

Reconhecimento de Caracteres Numéricos

Manuscritos

Luciana Ribeiro Veloso

Dissertação de Mestrado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba - Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Processamento da Informação

João Marques de Carvalho - Ph.D.
Orientador

Campina Grande, Paraíba, Brasil
©Luciana Ribeiro Veloso, Julho de 1998

RECONHECIMENTO DE CARACTERES NUMÉRICOS MANUSCRITOS

LUCIANA RIBEIRO VELOSO

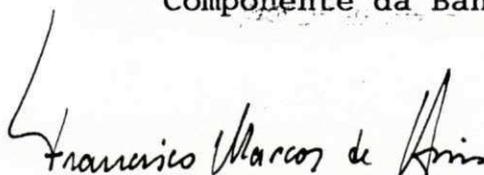
Dissertação Aprovada em 08.07.1998



PROF. JOÃO MARQUES DE CARVALHO, Ph.D., UFPB
Orientador



PROF. ELMAR UWE KURT MELCHER, Dr., UFPB
Componente da Banca



PROF. FRANCISCO MARCOS DE ASSIS, Dr., UFPB
Componente da Banca



PROF. MARCUS COSTA SAMPAIO, Dr., UFPB
Componente da Banca

CAMPINA GRANDE - PB
Julho - 1998

DIGITALIZAÇÃO:
SISTEMOTECA - UFCG

Dedicatória

Dedico este trabalho a memória de meu pai, Areoaldo, e a minha mãe, Dinalva, exemplos de luta e respeito à vida.

Agradecimentos

Agradeço, primeiramente à Deus, por poder concluir mais uma etapa na minha caminhada.

Ao prof. João Marques, por haver despertado em mim o interesse por processamento de imagens, sua valiosa orientação e dedicação, imprescindíveis em todas as fases deste trabalho.

Aos amigos Josemar e Madeiro pelo estímulo, críticas, sugestões e principalmente, por suas amizades. A minha amiga Marcia por sua ajuda e presteza e pelos vários artigos.

Aos professores Elmar e Ângelo Perkusich, por suas valiosas sugestões e críticas. Aos funcionários do DEE/UFPb: Rinaldo, Paulo Márcio, e Ângela pelo espírito de cooperação. Aos colegas de laboratório: Maria de Fátima, Aldenor, Yuri, Robson, Robson2.

Ao DEE/UFPb pela confiança depositada e à CAPES que proporcionou o suporte financeiro deste trabalho.

Resumo

O reconhecimento *off-line* de caracteres numéricos manuscritos é um sistema empregado na análise de imagens de documentos, que tem várias aplicações comerciais e industriais.

Neste trabalho, são analisados e avaliados dois sistemas de reconhecimento *off-line* de caracteres. O primeiro sistema utiliza um classificador baseado em análise sintática com decisão em árvore. O segundo consiste de um classificador baseado em redes neurais multicamadas com propagação direta, utilizando o algoritmo de *backpropagation* no treinamento da rede. No caso do primeiro sistema foram feitas modificações no classificador original, resultando em um novo algoritmo de classificação. O desempenho de cada sistema é avaliado em relação a um banco de dados obtido da *Concordia University, Montreal, Canada*.

A avaliação dos sistemas é feita utilizando os seguintes parâmetros: taxa de reconhecimento, taxa de erro, taxa de rejeição e taxa de confiabilidade, para cada caracter; além do tempo de processamento de cada etapa do sistema.

Abstract

Off-line handwritten numeral recognition systems can be used in the analysis of document images for several commercial and industrial applications.

This work concerns the analysis, implementation and evaluation of two different methods for characters recognition. The first one uses a classifier based on syntactical analysis by decision tree. The second method consists of a feed-forward backpropagation multilayer neural networks. In the case of the first system, modifications were introduced in the original classifier, leading to a new classification algorithm. A handwritten character database organized by the Concordia University in Montreal, Canada, was utilized for evaluation of both systems.

The following parameters have been used for performance evaluation: recognition rate, substitution rate, rejection rate and reliability rate, besides the processing time.

Índice

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Aspectos Gerais do Reconhecimento de Manuscritos | 1 |
| 1.2 | Sistemas <i>Off-line versus</i> Sistemas <i>On-line</i> | 3 |
| 1.3 | Identificação do Autor <i>versus</i> Determinação do Conteúdo | 3 |
| 1.4 | Sistema de Reconhecimento da Caracteres Óticos | 4 |
| 1.5 | Tipos de Sistemas de Reconhecimento | 5 |
| 1.6 | Aplicação | 5 |
| 1.7 | Objetivos da Dissertação | 6 |
| 1.8 | Organização do Texto | 7 |
| 2 | Reconhecimento de Caracteres Numéricos Manuscritos | 9 |
| 2.1 | Técnicas de reconhecimento | 10 |
| 2.2 | Sistemas de reconhecimento | 15 |
| 2.3 | Conclusão | 20 |
| 3 | Aquisição e Pré-processamento de Imagem | 21 |
| 3.1 | Aquisição das Imagens | 21 |
| 3.2 | Pré-processamento | 23 |
| 3.2.1 | Suavização | 24 |
| 3.2.2 | Normalização | 25 |
| 3.2.3 | Gradientes de Kirsch | 33 |
| 3.3 | Conclusão | 35 |
| 4 | Algoritmos Sintáticos | 36 |
| 4.1 | Extração das Características | 36 |
| 4.1.1 | Cavidades da Imagem dos Numerais | 37 |
| 4.1.2 | Seqüência de Cruzamento de um Numeral | 37 |
| 4.1.3 | Interseções com o Eixo Principal e o Eixo Secundário de um Numeral | 38 |

| | | |
|----------|--|-----------|
| 4.1.4 | Distribuição Pictorial | 38 |
| 4.2 | Algoritmo Sintático | 39 |
| 4.2.1 | Classificação dos Numerais que Possuem Cavidades Centrais . . | 40 |
| 4.2.2 | Classificação dos Numerais que não Apresentam Cavidades Centrais | 42 |
| 4.3 | Algoritmo Modificado | 44 |
| 4.3.1 | Numerais com Uma ou Duas Cavidades Centrais | 49 |
| 4.3.2 | Numerais sem Cavidades Centrais | 52 |
| 4.4 | Conclusão | 56 |
| 5 | Reconhecimento de caracteres numéricos via redes neurais | 57 |
| 5.1 | Introdução | 57 |
| 5.2 | Redes Neurais | 57 |
| 5.3 | Redes Neurais Multicamadas com Propagação Direta | 62 |
| 5.4 | Redes Neurais Recorrentes | 65 |
| 5.5 | Conclusão | 71 |
| 6 | Testes Efetuados e Resultados Obtidos | 72 |
| 6.1 | Algoritmo Sintático | 74 |
| 6.2 | Redes Neurais | 79 |
| 6.3 | Conclusão | 91 |
| 7 | Conclusão | 92 |
| 7.1 | Contribuições | 94 |
| 7.2 | Sugestões | 95 |

Lista de Tabelas

| | | |
|------|---|----|
| 6.1 | Número de amostras por classe, no banco de dados do CENPARMI . . . | 73 |
| 6.2 | Matriz de confusão - Algoritmo Sintático | 75 |
| 6.3 | Resultados do reconhecimento - Algoritmo Sintático | 75 |
| 6.4 | Matriz de confusão - Algoritmo Sintático Modificado | 76 |
| 6.5 | Resultado do reconhecimento - Algoritmo Sintático Modificado | 77 |
| 6.6 | Tempo de processamento para Algoritmo Sintático Modificado | 79 |
| 6.7 | Matriz de confusão - MLP treinada com 100 amostras | 82 |
| 6.8 | Matriz de confusão - MLP treinada com 1000 amostras | 83 |
| 6.9 | Matriz de confusão - MLP treinada com 2000 amostras | 83 |
| 6.10 | Matriz de confusão - MLP treinada com 3000 amostras | 84 |
| 6.11 | Matriz de confusão - MLP treinada com 4000 amostras | 84 |
| 6.12 | Resultados do reconhecimento - MLP treinada com 100 amostras . . . | 85 |
| 6.13 | Resultados do reconhecimento - MLP treinada com 1000 amostras . . . | 85 |
| 6.14 | Resultados do reconhecimento - MLP treinada com 2000 amostras . . . | 86 |
| 6.15 | Resultados do reconhecimento - MLP treinada com 3000 amostras . . . | 86 |
| 6.16 | Resultados do reconhecimento - MLP treinada com 4000 amostras . . . | 87 |
| 6.17 | Matriz de confusão - MLP treinada com 1000 amostras e $\varepsilon = 0,001$. . | 89 |
| 6.18 | Resultados do reconhecimento - MLP treinada com 1000 amostras e $\varepsilon = 0,001$ | 89 |
| 6.19 | Matriz de confusão - MLP treinada com 4000 amostras e $\varepsilon = 0,001$. . | 90 |
| 6.20 | Resultados do reconhecimento - MLP treinada com 4000 amostras e $\varepsilon = 0,001$ | 90 |
| 6.21 | Tempo de proc. do sistema de rec. baseado em redes neurais | 91 |
| 6.22 | Desempenhos de alguns sistemas existentes na literatura | 91 |

Lista de Figuras

| | | |
|------|--|----|
| 1.1 | Subdivisões dos sistemas de reconhecimento de caracteres manuscritos | 4 |
| 2.1 | Diagrama geral do sistema de reconhecimento de caracteres manuscritos | 9 |
| 3.1 | Amostras representativas do Banco de dados do CENPARMI | 23 |
| 3.2 | Amostras de dígitos deformados do banco de dados do CENPARMI . . | 23 |
| 3.3 | Máscaras de suavização | 24 |
| 3.4 | Vizinhança relativa ao ponto de contorno p_1 | 26 |
| 3.5 | Comparação entre a imagem do numeral 8 e seu esqueleto | 28 |
| 3.6 | Máscaras de modelos usados para o procedimento ênfase do ângulo agudo | 29 |
| 3.7 | Máscaras usadas para eliminação do efeito “escadaria” | 29 |
| 3.8 | Máscaras usadas pelo algoritmo de pruning | 30 |
| 3.9 | Comparação entre as imagens original e normalizada do numeral 8 . . . | 32 |
| 3.10 | Técnica de mapeamento direto e a propriedade de interpolação através da técnica de mapeamento reverso | 33 |
| 3.11 | Vizinhança relativa ao ponto (i, j) | 34 |
| 3.12 | Máscaras de Kirsch | 34 |
| 4.1 | Cavidades do numeral 6 | 37 |
| 4.2 | Seqüência de cruzamento do numeral 6 | 38 |
| 4.3 | Interseções com o eixo principal e o Eixo secundário do numeral 6 . . . | 39 |
| 4.4 | Fluxograma do algoritmo sintático, parte 1 | 40 |
| 4.5 | Fluxograma do algoritmo sintático, parte 2 | 41 |
| 4.6 | Comparação entre o estilo norte-americano e brasileiro de escrever o numeral 7 | 45 |
| 4.7 | Fluxograma do algoritmo sintático modificado - parte 1 | 46 |
| 4.8 | Fluxograma do algoritmo sintático modificado - parte 2 | 47 |
| 4.9 | Fluxograma do algoritmo sintático modificado - parte 3 | 48 |
| 4.10 | Fluxograma do algoritmo sintático modificado - parte 4 | 49 |
| 4.11 | Exemplos de numerais com uma cavidade central na parte inferior . . . | 50 |

| | | |
|------|--|----|
| 4.12 | Exemplos de numerais que possuem cavidades centrais na parte superior | 51 |
| 4.13 | Exemplo de resquícios no traçado dos numerais 2, 3 e 5 | 53 |
| 5.1 | Modelo do neurônio | 59 |
| 5.2 | Redes neurais multicamadas com propagação direta | 63 |
| 5.3 | Rede neural recorrente | 65 |
| 6.1 | Dígito 5 e Dígito 9 | 78 |
| 6.2 | Taxa de reconhecimento versus tamanho do conjunto de treinamento . . . | 80 |
| 6.3 | Taxa de erro versus tamanho do conjunto de treinamento | 81 |
| 6.4 | Taxa de rejeição versus tamanho do conjunto de treinamento | 81 |
| 6.5 | Taxa de confiabilidade versus tamanho do conjunto de treinamento . . | 82 |

Capítulo 1

Introdução

1.1 Aspectos Gerais do Reconhecimento de Manuscritos

O nosso século está marcado pelo início da informatização, onde os computadores fazem parte do cotidiano de milhões de pessoas. Quanto mais fácil for a comunicação entre o homem e a máquina, mais benefícios serão trazidos a todos os seus usuários.

Um dos maiores obstáculos para a integração dos computadores com outros sistemas de processamento de informações é o fato de que muitos dados estão armazenados em papéis manuscritos. Particularmente, quando se trata do público em geral, uma grande quantidade de formulários oficiais encontra-se manuscrita. Muitas vezes é desejável processar o conteúdo desses dados por máquinas e para tal, o reconhecimento de manuscritos é essencial.

Similarmente, quanto mais fácil for a interface homem-máquina, uma quantidade cada vez maior de usuários farão uso das facilidades dos computadores. Neste contexto, foram desenvolvidos o *mouse* e as telas sensíveis ao toque, e muitos trabalhos estão sendo realizados no campo do reconhecimento de voz. Entretanto, ainda há muito que se pesquisar para tornar a comunicação entre os usuários e os computadores a mais natural possível. A escrita manual é uma forma natural de comunicação entre milhões de pessoas. Pensando nisto, muitos pesquisadores e cientistas vêm investigando a utilização dos sistemas de reconhecimento de manuscritos para tornar a interface homem-máquina fácil e segura[1, 2, 3].

Fazer máquinas que podem “ler” tem sido um tema de grande interesse entre os pesquisadores desde a origem dos computadores. Entretanto, ainda não se conseguiu dotar uma máquina com a capacidade de “ler” dos seres humanos; muitos projetos

foram feitos e outros encontram-se em andamento na tentativa de construir máquinas que sejam capazes de interpretar a escrita, seja ela manuscrita ou não.

A análise de imagens de documentos é um excitante campo de pesquisa, que tem atraído o interesse dos pesquisadores de diversas áreas científicas e tecnológicas. O objetivo é não apenas o problema de reconhecimento ótico de caracteres (OCR), mas também a análise das estruturas lógicas e físicas dos documentos digitalizados, e mais genericamente, a interpretação do conteúdo dos documentos[2]. Atualmente, com os crescentes avanços tecnológicos no campo da arquitetura dos computadores e seus periféricos, tem sido possível o desenvolvimento de novos sistemas para análise de imagens de documentos. Muitos livros e artigos têm sido publicados sobre o assunto e vários encontros científicos (conferências, simpósios, *workshops*) são realizados anualmente, com o propósito de promover o intercâmbio entre pesquisadores e cientistas da área de Processamento Digital de Imagens de Documentos.

Atualmente, grandes centros de pesquisas como, por exemplo, o CEDAR (*Center of Excellence for Document Analysis and Recognition - State University of New York at Buffalo*) e o CENPARMI (*Center for Pattern Recognition and Machine Intelligence - Concordia University, Montreal, Canada*), estudam o aperfeiçoamento das técnicas utilizadas para reconhecimento de manuscritos. Porém, o reconhecimento de manuscritos continua sendo um problema difícil de ser resolvido com robustez e precisão pelos computadores, devido às inúmeras variações apresentadas pelos caracteres manuscritos. Esta variabilidade dos caracteres é devido a fatores externos e internos, como o tipo de papel e instrumento de escrita utilizados, habilidade, estilo, nível social, educação, origem, e o estado emocional do escritor. Todos estes fatores alteram a forma apresentada pelos caracteres que compõem as palavras e os números[4, 5].

O objetivo do processamento automático de documentos é reconhecer textos, gráficos e figuras em imagens digitais e extrair dos mesmos as informações desejadas. Dentro deste contexto, encontra-se o reconhecimento, identificação e verificação de textos manuscritos e/ou impressos e de assinaturas[1, 2, 4, 6, 7, 8, 9].

O reconhecimento de manuscritos tem se tornado uma das principais áreas de pesquisa, nas últimas décadas. Muitos esforços tem sido gastos na tentativa de dominar as variações e diversidades de estilo apresentados pelos documentos manuscritos. Como exemplo, cita-se pesquisas em segmentação e reconhecimento de dígitos manuscritos, segmentação de palavras, reconhecimento de palavras e reconhecimento de frases e/ou texto, dentre outras[10, 11, 12, 13].

1.2 Sistemas *Off-line* versus Sistemas *On-line*

Tradicionalmente, duas abordagens têm sido empregadas no processamento automático de manuscritos: reconhecimento *off-line* e reconhecimento *on-line*. Esta classificação está associada à forma com que os dados são adquiridos. No reconhecimento *off-line* os dados já encontram-se dispostos sobre uma folha de papel e é feita uma aquisição estática através de dispositivos do tipo *scanner*. No reconhecimento *on-line*, os dados são obtidos dinamicamente através de dispositivos acoplados ao computador, como por exemplo uma mesa digitalizadora ou uma caneta especial.

Em geral, as técnicas de reconhecimento *on-line* apresentam taxas de reconhecimento superiores aos sistemas de reconhecimento *off-line*. Este fato deve-se à natureza dinâmica da aquisição. Informações sobre a velocidade, pressão do lápis sobre o papel, pontos iniciais e finais do traçado, dentre outros, são exemplos de informações coletadas durante o processo de aquisição. Estas informações, quando usadas, facilitam a tarefa de reconhecimento dos caracteres. A localização dos pontos inicial e final do traçado, por exemplo, são de fundamental importância para os algoritmos de segmentação dos caracteres, que visam separar as palavras ou conjunto de dígitos em caracteres isolados.

Os sistemas de reconhecimento *off-line* são mais limitados, não apenas por apresentarem perda de informações dinâmicas, mais também por serem susceptíveis a ruídos introduzidos pelos dispositivos de aquisição dos dados. No entanto, o interesse nos sistemas de reconhecimento *off-line* deve-se ao fato desta abordagem preservar ao máximo a natureza da escrita e não requerer nenhum tipo de dispositivo especial de aquisição.

1.3 Identificação do Autor *versus* Determinação do Conteúdo

De acordo com o tipo de informação que se deseja extrair, os sistemas de processamento automático de manuscritos podem ser classificados em sistemas de reconhecimento, de identificação ou de verificação.

Os sistemas de reconhecimento procuram recuperar o conteúdo escrito, independentemente do seu autor. Os sistemas de verificação e identificação utilizam-se das variações da escrita peculiares ao autor do texto ou da assinatura com o objetivo de estabelecer a identidade do escritor independentemente do conteúdo do manuscrito[6, 14]. Um sistema de identificação do escritor deve determinar a identidade do escritor através da comparação de alguns atributos específicos de seus manuscritos com aqueles de todos os outros escritores armazenados em uma base de dados de referência, ou seja,

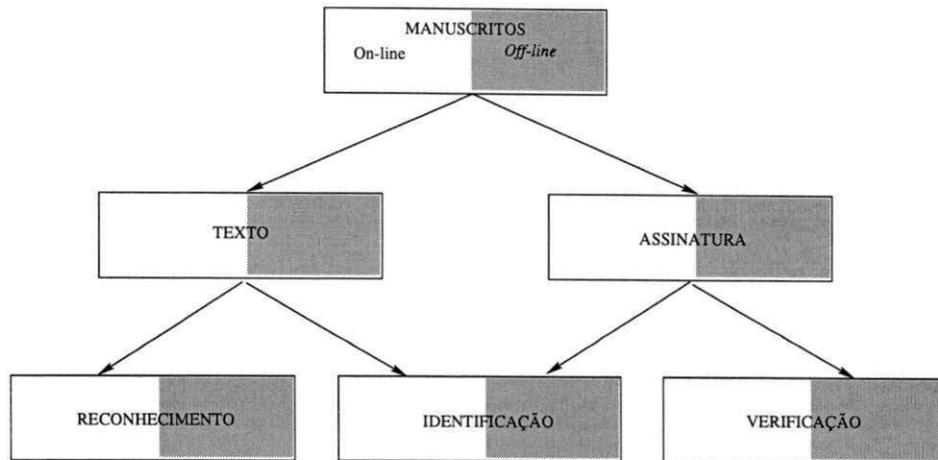


Figura 1.1: Subdivisões dos sistemas de reconhecimento de caracteres manuscritos

através de um processo de comparação de um para muitos. Um sistema de verificação, por outro lado, decide a identidade suposta de um escritor através de um processo de comparação de um-para-um, uma vez que sabe-se de antemão a que classe deverá ou não pertencer a assinatura[6].

1.4 Sistema de Reconhecimento da Caracteres Óticos

Reconhecimento *off-line* de manuscritos possui muito em comum com os sistemas de reconhecimento de caracteres óticos (OCR- Optical Character Recognition)- a leitura de caracteres impressos pelo computador. Esta aplicação recebeu muita atenção durante os anos 80, e muitas soluções foram encontradas, com pacotes comerciais disponíveis atualmente para microcomputadores, os quais podem ler uma variedade de letras com uma certa quantidade de ruído. Entretanto, em situações mais difíceis, estes pacotes comerciais ainda não apresentam desempenho satisfatório [14].

Os sistemas OCR realizam uma série de atividades. Admitindo-se que a entrada do sistema é uma imagem de uma página de texto, a primeira tarefa a ser realizada após a aquisição é a determinação da orientação do texto na página. Algumas vezes, a página é colocada no *scanner* de maneira a formar um ângulo com o eixo vertical do *scanner* diferente de 90°. Depois a imagem deve ser segmentada, primeiro em branco e preto, a seguir em linhas de texto e, finalmente, em palavras. Estas são segmentadas novamente em caracteres individuais. Após isso, a estratégia de reconhecimento é aplicada a cada caracter. Os caracteres reconhecidos são reunidos para formar as palavras e estas reunidas para formar as sentenças[15, 16, 17].

A razão pela qual as técnicas utilizadas pelos sistemas OCR não se aplicam aos sistemas de reconhecimento de manuscritos deve-se à grande diversidade da forma dos caracteres que compõem os manuscritos, bem como à dificuldade de se realizar a segmentação dos mesmos.

1.5 Tipos de Sistemas de Reconhecimento

Existem dois tipos principais de sistemas de reconhecimento de manuscritos, sistemas para reconhecimento de caracteres isolados ou dígitos e sistemas para o reconhecimento de escrita manuscrita cursiva. Os primeiros caracterizam-se pelos tipos de dados a serem utilizados: os caracteres são manuscritos, entretanto, encontram-se isolados. Os bancos de dados utilizados nestes sistemas apresentam imagens contendo apenas um dígito. Para este tipo de sistema, não se faz necessária a utilização de uma etapa de segmentação, etapa esta bastante difícil de ser realizada. É bastante complicado determinar a posição exata onde começa um caracter e termina o outro. Dentre as aplicações deste sistemas pode ser citada: o processamento automático de formulários, onde as informações são preenchidas em pequenos quadrados, cada quadrado preenchido com um único caracter.

Nos sistema de reconhecimento da escrita manuscrita cursiva, por sua vez, os caracteres que compõem a informação a ser reconhecida, geralmente, encontram-se unidos.

1.6 Aplicação

Nesta seção serão apresentadas algumas aplicações para os sistemas de reconhecimento *off-line* de manuscritos e as restrições comumente impostas aos sistemas comerciais.

O processamento de muitos documentos, tais como cheques, formulários e guias depende do uso da escrita. Portanto, o desenvolvimento de técnicas automáticas para o reconhecimento *off-line* acarretará em um impacto financeiro enorme, considerando-se o volume de informações que circulam na forma de manuscritos.

Uma importante aplicação comercial para os sistemas de reconhecimento *off-line* de manuscritos é o processamento automático de cheques bancários. Para se ter uma noção da importância dos sistemas automáticos ou semi-automáticos de processamento de documentos, convém mencionar o caso do Banco do Brasil, que ainda utiliza um processamento manual. A maior instituição financeira do país, executora do serviço de compensação de cheques e outros papéis, movimentou, apenas no primeiro semestre de 1992, cerca de 1,7 trilhões de cheques e outros documentos de compensação[9]. O

processamento automático de cheques requer, dentre outros, sistemas para reconhecimento da assinaturas, reconhecimento de caracteres numéricos e reconhecimento de palavras. Senior [14] menciona um projeto de reconhecimento de caracteres numéricos em cheques bancários, apoiado pelo governo francês, cujo objetivo é obter uma taxa de erro de 0,01%, embora permitindo que 50% dos cheques sejam rejeitados para a classificação manual.

Sistemas *off-line* capazes de reconhecer dígitos isolados possuem uma grande aplicação em agências de correios, onde são usados para controlar algum tipo de processo responsável pela distribuição automática de correspondência para seu destino apropriado. Sistemas *off-line* capazes de reconhecer dígitos isolados já foram criados e instalados em muitas empresas de correios ao redor do mundo, como parte de máquinas automáticas para classificação de correspondência[14].

Outra aplicação que está recebendo grande atenção é o processamento automático de formulários, amplamente utilizados para coletar informações em serviços públicos, como o imposto de renda, a previdência social e serviços de assistência médica.

O desenvolvimento de aparelhos de leitura para deficientes visuais constitui uma importante aplicação para os sistemas *off-line* de reconhecimento de documentos. A combinação de sistemas de reconhecimento de documentos com sistema de síntese de voz possibilita o entendimento de documentos escritos e livros por deficientes visuais.

Os sistemas comerciais de reconhecimento de caracteres numéricos devem satisfazer os seguintes requisitos:

1. Independência do escritor;
2. Capacidade de processamento de numerais de diferentes tamanhos e formas;
3. Robustez à presença de ruídos ou a variação da imagem de fundo;
4. Operação a baixa taxa de erro com uma pequena quantidade de dígitos não reconhecidos;
5. Operação a altas taxas de processamento.

1.7 Objetivos da Dissertação

O primeiro passo no processo de alfabetização é apreender o alfabeto. A habilidade para reconhecer letras ou dígitos (caracteres) é fundamental para a interpretação da linguagem escrita. Para os computadores, um caracter é simplesmente uma imagem ou um objeto a ser reconhecido.

Visando um objetivo maior ou uma aplicação específica, muitos pesquisadores desenvolveram sistemas para o reconhecimento de caracteres numéricos manuscritos. Entretanto, a comparação destes sistemas simplesmente pelo seus desempenhos relatados não pode ser realizado. Muitos sistemas são testados com base de dados contendo características diferentes como, por exemplo, resolução com que as imagens são captadas, imagens contendo diferentes tamanhos ou imagens de tamanho fixo.

Dentro do contexto de análise de imagens de documentos, mais especificamente no processamento de manuscritos, o reconhecimento *off-line* de caracteres numéricos constitui um dos mais simples problemas, não obstante este tipo de sistema possui uma grande perspectiva comercial.

O objetivo desta dissertação é investigar técnicas existentes e novas utilizadas no processo de reconhecimento *off-line* de caracteres numéricos, no contexto das abordagens de reconhecimento de padrões e processamento digital de imagens. Técnicas de suavização, normalização, esqueletização, gradientes direcionais, redes neurais, dentre outras, fazem parte do objeto de estudo deste trabalho. Alguns algoritmos de reconhecimento de caracteres numéricos foram implementados e um estudo comparativo dos mesmos foi realizado.

A seleção dos algoritmos testados foi realizada de acordo com a quantidade de informações disponíveis para a sua implementação. Foi selecionado um algoritmo de reconhecimento de caracteres utilizando análise sintática e outro algoritmo baseado em redes neurais artificiais.

1.8 Organização do Texto

O capítulo 2 faz uma revisão bibliográfica a respeito das técnicas mais utilizadas nos sistemas de reconhecimento de caracteres numéricos manuscritos.

O capítulo 3 tem como objetivo apresentar as fases de aquisição e pré-processamento utilizados pelos sistemas de reconhecimento. São apresentadas, técnicas de suavização, de esqueletização, de normalização em escala, gradientes direcionais e técnicas de *pruning*.

O capítulo 4 aborda o algoritmo de reconhecimento de caracteres numéricos manuscritos baseado em análise sintática desenvolvido por Gomes[6]. Neste capítulo, ainda, é proposto um novo algoritmo, resultante de modificações no algoritmo original analisado.

O capítulo 5 trata dos algoritmos de reconhecimento de caracteres numéricos baseados em redes neurais. Uma abordagem geral de redes neurais é realizada, onde

são apresentados a modelagem dos neurônios, os principais aspectos que caracterizam as redes neurais, arquitetura da rede e regras de treinamento. Neste capítulo, ainda, são descritos dois classificadores baseados em redes neurais: Rede neurais multicamadas com propagação direta e uma nova arquitetura de redes neurais recorrentes para reconhecimento de padrões visuais.

O capítulo 6 é dedicado à apresentação e análise comparativa dos resultados obtidos com os testes efetuados com a base de dados da Universidade de Concordia em Montreal, Canadá.

O capítulo 7 apresenta as conclusões do trabalho desenvolvido e sugestões de trabalhos futuros.

Capítulo 2

Reconhecimento de Caracteres Numéricos Manuscritos

Com base nos estudos realizados[14, 18, 10, 19, 20, 4, 21, 22], o sistema geral para o reconhecimento de caracteres manuscritos é descrito a seguir. O diagrama de blocos deste sistema é ilustrado na Figura 2.1.

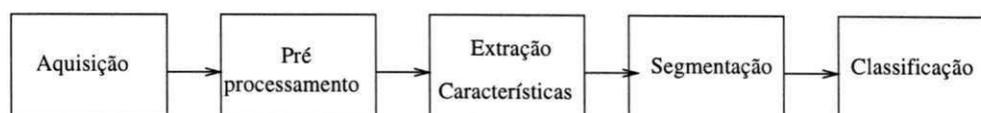


Figura 2.1: Diagrama geral do sistema de reconhecimento de caracteres manuscritos

Aquisição

É um processo através do qual a imagem é convertida em um conjunto de números, que podem ser manipulados pelo computador.

Pré-processamento

Tem o propósito de melhorar a informação pictorial, visando aumentar o desempenho dos algoritmos de reconhecimento e interpretação da imagem.

Segmentação

A segmentação é uma das fases mais críticas e complexas no processo de reconhecimento de caracteres. Seu objetivo é dividir a palavra em regiões, cada qual contendo um dígito isolado e completo[8, 7, 23, 24].

As palavras e sequências de números, que são formados por caracteres que encontram-se ligados ou unidos aos caracteres adjacentes, representam um grande desafio para os algoritmos de segmentação.

Extração das características

Tenta-se representar a imagem por um conjunto de características próprias, visando a remoção de redundâncias e redução da dimensão do conjunto de dados.

Classificação

Na classificação, é atribuído um rótulo para a imagem desconhecida, através da comparação das características extraídas com classes de padrões aprendidas pelo classificador durante o estágio de treinamento.

2.1 Técnicas de reconhecimento

Numerosos métodos têm sido desenvolvidos para realizar o reconhecimento automático de caracteres manuscritos. Estes métodos variam de acordo com as características escolhidas, o modo de extração dessas características e o esquema de classificação utilizado.

A extração de características é um importante passo na obtenção de bons desempenhos nos sistemas de reconhecimento. Porém, outras etapas do sistema devem ser também otimizadas visando obter o melhor desempenho possível e estas etapas não são independentes. A escolha dos métodos de extração das características limita ou mesmo descarta a natureza e a saída da etapa de pré-processamento. Alguns métodos de extração de características trabalham com imagens em níveis de cinza, outros trabalham com imagens binárias, com o esqueleto do caracter ou com o contorno dos numerais. Além disso, o tipo e o formato das características extraídas devem ser adequadas para o classificador escolhido. Descrição gráfica ou descrição baseada em uma gramática são adequadas para classificadores estruturais ou sintáticos. Características discretas que podem assumir um, dois ou três valores são ideais para decisões em árvore. Vetores de características com valores reais são ideais para a classificação estatística. De qualquer modo, classificadores de múltipla decisão podem utilizar um ou outro esquema de classificação em múltiplos estágios ou como classificadores paralelos, onde o resultado de cada classificação individual é utilizado para decidir a classificação final. Nestes casos, um ou mais tipos e/ou formatos de características podem ser extraídas dos caracteres de entrada.

Os métodos de classificação são divididos em duas classes distintas: métodos sintáticos e métodos estatísticos[25].

Os métodos sintáticos incluem técnicas tais como similaridade (*template matching*), medida de densidade de pontos, cálculo de momentos, características *loci* e transformações matemáticas. Na segunda categoria, métodos estatísticos, os esforços são concentrados na captura das características essenciais do formato dos numerais. Geralmente as características são extraídas do esqueleto ou do contorno dos numerais, e incluem *loops*, concavidades, junções, pontos finais e traços (*strokes*).

Suen[4], realizou uma análise comparativa dos méritos, dos principais métodos de reconhecimento de caracteres manuscritos, segundo os seguintes critérios: sensibilidade às deformações nas imagens dos caracteres e implementação prática:

1) Sensibilidade às deformações das imagens causadas pelos fatores:

- Ruído: elementos que causam descontinuidade dos segmentos de linha, lacunas, segmentos espúrios, pontos isolados, dentre outros;
- Distorções: incluem variações locais, cantos arredondados, bem como traços e inclinações para personalizar os caracteres, saliências impróprias, variações no tamanho;
- Variações de estilos: uso de diferentes formas para representar o mesmo caracter;
- Translação: movimento de todo o caracter ou de seus componentes;
- Rotação: variações na orientação dos caracteres.

Enquanto o ruído é, geralmente, introduzido devido ao processo de digitalização da imagem, superfície ou instrumento de escrita, distorções e variações de estilos são principalmente produzidas pelo escritor. Translações e rotações podem resultar de ambos, escritor e da parte mecânica do sistema (digitalizador, superfície e instrumento de escrita).

2) Os critérios de implementação prática das técnicas de reconhecimento são julgados segundo os seguintes fatores:

- Facilidade de implementação das máscaras: avalia a facilidade de se construir automaticamente as máscaras que correspondam a cada classe de caracter, com a finalidade de obter diferentes características usadas nos métodos em questão;

- Velocidade: quão rápido os caracteres são reconhecidos;
- Complexidade: determina o grau de dificuldade, custo, na implementação de um determinado método;
- Independência: diz respeito ao fato de uma única técnica ser capaz de atingir determinado desempenho sem ser necessário o uso de técnicas suplementares.

Seguindo o modelo proposto por Suen[4], será feita e apresentada a seguir uma classificação e análise comparativa das principais técnicas de reconhecimento encontradas na literatura especializada. Algumas das classes analisadas são propostas originais deste trabalho.

Similaridade e correlação (*template matching*)

Este tipo de técnica mede a similaridade entre a imagem a ser reconhecida e um padrão de referência, através da combinação e correlação entre os pontos ou grupos de pontos da imagem e do padrão de referência. Esta técnica pode ser utilizada em imagens em níveis de cinza, binárias e sobre o esqueleto.

No estágio de reconhecimento, a medida de similaridade entre cada modelo T_j e a imagem do caracter Z é calculada. O modelo T_k , que possui a mais alta medida de similaridade, é identificado e se esta medida de similaridade for maior do que um limiar especificado, o caracter é classificado como pertencente a classe k [26].

Embora simples, as técnicas de similaridade têm algumas limitações. Um modelo é capaz apenas de reconhecer caracteres de mesmo tamanho e mesma rotação e são muito vulneráveis a ruídos e pequenas variações que ocorram entre os caracteres de mesma classe. Deste modo, mais de um modelo por classe de caracter deve ser necessário para caracterizar os caracteres escritos por duas ou mais formas distintas. Portanto, vários modelos podem ser usados para cada classe de caracter, com o conseqüente aumento do tempo de processamento, desde que todo o caracter de entrada deve ser comparado com todos os modelos.

Transformações e expansões em série

Estes métodos examinam a imagem através da decomposição da mesma em séries de funções periódicas. Foram criados para diminuir a dimensão do vetor de características e extrair as características invariantes a alguma deformação global, tais como translação e rotação. As mais exploradas têm sido as séries de Fourier, a transformada de Fourier, a transformada de Walsh, a transformada de Haar e Hadmard, bem como a transformada de Karhunen-Loeve.

Shridhar e Badreldin[27] utilizaram a série de Fourier para obter as duas seqüências de coordenadas cartesianas ($x(m)$ e $y(m)$; $m = 1, 2, \dots, L$) dos elementos de contorno. Desde que os elementos de contorno dos caracteres são curvas fechadas, observa-se que $x(L) = x(1)$ e $y(L) = y(1)$. As séries de Fourier para $x(m)$ e $y(m)$ são dadas por:

$$\begin{aligned} x(m) &= \sum_{n=-\infty}^{\infty} a(n)e^{jn\omega_0 m}, \\ y(m) &= \sum_{n=-\infty}^{\infty} b(n)e^{jn\omega_0 m}, \end{aligned} \quad (2.1)$$

onde $\omega_0 = 2\pi/L$ e $a(n)$ e $b(n)$ são os coeficientes complexos de Fourier, dados por:

$$\begin{aligned} a(n) &= \frac{1}{L-1} \sum_{m=1}^{L-1} x(m)e^{-jn\omega_0 m}, \\ b(n) &= \frac{1}{L-1} \sum_{m=1}^{L-1} y(m)e^{-jn\omega_0 m}. \end{aligned} \quad (2.2)$$

Os coeficientes obtidos através do conjunto de equações 2.2 são utilizados na obtenção dos descritores de Fourier que possuem as propriedades de invariância à rotação e ao deslocamento e a independência ao tamanho do caracter. O descriptor $s(n)$ utilizado é obtido através da equação 2.3.

$$s(n) = \frac{\sqrt{[|a(n)|^2 + |b(n)|^2]}}{\sqrt{[|a(1)|^2 + |b(1)|^2]}} \quad (2.3)$$

Técnicas de transformações e expansões em série possuem grande facilidade de implementação e alta sensibilidade às distorções e variações de estilos. Por exemplo, manchas ou ruídos podem deslocar os pontos da imagem de sua posição de alinhamento, afetando seriamente as características obtidas por estas técnicas. Estes métodos provêm alguma liberdade à translação e rotação. A propriedade da invariância à rotação dos descritores de Fourier pode criar dificuldades em classificar os caracteres que são similares na forma e cuja a diferença pode ser atribuída a uma rotação e/ou a uma translação. Como exemplo, técnicas que utilizam transformações e expansões em séries geralmente, confundem-se na distinção entre as classes 2 e 5, 6 e 9 e 4 e 7.

Divisão em zonas

A matriz que contém o caracter é dividida em várias regiões, sobrepostas e não sobrepostas, denominadas zonas. As densidades de pontos nesta regiões formam as características.

Kimura e Shridhar[10], usaram a divisão em zonas das curvas do contorno dos numerais. Em cada zona, os segmentos de linhas de contorno entre pixels vizinhos foram agrupados pela orientação: horizontal(0°), vertical(90°) e duas orientações diagonais (45° e 135°). O número de segmentos de linha em cada orientação foi obtido.

Perfil de contorno

A motivação para usar perfil de contorno é que cada metade do contorno pode ser aproximada por uma função discreta de uma das variáveis espaciais x ou y . Então, as características podem ser extraídas da função discreta. Utiliza-se o perfil de contorno vertical e horizontal ou ambos.

Para construir o perfil vertical, primeiramente localiza-se os pixels extremos do contorno (os pixels com coordenada vertical mais alta e mais baixa respectivamente) na vertical. O contorno é dividido em duas partes, perfil direito e perfil esquerdo, nestes pontos. Para obter o perfil externo, para cada valor de y , seleciona-se o valor de x mais externo, para cada metade do contorno. Para obter o perfil interno, para cada valor de y , os valores mais internos de x são selecionados.

Kimura e Shridhar[10], extraiu características apenas do perfil externo. Os perfis podem ser usados eles próprios como características, como também as suas primeiras derivadas, a razão entre a altura e a máxima largura do caracter, a localização do máximo e do mínimo no perfil e a localização dos picos nas primeiras derivadas (as quais indicam discontinuidades).

Momentos

Os momentos estatísticos dos pixels em relação a um ponto de referência escolhidos no caracter, tal como o centro de gravidade ou uma outra coordenada qualquer do sistema são usados como características.

Distância e cruzamento

As características são obtidas do número de vezes que os segmentos de linha são cortados por vetores em direções específicas. Em outras palavras as distâncias medidas dos pontos que formam o caracter a um ponto específico da imagem, por exemplo, os pontos que determinam o limite do retângulo que contém o caracter.

Características *Loci*

Para todos os pixels brancos que compõem o fundo da imagem, vetores horizontais e verticais são gerados e a presença de interseções dos vetores com o caracter são usados para obter o vetor de características.

A considerável tolerância às distorções e às pequenas variações de estilos são observadas nas três últimas técnicas descritas acima. Este fato é evidenciado, principalmente nas técnicas de características *Loci*, distâncias e cruzamento. Nestes grupos de técnicas, existe uma relativa dificuldade na implementação das máscaras, devido ao grande número de possíveis variações dessas características e variações no formato dos caracteres. Por outro lado, estas técnicas provêm alta velocidade de processamento e baixa complexidade de implementação.

Segmentos de reta, linhas curvas em várias direções, pontos finais, interseções de linhas e *loops*

Devido a complexidade para extrair as características geométricas e topológicas e a grande variação de estilos apresentados pelos caracteres manuscritos, torna-se bastante difícil gerar máscaras que extraíam estas características. Entretanto, uma vez implementadas, estas técnicas permitem ao sistema uma grande velocidade de processamento e um alto índice de reconhecimento.

Realizada uma análise das principais técnicas utilizadas no reconhecimento de caracteres numéricos manuscritos são apresentadas a seguir alguns sistemas de reconhecimento existentes na literatura.

2.2 Sistemas de reconhecimento

O sistema de reconhecimento pode realizar uma simples decisão ou uma múltipla decisão. No caso dos sistemas de múltipla decisão, as decisões dos diferentes métodos de reconhecimento são combinadas em diferentes formas. Os métodos híbridos utilizam diferentes algoritmos para reconhecer as imagens de entrada e a decisão final é selecionada baseada nas decisões de cada algoritmo. Cada decisão é obtida independentemente das outras, ou seja, em paralelo. Nos métodos múltiplos estágios, por outro lado, alguns caracteres são reconhecidos por um estágio inicial de reconhecimento e as imagens restantes são passadas para os estágios subsequentes. A decisão em um estágio depende da decisão dos estágios anteriores, em um processo seqüencial.

Muita atenção tem sido dada à utilização de redes neurais na tarefa de reconhecimento de caracteres manuscritos.

Lee et al[4] propõem uma nova arquitetura de redes neurais recorrentes. A rede consiste de três camadas, onde a camada escondida é totalmente conectada a camada de entrada e cada unidade de saída é conectada a ela própria e também conectada as outras unidades de saída e as unidades escondidas. O *loop* recorrente introduzido na camada de saída garante um maior poder de discriminação e generalização no reconhecimento de padrões visuais.

Cho et al[18] descrevem um classificador de decisão múltipla, múltiplos perceptrons de multicamadas (*multiple multilayer perceptron-MLP*), treinados independentemente com características distintas (matriz normalizada, gradientes de Kirsch e características obtidas da seqüência do contorno). A classificação de um caracter é obtida levando em conta a classificação fornecida por cada rede individualmente, utilizando um esquema consensual para decidir a classificação final, pela utilização de métodos combinatórios. O método de fusão utilizado considera os diferentes desempenhos de cada rede, na combinação do resultado final da classificação, o qual é baseado na noção de lógica nebulosa (*fuzzy*).

Knerr[22] apresenta um classificador baseado em redes neurais com uma única camada de conexões. O problema do reconhecimento de caracteres manuscritos é decomposto em sub-problemas, os quais podem ser resolvidos por separadores lineares. A decomposição é realizada por um procedimento denominado de STEPNET, que consiste de três passos. No primeiro passo, realiza-se uma separação linear para cada classe utilizando um simples neurônio por classe. No segundo passo, tenta-se realizar a separação das classes que não foram separadas pelo primeiro passo, onde cada neurônio usado discrimina entre duas classes. Uma vez concluído o treinamento, as dez saídas da rede são obtidas através de operações com portas AND nos neurônios da camada de saída da rede. Os pares de classes que não encontram-se separados, podem ser separados por superfície de decisão linear por partes; implementado por uma sub-rede de simples camada que pode ser construída usando o procedimento de particionamento recursivo baseado em decisão binária por árvore. Neste tipo de procedimento a rede é construída e treinada simultânea e automaticamente sem interferência dos usuários.

Ahmede e Althorfer[28] utilizaram uma arquitetura paralela de redes neurais de única camada (*Single Layer Neural Network-SLNN*) composta de N redes separadas. Cada uma dessas redes é uma estrutura com um simples neurônio na camada de saída. A arquitetura SLNN é dividida em um nível de apresentação e um nível de decisão. No nível 1, a rede é treinada para calcular as saídas desejadas. O erro médio quadrático

é utilizado pela rede para ajustar os pesos . O algoritmo utilizado é *Window-Holf's Least Mean Square*. No nível 2, a rede calcula a soma dos erros no nível anterior. Estas somas dos erros é usada como critério básico para seleção e rejeição dos dígitos de entrada durante a fase de treinamento.

Freixinho et al[29] apresentam uma rede neural multicamadas com propagação direta - MLP, treinadas com o algoritmo de retropropagação (*Backpropagation*). A rede pode simultaneamente extrair as características e classificar os dígitos. As imagens adquiridas são suavizadas e esqueletizadas antes de serem apresentadas à entrada da rede.

Cao et al[30] propõem um sistema multi especialista para realizar a classificação dos numerais manuscritos. Neste sistema são utilizados dois vetores de características: um vetor de característica composto por histogramas direcionais e um vetor de características composto pelas características de nível de cinza extraídas pela divisão por zonas. O classificador consiste de dois tipos diferentes de rede neural. No primeiro estágio, as características do nível de cinza são aplicadas a uma camada de *incremental clustering*. Neste estágio gera-se um número de *clustering* para cada classe de caracter através de um processo de aprendizagem da rede. O segundo estágio, consiste de 45 redes neurais subclasses, cada qual correspondendo a um par de classes dos números. Cada rede neural subclasse possui três saídas, duas saídas correspondendo as subclasses do par de números e a outra como neurônio de rejeição.

Wang et al[31] descrevem um sistema de redes neurais, composto de um processo de pré-classificação e uma rede neural nebulosa. No primeiro estágio, um conjunto de características estruturais denominadas *four-zone code* são adotadas para classificar os numerais. A idéia da técnica *four-zone code* é particionar a matriz do esqueleto do caracter em quatro regiões. A cada região é atribuída um código segundo o traço padrão, o número de pontos de características e os pontos de interseção localizados na região. Os códigos atribuídos e os pontos de característica extraídos são combinados formando uma palavra código, que é usada como chave de pesquisa para pesquisar a classe correspondente no banco de dados. Embora a capacidade discriminatória destas características sejam consideradas boas, algumas subclasses ainda contém mais de um numeral. Desta forma, uma rede neural nebulosa com três camada é utilizada para realizar um classificação mais fina.

Chi et al[21] utilizam mapas auto-organizáveis (*Self Organizin Maps-SOM*) e regras nebulosas para realizar o reconhecimento de caracteres numéricos manuscritos. O algoritmo SOM é utilizado para produzir protótipos com correspondentes variâncias que são usados para determinar as regiões nebulosas e pertinência nebulosa (*fuzzy*

membership). As regras nebulosas são geradas pela aprendizagem sobre os padrões de treinamento. No estágio de reconhecimento, primeiramente são utilizadas as regras nebulosas. Os padrões que não foram reconhecidos neste estágio são reclassificados pelo classificador SOM. Os vetores de características apresentados a rede são imagens 8×8 em níveis de cinza obtidas das imagens 64×64 binárias. O número de pixels pretos em cada janela local 8×8 da imagem original é o nível de cinza do pixel correspondente na imagem 8×8 .

Diversos modelos de perceptrons de multicamadas e mapas de Kohonen auto-organizáveis são usados como poderosas ferramentas nos problemas de classificação de padrões. Suas forças encontram-se no poder discriminatório e capacidade de aprender e representar conhecimentos implícitos.

Uma vez fixados a estrutura da rede, seus pesos são ajustados através de regras de treinamento até que ótimos valores sejam obtidos. Os pesos correspondentes ao longo da rede criam uma superfície de decisão no espaço de características. Em muitos problemas práticos de reconhecimento de padrões, o classificador baseado em rede neural tende a não convergir para a solução especificada, e se convergir, o tempo necessário para a convergência pode ser proibitivo para propósitos práticos.

Shridhar e Badreleim[27] desenvolveram um sistema de decisão múltipla para o reconhecimento de caracteres numéricos manuscritos, que consiste de dois estágios de classificação. No primeiro estágio utilizam-se quinze descritores de Fourier para compor o vetor de características para cada subclasse dos caracteres. Os descritores de Fourier são obtidos através da análise de Fourier dos contornos das imagens digitalizadas. Após análise das formas de escritas dos diversos numerais, Shridhar e Badreleim constataram a existência de sub-classes para cada caracter. Estas sub-classes são decorrentes da diversidade de estilo de escrita utilizado. Uma medida de distância estatística foi adotada para medir a similaridade entre os vetores de características de referência e de teste. Devido a propriedade de invariância à rotação apresentada pelos descritores de Fourier, praticamente nenhuma distinção é feita entre as classes de numerais 2 e 5, 6 e 9 e ocasionalmente entre 4 e 7. Para solucionar este problema, Shridhar e Badreleim utilizaram um segundo classificador que utiliza descritores topológicos.

Em outros trabalhos Shridhar e Badreleim[32, 33] descrevem um método de classificação estrutural em árvore, onde a largura do caracter, a razão entre altura e máxima largura, a localização dos extremos e a descontinuidade do perfil dos caracteres são utilizados como as características locais dos numerais. Estas características locais são usadas para compor os atributos de cada primitiva, os quais possuem a forma de regras semânticas. Cada numeral e sua subclasse são descritos em termos de primitivas. O

algoritmo de reconhecimento utiliza uma estrutura em árvore para identificar a classe de um numeral desconhecido.

Suen et al[5] apresentam quatro algoritmos utilizados para realizar o reconhecimento de caracteres numéricos manuscritos. O primeiro algoritmo apresentado, decompõe o esqueleto do numeral em ramos (*branches*). Uma pré-classificação é realizada visando eliminar um ou mais elementos do conjunto de possíveis classes, utilizando para isto o conjunto de informações estruturais dos ramos. Informações sobre as posições dos pontos finais, junções do esqueleto e informações sobre regiões vazias são exemplo de algumas características estruturais utilizadas. A classificação é realizada através de onze módulos. Um destes módulos trata apenas dos dígitos que possuem apenas um ramo, utilizando as informações sobre o posicionamento dos pontos finais, curvatura e variação angular do ramo. Cada um dos dez módulos restantes é especialista no reconhecimento de uma das dez possíveis classes de numerais. Estes módulos dividem horizontalmente o numeral em duas partes iguais. A classificação é realizada conforme o casamento das partes constituinte do numeral analisado e as partes dos numerais de referência. São rejeitados os numerais que não foram reconhecidos ou aqueles que foram classificados como pertencentes a duas ou mais classes. Este tipo de sistema necessita ter o conhecimento à priori das possíveis formas de escrita dos numerais.

No segundo sistema proposto por Suen et al[5], a classificação é realizada em dois estágios. O primeiro estágio é baseado em uma decisão por árvore e destina-se a reconhecer as amostras mais legíveis. O segundo estágio é baseado no processo de relaxação. Ambos os módulos utilizam-se do mesmo conjunto de características obtidas das primitivas (segmentos de linha, polígonos convexos, e *loops*) selecionadas para representar os numerais manuscritos. As características das primitivas utilizadas são: seu tipo, sua direção, coordenada dos pontos iniciais e finais e informações sobre a sua conectividade. No módulo de relaxação é determinada a probabilidade do casamento entre as máscaras de primitivas e as primitivas de entrada. Baseado nesta probabilidade, a distância entre o exemplo e cada máscara são calculadas. Os dígitos são considerados como pertencentes a uma determinada classe se a distância obtida não exceder um determinado limiar (*threshold*).

No terceiro sistema, os dados estatísticos da ocorrência das características obtidas durante o treinamento são armazenados num banco de conhecimentos. Este banco é utilizado para deduzir a identificação do numeral. Após o exame de todas as características do numeral a ser reconhecido, o método de inferência utiliza o conjunto de interseção para operar sobre todos os correspondentes conjuntos de hipóteses e levantar o resultado final.

O quarto sistema proposto, utiliza-se das características estruturais do contorno dos numerais e de um classificador em árvore. A técnica utilizada emprega informações referentes ao número de cavidades e seu posicionamento como um primeiro indicador de a qual classe pertence o dígito.

Kimura e Shridhar[10] combinaram, utilizando diferentes formas, dois algoritmos de reconhecimento de numerais manuscritos para realizar a classificação dos numerais. O primeiro algoritmo é um classificador estatístico, que utiliza a função discriminante quadrática modificada com as características obtidas do código de cadeia de contorno dos caracteres. O segundo algoritmo é um classificador em árvore baseado nas características estruturais dos perfis direito e esquerdo dos numerais[32, 33].

Duerr et al[34] propõem uma combinação de métodos de reconhecimento estatístico e sintático. A estrutura hierárquica do sistema consiste de um classificador estatístico convencional (característica são os coeficientes de expansão em série ortonormal) e um classificador estrutural que analisa a composição topológica do contorno dos elementos direcionais. Usando simples grupos concatenados de pontos de padrões como primitivas a análise é feita através do casamento dos gráficos dos grupos com os gráficos e referência gerados. Um terceiro módulo é utilizado para reduzir o número de hipóteses geradas pelos dois primeiros módulos e é baseado na análise do contorno dos caracteres por elementos direcionais. O último módulo procura a máxima similaridade entre os protótipos sintaticamente gerados e os padrões.

2.3 Conclusão

Neste capítulo foram apresentadas classificação e análise comparativa das principais técnicas de reconhecimento de caracteres manuscritos tendo a seguir sido descritos alguns sistemas de reconhecimento de caracteres numéricos manuscritos encontrados em literatura específica.

Como pôde ser observado, existem muitas técnicas de extração de características e muitos tipos de classificadores. O primeiro passo na construção de um sistema de reconhecimento é a escolha do classificador e dos métodos de extração de características a serem utilizados.

Baseado nesta escolha, deve-se a seguir estudar os procedimentos necessários na etapa de aquisição e pré-processamento de imagens.

Capítulo 3

Aquisição e Pré-processamento de Imagem

Neste capítulo são apresentadas as fases de aquisição e pré-processamento de imagens, utilizadas nos sistemas de reconhecimento de caracteres manuscritos. Tanto a aquisição quanto o pré-processamento das imagens constituem etapas essenciais ao reconhecimento de manuscritos. A aquisição é a fase inicial de qualquer processo de reconhecimento de manuscritos.

3.1 Aquisição das Imagens

Uma imagem monocromática, ou imagem de intensidade ou simplesmente imagem, é definida como sendo uma função da intensidade de luz, uma vez que as imagens são captadas da energia radiante do ambiente por sensores. Matematicamente, uma imagem é uma função bi-dimensional contínua, ou seja, pode ser descrita por $f(x, y)$, onde x e y são variáveis contínuas que denotam as coordenadas espaciais do sistema de referência da imagem captada. O valor de f em qualquer ponto (x, y) é proporcional ao brilho (ou nível de cinza) da imagem naquele ponto. A imagem digital é discretizada (amostrada) nas coordenadas espaciais e no brilho. É considerada como uma matriz de pontos (linhas \times colunas) cujos elementos equivalem ao nível de cinza no ponto correspondente na imagem. Os elementos desta matriz são denominados elementos de imagem, elementos de quadro ou *pixels* (*picture elements*).

Os dispositivos utilizados para aquisição das imagens, nos sistemas de reconhecimento de caracteres numéricos, são divididos em duas classes principais: dispositivos *off-line* e dispositivos *on-line*. Na classe dos dispositivos *off-line*, os mais difundidos são os *scanners* óticos (*optical scanner*) e na classe dos dispositivos *on-line* são as mesas

digitalizadoras e canetas especiais que são conectadas aos computadores[35].

A validação dos sistemas de reconhecimento de caracteres é usualmente realizada através do reconhecimento de um conjunto de caracteres conhecidos, denominado conjunto de teste, contidos num banco de dados, distintos daqueles usados para compor o conjunto de treinamento do sistema. Para que os testes realizados sejam conclusivos, o conjunto de teste deve representar o universo de estilos de escrita, encontrados na aplicação desejada.

A validação dos sistemas de reconhecimento deve ser, idealmente, realizada com um banco de dados padrão. Desta forma, os resultados dos teste realizados com diferentes algoritmos de reconhecimento de caracteres podem ser facilmente comparados.

A não existência de um banco de dados padrão levou-nos à aquisição de um banco composto por uma grande diversidade de estilos de escrita, com numerais de diferentes tamanhos e diferentes larguras de traçado. Optamos pelo banco de dados do *Center for Pattern Recognition and Machine Intelligence (CENPARMI)*, da *Concordia University* em Montreal, Cánada, que consiste de 17.771 numerais desconectados, originalmente coletados dos envelopes de cartas “mortas” pelo *U.S. Postal Service* em diferentes localizações nos Estados Unidos. Os numerais deste banco de dados foram digitalizados em imagens binárias de 64 x 244 pixel para uma grade de 153 mm, dando uma resolução de aproximadamente 166 PPI (*Pixels Per Inch*)[18]. Assumindo que o código postal norte-americano seja composto por cinco dígitos, conclui-se que este banco de dados foi composto de dígitos escritos por aproximadamente 3.500 indivíduos.

A Figura 3.1 mostra algumas amostras representativas do banco de dados do CENPARMI. Diferentes tipos de estilos, diferentes tamanhos e larguras de traçado são visíveis nos caracteres apresentados.

Este banco de dados inclui, também, amostras de dígitos de difícil reconhecimento através de uma simples inspeção visual. Exemplos destes dígitos são apresentados na Figura 3.2. Na experiência realizada por Suen[5], com os 360 dígitos mais confusos deste banco de dados foram utilizados nove indivíduos: quatro estudantes e cinco especialistas em reconhecimento de caracteres. Obteve-se uma taxa de reconhecimento de 59,2% para o grupo de especialista e para o grupo de estudantes uma taxa de 46,3% apenas.

As amostras, neste banco de dados, são irregulamente distribuídas através das dez classes de numerais: por exemplo, existem apenas 716 dígitos “9” comparados aos 3697 dígitos “1”.

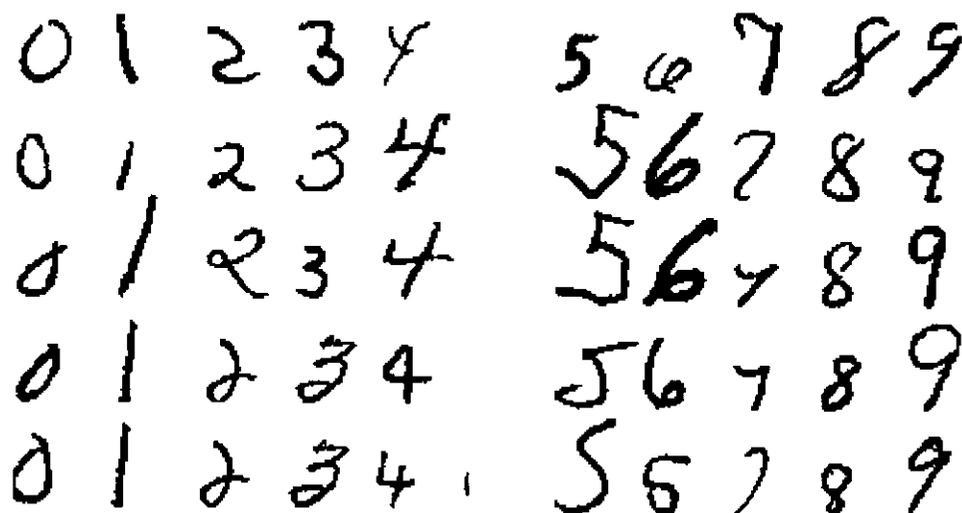


Figura 3.1: Amostras representativas do Banco de dados do CENPARMI

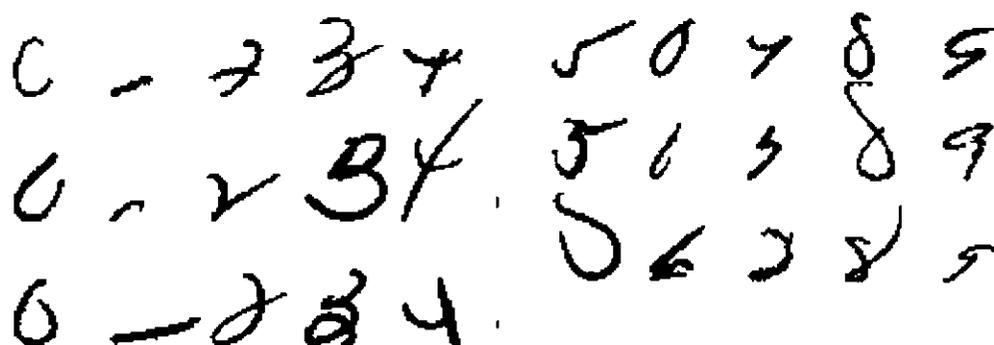


Figura 3.2: Amostras de dígitos deformados do banco de dados do CENPARMI

3.2 Pré-processamento

Os sistemas descritos neste trabalho foram projetados para reconhecerem os dígitos manuscritos obtidos através de imagens digitalizadas. Os sistemas podem, evidentemente, reconhecer os dígitos diretamente da imagem digitalizada, entretanto esta tarefa é extremamente simplificada pelo pré-processamento da imagem. Esta fase tem o propósito de melhorar a informação pictorial, visando aumentar assim, o desempenho dos algoritmos de reconhecimento e interpretação da imagem. Nesta fase a imagem de entrada é tratada de forma a retirar os ruídos e erros introduzidos pelo processo de aquisição.

3.2.1 Suavização

O processo de suavização tem como objetivo retirar da imagem original os pontos isolados (ruídos) e reduzir os picos e buracos existentes nos contornos dos caracteres. Estes ruídos são devidos à resolução com que o *scanner* capta a imagem, como também devido ao processo de escrita, ao tipo de papel utilizado e à sujeira na tela do *scanner*[31].

A necessidade de operações de suavização é inversamente proporcional à resolução utilizada no processo de aquisição da imagem, como também é função da qualidade da imagem adquirida. Quanto maior for a resolução menor a necessidade de realizar operações de suavização. Entretanto, neste caso maiores são também as necessidades de armazenamento e o tempo de processamento necessário para as operações de extração de contorno e esqueletização[6, 4].

O princípio de operação do algoritmo de suavização implementado neste trabalho é baseado no deslocamento de máscaras sobre a imagem do carácter. As máscaras básicas utilizadas neste procedimento são mostradas na Figura 3.3, além dessas, são usadas outras 17 máscaras, obtidas por rotacionamento das máscaras x_b e do espelhamento e rotacionamento das x_c e x_d de 90° , 180° e 270° . Os x , nestes elementos estruturantes, podem representar tanto pixels pretos como pixels brancos, o número 1 representa os pixels pretos e o número 0 os pixels brancos. Quando ocorrer o casamento entre qualquer uma dessas máscaras com uma janela da imagem, o elemento central da janela, definido pelas coordenadas (x, y) na imagem, tem seu valor modificado (0 para 1 ou 1 para 0).

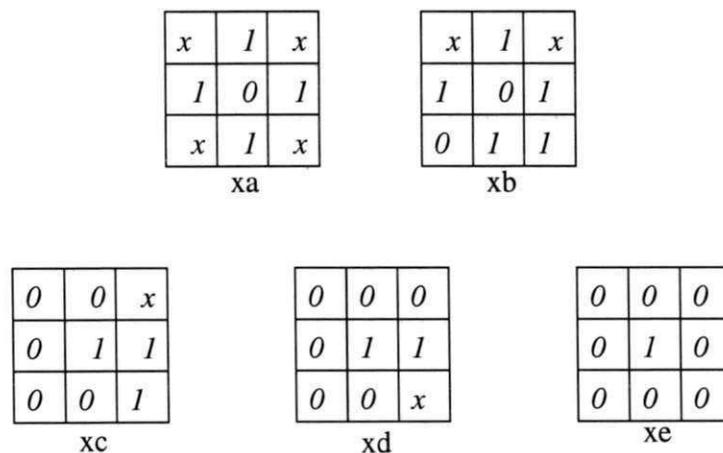


Figura 3.3: Máscaras de suavização

3.2.2 Normalização

A escrita cursiva apresenta uma grande quantidade de formas que são utilizadas para representar o mesmo caracter. Esta diversidade é causada por uma infinidade de variações na forma dos dígitos, resultado da habilidade, estilo, educação, região de origem, nível social e o estado emocional do escritor, como também de outros fatores, tais como o instrumento e superfície utilizados na escrita e os métodos de aquisição da imagem[4, 5], como anteriormente mencionado. Em aplicações de verificação ou identificação de assinaturas e da identidade do escritor, certas variações peculiares ao escritor são analisadas e utilizadas. Entretanto, para o processo de reconhecimento do conteúdo escrito, estas variações são consideradas ruídos, já que podem ser retiradas sem comprometer a identificação da palavra.

As variações comumente encontradas na escrita cursiva são[14]:

- Ângulo: Desvio dos traços que compõem os caracteres em relação à vertical.
- Largura dos traços(*strokes*): Depende de fatores tais como instrumento utilizado, pressão aplicada e papel utilizado.
- Rotação: É a mudança na orientação do caracter.
- Altura: Altura dos caracteres.

Estas variações podem ser atenuadas ou eliminadas na fase de pré-processamento da imagem. A variação na largura do traçado, por exemplo, pode ser eliminada através de algoritmos de esqueletização ou extração de contornos.

Esqueletização

O processo de esqueletização visa obter uma réplica “estruturada” da imagem que preserve as características estruturais da imagem original, porém sendo mais econômica em termos de memória [36, 37, 38]. Os algoritmos de afinamento (esqueletização) ocupam um papel central em uma larga faixa de problemas de processamento de imagens, variando desde a inspeção de circuitos impressos e reconhecimentos de imagens aéreas até a contagem de organismos em lâminas de microscópio[39].

Para os sistemas de reconhecimento de caracteres, além do objetivo acima citado, tem-se o objetivo de obter caracteres com a mesma espessura, tornando desta forma, o sistema independente das variações ocorridas nos caracteres devido a diferentes tipos de lápis e papéis, utilizados no processo de escrita manual[39].

Lam [40] comprovou que para os sistemas de reconhecimentos de caracteres manuscritos, a correta classificação dos caracteres é função muito mais do algoritmo de reconhecimento propriamente dito do que do algoritmo de esqueletização. Entretanto, distorções no esqueleto do caracter contribuem para erros na classificação.

Existem muitos algoritmos de esqueletização, entretanto o mais conhecido é o algoritmo de transformação do eixo mediano MAT (*Medial Axis Transformation*)[39]. A operação MAT em uma região R com bordas B é definida da seguinte forma. Para cada ponto P em R , deve-se encontrar seu vizinho mais próximo em B . Se P possui mais de um vizinho que satisfaz tal condição, então ele é dito pertencer ao eixo mediano (ou esqueleto) de R . O conceito de proximidade depende da distância utilizada, e portanto, os resultados de uma operação de MAT são influenciados pela escolha da medida de distância[39].

Apesar da operação de MAT produzir um esqueleto correto, a implementação direta desta operação é computacionalmente proibitiva. Tal implementação envolveria a computação das distâncias de cada ponto interior para todo o ponto de fronteira da região. Por este motivo, vários algoritmos têm sido propostos para produzir uma solução otimizada da operação MAT. Como exemplo, existem algoritmos de afinamento que interativamente removem pontos de borda de uma região sujeitos às seguintes restrições: não remover pontos terminais, não quebrar a conectividade e não causar erosão excessiva na região. Este é o caso do algoritmo proposto por Gonzales e Woods[39, 41], que foi implementado neste trabalho. Este método é aplicado a imagens binárias, onde os pontos da região de interesse possuem valor 1 e os pixels que formam o fundo da imagem tem valor igual a zero.

O método de Gonzales e Woods consiste na aplicação sucessiva de dois passos básicos sobre os pontos de contorno da região, onde pontos de contorno de uma região são aqueles pontos com valor 1 e que tenham pelo menos um dos seus oito vizinhos, mostrado na Figura 3.4 com valor igual a zero.

| | | |
|-------|-------|-------|
| p_9 | p_2 | p_3 |
| p_8 | p_1 | p_4 |
| p_7 | p_6 | p_5 |

Figura 3.4: Vizinhaça relativa ao ponto de contorno p_1

O passo 1 do algoritmo marca um ponto (p_1) de contorno para remoção se as

seguintes condições são simultaneamente verdadeiras:

$$(a) \quad 2 \leq N(p_1) \leq 6,$$

$$(b) \quad S(p_1) = 1,$$

$$(c) \quad p_2 \cap p_4 \cap p_6 = 0,$$

$$(d) \quad p_4 \cap p_6 \cap p_8 = 0.$$

$N(p_1)$ é o número de vizinhos não zero de p_1 e $S(p_1)$ é o número de transições 0-1 na seqüência ordenada: $p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9$. O passo 1 é aplicado sobre todos os pontos de borda da imagem sob consideração. Caso todas as condições (a),..., (d) sejam satisfeitas, o ponto p_1 é marcado para ser apagado. Caso contrário, o ponto não é modificado. Ao final do passo 1 todos os pontos marcados, são transformados em zeros. Isto previne mudanças na estrutura dos dados durante a execução do algoritmo.

O passo 2 do algoritmo é aplicado aos dados resultantes do passo 1 e da mesma maneira que o passo anterior marca o ponto para remoção se as seguintes condições são verdadeiras:

$$(a') \quad 2 \leq N(p_1) \leq 6,$$

$$(b') \quad S(p_1) = 1,$$

$$(c') \quad p_2 \cap p_4 \cap p_8 = 0,$$

$$(d') \quad p_2 \cap p_6 \cap p_8 = 0.$$

O algoritmo é aplicado sucessivamente sobre a imagem até que nenhum ponto possa mais ser marcado.

A condição (a) é violada quando o ponto de contorno p_1 tem apenas 1 ou 7 de seus vizinhos iguais a 1. Ter apenas um dos seus 8-vizinhos iguais a 1 significa que p_1 é um ponto final (*end point*) do esqueleto e não deverá ser apagado. No outro caso, apagar p_1 se ele tem 7 dos seus 8-vizinhos iguais a 1 causaria erosão na região, o que não é desejável. A condição (b) é violada quando é aplicada a pontos de traço (*strokes*) de espessura unitária. Esta condição previne a desconexão de pontos de um esqueleto durante o processo de afinamento. As condições (c) e (d) podem ser simplificadas para a única condição: $(p_4 \cup p_6) \cup (p_2 \cap p_8)$. Assim, de acordo com a definição de vizinhança da figura, um ponto que satisfaz a essas condições (bem como as (a) e (b)) ou é um ponto de borda leste, sul ou o único ponto de borda noroeste. Em qualquer destes casos, p_1 não pertence ao esqueleto e deve ser apagado. Do mesmo modo as condições

(c') e (d') podem ser simplificadas para a única condição: $(p_2 \cup p_8) \cup (p_4 \cap p_6)$. Estas correspondente aos pontos de borda norte, oeste ou o único ponto de borda sudeste. Da mesma forma que antes, p_1 não pertence ao esqueleto e deve ser apagado. Note que o ponto de borda nordeste tem $p_2 = 0$ e $p_4 = 0$ e desta forma satisfaz as condições (c) e (d), como também as condições (c') e (d'). O mesmo é verdade para os pontos de borda sudoeste, o qual tem $p_6 = 0$ e $p_8 = 0$.

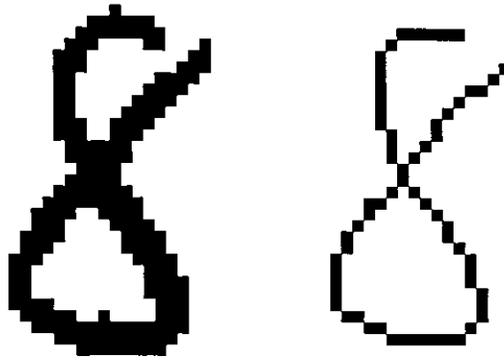


Figura 3.5: Comparação entre a imagem do numeral 8 e seu esqueleto

Existem problemas clássicos associados aos algoritmos de afinamento. O primeiro deste tipo de problema é chamado de *necking*, no qual pontos no limiar da interseção de duas linhas são unidos, formando um pequeno segmento. Desta forma, uma cauda pode ser gerada, onde antes existia um encontro de duas linhas formando um ângulo agudo. Outro exemplo, e talvez mais comum, é a criação de segmentos extras de linha junto ao esqueleto real da imagem, chamados segmentos espúrios[15].

Para lidar com os segmentos espúrios basicamente existem dois procedimentos: a utilização do método suferido Stentiford[15] e os métodos de *prunning*, a serem explicados ainda nesta seção.

Os segmentos espúrios são causados por irregularidades, geralmente pequenas cavidades ou buracos no contorno das imagens. Basicamente o algoritmo de Stentiford apaga todos os pixels pretos que possuem no máximo dois vizinhos pretos e têm o número de conectividade menor que dois[15]. O número de conectividade é uma medida do número de pixels aos quais um particular objeto pode se ligar. Uma forma de calcular é percorrer a vizinhança dos pixels na ordem $N_1, N_2, \dots, N_8, N_1$. O número de conectividade representa o número de variações da coloração (preto-branco) nesta seqüência.

Para tratar com o *necking*, utiliza-se um procedimento denominado ênfase do ângulo agudo (*Acute Angle Emphasis-AAE*), onde os pixels pretos próximos a junção entre

duas linhas são apagados quando entopem (*plug up*) o ângulo agudo. Isto é feito utilizando os modelos mostradas na Figura 3.6. O casamento com qualquer máscara marca o pixel central para remoção e causa outra interação AAE de menor severidade, usando apenas os três primeiros modelos de cada tipo.

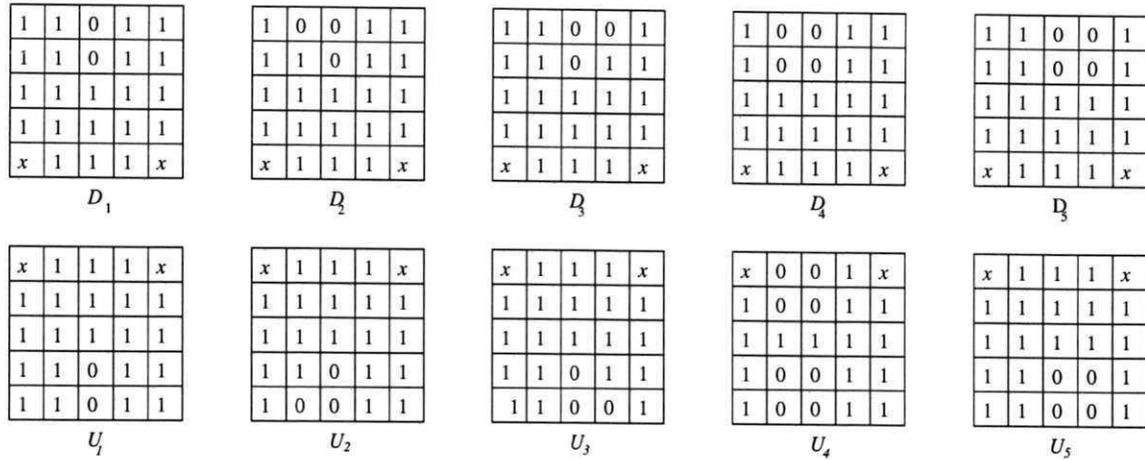


Figura 3.6: Máscaras de modelos usados para o procedimento ênfase do ângulo agudo

Algumas vezes, após o processo de afinamento, existem pixels que podem ser apagados, geralmente os que formam uma “escadaria”. Claramente, metade dos pixels que formam essas “escadarias” podem ser removidos sem afetar a forma ou a conectividade do objeto como um todo. Basicamente, o pixel central de cada uma das seguintes janelas pode ser apagado[15].

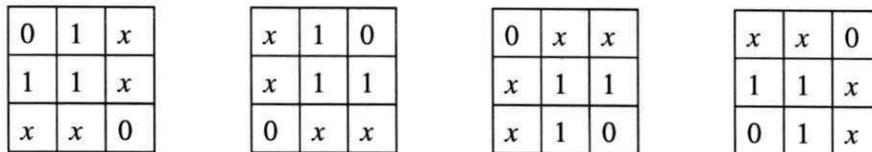


Figura 3.7: Máscaras usadas para eliminação do efeito “escadaria”

Para evitar a criação de novos buracos é adicionada uma condição extra: pelo menos um dos valores de x deve ser 0. Este procedimento é realizado após a operação do algoritmo de esqueletização.

Prunning

Os métodos de “*prunning*” são um complemento essencial aos algoritmos de afinamento e esqueletização. Estes métodos inserem muitas vezes nas imagens esqueletizadas segmentos de tamanho reduzido, denominados “pés de galinha” ou “ramos parasitas” que necessitam ser apagados por um pós-processamento[36]. Estes componentes parasitas são causados durante a erosão pela não uniformidade na composição dos traços do caracter.

O objetivo do processo de *prunning* é retirar da imagem afinada os pontos extremos. Este processo consiste em aplicar sobre a imagem afinada os oito elementos estruturantes 3×3, mostrados na Figura 3.8. Quando ocorrer o casamento do elemento estruturante com a janela 3×3 da imagem, o pixel central da janela deve ser apagado[36, 39].

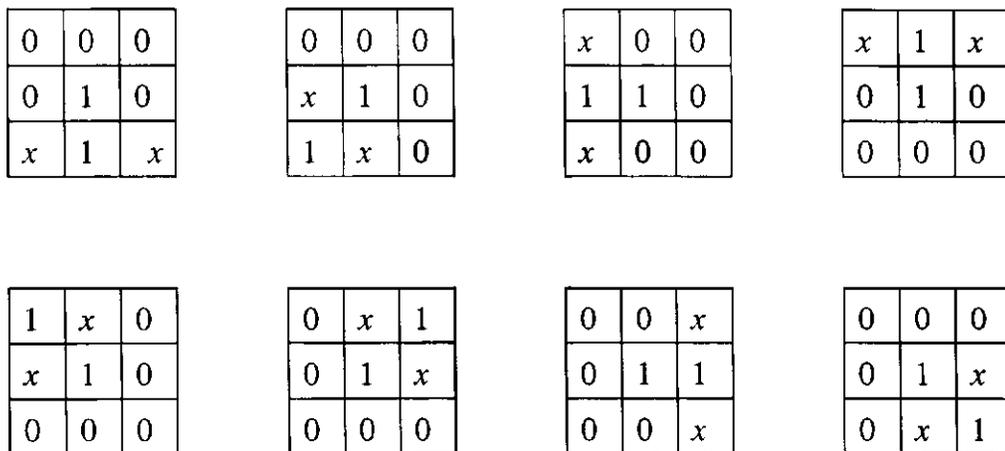


Figura 3.8: Máscaras usadas pelo algoritmo de pruning

O processo de *prunning* deve possuir um número pré-determinado de ciclos, pois a continuidade do processo pode resultar em uma redução, grande diminuição ou até mesmo em uma destruição parcial da imagem afinada[36, 9]. Na implementação deste algoritmo no presente trabalho utilizou-se o número pré-determinado de dois ciclos.

Normalização do Tamanho da Matriz da Imagem - Normalização em escala

Redes neurais com propagação direta, quando utilizadas como classificadores padrões são altamente sensíveis às variações das dimensões dos caracteres. Estudos experimentais mostraram que o procedimento de reconhecimento de caracteres manuscritos apresenta resultados insatisfatórios, quando os numerais armazenados como padrões

de treinamento encontram-se com tamanhos diferentes, em relação aos numerais a serem reconhecidos. Este comportamento tem origem na fase de treinamento, onde a principal área de interesse da rede concentra-se na região onde se encontra o padrão. Conseqüentemente, os pesos associados aos pixels de entrada que encontram-se fora desta região, decaem para zero durante a fase de treinamento. Embora, os neurônios da primeira camada realizem a soma ponderada de todos os pixels, estes pesos nulos bloqueiam as informações que podem resultar dos pixels correspondentes. Desta maneira, as imagens que possuem parte do seu padrão fora desta região, degradam drasticamente a performance da rede [42].

Portanto, para obter uma alta taxa de reconhecimento nos sistemas de classificação é útil que as imagens de entrada sejam de dimensões padronizadas, tanto na fase de treinamento quanto na fase de classificação. Entretanto, os algoritmos que compõem o pré-processamento devem ser facilmente computados, não aumentando demasiadamente a complexidade nem o tempo de processamento do sistema como um todo. Dentro deste contexto encontram-se os algoritmos de interpolação e decimação extremamente utilizados no processamento de imagens. Entretanto, devido à complexidade e tempo de processamento os mesmos não são usados no processo de tratamento das imagens para o reconhecimento de caracteres.

A escolha do algoritmo de normalização do tamanho da matriz do caracter utilizado pelo sistema de reconhecimento de caracteres manuscritos é crítica. Quando estes algoritmos são usados para diminuir as dimensões das imagens, ocorre sempre uma perda de informações. Se as informações são críticas, isto pode levar a um sério comprometimento da performance do sistema como um todo. Outro grave problema ocorre quando são usados interpoladores, pois os pixels adicionados à imagem original podem deformar a mesma, comprometendo o desempenho do sistema.

A escolha do algoritmo de normalização em escala da imagem constitui uma solução de compromisso entre a complexidade e o desempenho. Foram implementados dois algoritmos de baixa complexidade com princípios de operação similares. A diferença entre os dois encontra-se no fator de escala.

O primeiro algoritmo implementado foi baseado no método proposto por Yuceer [42], onde a correção é feita por fatores de escala diferentes para cada um dos eixos. Este tipo de correção de escala tem mostrado bons resultados em certas aplicações, como por exemplo, para padrões digitalizados com resoluções e tamanhos diferentes. A função de mapeamento para à invariância a escala das imagens é dada pela equação:

$$f_s(x_i, y_j) = f(S_x x_i, S_y y_j). \quad (3.1)$$

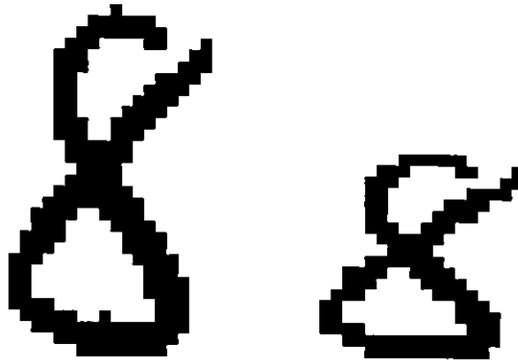


Figura 3.9: Comparação entre as imagens original e normalizada do numeral 8

A função $f(x, y)$ representa o valor do pixel nas coordenadas (x, y) da imagem original e a função $f_s(x, y)$ representa o valor do pixel nas coordenadas (x, y) da nova imagem, S_x e S_y são os fatores de escala na direção x e y , respectivamente, calculados através das seguintes equações:

$$S_x = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M f(x_i, y_j) \cdot x_i^2}{R_x \cdot \sum_{i=1}^N \sum_{j=1}^M f(x_i, y_j)}}, \quad (3.2)$$

$$S_y = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M f(x_i, y_j) \cdot y_j^2}{R_y \cdot \sum_{i=1}^N \sum_{j=1}^M f(x_i, y_j)}}, \quad (3.3)$$

onde R_x e R_y são as dimensões desejadas e N e M são as dimensões originais da imagem.

Na equação 3.1, a função de mapeamento embute a interpolação aplicada na imagem. Nesta equação os pixels de saída são mapeados pretos para seus correspondentes pixels na imagem de entrada. Isto é chamado mapeamento reverso e conduz a uma interpolação entre os pixels. Considere o caso onde a técnica de mapeamento direto é utilizada e dois pixels ativos que são adjacentes na imagem de entrada são mapeados para dois pixels não mais adjacentes. Existe então uma descontinuidade entre estes pixels na imagem de saída. Entretanto, usando o mapeamento reverso ambos os pixels separados e os pixels entre eles são mapeados pretos, de forma a manter a conectividade do padrão. Esta operação é ilustrada na Figura 3.10.

O segundo algoritmo implementado é semelhante ao primeiro. A diferença é que, neste caso, os fatores de escala são dados pelas equações:

$$S_x = \frac{N}{R_x}, \quad (3.4)$$

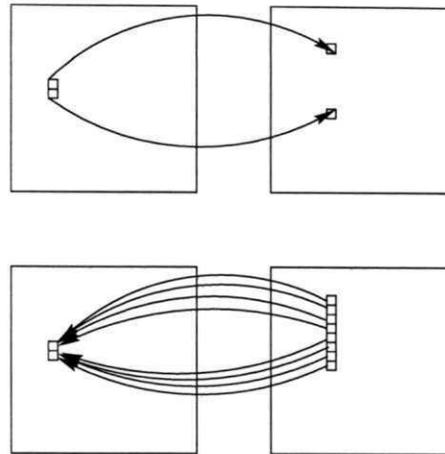


Figura 3.10: Técnica de mapeamento direto e a propriedade de interpolação através da técnica de mapeamento reverso

$$S_y = \frac{M}{Ry}. \quad (3.5)$$

3.2.3 Gradientes de Kirsch

Numerais, manuscritos ou datilografados, são essencialmente desenhos de linhas, ou seja: uma estrutura uni-dimensional no espaço bi-dimensional. Desta forma, a detecção local de segmentos de linhas parece ser um adequado método de extração das características. Para cada ponto específico da imagem, informações sobre a presença de segmentos de linha em uma dada direção são armazenadas no mapa de características. As máscaras *Kirsch* são amplamente usadas para extrair as características direcionais, neste tipo de imagens.

Kirsch definiu um algoritmo não linear de detecção de bordas que será examinado a seguir:

$$G(i, j) = \max\{1, \max_{k=0}^7 [|5S_k - 3T_k|]\}, \quad (3.6)$$

onde,

$$S_k = A_k + A_{k+1} + A_{k+2},$$

$$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7}.$$

$G(i, j)$ é o gradiente do pixel (i, j) , os índices k de A são calculados em módulo 8 e A_k ($k = 0, 1, \dots, 7$) são os níveis de cinza dos oito vizinhos do pixel (i, j) , definidos como mostra a Figura 3.11.

| | | |
|-------------------------------|----------------|----------------|
| A ₈ | A ₁ | A ₂ |
| A ₇ (<i>i,j</i>) | | A ₃ |
| A ₆ | A ₅ | A ₄ |

Figura 3.11: Vizinhança relativa ao ponto (*i, j*)

Os vetores de características direcionais horizontal, vertical, diagonal direita e diagonal esquerda são calculados pelas equações 3.7, 3.8, 3.9 e 3.10 ou através da convolução da imagem com as máscaras apresentadas pela Figura 3.12.

$$G(i, j)_H = \max(|5S_0 - 3T_0|, |5S_4 - 3T_4|) \tag{3.7}$$

$$G(i, j)_V = \max(|5S_2 - 3T_2|, |5S_6 - 3T_6|) \tag{3.8}$$

$$G(i, j)_R = \max(|5S_1 - 3T_1|, |5S_5 - 3T_5|) \tag{3.9}$$

$$G(i, j)_L = \max(|5S_3 - 3T_3|, |5S_7 - 3T_7|). \tag{3.10}$$

Estas máscaras representam uma tentativa de modelar o tipo de variação de nível de cinza, encontrada perto das bordas contendo várias orientações, ao invés de aproximações por gradiente. Por exemplo, uma larga resposta para a máscara K₅, Figura 3.12, implica bordas verticais (gradiente horizontal) no pixel correspondente ao centro da máscara. Portanto, o resultado destas máscaras assinala a presença de bordas em uma das direções vertical, horizontal e diagonal à direita ou diagonal à esquerda da imagem original.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----|----|---|----|---|----|----|----|----|---|----|----|----|----|---|----|---|----|---|---|----|----|----|----|---|---|----|----|----|---|----|----|----|---|---|----|----|----|----|
| <table border="1"><tr><td>5</td><td>5</td><td>5</td></tr><tr><td>-3</td><td>0</td><td>-3</td></tr><tr><td>-3</td><td>-3</td><td>-3</td></tr></table> <i>k</i> ₁ | 5 | 5 | 5 | -3 | 0 | -3 | -3 | -3 | -3 | <table border="1"><tr><td>-3</td><td>-3</td><td>-3</td></tr><tr><td>-3</td><td>0</td><td>-3</td></tr><tr><td>5</td><td>5</td><td>5</td></tr></table> <i>k</i> ₂ | -3 | -3 | -3 | -3 | 0 | -3 | 5 | 5 | 5 | <table border="1"><tr><td>-3</td><td>5</td><td>5</td></tr><tr><td>-3</td><td>0</td><td>5</td></tr><tr><td>-3</td><td>-3</td><td>-3</td></tr></table> <i>k</i> ₃ | -3 | 5 | 5 | -3 | 0 | 5 | -3 | -3 | -3 | <table border="1"><tr><td>-3</td><td>-3</td><td>-3</td></tr><tr><td>5</td><td>0</td><td>-3</td></tr><tr><td>5</td><td>5</td><td>-3</td></tr></table> <i>k</i> ₄ | -3 | -3 | -3 | 5 | 0 | -3 | 5 | 5 | -3 |
| 5 | 5 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 0 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 0 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 5 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 5 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 0 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 5 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"><tr><td>-3</td><td>-3</td><td>5</td></tr><tr><td>-3</td><td>0</td><td>5</td></tr><tr><td>-3</td><td>-3</td><td>5</td></tr></table> <i>k</i> ₅ | -3 | -3 | 5 | -3 | 0 | 5 | -3 | -3 | 5 | <table border="1"><tr><td>5</td><td>-3</td><td>-3</td></tr><tr><td>5</td><td>0</td><td>-3</td></tr><tr><td>5</td><td>-3</td><td>5</td></tr></table> <i>k</i> ₆ | 5 | -3 | -3 | 5 | 0 | -3 | 5 | -3 | 5 | <table border="1"><tr><td>-3</td><td>-3</td><td>-3</td></tr><tr><td>-3</td><td>0</td><td>5</td></tr><tr><td>-3</td><td>5</td><td>5</td></tr></table> <i>k</i> ₇ | -3 | -3 | -3 | -3 | 0 | 5 | -3 | 5 | 5 | <table border="1"><tr><td>5</td><td>5</td><td>-3</td></tr><tr><td>5</td><td>0</td><td>-3</td></tr><tr><td>-3</td><td>-3</td><td>-3</td></tr></table> <i>k</i> ₈ | 5 | 5 | -3 | 5 | 0 | -3 | -3 | -3 | -3 |
| -3 | -3 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 0 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | -3 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 0 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | 5 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 5 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 0 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| -3 | -3 | -3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figura 3.12: Máscaras de Kirsch

3.3 Conclusão

Neste capítulo foram feitas algumas considerações sobre uma das etapas principais dos sistemas de reconhecimento de caracteres manuscritos, a aquisição de imagens. A seguir, foi apresentado o banco de dados utilizado neste trabalho para avaliar os desempenhos dos sistemas escolhidos para implementação.

Na seqüência, foram apresentados os algoritmos de pré-processamento utilizados nos sistemas de reconhecimento de caracteres: suavização, esqueletização, normalização em escala e gradientes de *Kirsch*.

Os capítulos seguintes são dedicados aos sistemas de reconhecimento baseados em análise sintática e em redes neurais.

Capítulo 4

Algoritmos Sintáticos

No reconhecimento estrutural ou sintático, as características dos elementos, em termos de suas partes constituintes, propriedades e relacionamentos entre as partes, são representadas sintaticamente.

O estudo desenvolvido neste capítulo é baseado, em grande parte, no algoritmo de reconhecimento por análise sintática proposto por Gomes[6], chamado a partir de agora de “algoritmo sintático”. Este método classifica os numerais manuscritos conforme as características extraídas de sua topologia.

Visando proporcionar um bom entendimento do algoritmo sintático é feito, inicialmente, um estudo das características utilizadas pelo mesmo e suas formas de extração. A seguir, é apresentado o algoritmo propriamente dito. Por último, é apresentado o algoritmo de classificação de caracteres numéricos manuscritos desenvolvido durante o decorrer desta dissertação, baseado no algoritmo sintático.

4.1 Extração das Características

No reconhecimento de caracteres pelo método de análise sintática, a classificação dos padrões é realizada através da análise de um conjunto de atributos que descrevem aspectos importantes da geometria e da topologia do desenho do caracter. Este conjunto de atributos deve possibilitar a representação e discriminação do padrão a ser reconhecido em relação aos demais padrões.

Neste tipo de algoritmo, o numeral é representado pela matriz correspondente ao menor retângulo envolvente, ou MRE. O conjunto de atributos utilizados para o reconhecimento via análise sintática implementado envolve oito parâmetros, relacionados a seguir:

- Número de cavidades centrais;

- Número de cavidades à direita;
- Número de cavidades à esquerda;
- Posição da cavidade no numeral (Superior ou Inferior);
- Seqüência de cruzamento do numeral;
- Número de intersecções com o eixo principal;
- Número de intersecções com o eixo secundário;
- Posições das intersecções dos eixos principal e secundário;
- Distribuição pictorial.

4.1.1 Cavidades da Imagem dos Numerais

Os quatro primeiros parâmetros do conjunto de atributos são obtidos através da contagem e identificação das posições das cavidades no numeral a ser reconhecido. São denominadas cavidades, as regiões delimitadas pelo traçado do numeral e pelas bordas do menor retângulo que enquadra o numeral (MRE).



Figura 4.1: Cavidades do numeral 6

As cavidades centrais do numeral são cavidades cujos pixels não pertencem as bordas da matriz do numeral. As cavidades à direita são aquelas cavidades cujos pixels pertencem à última coluna da matriz e as cavidades à esquerda são as cavidades cujos pixels pertencem à primeira coluna da matriz. Por exemplo, para a Figura 4.1 tem-se uma cavidade central, duas cavidades à direita e duas cavidades à esquerda.

4.1.2 Seqüência de Cruzamento de um Numeral

A Figura 4.2 mostra a seqüência de cruzamentos horizontal e vertical localizadas no lado direito e na base da imagem do numeral 6, respectivamente. Cada elemento desta seqüência corresponde à quantidade de blocos, constituídos de um ou mais pixels pretos

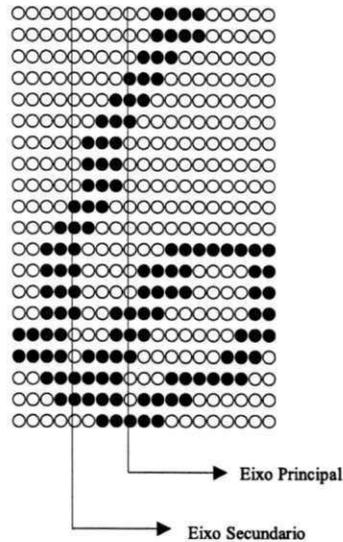


Figura 4.3: Interseções com o eixo principal e o Eixo secundário do numeral 6

classe (uma classe representa um número de 0 à 9). Como exemplo de características extraídas da distribuição pictorial, tem-se o número de pixels pretos existentes na última linha da matriz do carácter ou a existência de pixels pretos à direita do eixo principal entre a primeira e a segunda intersecções deste com os pixels pretos que formam o traçado do carácter.

4.2 Algoritmo Sintático

O algoritmo de reconhecimento de numerais por análise sintática, desenvolvido por Gomes[6] classifica os numerais de acordo com as características extraídas de sua topologia e distribuição pictorial. Os numerais são classificados de acordo com a existência ou não de cavidades centrais. De maneira a facilitar o entendimento deste algoritmo, o seu fluxograma apresenta-se dividido em duas partes, como ilustrado na Figuras 4.4 e 4.5. A primeira dedica-se à classificação dos numerais que apresentam cavidades centrais, a segunda parte dedica-se à análise dos dígitos que não possuem cavidades centrais.

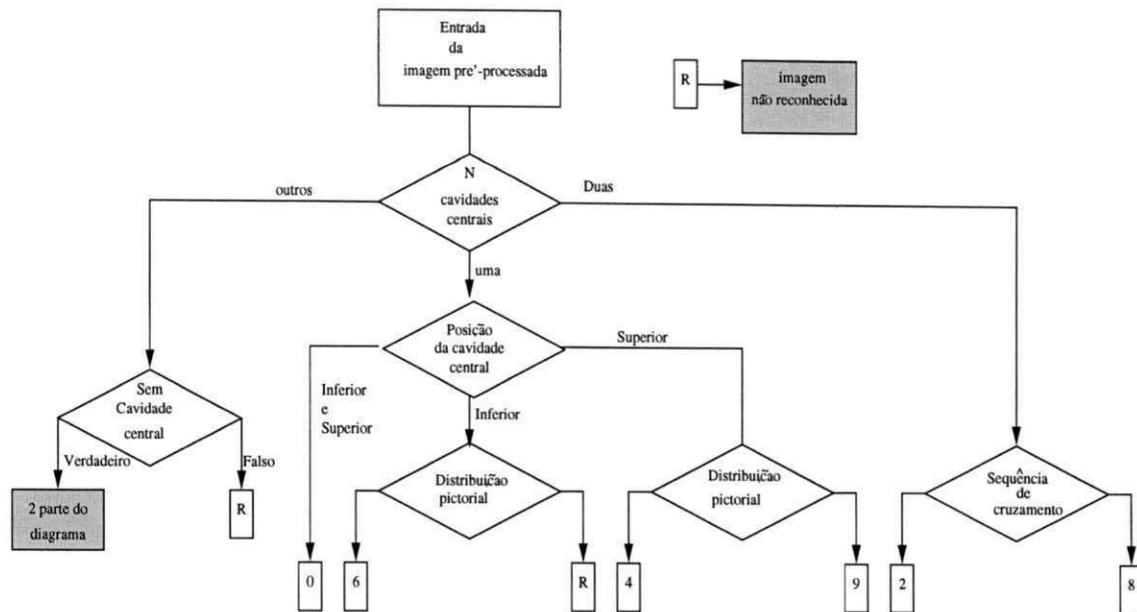


Figura 4.4: Fluxograma do algoritmo sintático, parte 1

4.2.1 Classificação dos Numerais que Possuem Cavidades Centrais

Através da análise das formas que os numerais manuscritos apresentam, os numerais que possuem cavidades centrais foram divididos em dois grupos. O primeiro grupo é constituído dos numerais com maior probabilidade de possuir duas cavidades centrais, como os numerais 2 e 8. O segundo grupo é composto pelos numerais com maior probabilidade de apresentar uma única cavidade central, que são os numerais 0, 2, 4, 6 e 8.

A classificação dos numerais que possuem duas cavidades centrais, ilustrada na Figura 4.4 é realizada mediante comparação entre as seqüências de cruzamento do numeral desconhecido e as seqüências de cruzamento dos dígitos 2 e 8 de referência, de acordo com a menor distância encontrada. As seqüências dos dígitos de referência para cada classe são calculadas e armazenadas pelo algoritmo na sua fase de implementação.

Os numerais que apresentam uma única cavidade central, são primeiramente analisados com relação à disposição desta cavidade no numeral. Se a cavidade central encontra-se simultaneamente na parte superior e inferior do caracter, então o numeral é classificado como o dígito 0.

Se a cavidade central ocupa a parte inferior do numeral, é analisada a existência de pixels pretos do lado direito do eixo principal entre a primeira e a segunda interseção deste eixo com os pixels pretos que compõem o traçado do caracter. Caso não exista

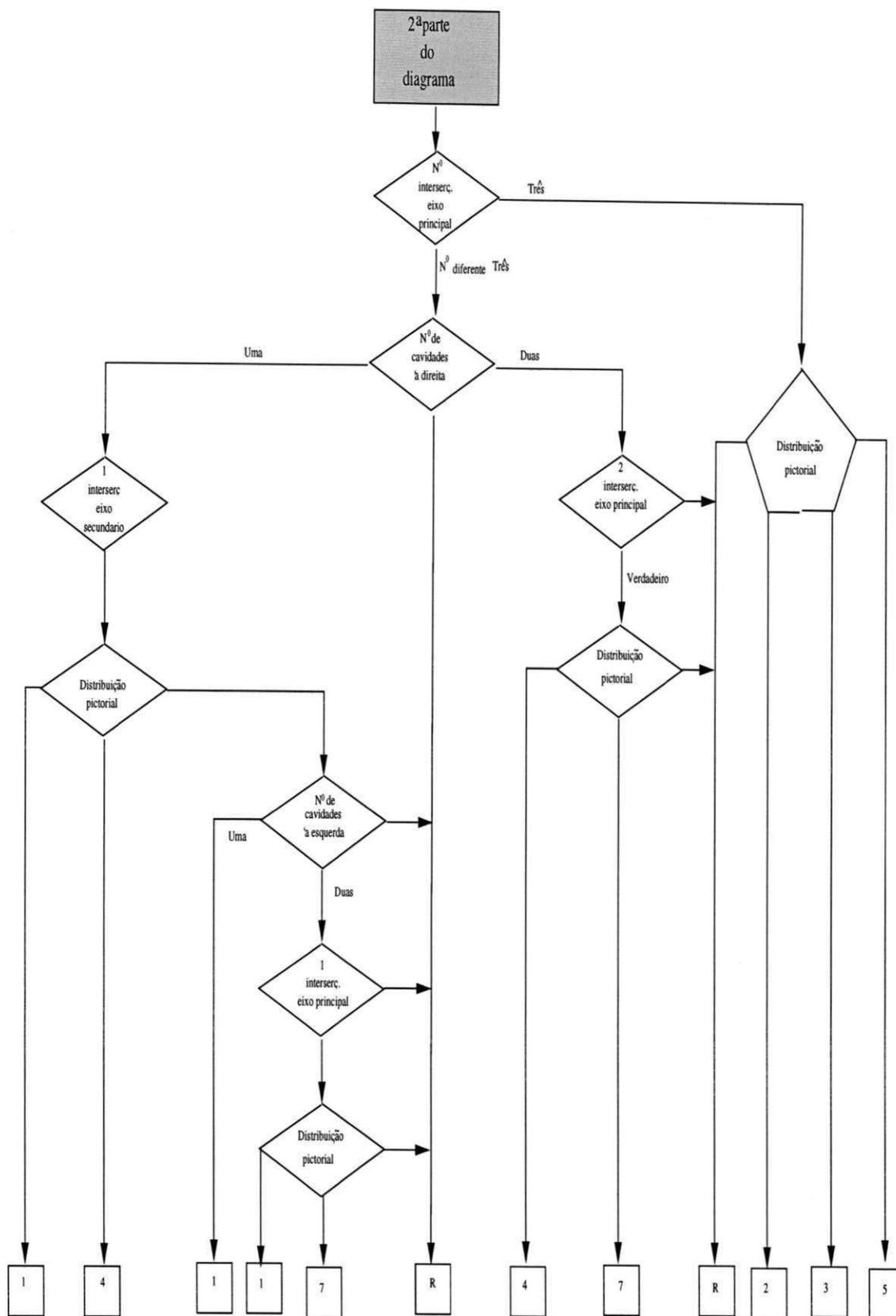


Figura 4.5: Fluxograma do algoritmo sintático, parte 2

pixels pretos nesta região, o numeral é classificado como o dígito 6. De outro modo o numeral não pode ser classificado.

Os numerais 4 e 9 geralmente, possuem uma cavidade na sua parte superior, como também não apresentam pixels pretos do lado esquerdo entre a segunda interseção do caracter com seu eixo principal e sua 18ª linha. A distinção entre estes numerais é realizada em dois passos. O primeiro, é verificado a quantidade de pixels pretos na última linha do caracter. Se o número de pixels pretos for maior do que três, o numeral desconhecido é classificado como 9. Caso contrário, é determinado o segmento de reta que passa por dois pontos específicos da matriz do numeral. O primeiro ponto é o último pixel preto da primeira linha da matriz do caracter, o segundo ponto é o primeiro pixel preto, localizado na linha da segunda interseção com o eixo principal do caracter. Com este segmento de reta calculado, para cada linha da matriz cortada por este segmento, a distância absoluta entre o ponto do segmento de reta naquela linha e o pixel preto mais próximo (caso haja mais de um pixel, somente um é o escolhido) é obtida. O numeral desconhecido é classificado como 4 se o somatório das distâncias calculadas for menor do que 70% do número de pixels que formam o segmento de reta. Caso contrário o numeral desconhecido é classificado como sendo o dígito 9.

4.2.2 Classificação dos Numerais que não Apresentam Cavidades Centrais

Os numerais que não apresentam cavidades centrais são divididos em dois grupos: os numerais que apresentam três interseções com o eixo principal e os numerais que apresentam outro número qualquer de interseções com este mesmo eixo. O algoritmo de reconhecimento destes numerais, ilustrado na Figura 4.5 é descrito a seguir.

Numerais que Apresentam Três Interseções com o Eixo principal

Os numerais com maior probabilidade de seu traçado interceptar o eixo principal em três posições distintas[6], são os dígitos 2, 3 e 5.

No processo de classificação são analisadas quatro condições:

1. Existência de pixels pretos à direita, entre a primeira e a segunda interseção do caracter com o eixo principal.
2. Existência de pixels pretos à esquerda, entre a primeira e a segunda interseção do caracter com o eixo principal.

3. Existência de pixels pretos à direita, entre a segunda e terceira interseção do caracter com o eixo secundário.
4. Existência de pixels pretos à esquerda, entre a segunda e terceira interseção do caracter com o eixo secundário.

O numeral é classificado como 5 quando somente as condições 2 e 3 são verdadeiras. O numeral é classificado como 3 quando somente as condições 1 e 2 são verdadeiras e se existirem mais de três pixels pretos na última linha da matriz do caracter. Esta última análise previne que o numeral 7 seja erroneamente classificado como sendo o numeral 3. O numeral é classificado como 2 quando somente as condições 1 e 4 são verdadeiras.

Numerais que Apresentam Número de Interseções com o Eixo Principal Diferente de Três

O objetivo deste grupo é classificar os numerais 1, 4 e 7. Para tanto os numerais são analisados conforme as seguintes características:

1. Numerais com duas cavidades à direita,
2. Numerais com uma cavidade à direita,
3. Numerais com uma cavidade à direita e uma cavidade à esquerda,
4. Numerais com uma cavidade à direita e duas cavidades à esquerda,

Os numerais com duas cavidades à direita podem tanto serem classificados como 4 ou como 7. Para isto, a matriz do caracter desconhecido deve satisfazer às seguintes condições: possuir duas interseções com o eixo principal e existirem mais de três pixels pretos em cada linha abaixo da segunda interseção do caracter com o seu eixo principal. A classificação é realizada conforme a existência de pixels pretos acima e à esquerda da segunda interseção com o eixo principal. Caso ocorra a situação acima especificada, o numeral é classificado como 4, caso contrário o numeral desconhecido é classificado como 7.

Os numerais que apresentam uma cavidade à direita são classificados como 1, 4 ou 7. A classificação é realizada após a verificação da existência de uma única interseção entre o traçado do caracter e seu eixo secundário. Caso esta interseção localize-se abaixo das três primeiras linhas e exista pixels pretos acima e à esquerda deste eixo, o

numeral é analisado. Caso esta interseção ocorra em uma das três primeiras linhas é analisado o número de cavidades à esquerda.

São classificados como 4, os numerais que possuem a primeira interseção do caracter com o eixo secundário localizada abaixo das três primeiras linhas e que possua pixels pretos acima e à esquerda desta interseção. Caso contrário, não havendo pixels pretos acima e à esquerda desta primeira interseção, o numeral desconhecido é classificado como 1.

Os numerais desconhecidos onde a primeira interseção do traçado do caracter com o eixo secundário localiza-se em uma das três primeiras linhas do caracter, que possua apenas uma cavidade à esquerda e sua matriz apresente no máximo três pixels pretos por linha, são classificados como o numeral 1.

Os numerais desconhecidos onde a primeira interseção do traçado do caracter com o eixo secundário esteja localizado em uma das primeiras três linhas do caracter e possua duas cavidades à esquerda são classificados como sendo o numeral 1 ou 7. Para que isto seja possível, deve ocorrer apenas uma interseção com o eixo principal do caracter e o número de pixels por linha após o ponto de interseção, deve ser no máximo três. O caracter é classificado como sendo 7 se existir o segmento de reta que caracteriza este numeral na sua parte superior. De outro modo, o numeral é classificado como sendo o dígito 1.

4.3 Algoritmo Modificado

O algoritmo desenvolvido por Gomes[6] classifica os numerais de acordo com as características extraídas de sua topologia e distribuição pictorial, como foi anteriormente mencionado. Desta forma, é desejável que as características utilizadas no processo de classificação sejam discriminatórias.

Entretanto, este algoritmo, explicado na seção anterior, foi desenvolvido visando utilização em sistemas de reconhecimento de cheques. Nesta aplicação, os algarismos ou números apresentam, em geral, um bom comportamento. Este fato deve-se ao extremo cuidado com que a maioria das pessoas escrevem os dígitos ao preencher cheques bancários. O mesmo não acontece em outras situações, como por exemplo, na escrita de códigos postais, onde os dígitos, muitas vezes, são escritos sem nenhum cuidado adicional. Por esta razão, os números coletados de cartas postais apresentam uma grande diversidade de formas. Além deste fato, o estilo de escrita sofre variações de um país para o outro, como também entre as várias regiões de um mesmo país. Esta variação pode ser facilmente comprovada através de uma comparação entre o estilo

brasileiro e o norte-americano de escrever o numeral 7, como ilustrado na Figura 4.6. Geralmente os norte-americanos não costumam passar um traço no meio do numeral 7, como os brasileiros o fazem.

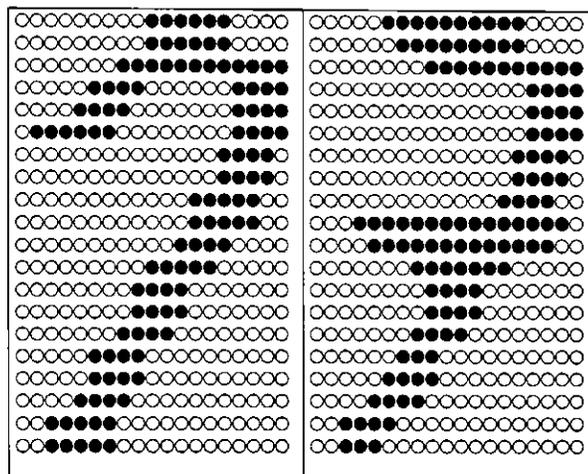


Figura 4.6: Comparação entre o estilo norte-americano e brasileiro de escrever o numeral 7

O teste realizado com o algoritmo sintático utilizando o banco de dados do CEN-PAMI, produziu, pelos motivos citados acima, uma taxa de apenas 30,45% de reconhecimento, como está relatado no Capítulo 6. Este desempenho está bastante abaixo daqueles relatados nos trabalhos que utilizam este método[6, 34, 27, 10, 5].

Devido ao desempenho insatisfatório do algoritmo original foi realizada uma análise da diversidade e do estilo de cada um dos dez dígitos constituintes do banco de dados. Com o resultado desta análise, foram realizadas modificações no algoritmo, de forma a adequar o mesmo ao seu universo de trabalho. As modificações entretanto foram realizadas de forma a preservar o conjunto de atributos utilizados pelo algoritmo original.

O diagrama de operação do algoritmo foi dividido em quatro partes, como ilustrado nas Figuras 4.7, 4.8, 4.9 e 4.10. A primeira parte é dedicada à classificação dos numerais que apresentam uma ou duas cavidades centrais. A segunda, terceira e quarta partes deste diagrama são dedicadas à análise dos numerais que não possuem cavidades centrais.

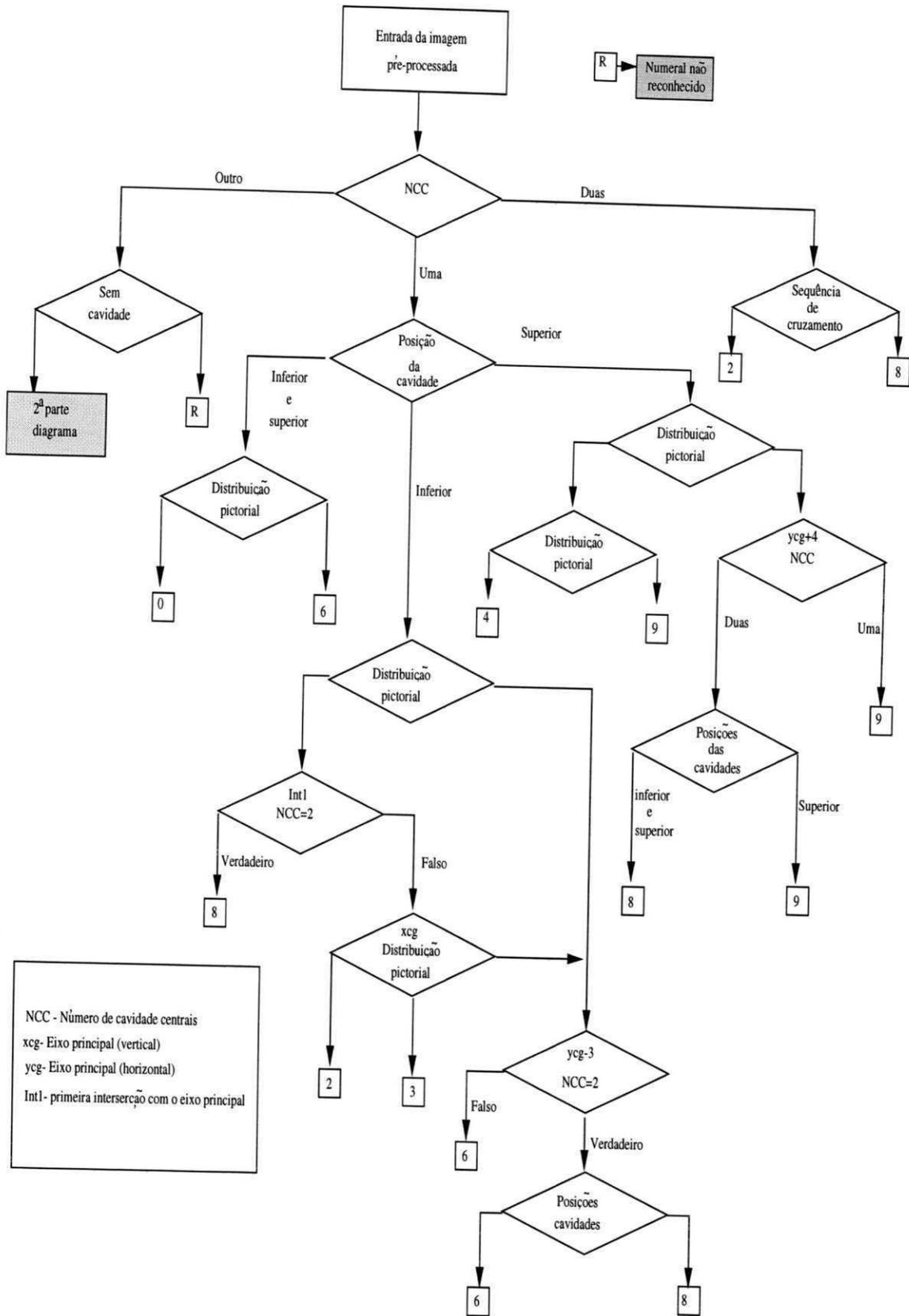


Figura 4.7: Fluxograma do algoritmo sintático modificado - parte 1

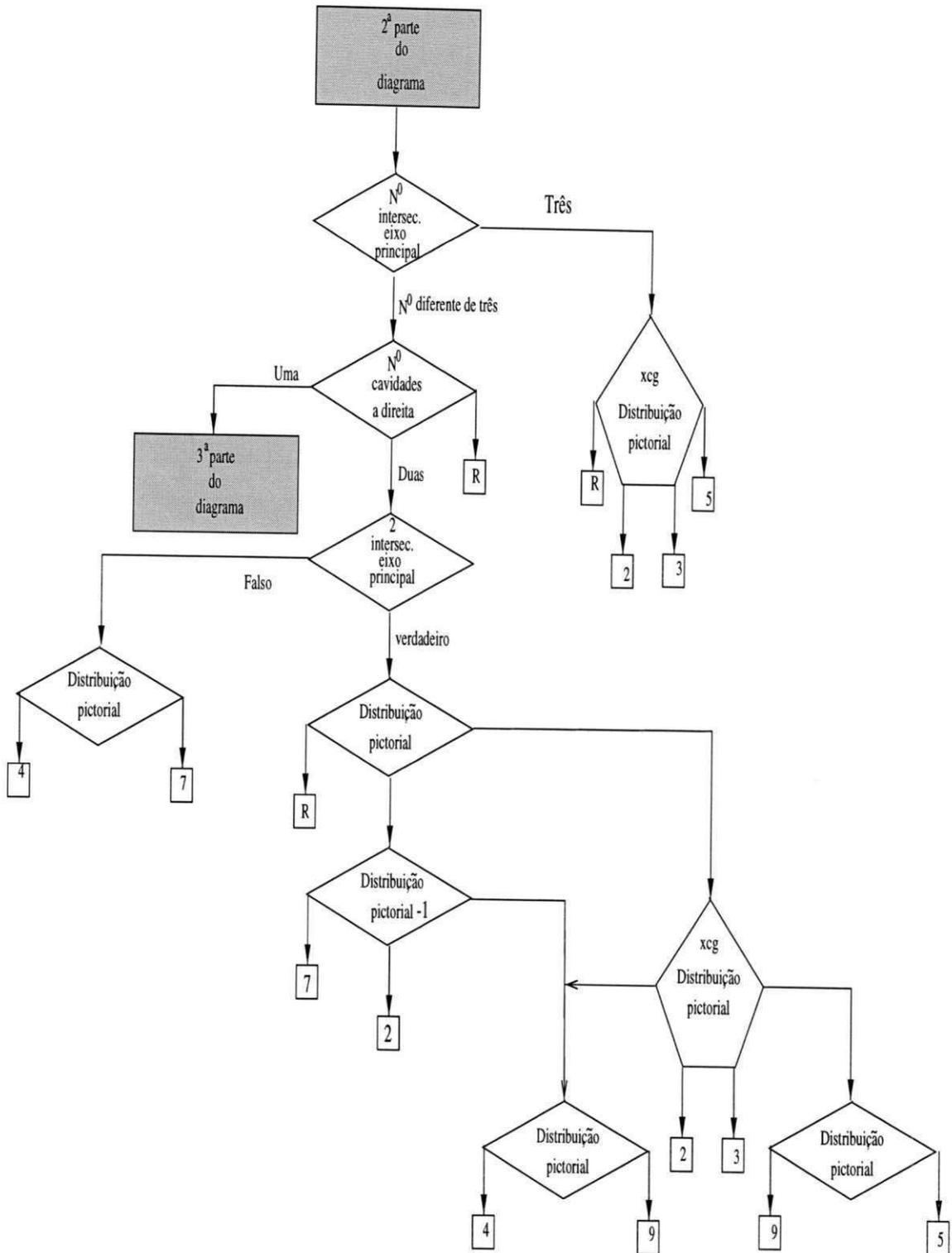


Figura 4.8: Fluxograma do algoritmo sintático modificado - parte 2

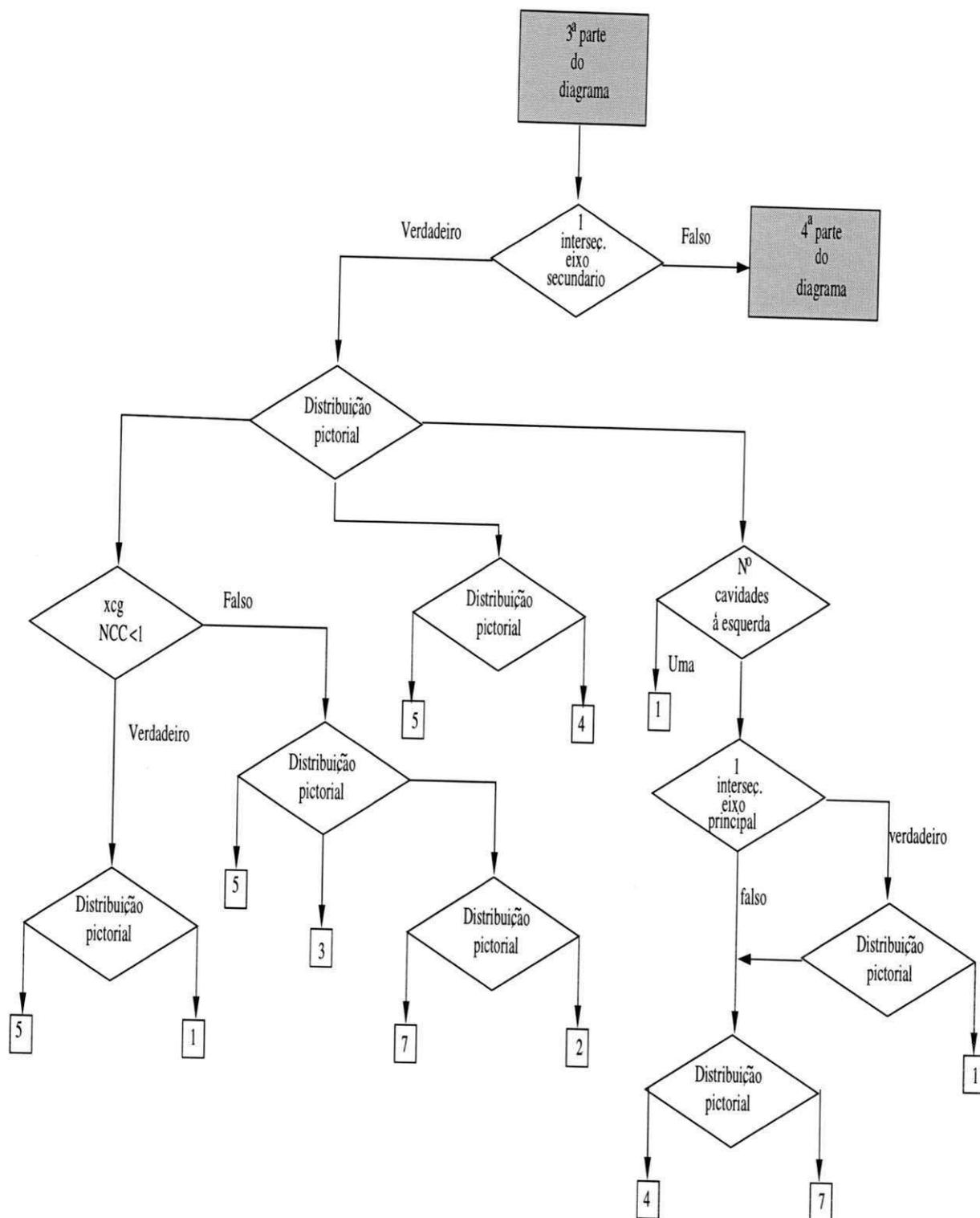


Figura 4.9: Fluxograma do algoritmo sintático modificado - parte 3

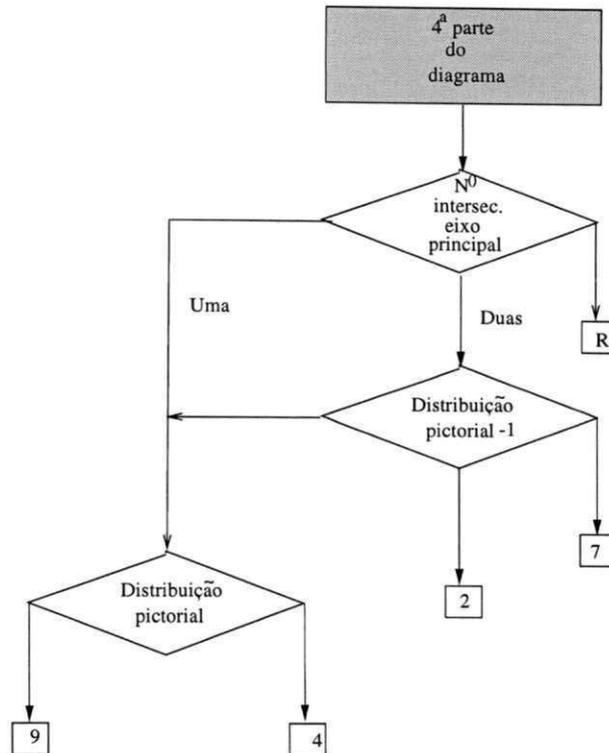


Figura 4.10: Fluxograma do algoritmo sintático modificado - parte 4

4.3.1 Numerais com Uma ou Duas Cavidades Centrais

A classificação dos numerais que apresentam duas cavidades centrais não sofreu nenhuma modificação. Ou seja, a distinção entre os numerais 2 e 8 é realizada da mesma forma como proposto no algoritmo original.

Entretanto, o mesmo não aconteceu na classificação dos numerais que apresentam uma cavidade central.

Numerais com Uma Cavidade Central Localizada na Parte Superior e Inferior

Através da análise dos dígitos, observou-se que a cavidade central do numeral 6 nem sempre encontra-se localizada na parte inferior de sua matriz. Alguns numerais 6 são escritos de forma tal, que a sua cavidade central se estende por ambas as partes, superior e inferior, confundindo assim o algoritmo original. Para solucionar este problema, os numerais que apresentam uma cavidade central na parte inferior e superior simultaneamente, são classificados como 0 ou 6, de acordo com sua distribuição pictorial. Testa-se a existência de pixels pretos à direita, entre a primeira e a segunda intersecções do eixo principal. Não existindo pixels pretos nesta região, o numeral é

classificado como sendo 6, de outro modo o numeral desconhecido é classificado como sendo 0.

Numerais com Uma Cavidade Central Localizada na Parte Inferior

Os numerais com maior probabilidade de apresentar uma cavidade central localizada na parte inferior de sua matriz são 2, 3, 6 e 8. Primeiramente, se analisa a existência de pixels pretos à direita entre a primeira e a segunda interseções com o eixo principal. Os numerais que possivelmente não possuem pixels pretos nesta região são o 6 e o 8. A distinção entre eles é feita inserindo um traço horizontal na terceira linha paralela que antecede o eixo horizontal principal. Calcula-se novamente o número de cavidades da imagem modificada. Se esta imagem apresentar uma única cavidade central, o numeral é reconhecido como 6, caso contrário é feita uma análise das posições destas cavidades. Ocorrendo uma cavidade na parte superior e outra na parte inferior o numeral é reconhecido como 8, de qualquer outro modo o numeral é reconhecido como sendo 6.

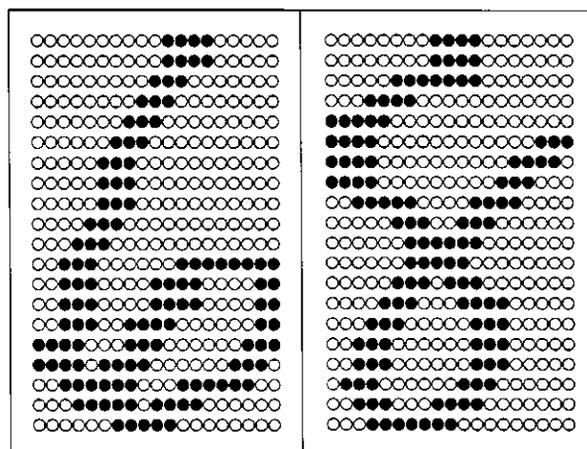


Figura 4.11: Exemplos de numerais com uma cavidade central na parte inferior

Os numerais desconhecidos que apresentam pixels pretos à direita entre a primeira e a segunda interseções com o eixo principal tem sua imagem modificada, pela inserção de um traço horizontal passando pela primeira interseção com o eixo principal. Calcula-se o número de cavidades centrais da nova imagem. As imagens com duas cavidades centrais são classificadas como sendo o numeral 8.

Caso contrário, a imagem original é reestabelecida e uma nova modificação é feita. Desta vez, é inserido um traço vertical na imagem, passando pelo centro de gravidade ou seja, sobre o eixo principal. A distinção entre os numerais é feita através da análise

da distribuição pictorial. O numeral é classificado como 2 se: 1) a imagem modificada possuir duas cavidades centrais, localizadas uma à direita, entre a primeira e a segunda interseções com o eixo principal e outra localizada à esquerda, entre a segunda e a terceira interseções com o eixo principal; ou 2) a imagem modificada apresentar uma única cavidade central localizada à esquerda, entre a segunda e terceira interseções com o eixo principal; ou 3) não existir pixels pretos à esquerda, entre a primeira e a segunda interseções e à direita, entre a segunda e terceira interseções com eixo principal. O numeral é classificado como 3 caso: 1) a imagem modificada possua duas cavidades centrais localizadas uma à direita, entre a primeira e a segunda interseções com eixo principal e outra localizada à direita, entre a segunda e a terceira interseções com o mesmo eixo; ou 2) a imagem modificada apresente uma única cavidade central localizada à direita, entre a segunda e a terceira interseções com o eixo principal; ou ainda 3) não existirem pixels pretos à esquerda, entre a primeira e segunda e entre a segunda e a terceira interseções com o eixo principal. Caso contrário o numeral pode ser reconhecido como 6 ou 8 e a distinção é realizada através de uma análise da distribuição pictorial que classifica estes numerais, anteriormente explicada.

Numerais com uma Cavidade Central Localizada na Parte Superior

Os numerais mais prováveis em de apresentarem uma cavidade central, localizada na parte superior são 4, 8 e 9. Os numerais 8 que encontram-se neste grupo são os que apresentam uma descontinuidade no traçado na sua parte inferior. Primeiramente é analisado o número de pixels pretos existentes na última linha da matriz do caracter. Se o número de pixels pretos for igual a três, a distinção é realizada de maneira similar à realizada pelo algoritmo original, classificando os numerais 4 e 9.

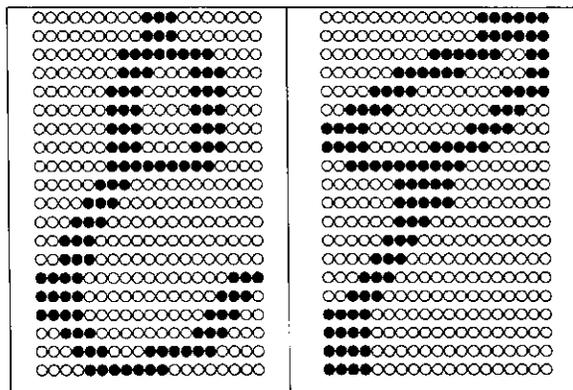


Figura 4.12: Exemplos de numerais que possuem cavidades centrais na parte superior

Entretanto, caso o número de pixels pretos na última linha da matriz do caracter seja maior que três, são feitas modificações no algoritmo original, permitindo, assim, uma distinção entre os numerais 8 e 9, mostrados na Figura 4.12 . Isto é feito através da alteração da forma do traçado do numeral pela inserção de um traço horizontal na quarta linha após o eixo horizontal principal. O numeral é classificado como 8, quando a sua imagem modificada, apresentar duas cavidades centrais localizadas uma na parte superior e outra na parte superior. Caso contrário o numeral é reconhecido como 9.

4.3.2 Numerais sem Cavidades Centrais

Do mesmo modo que no algoritmo original, os numerais sem cavidade central são divididos em dois grupos: os numerais que apresentam três interseções com o eixo principal e os numerais que apresentam outro número qualquer de interseções com o este principal.

Numerais com Três Interseções com o Eixo Principal

Através da análise feita com o conjunto de treinamento, comprovou-se que os numerais com maior probabilidade de apresentar três interseções com o eixo principal são os dígitos 2, 3 e 5. Entretanto, os testes com estes numerais utilizando o algoritmo sintático original, produziram resultados insatisfatórios. Após um minucioso estudo, objetivando encontrar as razões para as dificuldades encontradas pelo algoritmo sintático original no reconhecimento dos numerais 2, 3 e 5, notou-se que eles não eram reconhecidos, devido à forma de classificação utilizada, onde é analisada a existência de pixels pretos em certas regiões da matriz do caracter. Por exemplo, para classificar o numeral desconhecido como 2, é necessário que não existam pixels pretos à esquerda, entre a primeira e a segunda interseções com o eixo principal e à direita, entre a segunda e a terceira interseções com o eixo principal. Entretanto, a grande maioria dos números, na base de dados, apresentam “resquícios” na terminação dos seus traçados, como mostra a Figura 4.13. Estes “resquícios”, geralmente, são localizados nas regiões onde são analisadas a existência de pixels pretos. Desta forma, muitos dos dígitos, embora sejam visualmente identificáveis, não são reconhecidos pelo algoritmo.

A solução proposta foi introduzir um traço vertical na matriz do caracter, sobre o eixo principal da matriz do numeral e computar o número de cavidades centrais. A classificação é realizada mediante à análise da disposição destas cavidades centrais na matriz do numeral.

A matriz do numeral é dividida em quatro regiões, de acordo com as localizações das interseções do traçado do numeral com o seu eixo principal. A primeira região, é

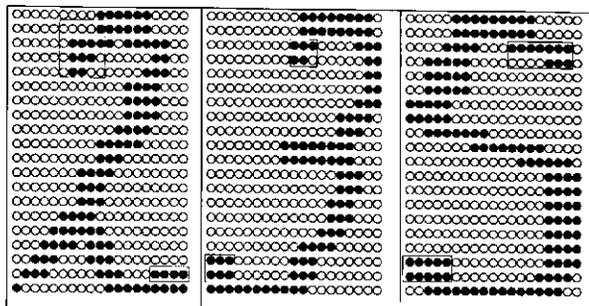


Figura 4.13: Exemplo de resquícios no traçado dos numerais 2, 3 e 5

formada pelos pixels que encontram-se no lado direito, entre a primeira e a segunda interseções com o eixo principal. A segunda região é formada pelos pixels que encontram-se no lado esquerdo, entre a primeira e a segunda interseções com o eixo principal. A terceira região, é formada pelos pixels que encontram-se no lado direito, entre a segunda e a terceira interseções. Finalmente, a última região é formada pelos pixels que encontram-se no lado esquerdo, entre a segunda e a terceira interseções.

Os numerais são, assim, classificados:

O numeral é classificado como 2, se possuir cavidades localizadas na primeira e quarta regiões, ou possuir uma única cavidade central localizada na quarta região.

O numeral é classificado como 3, se possuir cavidades localizadas na primeira e terceira regiões.

O numeral é classificado como 5, se possuir cavidades localizadas na segunda e terceira regiões, ou possuir uma única cavidade central localizada na segunda região.

Numerais que Apresentam Número de Interseções com o Eixo Principal Diferente de Três

Esta parte do algoritmo sofreu grandes alterações, para se adequar aos diferentes estilos de escrita encontrados no banco de dados. A comprovação deste fato é feita mediante uma rápida análise do fluxograma, mostrado na Figura 4.8. Os numerais que apresentam número de interseções com o eixo principal diferente de três são analisados conforme as seguintes características:

1. Numerais com uma cavidade à direita
2. Numerais com uma cavidade à direita e uma cavidade à esquerda
3. Numerais com uma cavidade à direita e duas cavidades à esquerda

4. Numerais com duas cavidades à direita.

Os numerais que possuem uma cavidade à direita e uma única interseção com o eixo secundário são 1, 2, 3, 4, 5 e 7. O processo de classificação é feito realizando uma análise da distribuição pictorial, onde verifica-se a posição da interseção do caracter com o eixo secundário e a existência de pixels pretos acima e à esquerda desta interseção.

Os numerais onde, a posição da interseção com o eixo secundário ocorre abaixo das três primeiras linhas do caracter e existem pixels pretos acima e à esquerda desta interseção, são classificados como 4 ou 5. A distinção é realizada através do número de pixels pretos existentes na última linha. O numeral desconhecido é classificado como 5, quando a sua imagem apresenta cinco ou mais pixels em sua última linha. De outro modo, o numeral é classificado como 4.

Os numerais que não apresentam pixels pretos acima e à esquerda da interseção com o eixo secundário, tem sua matriz alterada pela inserção de um traço vertical, sobre o eixo principal da matriz original do caracter. Se o número de cavidades da matriz alterada for maior ou igual a um, é realizada uma análise da distribuição pictorial. Esta análise é praticamente a mesma realizada com os numerais que não apresentam cavidades centrais e possuem três interseções com o eixo principal. A diferença está em uma análise pictorial adicional, objetivando a distinção entre os numerais 2 e 7. Esta distinção é realizada através do número de pixels por linha à partir da décima sexta linha. Caso entre as linhas analisadas ocorra pelo menos uma linha onde o número de pixels pretos seja maior do que cinco, o numeral desconhecido é classificado como 2, caso contrário tem-se o numeral 7. Por outro lado, se a matriz alterada não possuir cavidades centrais e o número de pixels pretos por linha entre a primeira e a quinta linha for maior que sete, o numeral é reconhecido como sendo 5, caso contrário tem-se o numeral 1.

Entretanto, para os numerais onde a interseção com o eixo secundário ocorre nas primeiras três linhas, analisa-se o número de cavidades à esquerda.

Os numerais desconhecidos que apresentarem uma única cavidade à esquerda são classificados como 1. De outra forma, o número de interseções com o eixo principal e a distribuição pictorial são analisada, de modo a classificar os numerais como sendo 1, 4 ou 7. Para os numerais que apresentarem uma única interseção com o eixo principal e o número de pixel pretos por linha, após a quinta linha depois desta interseção for maior ou igual a cinco, é feita uma comparação entre as seqüências de cruzamento da matriz e as seqüências de cruzamento dos dígitos 1 e 7 padrões. Se a distância absoluta entre as seqüências de cruzamento do numeral desconhecido e as seqüências de cruzamento padrão do dígito 1 for menor que as mesmas distâncias absolutas com relação ao dígito

7, o numeral é reconhecido como 1. Caso contrário é feita uma análise da distribuição pictorial de forma a classificar os numerais 4 e 7.

A distinção entre os numerais 4 e 7 é realizada através da análise do número de pixels pretos existentes por linha do caracter. Da quarta linha até o eixo horizontal principal são analisadas apenas as linhas que possuem seqüência de cruzamento horizontal igual a um. A partir do eixo principal todas as linhas subseqüentes são analisadas. Caso dentre essas linhas exista pelo menos uma com número de pixels pretos maior que cinco, o numeral desconhecido é classificado como sendo 4, caso contrário o numeral é reconhecido como 7.

Os numerais que possuem uma cavidade à direita e mais de uma interseção com o eixo secundário, ilustrados na Figura 4.10, não são analisados pelo algoritmo original. Na versão modificada porém são classificados de acordo com o número de interseções com o eixo principal. Os numerais que apresentam uma única interseção com o eixo principal são reconhecidos como sendo os numerais 4 ou 9, de acordo com sua distribuição pictorial. Caso o número de pixels pretos na primeira linha da matriz seja maior que cinco e a seqüência de cruzamento desta linha seja igual a um, o numeral desconhecido é classificado como sendo 9, de outro modo tem-se o numeral 4. Por outro lado os numerais que apresentarem duas interseções com o eixo principal, são analisados conforme a sua distribuição pictorial-1.

A análise da distribuição pictorial-1 visa classificar os dígitos em 2, 4, 7 e 9. É analisado o número de pixels pretos por linha. Da quarta linha até o eixo principal na horizontal são analisadas as linhas que possuem apenas uma seqüência de pixels pretos. A partir do eixo principal horizontal todas as linhas são analisadas. Se não existir pelo menos uma linha onde o número de pixels for maior que cinco o numeral desconhecido é classificado como 7. Caso contrário, é analisada a posição da primeira linha que possui mais de cinco pixels pretos. Se esta linha localiza-se entre as cinco últimas linhas, o numeral é reconhecido como 2. De outra forma, é analisada a distribuição pictorial que classifica os dígitos 4 e 9, anteriormente explicada.

Os numerais que apresentam duas cavidades à direita, ilustrados na Figura 4.9, são analisados conforme o número de interseções com o eixo principal. Os numerais que apresentam número de interseções com o eixo principal diferente de dois são o 4 e o 7. A classificação é feita com base na distribuição pictorial, que classifica estes numerais, como anteriormente explicado. Por outro lado os numerais que apresentarem duas interseções com o eixo principal e que possuam pelo menos uma linha com um número de pixel pretos superior a cinco abaixo da segunda interseção com o eixo principal, utilizam sua distribuição pictorial no processo de classificação. Os numerais que não

possuem pixels pretos à esquerda entre a primeira e a segunda interseção com o eixo principal são classificados através da distribuição pictorial-1, anteriormente explicada.

De outro modo é feita uma classificação similar à utilizada na classificação dos numerais sem cavidades centrais contendo três interseções com o eixo principal.

4.4 Conclusão

Neste capítulo foram descritas as características que descrevem aspectos importantes da geometria e da topologia do desenho do caracter utilizados pelos algoritmos sintáticos implementados.

Em seguida foram apresentados os algoritmos sintático original e sintático expandido ou modificado. Este algoritmo modificado foi desenvolvido no decorrer deste trabalho, através de alterações nas regras de classificação utilizadas pelo algoritmo sintático original, visando adequar o classificador aos estilos de escrita encontrados no banco de dados do CENPARMI.

No algoritmo sintático expandido foi proposta uma nova forma de detectar concavidades na matriz dos numerais, através de modificações na matriz original pela inserção de linhas ou traços em posições específicas.

A seguir serão descritos os sistemas de reconhecimento baseados em redes neurais.

Capítulo 5

Reconhecimento de caracteres numéricos via redes neurais

5.1 Introdução

Os recentes desenvolvimentos em redes neurais artificiais, máquinas de aprendizagem e computação paralela tem entusiasmado pesquisadores na área de reconhecimento de padrões. Diversos modelos de redes neurais foram implementados como classificadores de padrões. Em particular, redes neurais multicamadas com propagação direta tem demonstrado serem eficientes em reconhecimento de padrões visuais para uma grande variedade de estilos e tamanhos[43, 42, 4, 14].

Este capítulo é dedicado ao estudo dos algoritmos de reconhecimento de caracteres numéricos manuscritos baseados em redes neurais. Visando proporcionar um bom entendimento destes algoritmos é apresentado a seguir uma introdução às redes neurais. Em seguida é analisada a rede neural multicamada com propagação direta, implementada como classificadora de caracteres numéricos manuscritos. Finalizando é apresentado um estudo sobre uma nova arquitetura de rede neural recorrente proposta por Lee[43], para reconhecimento de padrões visuais.

5.2 Redes Neurais

A possibilidade de dar às máquinas a habilidade que o ser humano possui de aprender, reter informações, recordá-las e aplicá-las na solução de diversos tipos de problemas, tem entusiasmado muitos pesquisadores a procurar desenvolver modelos computacionais para os mesmos fins. Uma classe dentre estes modelos é a dos chamados sistemas neurais artificiais ou simplesmente redes neurais.

Uma rede neural tem uma estrutura massivamente paralela e distribuída e possui a capacidade de aprender um dado tipo de conhecimento, armazená-lo e usá-lo para executar certa tarefa. De acordo com Haykin [44], as redes neurais apresentam as seguintes propriedades:

1. **Não linearidade.** Um neurônio é um dispositivo não-linear. Conseqüentemente, a rede neural, interconexão de neurônios, é uma rede não linear. Além disso, a não linearidade ocorre de um modo bastante especial, uma vez que é distribuída ao longo da rede.
2. **Mapeamento entrada-saída.** Um popular paradigma de aprendizagem denominado de aprendizagem supervisionada envolve a modificação dos pesos sinápticos da rede neural pela aplicação de um conjunto de treinamento. Cada amostra do conjunto de treinamento consiste de uma única entrada e saída desejada. Os pesos sinápticos da rede são modificados de forma a minimizar a diferença entre a saída produzida pela rede e a saída desejada. O treinamento da rede é realizado com todas as amostras do conjunto de treinamento até que a diferença entre as saída produzida e desejada atinja um mínimo aceitável. Desta forma, a rede aprende através dos exemplos, pela construção de um mapeamento entrada-saída para o problema em questão.
3. **Adaptabilidade.** As redes neurais apresentam uma capacidade intrínseca de adaptar seus pesos sinápticos de acordo com a situação em questão. Em particular, uma rede neural treinada para operar em um ambiente específico pode ser re-treinada para lidar com pequenas modificações nas condições ambientais de operação. Além disso, ao operar em um ambiente não-estacionário, uma rede neural pode ser projetada para modificar seus pesos sinápticos em tempo real.
4. **Tolerância a falhas.** Implementada em *hardware*, uma rede neural é potencialmente tolerante a falhas no sentido de que sua performance é apenas gradativamente degradada sob condições de operações adversas, tendo em vista a natureza distribuída da informação ao longo da rede.
5. **Possibilidade de implementação em VLSI.** A natureza massivamente paralela de uma rede neural a torna potencialmente rápida para realização de certas tarefas. Esta mesma característica a torna idealmente adequada para implementação por hardware paralelo em VLSI. Este tipo de implementação provê um meio de capturar o comportamento complexo de uma rede neural em uma feição extremamente hierárquica, possibilitando o seu uso para operação em tempo real

de aplicações envolvendo reconhecimento de padrões, processamento de sinais e controle.

6. **Uniformidade.** Uma mesma notação é utilizada em todos os domínios que envolvem aplicações de redes neurais. Além disso, os neurônios, de uma forma ou de outra, representam o ingrediente comum a todas as redes, e neste contexto é possível aplicar teorias e algoritmos de aprendizagem em aplicações diversas.

Podemos citar, dentre outras, as seguintes aplicações de redes neurais: conversão de textos em representações fonéticas, reconhecimento de voz, reconhecimento de caracteres, reconstrução de imagens e compressão de imagens.

Os sistemas de redes neurais são utilizados para resolver muitos problemas de reconhecimento de padrões. Existem duas abordagens de classificação de padrões utilizando redes neurais. A primeira abordagem utiliza-se inicialmente de extratores de características e só então empregam redes neurais para classificar os padrões. A segunda e mais nova abordagem utiliza redes neurais para determinar e extrair as características dos padrões. Em particular, redes neurais de múltiplas camadas tem-se mostrado eficiente em reconhecimento de padrões visuais[26, 42].

O que é comumente chamado de redes neurais é um conjunto interconectado de elementos de processamento (PE), denominados de neurônios, células ou nós, cada qual realizando um simples cálculo. O modelo do neurônio apresentado na Figura 5.1, possui entradas (conjunto de *sinapse*), a cada uma das quais é associado um peso e uma saída, que pode ser usada como entrada de outros elementos de processamento. O valor associado com qualquer neurônio é chamado de sua ativação (*net*) e representa a soma ponderada das entradas. Ou seja, para um neurônio k :

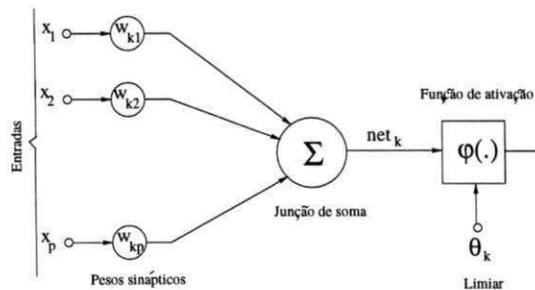


Figura 5.1: Modelo do neurônio

$$net_k = \sum_{j=1}^N x_j w_{kj}, \quad (5.1)$$

onde N é o número de entradas do neurônio, x_j são as entradas do neurônio e w_{kj} são os pesos sinápticos associados a cada entrada.

A saída de um elemento de processamento pode ser simplesmente o seu valor de ativação. Entretanto, na maioria das redes neurais a saída de um neurônio é a saída de uma função de ativação, algumas vezes chamada de função de saída expressa por:

$$y = \varphi(\text{net}_k - o_k), \quad (5.2)$$

onde o_k é o limiar.

A função de ativação $\varphi(\cdot)$ garante que o valor de saída do elemento de processamento encontra-se dentro de uma faixa pré-definida. Vários tipos de funções de ativação são usadas para ativar um neurônio artificial, porém o uso de uma função particular depende do tipo de dados de saída (contínuo ou discreto) e da faixa de valores assumidos por estes dados (por exemplo, de -1 a 1). As funções de ativação comumente encontradas na literatura são:

a) Função linear:

$$\varphi(x) = x; \quad (5.3)$$

b) Função degrau unitário:

$$\varphi(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases}; \quad (5.4)$$

c) Função bipolar:

$$\varphi(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ -1, & \text{se } x < 0 \end{cases}; \quad (5.5)$$

d) Função tangente hiperbólica:

$$\varphi(x) = \tan(x); \quad (5.6)$$

e) Função Sigmoidal:

$$\varphi(x) = \frac{1}{1 + e^{-ax}}. \quad (5.7)$$

Os elementos de processamento não são comumente encontrados isolados, mas estão conectados como um grafo (*graph*), normalmente chamado de rede (*network*). As saídas de um conjunto de PEs podem ser usadas como entradas para um outro ou para muitos PEs. Desta forma uma rede pode ser constituída de muitas camadas (*layers*), com um ou mais elementos (*units*) em cada camada.

Em geral, pode-se identificar quatro arquiteturas diferentes de redes: redes de única camada com propagação direta, redes multicamadas com propagação direta, redes recorrentes e estruturas reticuladas.

As redes de camada única com propagação direta possuem uma camada de entrada de neurônios fonte que projetam as entradas para a camada posterior de neurônios, ou seja, a rede é estritamente do tipo propagação direta. A designação de camada única deve-se ao fato da rede possuir apenas uma camada de neurônios computacionais. Redes multicamadas com propagação direta possuem uma ou mais camadas escondidas, situadas entre a camada de entrada e a saída, cujo os neurônios computacionais são chamados de neurônios escondidos. A rede neural é dita totalmente conectada quando todo o neurônio em cada camada da rede conecta-se a todos os neurônios da camada adjacente posterior. Se algumas dessas ligações ou pesos sinápticos estão ausentes, a rede é denominada parcialmente conectada.

Redes recorrentes distinguem-se das redes com propagação direta pela presença de pelo menos um laço de realimentação. É importante salientar que os *loops* de realimentação necessitam da presença de um elemento de atraso.

Estruturas reticuladas consistem de um arranjo unidimensionais, bidimensionais ou n -dimensionais de neurônios, cujas as entradas são fornecidas por um conjunto de neurônios de entrada. É importante salientar que cada neurônio de entrada conecta-se a todos os neurônios do reticulado.

Antes de uma rede neural executar determinada tarefa é necessário que ela seja treinada. O treinamento ou a aprendizagem, no sentido de redes neurais, significa determinar os pesos sinápticos para cada elemento de processamento, através de algoritmos de treinamento. O treinamento de uma rede neural consiste na apresentação de um conjunto de treinamento com propriedades desconhecidas à entrada da rede e em ajustar os seus pesos sinápticos até obter a saída desejada. Este processo é repetido diversas vezes com diferentes classes de dados até que os pesos sinápticos encontrem-se estabilizados. Neste ponto, o processo de aprendizagem fica completo e a rede pode ser usada para classificar as entradas.

Os principais tipos de algoritmos de treinamento encontrados na literatura são: treinamento supervisionado, treinamento não supervisionado e pesos sinápticos fixos.

No treinamento supervisionado, uma seqüência de vetores de entrada é apresentada à rede, cada qual associado a um vetor de saída. Os vetores de treinamento são aplicados sequencialmente à rede, os erros são calculados e os pesos sinápticos são ajustados para cada vetor, até que o erro para todo o conjunto atinja um valor mínimo aceitável. No treinamento não supervisionado são aplicados à rede apenas os vetores

de entrada, sem seus correspondentes vetores de saída. O algoritmo de treinamento modifica os valores de pesos, baseados na observação dos erros internos, para produzir uma saída consistente, isto é, se aplicado um dos vetores de treinamento ou um vetor suficientemente similar a ele, ambos deverão produzir a mesma saída.

E por último tem-se o algoritmo de pesos sinápticos fixos, onde não há nenhum ajuste iterativo dos pesos sinápticos. Os valores de pesos sinápticos são fixados de maneira a se ter a informação sobre a saída desejada de acordo com as restrições e quantidades a serem minimizadas ou maximizadas.

5.3 Redes Neurais Multicamadas com Propagação Direta

Modelos de redes neurais usados na implementação de sistemas de reconhecimento de padrões, em particular redes neurais multicamadas com propagação direta tem produzido resultados satisfatórios em reconhecimento de padrões visuais, como anteriormente afirmado[43].

O sistema de reconhecimento de caracteres numéricos manuscritos implementado neste trabalho utiliza padrões de entrada (imagens de caracteres) normalizados em tamanho para 16×16 . Vetores de características horizontais, verticais, diagonal direita e diagonal esquerda são calculados a partir da imagem normalizada através das máscaras *kirsch*[15, 22, 43, 18]. Adicionalmente, cada vetor direcional da imagem 16×16 é normalizado para um vetor 4×4 . Além disso, é utilizada a imagem original de entrada do sistema também normalizada em 4×4 , como vetor de características globais. Como resultado, são apresentados à rede cinco vetores de dimensões 4×4 : quatro vetores 4×4 de características locais e um vetor 4×4 de características globais. Estas características são usadas como valores de entrada da rede neural, por conseguinte o vetor de características de entrada da rede equivale a uma imagem com 80 pixels.

A rede neural implementada consiste de 80 neurônios na camada de entrada, 80 neurônios na camada escondida e 10 neurônios na camada de saída, ilustrada na Figura 5.2. Cada neurônio de saída corresponde a uma das dez classes de caracteres numéricos existentes. Para um dado padrão de entrada, uma destas saídas deve ser 1 (identificando o caracter) e as demais deverão ser zero.

O treinamento é realizado pela aplicação do vetor de características na entrada da rede. O papel dos neurônios da camada de entrada é simplesmente repassar cada um dos valores do vetor de característica a todos os neurônios da camada escondida. A ativação da saída associada com a camada escondida i é dada pela soma ponderada de

todas as entradas, ou seja:

$$net_i = \sum_{j=0}^{80} w_{ij}i_j, \quad (5.8)$$

onde net_i é o valor de ativação do i -ésimo neurônio escondido, i_j é o valor do j -ésimo neurônio de entrada e w_{ij} é o peso associado ao *link* entre o j -ésimo neurônio de entrada e o i -ésimo neurônio escondido. A saída da camada escondida é dada pela função sigmoïdal:

$$\varphi(net_i) = \frac{1}{1 + e^{-net_i}}. \quad (5.9)$$

Uma situação similar existe entre a camada escondida e a camada de saída da rede; cada neurônio da camada de saída tem seu valor associado à soma ponderada de todas as saídas dos neurônios da camada escondida:

$$O_k = \sum_{j=0}^{80} \varphi(net_i)w_{ki}, \quad (5.10)$$

onde O_k é o valor de ativação associado a saída do neurônio k , $\varphi(net_i)$ é a saída do i -ésimo neurônio escondido e w_{ki} é o peso sináptico aplicado entre o i -ésimo neurônio escondido e o k -ésimo neurônio de saída. A saída binária deste neurônio de saída é $\varphi(O_k)$ e existe apenas uma saída não nula por dígito.

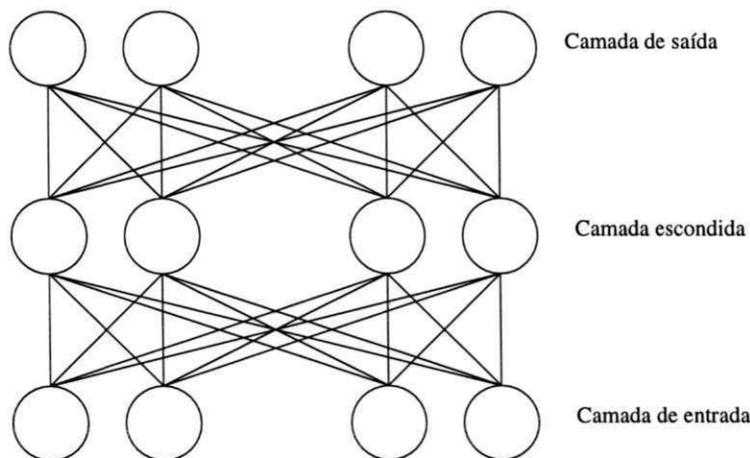


Figura 5.2: Redes neurais multicamadas com propagação direta

Antes da rede ser usada para classificar corretamente qualquer dígito ela deve ser treinada. Isto significa determinar os pesos sinápticos w_{ij} e w_{ki} que levam a uma correta classificação. Inicialmente os pesos sinápticos devem ser uniformemente distribuídos

dentro de uma estreita faixa. A razão da estreita faixa é reduzir a probabilidade dos neurônios da rede saturarem e produzirem pequenos gradientes de erro. A rede é treinada pela aplicação de padrões de classificação conhecidos. Os valores de saída para todos os neurônios são computados e o resultado é comparado com os valores desejados. Por exemplo, quando um três é utilizado como padrão de entrada, todos os neurônios de saída devem produzir saída 0, exceto o terceiro neurônio que deve produzir saída 1.

A função de erro (E_p) é definida como sendo proporcional ao erro quadrático entre a saída atual e a saída desejada, para todos os padrões a serem treinados. Ou seja, para um dado padrão p de entrada:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - O_{pj})^2, \quad (5.11)$$

onde O_{pj} representa a saída atual do j -ésimo neurônio, t_{pj} representa a saída desejada do neurônio j e E_p é a função erro para o padrão p .

O objetivo do treinamento é ajustar os pesos sinápticos da rede de forma a minimizar a função de erro. O algoritmo utilizado para treinar a rede é o algoritmo de retropropagação [45] e consiste dos seguintes passos:

1. Inicializa os pesos sinápticos e limiares (*thresholds*). Os pesos sinápticos da rede e os limiares devem ser inicializados com pequenos números randômicos, com o intuito de prevenir, por exemplo, que a rede fique saturada com grandes valores de peso.
2. Apresentar os valores das entradas e das saídas desejadas.
3. Ativar a rede para produzir as saídas.
4. Calcular o erro entre a saída produzida pela rede e a saída desejada.
5. Ajustar os pesos sinápticos da rede visando minimizar o erro, de acordo com a equação:

$$w_{ij}(n+1) = w_{ij}(n) + \eta \delta_{pj} O_{pj} + \alpha [w_{ij}(n) - w_{ij}(t-1)], \quad (5.12)$$

onde η representa o termo de ganho e α representa o momento. Para a camada de saída, tem-se

$$\delta_{pj} = k O_{pj} (1 - O_{pj}) (t_{pj} - O_{pj}), \quad (5.13)$$

e para a camada escondida, tem-se

$$\delta_{pj} = kO_{pj}(1 - O_{pj}) \sum_k \delta_{pk} w_{kj}. \quad (5.14)$$

6. Repetir os passos 2 à 6 até que o erro global (para todo o conjunto de treinamento) seja aceitavelmente baixo.

5.4 Redes Neurais Recorrentes

Nos casos de reconhecimento de padrões visuais com redes neurais multicamadas com propagação direta, as camadas escondidas aprendem a maximizar a informação útil dos padrões de entrada, enquanto as camadas de saída trabalham na tentativa de discriminar a informação recebida das unidades escondidas[43]. Por conseguinte, é razoável pensar no reaproveitamento das informações produzidas pelas unidades de saída de forma a elevar a potência discriminatória dos padrões visuais.

Redes neurais recorrentes oferecem uma estrutura adequada para reutilizar os valores de saída da rede na fase de treinamento. De acordo com o especificado acima, Lee [43] propôs uma nova arquitetura de redes neurais recorrentes com maior potência discriminatória para reconhecimento de padrões visuais.

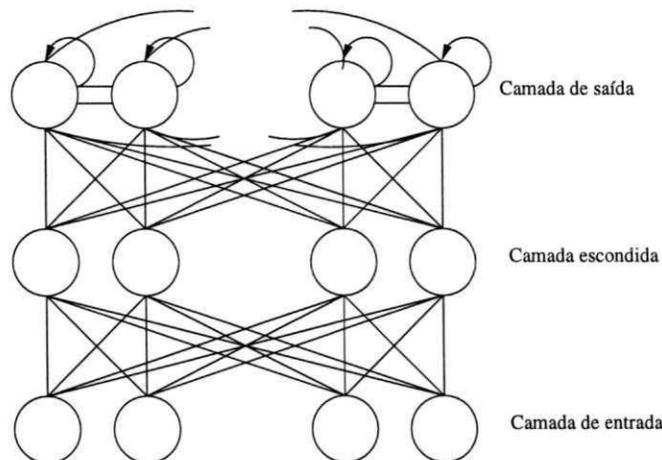


Figura 5.3: Rede neural recorrente

A arquitetura proposta consiste de três camadas, como mostra a Figura 5.3. A camada escondida é totalmente conectada a todos os neurônios de entrada e cada neurônio de saída é conectado a ele próprio e também totalmente conectado aos outros

neurônios de saída e a todos os neurônios escondidos. Desta forma, o valor de ativação da unidade de saída i no ciclo n para o padrão p é dado por:

$$net_{pi}^o(n) = \sum_{j=1}^l w_{ij} O_{pj}^h(n) + \sum_{k=1}^q z_{ik} O_{pk}^o(n-1), \quad (5.15)$$

onde $O_{pj}^h(n)$ é o valor de saída do j -ésimo neurônio escondido, w_{ij} é o peso entre o j -ésimo neurônio escondido e o i -ésimo neurônio de saída, z_{ik} é o peso sináptico entre o k -ésimo neurônio de saída e o i -ésimo neurônio de saída, l é o número de neurônios escondidos e q é o número de neurônios de saída. A saída do i -ésimo neurônio de saída para o padrão p é dado por:

$$O_{pi}^o(n) = \varphi(net_{pi}^o(n)), \quad (5.16)$$

onde a função de ativação $\varphi(\cdot)$ é a função sigmoid.

O valor de ativação da i -ésima unidade escondida (*hidden*) no ciclo n para o padrão p é obtido como segue:

$$net_{pi}^h(n) = \sum_{j=1}^m w_{ij} i_{pj}(n), \quad (5.17)$$

onde $i_{pj}(n)$ é o valor de saída do j -ésimo neurônio de entrada, w_{ij} é o peso entre o j -ésimo neurônio de entrada e o i -ésimo neurônio escondido e m é o número de neurônios de entrada. As unidades de entrada possuem características de função de transferência linear. Os valores de ativação das unidades de saída são inicialmente zerados. Os valores do vetor de característica são apresentados à entrada da rede e estes valores são diretamente propagados até as unidades de saída tornarem-se ativas.

O treinamento desta rede não é realizado utilizando o algoritmo de retropropagação, em virtude de sua arquitetura recorrente. O algoritmo de treinamento desta rede é chamado de treinamento supervisionado temporal, pois os valores dos neurônios de saída são comparados com os valores de saída desejados de acordo com o tempo especificado[46]. Nenhum algoritmo de treinamento para este tipo de rede foi encontrado na literatura pesquisada. Para a implementação pretendida neste trabalho, portanto, o algoritmo deve ser desenvolvido. A dedução deste algoritmo é mostrada a seguir.

O erro quadrático da rede é definido como sendo proporcional ao quadrado da diferença entre a saída real e a saída desejada no tempo n para o padrão p , ou seja:

$$E_p(n) = \frac{1}{2} \sum_i (d_{pi}(n) - O_{pi}^o(n))^2, \quad (5.18)$$

onde $d_{pi}(n)$ é a saída desejada no tempo n para o padrão p .

O algoritmo trabalha visando reduzir o valor da função de erro, variando para isto os pesos sinápticos da rede. Desta forma, tem-se:

$$\frac{\partial E_p(n)}{\partial w_{ij}(n)} = \frac{\partial E_p(n)}{\partial net_{pi}^o(n)} \frac{\partial net_{pi}^o(n)}{\partial w_{ij}(n)}. \quad (5.19)$$

Olhando para o segundo termo da equação 5.19 e de acordo com a equação 5.15 tem-se:

$$\begin{aligned} \frac{\partial net_{pi}^o(n)}{\partial w_{ij}(n)} &= \frac{\partial}{\partial w_{ij}(n)} \left[\sum_{z=1}^l w_{iz}(n) O_{pz}^h(n) + \sum_{k=1}^q z_{ik}(n) O_{pk}^o(n-1) \right] \\ &= \sum_j \frac{\partial w_{iz}(n)}{\partial w_{ij}(n)} O_{pz}^h(n) + \sum_k \frac{\partial}{\partial w_{ij}(n)} z_{ik}(n) O_{pk}^o(n-1) \\ &= O_{pj}^h(n) \end{aligned} \quad (5.20)$$

considerando que $\frac{\partial w_{iz}}{\partial w_{ij}}$ é igual a zero para $k \neq j$ e igual a um para $k = j$.

A taxa de variação do erro em relação à saída da rede para o i -ésimo neurônio é definida como:

$$-\frac{\partial E_p(n)}{\partial net_{pi}^o(n)} = \delta_{pi}(n) \quad (5.21)$$

Substituindo equações 5.20 e 5.21 na equação 5.19, tem-se:

$$-\frac{\partial E_p(n)}{\partial w_{ij}(n)} = \delta_{pi}(n) O_{pj}^h(n) \quad (5.22)$$

Reduzir o valor de E_p significa portanto, fazer a variação do peso proporcional a $\delta_{pi} O_{pj}^h(n)$, ou seja:

$$\Delta_p w_{ij} = \eta \delta_{pi}(n) O_{pj}^h(n). \quad (5.23)$$

É necessário saber δ_{pi} para cada i -ésima unidade. Sendo assim, pode-se reduzir o valor de E_p e conseqüentemente descobrir a variação dos pesos sinápticos da rede. Usando a equação 5.21 e a regra da cadeia podemos escrever

$$\delta_{pi}(n) = -\frac{\partial E_p(n)}{\partial net_{pi}^o(n)} = -\frac{\partial E_p(n)}{\partial O_{pi}^o(n)} \frac{\partial O_{pi}^o(n)}{\partial net_{pi}^o(n)}. \quad (5.24)$$

Considerando a equação 5.16, obtem-se:

$$\frac{\partial O_{pi}^o(n)}{\partial net_{pi}^o(n)} = \varphi' (net_{pi}^o(n)). \quad (5.25)$$

Derivando-se $E_p(n)$ em relação a $O_{pi}^o(n)$ na equação 5.18:

$$-\frac{\partial E_p(n)}{\partial O_{pi}^o(n)} = d_{pi}(n) - O_{pi}^o(n) \quad (5.26)$$

Desta forma, substituindo equações 5.25 e 5.26 na equação 5.24:

$$\delta_{pi}(n) = \varphi' \left(net_{pi}^o(n) \right) \left(d_{pi}(n) - O_{pi}^o(n) \right) \quad (5.27)$$

Esta equação é útil para a camada de saída, desde que a saída desejada e a atual sejam conhecidas, mas não para as camadas escondidas, onde as saídas desejadas não são conhecidas. Para a camada escondida tem-se:

$$\frac{\partial E_p(n)}{\partial w_{ij}(n)} = \frac{\partial E_p(n)}{\partial net_{pi}^h(n)} \frac{\partial net_{pi}^h(n)}{\partial w_{ij}(n)}. \quad (5.28)$$

Para obter o segundo termo da equação 5.28 deriva-se a equação 5.17:

$$\begin{aligned} \frac{\partial net_{pi}^h(n)}{\partial w_{ij}(n)} &= \frac{\partial}{\partial w_{ij}(n)} \sum_{k=1}^m w_{ik}(n) i_{pk}(n) \\ &= \sum_k \frac{\partial w_{ik}(n)}{\partial w_{ij}(n)} i_{pk}(n) \\ &= i_{pj}(n) \end{aligned} \quad (5.29)$$

considerando que $\frac{\partial w_{ik}}{\partial w_{ij}}$ é igual a zero para $k \neq j$ e igual a um para $k = j$.

A taxa de variação do erro como função da variação da saída da rede para a unidade é definida como:

$$-\frac{\partial E_p(n)}{\partial net_{pi}^h(n)} = \delta'_{pi}(n) \quad (5.30)$$

e de acordo com a equação 5.28 tem-se:

$$-\frac{\partial E_p(n)}{\partial w_{ij}(n)} = \delta'_{pi}(n) i_{pj}(n) \quad (5.31)$$

Reduzir o valor de E_p significa portanto, fazer a variação do peso sináptico proporcional a $\delta'_{pj} i_{pj}(n)$, ou seja:

$$\Delta_p w_{ij} = \eta \delta'_{pi} i_{pj}(n). \quad (5.32)$$

É necessário conhecer δ'_{pi} para cada unidade. Sendo assim, pode-se reduzir o valor de E_p e conseqüentemente conhecer a variação dos pesos sinápticos da rede. Usando a equação 5.30 e a regra da cadeia podemos escrever

$$\delta'_{pi}(n) = -\frac{\partial E_p(n)}{\partial net_{pi}^h(n)} = -\frac{\partial E_p(n)}{\partial O_{pi}^h(n)} \frac{\partial O_{pi}^h(n)}{\partial net_{pi}^h(n)} \quad (5.33)$$

Para o segundo termo da equação 5.33, tem-se:

$$\frac{\partial O_{pi}^h(n)}{\partial net_{pi}^h(n)} = \varphi' \left(net_{pi}^h(n) \right). \quad (5.34)$$

A derivada do erro em relação à saída da camada escondida, pode ser obtida como:

$$\frac{\partial E_p(n)}{\partial O_{pi}^h(n)} = \sum_y \frac{\partial E_p(n)}{\partial net_{py}^o(n)} \frac{\partial net_{py}^o(n)}{\partial O_{pi}^h(n)}, \quad (5.35)$$

Derivando-se a equação 5.15:

$$\begin{aligned} \frac{\partial E_p(n)}{\partial O_{pi}^h(n)} &= \sum_y \frac{\partial E_p(n)}{\partial net_{py}^o(n)} \frac{\partial}{\partial O_{pi}^h(n)} \sum_{j=1}^l w_{yj}(n) O_{pj}^h(n) + \sum_{y=1}^q y_{lk}(n) O_{pk}^o(n-1) \\ &= -\sum_y \delta_{py}(n) w_{yi}(n) \end{aligned} \quad (5.36)$$

Substituindo as equações 5.34 e 5.36 na equação 5.33, obtem-se:

$$\delta_{pi}(n) = \varphi' \left(net_{pi}^h(n) \right) \sum_l \delta_{pl}(n) w_{li}(n) \quad (5.37)$$

Utilizando a função de ativação sigmoid tem-se:

$$\varphi \left(net_{pj}(n) \right) = \frac{1}{1 + e^{-k \cdot net_{pj}(n)}} \quad (5.38)$$

$$\begin{aligned} \varphi' \left(net_{pj}^h(n) \right) &= \frac{ke^{-k \cdot net_{pj}(n)}}{\left(1 + e^{-k \cdot net_{pj}(n)} \right)^2} \\ &= k\varphi \left(net_{pj}(n) \right) (1 - \varphi \left(net_{pj}(n) \right)) \\ &= kO_{pj}(n) (1 - O_{pj}(n)) \end{aligned} \quad (5.39)$$

A variação dos pesos sinápticos do *loop* recorrente é deduzida a seguir:

$$\frac{\partial E_p(n)}{\partial z_{ij}(n)} = \frac{\partial E_p(n)}{\partial net_{pi}^o(n)} \frac{\partial net_{pi}^o(n)}{\partial z_{ij}(n)}. \quad (5.40)$$

Derivando a equação 5.40 em relação a z_{ij} :

$$\begin{aligned} \frac{\partial net_{pi}^o(n)}{\partial z_{ij}(n)} &= \frac{\partial}{\partial z_{ij}(n)} \sum_{j=1}^l w_{ij}(n) O_{pj}^h(n) + \sum_{k=1}^q z_{ik}(n) O_{pk}^o(n-1) \\ &= \sum_j \frac{\partial w_{ij}(n)}{\partial z_{ij}(n)} O_{pj}^h(n) + \sum_K \frac{\partial}{\partial z_{ij}(n)} z_{ik}(n) O_{pk}^o(n-1) \\ &= O_{pj}^o(n-1) \end{aligned} \quad (5.41)$$

A taxa de variação do erro como função da variação da saída da rede para a unidade é definida como:

$$-\frac{\partial E_p}{\partial net_{pi}^o(n)} = \delta_{pi} \quad (5.42)$$

e de acordo com a equação 5.40 tem-se:

$$-\frac{\partial E_p(n)}{\partial z_{ij}} = \delta_{pi}(n)O_{pj}^o(n-1) \quad (5.43)$$

Reduzir o valor de $E_p(n)$ significa fazer a variação do peso proporcional a $\delta_{pi}(n)O_{pj}^o(n-1)$, ou seja:

$$\Delta_p z_{ij} = \eta \delta_{pi}(n)O_{pj}^o(n-1). \quad (5.44)$$

Concluindo, são apresentados os passos realizados durante a fase de treinamento utilizando o algoritmo desenvolvido acima:

1. Inicializar os pesos sinápticos e *threshold*.

Os pesos sinápticos da rede e o *threshold* devem ser inicializados com pequenos números randômicos, com o intuito de prevenir, por exemplo, que a rede não fique saturada com grandes valores de peso.

2. Zerar os valores de ativação das unidades de saída.
3. Apresentar os valores das entradas e das saídas desejadas.
4. Ativar a rede para produzir as saídas.
5. Calcular o erro entre a saída produzida pela rede e a saída desejada.
6. Ajustar os pesos sinápticos da rede visando minimizar o erro, segundo a equação:

$$w_{ij}(n+1) = w_{ij}(n) + \eta \delta_{pi}(n)O_{pi}^h(n) + \alpha[w_{ij}(n) - w_{ij}(n-1)], \quad (5.45)$$

onde η representa o termo de ganho e α representa o momento. Para a camada de saída, tem-se

$$\delta_{pj}(n) = kO_{pj}(n)(1 - O_{pj}(n))(t_{pj}(n) - O_{pj}(n)), \quad (5.46)$$

e para a camada escondida, tem-se

$$\delta_{pj}(n) = kO_{pj}(n)(1 - O_{pj}(n)) \sum_k \delta_{pk}(n)w_{kj}(n). \quad (5.47)$$

O ajuste dos pesos sinápticos para os *loops* recorrentes é feito segundo a equação:

$$z_{ij}(n + 1) = z_{ij}(n) + \eta\delta_{pu}(n)O_{pj}^o(n - 1) + \alpha[z_{ij}(n) - z_{ij}(n - 1)]. \quad (5.48)$$

6. Repetir os passos 2 à 6 até que o erro global (para todo o conjunto de treinamento) seja aceitavelmente baixo.

Como a seqüência de treinamento é realizada em dois tempos, t_0 e t_1 , por caracter, as modificações dos pesos sinápticos ocorre somente no ciclo t_1 .

5.5 Conclusão

Vários sistemas vem usando as redes neurais na tarefa de reconhecimento de caracteres numéricos manuscritos, onde diversos modelos de redes neurais de multicamadas são utilizados. Neste capítulo foram apresentados dois classificadores baseados em redes neurais. O primeiro utiliza redes neurais multicamadas com propagação direta e o segundo utiliza uma arquitetura de redes neurais recorrentes.

No próximo capítulo são apresentados os testes realizados com os algoritmos implementados durante o decorrer deste trabalho.

Capítulo 6

Testes Efetuados e Resultados Obtidos

Neste capítulo são avaliados os sistemas de reconhecimento de caracteres numéricos manuscritos implementados: algoritmo sintático, algoritmo sintáticos modificado e algoritmo de classificação baseado em redes neurais multicamadas com propagação direta, através de resultados obtidos experimentalmente. Os testes foram realizados utilizando as imagens provenientes do Banco de Dados do CENPARMI, que consiste de 17.771 caracteres numéricos manuscritos isolados. As imagens dos caracteres foram digitalizadas por um *scanner* ótico e em seguida convertidas em arquivos no formato adequado ao processamento.

O banco de dados é composto por dez arquivos, cada qual contendo as imagens pertencentes a um único dígito. A quantidade de imagens é irregulamente distribuída através das dez classes de numerais, conforme ilustrado na Tabela 6.1. Para evitar que os algoritmos fossem treinados ou testados com um número distinto de amostras por classe, o conjunto de treinamento foi gerado com os primeiros 450 dígitos de cada arquivo, totalizando assim, 4.500 numerais. O conjunto de testes foi gerado com 266 dígitos de cada arquivo, obtidos após a retirada dos dígitos que irão compor o conjunto de treinamento, totalizando 2.600 numerais.

As fases de pré-processamento e de extração das características foram implementadas em linguagem C, máquina PENTIUM 200MHz, assim como os algoritmos de classificação baseados em análise sintática. Os algoritmos de treinamento e classificação utilizando redes neurais foram implementados em linguagem C, máquina 2-PENTIUM-PRO 200MHz, com 256Mbytes de memória sob sistema operacional LINUX.

Uma forma bastante utilizada de examinar o comportamento dos sistemas de reconhecimento de caracteres é a matriz de confusão, que é uma matriz bi-dimensional de

| Classe | Amostras |
|--------|----------|
| 0 | 2792 |
| 1 | 3697 |
| 2 | 1701 |
| 3 | 2213 |
| 4 | 1509 |
| 5 | 817 |
| 6 | 1372 |
| 7 | 1741 |
| 8 | 1213 |
| 9 | 716 |

Tabela 6.1: Número de amostras por classe, no banco de dados do CENPARMI

classificação. As colunas da matriz de confusão representam a classificação dos caracteres pelo algoritmo sob investigação e as linhas representam a classe a qual pertence o caracter[15].

Os parâmetros utilizados para avaliar o desempenho do sistema são a taxa de reconhecimento (Rec), a taxa de rejeição (Rej.), a taxa de erro (Err.) e a taxa de confiabilidade do sistema (Conf.), bem como o tempo de processamento.

A taxa de reconhecimento ou taxa de sucesso é a percentagem de imagens de entrada reconhecidas corretamente pelo sistema. A taxa de rejeição é a percentagem de imagens de entrada rejeitadas, ou seja não reconhecidas pelo sistema. A taxa de substituição ou taxa de erro é a percentagem de imagens de entrada reconhecidas erroneamente pelo sistema. A taxa de confiabilidade é a percentagem de imagens de entrada reconhecidas corretamente pelo sistema após a exclusão das amostras rejeitadas. O tempo de processamento indica o tempo gasto para o sistema realizar a tarefa especificada.

Em tarefas de reconhecimento de caracteres, observa-se que é preferível rejeitar um caracter a fazer uma classificação errada. Portanto, a relação entre a taxa de rejeição e a taxa de erro deve ser bem maior a um.

As seções seguintes descrevem os resultados obtidos nos testes efetuados com os algoritmos sintático, sintático modificado e o algoritmo de classificação baseado em redes neurais.

6.1 Algoritmo Sintático

As imagens dos caracteres numéricos selecionados para compor o conjunto de teste são armazenadas em arquivos contendo um único dígito cada, visando facilitar os procedimentos para o processamento. Em seguida, as imagens passam pelas etapas de pré-processamento: normalização em escala, suavização, afinamento ou esqueletização, eliminação dos ramos parasitas, dilatação e novamente pelo processo de normalização em escala. A necessidade de uma segunda etapa de normalização em escala deve-se à redução das dimensões dos caracteres provocada pelos processos de afinamento e eliminação dos ramos parasitas. A primeira normalização em escala efetuada pelo sistema utiliza a função de mapeamento apresentada pela equação 3.1 com os fatores de escala dados pelas equações 3.2 e 3.3. Este algoritmo de normalização garante que a proporção entre a largura e a altura do caracter seja preservada na imagem normalizada, preservando, assim, a forma do traçado do numeral. Entretanto, este algoritmo não garante que o menor retângulo que enquadra o numeral tenha suas dimensões iguais ao especificado. A segunda e última normalização em escala utiliza os fatores de escala dados pelas equações 3.4 e 3.5, que visa garantir que o menor retângulo que enquadra a imagem normalizada do numeral tenha as dimensões especificadas. Entretanto, isto provoca a perda da proporção entre a largura e a altura da imagem original na imagem normalizada.

A etapa seguinte consiste na extração das características a serem utilizadas na classificação dos caracteres.

Os resultados obtidos com o sistema de classificação de caracteres numéricos manuscritos baseado em análise sintática utilizando os caracteres que compõem o conjunto de teste é ilustrado nas Tabelas 6.2 e 6.3.

O algoritmo sintático obteve uma taxa de reconhecimento de 30,45% (Tabela 6.3). Este resultado reflete a dificuldade do algoritmo em reconhecer os estilos de escrita encontrados no banco de dados do CENPARMI. Uma análise da matriz de confusão ilustrada na Tabela 6.2, revela que o algoritmo não consegue reconhecer os dígitos 5 e 6. Entretanto, a maioria dos dígitos 0 e 1 são reconhecidos pelo sistema, a taxa de reconhecimento destes numerais é superior a 70%. Outro dado importante a ser analisado é a taxa de erro do sistema. Embora a maioria dos dígitos apresentados ao sistema não sejam reconhecidos corretamente, apenas 21,35% do conjunto de teste são reconhecidos erroneamente pelo sistema. A grande maioria dos dígitos, 48,19% do conjunto de teste é rejeitado, fato este refletido pela taxa de confiabilidade do sistema de 78,64%.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|----|-----|-----|----|---|----|-----|-----|------|------|------|
| 0 | 191 | 0 | 0 | 0 | 28 | 1 | 0 | 6 | 1 | 0 | 39 | 191 | 36 |
| 1 | 0 | 246 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 16 | 246 | 4 |
| 2 | 0 | 30 | 48 | 0 | 13 | 0 | 1 | 3 | 1 | 1 | 169 | 48 | 49 |
| 3 | 0 | 15 | 2 | 49 | 18 | 0 | 0 | 17 | 0 | 7 | 158 | 49 | 59 |
| 4 | 0 | 32 | 0 | 5 | 62 | 7 | 0 | 30 | 0 | 3 | 127 | 62 | 77 |
| 5 | 0 | 32 | 0 | 1 | 29 | 0 | 0 | 0 | 1 | 1 | 202 | 0 | 64 |
| 6 | 2 | 13 | 0 | 9 | 25 | 2 | 1 | 5 | 2 | 2 | 205 | 1 | 60 |
| 7 | 0 | 52 | 0 | 0 | 18 | 0 | 0 | 11 | 0 | 0 | 185 | 11 | 70 |
| 8 | 0 | 5 | 5 | 3 | 3 | 0 | 0 | 3 | 121 | 15 | 111 | 121 | 34 |
| 9 | 2 | 10 | 0 | 38 | 28 | 21 | 0 | 8 | 8 | 81 | 70 | 81 | 115 |
| Total | 191 | 435 | 55 | 105 | 226 | 31 | 2 | 84 | 134 | 111 | 1282 | 810 | 568 |

Tabela 6.2: Matriz de confusão - Algoritmo Sintático

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 266 | 14,66 | 13,53 | 71,80 | 84,14 |
| 1 | 266 | 6,01 | 1,50 | 92,48 | 98,40 |
| 2 | 266 | 65,53 | 18,42 | 18,04 | 49,48 |
| 3 | 266 | 59,39 | 22,18 | 18,42 | 45,37 |
| 4 | 266 | 47,74 | 28,94 | 23,30 | 44,60 |
| 5 | 266 | 75,93 | 24,07 | 0,00 | 0 |
| 6 | 266 | 77,06 | 22,55 | 0,37 | 1,64 |
| 7 | 266 | 69,54 | 26,31 | 4,13 | 13,18 |
| 8 | 266 | 41,72 | 12,78 | 45,48 | 78,06 |
| 9 | 266 | 26,31 | 43,23 | 30,45 | 41,33 |
| Total | 2660 | 48,19 | 21,35 | 30,45 | 58,78 |

Tabela 6.3: Resultados do reconhecimento - Algoritmo Sintático

As baixas taxas de reconhecimento, apresentadas pelas Tabelas 6.2 e 6.3, obtidas pelo algoritmo sintático demonstram que as regras de decisão utilizadas pelo processo de classificação em árvore não são adequadas ao estilo de escrita dos caracteres contidos no banco de dados do CENPARMI. Como exemplo deste fato, cita-se as regras de decisão utilizadas para classificar os dígitos 6. O numeral é classificado como 6, caso sua imagem possua uma única cavidade central na sua parte inferior e não existam pixels pretos à direita, entre a primeira e segunda interseções do traçado do numeral com o seu eixo principal. Foram analisados 266 numerais 6, deste conjunto quarenta e seis não possuíam cavidades centrais e apenas um numeral possuía as duas condições necessárias a uma correta classificação.

As Tabelas 6.4 e 6.5 mostram os resultados obtidos pelo sistema que utiliza o algoritmo sintático modificado na etapa de classificação. As etapas de pré-processamento e extração das características são as mesmas utilizadas no sistema de classificação baseado no algoritmo sintático original.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 191 | 1 | 26 | 2 | 9 | 0 | 18 | 1 | 5 | 12 | 1 | 191 | 74 |
| 1 | 0 | 235 | 2 | 0 | 2 | 10 | 0 | 3 | 0 | 1 | 13 | 235 | 18 |
| 2 | 0 | 4 | 181 | 5 | 10 | 3 | 9 | 9 | 5 | 7 | 33 | 181 | 52 |
| 3 | 0 | 2 | 35 | 156 | 16 | 20 | 4 | 10 | 1 | 13 | 9 | 156 | 101 |
| 4 | 0 | 21 | 6 | 9 | 182 | 19 | 1 | 6 | 0 | 16 | 6 | 182 | 78 |
| 5 | 0 | 3 | 33 | 9 | 13 | 173 | 9 | 19 | 1 | 2 | 4 | 173 | 89 |
| 6 | 2 | 1 | 38 | 2 | 17 | 7 | 168 | 2 | 9 | 5 | 15 | 168 | 83 |
| 7 | 0 | 8 | 11 | 1 | 12 | 10 | 0 | 179 | 0 | 6 | 39 | 179 | 48 |
| 8 | 0 | 3 | 23 | 7 | 6 | 3 | 18 | 2 | 156 | 19 | 29 | 156 | 81 |
| 9 | 2 | 0 | 7 | 15 | 17 | 52 | 2 | 14 | 11 | 144 | 4 | 144 | 118 |
| Total | 195 | 278 | 362 | 206 | 284 | 307 | 227 | 245 | 188 | 255 | 153 | 1765 | 742 |

Tabela 6.4: Matriz de confusão - Algoritmo Sintático Modificado

Os resultados, na Tabela 6.5 mostram significativo aumento na taxa de reconhecimento com relação às taxas de reconhecimento do sistema original, listada na Tabela 6.3. Através de uma comparação direta dos resultados nas Tabelas 6.3 e 6.5, percebe-se um aumento superior a 100% na taxa de reconhecimento do sistema modificado. Entretanto, ocorre uma diminuição na confiabilidade do sistema, este fato deve-se ao aumento na taxa de erro. As alterações nas regras de decisão proporcionaram uma maior taxa de reconhecimento, em contra partida tornou o sistema mais susceptível

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 266 | 0,37 | 27,81 | 71,80 | 72,07 |
| 1 | 266 | 4,88 | 6,76 | 88,34 | 92,88 |
| 2 | 266 | 12,40 | 19,54 | 68,04 | 77,68 |
| 3 | 266 | 3,38 | 37,96 | 58,64 | 60,70 |
| 4 | 266 | 2,25 | 29,32 | 68,42 | 70,00 |
| 5 | 266 | 1,50 | 33,45 | 65,03 | 66,03 |
| 6 | 266 | 5,63 | 31,20 | 63,15 | 66,93 |
| 7 | 266 | 14,66 | 18,04 | 67,29 | 78,85 |
| 8 | 266 | 10,90 | 30,45 | 58,64 | 65,82 |
| 9 | 266 | 1,50 | 44,36 | 54,13 | 54,96 |
| Total | 2660 | 5,75 | 27,89 | 66,35 | 70,40 |

Tabela 6.5: Resultado do reconhecimento - Algoritmo Sintático Modificado

a erros na classificação dos numerais. Embora os resultados nas taxas de reconhecimento do sistema modificado apresentem uma melhora sobre o sistema original, seu desempenho ainda pode ser considerado baixo.

Muitos fatores contribuem para o baixo desempenho destes sistemas. O primeiro e mais grave destes fatores é a dificuldade de encontrar características e regras de decisões baseadas nestas características que contemplem todas as variações do estilo de escrita dos numerais encontrados no banco de dados do CENPARMI. Outro fator importante é a perda de informações durante o processo de aquisição e pré-processamento das imagens. No processo de aquisição a imagem é discretizada (amostrada) nas coordenadas espaciais e no brilho. Neste processo os traços finos e com baixa intensidade na sua coloração, dependendo da resolução utilizada no processo de digitalização, podem não ser captados introduzindo assim, discontinuidades no traçado dos numerais ou mesmo cortes nas terminações dos seus traçados. No pré-processamento ocorre fato similar, principalmente durante a fase de normalização em escala e na fase de eliminação dos ramos parasitas. Quando o processo de normalização é utilizado para diminuir as dimensões das imagens, ocorre uma reamostragem da imagem, o que sempre provoca perdas de informações, e resulta no desaparecimento de traços que compõem o numeral. Este desaparecimento de traços compromete a classificação, principalmente quando estes traços fazem parte de um *loop* no traçado do numeral, podendo levar ao desaparecimento de cavidades centrais. Como o algoritmo sintático e o algoritmo sintático modificado utilizam o número de cavidades centrais como primeiro indicador

da classe a qual pertence a imagem, a perda destas informações é fatal para o sistema. No processo de eliminação de ramos parasitas, também ocorre perda de informações, geralmente estas perdas são localizadas nas terminações do traçado do numeral.

Muitos numerais pertencentes a classe distintas são escritos de maneira muito similar. As diferenças encontram-se em pequenas sutilezas no traçado, como ilustrado na Figura 6.1, onde mostram-se as ilustrações dos dígitos 5 e 9.

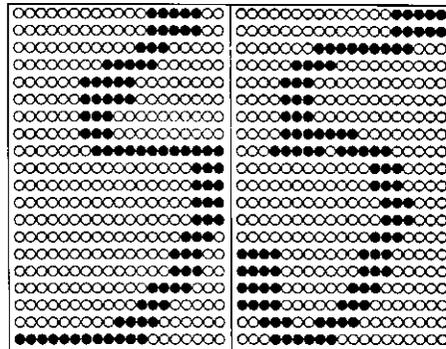


Figura 6.1: Dígito 5 e Dígito 9

O banco de dados do CENPARMI é constituído por dez arquivos, onde cada arquivo contém amostras de caracteres pertencentes a uma única classe. Entretanto, ocorre geralmente a existência de imagens nestes arquivos que são classificadas, através de uma inspeção visual, como pertencentes a outra classe. Existe também, o caso de muitas imagens onde não é possível o reconhecimento visual do dígito.

Como o conjunto de testes foi gerado a partir dos arquivos do banco de dados sem nenhuma intervenção humana, os arquivos que compõem o conjunto de teste contém exemplos de imagens onde não é possível a identificação da classe a qual pertence a imagem através de uma simples inspeção visual, como também imagens pertencentes à classe distintas do arquivo.

O arquivo de teste que contém o conjunto de caracteres pertencentes a classe 4, contém 266 imagens, sendo 18 imagens onde não é possível o reconhecimento visual do dígito e várias reconhecidas visualmente como pertencentes ao dígito 9. Outro exemplo é o arquivo que contém o conjunto de caracteres 3, onde existem doze imagens para as quais não é possível o reconhecimento visual do dígito e duas imagens reconhecidas visualmente como sendo o numeral 2.

O tempo de processamento de cada etapa do sistema é mostrado na Tabela 6.1.

| Processo | Tempo(ms/caracter) |
|------------------------------|--------------------|
| Normalização | 41,35 |
| Suavização | 48,87 |
| Esqueletização | 33,83 |
| Pruning | 78,95 |
| Dilatação | 48,87 |
| Extração das características | 22,56 |
| Classificação | 3,76 |

Tabela 6.6: Tempo de processamento para Algoritmo Sintático Modificado

6.2 Redes Neurais

As imagens dos caracteres numéricos que compõem os conjuntos de treinamento e teste são primeiramente normalizadas para o tamanho 16×16 e em seguida suavizadas. Na próxima etapa, os vetores de características direcionais horizontal, vertical, diagonal esquerda e diagonal direita são calculados a partir das imagens normalizadas e suavizadas utilizando as máscaras *Kirsch*[15, 22, 43, 18]. Como etapa final do processo de extração de características direcionais, cada vetor de característica direcional 16×16 é comprimido para vetor de características 4×4 . Além desses vetores, na hora de considerar as características globais da imagem do caracter, utiliza-se a imagem 16×16 comprimida para 4×4 , como vetor de característica global. Como resultado, as características finais consistem de cinco vetores de dimensões 4×4 : quatro vetores 4×4 de características locais e um vetor 4×4 de características globais, conforme mencionado na seção 5.3.

Estas características são usadas como entrada da rede neural multicamada com propagação direta, na qual a camada de entrada e a camada escondida consistem de 80 neurônios e a camada de saída de 10 neurônios. Devido a problemas de ordem prática, não foi possível realizar o treinamento da rede neural recorrente para obter o conjunto de pesos sinápticos.

Na geração do conjunto de treinamento, após os processos de normalização e suavização foi realizada uma inspeção de todas as imagens dos caracteres, com objetivo de retirar do conjunto de treinamento as imagens de baixa qualidade, ou seja imagens onde é difícil o reconhecimento através de uma simples inspeção visual. A inclusão de imagens de baixa qualidade no conjunto de treinamento introduz informações não representativas ou até mesmo informações falsas.

Neste trabalho, as imagens dos caracteres utilizados para compor o conjunto de testes foram selecionadas sem nenhuma intervenção humana. Desta forma, é de se esperar que o conjunto de treinamento não contenha número suficiente de imagens que represente todos os dados de entrada. É de se esperar que o conjunto de testes contenha imagens espúrias, não representativas dos caracteres (as mesmas que foram eliminadas do conjunto de treinamento).

Visando avaliar o efeito do número de amostras do conjunto de treinamento no desempenho do sistema, a rede neural multicamada com propagação direta foi treinada com conjuntos de treinamentos contendo 100, 1000, 2000, 3000 e 4000 amostras, igualmente distribuídas entre os caracteres. O algoritmo de retropropagação (*back-propagation*) foi usado para treinar a rede e o processo iterativo de estimação foi interrompido quando o erro médio ε foi menor do que 0,01 sobre todo o conjunto de treinamento. Os valores dos parâmetros usados no treinamento são: de 0,4 para termo de ganho e de 0,6 para o momento. Os valores iniciais dos pesos sinápticos e dos limiares da rede devem ser uniformemente distribuídos dentro de uma pequena faixa. A razão desta pequena faixa é reduzir a probabilidade da rede saturar e produzir pequenos gradientes de erro. Entretanto, esta faixa não deve ser excessivamente pequena, de forma a produzir gradientes de erros mínimos, porque desta forma tem-se uma aprendizagem muito lenta. Os pesos sinápticos foram inicializados dentro da faixa $\left(-\frac{2,4}{F_i}, +\frac{2,4}{F_i}\right)$, onde F_i é o número total de entradas no neurônio i [44].

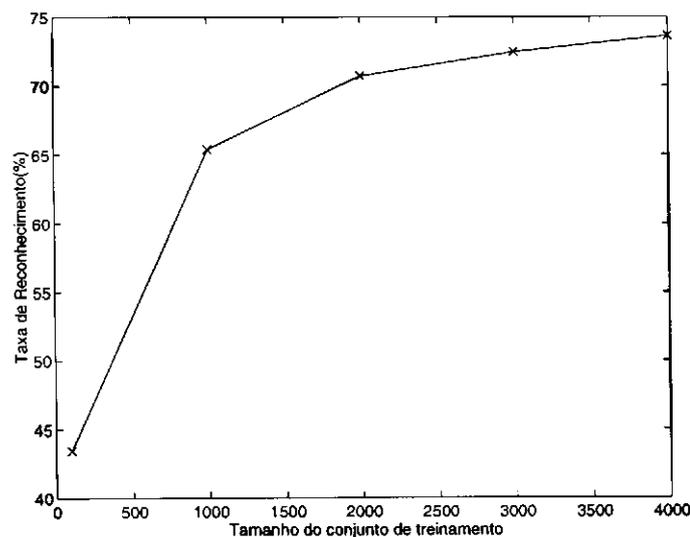


Figura 6.2: Taxa de reconhecimento versus tamanho do conjunto de treinamento

Os efeitos do tamanho do conjunto de treinamento no desempenho do sistema são ilustrados nos gráficos mostrados nas Figuras 6.2, 6.3, 6.4 e 6.5. As Tabelas 6.7, 6.8,

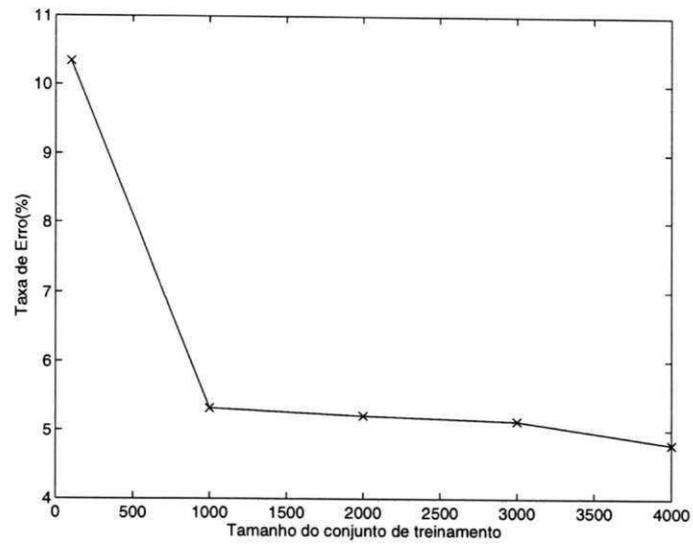


Figura 6.3: Taxa de erro versus tamanho do conjunto de treinamento

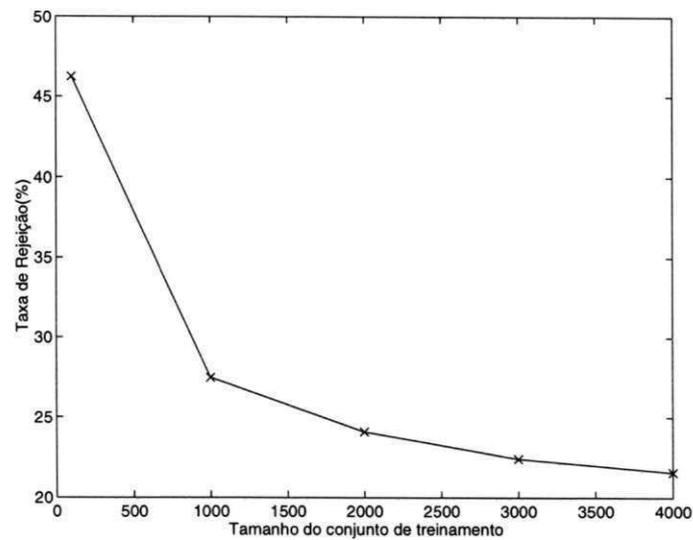


Figura 6.4: Taxa de rejeição versus tamanho do conjunto de treinamento

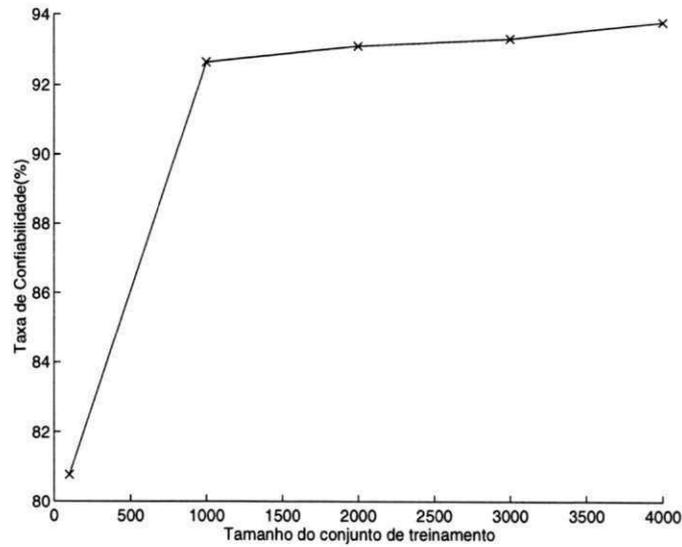


Figura 6.5: Taxa de confiabilidade versus tamanho do conjunto de treinamento

6.9, 6.10 e 6.11 mostram a matriz de confusão obtidas pelo sistema, quando a rede é treinada com 100, 1000, 2000, 3000 e 4000 amostras, respectivamente. Os desempenhos dos sistemas são mostrados nas Tabelas 6.12, 6.13, 6.14, 6.15 e 6.16.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|----|----|-----|-----|-----|----|----|-----|------|------|------|
| 0 | 128 | 2 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 130 | 128 | 7 |
| 1 | 0 | 182 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 2 | 77 | 182 | 6 |
| 2 | 0 | 10 | 47 | 0 | 5 | 0 | 32 | 2 | 0 | 0 | 169 | 47 | 49 |
| 3 | 0 | 9 | 1 | 66 | 0 | 11 | 0 | 2 | 0 | 7 | 169 | 66 | 30 |
| 4 | 0 | 1 | 0 | 0 | 206 | 0 | 1 | 0 | 0 | 4 | 53 | 206 | 6 |
| 5 | 0 | 1 | 0 | 3 | 10 | 103 | 2 | 0 | 1 | 6 | 139 | 103 | 23 |
| 6 | 0 | 11 | 0 | 0 | 1 | 1 | 181 | 0 | 0 | 0 | 71 | 181 | 13 |
| 7 | 0 | 10 | 0 | 0 | 6 | 0 | 0 | 52 | 0 | 60 | 137 | 52 | 76 |
| 8 | 0 | 4 | 0 | 0 | 15 | 7 | 2 | 0 | 34 | 11 | 192 | 34 | 39 |
| 9 | 0 | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 152 | 88 | 152 | 25 |
| Total | 128 | 230 | 48 | 69 | 272 | 125 | 220 | 56 | 35 | 242 | 1225 | 1151 | 274 |

Tabela 6.7: Matriz de confusão - MLP treinada com 100 amostras

O seguinte comportamento foi observado quando aumenta-se o número de amostras no conjunto de treinamento:

- Aumento da taxa de reconhecimento,

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|------|------|------|
| 0 | 208 | 2 | 0 | 0 | 0 | 2 | 1 | 0 | 2 | 0 | 50 | 208 | 7 |
| 1 | 0 | 231 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 31 | 231 | 3 |
| 2 | 0 | 2 | 108 | 1 | 1 | 2 | 6 | 2 | 3 | 1 | 139 | 108 | 18 |
| 3 | 0 | 6 | 3 | 174 | 0 | 3 | 0 | 0 | 0 | 3 | 76 | 174 | 15 |
| 4 | 0 | 0 | 0 | 0 | 233 | 0 | 3 | 0 | 1 | 0 | 28 | 233 | 4 |
| 5 | 0 | 1 | 0 | 2 | 0 | 196 | 1 | 0 | 2 | 3 | 60 | 196 | 9 |
| 6 | 0 | 3 | 0 | 0 | 0 | 3 | 223 | 0 | 1 | 0 | 35 | 223 | 7 |
| 7 | 0 | 8 | 0 | 0 | 5 | 0 | 0 | 78 | 0 | 28 | 146 | 78 | 41 |
| 8 | 0 | 5 | 0 | 1 | 4 | 2 | 3 | 0 | 168 | 2 | 80 | 168 | 17 |
| 9 | 0 | 1 | 0 | 0 | 18 | 0 | 0 | 0 | 1 | 162 | 83 | 162 | 20 |
| Total | 208 | 259 | 111 | 179 | 261 | 209 | 237 | 80 | 178 | 200 | 728 | 1781 | 141 |

Tabela 6.8: Matriz de confusão - MLP treinada com 1000 amostras

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 209 | 2 | 0 | 0 | 0 | 3 | 1 | 0 | 3 | 0 | 47 | 209 | 9 |
| 1 | 0 | 231 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 32 | 231 | 2 |
| 2 | 0 | 1 | 132 | 2 | 1 | 1 | 5 | 2 | 3 | 1 | 117 | 132 | 16 |
| 3 | 0 | 4 | 1 | 195 | 0 | 3 | 0 | 0 | 0 | 4 | 58 | 195 | 12 |
| 4 | 0 | 0 | 0 | 0 | 236 | 0 | 3 | 0 | 1 | 0 | 25 | 236 | 4 |
| 5 | 0 | 1 | 0 | 7 | 0 | 194 | 1 | 0 | 2 | 2 | 58 | 194 | 13 |
| 6 | 1 | 2 | 0 | 0 | 0 | 3 | 217 | 0 | 1 | 0 | 41 | 217 | 7 |
| 7 | 0 | 6 | 0 | 0 | 9 | 0 | 0 | 102 | 1 | 24 | 123 | 102 | 40 |
| 8 | 0 | 4 | 0 | 1 | 3 | 2 | 3 | 0 | 183 | 2 | 67 | 183 | 15 |
| 9 | 0 | 1 | 0 | 0 | 18 | 0 | 0 | 0 | 1 | 175 | 70 | 175 | 20 |
| Total | 210 | 252 | 133 | 205 | 267 | 207 | 230 | 104 | 195 | 209 | 638 | 1874 | 138 |

Tabela 6.9: Matriz de confusão - MLP treinada com 2000 amostras

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 217 | 1 | 0 | 0 | 0 | 4 | 1 | 0 | 2 | 0 | 40 | 217 | 8 |
| 1 | 0 | 231 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 33 | 231 | 1 |
| 2 | 0 | 1 | 144 | 2 | 0 | 1 | 3 | 2 | 2 | 1 | 109 | 144 | 12 |
| 3 | 0 | 3 | 1 | 201 | 0 | 3 | 0 | 0 | 0 | 3 | 54 | 201 | 10 |
| 4 | 0 | 0 | 0 | 0 | 235 | 0 | 2 | 0 | 1 | 1 | 26 | 235 | 4 |
| 5 | 0 | 1 | 0 | 8 | 0 | 199 | 1 | 0 | 2 | 2 | 52 | 199 | 14 |
| 6 | 1 | 2 | 0 | 0 | 2 | 2 | 220 | 0 | 2 | 0 | 36 | 220 | 9 |
| 7 | 0 | 9 | 0 | 0 | 6 | 0 | 0 | 105 | 0 | 30 | 115 | 105 | 45 |
| 8 | 0 | 3 | 0 | 1 | 3 | 2 | 3 | 0 | 185 | 1 | 67 | 185 | 13 |
| 9 | 0 | 1 | 0 | 0 | 18 | 0 | 0 | 0 | 1 | 183 | 62 | 183 | 20 |
| Total | 218 | 252 | 145 | 212 | 264 | 211 | 230 | 107 | 195 | 222 | 594 | 1920 | 136 |

Tabela 6.10: Matriz de confusão - MLP treinada com 3000 amostras

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 219 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 1 | 0 | 39 | 219 | 7 |
| 1 | 0 | 231 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 33 | 231 | 1 |
| 2 | 0 | 1 | 149 | 2 | 0 | 1 | 3 | 2 | 2 | 1 | 104 | 149 | 12 |
| 3 | 0 | 3 | 1 | 211 | 0 | 3 | 0 | 0 | 0 | 3 | 44 | 211 | 10 |
| 4 | 0 | 0 | 0 | 0 | 235 | 0 | 2 | 0 | 1 | 0 | 27 | 235 | 3 |
| 5 | 0 | 1 | 0 | 9 | 0 | 201 | 1 | 0 | 3 | 2 | 48 | 201 | 16 |
| 6 | 1 | 2 | 0 | 0 | 0 | 2 | 221 | 0 | 2 | 0 | 37 | 221 | 7 |
| 7 | 0 | 9 | 0 | 0 | 6 | 0 | 0 | 112 | 0 | 25 | 113 | 112 | 40 |
| 8 | 0 | 2 | 0 | 1 | 3 | 1 | 3 | 0 | 188 | 1 | 66 | 188 | 11 |
| 9 | 0 | 1 | 0 | 0 | 16 | 0 | 0 | 2 | 1 | 184 | 61 | 184 | 20 |
| Total | 220 | 250 | 150 | 223 | 260 | 212 | 232 | 116 | 198 | 217 | 572 | 1951 | 127 |

Tabela 6.11: Matriz de confusão - MLP treinada com 4000 amostras

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|----------|----------|
| 0 | 265 | 49,05 | 2,64 | 48,30 | 94,81 |
| 1 | 265 | 29,05 | 2,26 | 68,67 | 96,80 |
| 2 | 265 | 63,77 | 18,49 | 17,73 | 48,96 |
| 3 | 265 | 63,77 | 11,32 | 24,91 | 68,75 |
| 4 | 265 | 20 | 2,26 | 77,73 | 97,16 |
| 5 | 265 | 52,45 | 8,68 | 38,86 | 81,75 |
| 6 | 265 | 26,79 | 4,90 | 68,30 | 93,30 |
| 7 | 265 | 51,70 | 28,68 | 19,62 | 40,62 |
| 8 | 265 | 72,45 | 14,72 | 12,83 | 46,57 |
| 9 | 265 | 33,21 | 9,43 | 57,35 | 85,87 |
| Total | 2650 | 46,24 | 10,34 | 43,43 | 80,77 |

Tabela 6.12: Resultados do reconhecimento - MLP treinada com 100 amostras

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 18,87 | 2,64 | 78,49 | 96,74 |
| 1 | 265 | 11,70 | 1,13 | 87,16 | 98,72 |
| 2 | 265 | 52,45 | 6,79 | 40,75 | 86,88 |
| 3 | 265 | 28,68 | 5,66 | 65,66 | 92,06 |
| 4 | 265 | 10,57 | 1,51 | 87,92 | 98,31 |
| 5 | 265 | 22,64 | 3,40 | 73,96 | 95,61 |
| 6 | 265 | 13,21 | 2,64 | 84,15 | 96,96 |
| 7 | 265 | 55,09 | 15,47 | 29,43 | 65,54 |
| 8 | 265 | 30,18 | 6,41 | 63,39 | 90,81 |
| 9 | 265 | 31,32 | 7,55 | 61,13 | 98,01 |
| Total | 2650 | 27,49 | 5,32 | 65,37 | 92,66 |

Tabela 6.13: Resultados do reconhecimento - MLP treinada com 1000 amostras

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 17,74 | 3,40 | 78,86 | 95,87 |
| 1 | 265 | 12,07 | 0,75 | 87,16 | 99,14 |
| 2 | 265 | 44,15 | 6,04 | 49,81 | 89,19 |
| 3 | 265 | 21,89 | 4,52 | 73,58 | 94,20 |
| 4 | 265 | 9,44 | 1,51 | 89,06 | 98,33 |
| 5 | 265 | 21,89 | 4,91 | 73,21 | 93,72 |
| 6 | 265 | 15,47 | 2,64 | 81,89 | 96,87 |
| 7 | 265 | 46,41 | 15,09 | 38,49 | 71,83 |
| 8 | 265 | 25,28 | 5,66 | 69,06 | 92,42 |
| 9 | 265 | 26,42 | 7,55 | 66,04 | 89,74 |
| Total | 2650 | 24,08 | 5,21 | 70,72 | 93,14 |

Tabela 6.14: Resultados do reconhecimento - MLP treinada com 2000 amostras

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 15,04 | 3,02 | 81,87 | 96,44 |
| 1 | 265 | 12,45 | 0,38 | 87,17 | 99,57 |
| 2 | 265 | 41,13 | 4,53 | 54,33 | 92,31 |
| 3 | 265 | 20,38 | 3,77 | 75,85 | 95,26 |
| 4 | 265 | 9,81 | 1,51 | 88,68 | 98,33 |
| 5 | 265 | 19,62 | 5,28 | 75,09 | 93,43 |
| 6 | 265 | 13,58 | 3,40 | 83,02 | 96,07 |
| 7 | 265 | 43,40 | 16,98 | 39,62 | 70,00 |
| 8 | 265 | 25,28 | 4,91 | 69,81 | 93,43 |
| 9 | 265 | 23,39 | 7,54 | 69,05 | 90,14 |
| Total | 2650 | 22,41 | 5,13 | 72,45 | 93,38 |

Tabela 6.15: Resultados do reconhecimento - MLP treinada com 3000 amostras

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 14,72 | 2,64 | 82,64 | 96,90 |
| 1 | 265 | 12,45 | 0,38 | 87,17 | 99,57 |
| 2 | 265 | 39,25 | 4,53 | 56,23 | 92,54 |
| 3 | 265 | 16,60 | 3,77 | 79,62 | 95,47 |
| 4 | 265 | 10,19 | 1,13 | 88,68 | 98,74 |
| 5 | 265 | 18,11 | 6,04 | 75,85 | 92,62 |
| 6 | 265 | 13,96 | 2,64 | 83,40 | 96,93 |
| 7 | 265 | 42,64 | 15,09 | 42,26 | 73,68 |
| 8 | 265 | 24,91 | 4,15 | 70,94 | 94,47 |
| 9 | 265 | 23,02 | 7,55 | 69,43 | 90,19 |
| Total | 2650 | 21,58 | 4,79 | 73,62 | 93,88 |

Tabela 6.16: Resultados do reconhecimento - MLP treinada com 4000 amostras

- Diminuição da taxa de erro,
- Diminuição da taxa de rejeição,
- Aumento da taxa de confiabilidade do sistema.

Quando o número de amostras do conjunto de treinamento aumenta de 100 para 1000, ocorre um aumento significativo na taxa de reconhecimento e de confiabilidade do sistema e uma diminuição na taxa de erro e de rejeição, apesar da taxa de erro para o dígito 0 permanecer constante.

O mesmo comportamento ocorre quando aumenta-se o número de amostras de 1000 para 2000 no conjunto de treinamento, embora com menos intensidade. Entretanto, vale salientar os resultados obtidos no reconhecimento dos dígitos 0, 1, 5 e 6. A confiabilidade do sistema para o reconhecimento dos dígitos 0 e 5 sofreu uma diminuição com o aumento do conjunto de treinamento. O percentual de dígitos 0 reconhecidos erroneamente pelo sistema aumentou de 2,64% para 3,40% e o percentual dos dígitos 5 reconhecidos erroneamente pelo sistema aumentou de 3,40% para 4,91%. Apesar destes resultados a taxa de reconhecimento e de confiabilidade total do sistema aumentaram e as taxas de substituição e de erro diminuíram.

Os efeitos resultantes do aumento no número de amostras do conjunto de treinamento de 2000 para 3000 sobre o desempenho do sistema foram mínimos. A taxa de reconhecimento aumentou de 70,72% para 72,45%, um aumento de menos de 2%, embora tenham sido adicionados ao conjunto de treinamento 1000 amostras. Comparando-se

as Tabelas 6.14 e 6.15, percebe-se que ocorreu um aumento nas taxas de erro dos dígitos 5, 6 e 7 e uma diminuição na taxa de reconhecimento do dígito 4. Resultados semelhantes foram encontrados no desempenho dos sistema quando são adicionados 1000 amostras ao conjunto de treinamento contendo 3000 amostras.

Atráves da comparação das Tabelas 6.12, 6.13, 6.14, 6.15 e 6.16, percebe-se que em geral as taxas de confiabilidade e de reconhecimento do sistema aumentam e que as taxas de rejeição e de erro diminuem quando aumenta-se o número de amostras do conjunto de treinamento. Entretanto, uma análise das taxas individuais de cada dígito mostra que este comportamento não é regra geral. Por vezes ocorre, por exemplo, uma diminuição na taxa de reconhecimento ou um aumento na taxa de erro de um dígito. Este comportamento contraditório no desempenho do sistema para alguns dígitos ao aumentar-se o número de amostras do conjunto de treinamento, deve-se à inclusão de informações não representativas ou até mesmo à inclusão de informações falsas no conjunto de treinamento.

Dois outros conjuntos de pesos sinápticos foram gerados utilizando o conjunto de treinamento contendo 1000 e 4000 amostras, entretanto o processo iterativo de estimação dos pesos não foi interrompido antes do erro médio ϵ sobre todo o conjunto de treinamento se tornar menor que 0,001. Estes últimos conjuntos de pesos sinápticos foram gerados visando avaliar o desempenho do sistema em relação ao critério de parada do treinamento. As Tabelas 6.19, 6.17, 6.20, 6.18 mostram as matrizes de confusão e os resultados obtidos quando a rede neural é treinada com 1000 e 4000 amostras, utilizando como critério de parada um erro médio ϵ de 0,001.

Comparando os resultados listados nas Tabelas 6.18 e 6.16, percebe-se um desempenho superior do sistema quando este é treinado com 1000 amostras com critério de parada $\epsilon = 0,001$ do que quando a rede neural é treinada com 4000 amostras e $\epsilon = 0,01$. Entretanto, a confiabilidade diminui em virtude do aumento da taxa de erro do sistema.

O melhor desempenho do sistema foi obtido quando este utiliza os pesos sinápticos gerados pelo treinamento de 4000 amostras com critério de parada $\epsilon = 0,001$. Obtem-se para este sistema uma taxa de reconhecimento de 84,72%, 4,52% de erro, 10,75% de rejeição e 95,47% de confiabilidade.

Os resultados ilustrados nas Tabelas 6.12, 6.13, 6.14, 6.15, 6.16, 6.19 e 6.17 mostram que a confiabilidade do sistema está mais ligada ao número de amostras do conjunto de treinamento, enquanto a taxa de reconhecimento está mais ligada ao critério de parada do treinamento.

Outro fato que merece destaque é a taxa de reconhecimento dos dígitos 2 e 7 do

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 216 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3 | 0 | 44 | 216 | 5 |
| 1 | 0 | 251 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 251 | 0 |
| 2 | 1 | 1 | 200 | 1 | 1 | 0 | 3 | 2 | 1 | 0 | 55 | 200 | 10 |
| 3 | 0 | 5 | 3 | 214 | 0 | 3 | 0 | 0 | 0 | 2 | 38 | 214 | 13 |
| 4 | 0 | 0 | 0 | 0 | 250 | 0 | 2 | 0 | 2 | 0 | 11 | 250 | 4 |
| 5 | 0 | 1 | 0 | 10 | 0 | 169 | 1 | 0 | 3 | 5 | 76 | 169 | 20 |
| 6 | 2 | 2 | 0 | 0 | 1 | 1 | 227 | 0 | 2 | 0 | 30 | 227 | 8 |
| 7 | 0 | 6 | 0 | 0 | 10 | 0 | 0 | 100 | 1 | 40 | 108 | 100 | 57 |
| 8 | 0 | 3 | 2 | 2 | 2 | 0 | 3 | 0 | 187 | 3 | 63 | 187 | 15 |
| 9 | 0 | 3 | 0 | 0 | 33 | 0 | 0 | 1 | 0 | 195 | 33 | 195 | 37 |
| Total | 219 | 272 | 205 | 227 | 297 | 174 | 237 | 103 | 199 | 245 | 472 | 2009 | 169 |

Tabela 6.17: Matriz de confusão - MLP treinada com 1000 amostras e $\varepsilon = 0,001$

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 16,60 | 1,89 | 81,51 | 97,74 |
| 1 | 265 | 5,28 | 0 | 94,72 | 100 |
| 2 | 265 | 20,75 | 3,77 | 75,47 | 95,24 |
| 3 | 265 | 14,34 | 4,91 | 80,75 | 94,27 |
| 4 | 265 | 4,15 | 1,51 | 94,34 | 98,42 |
| 5 | 265 | 28,68 | 7,55 | 63,77 | 89,42 |
| 6 | 265 | 11,32 | 3,02 | 85,66 | 96,60 |
| 7 | 265 | 40,75 | 21,51 | 37,74 | 63,69 |
| 8 | 265 | 23,77 | 5,66 | 70,57 | 92,57 |
| 9 | 265 | 12,45 | 13,96 | 73,58 | 84,05 |
| Total | 2650 | 17,81 | 6,38 | 75,81 | 92,24 |

Tabela 6.18: Resultados do reconhecimento - MLP treinada com 1000 amostras e $\varepsilon = 0,001$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Rej. | Rec. | Err. |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| 0 | 241 | 0 | 0 | 0 | 0 | 4 | 3 | 1 | 0 | 0 | 17 | 241 | 7 |
| 1 | 0 | 256 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 8 | 256 | 1 |
| 2 | 1 | 0 | 220 | 0 | 1 | 2 | 2 | 3 | 2 | 0 | 34 | 220 | 11 |
| 3 | 0 | 1 | 0 | 235 | 0 | 4 | 0 | 0 | 0 | 3 | 22 | 235 | 8 |
| 4 | 1 | 0 | 0 | 0 | 256 | 0 | 1 | 0 | 0 | 0 | 7 | 256 | 2 |
| 5 | 0 | 0 | 0 | 5 | 0 | 205 | 1 | 0 | 2 | 1 | 51 | 205 | 9 |
| 6 | 2 | 0 | 0 | 0 | 1 | 4 | 236 | 1 | 4 | 0 | 17 | 236 | 12 |
| 7 | 1 | 3 | 0 | 0 | 5 | 0 | 0 | 181 | 0 | 18 | 57 | 181 | 27 |
| 8 | 0 | 1 | 2 | 1 | 1 | 2 | 4 | 0 | 225 | 0 | 29 | 225 | 11 |
| 9 | 0 | 0 | 0 | 0 | 26 | 1 | 0 | 5 | 0 | 190 | 43 | 190 | 32 |
| Total | 246 | 261 | 222 | 241 | 290 | 222 | 247 | 191 | 233 | 212 | 285 | 2245 | 119 |

Tabela 6.19: Matriz de confusão - MLP treinada com 4000 amostras e $\epsilon = 0,001$

| Numeral | Total | Rej.(%) | Err.(%) | Rec.(%) | Conf.(%) |
|---------|-------|---------|---------|---------|----------|
| 0 | 265 | 6,41 | 2,64 | 90,94 | 97,18 |
| 1 | 265 | 3,02 | 0,38 | 96,60 | 99,61 |
| 2 | 265 | 12,83 | 4,14 | 83,02 | 95,24 |
| 3 | 265 | 8,30 | 3,02 | 88,67 | 96,70 |
| 4 | 265 | 2,64 | 0,75 | 96,60 | 99,22 |
| 5 | 265 | 19,24 | 3,40 | 77,36 | 96,79 |
| 6 | 265 | 6,41 | 4,53 | 89,06 | 95,16 |
| 7 | 265 | 21,51 | 10,19 | 68,31 | 87,02 |
| 8 | 265 | 10,94 | 4,15 | 84,31 | 95,34 |
| 9 | 265 | 16,23 | 12,07 | 71,70 | 85,58 |
| Total | 2650 | 10,75 | 4,53 | 84,72 | 94,92 |

Tabela 6.20: Resultados do reconhecimento - MLP treinada com 4000 amostras e $\epsilon = 0,001$

sistema, mostrados nas Tabelas 6.12, 6.13, 6.14, 6.15, 6.16, 6.19 e 6.17. Estes dígitos possuem as mais baixas taxas de reconhecimento. Este fato deve-se à baixa diversidade de estilos encontrada no conjunto de treinamento ou seja, no conjunto de treinamento não existe número de amostras dos dígitos 2 e 7 que represente todos os possíveis estilos de escrita encontrados no conjunto de teste.

O tempo de processamento de cada etapa do sistema é mostrado na Tabela 6.21.

| Processo | Tempo(ms/caracter) |
|-----------------------------|--------------------|
| Extração de características | 55,56 |
| Classificação | 3,66 |

Tabela 6.21: Tempo de proc. do sistema de rec. baseado em redes neurais

A tabela 6.22 mostra os desempenhos de alguns sistemas de reconhecimento de caracteres numéricos manuscritos existentes na literatura que utilizam o banco de dados do CENPARMI no treinamento e nos testes.

| Método | Rec. | Err. |
|----------|-------|------|
| Lam [40] | 93,15 | 4,60 |
| Suen[5] | 86,05 | 2,25 |
| Suen [5] | 93,05 | 0,00 |
| Cho [18] | 89,05 | 7,00 |

Tabela 6.22: Desempenhos de alguns sistemas existentes na literatura

6.3 Conclusão

Neste capítulo foram apresentados os testes realizados e os resultados obtidos com os algoritmos implementados, utilizando como conjunto de testes os caracteres obtidos do banco de dados do CENPARMI. Foi realizada também, uma análise comparativa dos resultados obtidos.

No próximo capítulo são apresentadas a comparação entre os métodos implementados; análise sintática e redes neurais, as conclusões obtidas em relação ao conjunto do trabalho, as sugestões para desenvolvimentos subsequentes baseado no estudo descrito, além de suas contribuições para a área.

Capítulo 7

Conclusão

A análise de imagens de documentos é um excitante campo de pesquisa, que tem atraído interesse de pesquisadores de diversas áreas científicas e tecnológicas. Neste contexto, o reconhecimento de caracteres numéricos manuscritos tem sido bastante estudado, apresentando desempenhos satisfatórios, isoladamente ou como parte integrante de sistemas mais complexos.

Durante o desenvolvimento deste trabalho, foram implementados dois sistemas de reconhecimento de caracteres numéricos manuscritos. O primeiro baseado em análise sintática e o segundo baseado em redes neurais artificiais. Durante a fase de implementação e de avaliação do algoritmo sintático foram propostas modificações na estrutura do classificador.

As avaliações dos algoritmos implementados; algoritmo sintático, algoritmo sintático modificado e redes neurais multicamadas com propagação direta, foram realizadas utilizando o conjunto de testes obtidos com os caracteres provenientes do banco de dados do CENPARMI.

Os testes realizados com o algoritmo sintático original, não apresentaram resultados satisfatórios. Nestes experimentos, ficou evidenciado que as regras sintáticas de decisão utilizadas por este algoritmo não são adequadas para o estilo de escrita dos dígitos encontrados no banco de dados utilizado. Os resultados obtidos dos testes realizados com o algoritmo sintático modificado mostram um aumento superior a 100% na taxa de reconhecimento do sistema com relação à taxa de reconhecimento do sistema original. Entretanto, os resultados obtidos não são, ainda, satisfatórios.

Em sistemas que utilizam algoritmos sintáticos, durante o estágio de treinamento, os caracteres que compõem o conjunto de treinamento são analisados por humanos. Através deste exame e da análise efetuada, um número de características ou medidas são selecionadas e a seguir definem-se os métodos para extração destas características.

O conhecimento do classificador é dado pela experiência humana, na forma de decisão em árvore. No estágio de reconhecimento, o computador compara as características obtidas do caracter desconhecido com as informações armazenadas no classificador e decide a qual classe pertence o caracter. Os problemas existentes nestes métodos são: 1) o reconhecimento independente do treinamento: não é fácil melhorar o classificador usando o conhecimento obtido através da análise dos erros de reconhecimento; 2) o conhecimento armazenado no classificador depende do conjunto de treinamento e das características encontradas pelo especialista humano; 3) o especialista humano pode decidir excluir um número de amostras de baixa qualidade, o que introduz subjetividade no projeto do classificador; 4) as características são predefinidas e dependem do conjunto de treinamento utilizado, se o conjunto de treinamento não for representativo, o método não irá trabalhar bem. As regras utilizadas pelo classificador podem continuar válidas para diferentes aplicações, desde que o conjunto de referência utilizado seja válido. Estes problemas implicam em um desempenho limitado para algoritmos sintáticos.

Os testes realizados com o sistema de reconhecimento de caracteres numéricos manuscritos utilizando o classificador de redes neurais artificiais apresentou resultados melhores, comparados aos sistemas de classificação baseados em análise sintática. Neste experimento, ficou caracterizada a eficiência do algoritmo de retropropagação (*backpropagation*) no treinamento da rede neural. Os problemas existentes nestes métodos são: 1) dificuldade na detecção de erros no algoritmo e na sua implementação; 2) em muitos problemas práticos o classificador baseado em redes neurais não converge para uma solução ótima, caindo em mínimos locais; e se convergir o tempo necessário para a sua convergência pode ser elevado; 3) a escolha dos valores iniciais para os pesos sinápticos, pois valores elevados podem causar a saturação da rede; 4) necessidade que todos os numerais a serem apresentados a rede tenham as mesmas dimensões, evitando assim, que a rede tenha pesos sinápticos com valores nulos que bloqueiem as informações que podem resultar dos pixels correspondentes.

Embora a taxa de reconhecimento dos dígitos 2 e 7 tenha sido relativamente baixa em comparação à dos demais dígitos, a taxa geral de reconhecimento do sistema foi satisfatória. A baixa taxa de reconhecimento dos dígitos 2 e 7 deve-se à pequena diversidade de estilos de escritas destes dígitos encontrados no conjunto de treinamento. Observou-se que a taxa de reconhecimento do sistema está diretamente relacionada às amostras do conjunto de treinamento e que a taxa de confiabilidade está diretamente relacionada ao parâmetro utilizado para interromper o treinamento (erro médio).

Com relação aos métodos de reconhecimento implementados, pode-se concluir que o

tempo de operação requerido nos testes realizados é relativamente pequeno, tipicamente algumas frações de segundos por caracter, excluindo o tempo de pré-processamento e extração das características. No caso do algoritmo sintático esta eficiência, deve-se ao esquema de classificação em árvore. Sobre a complexidade, pode-se dizer que o sistema de classificação utilizando redes neurais requer uma etapa de pré-processamento com menor complexidade do que os sistemas de classificação baseados em análise sintática e são menos susceptíveis às deformações na forma do traçado do numeral ocasionalmente introduzidas pelos algoritmos que compõem o pré-processamento. Com relação à robustez, pode-se dizer que o algoritmo baseado em redes neurais é mais tolerante a imagens de caracteres de baixa qualidade, devido a ruídos provavelmente introduzidos durante as fases de escrita e aquisição. O parâmetro tempo de processamento do sistema é muito relativo, depende da máquina, da aplicação, do tamanho das imagens de entrada, etc. O ideal seria comparar todos os sistemas rodando na mesma máquina.

7.1 Contribuições

Podem ser citadas como desenvolvimentos originais resultantes do trabalho até agora realizado:

- Processamento e normalização do banco de caracteres do CENPARMI, colocando o mesmo em um formato que possibilita o seu uso no treinamento e teste de sistemas de reconhecimento. Embora bastante citado e utilizado em trabalhos internacionais este banco de dados nunca havia sido utilizado por pesquisadores no Brasil.
- Elaboração de um algoritmo sintático ampliado para o reconhecimento de caracteres manuscritos. Este algoritmo atingiu desempenho bastante superior ao algoritmo original, para o banco de dados do CENPARMI.
- Avaliação comparativa objetiva do desempenho entre métodos sintáticos e redes neurais, no reconhecimento de caracteres numéricos manuscritos. Tanto quanto foi possível investigar, esta comparação é inédita. O fato de que foi feita usando o banco de dados do CENPARMI a torna ainda mais valiosa, pois permite a fácil comparação dos resultados obtidos com outros trabalhos publicados internacionalmente.
- Derivação de um algoritmo de treinamento para uma rede neural recorrente. Embora a estrutura da rede neural tenha sido proposta por Lec[43], nenhum

algoritmo de treinamento é apresentado no trabalho ou em outros trabalhos pesquisados.

7.2 Sugestões

Para continuação das atividades de pesquisa desenvolvidas, podem ser citadas as seguintes sugestões:

- Investigação de técnicas de normalização, visando melhorar a qualidade das imagens normalizadas.
- Avaliação do desempenho destes algoritmos no reconhecimento de caracteres pertencentes a outra base de dados, que contenha dígitos coletados em outra região (de preferência brasileiro), desta forma contendo estilos de escrita distintos dos aqui utilizados.
- Investigação de técnicas de afinamento seletivo do caracter, que produzam o esqueleto do caracter e eliminem os segmentos espúrios de determinado comprimento.
- Investigar técnicas de segmentação que possibilitem o reconhecimento de caracteres não isolados.
- Estudo aprofundado da topologia dos sistemas baseados em redes neurais artificiais, melhorando a etapa de extração de características, de maneira a se obter uma melhor taxa de reconhecimento.
- Análise aprofundada dos caracteres que compõem o conjunto de treinamento, para obtenção de características e regras de classificação mais apuradas, que permitam uma melhor descrição dos dígitos que compõem a base de dados.

Conclue-se esta dissertação salientando sua contribuição para o início de uma nova linha de pesquisa dentro do grupo de pesquisa “Análise e Classificação de Sinais e Imagens” deste departamento, que servirá de incentivo a outros estudantes da área, permitindo o desenvolvimento de novos trabalhos nesta linha, ainda não muito explorada no país.

Bibliografia

- [1] Sabourin, R. Off-line signature verification: Recent advances and perspectives. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 84–98. Springer, November 1997.
- [2] Kasturi, R. and Luo, H. Research advances in graphics recognition: An update. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 99–110. Springer, November 1997.
- [3] Belaid, A. Future trends in retrospective document conversion. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 111–128. Springer, November 1997.
- [4] Suen, C. T. and Berthod, M. and Mori, S. Automatic recognition of handprinted characters - the state of the art. *Proceeding of IEEE*, 68(4):469–487, april 1980.
- [5] Suen, C. Y. and Nadal, C. and Legault, R. and Mai, T. A. and Lam, L. . “Computer Recognition of Unconstrained Handwritten Numerals. *Proceeding of the IEEE*, 80(7):1162–1180, July 1992.
- [6] Gomes, H. M. Investigação de técnicas automáticas para o reconhecimento off-line de assinaturas. Master’s thesis, Universidade Federal de Pernambuco.
- [7] Lu, Y. and Shridhar, M. Character Segmentation in Handwritten Words - An Overview. *Pattern Recognition*, 29(1):77–96, 1996.
- [8] Lu, Y. Machine printed Character Segmentation - An Overview. *Pattern Recognition*, 28(1):67–80, 1995.
- [9] Gomes, N. R. Algoritmo sequencial para reconhecimento de numerais manuscritos desconectados utilizando redes neurais. Master’s thesis, Universidade Estadual de Campinas- Faculdade de Engenharia Eletrica- Departamento de Computacao, 1996.

- [10] Kimura, F. and Shridhar, M. Handwritten Numerical Recognition Based on Multiple Algorithms. *Pattern Recognition*, 24(10):969–983, 1991.
- [11] El-Yacoubi, A. and Gilloux, M. and Sabourin, R. and Suen, C. Y. . Objective evaluation of the discriminant power of features in an hmm-based word recognition system. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 60–73. Springer, November 1997.
- [12] Suen, C. Y. and Li, Z. C. Crucial combinations for the recognition of handwritten letters. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 19–39. Springer, November 1997.
- [13] Amin, A. Recognition of printed and handwritten arabic characters. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 40–59. Springer, November 1997.
- [14] Senior, A. W. *Off-Line Cursive Handwriting Recognition Using Recurrent Neural Networks*. PhD thesis, University of Cambridge, September 1994.
- [15] Parker, J. R. *Algorithms For Image Processing and Computer Vision*. Wiley Computer Publishing, 1997.
- [16] Mantas, J. An overview of character recognition methodologies. *Pattern Recognition*, 19(6):425–430, 1986.
- [17] Mori, S. and Suen, C. Y. and Fellow and Yamamoto, K. Historical Review of OCR Research and Development. *Proceeding of IEEE*, 80(7):1029–1058, July 1992.
- [18] Cho, S.-B. Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numeral. *Transactions on Neural Networks*, 8(1):43–53, January 1997.
- [19] Tsirikolias, K. and Mertzios, B. G. Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition. *Pattern Recognition*, 26(6):877–882, 1993.
- [20] Gomes, N. R. and Ling, L. L. . Um Algoritmo Sequencial Utilizando Redes Neurais Para o Reconhecimento de numerais manuscritos. In *XIV- Simposio brasileiro de telecomunicacoes*, 1996.
- [21] Chi, Z. and Wu, J. and Yan, H. Handwritten Numeral Recognition using Self-Organizing Maps and Fuzzy Rules. *Pattern Recognition*, 28(1):59–66, 1995.

- [22] Knerr, S. and Personnaz, L. and Dreyfus, G. Handwritten Digit Recognition by Neural Networks with Single-Layer Training. *IEEE trans. Neural Networks*, 3:962–968, 1992.
- [23] Wang, J. and Jean, J. Segmentation of Merged Characters By Neural Networks and Shortest Path. *Pattern Recognition*, 27(5):649–658, 1994.
- [24] Liang, S. and Shridhar, M. and Ahmadi, M. Segmentation Of Touching Characters in Printed Document Recognition. *Pattern Recognition*, 27(6):825–840, 1994.
- [25] Mai, T. A. and Suen, C. Y. A generalized knowledge-based system for the recognition of unconstrained handwritten numerals. *IEEE- Transactions on Systems, Man, and Cybernetics*, 20(4):835–848, July 1990.
- [26] Trier, O. D. and Jains, A. K. and Taxt, T. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.
- [27] Shridhar, M. and Badreldin, A. High accuracy character recognition algorithm using fourier and topological descriptors. *Pattern Recognition*, 17(5):515–524, 1984.
- [28] Ahmed, J. and Althofer, K. A single layer neural network architecture for handwritten digits recognition. In *International Conference on Engineering Applications of Neural Networks*, pages 67–70, August 1995.
- [29] Freixinho, M. A. P. and Salazar, F. A. and Vellasco, M. and Pacheco, M. A. Handwritten Digits Classification Using Neural Networks. In *International Conference on Engineering Applications of Neural Networks*, pages 71–74, August 1995.
- [30] Cao, J. and Ahmadi, M. and Shridhar, M. Recognition of handwritten numerals with multiple feature and multistage classifier. *Pattern Recognition*, 28(2):153–160, 1995.
- [31] Wang, G.-E. and Wang, J.-F. A new hierarchical approach for recognition of unconstrained handwritten numerals. *IEEE- Transactions on Consumer Electronics*, 40(3):428–436, 1994.
- [32] Shridhar, M. and Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition*, 19(1):1–12, 1986.

- [33] Shridhar, M. and Badreldin, A. A high-accuracy syntactic recognition algorithm for handwritten numerals. *IEEE- Transactions on Systems, Man, and Cybernetics*, Smc-15(1):152–158, January 1985.
- [34] Duerr, B. and Haettich, W. and Tropf, H. and Winkler, G. A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals. *Pattern Recognition*, 12:189–199, 1980.
- [35] Dimauro, G. and Impedovo S. and Pirlo, G. and Salzo, A. Handwriting recognition: State of the art and future trends. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 1–18. Springer, November 1997.
- [36] Facon, J. *Morfologia Matemática: Teoria e Exemplos*. PUC-PR, 1996.
- [37] Frucci, M. and Marcelli, A. Contour pixel classification for character skeletonization. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 141–152. Springer, November 1997.
- [38] Borgefors, G. and Ramella, G. and Baja, G. S. . Using binary pyramids to create multi-resolution shape descriptors. In N. Murshed and F. Bortolozzi, editors, *Advances in Document Image Analysis*, pages 129–140. Springer, November 1997.
- [39] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*. Addison-Wesley, 1992.
- [40] Lam, L. and Suen, C. Y. Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. *Pattern Recognition*, 21(1):19–31, 1988.
- [41] Lam, L. and Suen, C. Y. An evaluation of parallel thinning algorithms for character recognition. *IEEE- Transactions on Pattern Analysis and Machine Intelligence*, 17(9):914–919, September 1995.
- [42] Yuceer, C. and Oflazer, K. A rotation, scaling, and translation invariant pattern classification system. *Pattern Recognition*, 26(5):687–710, 1993.
- [43] Lee, S.-W. and Song, H.-H. A new recurrent neural-network architecture for visual pattern recognition. *IEEE - Transactions on Neural Networks*, 8(2):331–339, 1997.
- [44] Haykin, S. *Neural Networks - A Comprehensive Foundation*. IEEE Computer Society Press, 1994.

- [45] Beable, R. and Jackson, T. *Neural Computing: An Introduction*. Jop publishing LTd, 1992.
- [46] Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computa*, 1:270-280, 1989.