 输出项

联机模式下输出字符库中与用户输入内容对比最相似的项。

脱机模式下将输出对应字符的编码。

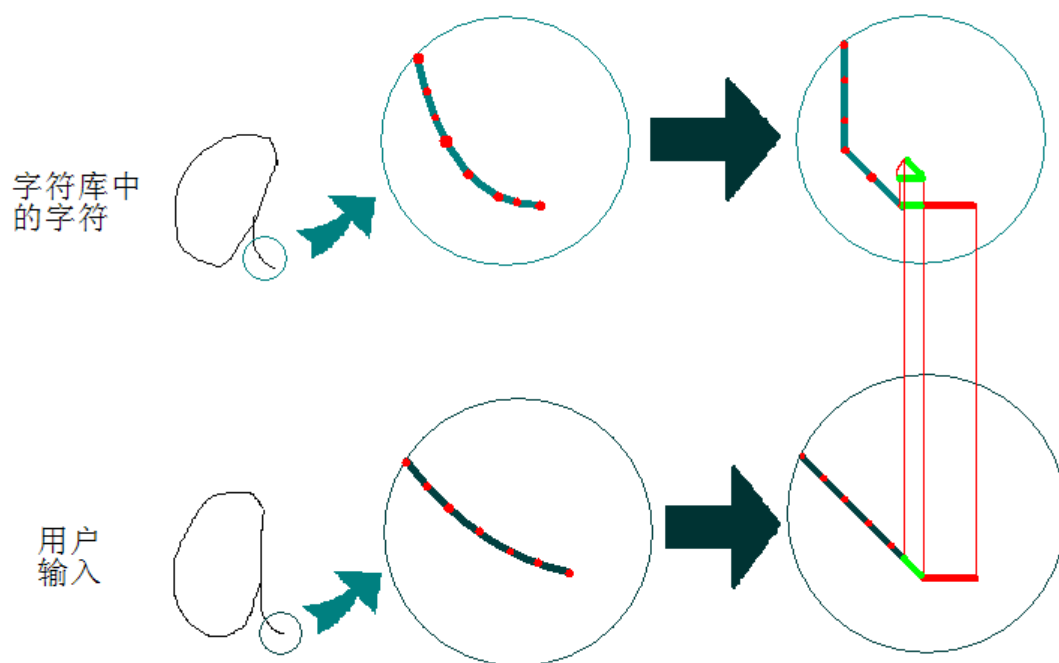
4.5 算法

4.5.1 联机模式

分析模块会根据处理模块输出的字符笔画数，定位到字符库中笔画数与字符笔画数相同的段，与段中所有的字符进行比较。

将用户输入的字符与字符库中的字符比较的算法如下：

方向的处理：将小笔画按顺序比较长度，取出较小的长度比（小笔画的真实长度比上全字总笔画的值） L ，计算出方向差 d 。再将原来较长的小笔画长度减去 L ，再作为单独的小笔画，按照前面的规则继续处理。最后将 $L*d$ 的值累加。



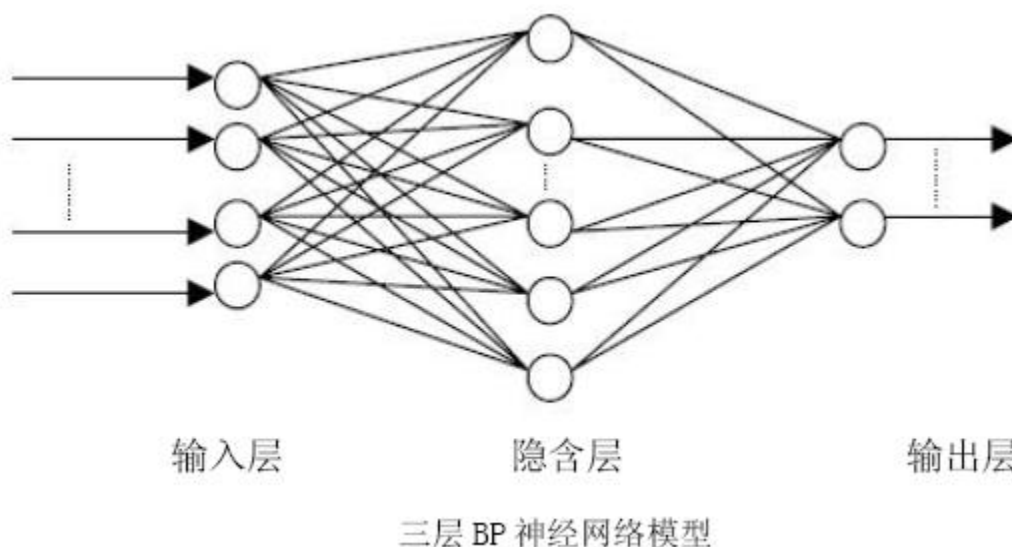
笔画对比示意图

重心的处理：先用每个小笔画的标准重心（重心真实坐标值与图像宽，高的比值） (x_i, y_i) ，计算出重心距离差，与相对应小笔画的长度比相乘，最后累加，得到权值 T 。

最后，将 $L*d+T$ 的值作为对应字符的得分，最后输出得分最高的 10 个字符。

4.5.2 脱机识别模式

本系统使用改进的三层 BP 神经网络对输入项进行分析。BP 网络是采用误差反向传播算法对网络权值进行训练的多层前向网络。一个 BP 网络设计需要计算网络加权输入矢量以及网络输出和误差矢量然后求得误差平方和。当所训练矢量的误差平方和小于误差目标训练则停止，否则在输出层计算误差变化，且采用反向传播学习规则来调整权值并重复此过程。



将 BP 算法用于具有非线性转移函数的三层前馈网可以任意精度逼近任何非线性函数。这一优势使多层前馈网络得到越来越广泛的应用，但也存在自身的限制与不足其主要表现在于他的训练过程的不确定。具体如下：

- 1 训练次数多使得学习效率低收敛速度慢
- 2 易形成局部极小而得不到全局最优

3 存在麻痹现象

针对以上存在的缺陷和不足在本系统中采用以下改进算法以提高神经网络分类器的性能

- (1) 带动量因子算法
- (2) 自适应学习速率
- (3) 误差函数的改进
- (4) 附加动量因子及动态自调整学习算法(又称综合法)

神经网络能通过对样本的学习训练,不断改变网络的连接权值及拓扑结构,使网络的输出不断地接近期望的输出,这一过程就是神经网络的训练过程。

训练的输入层有 128 层,来自于前面提到的 128 维字符特征向量。

训练输出层数是 5 层,输出项目编码列表如下图所示

优	良	中	差	不
00001	00010	00100	01000	10000

训练集使用的是从摄像头拍摄到的手写字符。



训练字符示例

本系统使用了 20 个隐藏节点,进行了 25000 次迭代。使

识别的正确率达到 90% 以上。

Exceed Technology