摘　要

字符识别是模式识别的一个分支，它能大大提高信息的采集录入速度，减轻人们的工作强度。随着计算机技术的飞速发展，字符识别技术多年来不断改进和完善，现在已经广泛应用于各个领域，使大量的文档资料能快速、方便、省时省力和及时地自动输入计算机，实现信息处理的电子化。到目前为止，尽管人们在字符识别的研究中已取得很多可喜成就，但还不能满足我们日常的需求。研究字符识别技术，提高字符识别率具有非常重要的意义。

本文针对字符识别的相关方法进行研究，主要工作包括：

1. 进行字符识别前，先要进行图像预处理，本文对图像预处理的一些方法进行了研究，像灰度化、二值化、噪声处理、特征值提取等。对灰度化和二值化的各种方法进行了性能比较，针对光照不均匀的车牌图像二值化效果不好的问题，给出了一种改进的局部阈值法的二值化方法，使用这种改进的方法能够有效的得到二值化图像。

2. 图像预处理阶段，重点分析了图像边缘检测方法，边缘检测对字符轮廓的提取、车牌定位、字符特征值提取等都非常重要。本文在深入研究传统 Canny 算子的基础上，对双阈值为极值点给予了证明，并给出了一种改进的 Canny 算子，通过对添加了椒盐噪声的图像进行大量实验，证明了本文给出的方法性能好于传统 Canny 方法。

3. 形状上下文(Shape Context)是一种形状描述方法，本文给出了一种改进的形状上下文方法用于字符粘连的复杂验证码识别。传统的逐个像素点特征提取和模板匹配的方法，只能对简单验证码进行识别，而字符粘连的复杂验证码还无有效的识别方法。针对字符粘连不

能有效的提取单个字符特征的问题，本文给出了改进的形状上下文方法进行特征提取，并采用字符整体识别的方法，实现了对复杂验证码的识别。

4. BP 神经网络作为一种应用最为广泛的神经网络学习算法，在车牌识别中得到了广泛应用。针对 BP 算法存在的收敛速度慢、易陷入局部极小等缺点，本文引入动量因子和自适应学习速率、改进激励函数以及使用 LM 算法对其进行改进。通过大量的车牌识别实验，本文将各种改进的 BP 算法性能进行了比较，并将本文方法与传统模板匹配方法进行性能比较，证明了本文给出的算法性能优于传统模版匹配方法。

关键词：字符识别，灰度化，二值化，边缘检测，形状上下文，BP 神经网络

Abstract

Character recognition, a branch of pattern recognition, could increase the speed of collecting and inputting information greatly, meanwhile the work intensity can be lightened. With the rapid development of computer technology, character recognition technology was also improved and wide applied in many fields, the result of which is electronization of information disposal, that is a great deal of document information can be input into computer quickly, conveniently and automatically in time. Up to now, though the achievements in the study of character recognition are fruitful, they are still not enough to meet our daily needs. Therefore, it is of high importance to study the character recognition technology and improve recognition rate.

Aiming at the methods related to character recognition, this paper did some research on the following aspects:

1. Before recognizing, the picture should be pretreated. This paper studied some pretreating methods, such as graying, binaryzation, noise processing, Eigenvalue Extraction. Through comparing performance of different methods in graying and binaryzation, aiming at the bad binaryzation effect problem of vehicle license plate caused by uneven light, this paper bring up an improved binaryzation called local threshold method, by which we can got better binary image.

2. In the image pretreatment, this paper introduced image edge-detection method first. Edge detection is very important to character outline extraction, license plate location, character eigenvalue extraction, etc. Based on the deliberate research of traditional Canny operator, this paper proved that the double threshold is extreme point, and put forward an improved Canny operator. Through carrying out a lot of experiment for picture increased salt and pepper noise, the method proposed in this paper can perform better than the traditional Canny.

3. Shape Context is a shape description method. This paper proposed a improved shape context, using for recognizing the adhered and complicated CAPTCHA. The traditional method of extracting pixel points one by one and template matching, can only recognize simple CAPTCHA, while there is no efficient method to recognize the adhered and complicated CAPTCHA. Aiming at the problem that single character feature can't be extracted efficiently because of conglutination, this paper proposed improved shape context to extract character feature, combined with character global

recognition, and realized the complicated CAPTCHA recognition.

4. As one of the most popular neural network algorithm, BP neural network was widely used in the recognition of vehicle license plate. For there are some defects existed in BP, such as low convergence rate, easy to trap into local minimum, this paper improved it through introducing momentum factor and self-adaptive learning rate, improving activation function, and using LM algorithm. Through a great deal of vehicle plate experiments, this paper compared the performance of all kinds of improved BP algorithm, and also compare the method proposed in this paper with the traditional template matching method, proved that the algorithm proposed in this paper is better than the traditional template matching method.

Key words: character recognition, graying, binaryzation, edge detection, Shape Context, BP neural network

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权江苏大学可以将本学位论文的全部内容或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保　密 □，在　　年解密后适用本授权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：贺强　　指导教师签名：

2010年 6月13日　　2010年 6月13日

独 创 性 申 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容以外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：贺强

2010 年 6 月 13 日

第一章 绪论

1.1 研究背景和意义

光学字符识别(Optical Character Recognition，OCR)技术[1][2][3]是模式识别的一个分支，通过扫描仪把印刷体或手写体文稿扫描成图像，然后识别成相应的计算机可直接处理的字符。它能大大提高信息的采集录入速度，减轻人们的工作强度。OCR技术伴随着计算机技术的飞速发展，多年来不断改进和完善，现在已经广泛应用于各个领域，使大量的文档资料能快速、方便、省时省力和及时地自动输入计算机，实现信息处理的电子化。到目前为止，尽管人们在字符识别的研究中已取得很多可喜成就，但还不能满足我们日常的需求。

研究字符识别技术的一个直接而现实的意义[4]是解决文字信息的高速、有效、方便、自动地输入到计算机的问题。另一方面，研究光学字符识别技术，在社会生活的其它领域也有着重要意义，例如：企业生产自动化(生产过程中的自动包装、印刷质量管理)、智能交通管理系统(车牌识别)、信息采集中的证件识别(身份证识别、名片识别等)、网络安全(验证码识别)等方面。

1.2 国内外研究现状

1.2.1 字符识别的研究现状

光学字符识别概念的诞生可以追溯到1809年一项帮助盲人阅读的装置的发明专利。1929年，由德国的科学家Tausheck首先正式提出OCR的概念，并且申请了专利，美国科学家Handel也提出了对文字进行识别的方案。但真正的OCR系统，直到电子计算机诞生后才变成为现实。

第一个OCR软件是在1957年开发的ERA（Electric Reading Automation），它是基于窥视孔方法实现的，识别的速度是每秒120个英文字母。在此以后，世界范围内广泛地进行着 OCR 技术的研究和开发工作。现代典型的商品化 OCR 系统可分为三代：

1960年到1965年出现的商品化OCR产品属于第一代，NCR公司、Farrington

公司、IBM 公司分别研制出了自己的 OCR 软件。这一代产品的特点是被识别的字符的字体很少，是经过特殊设计或指定的，甚至于某些字符看上去都不太自然。

二十世纪 60 年代中期到 70 年代初期的 OCR 产品属于第二代。在这个时期，有重要意义的事件是建立了一些供字符识别研究的标准化字符数据库，使研究人员可以很容易地评估自己的进展。

从 20 世纪 70 年代中期至今属于第三代，主要解决的技术问题就是对于质量较差的文档及大字符集的识别，以及达到非常高的识别精度，例如汉字的识别，高精度的手写数字识别等。

我国在 OCR 技术方面的研究工作起步较晚，在 70 年代开始对数字、英文字母及符号的识别进行研究；70 年代末开始进行汉字识别的研究；到 1986 年汉字识别的研究进入一个实质性阶段，取得了较大的成果，不少研究单位相继推出了中文 OCR 产品。时至今日，对印刷体汉字的识别率最高可达到 99%以上，可识别宋体、黑体、楷体、仿宋体、繁体等多种字体，并且可以对多种字体、不同字号混合排版进行识别；对手写体汉字的识别率最高达到 70%以上。

尽管文字识别率较高，但对于印刷质量较低、图像倾斜、以及字符粘连断裂等干扰，识别正确率有明显下降。同时，目前的版面分析技术，对于版面设计较为简单的文本可以达到很高的切分水平，但对于版面设计复杂，或印刷质量不高，如图像倾斜或噪声点过多的文本的分析与切分实用化程度仍有待提高。

随着计算机技术的飞速发展，为了适应计算机高速信息处理，作为计算机智能接口重要组成部分之一的 OCR 技术已经受到越来越多的人的重视。

1.2.2 相关方法的研究现状

1. 边缘检测

边缘检测是图像处理领域中最基本的问题，它的解决对于进行高层次的特征提取、特征描述、目标识别和图像理解等有着重大的影响，因此，被广泛应用与模式识别、计算机视觉、图像分割等众多领域。近来，随着科学技术的发展，一些新的理论工具被运用到边缘检测中，如形态学、统计学方法、模糊理论、遗传算法、神经网络等等[5][6][7][8]。

Canny 算子[9]作为一种优化的边缘检测算子，具有比较好的信噪比和检测精

度，得到了广泛的应用。然而实际应用中，对于高噪声的模糊图像，Canny 算子在抑制噪声的同时往往错过一些低强度的边缘，而一些高强度噪声被检测为边缘。针对这些不足，国内外学者进行了很多研究，提出了许多的改进方法。文献[10]提出了一种基于模糊理论的边缘检测方法，将 Canny 算子和模糊推理相结合进行边缘检测，取得不错的效果；文献[11]提出一种基于二维经验模态分解(BEMD)的Canny算子边缘检测算法，通过BEMD将图像分解成多层本征模函数，利用 Canny 算子对各分量进行边缘检测，获得较好的检测性能；文献[12]提出了一种改进的 Canny 边缘检测算法，采用新的基于梯度方向的检测和连接方法取代了传统的双阈值法，充分利用边缘点和噪声点在梯度方向特性上的差异；文献[13]提出基于梯度幅度直方图和类内方差最小化自适应的确定高低阈值的方法，可针对不同的图像，实现双阈值的自适应提取，不需要人为设定任何参数,采用模糊控制技术提取边缘像素。

2. 形状上下文(Shape Context)

形状上下文方法[14][15]是一种以目标轮廓的有限点集合来表示物体特征的描述方法。该方法首先对目标图像进行轮廓提取操作(或者是边缘检测操作) ，然后选择轮廓或边缘上的一组离散点的集合来表示该目标的形状信息。形状上下作为一种丰富的形状特征描述方法，被广泛的应用于验证码识别，人脸识别，基于内容的图像检索技术，形状匹配等。

传统形状上下文主要针对简单形状的形状匹配，为了表示复杂形状的形状特征，国内外一些学者进行了大量研究，提出了一些改进方法。文献[16]选择字符笔画的一些转弯点作为样本像素点，极大的减少样本像素点数量，加快形状上下文匹配速度；文献[17] 改进边界提取算法，并融入了扩散滤波的预处理算法和数学归一化方法对形状上下文进行改进，能处理简单形状和人脸形状匹配；文献[18]选择细化后的字符进行形状上下文建模，减少了形状上下文的复杂度，实现对简单验证码识别；文献[19]提出了一种结合形状上下文分析的Laplace谱匹配算法，首先使用Laplace 矩阵的特征向量和特征值以及双随机矩阵的方法计算初始匹配概率，然后借助于概率松弛算法，将用形状上下文表示的局部相似性融入Laplace谱匹配算法以优化谱匹配的结果，获得了比较高的精度。

3. BP神经网络

BP 神经网络[20][21][22]是基于误差反向传播算法(Back-propagation)的多层前向神经网络，它是 D.E.Rumelhart 和 J.L.McCelland 及其研究小组在 1986 年研究并设计出来的，已经成为目前应用最为广泛的神经网络学习算法。

由于BP神经网络采用的是经典的BP算法，而BP算法是基于梯度的最速下降法，以误差平方为目标函数，所以不可避免的存在四个缺陷：网络训练易陷入局部极小值；学习过程收敛速度缓慢；网络的结构难以确定(包括隐层数及各隐层节点数的难以确定)；所设计网络泛化能力不能保证。近十几年来，许多专家学者对其性能做了大量的工作。文献[23]提出在激励函数中引入陡度因子；文献[24]提出分段函数作为激励函数；文献[25]提出在标准误差函数中加上一个惩罚项 $p(w)$，$p(w)=\frac{\lambda}{n}\sum_{i,j}|w_{ij}|^n$，从而能提高网络的容错能力；文献[31]通过研究K-L信息距离和神经网络泛化能力的关系，构造一个新的神经网络学习函数，取得了较好的效果；文献[26]通过引入求解大规模线性方程组的共轭梯度法，提出了一种新的基于LM的前馈网络学习算法，该算法不仅具有LM优化学习方法的快速收敛性，并且降低了LM法的计算复杂度，有较好的学习精度和推广预测能力；文献[27]提出了基于Block Hessian矩阵的二阶学习算法，能避免标准BP算法收敛速度慢、易陷入局部极小的缺点，并且能克服Newton法的计算复杂、需大量内存的缺点。

1.3 研究内容

本文工作的研究内容是以字符识别为研究载体，以字符识别的相关方法为研究对象，对图像处理、验证码识别、车牌识别进行大量实验，通过实验结果对本文研究的相关方法进行详细分析。具体的说，本文主要完成了如下几个方面的不同分析：

1. 图像预处理阶段，对灰度化和二值化进行了研究，对现有的各种方法进行了比较，给出了一种改进的局部阈值法的二值化方法，针对光照不均匀的车牌图像二值化效果不好的问题，使用这种改进的方法能够有效的得到二值化图像。

2. 针对传统 Canny 算子在抑制噪声和检测低强度边缘能力不足的问题，本文给出一种 LOG 算子和 Canny 算子相结合的边缘检测方法。首先采用 LOG 算

子对图像进行噪声过滤，然后使用改进的 Canny 算子实现边缘检测。本文还对 Canny 算子双阈值为极值点给予了证明。又通过对添加了椒盐噪声的图像进行大量实验，证明了本文给出的方法性能好于传统 Canny 方法。

3. 传统的逐个像素点特征提取和模板匹配的方法，只能对简单验证码进行识别，而字符粘连的复杂验证码还无有效的识别方法。本文给出一种形状上下文(Shape Context)的改进方法，采用不对图片进行切割，整体识别的方法，使用单像素跟踪算法获取字符的轮廓，有效的减少了像素点数目，降低了形状上下文描述的复杂度，采用半圆形式的对数极坐标建模，解决了两个字符粘连处字符建模时互相干扰的问题，准确的描述出字符的特征，实现了对字符粘连的复杂验证码的识别。

4. 针对 BP 算法存在的收敛速度慢、易陷入局部极小等缺点，本文从引入动量因子和自适应学习速率、改进激励函数以及使用 LM 算法对其进行改进。通过大量的车牌识别实验，本文将各种改进的 BP 算法性能进行了比较，并将本文方法与传统模板匹配方法进行性能比较，证明了本文给出的算法性能优于传统模版匹配方法。

1.4 论文的组织结构

论文共分六章，主要内容概要如下：

第一章：介绍了本文的研究背景和意义及国内外的研究现状，说明本文的研究内容和论文的组织结构。

第二章：详细介绍了字符识别的相关理论，对常见的字符识别进行了描述。

第三章：字符识别的预处理过程，对灰度化的各种方法进行了比较，对二值化方法也进行了研究，重点对边缘检测方法进行了研究和实验分析，给出了一种改进的边缘检测算法。

第四章：形状上下文方法和验证码识别理论，给出了一种改进的形状上下文方法，通过对复杂验证码识别进行实验分析。

第五章：对BP算法的学习过程进行了分析，给出了改进的BP算法，通过车牌识别实验，分析了改进的BP算法性能。

第六章：对本文的研究工作进行归纳和总结，探讨进一步的研究方向。

第二章 字符识别方法

模式识别(Pattern Recognition)又常称作模式分类，是指对表征事物或现象的各种形式的(数值的、文字的和逻辑关系的)信息进行处理和分析，以对事物或现象进行描述、辨认、分类和解释的过程，是信息科学和人工智能的重要组成部分。

字符识别是模式识别的一个分支，一般分为数据获取、预处理、特征抽取、判别、后处理等模块。其中预处理是字符识别中特别重要的一环，它把原始的图像转换成识别器所能接受的二进制形式。要识别字符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、冗余信息，得到规范化的点阵，为识别做好准备。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。字符识别在中文信息处理、办公室自动化、机器翻译、人工智能等高技术领域，都有着重要的实用价值和理论意义。

2.1 光学字符识别系统

一个完整的 OCR 系统，如图 2.1，包括图像获取、预处理、特征提取、识别分类、后处理、识别结果。

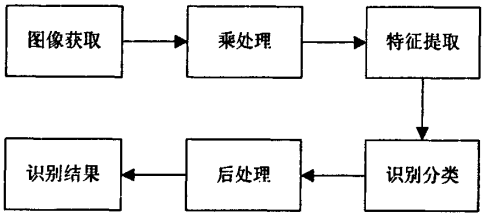


图 2.1 OCR 识别的一般步骤

2.1.1 图像获取

通常采用 CCD(Charge Coupled Device)摄像机或光学扫描仪等获取图像，其主要参数是图像分辨率，分辨率包括空间分辨率和灰度分辨率，前者反映了像素描述在空间上的精细程度，而后者则反映了像素描述在灰度(色彩)空间的明暗程度。

2.1.2 预处理

预处理效果的好坏会直接影响到整个 OCR 系统的性能。一般而言，预处理的目的是为了滤除噪声增强有用信息、对退化的信息进行复原。预处理方法一般有灰度化、二值化、图像去噪、字符细化、字符规范化等等。

1. 灰度化

手写体数字图像主要是通过扫描仪或摄像机等工具对纸张上的数字进行数据采集在计算机中成为一幅图像。在图像预处理中，人们只关心笔划， 而不关心其色彩。先要对读入的图像进行灰度化处理。由于 256 色的位图的调色板内容比较复杂，使得图像处理的许多算法都没有办法运用，因此首先要对它进行灰度处理。所谓灰度图像就是图像的每一个象素的 R、G、B 分量的值都是相等的。彩色图像的每个像素的 R、G、B 值是不同的，所以显示红绿蓝等各种颜色。灰度图像没有这些颜色差异，只是亮度不同。灰度值取值范围是 0—255，取值为 0 时是黑色，取值为 255 时是白色，所以灰度值大的像素点比较亮，反之则比较暗。图像灰度化有各种不同的算法，常用的一种就是给每个像素的 R、G、B 值各自一个加权系数，然后求和，同时对调色板表项进行相应的处理。

2. 二值化

在对图片进行了灰度化处理后，图像中的每个像素只有一个灰度值，它的大小决定了像素的亮暗程度。为了以后更方便的对图像进行处理，先进行二值化处理。

图像的二值化处理就是将图像上点的灰度值置为 0 或 255，也就是将整个图像呈现出明显的黑白效果。即将 256 个亮度等级的灰度图像通过选取适当的阈值而获得可以反映图像整体和局部特征的二值化图像。

图像二值化的关键在于阈值的选择。图像的二值化有很多成熟的算法， 有整体阈值二值化法、局部阈值二值化法、动态阈值二值化法等。

整体阈值化方法实现简单，对于具有明显双峰直方图的图像效果明显，而且速度也是最快的。但是给定整体阈值法有很多缺点，不能根据每个字符来选择最佳阈值，整体阈值一旦给定就不能更改了。并且对于低对比度和光照不均匀的图像效果不佳，抗噪声能力差，因而应用范围受到极大限制，如分水岭算法是一种经典的整体阈值方法，它反映了整个图像灰度分布情况，但它单一阈值的特性决定了它的抗噪声能力较差。

还有一种由灰度值直方图确定的整体阈值法。这种方法根据图像和背景的灰度值自动确定整体阈值。灰度级直方图给出了一幅图灰度值的概貌描述。数字图像的直方图一般有两个峰值，一个峰值对应数字笔划部分，另一个峰值对应背景部分，阈值取在两个峰值的波谷处，波谷越深陡，二值化效果越好。

局部阈值法能处理较为复杂的情况，但往往忽略了图像的边缘特征，容易出现伪影现象。如经典的局部阈值化 Bernsen 算法，其阈值由考察点邻域的灰度确定，算法中不存在预定阈值，适应性比整体阈值法广，但是当窗口的宽度较小时，很容易出现伪影现象和目标的丢失；而当窗口宽度增大时，算法的速度将受到很大影响。

动态阈值法充分考虑了像元的邻域特征，能够根据图像的不同背景情况自适应地改变阈值，可较精确地提取出二值图像，但它过渡地夸大了像元的邻域灰度的变化，会把不均匀灰度分布的背景分割到目标中去，带来许多不应出现的假目标。

在数字图像处理中，二值图像占有非常重要的地位，特别是在实用的图像处理中，以二值图像处理实现而构成的系统是很多的，要进行二值图像的处理与分析，首先要把灰度图像二值化，得到二值化图像，这样有利于在对图像做进一步处理时，图像的集合性质只与像素值为 0 或 255 的点的位置有关，不再涉及像素的多级值，使处理变得简单，而且数据的处理和压缩量小。为了得到理想的二值图像，一般采用封闭、连通的边界定义不交叠的区域。所有灰度大于或等于阀值的像素被判定为属于特定物体，其灰度值用 255 表示，否则这些像素点被排除在物体区域以外，灰度值为 0，表示背景或者例外的物体区域。如果某特定物体在内部有均匀一致的灰度值，并且其处在一个具有其他等级灰度值的均匀背景下，使用阈值法就可以得到比较好的分割效果。如果物体同背景的差别表现不在灰度值上(比如纹理不同)，可以将这个差别特征转换为灰度的差别，然后利用阈值选取技术来分割该图像。动态调节阈值实现图像的二值化可动态观察其分割图像的具体结果。

3. 图像噪声

(1) 图像噪声产生的途径

一幅图像在实际应用中可能存在各种各样的噪声，这些噪声可能在传输中产

生，也可能在量化等处理中产生。根据噪声和信号的关系可将其分为三种形式[28]：

假设 $f(x,y)$ 表示给定原始图像，$g(x,y)$ 表示图像信号，$n(x,y)$ 表示噪声。

① 加性噪声，此类噪声与输入图像信号无关，含有噪声的图像可表示为 $f(x,y)=g(x,y)+n(x,y)$，信道噪声及光导摄像管的摄像机扫描图像时产生的噪声就属这类噪声。

② 乘性噪声，此类噪声与图像信号有关，含有噪声的图像可表示为 $f(x,y)=g(x,y)+n(x,y)\ g(x,y)$，电视图像中的相干噪声，胶片中的颗粒噪声就属于此类噪声。

③ 量化噪声，此类噪声与输入图像信号无关，是量化过程中存在的量化误差，在接收端产生。

(2) 去除图像噪声的几种方法

① 均值滤波器

采用邻域平均法的均值滤波器非常适用于去除通过扫描得到的图像中的噪声。邻域平均法有力地抑制了噪声，同时也由于平均而引起了模糊现象，模糊程度与领域半径成正比。

几何均值滤波器所达到的平滑度可以与算术均值滤波器相比，但在滤波过程中会丢失更多的图像细节。

谐波均值滤波器对盐噪声效果更好，但是不适用于胡椒噪声。它善于处理像高斯噪声那样的噪声。

逆谐波均值滤波器更适合于处理脉冲噪声，但它有个缺点，就是必须要知道噪声是暗噪声还是亮噪声，以便于选择合适的滤波器阶数符号，如果阶数的符号选择错了可能会引起相当严重的后果。

这个均值模板是一个低通滤波器，从上到下，从左到右是一个卷积过程，整个取均值的过程是一个低通滤波过程。

用函数方法来描述均值法的效果原理：设有一幅数字噪声图像为

$$g(x,y)=f(x,y)+h(x,y) \tag{2.1}$$

经均值滤波处理后的平滑图像为

$$\widehat{g}(x,y)=\frac{1}{M}\sum_{(i,j)}g(i,j)=\frac{1}{T}\sum_{(i,j)}f(i,j)+\frac{1}{T}\sum_{(i,j)}h(i,j) \qquad (2.2)$$

在式(2.1)和(2.2)中，$f(x,y)$是原始图像，$h(x,y)$是噪声，s是点(x,y)邻域内的点集，T是点集s中的总点数。根据图像是由许多灰度恒定的小块组成的假设，可以看出式(2.2)的第一项非常接近于原始图像；而第二项代表平滑后图像中的噪声，它的均值仍为零，方差为

$$D\left\{\frac{1}{T}\sum_{(i,j)}h(i,j)\right\}=\frac{1}{T^2}\sum_{(i,j)}D\{h(i,j)\}=\frac{1}{T}\sigma_n^2 \qquad (2.3)$$

可见，图像经平滑处理后可使噪声方差减小T倍。因图像细节信息主要分布在高频区域，因此均值滤波的过程会导致图像变模糊。如果模板过大，则这种模糊会加剧。模板选择越小，去噪能力会下降。因此模板大小的选择实际上是去噪能力和保留图像细节的一种折中。

② 自适应维纳滤波器

在线性滤波理论中，维纳滤波器是所要解决的最小均方误差准则下的线性滤波问题。这种滤波方法是在已知信号与噪声的相关函数或功率谱的情况下，通过求解维纳－霍夫方程，对平稳随机信号进行最优预测和滤波的。

该方法的滤波效果比均值滤波器效果要好，对保留图像的边缘和其他高频部分很有用，不过计算量较大。维纳滤波器对具有白噪声的图像滤波效果最佳。

③ 中值滤波器

它是一种常用的非线性平滑滤波器，其基本原理是把数字图像或数字序列中一个点的值用该点的一个邻域中各点值的中间值来代换，其主要功能是让周围象素灰度值的差比较大的像素改为与周围的像素值接近的值，从而可以消除孤立的噪声点，所以中值滤波对于滤除图像的椒盐噪声非常有效。中值滤波器可以做到既去除噪声又能保护图像的边缘，从而获得较满意的复原效果，而且，在实际运算过程中不需要图像的统计特性，相对来说这也减少了很多的麻烦，但对一些细节特别是点、线、尖顶细节较多的图像不宜采用中值滤波的方法。

④ 形态学噪声滤除器

将开和闭操作结合起来可用来滤除噪声，首先对有噪声图像进行开操作，可选择结构元素矩阵比噪声的尺寸大，因而开操作的结果是将背景上的噪声去除。

然后是对前一步得到的图像进行闭操作，将图像上的噪声去掉。根据此方法的特点可以知道，此方法适用的图像类型是图像中的对象尺寸都比较大，且没有细小的细节，对这种类型的图像去噪的效果会比较好。

⑤ 小波去噪

这种方法保留了大部分包含信号的小波系数，因此可以较好地保持图像细节。小波分析进行图像去噪主要有 3 个步骤，第一：对图像信号进行小波分解。第二：对经过层次分解后的高频系数进行阈值量化。第三：利用二维小波重构图像信号。

4. 图像细化

在数字图像处理中，很重要的环节就是进行分支的细化工作。细化就是在不改变图像像素的拓扑连接性关系的前提下，连续地剥落图像的外层像素，使之最终成为单像素宽的图像骨架，细化后骨架的存储量要比原来的图像点阵少得多，降低了图像处理的工作量。它是在图像目标形状分析、信息压缩、特征提取与描述的模式识别等应用中经常运用的基本技术。几乎所有的光学字符识别都是基于细化算法的。因此，细化算法的好坏很大程度上决定了 OCR 系统的好坏。一个好的细化算法可以减少细化造成的形变，找到能反映字符真实形状的特征点，使系统有较高的识别率；相反，一个不好的细化算法会产生伪特征点，给字符分类带来困难，甚至导致误识或拒识。

经典的细化算法有：Hilditch 算法[29]、Rosen 算法[30]，索引表算法，细化根据是否使用迭代运算分为两类：一类是非迭代算法，一次性的产生骨架，常见的方法如基于距离变换和游程长度编码等。另一类是迭代算法，即重复删除满足一定条件的图像边缘像素，最终得到骨架。迭代方法又分成串行算法和并行算法，在串行算法中，是否删除像素在每次迭代的执行过程中顺序是固定的，它不仅取决于前次迭代的结果，也取决于本次迭代中已处理过的像素点分布情况，而在并行算法中，像素点删除与否与像素值在图像中的顺序无关，仅取决于前次迭代的结果。

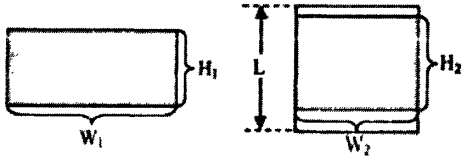
Hilditch、Pavlidis、Rosenfeld 细化算法：这类算法则是在程序中直接运算，根据运算结果来判定是否可以删除点的算法，差别在于不同算法的判定条件不同。

索引表细化算法：经过预处理后得到待细化的图像是 0、1 二值图像。像素值为 1 的是需要细化的部分，像素值为 0 的是背景区域。基于索引表的算法就是依据一定的判断依据，所做出的一张表，然后根据要细化的点的八个邻域的情况查询，若表中元素是 1，则删除该点，若是 0 则保留。因为一个像素的 8 个邻域共有 256 中可能情况，因此，索引表的大小一般为 256。

5. 变形矫形

在图 2.2 中，原字符图像 $f(x,y)$ 的宽度和高度用 W1 和 H1 表示，规整后字符 $g(x',y')$ 的宽度和高度表示为 W2 和 H2。规整化处理的字符局中放置在规整图像内。规整图像大小假设为 L×L，L 一般设为 32 或者 64 像素。原字符图像和规整字符的纵横比分别用 R1 和 R2 表示：

$$\begin{cases} R_1 = \dfrac{\min(W_1, H_1)}{\max(W_1, H_1)} \\ R_2 = \dfrac{\min(W_2, H_2)}{\max(W_2, H_2)} \end{cases} \tag{2.4}$$



a) 原始字符图像　　b) 规整后字符图像

图 2.2 字符图像大小和位置的规整化

常用的纵横比设置包括固定比例 $(R_2 = 1)$，保持比例 $(R_2 = R_1)$，平方根比例 $(R_2 = \sqrt{R_1})$，三次方根比例 $(R_2 = \sqrt[3]{R_1})$ 和 sin 比例 $(R_2 = \sqrt{\sin\frac{\pi R_1}{2}})$ 等。

一旦设置了纵横比关系，可以确定水平和竖直的缩放比例：

$$\begin{cases} \alpha = \dfrac{W_2}{W_1} \\ \beta = \dfrac{H_2}{H_1} \end{cases} \tag{2.5}$$

按照缩放比例，可以由前向映射(forward mapping)或者后向映射(backward mapping)实现坐标变换。前向映射可表示为：

$$\begin{cases} x' = x'(x,y) \\ y' = y'(x,y) \end{cases} \tag{2.6}$$

为了实现简便，上式常用一维坐标近似：

$$\begin{cases} x' = x'(x) \\ y' = y'(y) \end{cases} \tag{2.7}$$

式(2.7)表示的变换称为一维规整方法。最简单的是线性规整法(linearnormalization，LN)：

$$\begin{cases} x' = \alpha x \\ y' = \beta y \end{cases} \tag{2.8}$$

LN 没有用考虑像素点的局部分布，而基于线密度均衡的非线性规整法(nonlinearnormaliztion，NLN)则考虑到这方面的因素：

$$\begin{cases} x' = W_2 \sum_{u=0}^{x} h_x(u) \\ y' = H_2 \sum_{v=0}^{y} h_y(v) \end{cases} \tag{2.9}$$

式中 $h_x(x)$ 和 $h_y(y)$，分别为水平和竖直方向上的归一化线密度投影。进一步的，$h_x(x)$ 和 $h_y(y)$ 分别表示为：

$$\begin{cases} h_x(x) = \dfrac{\sum_y d_x(x,y)}{\sum_x \sum_y d_x(x,y)} \\ h_y(x) = \dfrac{\sum_x d_y(x,y)}{\sum_y \sum_x d_y(x,y)} \end{cases} \tag{2.10}$$

式中的 $d_x(x,y)$ 和 $d_y(x,y)$ 分别为水平和竖直方向上的局部线密度。文献中，有大量计算 $d_x(x,y)$ 和 $d_y(x,y)$ 的方法。

在 NLN 之后，发展了大量其它一维和伪二维规整方法。常用的一维规整方法包括一维矩规整法[31](moment normalization，MN)、一维双矩规整法[32](bi-moment normalization，BMN)、一维中心边界调整规整法[33](centroid boundary alignment，CBA)等。最近提出的线密度投影插值法是一种比较有效的伪二维规整方法[34]。以类似 LDPI 的策略，可以推广 MN、BMN、CBA 到二维情形，分别得到了 P2DMN、P2DBMN、P2DCBA。

2.1.3 特征抽取

特征提取一般分为结构特征和统计特征。

1. 结构特征

基于模式的结构特征用于模式识别众多领域。在汉字识别研究中，结构模式识别方法是人们最初用来进行手写汉字识别研究的方法.一般需要先抽取笔段或基本笔画作为基元，由这些基元再构成部件(子模式)，由部件的组合来描述汉字(模式)，最后再利用形式语种及自动机理论进行文法推断，即识别。然而，这种方法效果也不尽如人意，这是因为从汉字图像中抽取笔画等基元比较困难。通常，为了抽取笔画需要将原始点阵图像进行细化处理，但是细化算法不仅速度慢，而且容易产生伪笔画段，如将一个四叉点变成了二个三叉点，给准确抽取基元造成了困难。为了解决这个问题，有些学者试图不经过细化直接从汉字点阵图像中抽取笔画等基元，但效果仍不尽如人意。因此，有些研究人员放弃了抽取笔画或笔段作为基元然后进行文法推断的思路，采用汉字轮廓结构信息作为特征，这一方案的识别结果优于基于基元抽取的方法，但识别方法需要进行松弛迭代匹配，耗时严重，而且对于笔画较模糊的汉字图像，抽取内轮廓会遇到极大困难，外轮廓的抽取也不太稳定。也有些学者采用抽取汉字图像中关键特征点来描述汉字，字符的关键特征点包括端点、折点、交点、歧点、背景特征点、局部曲率最大点等，但是特征点的抽取易受噪声点、笔画的粘连与断裂等影响。

2. 统计特征

依据特征抽取区域的不同，统计特征分为全局统计和局部统计特征两大类。

全局统计特征是将整个汉字点阵作为研究对象，从整体上抽取特征。主要包括变换特征：对汉字图像进行各种变换，利用变换系数作为特征，常用的变换有Fourier 变换、Hadamard 变换、DCT 变换、Walsh 变换、Rapid 变换、K－L 变换等。不变矩(Moment)特征，Zemike 矩；笔画穿透数目特征；全局笔划方向特征(这种特征反映个汉字点阵中笔画的复杂度、方向及连接关系)；背景特征(汉字图像的空白部分和周围笔画的关系也含有一定的结构信息，提取背景点在各个方向的笔画密度特征，通常可选取位于汉字图像两对角线上的背景点)。

局部统计特征是将汉字点阵图像分割成不同区域或网格，在各个小区域内分别抽取统计特征，主要包括：局部笔画方向特征；细胞特征；方向线素特征；

Gabor 特征；四角特征等。根据抽取特征的不同，可以选用不同的匹配方法，常用的有统计匹配、模板匹配、相关匹配、树分类器等。常用的距离度量有欧氏距离、城市街区距离、马氏距离等。

与结构法相比，统计法具有良好的抗噪声、抗干扰的性能，其鲁棒性主要体现在统计特征的抽取和模式匹配方法上。

可见，统计与结构方法各有优缺点。统计方法具有良好的鲁棒性，较好的抗干扰的能力，一般按一定的距离度量匹配准则，采用多维特征值累加的办法，把局部噪声和微小畸变淹没在最后的累加和里，但是，可以用来区分“敏感部位”的差异也随失，因此区分相似字的能力较差。结构方法对结构特征较敏感，区分相似字的能力较强，但是结构特征难以抽取，不稳定。因此，可以考虑将两种方法结合使用。

2.2 几种字符识别介绍

2.2.1 手写体字符识别

手写体识别在学科上属于人工智能与模式识别的范畴。在过去的四十年中，人们想出了很多办法获取手写字符的关键特征。主要分两大类：全局分析和结构分析。对前者，我们可以使用模板匹配、象素密度、矩、特征点、数学变换等技术，这类特征常常和统计分类方法一起使用。对后者，多半需要从字符的轮廓或骨架上提取字符形状的基本特征，包括：圆圈、端点、凸弧、凹弧、笔划等。与这些结构特征配合使用的往往是句法的分类方法。多年的研究实践表明，对于自由手写体字符，几乎可以肯定：没有一种简单的方法能达到很高的正确识别率。因此，最近在这方面正加快向着更为成熟、综合的方向发展。一方面，研究工作者努力把新的知识运用到预处理，特征提取，分类当中，如：神经网络、数学形态学等方法。到目前为止，尽管人们在脱机手写体字符识别的研究中已取得很多可喜成就，但距实用要求还有一定距离。而在手写数字识别这个方向上，经过多年研究，研究工作者已经开始把它向各种实际应用推广，为手写数据的高速自动输入提供了一种解决方案。

2.2.2 验证码识别

验证码是指包含有一串数字或其它字符的一幅图片，图片里加上一些干扰象素。攻击者编写的攻击程序，难以识别验证码字符，顺利的完成自动注册，登录。而用户可以识别填写，所以这就实现了阻挡攻击的作用。

在国外，称为 CAPTCHA，是 Completely Automated Public Turing Test to Tell Computers and Humans Apart(全自动区分计算机和人类的图灵测试)的简称，已由卡内基梅隆大学注册成商标。CAPTCHA 的目的是区分计算机和人类的一种程序算法，这种程序必须能生成并评价人类能很容易通过但计算机却通不过的测试。这个要求本身就是悖论，因为这意味着一个 CAPTCHA 必须能生成一个它自己不能通过的侧试。

2.2.3 车牌识别

汽车牌照识别(VLPR，Vechicle License Plate Recognition)是智能交通系统(ITS，Intelligent Transportation Systems)中的关键技术之一，已广泛应用于各级公路和城市道路交通管理(如道路收费系统、交通管理系统)，具有巨大的经济价值和现实意义。目前汽车牌照识别技术主要有：图像处理技术、条形码识别技术、IC 卡识别技术、射频识别技术、模式识别技术. 其中 IC 卡识别技术、条形码识别技术和无线射频技术都要求预先在车身上印刷条形码或在车内安装标识卡，因此，这 3 种技术在全国范围内推广有一定的困难。而汽车牌照识别系统是基于图像处理技术的识别系统，具有更广阔的应用前景。

2.4 本章小结

本章介绍了字符识别的相关概念。首先介绍了字符识别系统，详细介绍了字符识别系统个个处理阶段相关方法，然后对常见的几种字符识别进行了介绍。

第三章 图像的预处理

图像的预处理一般包括[35]灰度化、二值化、图像增强、图像校正、噪声去除、图像滤波、边缘检测等。不同的识别方法，对预处理的要求有所差别。如结构识别方法，对字符的规范化预处理可以从简，甚至不要。而有的识别方法对细化预处理要求很高，有的则不需要细化等。

图像的预处理在字符识别研究中有着重要的作用，它可以把原始图像转换成识别器所能接受的形式(二值化图像)，消除一些与类别无关的因素(尺寸和位置的标准化)。由于一般都在预处理后的图像上进行分割、提取特征等，因此如果预处理的结果不理想，往往会给后面的分割和识别环节带来无法纠正的错误。

3.1 RGB 图像的灰度化

3.1.1 灰度化的概念

位图图像一般分为单色图像、灰度图像和彩色图像。单色图像只有黑色和白色两种颜色，整个图像由单纯的黑色点和白色点组成。彩色图像的像素点是由R(红色)、G(绿色)、B(蓝色)三原色混合而成的，不同含量的 R、G、B 组成不同的颜色，每一个记录单个像素的位数据单元可表示任意一种颜色。而灰度图是指只含亮度信息，不含色彩信息的图像。灰度图像与单色图像的区别是加上颜色深度的概念，单纯的看，灰度图也是黑白的，就像黑白电视显示的图像一样，但是点与点之间黑的程度是不一样的，这就是深度。如果称不同深度的颜色为一色的话，灰度图像就不止只有黑色和白色两种颜色，一般使用的灰度图为 256 级灰度图。

灰度化处理[36]是指把含有亮度和色彩的彩色图像变换成灰度图像的过程。在 RGB 模型中，如果 R=G=B，则颜色(R、G、B)表示一种黑白颜色，其中 R=G=B 的值叫做灰度值。灰度化处理在许多图像处理中是很重要的一步，其结果就是后续处理的基础。把 24 位彩色图像转化为灰度图像有利于减小计算量，减少程序处理时间，而且彩色验证码图像会刻意的用各种颜色干扰用户对字符的识别，把图像转化为灰度图后同样可以获得足够的图像细节，并且还能减少不必要的干

扰，降低处理的复杂度。所以，寻求一种正确有效的灰度化处理方法尤其重要。

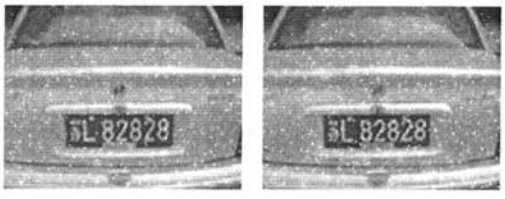
3.1.2 灰度化的处理方法

灰度化处理的方法[37]主要有三种：最大值法，平均值法和加权平均值法。

(1)最大值法：使 R、G、B 的值等于三者中最大的一个作为该点的灰度，即：R=G=B=max(R，G，B)，这种方法会形成亮度很高的图像。

(2)平均值法：用 R、G、B 的值求出平均值作为该点的灰度，即：R=G=B=(R+G+B)/3，这种方法会形成较为柔和的灰度图像。

(3)加权平均值法：根据某种指标给 R、G、B 赋予不同的权值，取 R、G、B 的加权平均值作为该点的灰度，即：R=G=B=(W_R×R+W_G×G+W_B×B)/3，式中的 W_R、W_G 和 W_B 分别为 R、G、B 的权值。W_R、W_G 和 W_B 取不同的值，加权平均值法就会形成不同的灰度图像。研究表明人眼对绿色的敏感度最高，对红色的敏感度次之，对蓝色的敏感度最低，所以使 W_G> W_R >W_B 将得到较合理的灰度图像。实验和理论推导证明，当 W_G=0.587，W_R =0.299，W_B=0.114 时能得到最合理的灰度图像。图 3.1 为使用加权平均值法灰度化。



D）原图　　E）加权平均值法

图 3.1 加权平均值法灰度化

在本文中我们选用适合人眼感受的加权平均值法来灰度化彩色图像。我们使用下面的式(3.1)把图像化为 256 级的灰度图像，

$$Gray = (\mathrm{Re}\,d \times 299 + Green \times 587 + Blue \times 114)/1000 \qquad (3.1)$$

式(3.1)中 Gray 是灰度值，Red、Green、Blue 分别是像素的红、绿、蓝色分量。

3.2 图像的二值化

3.2.1 二值化的概念

通过闭值把灰度图像处理成二值图像的过程，称为对数字图像的二值化[38]。二值化处理后的图像呈现出明显的黑白效果，图像上的点要么是黑，要么是白。验证码图像二值化的目的就是使字符区域呈现为黑色，背景区域呈现为白色。

在数字图像处理中，二值图像占有非常重要的地位，特别是在实用的图像处理中，以二值图像处理实现而构成的系统是很多的，要进行二值图像的处理与分析，首先要把灰度图像二值化，得到二值化图像，这样有利于对图像做进一步处理时，图像的集合性质只与像素值为 0 或 255 的点的位置有关，不再涉及像素的多级值，使处理变得简单，而且数据的处理量和压缩量小了很多。为了得到理想的二值图像，一般采用封闭、连通的边界定义不交叠的区域。所有灰度小于或等于阈值的像素被判定为属于特定物体，其灰度值为 0；否则这些像素点被排除在物体区域以外，灰度值为 255，表示背景或者例外的物体区域。设原始图像为 $f(x,y)$，按照一定的准则在 $f(x,y)$ 中找到特征值 T，将图像分割为两个部分，分割后的图像为：

$$f(x,y)=\begin{cases}0 & f(x,y)\le T\\ 255 & f(x,y)>T\end{cases} \tag{3.2}$$

其中 T 称为二值化运算的阈值或门限值。当图像中某点 (x,y) 的灰度值 $f(x,y)\le T$ 时，$f(x,y)=0$，表示将此点转换为前景；当图像中某点 (x,y) 的灰度值 $f(x,y)>T$ 时，$f(x,y)=255$，表示将此点转换为背景。

对图像进行二值化处理的关键是阈值的选择与确定。不同的阈值设定方法对一幅图像进行二值化处理后产生不同的处理结果。各种图像上目标物性质的变化及灰度变化的多样性，使得固定的二值化方法难以奏效，处理效果不太理想，影响了对各种图像的后续处理。阈值设置过大会保留过多噪声阈值设置过小会降低字符区域的分辨率，使非噪声信号被视为噪声而滤掉。实际应用及研究表明，普适的阈值选取方法应满足不受图像质量及图像类型的限制、能保留足够的图像特征信息、可实现对不同图像阈值的自动化选择、时间开销尽可能少等几方面的要求。

3.2.2 常用的二值化方法

本节将介绍几种常用的二值化方法[39][40]并对两种不同的验证码进行二值化，观察其效果。

1. p-分位数法

1962 年 Doyle 提出的 p-分位数法[41]可以说是最古老的一种阈值选取方法。该方法使目标或背景的像素比例等于其先验概率来设定阈值。假如已知对象区域面积(像素点数)在全图中所占的比率 p，就可以据此在直方图中选取相应的分割阈值。若对象区域较暗，则从灰度值 0 开始累加象素点数，并计算它在全图中的比率，将此比率与 p 作比较，等到前者等于或超过 p 值时，这时的灰度值就是所需的分割阈值。

由于这种方法简单，容易实现，也可用穷举方法来逼近。即先根据估计设定一个可能比率对图像进行二值化。然后根据某个指标判别分割阈值是否合适，如字符识别时字体的宽度是否均匀，图形是否完整，噪声是否严重等，再根据二值化后的效果调整比率重新进行二值化，如此反复直至找到所需的阈值。

2. 最大类间方差法

由 Otsu 于 1978 年提出的最大类间方差法[42]以其计算简单、稳定有效，一直广为使用。这种方法是在最小二乘法原理基础上推导出来的，它的基本思想是将直方图在某一阈值处分割成两组，一组对应于背景干扰噪声部分，一组对应于前景字符部分，当被分成的两组的组内方差最小，组间方差最大时，确定阈值。

设给定图像具有 L 级灰度值，对 1<K<L 中的每个 K 将 1 到 L 分成两组，1 到 K 为组 1，K 到 L 为组 2。计算组 1 的像素数 $\omega_1(k)$，平均灰度 $M_1(k)$，方差 $\sigma_1^2(k)$，组 2 的像素数 $\omega_2(k)$，平均灰度 $M_2(k)$，方差 $\sigma_2^2(k)$。则：

组内方差：

$$\sigma_w^2(k)=\sigma_1^2(k)\omega_1(k)+\sigma_2^2(k)\omega_2(k) \tag{3.3}$$

组间方差：

$$\sigma_B^2(k)=\omega_1(k)\omega_2(k)[M_1(k)-M_2(k)] \tag{3.4}$$

使组间方差最大的 K 值即为所求的阈值。

算法描述如下：

(1)遍历图中所有点，得到灰度直方图 histogram[255]，最大像素值 max 和最

小像素 min；

(2)令 $t=\min$，$tempG=0$；

(3)令 min 到 t 为组 1，t 到 max 为组 2，求出组 1 的平均灰度值 mean1 及其像素占整幅图片的比例 w1 和组 2 的平均灰度值 mean2 及其像素占整幅图片的比例 w2；

(4)计算组间方差：$G=w1\times w2\times(mean1-mean2)^2$，若 $G>tempG$，则 $tempG=G$，$threshold=t$；

(5)$t=t+1$，循环 3-5 步，直到 $t=\max$；

(6)此时得到的 *threshold* 值即为最佳的阈值。

3. 熵方法

1948 年，香农提出了“信息熵”的概念，解决了对信息的量化度量问题。“信息熵”是信息论中用于度量信息量的一个概念。一个系统越是有序，信息熵就越低；反之，一个系统越是混乱，信息熵就越高。所以，信息熵也可以说是系统有序化程度的一个度量。

当将熵的概念用于图像分割时，基本是研究图像灰度直方图的熵测量，并由此自动找出最佳门限去分割图像。从不同的角度出发，会定义不同的熵测量以及选择最佳门限的方法。下面我们来介绍一种 KSW 熵方法[43]。

设图像有 L 级灰度 $Gray=\{0,1,...,L-1\}$，每个灰度级的概率分布为 $p_0,p_1,...,p_k$，由此可以导出两个灰度分布前景 A 和背景 B，其中 A 是 $\{0,1,...,T\}$ 的灰度分布，B 是 $\{T+1,T+2,...,L-1\}$ 的灰度分布，这两个分布分别为：

A：$p_0/p_T,p_1/p_T,...,p_T/p_T$；

B：$p_{T+1}/1-p_T,p_{T+2}/1-p_T,p_{L-1}/1-p_T$；

这两个分布对应的熵分别是：

$$H_A(T)=\sum_{t=0}^{T}\frac{p_t}{p_T}\ln\frac{p_t}{p_T} \tag{3.5}$$

$$H_B(T)=\sum_{t=T+1}^{L-1}\frac{p_t}{1-p_T}\ln\frac{p_t}{1-p_T} \tag{3.6}$$

整个图像的总体熵为：$H(T)=H_A(T)+H_B(T)$。使得$H(T)$达到最大值的那个T即为图像二值化分割的最佳阈值。

4. 迭代法

迭代法[44][45]是基于逼近的思想。从图像的直方图中可以看到，以波谷的任一点作为阈值点，都可以很好地将字符和背景分开，而以波谷这些点作为阈值点变换时，被阈值点分开的两部分像素点的平均灰度值应该是趋于稳定的。最佳阈值点的选取就是利用了这个特点作为是否为最佳阈值点的判断标准。算法如下：

(1)首先对图像每个像素点的灰度值遍历一遍，通过两两对比找出最小灰度值$gray_{\min}$和最大灰度值$gray_{\max}$；

(2)其次初始化阈值$T_0=(gray_{\min}+gray_{\max})/2$；

(3)然后以T_0的值作为图像像素灰度值的划分点，分别求出两侧的平均灰度值$meanvalue1$和$meanvalue2$，选取$T_1=(meanvalue1+meanvalue2)/2$作为新的划分点。

(4)通过前面的分析，可以看出经过多次迭代后，T_0和T_1的差会逐渐收敛，当这个差足够小时，就可以说明以T_1作为阈值点时，图像两边的平均灰度值已经趋于稳定，此时的T_1就是我们要找的最佳阈值点。

5. Niblack 方法

上面的四种二值化方法都属于全局阈值法的二值化方法，而 Niblack[46]方法是属于局部阈值法的二值化方法，其思想是根据局部平均值和局部标准差在图像中变动阈值。点(x,y)处的阈值这样来计算：

$$T(x,y)=m(x,y)+k\times s(x,y) \tag{3.7}$$

其中$m(x,y)$和$s(x,y)$依次是(x,y)的局部邻域的样本平均值和标准差。这个邻域的大小不能太小，那样将得不到足够的信息，以致算出来的阈值效果不好，容易保留太多噪声；但是也不能太大，那样一来计算量很大，而且这个阈值也不能体现局部的细节信息。通常，取 15×15 大小的邻域效果较好。而这个 K 值是

用来调整决定多大的字符目标边界被作为给定目标的一部分。一般取 K=-0.2。

Niblack 方法应用得非常广泛，效果在局部阈值法中也是比较好的。但是它也有不足，那就是会产生连续大块的非目标黑色区域，要经过后续处理去除。另外，由于 Niblack 方法对一个像素点做二值化时，需要考察其 15×15 邻域的像素值，因此，对于输入图像边缘宽为 7 的带状方框，由于其阈值计算没有足够的邻域用以分析，会存在边缘效应，边缘部分的二值化是不准确的。在本实验中，对边缘部分我们采取用平均灰度值作为阈值的方法来处理。

图 3.2 为上面几种二值化方法效果图。

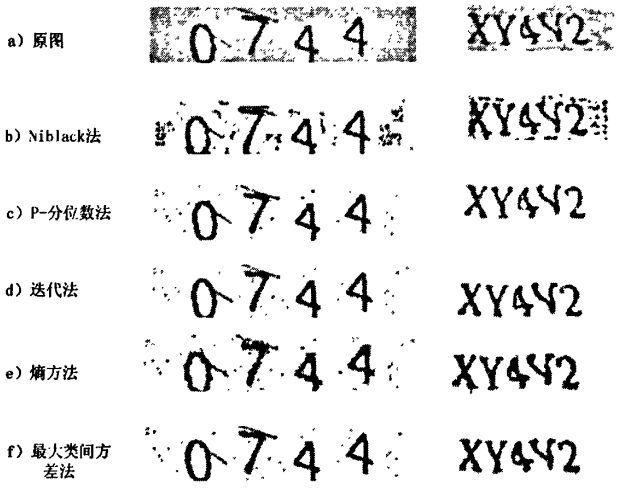


图 3.2 几种二值化方法比较

3.2.3 一种改进的 Bernsen 算法

目标和背景分离明显、灰度直方图呈明显双峰模式的图像，采用全局阈值化方法可以取得较好的效果，如 OTSU 算法和平均灰度阈值法等。由于在实际环境中光照不均匀、图像噪声干扰等原因，图像的灰度直方图分布不呈双峰性。局部阈值法常用于识别干扰比较严重，或者非均匀光照的图像，如 Bernsen 算法、Niblack 算法等。其中 Bernsen 算法凭借其优异的综合性能，在局部二值化算法效果明显，较适合解决光照不均的问题。当字符图像有大量的阴影时，单个字符的提取成为一个困难的问题，如何去除阴影是图像预处理的一个课题。

设 $f(x,y)$ 表示 (x,y) 处的像素灰度值，考虑以 (x,y) 为中心大小为 $(2w+1)^2$ 的

区域 S，首先计算 $f(x,y)$ 点的阈值 $T(x,y)$：

$$T(x,y)=\frac{\max\limits_{-w\le k,l\le w} f(x+k,y+l)+\min\limits_{-w\le k,l\le w} f(x+k,y+l)}{2} \tag{3.8}$$

其次，对 $f(x,y)$ 逐点二值化：

$$b(x,y)=\begin{cases}0, & if \quad f(x,y)<T(x,y)\\ 255, & else\end{cases} \tag{3.9}$$

然而，Bernsen 算法对噪声比较敏感，因此本文给出一种改进的 Bernsen 算法，设 $f(x,y)$ 表示 (x,y) 处的像素灰度值，考虑以 (x,y) 为中心大小为 $(2w+1)^2$ 的区域 S，$\hat{f}(x,y)$ 表示 (x,y) 处经过平滑滤波后的灰度值，σ 为平滑尺度 k 、l 为窗口内的位置参数，$b(x,y)$ 表示 (x,y) 处的二值化结果，改进的 Bernsen 算法可描述为：

(1)计算 $f(x,y)$ 点的阈值 $T_1(x,y)$

$$T_1(x,y)=\frac{\max\limits_{-w\le k,l\le w} f(x+k,y+l)+\min\limits_{-w\le k,l\le w} f(x+k,y+l)}{2} \tag{3.10}$$

(2)对 $f(x,y)$ 点在 $(2w+1)^2$ 窗口内进行高斯平滑滤波

$$\hat{f}(x,y)=\frac{1}{(2w+1)^2}\sum_{x,y<s} f(x,y)\times\exp\{-\frac{1}{2}[(\frac{x}{\sigma})^2+(\frac{y}{\sigma})^2]\} \tag{3.11}$$

(3)计算滤波后 $\hat{f}(x,y)$ 点的阈值 $T_2(x,y)$

$$T_2(x,y)=\frac{\max\limits_{-w\le k,l\le w} \hat{f}(x+k,y+l)+\min\limits_{-w\le k,l\le w} \hat{f}(x+k,y+l)}{2} \tag{3.12}$$

(4)取 $\alpha\in(0,1)$，对 $f(x,y)$ 逐点二值化

$$b(x,y)=\begin{cases}0, & if \quad f(x,y)<(1-\alpha)T_1(x,y)+\alpha T_2(x,y)\\ 255, & else\end{cases} \tag{3.13}$$

当 α 取 0 时，该算法即为原 Bersen 算法，当 α 取 1 时，该算法即为高斯平滑滤波 Bernsen 算法。

a)光照不均车牌　　b)传统方法　　c)本文方法

图 3.3 改进的 Bernsen 算法处理效果

3.3 图像边缘检测

边缘检测是图像处理领域中最基本的问题，也是经典的技术难题之一，它的解决对于进行高层次的特征提取、特征描述、目标识别和图像理解等有着重大的影响。因此，边缘检测在图像分割、模式识别、计算机视觉等众多方面都有着非常重要的地位。然而由于成像过程中的投影、混合、畸变和噪声等导致图像的模糊和变形，边缘往往难于检测，这使得人们一直致力于构造具有良好性质的边缘检测算子。边缘检测的研究有着久远的历史，其原因一方面是由于课题本身的重要性，另一方面也反映了这个课题的深度和难度。所以，边缘检测方面的研究具有非常重要的理论意义。

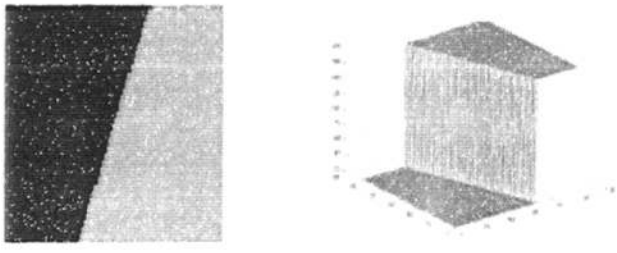
3.3.1 边缘检测概念

边缘是图像局部灰度变化不连续部分，是图像中灰度的急剧变化，主要存在于目标与目标、目标与背景、区域与区域(包括不同色彩)之间，是图像分割、纹理特征提取和形状特征提取等图像分析的重要基础。图像分析和理解的第一步常常是边缘检测，所以边缘检测技术的研究十分重要。

通常把边缘划分为阶跃边缘和线状边缘两种类型：斜坡边缘可以视为阶跃型边缘的特例[46]。对于阶跃性边缘，如图 3.3a)所示，在其正交切面上，阶跃边缘点周围的图像灰度值表现为一维阶跃函数，边缘点位于图像灰度的跳变点，其一阶方向导数在边缘处取极值，二阶方向导数在边缘处呈零交叉；而对于线状边缘，如图 3.3b)所示，在其正交切面上，边缘点周围图像灰度值呈屋脊状变化，边缘点位于局部极值的位置，其二阶方向导数在边缘处取极值。图 3.4 和图 3.5 分别是理想阶跃边缘和理想线状边缘的二维图和三维图。

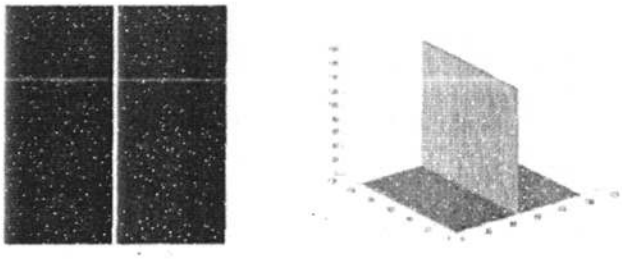
a）阶跃边缘　　b）线状边界

图 3.4 边缘种类



D）理想的阶跃边界　　E）三维图

图 3.5 理想阶跃边缘及其三维图



a）理想的线状边缘　　b）三维图

图 3.6 理想线状边缘及其三维图

边缘检测的定义有很多种，其中最常用的一种定义为：边缘检测是根据引起图像灰度变化的物理过程来描述图像中灰度变化的过程。而实际应用中，图像数据往往被噪声污染。因此，边缘检测方法要求既能检测到边缘的精确位置，又可以抑制无关细节和噪声。

3.3.2 基本实现方法

边缘是图像中灰度发生剧烈变化的地方。因为导数可以表示函数斜率的变化，因而，早期的边缘检测方法利用一阶导数的极大值或二阶导数的过零点来检测边缘点，由此衍生出一系列的不同形式的微分算子，如 Sobel 算子、Roberts

算子、Prewitt 算子和 Laplacian 算子等等。

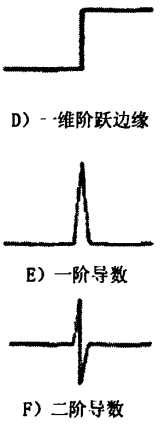


图 3.7 阶跃边缘的微分特性

微分算子的阶数是由偏导数的阶数确定的。一阶微分方法[46](Sobel 算子、Robert 算子等)和二阶微分方法[47](Laplacian 算子等)是图像边缘检测最基本的方法。图 3.7 分别是一维阶跃边缘的一阶和二阶导数示意图。由图可见，对于一维阶跃边缘，在一阶导数中为极大值，而在二阶导数中对应过零点。

1. 一阶微分方法

图像的梯度函数即函数灰度变化的速率，它在边缘处为局部极大值。通过梯度算子或一阶导数算子估计图像灰度变化的方向，增强图像中的灰度变化区域，然后对增强的区域进一步判断边缘。

对于连续函数 $I(x,y)$，它在点 (x,y) 的 x 方向，y 方向和 θ 方向的一阶方向导数为：

$$I_x(x,y)=\frac{\partial I(x,y)}{\partial x} \tag{3.14}$$

$$I_y(x,y)=\frac{\partial I(x,y)}{\partial y} \tag{3.15}$$

$$I_\theta(x,y)=\frac{\partial I(x,y)}{\partial x}\cos\theta+\frac{\partial I(x,y)}{\partial y}\sin\theta \tag{3.16}$$

它在点 (x,y) 处的梯度为一个矢量，定义为：

$$\nabla I(x,y)=[G_x G_y]^T[\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}]^T \tag{3.17}$$

梯度幅值为：

$$|\overrightarrow{grad}|=\sqrt{(\frac{\partial}{\partial x})^2(\frac{\partial}{\partial y})^2}\ 或|\overrightarrow{grad}|=|\frac{\partial}{\partial x}|+|\frac{\partial}{\partial y}| \tag{3.18}$$

梯度方向为垂直于边缘的方向：

$$\varphi=arctg(\frac{\partial}{\partial y}/\frac{\partial}{\partial x}) \tag{3.19}$$

根据以上的理论，人们提出了很多算法，经典的有：Robert 边缘检测算子、Sobel 边缘检测算子、Prewitt 边缘检测算子等等。所有基于梯度的边缘算子之间的根本区别在于算子应用的方向，以及在这些方向上逼近图像一维导数的方式和将这些近似值合成为梯度幅值的方式不同。在数字图像中常常以图像的一阶差分运算代替图像的一阶微分运算。

$$\Delta_x I(i,j)=I(i,j)-I(i-1,j) \tag{3.20}$$

$$\Delta_y I(i,j)=I(i,j)-I(i,j-1) \tag{3.21}$$

2. 二阶微分方法

在阶跃边缘处灰度的变化速率最大，如图 3.7 所示，边缘点对应的二阶导数为过零点。它在点(x,y)的 x 方向，y 方向的二阶导数为：

$$I_{xx}(x,y)=\frac{\partial^2 I(x,y)}{\partial x^2} \tag{3.22}$$

$$I_{yy}(x,y)=\frac{\partial^2 I(x,y)}{\partial y^2} \tag{3.23}$$

设$\vec{n}$是梯度方向，则图像$I(x,y)$在点(x,y)处沿着梯度方向的二阶导数为：

$$\frac{\partial^2 I(x,y)}{\partial \vec{n}^2}=\frac{\partial^2 I(x,y)}{\partial x^2}\cos^2(\theta)+2\frac{\partial^2 I(x,y)}{\partial x\partial y}\sin(\theta)\cos(\theta)+\frac{\partial^2 I(x,y)}{\partial y^2}\sin^2(\theta) \tag{3.24}$$

由于二阶方向导数算子不具备线性和旋转不变性，常采用具有线性和旋转对称性的 Laplacian 算子：

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.25}$$

Laplacian 算子是一个标量，属于各向同性运算，具有旋转不变性，对灰度突变敏感。在数字图像中，仍然可以用差分来近似微分运算，则 $I(x,y)$ 的 Laplacian 算子为：

$$\begin{aligned} &\nabla^2 I(i,j) = \Delta_x^2 I(i,j) + \Delta_y^2 I(i,j) \\ &= I(i+1,j) + I(i-1,j) + I(i,j-1) + I(i,j+1) - 4I(i,j) \end{aligned} \tag{3.26}$$

3.3.3 双阈值为极值点证明

1. Canny 的三个准则

(1)信噪比准则。

信噪比的数学表达式为：

$$SNR = \frac{|\int_{-w}^{+w} G(-x) f(x) dx|}{\partial \sqrt{\int_{-w}^{+w} f^2(x) dx}} \tag{3.27}$$

其中，$f(x)$ 是边界为 $[-w,+w]$ 的滤波器的脉冲响应；$G(-x)$ 代表边缘函数；σ 是高斯噪声的均方差，信噪比越大，提取的边缘质量越高。

(2)定位精度准则。

定位精度的数学表达式为：

$$Localization = \frac{|\int_{-w}^{+w} G'(-x) f'(x) dx|}{\partial \sqrt{\int_{-w}^{+w} f'^2(x) dx}} \tag{3.28}$$

其中，$G'(-x)$ 和 $f'(x)$ 分别表示 $G(-x)$ 及 $f(x)$ 的一阶导数，Localization 值越大，表明定位精度越高。

(3)单边缘响应准则。

定义 f 对噪声的响应中，两个相邻极大值间距离 $x_{\max}(f)$ 为：

$$x_{\max}(f) = 2\pi \left\{ \frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''^2(x) dx} \right\}^{\frac{1}{2}} \tag{3.29}$$

其中，$f''(x)$为$f(x)$的二阶导数。要保证单边缘只有一个像素响应，要求$x_{max}(f)$尽可能大。

2. 双阈值为极值点证明

Canny 算子首先采用高斯滤波器平滑图像，根据高斯函数的一阶偏导数计算梯度，检测梯度幅值的局部极大值，然后用低阈值 T1 得到弱边缘 E1，用高阈值 T2 得到边缘 E2，最后 E1 中仅保留与 E2 有连通关系的连通分量作为输出边缘 E。canny 算子进行边缘检测的过程如图 3.7：

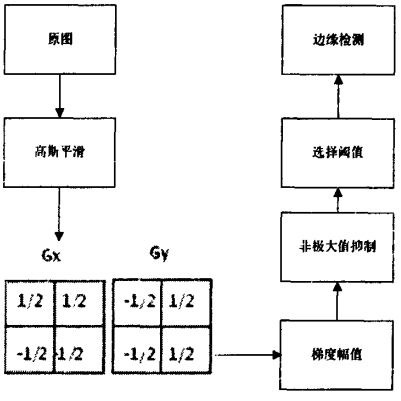


图 3.8 Canny 算子原理

定理 3.1 将经过非极大值抑制后的梯度幅值分为 L 级，模极大值分成 3 类：C_0、C_1、C_2，C_0类为非边缘点的像素，C_2类为边缘点的像素，C_1类包含的像素可能是边缘点，也可能不是边缘点。设 m 为C_0和C_1的边界点，而 k 为C_1和C_2的边界点，则 m，k 就是最适合分割的极值点。

证明：设定n_i为模数为 i 的像素的总数，p_i为该模级像素数占整个图像像素的比率：

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \tag{3.30}$$

令C_0包含模级$[0,1,...,k]$的像素，C_1包含模级$[k+1,k+2,...,m]$的像素，C_2包含模级$[m+1,m+2,...,l-1]$的像素：

$$u_r = u(l-1) = \sum_{i=0}^{l-1} ip_i \tag{3.31}$$

$$\begin{cases} \omega_0(k)=\sum_{i=0}^{k} p_i, & \omega_1(k,m)=\sum_{k+1}^{m} p_i \\ \omega_2(m)=\sum_{m+1}^{l-1} p_i, & u_0(k)=\dfrac{\sum_{i=0}^{k} ip_i}{\omega_0} \\ u_1(k,m)=\dfrac{\sum_{i=k+1}^{m} ip_i}{\omega_1}, & u_2(m)=\dfrac{\sum_{i=m+1}^{l-1} ip_i}{\omega_2} \end{cases} \tag{3.32}$$

$$\begin{cases} \sigma_0^2=\dfrac{\sum_{i=0}^{k}(i-u_0)^2 p_i}{\omega_0} \\ \sigma_1^2=\dfrac{\sum_{i=k+1}^{m}(i-u_1)^2 p_i}{\omega_1} \\ \sigma_2^2=\dfrac{\sum_{i=m+1}^{l-1}(i-u_2)^2 p_i}{\omega_2} \end{cases} \tag{3.33}$$

基于梯度幅度直方图和类内方差最小化确定双阈值的评价函数：

$$J(k,u)=Arg\min(\sigma_w^2)=Arg\min(\omega_0\sigma_0^2+\omega_1\sigma_1^2+\omega_2\sigma_2^2) \tag{3.34}$$

式(3.34)可化为：

$$\begin{cases} J(k,m)=\int_0^k (i-u_0(k))^2 p_i di+\int_{k+1}^m (i-u_1(k,m))^2 p_i di+\int_{m+1}^{l-1}(i-u_2(m))^2 p_i di \\ \dfrac{\partial J(k,m)}{\partial k}=(k-u_0(k))^2 p_k-2u_0^{(1)}(k)\int_0^k (g-u_0(k))p_i di-(k-u_1(k,m))^2 p_k \\ \qquad -2u_1^{(1)}(k,m)\int_m^{k+1}(g-u_1(k,m))p_i di \end{cases} \tag{3.35}$$

出数理统计可知：$\int_0^k (g-u_0(k))p_i di=0$，则：

$\dfrac{\partial j(k,m)}{\partial k}=[k-u_0(k)]^2 p_k-[k-u_1(k,m)]^2 p_k$，令$\dfrac{\partial J(k,m)}{\partial k}=0$，化简可以得到：

$$2k-u_0(k)-u_1(k,m)=0 \tag{3.36}$$

同理：

$$2m-u_1(k,m)-u_2(m)=0 \tag{3.37}$$

求二阶导数：

$$\begin{aligned}\frac{\partial^2 J(k,m)}{\partial k^2} &= p_k^2(k-u_0(k))[1-\frac{ku_0(k)}{\omega_0(k)}p_k] \\ &= 2p_k[(u_1(k,m)-u_0(k))-\frac{(k-u_0(k))^2}{\omega_0(k)}p_i-\frac{(k-u_1(k,m))^2}{\omega_1(k,m)}p_k]\end{aligned} \tag{3.38}$$

同理：

$$\begin{cases}\dfrac{\partial^2 J(k,m)}{\partial m^2} = 2p_m[(u_2(m)-u_1(k))-\dfrac{(m-u_1(k,m))^2}{\omega_1(k,m)}p_m-\dfrac{(m-u_2(m))^2}{\omega_2(m)}p_m] \\ \dfrac{\partial^2 J(k,m)}{\partial m\partial k} = 2p_k(k-u_1(k,m))\dfrac{m-u_1(k,m)}{\omega_1(k,m)}p_m\end{cases} \tag{3.39}$$

因为式(3.29)恒小于 0，则 m、k 为极值点。因为：

$$\begin{cases}u_0(k)=\dfrac{\sum_{i=0}^{k} ip_i}{\omega_0} \\ u_1(k,m)=\dfrac{\sum_{i=k+1}^{m} ip_i}{\omega_0} \\ \dfrac{\partial u_0(k)}{\partial k}=\dfrac{kp_k-p_k\int_0^k ip_i di/\int_0^k p_i di}{\int_0^k p_i di}=\dfrac{p_k(k-u_0(k))}{\int_0^k p_i di} \\ \dfrac{\partial u_1(k,m)}{\partial k}=\dfrac{p_k(u_1(k,m)-k)}{\int_k^m p_i di}\end{cases} \tag{3.40}$$

出数理统计知：

$k-u_0(k)\geq 0, u_1(k,m)-k>0$

所以：$\dfrac{\partial u_0(k)}{\partial k}\geq 0, \dfrac{\partial u_1(k)}{\partial k}\geq 0$

同理可证：$\dfrac{\partial u_1(k,m)}{\partial m}\geq 0, \dfrac{\partial u_2(m)}{\partial m}\geq 0$

因此，$u_0(k)$、$u_1(k,m)$、$u_2(m)$ 为非减函数。根据数理统计的均值性质知：

$$2\times 0-u_0(0)-u_1(0,m)<0 \tag{3.41}$$

$$2\times m-u_0(m)-u_1(m,m)=\frac{\int_0^m(2\times m-i)p_i di}{\int_0^m p_i di}>0 \tag{3.42}$$

由罗尔定理以及梯度直方图的性质知，$2k-u_0(k)-u_1(k,m)=0$ 在区间[0，m]

上必有解。前已经证明，所求 m、k 为极值点。

同理：

$$2k-u_1(k,k)-u_2(k)<0 \tag{3.43}$$

$$2l-u_1(k,l)-u_2(l)>0 \tag{3.44}$$

$2m-u_1(k,m)-u_2(m)=0$ 在区间$[k,l]$上必有解。因此，同时满足式(3.36)、(3.37)的一组 m、k 为所求得的阈值。

3.3.4 一种改进的 Canny 算子

针对传统Canny算子在抑制噪声和检测低强度边缘能力不足的问题，本文给出一种LOG算子和Canny算子相结合的边缘检测方法。首先采用LOG算子对图像进行噪声过滤，然后从以下三个方面改进Canny算子实现边缘检测：设计高斯滤波核对过滤掉噪声的图像进行边缘增强，使低强度边缘更容易被检测；提出在M×N的邻域中计算梯度幅值和方向；将梯度方向结合到梯度幅值的计算中，使梯度幅值在边缘检测中更具依据性。通过对大量增加椒盐噪声的图像进行实验，本文给出的方法在最大程度抑制噪声的同时，能够更多的检测到低强度边缘。

1. LOG 函数对图像去噪

拉普拉斯算子是最常用的二阶基于导数的边缘检测算子，它易受噪点影响，为了减少对噪点敏感度，Marr 和 Hildreth 将高斯滤波和拉普拉斯边缘检测结合在一起，形成了高斯-拉普拉斯(LOG)算子。LOG 首先进行高斯平滑，然后进行拉普拉斯运算。它对噪点不太敏感，因为高斯函数减少噪点，并且拉普拉斯模版使检测到假边缘的概率减到最小。

定义 1：设原图像为 $f(x,y)$，高斯滤波函数为 $G(x,y)$，两者进行卷积运算，然后利用拉普拉斯算子 ∇^2 实现边缘检测，输出的图像 $h(x,y)$，则 LOG 算子的推导如下：

$$h(x,y)=\nabla^2(G(x,y)\otimes f(x,y)) \tag{3.45}$$

式(3.45)中 $\otimes$ 为卷积符号，∇^2 公式为：

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.46}$$

由卷积的可交换行得：

$$h(x,y) = \nabla^2 G(x,y) \otimes f(x,y) \tag{3.47}$$

式(3.47)中$\nabla^2 G(x,y)$即为 LOG 算子：

$$LOG(x,y) = \nabla^2 G(x,y) = \frac{1}{\pi\delta^4}[\frac{x^2+y^2}{2\delta^2} - 1]e^{-\frac{x^2+y^2}{2\delta^2}} \tag{3.48}$$

用 LOG 算子处理图像时，是先用$G(x,y)$进行平滑处理，然后用式(3.46)对图像进行二阶导数增强。

本文采用 LOG 函数进行噪声过滤，研究中发现，LOG 算子在进行边缘检测时，能很大程度的抑制噪声。通过将图像原值与 LOG 函数值进行比较，发现图像原值小于 LOG 函数的部分由图像上绝大部分噪声组成，本文正是利用此原理进行噪声过滤。

本文用下列公式对原图进行噪声过滤：

$$f(x,y) \geq LOG(x,y) \tag{3.49}$$

其中$f(x,y)$为图像的原值，原值如果小于 LOG 函数值，可以肯定为噪声部分，所以用式(3.49)过滤后的图像包含原图主要信息并且主要的噪声部分都被过滤掉。过滤后的图像设为$f'(x,y)$，作为下面 Canny 算子处理的输入图像。

2. 设计高斯滤波核

Canny 算子首先用二维高斯函数的一阶导数，对图像进行平滑，二维高斯函数为：

$$G(x,y) = \frac{1}{2\pi\delta^2}\exp(\frac{1}{2\delta^2}(x^2+y^2)) \tag{3.50}$$

式中 δ 为高斯曲线标准差，用于控制平滑程度。

对于任意输入图像$f(x,y)$，高斯平滑的过程是一个卷积的过程：$f(x,y) \otimes G(x,y)$。常用的高斯算子可以用 3×3 的核对图像平滑：

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

通过 LOG 算子过滤后的图像，已经过滤掉绝大部分噪点，这里需要对处理后的图像边缘进行增强，本文使用下面的高斯滤波核增强图像的边缘：

$$\begin{bmatrix}-1 & -1 & -1\\ -1 & \alpha & -1\\ -1 & -1 & -1\end{bmatrix} \tag{3.51}$$

这里$\alpha=2^i$(i=1，2，3)。

输入图像$f'(x,y)$，用上面的高斯滤波核进行图像边缘增强后的图为

$$f''(x,y)=f'(x,y)\otimes G(x,y) \tag{3.52}$$

3. 改进梯度幅值的计算

传统 Canny 算法利用 2×2 邻域一阶偏导的有限差分来计算平滑后图像$f''(x,y)$各点处的梯度幅值和梯度方向，点(i,j)处水平和垂直两个方向的偏导数$G_x(i,j)$和$G_y(i,j)$分别为：

$$G_x(i,j)=\frac{1}{2}(f''[i,j+1]-f''[i,j]+f''[i+1,j+1]-f''[i+1,j]) \tag{3.53}$$

$$G_y(i,j)=\frac{1}{2}(f''[i,j]-f''[i+1,j]+f''[i,j+1]-f''[i+1,j+1]) \tag{3.54}$$

一阶偏导数$G_x(i,j)$和$G_y(i,j)$即为梯度矢量，点的梯度幅值$M(i,j)$和梯度方向$\theta(i,j)$用梯度矢量根据直角坐标到极坐标的坐标转化公式来计算，则此时点(i,j)处的梯度幅值和梯度方向分别为：

$$M(i,j)=\sqrt{G_x(i,j)^2+G_y(i,j)^2} \tag{3.55}$$

$$\theta(i,j)=\arctan(G_y(i,j)/G_x(i,j)) \tag{3.56}$$

由图 3.7 可以看出，传统 canny 算子判定边缘的唯一依据是图像的梯度幅值，本文结合梯度矢量和方向，给出下面公式计算梯度幅值：

$$M(i,j,\theta)=Max(\cos\theta G_x(i,j),\sin\theta G_y(i,j)) \tag{3.57}$$

实际应用中，仅依据式(3.55)计算出来的梯度幅值信息，在进行双阈值的判定时，容易混淆低强度边缘和噪声，为此需要找到描述边缘与噪声点之间差异的特征信息。图像中的像素点除了具有梯度幅值信息外，还具有梯度方向信息，而且边缘点的梯度方向一般指向边缘的法线方向，而孤立的噪声点则通常没有特定的梯度方向，根据两者在梯度方向特性上的差异，就能够有效地区分边缘点与噪声点。为此，本文将梯度方向信息融合到梯度幅值的计算中，得出了式(3.57)改进的梯度幅值计算方法。

4. 改进梯度矢量的计算

传统 Canny 算子在 2×2 领域内求有限差分均值来计算梯度幅值和梯度方向，对边缘的定位比较准确，但对噪声过于敏感，容易检测出假边缘和丢失一些真实边缘的细节部分。一个好的解决方法是在 M×N 的领域中计算梯度幅值和方向。水平和垂直的梯度矢量分别由式(3.53)和(3.54)计算的到，传统 Canny 算子在 2×2 邻域内计算得到：

$$G_x(i,j)=\begin{bmatrix}\frac{1}{2} & \frac{1}{2}\\ -\frac{1}{2} & -\frac{1}{2}\end{bmatrix},\quad G_y(i,j)=\begin{bmatrix}-\frac{1}{2} & \frac{1}{2}\\ -\frac{1}{2} & \frac{1}{2}\end{bmatrix} \tag{3.58}$$

本文改进梯度幅值和方向在 5×7 的邻域内计算，由下面的核去计算梯度矢量：

$$G_x(i,j)=\begin{bmatrix}1 & \sqrt{2} & 2 & 0 & -2 & -\sqrt{2} & -1\\ \sqrt{2} & 2 & 2\sqrt{2} & 0 & -2\sqrt{2} & -2 & -\sqrt{2}\\ 2 & 2\sqrt{2} & 4 & 0 & -4 & -2\sqrt{2} & -2\\ \sqrt{2} & 2 & 2\sqrt{2} & 0 & -2\sqrt{2} & -2 & -\sqrt{2}\\ 1 & \sqrt{2} & 2 & 0 & -2 & -\sqrt{2} & -1\end{bmatrix},\quad G_y(i,j)=\begin{bmatrix}1 & \sqrt{2} & 2 & \sqrt{2} & 1\\ \sqrt{2} & 2 & 2\sqrt{2} & 2 & \sqrt{2}\\ 2 & 2\sqrt{2} & 4 & 2\sqrt{2} & 2\\ 0 & 0 & 0 & 0 & 0\\ -2 & -2\sqrt{2} & -4 & -2\sqrt{2} & -2\\ -\sqrt{2} & -2 & -2\sqrt{2} & -2 & -\sqrt{2}\\ -1 & -\sqrt{2} & -2 & -\sqrt{2} & -1\end{bmatrix} \tag{3.59}$$

5. 两种不同的边缘检测方式进行融合

本文将使用上面给出的梯度幅值和梯度矢量计算方法分别进行边缘检测，然后进行融合。上面对图像的处理进行到高斯滤波，得到的图像为 $f''(i,j)$，本文首先使用式(3.57)计算出梯度幅值和梯度方向，式中的梯度矢量由传统方法计算得到，依据梯度幅值完成边缘检测，得到输出图像 $f''_A(i,j)$；同时，本文使用改进的核(3.59)计算梯度矢量，然后再用式(3.57)计算出梯度幅值和梯度方向，并依据计算出的梯度幅值完成边缘检测，得到输出图像 $f''_B(i,j)$。最后对两种方法得

到的输出图像 $f''_A(i,j)$ 和 $f''_B(i,j)$ 进行融合，将两种方法分别漏选的真实边缘进行互补，完成最后的边缘检测。

6. LOG 算子和 Canny 算子相结合的边缘检测算法实现

本文基于.NET 平台，使用 C#语言进行算法实现。

Input：24 位 bmp 图像，使用用 *Bitmap* 类型变量 *Detbmp* 保存。

Recognition：

Step1. 给输入图像增加椒盐噪声，添加的椒盐噪声强度为 15%；

Step2. 设 *Detbmp* 图像原值为 $f(i,j)$，利用式(3.48)计算 LOG 函数值 $LOG(i,j)$，然后使用式(3.49)对 *Detbmp* 进行噪声过滤。

Step3. 使用高斯滤波器进行图像边缘增强。使用本文设计的高斯滤波核(3.51)，α 取值为 2，使用式(3.50)进行高斯平滑。

Step4.分别使用两种方法进行边缘检测；

Step4.1. 在 2×2 邻域内计算梯度幅值和方向。首先使用式(3.53)和(3.54)计算梯度矢量 $G_x(i,j)$ 和 $G_y(i,j)$，然后依据式(3.56)计算梯度向量，使用本文给出的式(3.57)计算梯度幅值。接下来进行非极大值抑制处理，使图像上的边缘尽量接近一个像素宽度。最后利用双阈值进行边缘检测，得到边缘检测图像 *DetAbmp*；

Step4.2. 在 M×N 邻域内计算梯度幅值和方向。首先利用本文给出的 5×7 的核(3.59)计算梯度矢量 $G'_x(i,j)$ 和 $G'_y(i,j)$，然后依据式(3.56)计算梯度向量，并使用本文给出的式(3.57)计算梯度幅值。接下来进行非极大值抑制处理，使图像上的边缘尽量接近一个像素宽度。最后利用双阈值进行边缘检测，得到边缘检测图像 *DetBbmp*；

Step5. 图像边缘融合：上面一步得到的边缘检测图像 *DetAbmp* 和 *DetBbmp*，在一定程度上都有对低强度边缘的漏选，存在一些非连贯边缘。本文最后结合两个图像进行处理，以 *DetAbmp* 为参考，对图像 *DetBbmp* 进行边缘修复，最后得到的边缘图像为 *DetAbmp* 和 *DetBbmp* 的融合，在最大程度上检测到真实边缘。

3.4 实验结果与分析

本章实验为边缘检测，将本文改进的 Canny 算法与传统算法进行比较。本文在.NET2008 的环境下进行实验，输入图像都是 24 位灰度图。得到的实验结果如图 3.8 所示：

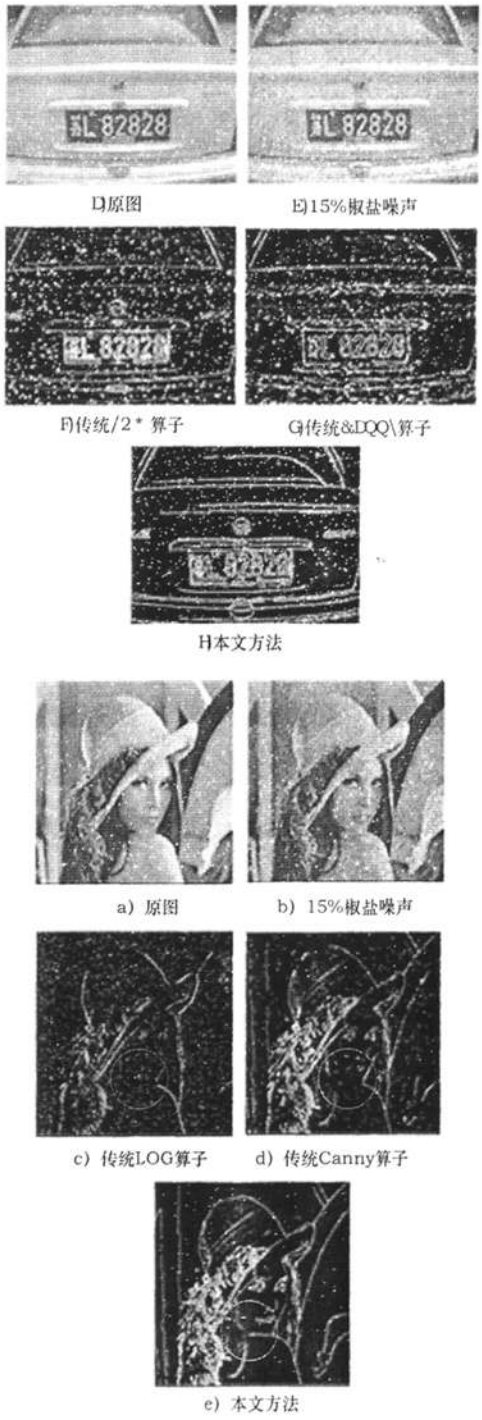


图 3.9 边缘检测实验

图 3.8 的两个实验中，a)为原图，b)为增加了椒盐噪声的图像，噪声强度为15%，c)和 d)分别使用传统 Canny 算子和本文方法对噪声图像进行边缘检测。由结果可以明显看出，Canny 算子对于一些高强度的噪声比较敏感，对于低强度的边缘又容易漏选，而本文给出的方法很好的克服了上面的问题，处理后的图像基本上没有噪声的影响，并且对于一些低强度的边缘也能够准确的检测出来。实验证明了，本文给出的方法在最大的抑制噪声的同时，具有比传统 Canny 算子更加优良的边缘检测性能。

3.5 本章小结

本章主要介绍字符识别中图像预处理的相关方法，包括灰度化、二值化、边缘检测等等。对灰度化和二值化的相关方法进行了比较，重点对边缘检测方法进行了介绍。本章给出了一种改进的 Canny 算子方法用于边缘检测，并且进行了大量的实验，对实验结果进行分析得出了本文给出的方法性能优于传统方法。

图像预处理在字符识别中占有重要地位，好的预处理结果对于字符识别有重要的帮助。

第四章 基于 Shape Context 的字符识别

本章主要介绍形状上下文(Shape Context)算法，结合验证码识别对形状上下文算法的一些相关问题进行探讨，并采用改进的形状上下文算法对复杂验证码进行识别，最后通过实验验证了本文方法的可行性，并将传统模版匹配算法与形状上下文方法的识别结果进行了比较。

4.1 形状上下文概念

4.1.1 形状上下文基本思想

形状上下文是以有限点集来表示待匹配的物体，并假设这些物体内部或者表面轮廓上的离散点足以表征该物体的形状信息。这些点能够用边界检测器来检测边缘像素的位置而得到 $p=\{p_1,p_2,...,p_n\}$。这些点不需要是特殊点，比如曲率最大，变形点等，只要求这些点尽可能均匀一致分布在物体的边缘上。

对于在第一个形状上的每个点 p_i，我们要在第二个形状上找到一个最好的匹配点 q_i。考虑到这样的矢量集，它是形状上的一个取样点到其余点的矢量集，这 $n-1$ 个矢量代表了相对于参考点的整个形状。

构造类似雷达扫描系统的极坐标系：它将整个空间从方向上将 e 平均划分出 8 个方向，在弦上按近似于 $\log_2 r$ 的规则划为 4 份，即 e 代表角度，r 代表周长。这样，整个空间就被分为 32 个区域(32 个 bin)，每个 bin 所占空间从里向往外增大，这是因为对该点具有鉴别力的点是离得最近的点，离该点越远的点对该点的鉴别力越弱，如图 4.1 所示。在同一环上，每个 bin 所占空间是一样大小的，这是因为左、右、前、后相同距离的点对该点的鉴别力是一样的。这种对数极坐标区域表示法用来描述圆状物体每个区域的构成情况，计算区域内每个具体点的极坐标位置，其中的每个环的半径是相等的；而本算法则是用它来描述相对于任意点的其余轮廓点的方向位置及邻居状态，每个环半径不同，不计算区域内具体点的极坐标位置，只统计区域内点的大致个数，两者表示方法不同，目的不同。

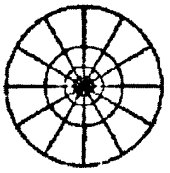


图 4.1 划分成 4×8 的极坐标系

对轮廓上的一给定点 p_i，其属性用以 p_i 点为中心的坐标系的 32 个区域(bin)中落入每个区域的像素点个数 $h_i(k)$ 来描述，按以下公式计算：

$$h_i(k)=\#\{q\neq p_i \,\&\, (q-p_i)\in bin(k)\} \tag{4.1}$$

其中，$k=0,1,2,...,32$，#操作表示 q_i 为落入第 k 个区域中的不同于 p_i 点的轮廓上的其余点的数量[48]。如果落入第 25 个区域的离散点个数为 24，即 $h_i(25)=24$，落入第 31 个区域的离散点个数为 26，即 $h_i(31)=26$。如此由 n 个离散点组成的物体轮廓可在 $O(n^2)$ 时间里获取所有轮廓点的属性描述。

这样就得到了一个有 32 个分量的形状直方图，点集中的每一个点，按照式(4.1)，依次计算与剩下的 n-1 个点构成的形状直方图，最终得到 n 个形状直方图，以 $n\times32$ 大小的矩阵存储。这样，对于任何一个物体，可以用 $n\times32$ 大小的矩阵表示其形状信息。

由上文的点属性描述知，表征物体形状的每个轮廓采样点都有 32 个属性值，令第 i 个轮廓采样点的 j 个属性值为：

$$a_{i,j}(k)=\#\{q_k\neq p_i \mid q_k\in bin(j)\} \tag{4.2}$$

则第 i 个轮廓采样点的 32 个属性值可构成一个序列 $(a_{i1},a_{i2},a_{i3},...,a_{i32})$，将这一序列作为矩阵的第 i 行，则对一个用 $2n$ 个采样点表征的形状轮廓，可构造出一个 $2n\times32$ 形状矩阵：

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,32} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i,1} & a_{i,2} & a_{i,3} & \cdots & a_{i,32} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{2n,1} & a_{2n,2} & a_{2n,3} & \cdots & a_{2n,32} \end{bmatrix}$$

图像对数极坐标变换是指将图像从笛卡尔坐标系转换至对数极坐标系。这

样，笛卡尔坐标系下的图像匹配问题就转化为对数极坐标下的图像匹配问题。图像 $I(x,y)$ 到 $I(r,\theta)$ 的对数极坐标变换定义为：

$$r=\sqrt{(x-x_0)^2+(y-y_0)^2},\quad \theta=\arctan(\frac{y-y_0}{x-x_0}) \tag{4.3}$$

取坐标原点 (x_0,y_0) 为 $(0,0)$，其用复数 z 表示为：

$$z=x+jy=r(\cos\theta+j\sin\theta)=re^{j\theta} \tag{4.4}$$

令 $w=\ln z=\ln r+j\theta$，则笛卡尔坐标转换为对数极坐标的映射方程为 $p(r,\theta)=\ln r$，$q(r,\theta)=\theta$。

当笛卡儿空间中的图像相对于坐标原点发生了缩放和旋转变化，例如图像放大了 r_0 倍，旋转了 θ_0 角度，变化后新的相应极坐标为 $(r_0+r,\theta_0+\theta)$ 时，对数极坐标空间 $w(u,v)$ 的坐标将产生平移，即：

$$u=\ln r_0+\ln r \tag{4.5}$$

$$v=\theta+\theta_0 \tag{4.6}$$

当笛卡儿空间 z 图像的比例变化相当于对数极坐标空间 ω 图像的水平位移，空间 z 图像的旋转变化相对于空间 ω 图像的垂直位移。这就是所谓的对数极坐标映射的二维不变性。

如图 4.2 所示，a)和 b)表示两个形状的边界取样点，c)为计算形状上下文的对数极坐标变换图，距离参数 $\log r$ 取 5，方向参数取 θ 取 12。可以发现越靠近原点，区域分得越细。(d-f)分别是标注○，◇，△的三个点的对数极坐标直方图，可以发现，用○和◇标注的两个点的形状直方图比较相似，因为这两个点的位置大致上是对应的。每个点的形状上下文是不同的，相似图像的对应的点有相似的形状上下文。点集 $p=\{p_1,p_2,...,p_n\}$ 中的每一个点，按照式(4.1)，依次计算与剩下的 n-1 个点构成的形状直方图，最终可得到 n 个形状直方图。

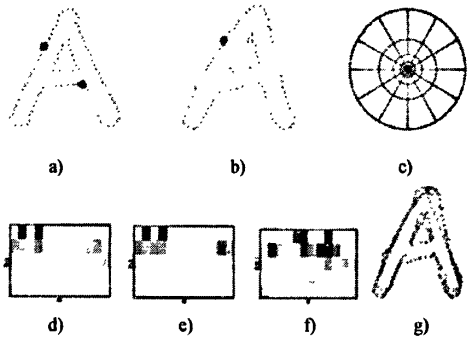


图 4.2 a)、b)是两个形状的边界取样点 c)对数极坐标变换图 (d-f)分别是标注○，◇，△的三个点的形状直方图 g)使用双向匹配的效果图

4.1.2 相似度度量

在图 4.2 中，要判定图 a)和 b)图之间的相似度，应比较这两个图形中对应的边界取样点的相似程度。我们可以根据每个取样点的形状直方图，来找到与该点最相似的点，然后综合所有点，以确定形状的相似程度。

设 p_i 为第一个形状上的一点，而 q_j 是第二个形状上的一点。定义 $C_{ij}=C(p_i,q_j)$ 代表这两个点匹配的 *Cost* 值。因为形状上下文由形状直方图来表示的，所以就自然用 x^2 距离：

$$C_{ij}=\frac{1}{2}\sum_{k=1}^{k}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \tag{4.7}$$

其中，$h_i(k)$ 代表形状 p 和 p_i 的形状直方图，$h_j(k)$ 代表形状 p 和 p_j 形状直方图。根据式(4.7)，可以计算得到 p_i 和 p_j 的 *Cost* 值，为找到与 p_i 最匹配的点，要使这个 *Cost* 值最小。对于两个形状上所有的点的 *Cost* 值 C_{ij} 的集合，我们可以求得总的 *Cost* 值，其值越小，说明两个形状越相似；反之，两个形状相差越大。上述点的匹配问题即为典型的双向图的匹配问题，利用匈牙利算法，该问题得到解决[49]。

4.2 验证码识别

4.2.1 验证码概念

1. 验证码定义

随着互联网技术的发展和应用，现代社会飞速前进，信息技术日新月异，伴随而来的就是 web 系统的安全性问题。web 系统的安全性涉及两方面，即服务器程序与 B/S 程序，验证码的出现正是加强 web 系统安全的产物。

在国内，验证码是指包含有一串数字或其它字符的一幅图片，图片里加上一些干扰象素。攻击者编写的攻击程序，难以识别验证码字符，顺利的完成自动注册，登录。而用户可以识别填写，所以这就实现了阻挡攻击的作用。

在国外，称为 CAPTCHA，是 Completely Automated Public Turing test to tell Computers and Humans Apart(全自动区分计算机和人类的图灵测试)的简称，已由卡内基梅隆大学注册成商标。CAPTCHA 的目的是区分计算机和人类的一种程序算法，这种程序必须能生成并评价人类能很容易通过但计算机却通不过的测试。这个要求本身就是悖论，因为这意味着一个 CAPTCHA 必须能生成一个它自己不能通过的测试。

2. 验证码的作用

(1)防止对网站的批量注册，重复发贴。

(2)防止密码被暴力破解。在浏览器中如需输入用户名与密码才能登录某个服务，该用户名一旦被他人获取，就可对密码进行猜解。如用户在登录某个服务时，需要输入随机产生的一个验证码，让用户填写验证码上显示的字符，这可使他人猜解密码不能轻易实现。

(3)防止他人使用广告软件发布大量的垃圾信息。

(4)防止对网站的恶意攻击。网站常会遇到合法客户机的恶意攻击，其中一种很常见的攻击手段就是身份欺骗，攻击者利用客户端程序的漏洞，或通过木马在页面里置入脚本，当用户登陆后，这些脚本就可以偷偷的利用“http-post”向服务器发送请求，或提交垃圾数据，或窃取数据提交到攻击者操纵的服务器，或者密集执行消耗服务器资源的操作，降低网站性能，乃至使网站崩溃。在关键的部分使用验证码可以在一定程度上，防止合法客户被攻击后，对服务器造成大的

伤害。

3. 验证码的分类

根据分割和识别的难易，本文将验证码分为以下几种，见图 4.3 所示：

1） 字符没有粘连。这类验证码人和计算机程序都比较容易识别，只要有足够多的样本数量，识别率都能接近于 100%。

2） 字符粘连且轻度扭曲。这类验证码人很容易识别，但是计算机程序比较难识别，是现在网络上使用比较多的一种验证码。

3） 字符粘连且极度扭曲。这种验证码人和计算机程序都很难识别。用户体验行比较差，国内网站很少用。

图 4.3 根据识别难易对验证码分类

4.2.2 验证码识别过程

验证码识别[50][51][52]的一般步骤如下：从网站上获得验证码原图；由于获得的图像一般是彩色的，所以要先对彩色图像进行灰度化，得到 256 色灰度图；然后对灰度图进行二值化，去除干扰背景，保留字符信息；二值化后的图像上会存在很多噪声，要通过去噪去除干扰噪声；之后要把图像上的单个字符分割出来，能否完整地分割出单个字符对于后面的识别是至关重要的；最后就是识别字符并输出结果了。

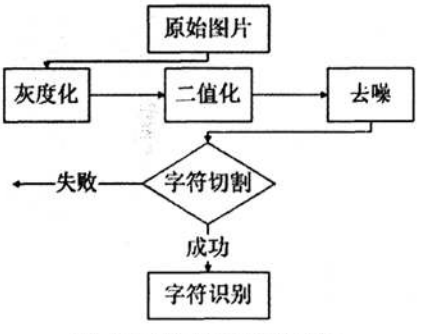


图 4.4 验证码识别流程图

4.2.3 简单验证码识别

图 4.3 中的第一类验证码属于比较简单的验证码，字符间间隔比较大，很容易对字符进行切割，只要有足够多的样本数量，识别的精度能接近于 100%。本文以第一类验证码为例进行简单介绍。

验证码图像的预处理是指在对验证码图像进行分割、特征提取和识别前所进行的处理。预处理一般包括灰度化、二值化、平滑、去噪等。

灰度化处理是指把含有亮度和色彩的彩色图像变换成灰度图像的过程。这里采用的是加权平均值法，利用公式 Grey = 0.29×red + 0.587×green + 0.114×blue 算出每个像素的灰度值 Grey，采用 128 作为阈值进行二值化。

图 4.5 验证码二值化

字符分割是指从背景中将单个有意义的字符提取出来，以便于下一步的字符识别算法对单个字符进行逐个识别。这里采用的是投影法进行切割。

图 4.6 投影效果图及切割出来的单个字符

对切割出来的单个字符再进行编码，编码方式很简单，对于图片上像素点为黑色的编码为 1，像素点为白色的编码为 0，然后把编码后的字符串保存为样本。

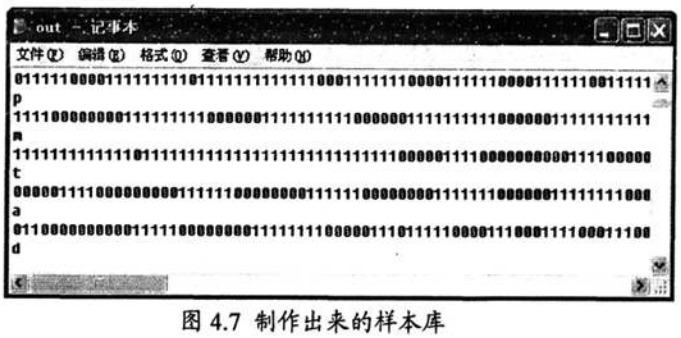


图 4.7 制作出来的样本库

当样本数量足够多时，就可以进行程序自动识别，识别的过程与样本制作的过程一样，就是在最后一步时不同，将编码后的字符串到样本库去匹配，找出相似度最高的，判定为识别出来的字符。

4.3 改进形状上下文用于复杂验证码识别

4.3.1 复杂验证码

图 4.3 中的第一类验证码属于简单验证码，这种验证码的共同点是能够对图片上字符进行切割，得到单个字符，然后进行匹配识别。本文研究的对象是复杂验证码，如图 4.3 中第二类验证码，这类验证码字符粘连，且有一定扭曲，没有有效的算法能够准确的切割出单个字符，所以很难进行识别。

图 4.8 复杂验证码

本文给出一种形状上下文的改进方法，采用不对图片进行切割，整体识别的方法，使用单像素跟踪算法获取字符的轮廓，有效的减少了像素点数目，降低了形状上下文描述的复杂度，采用半圆形式的对数极坐标建模，解决了两个字符粘连处字符建模时互相干扰的问题，准确的描述出字符的特征，实现了对字符粘连的复杂验证码的识别。识别过程如下。

4.3.2 单像素取字符轮廓

首先对验证码图片进行预处理。预处理一般包括灰度化、二值化、平滑、去噪等。灰度化处理是指把含有亮度和色彩的彩色图像变换成灰度图像的过程。这里采用的是加权平均值法，利用公式(4.8)算出每个像素的灰度值 Grey。

$$Grey = 0.29 * red + 0.587 * green + 0.114 * blue \tag{4.8}$$

根据灰度图片的 Grey 值进行二值化处理，将灰度图转化为黑白两色图。取 128 作为阈值，根据计算得到的 Grey 值，值大于 128 的像素点 RGB 值设置为(255，255，255)，而值小于 128 的像素点设置为(0，0，0)。

示例图片背景简单，没有噪点，二值化完成后就得到了预处理后的图片，如图 4.9 所示。

如果直接对预处理后的图片进行形状上下文的建模，将使算法的时间开销非常大，根据形状上下文定义可知，只需要对字符轮廓像素点进行建模，就足够描述出字符的特征。为了减少算法复杂度，这里只对字符的轮廓点进行形状上下文描述，舍去字符上非轮廓点。

定义 1 围绕该区域生成的，区域中所有像素均在其内部的区域轮廓，称为

区域外边界。

定义 2 围绕该区域生成的，区域中所有像素均在其外部的区域轮廓，称为区域内边界。

根据定义 1 和 2，要分别获取字符的外边界和内边界。有的字符只有外边界，而有的字符包括外边界和内边界。

本文参考文献[53]提出的轮廓跟踪算法，并针对验证码字符特征，进行了一些改进。

示例验证码图片上 4 个字符都两两粘连，对于外部轮廓，只需使用带标记信息的轮廓跟踪算法进行获取。从图片左上角开始扫描，由上到下，由左至右找到第一个黑色像素点，设置为外部边界的开始点，使用约定符号对该点的 8 个相邻像素点进行标号(见左图)。用 0 和 1 分别表示该边界像素点没有被搜索和已经被搜索。对于图中心的边界像素点，找到标号的 8 个相邻的一个没有被搜索的黑色像素点，并且该像素点的相邻一个像素点为白色，即为下一个开始搜索的轮廓像素点，将该点的标记信息设置为 1，从该点开始接着找相邻的未搜索过的边界像素点。以此类推可以得到一个封闭的回路，直到搜索到第一个被标记的边界像素点。便得到了 4 个字符的整个外部边界。

3	2	1
4		0
5	6	7

有的字符还有内边界，需要对图片从左到右，从上到下再扫描一遍。从图片上方到下方扫描像素点时，找到第一个有梯度变化，并且没有被标记的黑色像素点，即为内边界的开始点，扫描一遍后可以得到所有内边界的开始点，然后使用带标记的轮廓跟踪算法进行处理，得到每一个内边界。如果没有搜索到内边界的开始点，即表示验证码上的 4 个字符都没有内边界。得到所有的内边界后，就可以得到单像素表示的字符轮廓图，如图 4.9 c)所示。

a)原图　b)二值化图　c)单像素轮廓图

图 4.9 图像二值化及单像素取轮廓

4.3.3 半极坐标圆建模制作样本字符

形状上下文采用对数极坐标圆描述圆心像素点的特征，描述出该像素点与周围像素点的相对分布角度与距离。然而对于字符粘连的验证码字符，两个字符粘连处的像素点在使用对数极坐标圆建模时，会受到另一个字符像素点的干扰，而

不能准确描述出目标像素点的形状上下文特征。为了解决两字符粘连处像素点建模互相干扰的问题，本文给出了半极坐标圆建模的形状上下文方法。

在制作样本时，由于没有算法能够对验证码上字符进行自动切割，只能手工切割字符进行样本的制作。先从单像素轮廓图上手工切割出单个字符，然后进行特征值的提取。样本特征值提取采用了形状上下文方法，对每个轮廓像素点进行半极坐标圆建模。

一般图像中像素的位置可以用笛卡尔坐标表示为(x,y)，也可以用极坐标表示为(r,θ)。设圆心的坐标为(x_0,y_0)，相对于圆心坐标，图像中其它任意一点坐标值设为(x,y)，则，笛卡尔坐标系转换到极坐标系的变换过程如下：

$$r=\sqrt{(x-x_0)^2+(y-y_0)^2},\quad \theta=\arctan(\frac{y-y_0}{x-x_0}) \tag{4.9}$$

取坐标原点(x_0,y_0)为$(0,0)$，其用复数z表示为：

$$z=x+jy=r(\cos\theta+j\sin\theta)=re^{j\theta} \tag{4.10}$$

令$\omega=\ln z=\ln r+j\theta$，则笛卡尔坐标转换为对数极坐标的映射方程为$p(r,\theta)=\ln r$，$q(r,\theta)=\theta$。本文的对数极坐标$\theta$值被划分为8份，而$r$值被划分为3份，$\theta$值取-90°到90°，或者90°到270°的半极坐标圆进行建模。这样可以得到一个4×3的矩阵。相对于坐标原点(x_0,y_0)，根据坐标(x,y)的r和θ值可以判定该点位于哪一个格内。

取字符宽度的一半作为对数极坐标圆的半径值，字符左半边的像素点只编码右半圆的值，而右半边的像素点只编码左半圆的值，这样处理有利于进行后面的字符整体识别。

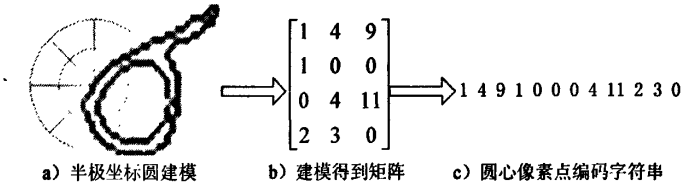


图 4.10 改进的形状上下文建模

图4.10中字符6一共有像素点115个，每个像素点都可以编码成这样的一个

4×3 的矩阵，而得到一个描述非常丰富的样本。

4.3.4 字符整体识别

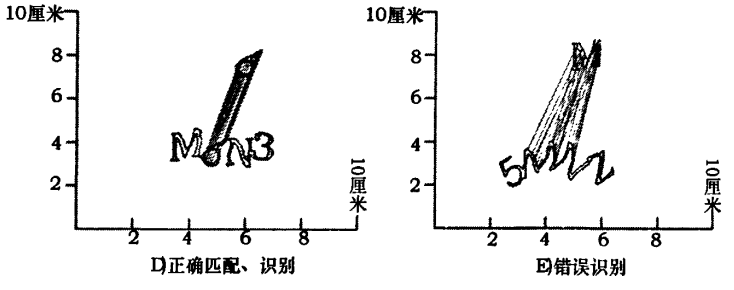
在样本足够多以后，就可以实现程序的自动识别。识别过程中不对测试验证码图片进行切割，以样本的每个字符作为匹配源字符，将样本的每个字符到图片上去比对，以该样本的宽度作为直径，对图片上的每个像素点进行左右半圆的形状上下文建模，找出编码结果与样本中的点能够匹配的点，本文设定相似度 95%以上即判定为匹配成功，如果相似度没有达到 95%以上的，选择相似度最高的进行匹配。对整个样本库扫描一遍之后，就可以识别出图片上有哪几个字符，再根据各个字符像素点在图片上的相对位置排好顺序，从而实现程序的自动识别。

假设样本上一个点的形状上下文为：$h_i=\{h_i(1),...,h_i(k)\}$，测试图片上任意一点的形状上下文为：$h_j=\{h_j(1),...,h_j(k)\}$，这里 k=12。则相似度 *Cost* 值由下面公司计算：

$$C_{i,j}=1/2\sum_{k=1}^{k}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \quad \text{k=12} \tag{4.11}$$

Cost 值在 0~1 之间，*Cost* 值越小，相似度越高，两点的匹配度越高。

本文只所以取半圆，在于半圆就足够对像素点形状描述，而且消除了两个字符连接点附近像素点建模受到旁边字符的干扰，半圆就能够准确的描述出该点的形状上下文信息。整体识别的效果如图 4.11 所示。



a)正确匹配、识别　　b)错误识别

图 4.11 字符整体识别

4.3.5 算法思想

本文基于.NET2008 平台，使用 C#语言进行算法实现，参考了文献[23]的部

分代码。

Input：网络上随机获取验证码，用 Bitmap 类型变量 *Recbmp* 保存。

Recognition：

Step1. 用式(4.8)对 *Recbmp* 进行灰度化，取 128 作为阈值，使用 SetPixel 类将图片设置为只有(255，255，255)和(0，0，0)的黑白图；

Step2. *0~Recbmp. Height-1* 和 *0~ Recbmp .Width-1* 搜索到第一个外边界像素点(x_0, y_0)，将该点标记值设置为 1；

Step2.1.从(x_0, y_0)的 8 个相邻像素点中搜索下一个边界点(x_1, y_1)，标记值设置为 1；

Step2.2.重复步骤 2.1 直到搜索到边界点(x_0, y_0)为止，得到所有字符外边界。

Step2.3.根据梯度变化，从 *0~Recbmp.Height-1* 和 *0~ Recbmp.Width-1* 搜索未标记的黑色像素点为内边界像素点开始点$(x_i, y_j)...(x_n, y_n)$，使用步骤 2.1 和 2.2 得到每一个内边界。

Step3. 取一个样本字符，取样本字符宽度 *Height* 作为极坐标直径 *r*，根据样本字符取左(右)半极坐标圆建模，使用式(4.9)和(4.10)，对验证码图片上每一个像素点(x_0, y_0)分别进行左(右)半极坐标圆建模。

Step4.使用式(4.11)，根据每个像素点编码值与测试图片上像素点编码值，计算样本字符与测试图片上像素点组成字符的相似度 *Cost* 值；

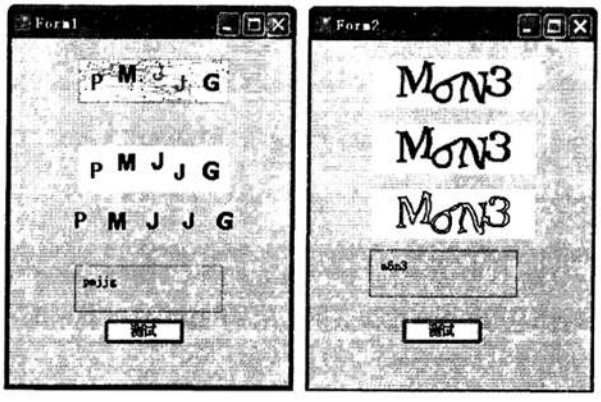
Step5. 根据当前的 *Cost* 值判定测试图片上是否存在该样本字符，然后接着取下一个样本字符，重复 Step3 和 Step4；

Step6.找出 *Cost* 值最小的 4 个字符后，根据像素点相对位置对字符排序，完成字符识别。

4.4 实验结果与分析

本文用两种方法对复杂验证码识别进行实验，一种方法是制作不同数量的样本进行实验，将识别率与简单验证码识别(采用传统模版匹配方法)进行比较。另

一种方法，每个样本选取不同像素点数目进行建模，观察识别率随建模像素点数目改变情况。



a) b)

图 4.12 识别结果，其中 a)为简单验证码识别(模版匹配方法实现)b)为复杂验证码识别(本文方法实现)

实验一：示例验证码有数字和字母，每个字符构造一个样本，即有 36 个样本，这种情况下随机识别 100 张图片，识别率为 60%。本文对简单验证码也进行了大量实验，同样只有一个样本的情况下，普通方法对简单验证码的识别率也只有 46%。然而，随着样本数量的增加，简单验证码的识别率能很快接近于 100%，而复杂验证码的识别率最高只能到达 80%左右，如表 4.1 所示。

表 4.1 每个字符样本数量不同的识别率

字符样本数	1	2	3	4
简单验证码	46%	67%	84%	93%
复杂验证码	60%	71%	78%	82%

实验二：对字符形状上下文建模时选取不同数量的像素点数目。图 4.10 中字符 6 包含有 115 个像素点，其实两个相邻的像素点形状上下文基本相同。实验时选择全部像素点建模，相邻的两个像素点选择一个建模以及相邻的三个像素点建模一个。得到的结果如表 4.2。

表 4.2 选择不同数量的像素点建模得到的识别率

	全部像素点	1/2 像素点	1/3 像素点
识别率	78%	70%	67%

表 4.2 选择不同数目的像素点进行建模，随机识别 100 张图片的结果。由图可以看出，减少建模的像素点数目，程序自动识别的正确率变化很小，然而程序

运行速度明显加快。

4.5 本章小结

本章研究了形状上下文方法，通过验证码识别进行实验。使用改进的形状上下文方法对复杂验证码进行识别，采用整体识别的方法，不对图片进行切割，使用半极坐标圆进行建模的方式，解决了两个字符连接处像素点建模互相干扰的问题，设计实现了复杂验证码的识别算法，并与简单验证码的实现过程进行了比较。通过大量实验证明，本文给出的方法能够对字符粘连的复杂验证码进行识别。

第五章 基于 BP 神经网络的字符识别

本章主要介绍 BP 神经网络及 BP 算法，结合车牌字符识别对 BP 网络的一些相关问题进行探讨，并采用改进的 BP 网络进行车牌字符的识别，最后通过实验对各种改进的 BP 算法进行性能比较，并将传统模版匹配算法与 BP 神经网络的识别结果进行了比较。

5.1 BP 神经网络

5.1.1 BP 神经网络算法原理

BP 神经网络是由非线性变换单元组成的有教师指导型前馈网络。BP 网络由输入层、隐含层(可为一层或多层)和输出层构成，各层之间实现全连接。如图 5.1 所示。

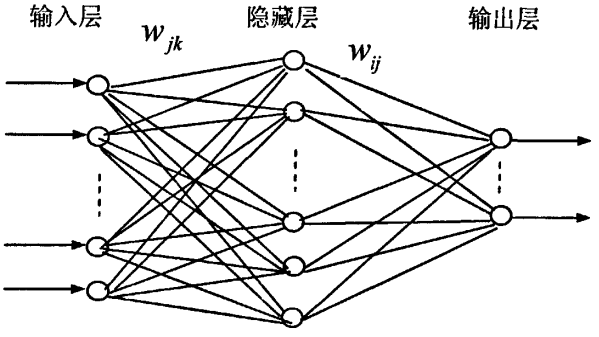


图 5.1 三层 BP 神经网络的结构

BP 网络的学习由四个过程组成：输入模式由输入层经中间层向输出层的“模式顺传播”的过程，网络的希望输出与网络实际输出之差的误差信号由输出层经中间层向输入层逐层修正连接权的“误差反向传播”的过程，由“模式顺传播”与“误差反向传播”的反复交替进行的网络“记忆训练”过程，网络趋向收敛即网络的全局误差趋向极小值的“学习收敛”过程。归结起来为，“模式顺传播”→“误差反向传播”→“记忆训练”→“学习收敛”过程。BP 网络学习规则有时也称广义 δ 规则。

下面分别介绍和分析这四个过程。

1. 模式的顺传播

模式的顺传播过程是从输入模式提供给网络的输入层开始的。输入层各个单元对应于输入模式向量的各个元素。设输入模式向量为：

$$\begin{aligned} A_k &= (a_1, a_2, ..., a_n) \\ k &= 1, 2, ..., m \end{aligned} \tag{5.1}$$

M 为学习模式对数，N 为输入单元个数；

对应输入模式的希望输出向量为：

$$Y_k = (y_1, y_2, ..., y_q) \tag{5.2}$$

q 为输出层单元数；

首先，按式(5.3)计算中间层各单元的输入。

$$\begin{aligned} S_j &= \sum_{i=1}^{n} w_{ij} a_i - \theta_j \\ j &= 1, 2, ..., p \end{aligned} \tag{5.3}$$

式中， w_{ij} 为输入层至中间层的连接权；

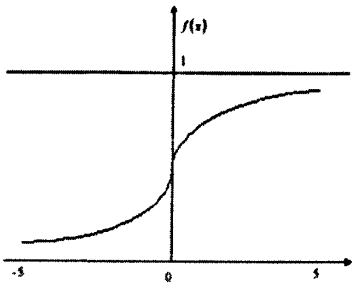
θ_j 为中间层单元的阀值；

p 为中间层单元个数。

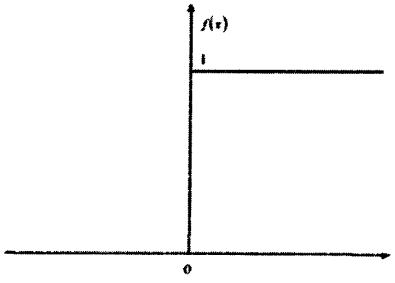
为模拟生物神经的非线性特性，以 S_j 作为 S 函数(Sigmoid 函数)的自变量，计算中间层各单元的输出。 S 函数的数学表达式如式(5.4)所示：

$$f(x) = \frac{1}{1 + e^{-\frac{x}{x_0}}} \tag{5.4}$$

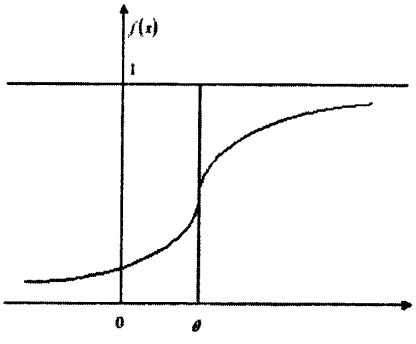
S 函数的输出曲线如图 5.2 a)所示：



D)6函数曲线



E)阶跃函数曲线



F)向右平易 θ 单位的6函数曲线

图 5.2 非线性输出曲线

由图可知，S 函数输出曲线两端平坦，中间部分变化剧烈。当 x<-5 时，f(x)接近于 0；当 x>5 时，f(x)接近于 1；而当 x 在 0 附近时(-5<x<5)，f(x)才真正起转移作用。S 函数与阶跃函数(如图 5.2 b)所示)相比，从形式上看具有“柔软性”；从生理学角度看，一个人对远远低于或高于他智力和知识水平的问题，往往很难产生强烈的思维反映；从数学角度看，S 函数具有微分性。正是因为 S 函数更接近于生物神经元的信号输出形式，所以选用 S 函数作为 BP 网络的输出函数。同时 BP 学习规则本身也要求网络的输出函数是可微分的(后面将介绍其原因)，S

函数不但具有可微分性，而且具有饱和非线性特性，这又增强了网络的非线性映射能力。把式(5.3)代入式(5.4)得：

$$b_j = f(s_j) = \frac{1}{1+e^{-s_j}} = \frac{1}{1+e^{-\sum_{i=1}^{n} w_{ij}a_i + \theta_j}} \qquad (5.5)$$
$$j = 1,2,...,p$$

式中，b_j为中间层j单元的激活值。单元输出阀值θ_j是为了模拟生物神经元的阀值电位而设置的。实际上在网络的学习过程中，它和各连接权一样也不断地被修正。阈值θ_j的作用反映在S函数的输出曲线上，是使曲线向右平行移动了θ_j个单位，如图5.2 c)所示，它起到了调节神经元兴奋水平的作用。

按模式顺传播的思路，以式(5.6)、式(5.7)计算输出层各单元的输入、输出。

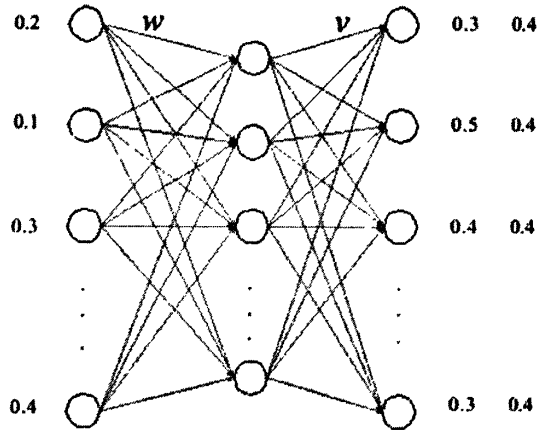
$$L_t = \sum_{j=1}^{n} v_{jt} \times b_j - \gamma_t \qquad (5.6)$$

$$C_t = f(L_t) \qquad (5.7)$$
$$t = 1,2,...,q$$

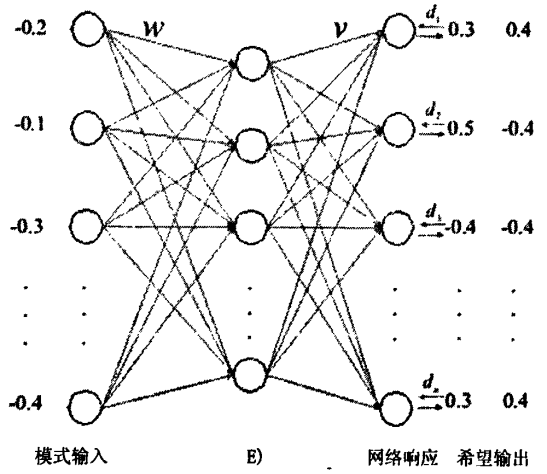
式中，v_{jt}为中间层至输出层连接权；γ_t为输出层单元阈值；f为S函数；至此，一个输入模式完全了一遍顺传播过程。

2. 误差的逆传播

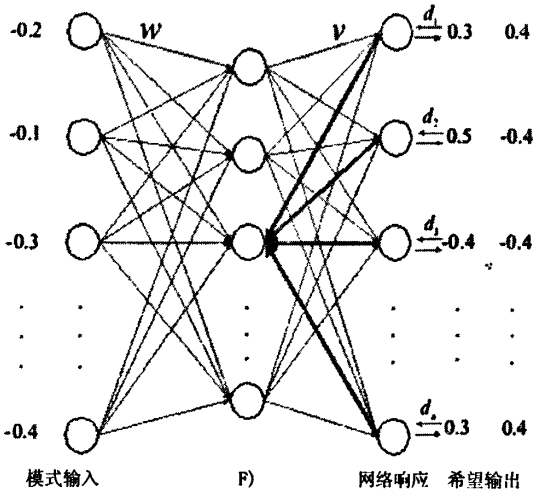
作为误差逆传播的第一步，是进行误差计算。如图5.3所示，误差逆传播过程是由输出层的误差d_j向中间层的误差e_i传递的过程：



模式乎序传播后网络响应和希望输出比较



计算输出层校正误差，并修正连接层Y



计算E间层校正误差（E粗箭头表示误差从E间层向传播）

图 5.3 “顺传播”和“逆传播”示意图

在这里输出层的校正误差按式(5.8)计算：

$$d_t^k = (y_t^k - C_t^k) f'(L_t)$$
$$t = 1,2,...,q \tag{5.8}$$
$$k = 1,2,...,m$$

上式的物理意义是：$(y_t^k - C_t^k)$项表示网络希望输出与实际输出的绝对误差；$f'(L_t)$项根据各单元的实际响应调整偏差量。如图 5.4、5.5 所示，当输出层某单元的输入L_t在 0 附近时，其输出变化幅度较大，而此时$f(L_t)$的导数$f'(L_t)$正好处于峰值附近，当与$(y_t^k - C_t^k)$项相乘后，增强了偏差的校正作用。反之，当L_t的绝对值比较大时，其输出变化幅度很小，也就是说此单元所起的校正作用不大，而此时$f'(L_t)$正好处于较小值的部分，当与$(y_t^k - C_t^k)$项相乘后，减弱了偏差的校正作用。

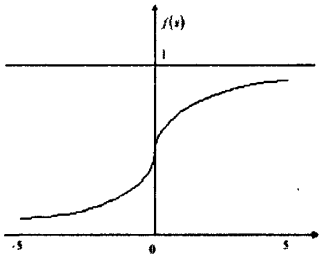


图 5.4 激励函数 f(x)的波形图

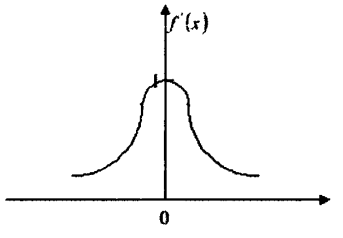


图 5.5 激励函数的导数$f'(x)$的波形图

为完成误差向中间层的传递，需要按式(5.9)计算中间各单元的校正误差：

$$e_j^k = \left[\sum_{t=1}^{q} v_{jt} \times d_t^k\right] f'(s_j)$$
$$j = 1,2,...,p \tag{5.9}$$
$$k = 1,2,...,m$$

得到校正误差d_t^k与d_j^k之后，沿逆方向调整输出层至中间层、中间层至输入层之间的连接权，以及各单元的输出阀值。其调整量按式(5.10)～(5.13)计算：

$$\Delta v_{jt} = \eta \times d_t^k \times b_j^k \tag{5.10}$$

$$\Delta \gamma_t = \eta \times d_t^k$$
$$j = 1,2,...,p \tag{5.11}$$
$$t = 1,2,...,q$$
$$k = 1,2,...,m$$

$0<\eta<1$(学习系数)

$$\Delta w_{ij} = \alpha \times e_j^k \times \alpha_i^k \tag{5.12}$$

$$\Delta \theta_j = \alpha \times e_j^k \tag{5.13}$$

$0<\alpha<1$(学习系数)

$$i = 1,2,...,n$$
$$j = 1,2,...,p$$
$$k = 1,2,...,m$$

下面以式(5.10)为例，说明连接权的调整机理。由式(5.10)可知，连接权的调整量依赖于三个因素：η、d_i^k和b_j^k。首先，调整量与误差d_i^k成比例，即误差越大，调整的幅度也就越大，这一物理意义是显见的；其次，调整量与连接权所对应的中间层单元的输出值b_j成比例，其物理意义是，某中间层单元的激活值越高，它在这次学习过程中就显得越活跃，则与其相关的连接权的调整幅度也就应该越大。调整量与学习率η成比例，通常η的值在 0.25 至 0.75 之间。η的大小影响学习的速度，为了使整个学习过程适当加快，又不至于引起振荡，可采用变学习率的办法，即在学习初期取较大的η值，随着学习过程的进行逐渐减小η值。

3. 训练过程

所谓训练过程[54]，是指反复学习的过程，也就是根据教师示教的希望输出与网络实际输出的误差调整连接权的过程。而希望输出实际上是对输入模式分类的一种表示，是人为设定的，所以也因人而异。随着“模式顺传播”与“误差逆传播”过程的反复进行，网络的实际输出逐渐向各自所对应的希望输出逼近。图 5.6 所示的是一组由三个元素构成的三种学习模式对。

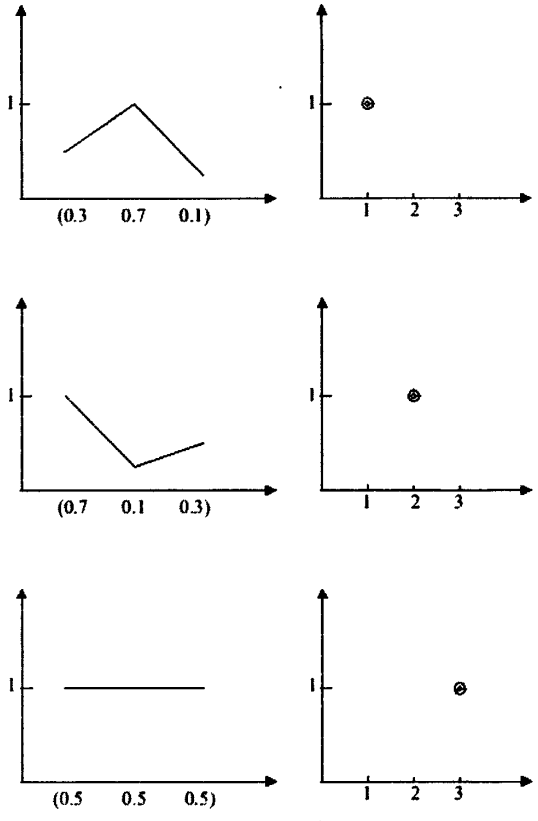


图 5.6 训练集合举例

对于典型的 BP 网络，一组训练模式，一般要经过数百次乃至几千次的学习过程，才能使网络收敛。

4. 收敛过程

前面已经多次指出，学习或者说训练的收敛过程就是网络全局误差趋向于极小值的过程。一般全局误差有公式(5.14)和式(5.15)两种定义：

$$E_k = \sum_{j=1}^{k} (y_j^k - y_j)^2 / 2 \tag{5.14}$$

$$E = \sum_{k=1}^{M} E_k \tag{5.15}$$

值得注意的是，虽然网络的收敛标志着网络的响应非常逼近了希望输出模式，但是，这种逼近不能简单地用“是”或“不是”来衡量。网络地响应往往是以实数的形式出现，而不是逻辑值。在这里“逼近”是一个模糊的概念，有点像模糊数学中的隶属度。例如，当某 BP 网络的希望输出模式为 Y =(1，0，0)时，

学习收敛后的网络的实际响应为 C=(0.988，0.012，0.036)，这时可以说网络的实际响应已经十分逼近希望输出模式，但不能认为这就是最好的逼近。在网络进行回想时，网络的响应也往往是一个近似的模式分类。例如，当三种标准分类 A、B、C 分别用希望输出模式 Y_A=(1，0，0)，Y_B=(0，1，0)，Y_C=(0，0，1)表示时，当某一被识别的输入模式提供给网络回想后，其输出为(0.03，0.76，0.99)。这时，我们应该认为，输入模式的第一可能类别时 C 类，第二可能是 B 类，而不可能是 A 类。

对于 BP 网络，其收敛过程存在着两个很大的缺陷：收敛速度慢；存在所谓“局部最小值”问题。在学习过程中有时会发现，当学习反复进行到一定次数以后，虽然网络的实际输出与希望输出还存在很大的误差，但无论如何学习下去，网络全局误差的减小速度都变得十分缓慢，或者根本不再变化。这种现象就是因为网络收敛于局部极小点所致。我们知道，BP 网络的全局误差 E(也称为代价函数或者能量函数)，是一个以 S 函数为自变量的非线性函数。这就意味着由 E 构成的连接权空间不是只有一个极小点的抛物面，而是存在多个局部最小点的超曲面。因此，BP 网络的收敛过程，很可能在遇到局部极小值便被“冻结”住，而无法最终收敛于全局最小点，也就无法对学习模式准确记忆。导致 BP 网络这一缺陷的原因，是由于 BP 学习规则采用了按照误差函数梯度下降的方向进行收敛，如图 5.7 所示：

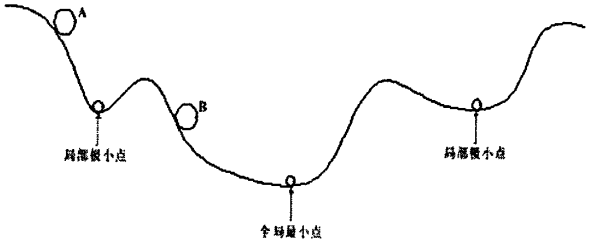


图 5.7 BP 网络的非线性空间

梯度下降学习法，有些像高山滑雪运动员，总是在寻找坡度最大的地段向下滑行，当他处于 A 点位置时，沿最大坡度路线下降，到达局部极小点，则停止滑行。如果他是从 B 点开始向下滑行，则最终他将到达全局最小点。由此可知，BP 网络的收敛依赖于学习模式的初始位置。适当改变 BP 网络中间层的单元个数、或者给每个连接权加上一个很小的随机数，都有可能使收敛过程避开局部极

小点，但是保证网络收敛于全局最小值的有效和常用的方法还是随机学习算法[55]。

5.1.2 BP 算法的程序实现

用 BP 神经网络进行字符样本训练的具体算法如下：

Step1. 选取 M 个样本组成训练样本集合$(x_{k1},x_{k2},x_{k3},...x_{kM})$；

Step2. 根据训练样本集合确定输入层节点数 M，输出层节点数 N，同时设定隐含层节点数 h；

Step3. 权值初始化为均匀分布的较小数值，比如 0-1 之间的随机数；

Step4. 设置训练参数，如学习速率 η，训练目标误差 ε，训练迭代的上限次数 $T_{\max}$。

Step5. 取训练样本中的一个输入向量 $(x_1,x_2,...x_p),(p\text{是特征维数})$ 输入网络，指定期望输出 $(t_0,t_1,...t_N)$。

Step6. 按照上述 BP 算法训练网络，计算

输出端平方误差的瞬时值 $E_j^* = \frac{1}{2}\sum_{k=0}^{M-1}(E_k - O_k)^2$

求出平方误差的均值 $Error = \frac{1}{M}\sum_{j=0}^{M-1}E_j^*$；

Step7. 用所有的训练样本反复训练，多次迭代。当满足下列之一的条件时结束训练 $Error \le \varepsilon$

或 训练步数$\ge T_{\max}$

结束训练后，返回训练后的神经网络权值；

Step8. 进行字符识别时，读入保存的网络权值，进行计算得到输出值。

5.1.3 BP 算法的流程图

传统 BP 算法流程图如图 5.8 所示：

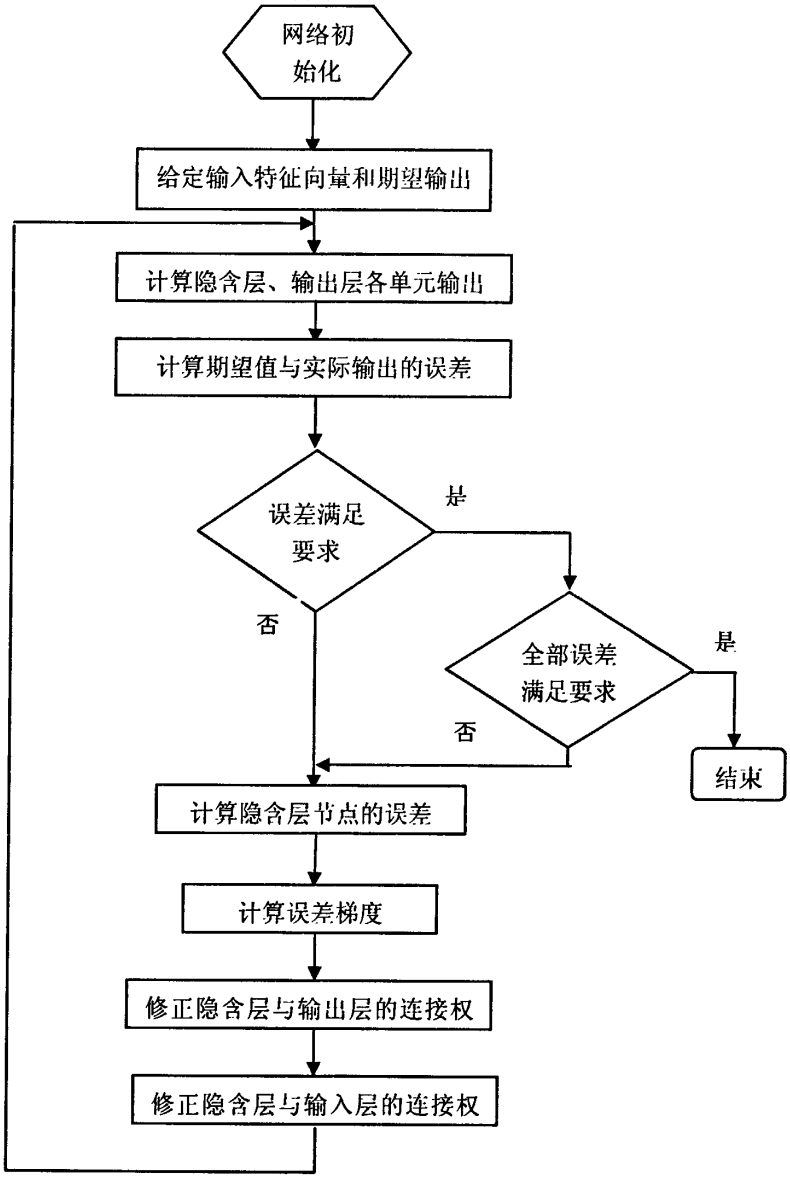


图 5.8 BP 算法流程图

5.2 车牌识别

5.2.1 车牌识别概述

汽车牌照识别(VLPR，Vechicle License Plate Recognition)是智能交通系统(ITS，Intelligent Transportation Systems)中的关键技术之一，已广泛应用于各级公路和城市道路交通管理(如道路收费系统、交通管理系统)，具有巨大的经济价值和现实意义。目前汽车牌照识别技术主要有：图像处理技术、条形码识别技术、IC 卡识别技术、射频识别技术、模式识别技术。其中 IC 卡识别技术、条形码识别技术和无线射频技术都要求预先在车身上印刷条形码或在车内安装标识卡，因此，这 3 种技术在全国范围内推广有一定的困难。而汽车牌照识别系统是基于图像处理技术的识别系统，具有更广阔的应用前景。

车牌识别系统可分为 5 大部分：视频采集、图像预处理、车牌定位、字符分割和字符识别，其流程见图 5.9：

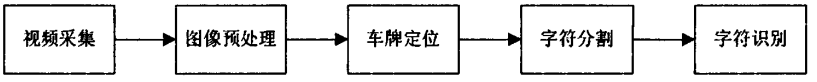


图 5.9 车牌识别系统流程图

5.2.2 图像预处理

图像的预处理一般包括灰度化、二值化、图像增强、图像校正、噪声去除、图像滤波、边缘检测等。这部分内容在第三章有详细介绍，这里简单介绍本文所使用的相关方法。

1. 灰度化

在本文中我们选用适合人眼感受的加权平均值法来灰度化彩色图像。我们使用公式(5.16)把图像化为 256 级的灰度图像，

$$Gray = (\mathrm{Re}d \times 299 + Green \times 587 + Blue \times 114)/1000 \qquad (5.16)$$

式(5.16)中 Gray 是灰度值，Red、Green、Blue 分别是像素的红、绿、蓝色分量。

2. 车牌定位

本文采用基于车牌形状特征的车牌定位方法。车牌最明显的形状特征就是矩形，且车牌长宽比固定，根据车牌形状定位车牌简单实用。图 5.10 为车牌定位后得到的车牌图片。

图 5.10 车牌定位

3. 二值化

目标和背景分离明显、灰度直方图呈明显双峰模式的图像，采用全局阈值化方法可以取得较好的效果，如 OTSU 算法和平均灰度阈值法等。由于在实际环境中光照不均匀、图像噪声干扰等原因，图像的灰度直方图分布不呈双峰性。局部阈值法常用于识别干扰比较严重，或者非均匀光照的图像，如 Bernsen 算法、Niblack 算法等。

图 5.11 二值化效果图

4. 字符分割

本文使用传统的垂直投影方法进行字符分割，字符大小归一为 15×27，效果图如下：

图 5.12 字符分割效果图

5.2.3 字符特征提取

经过上面一系列的变换，原来大小不同，分布不规律的各个字符变成了一个个大小相同，排列整齐的字符。下面就要从处理完毕的字符中，提取出最能体现字符特点的特征向量。将从训练样本中提取出的特征向量代入 BP 神经网络之中就可以对网络进行训练，将从待识别的样本中提取出的特征向量代入到训练好的 BP 神经网络中，就可以对字符进行识别。特征向量的提取方法有很多，如逐象素特征提取法，骨架特征提取法，垂直方向数据统计特征提取法，13 点特征提

取法，弧度梯度特征提取法等，根据具体情况的不同可以选用不同的提取方法。下面介绍几种简单的特征提取方法：

1. 逐象素特征提取法

这是一种最简单的特征提取方法，对图像进行逐行逐列的扫描当遇到黑色象素点时取其特征值为 1，遇到白色象素点时取其特征值为 0，这样当扫描结束以后就形成了一个维数与图像中象素点的个数相同的特征向量矩阵。

这种特征提取方法的特点是算法简单，运算速度快，可以使 BP 神经网络很快地收敛，训练效果好，缺点是适应性不强。但可以通过加大训练样本数目的方法来增强其适应性。

2. 骨架特征提取法

两幅图像由于它们线条粗细的不同，使得两幅图像差别很大，但是将它们的线条进行细化以后，统一到相同的宽度(如一个象素)，这时两幅图像的差别就不那么明显了。利用图形的骨架作为特征来进行字符识别，就使得识别有了一定的适应性。一般使用细化的方法来提取骨架。骨架特征提取的方法对于线条粗细不同的字符有一定的适应性，但是当图像一旦出现偏移就难以识别了。

3. 垂直方向数据统计特征提取法

这种特征提取方法的算法就是自左向右对图像进行逐列的扫描，统计每列的黑色象素点的个数，然后自上而下逐行扫描，统计每行的黑色象素点的个数，将统计结果作为字符的特征向量，如果字符的宽度为 w，长度为 h，则特征向量的维数就是 $w+h$。

本文采用第一种字符特征提取方法。

5.3 改进的 BP 算法用于车牌识别

5.3.1 BP 神经网络的缺陷

BP 神经网络虽然存在着这么多的优点和和应用前景，但是也存在着自己固有的缺陷。单隐层 BP 神经网络的误差是各层权值和输入样本对的函数，表达式为：$E=F(X^p,W,V,d^p)$ 函数中参数的个数为 $nm+ml$，函数为一个形状及复杂的曲面该曲面在某些区域比较平坦，误差的梯度变化很小，即使权值的调整量很大误差仍然变化很小，严重影响了收敛速度。因此训练非常缓慢且需要很大的迭代次数。在曲面上存在很多的记小点出现凸凹不平，低凹的部分就是误差曲面的极小点，曲面上分布着大量的极小点。如图 5.13 为单维时的情况：

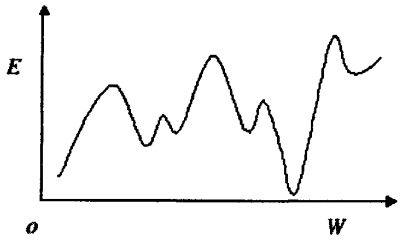


图 5.13 一维误差曲面

当用一般的 BP 算法即用梯度下降算法时，训练会陷入局部极小点而无法继续进行所以需要使训练进入全局极小点。

产生的问题有：

1. 收敛速度慢而加快收敛速度易产生振荡，平坦区域的存在也是造成收敛速度慢一个原因。

2. 多层 BP 网络可以任意逼近任何连续函数，但是从数学上看，它可归结为一非线性的梯度优化问题，因此不可避免地存在局部极小问题；误差函数容易陷入局部极小点和在误差曲面平坦区导致收敛十分慢，形成局部极小而得不到全局最优。BP 神经网络算法既然是一个非线性优化问题，这就不可避免地存在局部极小点问题。

3. 隐节点数和初始权值的选取缺乏理论指导；未考虑样本选择对学习的影响。

5.3.2 BP 神经网络的改进

1. 引入自适应学习效率

标准 BP 算法中，学习速率η在整个学习训练过程中始终保持恒定。但是如果η过大，那么权值的修改量也较大，在接近全局最小点时容易造成误差总不能小于某个小的值，从而引起振荡。反之如果η过小，这时会大大增加训练时间。所以要选择合适的学习速率不是很容易，同时对训练初期较好的η不见得适合后来的训练。这就需要一种自适应的学习率。训练刚刚开始时，η值较大，学习速度加快，将到极值点时，η逐渐变小利于收敛。自适应学习的准则是：当新误差超过旧误差一定的倍数时，学习速率将减少；否则起学习速率保持不变；当新误差小于旧误差时，学习速率将被增加。以此保证网络总是以最大可接受的学习速率进行训练。自适应公式如下：

$$\eta(t+1)=\begin{cases}1.05\eta(t) & SSE(t+1)<SSE(t)\\ 0.7\eta(t) & SSE(t+1)>1.04SSE(t)\\ \eta(t)\end{cases} \tag{5.17}$$

其中，SSE 为网络输出误差和。t 为第 n 次训练。

实验中仍采用以上的训练样本，隐层节点 25，误差上界10^{-3}，网络改进前后的收敛情况如 5.14，5.15 图所示。

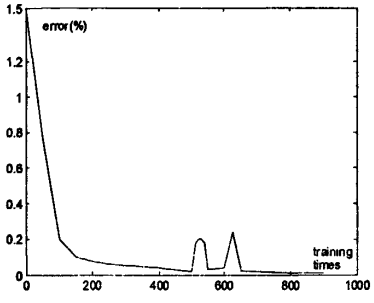


图 5.14 标准 BP 收敛曲线

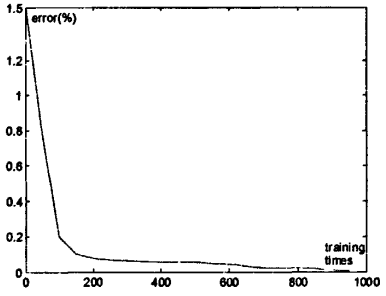


图 5.15 引入自适应率的网络收敛曲线

2. 激励函数的改进

激励函数又称激活函数或传递函数，典型的激励函数有四种：线性函数、非线性斜面函数、阶跃函数、S 型函数。传统 BP 网络采用的是 S 型函数。常见的激活函数见表 5.1：

表 5.1 常见的激活函数

函数类别	函数一般表达形式
线性函数	$f(net)=k\times net+c$ k 为放大系数，c 为位移
非线性斜面函数	$f(net)=\begin{cases}\gamma\text{或}-\gamma & net\geq\theta\text{或}\leq-\theta\\ k\times net & \lvert net\rvert<\theta\end{cases}$ $f(.)\in[-\gamma,\gamma]$
阶跃函数(阈值函数)	$f(net)=\begin{cases}\beta & net>\theta\\ -\gamma & net\leq\theta\end{cases}$ $\beta,\gamma,\theta\geq 0$
S 型函数	$f(net)=a+\dfrac{b}{1+\exp(-d\times net)}$ a,b,d为常数

传统 BP 网络采用的是 S 型函数即 Sigmoid 函数 $f(x)=\dfrac{1}{1+e^{-x}}$，由于作用函数形状固定不变，影响了网络收敛速度，通过加大陡峭度来加快网络收敛速度，即加入形态因子 λ。

激励函数变为：

$$f(x)=\frac{1}{1+e^{-\lambda x}}, x=u_j, f(u_j)=\frac{1}{1+e^{-\lambda u_j}}=\frac{1}{1+e^{-\lambda(\sum_i w_{ij}x_i-\theta_j)}} \quad (5.18)$$

其中，u_j为第j个神经元的状态值，λ为形态因子w_{ij}代表神经元的权重，θ_j则为该神经元的阈值。由于形态因子的引入，因此 Sigmoid 函数对于输入而言可以自由地进行伸缩和平移变换。

λ取不同的值时激励函数的形状如图 5.16 所示。

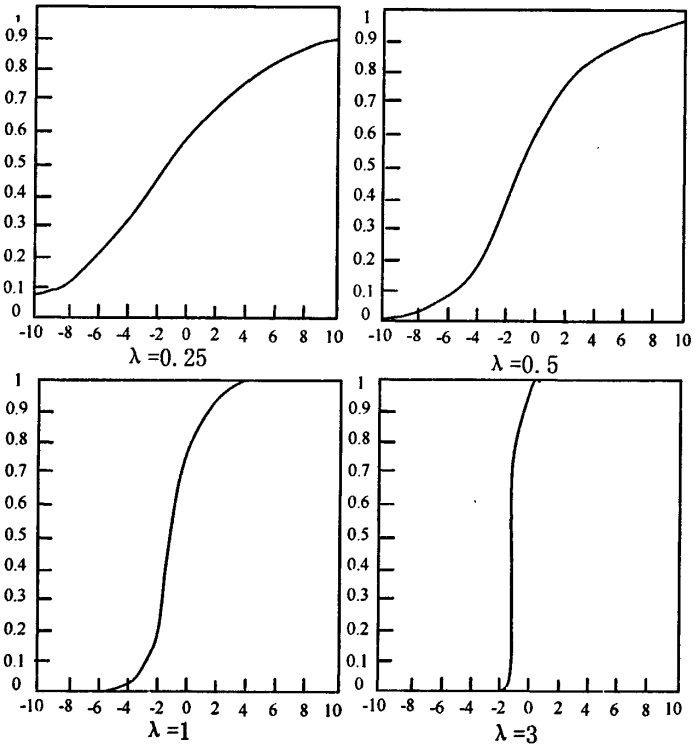


图 5.16 λ取不同的值时激励函数的形状

在设定隐节点数为 10，最小均方误差为 0.001，动量因子为 0，学习速率为 0.015，最大训练次数为 15000 的情况下，改变形态因子，用一系列的训练样本进行测试并记录其训练次数，表 5.2 给出了部分实验数据。

表 5.2 λ 取不同值时训练次数的比较

λ	样本 1	样本 2	样本 3	λ	样本 1：平均误差	样本 2：平均误差	样本 3：平均误差
0.1	8108	7500	7278	1.00	5095	4031	3683
0.25	4023	6897	3587	1.10	4866	5387	6369
0.30	3692	7052	6055	1.15	13116	3759	4830
0.35	3395	5344	2786	1.20	9712	3036	3153
0.45	2653	5313	2189	1.25	3496	4422	5694
0.50	2588	5657	3056	1.30	9493	4605	14254
0.55	2419	5652	2385	1.40	9111	4416	7220
0.60	2697	5853	2436	1.50	10628	2534	7695
0.65	2539	4227	4207	2.00	15000：0.0013	15000：0.0013	15000：0.0016
0.70	2304	5638	2677	2.50	15000：0.0018	15000：0.0018	15000：0.0028
0.80	3818	5277	2716	4.00	15000：0.0087	15000：0.0087	15000：0.0106
0.90	3813	4574	2715	5.00	15000：0.0178	15000：0.0178	15000：0.0192

出表 5.2 可以看出，对于本论文的训练样本，λ 取 0.30-0.75 之间，可以较大程度地提高网络的收敛速度。而 λ 取值过大(λ=2)，会使网络容易陷入局部极小点而跳不出来。

3. 引入动量项

标准 BP 算法实质是一种简单的最速下降静态寻优算法，在修正权值 $w(k)$ 时，只是按照 k 时刻的负梯度方式进行修正，而没有考虑到以前积累的经验，即以前时刻的梯度方向，从而常常使学习过程发生振荡，收敛缓慢。在学习公式中添加动量因子，可以降低网络对于误差曲面局部细节的敏感性，在一定程度上抑制网络陷入局部极小，如公式(5.19)所示

$$\Delta\omega_{ji}(t+1)=(1-\alpha)\eta\delta_{kj}o_{ki}+\alpha\Delta\omega_{ji}(t) \tag{5.19}$$

其中 α 是动量因子，表示以往权值修正量对目前权值修正的贡献程度，通常取值 $0<\alpha<1$。当误差处在某个极小值的一个非常小的领域内，此刻如果增大学习步长 η 和动量因子 α，就可以跳出这个局部领域，可能使误差进入到另一个极小值的领域中，实现在全局范围内搜索误差最小值。

对于动量因子 α 的取值，我们根据仿真试验进行比较后选取。我们用隐层节点 25，学习率 0.01，误差上界 10^{-2}，表 5.3 为动量因子对网络训练的影响。

表 5.3 动量因子对网络训练的影响

动量因子	训练时间(s)	迭代次数
0	210	340
0.1	195	313
0.2	190	310
0.3	176	290
0.4	154	237
0.5	128	178
0.6	107	165
0.7	80	110
0.8	58	90
0.9	20	39

从表 5.3 中可以看出，动量因子越大网络性能越优越。

4. LM(levenberg-marquardt)算法

LM 算法可以看作高斯牛顿法和梯度下降算法的一个折中算法。高斯牛顿法有二次收敛的特点，因此训练非常快。然而这种算法高度依赖于初始权值的选择，如果初始权值选择不慎，有时收敛很慢。由于初始权值的选择是困难的，实验中往往训练几次取平均情况。梯度下降法较少依赖于初始权值的选择，而且在最小点附近性能尚好，但它是线性收敛的，因此收敛速度非常缓慢。通过综合高斯牛顿法和梯度下降算法的优点，弃去各自的缺点才是研究出发点。LM 算法在不太靠近最小点时有良好的二次收敛特性，收敛快。因此在开始时“类似”于高斯牛顿法进行训练，加快训练速度。随着训练的进行，当靠近目标函数的最小点时再转换用梯度下降法。因此许多研究都使用 LM 优化方法处理各种优化问题。尽管 LM 算法优化能力很强但是直到近年来它才被用来训练神经网络，尽管 LM 算法用于神经网络尚处于早期研究阶段，但是它在训练过程中训练速度非常快的优点已很受青睐。Levenberg-Marquardt 算法类似于最优化设计中的牛顿法，其误差函数仍旧采用均方误差。设牛顿法误差函数为形式 $F(\vec{x})$，A_k 为 Hessian 矩阵，$\vec{g}_k$ 为梯度。

$$A_k=\nabla^2F(\vec{x}_k),\vec{g}_k=\nabla F(\vec{x}_k),F(\vec{x})=\sum_{i=1}^{N}v_i^2(\vec{x})=\vec{v}^T(x)\vec{v}(x) \tag{5.20}$$

$$\left[\nabla F(\vec{x})\right]_j=\frac{\partial F(\vec{x})}{\partial x_j}=2\sum_{i=1}^{N}v_i(\vec{x})\frac{\partial v_i(\vec{x})}{\partial x_j} \tag{5.21}$$

写成矩阵形式为 $\nabla F(\vec{x})=2J^T(\vec{x})V(\vec{x})$，$J^T(\vec{x})$ 为雅可比矩阵，表达式为：

$$J^T(\vec{x})=\begin{bmatrix} \frac{\partial v_1(\vec{x})}{\partial x_1} & \frac{\partial v_1(\vec{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\vec{x})}{\partial x_n} \\ \frac{\partial v_2(\vec{x})}{\partial x_1} & \frac{\partial v_2(\vec{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\vec{x})}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial v_N(\vec{x})}{\partial x_1} & \frac{\partial v_N(\vec{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\vec{x})}{\partial x_n} \end{bmatrix} \tag{5.22}$$

Hessian 矩阵的第 k，j 个元素为：

$$\left[\nabla^2 F(\vec{x})\right]_{k,j}=\frac{\partial^2 F(\vec{x})}{\partial x_k \partial x_j}=2\sum_{i=1}^{N}\left[\frac{\partial v_i(\vec{x})}{\partial x_k}\frac{\partial v_i(\vec{x})}{\partial x_j}+v_i(\vec{x})\frac{\partial^2 v_i(\vec{x})}{\partial x_k \partial x_j}\right] \tag{5.23}$$

其矩阵形式为 $\nabla^2 F(\vec{x})=2J^T(\vec{x})J(\vec{x})+2S(\vec{x})$，其中 $S(\vec{x})=\sum_{i=1}^{N}v_i(\vec{x})\nabla^2 v_i(\vec{x})$。

如果 $S(\vec{x})$ 很小可以忽略不计，则有 $\nabla^2 F(\vec{x})\approx 2J^T(\vec{x})J(\vec{x})$。

算法思想可以从下面的优化比较得到公式：

(1) 高斯-牛顿法：

$$\vec{x}_{k+1}=\vec{x}_k-[2J^T(\vec{x}_k)J(\vec{x}_k)]^{-1}2J^T(\vec{x}_k)V(\vec{x}_k)=\vec{x}_k-[J^T(\vec{x}_k)J(\vec{x}_k)]^{-1}J^T(\vec{x}_k)V(\vec{x}_k) \tag{5.24}$$

(2) Newton 法必须先求出 Hessian 阵，高斯-牛顿法只需要已知误差向量的 Jacobi 矩阵，但是 $J^T(\vec{x}_k)J(\vec{x}_k)$ 必须为非奇异的，因此为保证非奇异对其加入一个对角阵 $u_K I$，参数 u 为一个小的正数，从而也保证了 $J^T(\vec{x}_k)J(\vec{x}_k)+ul$ 为对称的。加入因子 u_K 得到所要求的 LM 优化方法：

$$\vec{x}_{k+1}=\vec{x}_k-[J^T(\vec{x}_k)J(\vec{x}_k)+u_K I]^{-1}J^T(\vec{x}_k)V(\vec{x}_k) \tag{5.25}$$

(3) 随着 u_K 的增加就变为最速下降法：

$$\vec{x}_{k+1}=\vec{x}_k-\frac{1}{u_K}J^T(\vec{x}_k)V(\vec{x}_k)=\vec{x}_k-\frac{1}{u_K}\nabla F(\vec{x}), y=[y_1,y_2,...,y_n] \tag{5.26}$$

对于一个单隐层 BP 神经网络，设输入向量为 $x=[x_1,x_2,...,x_n]$，期望输出向量为 $t=[t_1,t_2,...,t_n]$，输出向量为 $y=[y_1,y_2,...,y_n]$。

目标函数为 $\sum_{i=1}^{q}\sum_{i=1}^{n}e_n^2$，$e_{il}$ 为期望输出与实际输出的误差令 $e_{il}=t_{il}-y_{il}$，

$e=[e_{11},e_{21},...,e_{n1},e_{12},e_{22},...,e_{n2},...,e_{1q},e_{2q},...,e_{nq}]$，共有 nq 个元素，令权值向量为 $W=[w_{10}^{p},w_{11}^{p},...,w_{an}^{p},w_{10}^{h},w_{11}^{h},...,w_{ba}^{h},w_{10}^{o},...,w_{nb}^{h}]$，共有 $(m+1)a+(a+1)b+(b+1)n$ 个元素。误差向量的雅可比矩阵为：

$$\begin{bmatrix} \frac{\partial e_{11}}{\partial w_{10}^{p}} & \frac{\partial e_{11}}{\partial w_{11}^{p}} \cdots & \cdots \frac{\partial e_{11}}{\partial w_{nb}^{o}} \\ \frac{\partial e_{21}}{\partial w_{10}^{p}} & \frac{\partial e_{21}}{\partial w_{11}^{p}} \cdots & \cdots \frac{\partial e_{n1}}{\partial w_{nb}^{o}} \\ \cdots & \cdots\cdots & \cdots \\ \frac{\partial e_{nq}}{\partial w_{10}^{p}} & \frac{\partial e_{n1}}{\partial w_{11}^{p}} \cdots & \cdots \frac{\partial e_{n1}}{\partial w_{nb}^{o}} \end{bmatrix} \tag{5.27}$$

根据上述的 LM 方法再经过网络的运算，输出层，隐含层，输入层分别有

$$J_{ki}=\frac{\partial e_{kl}}{\partial w_{ij}^{o}}=\frac{\partial e_{kl}}{\partial I_{ij}^{h}}\frac{\partial I_{ij}^{h}}{\partial w_{ij}^{o}}=\delta_{ik}^{o}y_{jl}^{h},J_{ki}=\frac{\partial e_{kl}}{\partial w_{ij}^{h}}=\frac{\partial e_{kl}}{\partial I_{il}^{h}}\frac{\partial I_{il}^{h}}{\partial w_{ij}^{h}}=\delta_{ik}^{h}y_{jl}^{p},J_{ki}=\frac{\partial e_{kl}}{\partial w_{ij}^{h}}=\frac{\partial e_{kl}}{\partial I_{il}^{h}}\frac{\partial I_{il}^{h}}{\partial w_{ij}^{h}}=\delta_{ik}^{h}y_{jl}^{p} \tag{5.28}$$

参考上述 LM 优化方法以及 BP 算法的思想，最终得到权值修正方法为：

$$w_{k+1}=w_k-[J^T(w_k)J(w_k)+u_kI]^{-1}J^T(w_k)e(w_k) \tag{5.29}$$

其中 w_k 为权值，e_k 为误差函数。当 u_k 非常大时，方法类似于梯度下降算法，反之当 u_k 很小时类似于高斯牛顿法。

5.3.3 改进的 BP 神经网络识别器设计

1. BP 网络结构设计

输入、输出神经元个数：第四章已经确定了输入、输出样本的格式。输入样本为一个 15×27 的二维矩阵，对于 BP 网络而言，一个 15×27 的二维矩阵的输入和一个含有 405 个元素的一维矩阵的输入是没有区别的。因此可以确定所使用的 BP 网络的输入神经元个数为 405。输出神经元个数如果采用“n 取 1 表示法”，即输出样本为含有 54 个实数元素的一维矩阵，需将输出节点置 54，但出于实际应用考虑，网络的输出样本最好能够是二值形式的，因为这样是最便于逼近的。所以可以用 6 个输出节点，输出以二进制表示的 1—54，这个二进制数就是该输入样本应属于的类别号。另外，取 Sigmoid 函数作为激活函数时，输出值限制在 (0，1)范围内，而实际上输出值不可能达到 0 和 1。所以参照 ART 网的设计思想，

在应用中，将输入样本输入分类器。当某个输出节点的值大于0.5认为该节点输出为1，否则认为0，这些输出组成一个二进制数，就是应属的类别号。

隐含层数的确定：形象地说，正是由于隐含层的存在才使得BP网络具有识别非线性模式的能力。但是，具体取多少层合适却没有确定的规律可循。1989年Nielsen已经证明对于闭区间的连续函数，只要隐层神经元足够多就可以用一个隐层的神经网络以任意精度逼近，三层前向网络不仅能以任意精度逼近任意函数，还能以任意精度逼近其各阶导数。理论上，增加网络层数可以更进一步地降低误差，提高精度，但同时也使网络复杂化，增加了网络权值的训练时间。

针对车牌字符识别问题，我们认为使用单隐含层就可以解决，没有必要使用复杂度高的双隐含层结构，这样可以实现快速识别。

隐含层神经元个数的确定：确定隐含层神经元的个数是个复杂而重要的问题。它与具体的问题、训练样本数和输入输出层神经元的个数有关。同时，隐层神经元的过少或过多将会导致网络的学习能力不够或归纳能力下降。数目较少时，网络每次学习时间相对较短，但有可能因网络映射容量不够而不能很好学习，从而导致权值疲于来回调整而无法达到全局最小，网络训练的精度也不够高。当数目较大时，学习能力增强，但网络每次所需学习时间相对较长，所需的存储容量也随之变大。

目前常用试凑法确定隐层神经元个数，我们试验了以下两种选取方法，并利用同一组样本进行了比较，以便确定适合需要的隐层神经元的数目。这里规定k为样本数，h为隐层单元数，m为输入层节点数，n为输出层节点数。

(1)Hecht—Nielsen认为单隐层神经网络隐层节点数为$h=2m+1$；

(2)输入层节点数为m时，$h=\log_2^m$。

根据样本情况，由(1)得到$h=257$；由(2)得到$h=11$。实验中，设置训练速率$\eta=0.015$；最小均方误差上限10^{-2}；训练步数上限$T_{\max}=20000$。表5.4为隐含层节点数对网络性能影响的实验数据。图5.17、5.18分别显示了网络识别率与训练时间(包括迭代次数和迭代计算时间)随隐层节点数变化而变化的趋势图。开始时根据(1)(2)建立三个不同的网络结构进行训练，结果发现由(1)得到的隐层节点数目过多时($h=257$)网络结构过于庞大，运行很长时间后仍与期望误差相距较大。实际上理论上也证明了过多的隐层节点会增加网络结构的复杂性，降低网络

的泛化能力，产生过拟合现象。而由(2)建立的网络结构节点数相对较小($h=11$)，从图上可以看出训练时间远远高于$h=20$，识别率也低于后者，这时应该是由于隐层神经元过少，网络不强壮，识别能力不足，难以收敛易陷入局部极小值。此时我们采用了试凑法继续训练，通过对h进行加1操作增加网络结构的复杂性观察网络的性能，找到h最佳值。

表5.4 隐层节点数对网络性能的影响表

隐层节点数	训练误差	迭代计算时间(s)	迭代次数	识别率(%)
11	0.005	1420	20000	82.2
15	0.001	445	2734	86.3
20	0.001	360	1200	90.1
25	0.001	180	1980	91.7
30	0.001	347	1718	92.3
35	0.001	453	1435	91.2
40	0.001	518	1442	91.2
45	0.001	470	1098	90.4
50	0.001	550	1465	90.1
55	0.001	614	1083	89.5
60	0.001	760	1280	87.4
65	0.001	1185	1100	84.7

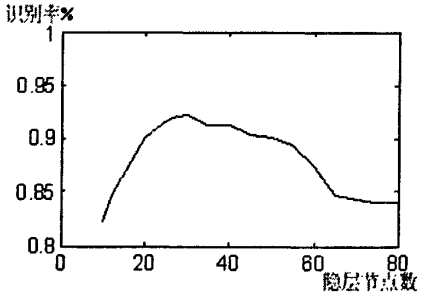


图5.17 网络识别率随隐层节点数变化曲线图

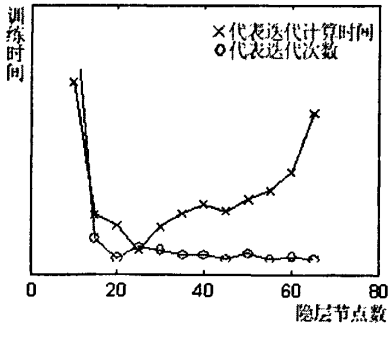


图5.18 网络训练时间随隐层节点数变化曲线图

2. 网络训练与学习

样本集选取：样本的学习就如同人记忆汉字，学习样本时的网络数值的调整就相当于记住各个汉字的形象，网络数值就是记住的内容，网络学习阶段就如同由不认识字符到认识字符反复学习的过程。神经网络分类器是按照整个特征向量的整体来记忆字符的，所以只要测试集的样本大多数特征符合曾经学习过的样本，即使存在噪声仍可以进行正确的分类。实验中为了神经网络的学习，选取各类车牌 18 个，共 270 个字符图像作为训练样本。对每个训练样本提取 405 维特征。为减少神经网络计算代码的大小，同时也为了使网络训练一开始就给各个输入分量以同等重要的地位，提高神经网络的训练速度和精度，对神经网络的输入向量进行了归一化工作，使特征向量的分量值落入[0，1]区间。归一化的变换公式如下所示：其中 x_i 代表输入或输出数据，$x_{\min}$ 代表数据变化的最小值，$x_{\max}$ 代表数据的最大值。如公式(5.30)所示：

$$\overline{x_i} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5.30}$$

权值初始化：网络权值的初始化决定了网络的训练误差从误差曲面的那一点开始，而 BP 网络一个重要的缺陷就是存在局部极小值，如图 5.4 可以看出，如果所选的初始权值使网络训练从误差曲面的 b 点开始就可以避免陷入局部极小值。所以 BP 网络自身的特点决定了权值矩阵初始化的重要性。同时，网络初始权值对于学习是否能够收敛以及训练时间的长短的关系也很大。在训练开始前，所有的权应该用一些不同的小随机数进行初始化。“不同”用来保证网络可以正常学习，如果用相同的数去初始化权矩阵，则网络将无能力学习；“小随机数”用来保证网络不会因为权过大而进入 S 型传递函数的饱和区，导致其导数非常小，而在计算权值修正公式中，因 $\delta \propto f'$，当 $f' \to 0$ 时，则有 $\delta \to 0$，这使得 $\Delta\omega \to 0$，从而使调解过程几乎停顿下来。而一般都是希望经过初始加权后的每个神经元的输出均在零点附近，这个位置不仅远离 S 型函数的两个饱和区，而且使其变化最灵敏的区域，必然使网络的学习速率较快。所以一般取初始权值在(-1，1)之间的随机数，而且要求数值比较小。

5.4 实验结果与分析

本章实验为车牌识别，对 4 种改进的 BP 算法进行性能比较，并将 BP 算法识别结果与传统方法进行性能比较。本文在.NET2008 的环境下进行实验，输入图像都是 24 位灰度图。实验界面效果如图 5.19：

图 5.19 识别效果图

5.4.1 改进后的 BP 算法性能比较

分别采用四种改进 BP 算法：动量项 BP 神经网络算法、自适应学习率动量 BP 算法、改进激励函数 BP 算法和 LM-BP 算法进行实验。训练和测试性能比较如表 5.5 所示

表 5.5 训练和测试性能比较

BP 神经网络算法	训练次数	训练时间	结束误差	测试样本识别率
动量项 BP 算法	1000	20.578000s	0.368027	训练测试均失败
自适应学习率动量 BP 算法	166	3.656000s	0.00984561	93.00%
改进激励函数 BP 算法	52	5.758000s	0.00909033	89.00%
LM-BP 算法	7	8.891000s	0.00857961	97.00%

5.4.2 BP 神经网络与模版匹配性能比较

除了使用神经网络外，还可以使用模板匹配进行分类。模板匹配一般采用最小距离分类器，将字符图像提取的特征与样本集中的字符特征进行比较，最匹配的特征所属的类就是字符所属的类别。最小距离分类器(无论用何种距离尺度)都

有着图 5.20 描述的结构，其中，x是输入特征向量，它将被分配到c个类别中的某一个$\omega_k(k=1,\ldots,c)$，这些类由各自的典型模式m_k表示。

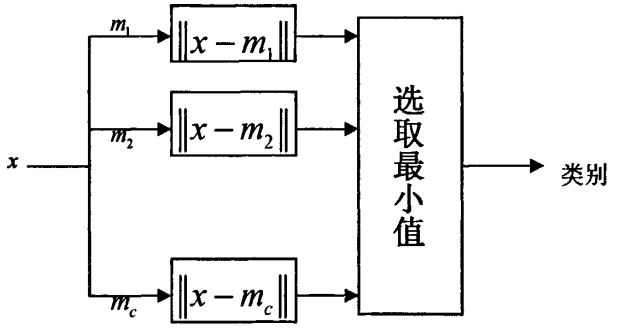


图 5.20 对于输入特征向量x运用最小距离分类器的系统图

两矢量间的距离测度的具体算法有很多种，设$x=(x_1,x_2,\ldots,x_n)'$，$y=(y_1,y_2,\ldots,y_n)'$，矢量x和y的距离记为$d(x,y)$。常见的距离测度有：

（1） 欧氏(Euclidean)距离：

$$d(x,y)=\|x-y\|=\left[\sum_{i=1}^{n}(x_i-y_i)^2\right]^{1/2} \tag{5.31}$$

（2） 绝对值距离：

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i| \tag{5.31}$$

（3） 明氏距离：

$$d(x,y)=\left[\sum_{i=1}^{n}|x_i-y_i|^m\right]^{1/m} \tag{5.32}$$

每个字符图像都可以用一系列的特征值即特征向量来表示，要判断一幅图像属于哪个类别，需要计算此图像的矢量特征与各类中心特征矢量之间的距离，距离小者对应的类被认为是该图像属于的类。在实验中，我们采用改进的欧氏距离进行分类，计算字符图像的特征向量 X 与标准样本库中的类中心特征向量的欧氏距离d_i，同时设定阈值 T，若$d_i<T$，则认为识别成功；如果$d_i\geq T$则认为识别失败。这里阈值 T 的取值显得非常关键，需要通过不断地试验，得出一个比较适中的阈值。

用距离分类器识别时，算法简单可以使识别的实时性能得到提高，但是识别的可靠性下降了。为了比较传统模式识别方法与神经网络模式识别方法的性能，我们的测试集里的每类字符由80个在白天实时环境下取得的车牌图像和80个在夜间实时环境下取得的车牌图像，用这两种不同的方法进行识别实验，识别结果如表5.6所示：

表5.6 两种分类器识别率的比较

分类器 / 样本类型	改进的欧氏距离	改进BP算法
白天取得的车牌图像	89.5%	97.2%
夜间取得的车牌图像	83.5%	93.7%

由表5.6可以看出，神经网络分类器比传统模板分类器效果好。所以对于车牌字符识别，采用最小距离法分类是不适合的。

5.5 小结

本章对BP神经网络进行研究，通过车牌识别进行实验。针对传统BP算法存在收敛速度慢、易陷入局部最小的缺点对其进行了一系列改进：引入了自适应学习速率、引入动量因子、改进激励函数以及使用LM(levenberg-marquardt)算法。通过大量实验，对各种改进方法进行了性能比较，并且将BP神经网络算法与传统模版匹配算法进行了比较。实验证明了改进方法对BP算法性能的提高，以及BP算法明显优于传统模版匹配算法。

第六章 总结与展望

6.1 总结

在这篇论文里，研究了字符识别的几个相关方法，以车牌识别、验证码识别等实际应用作为载体进行性能分析。本文研究的几个相关方法，都是在理论和方法上进行创新的基础上，再利用实际的应用去验证改进方法的优点。

字符识别之前为图像的预处理阶段，本文分析了几个重要的图像预处理方法，像灰度化、二值化、边缘检测等。针对传统方法的缺陷，本文给出了相应的改进，都取得了很好的性能，为字符识别打好基础。

形状上下文对于提取复杂字符的特征非常有用，形状上下文建模时结合了距离与角度，相对于单一的特征提取方法，形状上下文能丰富的描述出字符的特征。本文对形状上下文方法进行了大量研究，传统的形状上下文方法都是利用极坐标圆进行建模，本文给出了半极坐标圆进行建模的方法，实际应用中能有效的描述出复杂验证码的特征，对复杂验证码进行识别。

BP 神经网络作为一种广泛应用的神经网络学习算法，在车牌识别中得到了广泛应用。针对 BP 神经网络具有收敛速度慢、易陷入局部最小的特点，本文从四个方面对 BP 神经网络算法进行改进，并进行了大量的车牌识别实验，比较了各种改进的 BP 算法性能，并将本文方法与传统模板匹配方法进行性能比较，证明了本文给出的算法性能优于传统方法。

6.2 进一步的工作

本文对字符识别相关方法研究，进行了一些验证码识别、车牌识别实验。然而实际应用时，还需要考虑很多现实情况中的一些复杂因素，作为进一步的工作，需要将本文给出的创新方法与现实因素相结合去解决问题，以现实中的各种复杂条件去考验本文的创新方法。

另外，本文给出的改进形状上下文方法对于字符粘连的手写体字符识别也非常适合，在以后的工作中会进一步进行研究和实验。

参考文献

[1] 边肇祺，张学工等.模式识别[M].北京：清华大学出版社.2000.

[2] 张成海，张铎. 现代自动识别技术与应用[M].北京：清华大学出版社.2003.

[3] 厉小润.模式识别的核方法研究[D].浙江大学.2007.

[4] 文静.脱机签名识别中的关键问题研究[D].重庆大学.2009

[5] Saman Sinaie,Afshin Ghanizadeh, Siti Mariyam Shamsuddin et al. A Hybrid Detection Method Based on Fuzzy Set Theory and Cellular Learning Automata[J]. IEEE International Conference on Computational Science and Its Applications，2009，208-214.

[6] 狄文辉，王果，戴冬. 改进遗传算法在图像边缘检测中的应用[J].计算机工程与设计，2009，30(15)：3608-3611.

[7] 王志衡，吴福朝. 内积能量与边缘检测[J].计算机学报，2009，32(11):2211-2220.

[8] 李峰，刘雄飞.自适应多结构元形态学边缘检测方法[J]. 计算机仿真，2009，26(10)：265-269.

[9] J. F. Canny. A computational approach to edge detection，IEEE Trans.Pattern Anal. Machine Intell，vol. 8， no. 6， pp. 679.698， Nov. 1986.

[10] Sunita Sarangi，Dr.N.P.Rath. Performance Analysis of Fuzzy-based Canny Edge Detector[J]. IEEE International Conference on Computational Intelligence and Multimedia Applications ，2007，272-276.

[11] 古昱，汪同庆. 基于 BEMD 的 Canny 算子边缘检测算法[J].计算机工程，2009，35(11)：212-213.

[12] 吕哲，王福利，常玉清. 一种改进的 Canny 边缘检测算法[J].东北大学学报，2007，28(12)：1681-1684.

[13] 李牧，闫继红，李戈等. 自适应 Canny 算子边缘检测技术[J].哈尔滨工程大学学报，2007，28(9)：1002-1007.

[14]Serge Belongie，Jitendra Malick and Jan Puzicha. Mathching Shapes.Eighth IEEE International Conference on Computer Vision，2001.

[15] Belongie S， Malik J. Matching with Shape Contexts[D]. Berkeley，CA， USA：University of California at Berkeley， 2000.

[16]L. Basantakumar Singh. Enhanced Shape Context for Object Recognition[J]. IEEE

International Conference on Advanced Computing and Communications，2007，529-534.

[17]夏小玲，柴望.基于 Shape Context 的形状匹配方法的改进[J].东华大学学报，2009，35(1)：79-83.

[18]苏磊，马良.形状上下文在验证码识别中的应用[J].微计算机信息，2007，23(2)：252-253.

[19]唐俊，王年，梁栋等.一种结合形状上下文分析的 Laplace 谱匹配算法[J].系统仿真学报，2009，21(14)：4345-4350.

[20] 张乃尧，阎平凡. 神经网络与模糊控制. 清华大学出版社.1998，1.

[21]韩力群. 人工神经网络理论设计及应用. 北京：化学工业出版社.2002，6.

[22] Simon Haykin. 神经网络原理[M]. 北京：机械工业出版社，2004.

[23] 杨安华，彭清娥，刘光中. BP 算法固定学习率不收敛原因分析及对策[J].系统工程理论与实践，2002，12：22-25.

[24] TAKASE Haruhiko，KIKA Hidehiko，HAYASHI Terumine. A study on the simple penalty term to the error function from the viewpoint of fault tolerant training[J]. Proceeding of the 2004 International Joint conference on Neural Network，2004：1045-1050.

[25] 李杰，韩正之.神经网络的学习误差函数及泛化能力[J].华南理工大学学报，2004，30(2)：207-210.

[26] 杨斌，聂在平，夏耀先等.基于改进共轭梯度法的前馈网络快速监督学习算法[J].电子学报，2002，12.

[27] Yi-Jen Wang，Chin-Teng Lin. A second-order learning algorithm for multilayer network based on block Hessian matrix[J]. Neural Networks，1998：1607-1622.

[28] 周明月，姜文龙.基于小波变换的图像混合去噪算法[J].长春工业大学学报(自然科学版)，2008，29(2)：162—164.

[29] Joseph Vybihal，Danielle Azar. Hilditch's Algorithm for Skeletonization Software Systems. Kendall Hunt Pub.2008.

[30] Stefanelli S，Rosenfeld A. Some parallel thinning algorithm's for digital Pictures. J.ACM，1971，(18)：255—264.

[31] R. C. Casey. Moment Normalization of Handprinted Characters. IBM Journal of Research and Development . 1970，14(5):548-557.

[32] C.-L. Liu，H. Sako，H. Fujisawa. Handwritten Chinese Character Recognition：Alternatives to Nonlinear Normalization. Proceedings of the 7th Inernational Conference on Document

Analysis and Recognition. Edinburgh， Scotland， 2003：524-528.

[33] C.-L. Liu，K.Marukawa. Global Shape Normalization for Handwritten Chinese Character Recognition：A New Method. Proceedings of the 9th Inernational Workshop on Frontiers in Handwriting Recognition. 2004：300-305.

[34] C.-L. Liu，K. Marukawa. Pseudo Two-dimensional Shape Normalization Methods for Handwritten Chinese Character Recognition. Pattern Recognition. 2005，38(12)：2242-2255.

[35] Castleman K R. Digital Image Processing[M].Prentice-Hall International，Inc，1998.

[36] 张桂华，冯艳波，陆卫东.图像处理的灰度化及特征区域的获取[J].齐齐哈尔大学学报.2007，23(4)：49-52.

[37] 游达章，张建钢，甘勇.位图图像灰度化的方法及编程实现[J].广西工学院学报，2004，15(1)：23-26.

[38]Trier，Taxt. Evaluation of binarization methods for document images[M].Pattern Analysis and Machine Intelligence.1995，17(3)：312-315.

[39]Ye Xiang-yun. Fast binarization algorithm for document image[J].
Journal of Infrared and Millimeter Waves，1997，16(5)：344-350.

[40]Blayvas L，Brnckstein A，Kimmel R.Efficient Computation of Adaptive Threshold Surfaces for Image Binarization[A].Pattern Recognition，2006，39(1)：89-101.

[41]Doyle W. Operations useful for similarity-invariant pattern recognition [M].Journal of the Association for Computing Machinery ，1962，9：259-267.

[42]Otsu N. A threshold selection method from gray-level histograms[J].IEEE Transactions on Systems，Man，and Cybernetics，1979，9(1)：62-66.

[43]Pun T.A new method for gray-level picture thresholding using the entropy of histogram[M].Computer vision，graphics，and image processing，1985，29(3)：273-275.

[44]Sahoo P K，Wong A K C.A survey of thresholding techniques computer vision[J].Computer Vision，Gaphics，and Image Processing，1988，41(2)：233-260.

[45]高永英，张利，吴国威.一种基于灰度期望值的图像二值化算法[J].中国图像图形学报.1999，6：524-528.

[46] TinKu Acharya，Ajoy K.Ray. 数字图像处理原理与应用[M].清华大学出版社，2007.

[47] Ziou D.Tabbone S.Edge detection techniques-an overview.Massachusetts Institute of Technology，Artificial Intelligence Laboratory，Technical Report：AI Memo 833，1998.

[48] Saman Sinaie，Afshin Ghanizadeh，Siti Mariyam Shamsuddin et al. A Hybrid Detection Method Based on Fuzzy Set Theory and Cellular Learning Automata[J]. IEEE International Conference on Computational Science and Its Applications，2009，208-214.

[49]C Papadimitriou，K Stieglitz. Combinational Optimization，Prentice Hall，1982.

[50]文晓阳，高能，夏鲁宁等. 高效验证码识别技术与验证码分类思想[J].计算机工程，2009，35(8)：186-190.

[51]苏磊，马良. 形状上下文在验证码识别中的应用[J].微计算机信息，2007，12(2)：252-253.

[52]王虎，冯林，孙宇哲.数字验证码算法的研究和设计[J].计算机工程与应用，2007，43(32)：86-88.

[53]桑红石，傅勇，张天序等.基于标记信息的快速轮廓跟踪算法[J].华中科技大学学报，2005，33(9) ：1-4.

[54]高小榕，杨福生. 采用同伦BP算法进行多层前向神经网络的训练[J]. 计算机学报. 1996，(9)：687-694.

[55] 王伟. 人工神经网络原理—入门与应用[M]. 北京航空航天大学出版社， 1995.

攻读硕士期间发表的论文和科研情况

[1]贺强，晏立.基于 LOG 和 Canny 算子的边缘检测算法.计算机工程.(录用待刊)

[2]贺强，晏立.基于形状上下文的复杂验证码识别算法.计算机工程.(录用待刊)

致谢

在论文完成之际，首先向尊敬的导师晏立教授表示衷心的感谢！在三年的学习和研究工作中，晏老师自始至终都给予不遗余力的悉心指导和督促，不断鼓励我要勇于创新、开拓进取。晏老师崇高的品德、开明的作风、严谨的治学态度、渊博的知识和忘我的工作精神使我受益匪浅，令我终身难忘，鼓舞我不断追求上进。在此谨向晏老师致以衷心的感谢和崇高的敬意。并祝愿晏老师身体健康，工作顺利，生活幸福！

感谢我的学长张洪军、张婧颖、彭晨辉在学习上给予的指导和帮助，他们扎实的理论基础，过硬的动手能力为我树立了良好的榜样，以及在生活上热心的帮助，这一切都让我难忘。

感谢我的同门周亮、陈文在平时生活和学习上都给予我无私的帮助，正是从他们的身上让我看到了自己的不足，正是从他们的身上让我学到了很多新的东西，正是有了与他们和睦愉快的相处才使得原本枯燥、单调的求学生活不再平板。

感谢我的诸位师弟，他们友善的为人态度让我如沐春光，他们刻苦的学习精神让我佩服不已，也将激励着我永远前行。

特别要感谢我的家人，他们的支持和鼓励使我不断进步和顺利完成学业，我会不断地努力，不辜负他们对我的期望。

感谢所有关心和帮助过我的人们，祝福你们！

贺强

2010-4-24