

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM ESTUDO SOBRE RECONHECIMENTO
VISUAL DE CARACTERES ATRAVÉS DE
REDES NEURAIS

5

por

FERNANDO SANTOS OSÓRIO

Dissertação submetida como requisito parcial para
a obtenção do grau de Mestre em
Ciência da Computação

Prof. Anatólio Laschuk

Orientador



Porto Alegre, outubro de 1991

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

2.

CATALOGAÇÃO DA PUBLICAÇÃO

Osório, Fernando Santos

Um estudo sobre o reconhecimento visual de caracteres através de redes neurais. Porto Alegre, CPGCC da UFRGS, 1991.

302p.

Dissertação (mestrado), Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Ciências da Computação, Porto Alegre, BR-RS, 1991.
Orientador: Laschuk, Anatólio.

Dissertação: Processamento de Imagens: OCR
Reconhecimento de Caracteres: Redes Neurais

"O mundo está dividido entre aqueles que fazem as coisas e aqueles que levam as glórias. Tente ficar no primeiro grupo, porque ali há menos competição."

Lord Chesterfield

"O assunto mais importante do mundo pode ser simplificado até o ponto em que todos possam apreciá-lo e compreendê-lo. Isso é, ou deveria ser, a mais elevada forma de arte."

Charles Chaplin

"Considero a ciência simplesmente como uma maneira de entender como o mundo é e por que ele é assim. Em qualquer época, nosso conhecimento científico é apenas a modernização do nosso entendimento. Não acredito em verdades absolutas. Tenho medo delas porque bloqueiam a busca de um melhor entendimento. Sempre que pensamos ter chegado à resposta final, o progresso, a ciência e o entendimento acabam. No entanto, o conhecimento que nos rodeia não é um objetivo a ser alcançado por si só. Ele deve ser perseguido, acredito, para tornar o nosso mundo um lugar melhor e a vida mais gratificante."

Eliyahu M. Goldratt

Aos meus pais,
à minha amada esposa.

AGRADECIMENTOS

Agradeço ao meu orientador, professor Anatólio Laschuk, pela paciência, apoio e confiança em mim e no meu trabalho;

A todo corpo docente do CPGCC, em especial aos professores Philippe Navaux e Dante Barone, os quais acompanharam e incentivaram meus trabalhos junto ao CPGCC.

Aos funcionários do II, pela ajuda prestada e o "apoio técnico" necessário para o bom andamento de meu trabalho. Agradeço em especial aos funcionários da secretaria, dos laboratórios e em destaque aos da biblioteca, indispensáveis a um bom trabalho científico;

Aos colegas da pós-graduação pelo seu companherismo, incentivo, e pelas inúmeras discussões e sugestões dadas durante a evolução deste trabalho. Agradeço especialmente a: Alceu Frigeri, Gilberto Marchioro, Fernando Moraes, Carlos Eduardo Pereira, Marcelo Walter, Gladimir Baranovsky, Alexandre Casacurta, Alexandre Agustini e Alex Guazzelli.

Em especial a Adelmo Cechin e Remis Balaniuk, meus verdadeiros "mestres", que me apresentaram as redes neurais e me incentivaram na escolha do tema abordado em minha dissertação de mestrado. O estudo das redes neurais junto ao CPGCC-UFRGS é fruto do trabalho e dedicação destas duas pessoas, aliadas ao incentivo dado pelo prof. Navaux;

A CAPES e CNPQ pelo auxílio financeiro, na forma de bolsa de estudos;

A UFRGS e ao CPGCC pelas condições de estudo e trabalho oferecidas;

A minha família e meus amigos, os quais por mais difícil que tenha sido esta caminhada, nunca me deixaram sem o apoio tão necessário para que eu pudesse atingir meus objetivos;

Obrigado a todos vocês !

SUMÁRIO

GLOSSARIO	9
LISTA DE ABREVIATURAS	10
LISTA DE FIGURAS	11
LISTA DE TABELAS	15
RESUMO	16
ABSTRACT	18
1 INTRODUÇÃO	20
2 PROCESSAMENTO DE IMAGENS	24
2.1 Introdução	24
2.2 Imagem Digital	26
2.2.1 Digitalização	27
2.2.1.1 Digitalizadores	28
2.2.1.2 Amostragem	30
2.2.1.3 Quantização	33
2.2.1.4 Classes de Imagens	35
2.2.2 Armazenamento	38
2.2.2.1 Compactação	39
2.2.2.2 Formato de Arquivos	40
2.2.3 Exibição	42
2.2.3.1 <i>Halftoning</i>	42
2.2.3.2 <i>Dithering</i>	46
2.3 Tratamento de Imagens	47
2.3.1 Histograma	48
2.3.2 Modificação da Escala de Cinza	50
2.3.2.1 Equalização de Histograma	50
2.3.2.2 Limiarização	51
2.3.3 Filtros	53
2.3.3.1 Operações no Domínio Espaço	53
2.3.3.1.1 Convolução	54
2.3.3.2 Operações no Domínio Freqüência	57

2.4 Reconhecimento de Padrões	57
2.4.1 Obtenção de Descrições	59
2.4.2 Obtenção de Classes	59
2.4.2.1 Análise de Padrões	63
2.4.2.2 Análise de Elementos	63
2.4.2.3 Análise de Contornos	63
2.4.2.4 Casamento de Padrões	64
2.4.2.5 Abordagem do Discriminante	65
2.4.2.6 Abordagem Sintática	65
2.4.2.7 Avaliação do Processo de Classificação	65
 3 SISTEMAS DE RECONHECIMENTO DE CARACTERES	67
3.1 Origens e Evolução	67
3.2 Tipos de Sistemas de Reconhecimento de Caracteres ..	69
3.3 Descrição dos Sistemas OCR	76
3.3.1 Etapas de Processamento	77
3.3.2 Técnicas de Reconhecimento	80
3.3.2.1 Atributos Globais	80
3.3.2.1.1 Casamento de Padrões e Correlações	80
3.3.2.1.2 Transformações e Expansões em Séries	80
3.3.2.2 Distribuição de Pontos	81
3.3.2.2.1 Zoneamento	81
3.3.2.2.2 Momentos	81
3.3.2.2.3 N-Uplas	81
3.3.2.2.4 <i>Characteristic Loci</i>	81
3.3.2.5 Cruzamentos e Distâncias	82
3.3.2.3 Atributos Geométricos e Topológicos	82
3.3.3 Avaliação do Reconhecimento	83
3.3.4 Principais Dificuldades	87
3.4 Análise de Algoritmos	94
3.4.1 Segmentação	94
3.4.1.1 Separação de Textos e Gráficos	94
3.4.1.2 Separação de caracteres	98
3.4.2 Extração de Atributos	104
3.4.2.1 Vizinhança Global	105
3.4.2.2 Vizinhança Local	105
3.4.2.3 Projeção	106

3.4.2.4 Alternância	106	
3.4.2.5 Simetria	107	
3.4.2.6 Densidade	107	
3.4.3 Alteração de Escala e Posição	108	
3.5 Perspectivas	111	
3.5.1 Adaptação	111	
3.5.2 Processamento Paralelo	112	
3.5.3 Uso de Contexto	113	
4 REDES NEURAIS		115
4.1 Conceitos Básicos	115	
4.2 Origem e Evolução	121	
4.3 Características e Aplicações das Redes Neurais	123	
4.4 Modelos de Redes Neurais	125	
4.4.1 Classificação dos Modelos	126	
4.4.2 Modelos Adaptativos	130	
4.4.2.1 Perceptron	131	
4.4.2.2 Adaline	133	
4.4.2.3 Madaline	137	
4.4.2.4 Redes Multinível	140	
4.4.2.5 ART	142	
4.4.3 Modelos Competitivos	145	
4.4.3.1 Modelo de Hopfield	145	
4.4.3.2 Máquina de Boltzmann	149	
4.4.3.3 BAM	149	
4.4.3.4 Modelo de Kohonen	152	
4.5 Aplicação das Redes Neurais Junto aos Sistemas OCR	155	
5 SISTEMA N ² OCR		157
5.1 Organização Geral do Sistema	161	
5.2 Aquisição de Imagens	164	
5.3 Tratamento da Imagem	172	
5.4 Preparação dos Dados para a Rede Neural	180	
5.4.1 Segmentação	184	
5.4.2 Rotulação	189	
5.4.3 Ajuste de Escala e Posição	195	

5.5 Rede Neural	195
5.5.1 Modelo Adotado	200
5.5.1.1 Escolha do Modelo	200
5.5.1.2 Descrição do Modelo	204
5.5.2 Aprendizado	204
5.5.3 Reconhecimento	212
5.5.4 Extensão do Modelo	213
5.5.5 Implementação em Hardware	217
5.6 Arquivo de Configuração	219
5.7 Avaliação do Desempenho do Sistema	223
5.8 Evolução do Sistema	229
 6 CONCLUSÃO	233
 ANEXO A.1 - Padrões de Descrição de Imagens	236
ANEXO A.2 - Padrões de Descrição de Caracteres	244
ANEXO A.3 - Padrões de Teste Empregados	247
ANEXO A.4 - Avaliação do Desempenho do Sistema N ² OCR ...	265
ANEXO A.5 - Listagens de Arquivos do Sistema N ² OCR	283
 BIBLIOGRAFIA CITADA	291
BIBLIOGRAFIA CONSULTADA	301

GLOSSÁRIO

- Amostragem - Discretização da imagem no espaço
- Artefatos - Pontos de "sujeira" em uma imagem. São elementos da imagem que não fazem parte do objeto alvo da análise
- Bit-Map* - Conjunto de pontos de uma imagem digital
- Desempenho - Avaliação do funcionamento de um sistema em relação a sua eficácia e eficiência
- Eficácia - Desempenho em termos de taxa de acertos
- Eficiência - Desempenho em termos de rapidez do sistema
- Performance* - Desempenho de um sistema
- Pixel ou Pel* - Elemento de imagem, um ponto de um padrão bit-map ("picture element")
- Quantização - Discretização da imagem na amplitude

LISTA DE ABREVIATURAS

A/D	- Conversor de sinais analógicos para digitais
ADALINE	- Adaptive Linear Element
ADANLIENE	- Adaptive Non-Lienar Expandible Network
ANSI	- American National Standards Institute
ART	- Adaptive Resonance Theory
ASCII	- American Standard Code for Information Interchange
B&W	- Branco e Preto (dispositivo de 2 cores apenas)
BRASCI	- Código ASCII brasileiro, inclui acentuação
BTC	- Block Truncation Encoding
CCD	- Charged Coupled Devices
CPGCC	- Curso de Pós-Graduação em Ciência da Computação
D/A	- Conversor de sinais digitais para analógicos
DPI	- Dots per Inch
FFT	- Fast Fourier Transform
HDTV	- High Definition Television
IEEE	- Institute of Electrical and Electronic Engineers
II	- Instituto de Informática
LUT	- Look Up Table
LZW	- Lempel-Ziv & Welch Algorithm
MADALINE	- Many ADALINE (Adaptive Linear Element)
N ² OCR	- Neural Network for Optical Character Recognition
P&B	- Preto e Branco, o mesmo que B&W
OCR	- Optical Character Recognition
OHCR	- Optical Handwritten Character Recognition
OPCR	- Optical Printed Character Recognition
RAC	- Relative Address Coding
RGB	- Red, Green, Blue
RLE	- Run Length Encoding
RLSA	- Run Length Smoothing Algorithm
SCR	- Single Character Recognition
SIRENE	- Projeto "Ambiente para SIMulação de REdes NEurais"
UFRGS	- Universidade Federal do Rio Grande do Sul
XOR	- Função Ou Exclusivo (Exclusive OR)

LISTA DE FIGURAS

Figura 2.1 - Relações entre as áreas constituintes do processamento Gráfico	25
Figura 2.2 - Esquema de um digitalizador com câmera de vídeo	29
Figura 2.3 - Classificação dos dispositivos de digitalização	30
Figura 2.4 - Exemplos de matrizes de caracteres	32
Figura 2.5 - Exemplo de compactação RLE	40
Figura 2.6 - Exemplos de aproximação de intensi- dades através do uso de <i>halftoning</i>	45
Figura 2.7 - Matrizes de <i>halftoning</i> usadas na geração de cinco tons	45
Figura 2.8 - Histograma da figura 2.6 (b) com o <i>halftoning</i> desfeito (matriz 6x6)	49
Figura 2.9 - Binarização da imagem do texto da figura 2.6 (a)	52
Figura 2.10 - Imagem original e imagem após convolução para a detecção de contornos	56
Figura 2.11 - Esquema geral de um processo de reconhecimento e classificação de imagens	61
Figura 3.1 - Classificação dos sistemas de reconhecimento de caracteres	70
Figura 3.2 - Exemplos de padronização de conjuntos de caracteres para sistemas de OCR	76
Figura 3.3 - Esquema de etapas de processamento em um sistema OCR	77
Figura 3.4 - Complexidade dos sistemas OCR	91
Figura 3.5 - Etapas do processamento do algoritmo RLSA	96
Figura 3.6 - Separação dos textos do documento da figura 3.5	98
Figura 3.7 - Algoritmo geral de separação de caracteres	101

Figura 3.8 - Localização do início de uma linha	101
Figura 3.9 - Algoritmo para isolar um caractere	103
Figura 3.10 - Extração de um caractere desconexo	104
Figura 3.11 - Exemplo de vizinhança global	105
Figura 3.12 - Exemplo de padrões de vizinhança local	105
Figura 3.13 - Exemplo de cálculo de projeções	106
Figura 3.14 - Exemplo de cálculo de simetria	107
Figura 3.15 - Exemplo de cálculo da densidade de sub-regiões	107
Figura 3.16 - Separação e posicionamento de carac- teres	109
Figura 3.17 - Redução de um padrão 3x3 para 2x2	110
Figura 3.18 - Algoritmo de alteração de escala	110
Figura 4.1 - Aparência externa de um neurônio	116
Figura 4.2 - Modelo do <i>Psychon</i> de McCulloch & Pitts	119
Figura 4.3 - Topologias de interconexões em redes neurais	121
Figura 4.4 - Estrutura de um neurônio artificial	126
Figura 4.5 - Taxonomia das redes artificiais de neurônios	129
Figura 4.6 - Funções de transferência	130
Figura 4.7 - Exemplo de uso do Perceptron	132
Figura 4.8 - Esquema do neurônio tipo Adaline	134
Figura 4.9 - Algoritmo de aprendizado <i>Delta Rule</i>	135
Figura 4.10 - Níveis de uma rede versus regiões de decisão	138
Figura 4.11 - Solução do problema do XOR	139
Figura 4.12 - Funcionamento do modelo do ART	144
Figura 4.13 - Organização de uma rede de Hopfield	146
Figura 4.14 - Utilização da rede de Hopfield	146
Figura 4.15 - Interação entre os neurônios do modelo de Hopfield	147
Figura 4.16 - Algoritmos da rede de Hopfield	148
Figura 4.17 - Processo de codificação de uma BAM	151

Figura 4.18 - Organização de uma rede neural de Kohonen	153
Figura 4.19 - Agrupamento da atividade das saídas de uma matriz bidimensional de neurônios	154
Figura 4.20 - Exemplos de topologias de vizinhança	155
Figura 5.1 - Esquema de funcionamento do sistema N ² OCR	161
Figura 5.2 - Protótipo do sistema N ² OCR	162
Figura 5.3 - Tela de ajuda do módulo gerador de imagens	165
Figura 5.4 - Configuração do módulo gerador de imagens	166
Figura 5.5 - Padrão do caractere 'a' gerado sem distorção	168
Figura 5.6 - Padrão do caractere 'a' gerado com a inversão aleatória de 5% dos pixels	169
Figura 5.7 - Padrão do caractere 'a' gerado com a inversão aleatória de 10% dos pixels	170
Figura 5.8 - Padrão do caractere 'a' gerado com a rotação de 10 graus	171
Figura 5.9 - Tela de ajuda do módulo de tratamento de imagens	173
Figura 5.10 - Conversão dos formatos de arquivos	174
Figura 5.11 - Visualização de uma imagem de um texto ..	175
Figura 5.12 - Visualização de uma imagem com halftoning ..	176
Figura 5.13 - Histograma da imagem da fig. 5.12 com halftoning desfeito	177
Figura 5.14 - Binarização da imagem cujo histograma aparece na fig. 5.13	178
Figura 5.15 - Convolução para detecção de bordas na imagem da fig. 5.14	179
Figura 5.16 - Tela de ajuda do módulo de preparação de dados	181
Figura 5.17 - Imagem exibida pelo módulo de preparação de dados	182
Figura 5.18 - Exibição de outra área da imagem da fig. 5.17 (deslocamento interativo)	183
Figura 5.19 - Parâmetros de configuração da segmentação	185

Figura 5.20 - Primeiro caractere isolado da imagem	186
Figura 5.21 - Segundo caractere isolado da imagem	187
Figura 5.22 - Terceiro caractere isolado da imagem	188
Figura 5.23 - Rotulação de caracteres	190
Figura 5.24 - Configuração do processo de rotulação ...	192
Figura 5.25 - Caractere isolado com código de identificação	193
Figura 5.26 - Caractere isolado com rótulo de desconhecido	194
Figura 5.27 - Ajuste de escala e posição	196
Figura 5.28 - Padrão com ajuste de escala 10 x 10	197
Figura 5.29 - Padrão com ajuste de escala 16 x 16	198
Figura 5.30 - Padrão com ajuste de escala 20 x 20	199
Figura 5.31 - Organização da rede neural do modelo ADAn-LIeNE	203
Figura 5.32 - Expansão da rede neural	206
Figura 5.33 - Tabela de controle de aprendizado	210
Figura 5.34 - Proposta simplificada de uma arquitetura	218
Figura 5.35 - Avaliação automática do reconhecimento ..	226
Figura 5.36 - Recodificação de um padrão	230

LISTA DE TABELAS

Tabela 3.1 - Taxas de reconhecimento obtidas por diferentes sistemas OHCR	86
Tabela 3.2 - Comportamento das técnicas de reconhecimento	90
Tabela 3.3 - Erros de substituição mais comuns	92
Tabela 5.1 - Funções previstas no sistema N ² OCR e funções implementadas no protótipo	158

RESUMO

Este trabalho apresenta um estudo sobre o reconhecimento visual de caracteres através da utilização das redes neurais. São abordados os assuntos referentes ao Processamento Digital de Imagens, aos sistemas de reconhecimento de caracteres, e às redes neurais. Ao final é apresentada uma proposta de implementação de um sistema OCR orientado ao reconhecimento de caracteres impressos, que utiliza uma rede neural desenvolvida especificamente para esta aplicação. O sistema proposto, que é denominado de sistema N²OCR, possui um protótipo implementado, que também é descrito neste trabalho.

Em relação ao Processamento Digital de Imagens são apresentados diversos temas, abrangendo os assuntos referentes à aquisição de imagens, ao tratamento das imagens e ao reconhecimento de padrões. A respeito da aquisição de imagens são destacados os aspectos referentes aos dispositivos de aquisição e os tipos de imagens obtidas através destes. Sobre o tratamento de imagens são abordados os aspectos referentes a imagens textuais, incluindo: halftoning, geração e modificação de histograma, limiarização e operações de filtragem. Quanto ao reconhecimento de padrões é feita uma breve análise das técnicas relacionadas a este tema.

Os diversos tipos de sistemas de reconhecimento de caracteres são abordados, assim como as técnicas e algoritmos empregados por estes. Além destes tópicos é apresentada uma discussão a respeito da avaliação dos resultados obtidos por estes sistemas, assim como é feita uma análise das principais dificuldades enfrentadas por estas aplicações.

Neste trabalho é feita uma apresentação a respeito das redes neurais, suas características, histórico e evolução das pesquisas nesta área. É feita uma descrição dos principais modelos de redes neurais em destaque na atualidade: Perceptron, Adaline, Madaline, redes multinível, ART, modelo de Hopfield, máquina de Boltzmann, BAM e modelo de Kohonen.

A partir da análise dos diferentes modelos de redes neurais empregados na atualidade, chega-se a proposta de um novo modelo de rede a ser utilizado pelo sistema N²OCR. São descritos os itens referentes ao aprendizado, ao reconhecimento e as possíveis extensões deste novo modelo. Também é abordada a possibilidade de implementação de um hardware dedicado para este modelo.

No final deste trabalho é fornecida uma visão global do sistema N²OCR, descrevendo cada um de seus módulos. Também é feita uma descrição do protótipo implementado e de suas funções.

Palavras-chave: Processamento de Imagens, Sistemas OCR, Reconhecimento de Caracteres, Redes Neurais.

ABSTRACT

This work presents a study of visual character recognition using neural networks. It describes some aspects related to Digital Image Processing, character recognition systems and neural networks. The implementation proposal of one OCR system, for printed character recognition, is also presented. This system uses one neural network specifically developed for this purpose. The OCR system, named N²OCR, has a prototype implementation, which is also described.

Several topics related to Digital Image Processing are presented, including some referent to image acquisition, image processing and pattern recognition. Some aspects on image acquisition are treated, like acquisition equipments and kinds of image data obtained from those equipments. The following items about text image processing are mentioned: halftoning, histogram generation and alteration, thresholding and filtering operations. A brief analysis about pattern recognition related to this theme is done.

Different kinds of character recognition systems are described, as the techniques and algorithms used by them. Besides, a discussion about performance estimation of this OCR systems is done, including typical OCR problems description and analysis.

In this work, neural networks are presented, describing their characteristics, historical aspects and research evolution in this field. Different famous neural network models are described: Perceptron, Adaline, Madaline, multi-level networks, ART, Hopfield's model, Boltzmann machine, BAM and Kohonen's model.

From the analysis of such different neural network models, we arrive to a proposal of a new neural network model, where are described items related to learning, recognition and possible model extensions. A possible hardware implementation of this model is also presented.

A global vision of N²OCR system is presented at the end of this work, describing each of its modules. A description of the prototype implementation and functions is also provided.

Key-words: Image Processing, OCR Systems, Character Recognition, Neural Networks.

1 INTRODUÇÃO

Este trabalho apresenta um estudo de sistemas de reconhecimento de caracteres onde são empregadas técnicas especiais, como é o caso da utilização das redes neurais. Mas o que vem a ser um sistema de reconhecimento de caracteres, e por que é importante reconhecer caracteres? O que são e para que servem as redes neurais? Um dos objetivos deste trabalho é o esclarecimento destas perguntas através de uma dissertação sobre estes temas, possibilitando ao leitor um entendimento geral sobre tais assuntos. Além de abordar estes temas, como objetivo principal, será apresentada uma proposta de implementação de um sistema completo de reconhecimento de caracteres, sendo descrita a implementação de um protótipo deste sistema.

Um sistema de reconhecimento de caracteres permite que o computador possa adquirir informações através da "leitura" de textos, ou seja, ao invés de entrar as informações pelo teclado, é possível implementar um sistema computadorizado, associado a um dispositivo ótico, para a aquisição automática de informações descritas sob a forma textual.

Os sistemas de reconhecimento de caracteres possuem uma grande importância junto ao processamento de dados. Isto é devido ao fato da linguagem escrita ser a forma mais usual do ser humano armazenar e transmitir informações. Basicamente as duas formas de comunicação mais utilizadas pelo ser humano são a fala e a escrita, sendo que ambas possuem uma inestimável importância na aquisição, difusão e aprimoramento dos conhecimentos humanos. A partir destas constatações, chegamos à conclusão que as duas formas mais naturais e adequadas de interface entre homem-máquina, são através da escrita e da fala. Sistemas que permitam a troca de informações com o ser humano através de uma destas formas, são sistemas que possibilitam uma alta interação homem-máquina, e sua principal vantagem é o fato de ser desnecessário o ser humano alterar o seu comportamento frente a uma máquina, para poder ter acesso a esta.

A linguagem falada, na época atual, ainda é uma forma de interação que possui muitas dificuldades de implementação prática. Por isso, atualmente os sistemas de aquisição de informações textuais, que são mais viáveis, tem crescido muito em sua utilização prática. Encontra-se atualmente o emprego de sistemas de reconhecimento de caracteres desde os níveis empresariais, onde o volume de informações é muito grande, até em aplicações domésticas. Estes sistemas estão se tornando cada vez mais difundidos conforme demonstram [DIE 91, JOC 91, FAL 88 e KAN 89]. E foi baseando-se em fatos como estes que surgiu a motivação do desenvolvimento deste trabalho.

Além de servirem como uma forma alternativa de interação homem-máquina, os sistemas de reconhecimento de caracteres tem sido muito utilizados também devido a sua alta capacidade de compactação de imagens textuais. Este caso se aplica muito na utilização de sistemas de fac-símile para transmissão de textos. Através do reconhecimento dos caracteres, em uma imagem a ser transmitida por este tipo de equipamento, pode-se obter uma ótima compactação das informações, resultando em uma grande economia (menor número de informações transmitidas). Estes conceitos, superficialmente decretos acima, serão esclarecidos de uma forma mais detalhada posteriormente neste trabalho.

As redes neurais são um novo paradigma de desenvolvimento de sistemas, que tem se difundido muito na atualidade. As redes neurais tem se apresentado como uma solução muito adequada para sistemas de reconhecimento de padrões, como é o caso de um sistema de reconhecimento de caracteres, onde os padrões a serem reconhecidos são os próprios caracteres. Esta característica, a ser demonstrada em maiores detalhes posteriormente, foi o que levou ao estudo e emprego das redes neurais junto a este trabalho.

Além de serem apresentados temas como os sistemas de reconhecimento de caracteres e as redes neurais, também será abordado o projeto completo de um destes sistemas.

O sistema descrito, cujo protótipo foi desenvolvido junto a este trabalho, é denominado de Sistema N²OCR. O nome N²OCR é a sigla que representa o sistema para reconhecimento ótico de caracteres através de redes neurais - *Neural Network for Optical Character Recognition*. Este sistema é constituído por vários módulos que implementam as tarefas desde a entrada das imagens de um texto até o reconhecimento dos caracteres (textos impressos com múltiplas fontes). O reconhecimento dos caracteres é implementado através do emprego de uma rede neural. A rede neural que implementa o reconhecimento de caracteres foi desenvolvida a partir dos modelos estudados (descritos no capítulo 4), constituindo uma proposta original, criada pelo autor, para a solução do problema de reconhecimento de caracteres.

Este trabalho se divide em um total de sete capítulos. Os capítulos 1 e 7 são referentes à introdução e à conclusão do assunto. Os demais capítulos abordam os temas referentes aos assuntos descritos acima.

O capítulo 2 apresenta uma introdução ao processamento de imagens, onde são destacados os principais aspectos referentes à aquisição de imagens de textos, tratamento destas imagens e técnicas tradicionais de reconhecimento de padrões empregadas junto ao processamento de imagens.

O capítulo 3 aborda os sistemas de reconhecimento de caracteres, onde é apresentado um estudo a respeito dos diferentes tipos de sistemas utilizados para este fim, bem como métodos de avaliação de desempenho, etapas de processamento, técnicas e algoritmos empregados nos sistemas tradicionais, e, ao final, é feita uma avaliação geral dos problemas e tendências de tais sistemas.

O capítulo 4 é dedicado a um estudo sobre as redes neurais. Neste capítulo são apresentados os principais conceitos a respeito desta área de estudos, modelos de redes neurais de destaque na atualidade, bem como a caracterização

de cada um destes modelos em termos de propriedades e capacidade para o emprego junto ao reconhecimento de padrões.

O capítulo 5 apresenta o projeto completo de um sistema de reconhecimento de caracteres. É apresentado o sistema N²OCR, que engloba as etapas de captura da imagem, tratamento desta, preparação dos dados para a rede neural e utilização da rede neural. Neste capítulo é proposto um modelo específico de rede neural, desenvolvido para aplicações de reconhecimento de caracteres. Também é descrita a implementação do protótipo do sistema N²OCR.

O capítulo 6 apresenta as perspectivas de evolução do sistema N²OCR. São apresentadas as possíveis extensões do sistema através da ampliação de suas potencialidades, além dos resultados em termos da criação de novos projetos de pesquisas em desenvolvimento junto ao CPGCC-UFRGS.

Os capítulos de 2 a 4 apresentam uma introdução aos assuntos de processamento de imagens, reconhecimento de caracteres e redes neurais respectivamente. O leitor que já possuir conhecimentos bem fundamentados sobre algum destes itens pode avançar diretamente para os capítulos seguintes. A partir do capítulo 5 são descritas as propostas originais do autor, sendo os capítulos anteriores uma revisão de aspectos básicos relevantes para a compreensão completa do trabalho desenvolvido.

O protótipo do sistema N²OCR foi implementado em equipamentos do tipo IBM-PC compatíveis, sendo desenvolvido na linguagem "C" sob o sistema operacional MS-DOS. Este sistema pode ser portado para outras máquinas, sendo que o simulador da rede neural já foi adaptado para ser utilizado em estações de trabalho SUN SparcStation.

2 PROCESSAMENTO DE IMAGENS

Neste capítulo será abordado o "Processamento de Imagens", onde serão destacados os aspectos: caracterização do Processamento de Imagens, as técnicas e procedimentos empregados na manipulação das imagens, e o reconhecimento de padrões.

2.1 Introdução

O Processamento de Imagens é uma área de estudos da computação que tem crescido muito nos últimos anos. Seu grande crescimento se deve principalmente ao desenvolvimento de equipamentos, cada vez mais sofisticados e baratos, utilizados para a captura e tratamento de imagens digitais.

O Processamento de Imagens, mais exatamente, o *Processamento Digital de Imagens* é uma área de estudos que está inserida dentro do *Processamento Gráfico*, junto a *Computação Gráfica*. O esquema da figura 2.1 apresenta as ligações e relações entre estas áreas, caracterizando cada uma delas em relação ao tipo de informações manipuladas. O Processamento Gráfico engloba as áreas de Processamento de Imagens e Computação Gráfica, onde o Processamento de Imagens está relacionado ao tratamento e reconhecimento de padrões em imagens, e a Computação Gráfica está ligada à síntese de imagens e à geração de descrições (dados não pictóricos) utilizadas na obtenção destas imagens.

A área de Processamento Digital de Imagens teve início em meados da década de 60, com a motivação criada pelos programas espaciais da NASA nos EUA [MAS 89]. O aumento da capacidade de memória e velocidade dos computadores contribuiu para dar impulso à nova tecnologia, que passou a ser aplicada em outras áreas como: medicina, microscopia, ótica eletrônica, transmissão digital de sinais de TV ou fac-símile e em sensores visuais de robôs.

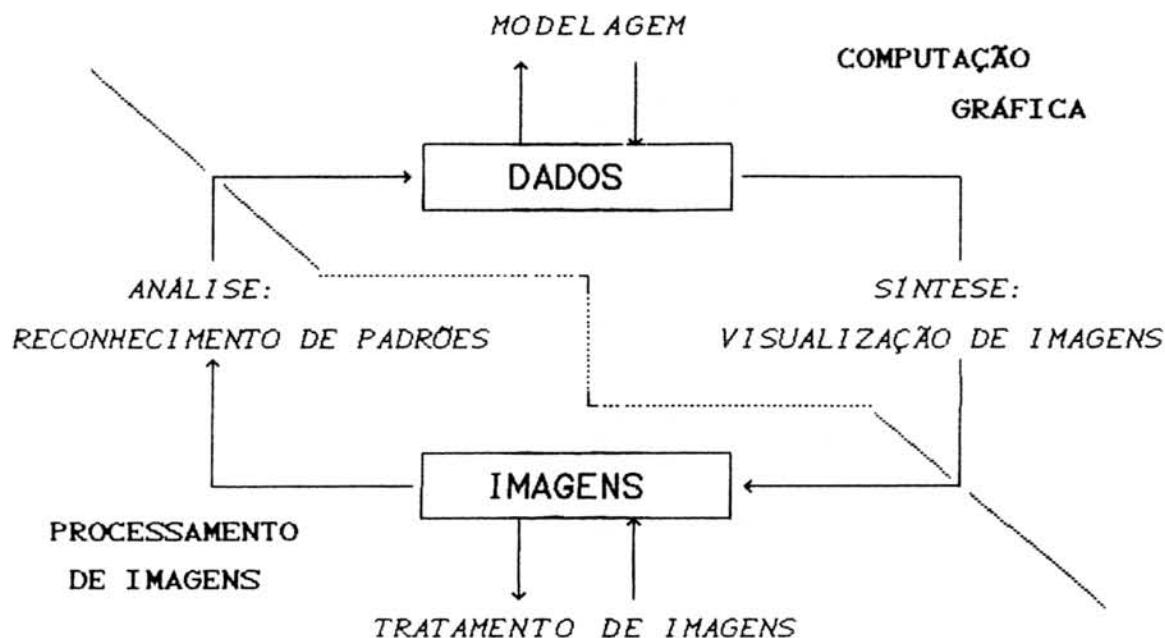


Figura 2.1 - Relações entre as áreas constituintes do processamento gráfico

Na área espacial, deve ser mencionada a grande utilização de imagens digitais a partir do lançamento da série de satélites LANDSAT em 1972, bem como dos satélites meteorológicos. Em 1975, a invenção do tomógrafo computadorizado por G. N. Hounsfield, abriu amplas perspectivas de utilização das técnicas digitais em imagens radiográficas. O número de aplicações tem se multiplicado na atualidade e hoje em dia o Processamento Digital de Imagens pode ser encontrado junto às mais diversas áreas de atuação. Novas aplicações do processamento digital de imagens tem surgido, destacando-se a HDTV e a transmissão e reprodução de imagens (textos) através do fac-símile.

O processamento digital de imagens é essencialmente uma área multidisciplinar, abrangendo conhecimentos da Ótica, Colorimetria, Neurofisiologia, Engenharia Elétrica e Ciência da Computação. Estes conhecimentos são utilizados para o desenvolvimento tanto de equipamentos, como de algoritmos empregados junto ao Processamento de Imagens.

Serão abordados em destaque neste capítulo os aspectos relacionados com as técnicas e algoritmos usados no processamento de imagens, sendo que será apresentada apenas uma visão global a respeito do funcionamento e implementação dos equipamentos utilizados para este fim.

2.2 Imagem Digital

O objeto de trabalho no Processamento de Imagens é a imagem digital, onde esta é definida como sendo uma matriz de $M \times N$ elementos (vetores com informações referentes aos pontos da imagem). Cada elemento da imagem digital é denominado de *pixel* (*picture element*), tendo associado a si uma informação referente à luminosidade e à cor. Esta informação pode ser um valor indicativo de intensidade luminosa ou um índice para acesso indireto à cor real através de uma LUT (*look up table*), conforme será apresentado posteriormente.

Esta representação através de uma matriz bidimensional de pontos é resultante da manipulação da imagem como sendo uma área de memória do computador. A cada pixel é associada uma ou mais posições de memória, as quais devem armazenar as diversas informações referentes a estes. Com isto pode-se armazenar e processar as informações referentes a uma imagem para posteriormente exibi-la em um dispositivo de saída matricial (vídeo "raster scan", impressoras matriciais, laser, etc).

Uma imagem digital pode ser vista como uma função $i: R \rightarrow V$, onde R é um subconjunto do plano, e V é um espaço vetorial. Para cada pixel $(u, v) \in R$ da imagem, $i(u, v)$ contém as informações da estrutura do pixel, e é chamado valor do pixel. Na maioria das aplicações, R é um reticulado uniforme do plano, V é um espaço de cor, e i associa a cada ponto do plano (pixel) a sua intensidade de cor. A dimensão $M \times N$ é chamada de "resolução geométrica" da imagem, e o número de bits usado para armazenar o valor do pixel é denominado de "resolução de cor" da imagem.

2.2.1 Digitalização

O processo de digitalização consiste em realizar a aquisição de uma cena, a qual é passada para o computador em um formato adequado para que este possa manipulá-la. As informações visuais são convertidas em sinais elétricos por sensores óticos, e estes sinais são quantificados em valores binários e armazenados na memória do computador. No processo de digitalização, os sinais são amostrados espacialmente e quantificados em amplitude, de forma a obter a imagem digital.

2.2.1.1 Digitalizadores

Uma imagem digital é definida como uma matriz bidimensional, onde cada ponto possui um valor relativo a sua intensidade luminosa. Existem vários dispositivos associados às várias aplicações que manipulam imagens. Como exemplos, pode-se citar: o radar, usado na obtenção de imagens de objetos distantes; aparelhos médicos baseados em ultra-som, usados para analisar o interior do corpo humano e a câmera de TV usada na obtenção de imagens locais. Basicamente o processo de formação das imagens nestes diversos dispositivos é o mesmo.

O interesse deste trabalho está voltado à captura de imagens de textos, a qual será realizada principalmente através de câmeras de TV e scanners.

Os digitalizadores de imagens são constituídos basicamente pelos seguintes elementos:

1) ILUMINADOR: dispositivo responsável pela iluminação do objeto alvo da digitalização. A iluminação pode ser controlada/focalizada (feixe de luz) ou através de uma fonte luminosa simples ou até mesmo utilizando-se da luz ambiente natural (luz solar).

2) SENSOR: elemento fotossensível (fotodiodo ou fotomultiplicador) capaz de traduzir uma intensidade luminosa incidente, em valores de corrente elétrica proporcionais a esta luminosidade. Estes sensores possuem uma curva de resposta espectral, gerando intensidades de corrente elétrica diferentes para diferentes intensidades luminosas de um dado comprimento de onda.

3) CONVERSOR A/D: o valor contínuo da corrente elétrica gerada pelo sensor é convertido em um valor discreto (análogo → digital). Este valor discreto pode então ser armazenado sob a forma numérica na memória de um computador.

Os digitalizadores diferem entre si principalmente quanto ao número de sensores: pontual, vetor ou matriz (fig. 2.3). Isto implica diferentes formas de realizar a varredura sobre a cena, influindo na formação da imagem digital.

O primeiro tipo de digitalizador que será analisado é o baseado em câmeras de vídeo, também denominado de *frame grabber* (figura 2.2). Existem dois tipos principais destes digitalizadores, os que utilizam câmeras VIDICOM e os de câmeras CCD. As câmeras VIDICOM possuem um tubo de imagem capaz de gerar, por exemplo, 30 quadros por segundo, onde cada quadro possui uma resolução vertical discreta útil de 480 linhas. Nestas câmeras, a imagem é formada sobre um plano fotossensível, o qual sofre uma varredura, gerando o padrão elétrico que depois será convertido para os valores digitais que representam a cor de cada pixel. As câmeras tipo VIDICOM são muito frágeis e estão sujeitas a problemas de calibração frequente, além de possuírem um alto consumo. O outro tipo de câmeras, as CCD (*Charged Coupled Devices*) ou como também são conhecidas, os "area imaging devices" possuem uma matriz CCD de elementos fotossensíveis discretos. Estas câmeras também são conhecidas por "câmeras de estado sólido", sendo mais leves, resistentes, de baixo consumo e não estão sujeitas a problemas de calibração como no caso da VIDICOM.

Os dispositivos baseados em câmeras de vídeo são capazes de capturar cenas de natureza tridimensional, já o segundo tipo de digitalizadores, os scanners, não possuem a capacidade para adquirir este tipo de imagens. O objeto alvo da digitalização por um scanner tem que ser plano, podendo ser uma fotografia, impresso ou qualquer outro tipo de objeto plano. Os scanners são muito utilizados na atualidade, podendo ser de três tipos: de mesa, manual ou "flying spot scanners".

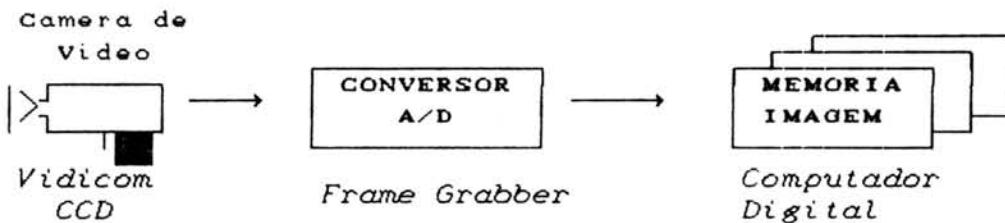


Figura 2.2 - Esquema de um digitalizador com câmera de vídeo

Os scanners de mesa possuem normalmente um arranjo em vetor dos elementos fotossensíveis, os quais são deslocados sobre o plano de digitalização através de um sistema mecânico. Este tipo de scanners realiza a varredura de maneira similar a uma máquina fotocopiadora. Os scanners manuais (*handy scanners*) são muito semelhantes aos de mesa, mas a varredura sobre a imagem é feita de forma manual, onde o usuário substitui o dispositivo mecânico responsável pelo deslocamento dos sensores sobre a imagem. Este tipo de scanner está sujeito a um número maior de perturbações, possíveis de serem causadas por um manejo imperfeito dos sensores na hora da varredura. Em compensação, este último tipo de scanner possui um preço bem mais acessível. O último tipo de scanner a ser analisado, o *flying spot scanner*, é um dispositivo onde há um sensor pontual, o qual é deslocado sobre a imagem, gerando assim uma descrição desta. A descrição gerada por este tipo de scanner pode ser diferente daquela obtida em outros digitalizadores, sendo muito comum neste caso, sensores que percorrem o contorno do objeto, devolvendo uma descrição vetorial do mesmo.

Por fim, um último tipo de digitalizador a ser analisado é o que abrange o denominado "tablet" e a mesa digitalizadora. Este tipo de dispositivo deve ser considerado em separado dos demais, pois gera uma descrição da imagem completamente diferente da obtida pelos outros dispositivos. O tablet permite a transcrição da imagem em termos de coordenadas do plano cartesiano (x, y), coordenadas estas referentes às posições sobre as quais é deslocado um cursor, movimentado sobre uma mesa especial sensora. A descrição final da imagem, neste caso, não será portanto uma descrição matricial, mas sim uma descrição que representa a trajetória realizada com o cursor sobre a mesa.

Sensores	Matriz	<ul style="list-style-type: none"> VIDICOM Matriz CCD
	Vetor	<ul style="list-style-type: none"> Scanner c/ varredura fixa automática (Scanner de Mesa) Scanner c/ varredura fixa manual (Handy Scanner)
	Pontual →	Scanner c/ varredura aleatória (Flying Spot Scanner)

Figura 2.3 - Classificação dos dispositivos de digitalização

2.2.1.2 Amostragem

Como já foi mencionado, uma imagem durante o processo de digitalização sofre uma discretização no espaço denominada de amostragem. Para realizar a amostragem de uma imagem deve ser fixado um intervalo de tempo que vai determinar a frequência de amostragem e consequentemente o número de pontos discretizados (resolução da imagem). É assumido na prática que o intervalo de amostragem entre duas células da imagem seja ao menos igual à metade do menor intervalo significativo da cena capturada, ou seja, a resolução deve ser o dobro da resolução necessária para visualizar o menor elemento significativo presente na cena [PAZ 88].

Para se obter uma resolução igual a da televisão, é preciso uma matriz com cerca de 512×512 elementos. No caso da digitalização utilizando-se uma câmera, a distância do objeto alvo até a câmera será de vital importância na sua captura, pois esta distância influí diretamente sobre o tamanho dos objetos capturados (quanto mais afastada a câmera, menores serão os objetos).

No caso dos textos digitalizados, a resolução ideal de digitalização, para que não seja prejudicada a forma dos caracteres, deve ser tal que um único caractere resulte em uma matriz de 20 pontos na vertical por 15 pontos na horizontal [PAV 82]. Esta resolução média de 20×15 (300 pixels/caractere) é necessária para uma boa representação dos caracteres, incluindo os espaçamentos horizontal e vertical. Caso estas dimensões mínimas da matriz não sejam respeitadas, poderão ser perdidas informações importantes a respeito da geometria dos caracteres. Esta especificação também é muito relativa, pois ainda devem ser considerados: o tipo de dispositivo utilizado na impressão do texto; o tipo de letras utilizadas (típico, negrito, etc, podendo ser necessárias matrizes de pontos bem maiores) e a relação sinal/ruído presente na digitalização.

A matriz de representação mínima para caracteres, utilizada em muitos dispositivos de saída matricial, é de cerca de 8×8 pixels (impressoras matriciais) ou algo em torno de 10×7 pontos [PAV 82]. A menor matriz possível para representação de caracteres é de 4×6 pontos, como é demonstrado na figura 2.4, mas este tipo de matriz permite apenas a representação de caracteres maiúsculos. Portanto, para representar com a menor matriz de pixels possível, todo o conjunto de caracteres alfanuméricos (letras minúsculas, maiúsculas, dígitos e caracteres especiais) é necessário uma matriz 8×8 ; e como para uma digitalização sem perdas de detalhes deve-se usar o dobro da resolução de amostragem do menor elemento, chega-se a uma matriz 16×16 , com dimensões bem aproximadas daquela proposta por Pavlidis [PAV 82].

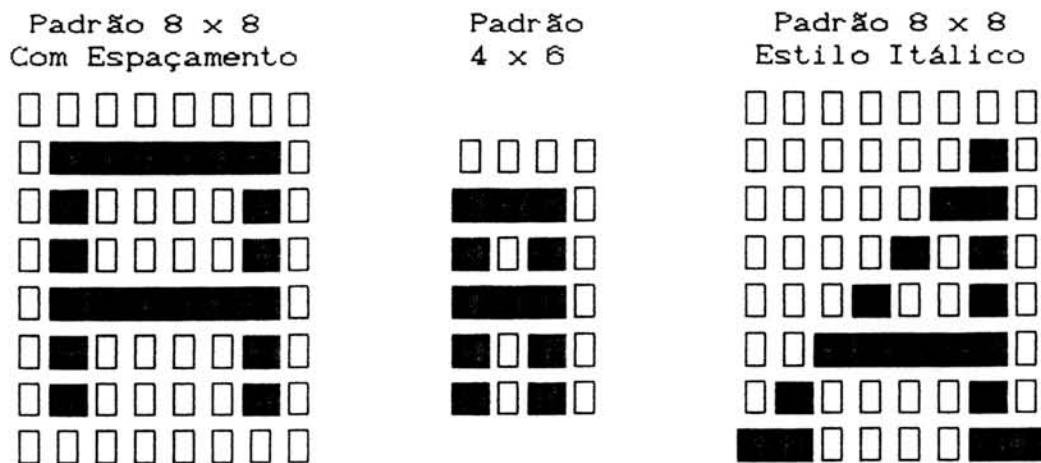


Figura 2.4 - Exemplos de matrizes de caracteres

Para se ter uma noção ainda melhor a respeito da resolução ideal de amostragem a ser utilizada com textos, é necessário se conhecer a resolução de impressão de diferentes dispositivos. Nos dispositivos matriciais, a resolução é normalmente medida em DPI (Dot Per Inch - Pontos por Polegada), onde uma impressora matricial possui resoluções de impressão que variam de 10 a 20 caracteres por polegada, com resolução gráfica de 60 a 240 DPI, obtendo assim matrizes de 8 x 8 à 24 x 24 pixels/caractere. Uma impressora laser atinge uma resolução de 300 DPI, com impressão de cerca de 12 caracteres por polegada e utilizando matrizes de cerca de 25 x 25 pixels/caractere.

Além da qualidade e resolução de impressão do texto, é importante se destacar que cada dispositivo de digitalização possuirá uma resolução de amostragem própria, a ser considerada. Os "handy scanners" possuem resoluções programáveis de 100 a 400 DPI e no caso da digitalização de textos, impressos com o uso de uma impressora matricial, através destes dispositivos, a resolução de amostragem deve ficar em torno de 200 DPI e para textos impressos em impressoras laser, uma resolução de 400 DPI é aconselhável para que não seja perdido nenhum detalhe.

Caso seja utilizada uma câmera de vídeo na digitalização, estas fornecem usualmente resoluções em torno de 512 x 512 pontos, e neste caso para que se obtenha uma definição dos caracteres adequada ao processamento de reconhecimento, aconselha-se que a distância entre texto e câmera seja ajustada de forma a obter uma resolução de caracteres conforme o descrito anteriormente.

O processo de digitalização de um texto deverá levar em consideração todos os dados acima, em relação à amostragem. Obviamente, o aumento da resolução de amostragem implicará um aumento considerável da massa de dados a ser manipulada. Concluindo, para fins de reconhecimento de textos uma resolução considerada aceitável, em relação ao nível de detalhes e quantidade de informações, é de 16 x 16 pontos por caractere.

2.2.1.3 Quantização

No processo de digitalização, a imagem sofrerá uma amostragem, conforme foi descrito no item anterior, e uma discretização da intensidade luminosa, que é denominada de quantização. No processo de quantização, uma imagem com tons contínuos é convertida em uma de tons discretos. Para o armazenamento e processamento por um computador, cada tonalidade (intensidade da luz refletida por cada ponto da imagem) é representada por um valor armazenado de forma binária. Cada ponto amostrado possuirá portanto um valor binário correspondente à intensidade luminosa da imagem naquele ponto.

As imagens podem ser do tipo monocromáticas ou policromáticas. Nas imagens monocromáticas somente uma faixa de comprimentos de onda (uma cor) é analisada pelo sensor, determinando as intensidades de luminosidade para esta faixa. As intensidades de luminosidades descritas acima são denominadas de tonalidades ou níveis de intensidade de cor.

Para as imagens policromáticas a digitalização é feita para diferentes faixas de comprimentos de onda (diferentes cores).

Através da análise do olho humano pode-se compreender a noção de cor. De acordo com a "tri-stimulus theory", o olho humano é constituído de cones (sensores) capazes de receber estímulos em três faixas de comprimentos de onda referentes às cores: azul, vermelho e verde [FOL 82]. A sensibilidade a estas três cores RGB (red-green-blue) determinou o desenvolvimento de aparelhos capazes de, no processo de digitalização, quantizar as intensidades para estas três faixas de comprimentos de onda. A digitalização em um sistema policromático consiste em determinar as intensidades para as faixas de vermelho, verde e azul, formando um vetor de três componentes que descreve a cor de um único ponto.

O olho humano é capaz de distinguir diferentes cores, e para uma determinada cor é capaz de distinguir diferentes níveis de intensidade (tons). Estudos psicológicos e fisiológicos demonstram que o olho humano não é capaz de distinguir níveis de iluminação que diferem menos que 1% um do outro [PAV 82]. Sua capacidade de distinção de cores resulta em um reconhecimento total de 350.000 cores e/ou tonalidades, sendo que para cada cor o número aproximado de tonalidades distinguíveis fica em torno de 128 níveis [FOL 82].

Portanto, em sistemas monocromáticos, os valores de quantificação da intensidade luminosa de uma certa cor, necessitam de 6 a 8 bits/pixel (64 a 256 níveis). O uso de um byte por pixel já permite a obtenção de um número de tons de uma dada cor, superior à capacidade do olho humano em distinguí-los. Logo, em um sistema policromático será necessário o uso de um byte por cada cor RGB relativa a um dado pixel, obtendo-se assim um número total de 16 milhões de cores e tonalidades representáveis, também superior a capacidade do olho humano em distinguí-las.

A opção pelo número de níveis de uma determinada coloração está diretamente relacionada ao tipo de aplicação na qual será utilizada a imagem capturada. A decisão a respeito do número de níveis também influenciará na quantidade de informações a serem manipuladas e na necessidade do aumento de sensibilidade dos sensores do dispositivo de digitalização. Nos sistemas monocromáticos, que são os mais utilizados na atualidade, a quantização irá fornecer valores usualmente entre 0 e 255, representando as tonalidades de cinza, que variam do preto - 0, até o branco - 255. Este tipo de representação é uma das mais utilizadas, constituindo a denominada "escala de cinza" (*gray scale*).

Para sistemas de processamento de imagens orientados ao tratamento de textos, somente dois níveis são necessários, fazendo uso apenas do preto e do branco. Nestes sistemas a informação presente sempre pode ser distinguida apenas através da identificação de uma cor de fundo e a cor dos caracteres. É possível se capturar imagens de 256 níveis (escala de cinza) ou até uma imagem colorida, para posteriormente utilizá-la no tratamento de textos, só que neste caso a imagem deverá sofrer um tratamento prévio visando a obtenção de uma descrição em apenas dois níveis; este tipo de abordagem será tratada no item 2.3.2.

2.2.1.4 Classes de Imagens

Como já foi discutido, um sensor possuirá uma curva de resposta espectral, fornecendo a intensidade de uma dada frequência, para uma posição específica da malha de amostragem. Devido ao fato de se poder trabalhar com diferentes configurações destas quantificações realizadas a nível espectral, é possível trabalhar-se com as seguintes classes de imagens baseado em [PAV 82]:

1) *Imagen P&B com dois níveis de intensidade:* São imagens cujos pixels possuem apenas dois níveis, indicando apenas se este ponto está aceso (branco), ou apagado (preto). Este é o tipo ideal de imagem utilizada no processamento de textos digitalizados.

2) *Imagen P&B com N níveis de intensidade:* São imagens onde é possível distinguir diferentes níveis de intensidades de cinza, obtidos através da quantificação das intensidades de luz intermediárias entre o preto e o branco. O número N de tons é normalmente uma potência de dois, e depende do número de bits usados para armazenar o valor da intensidade do pixel. Usualmente trabalha-se com 256 tonalidades, o que equivale a 8 bits/pixel. Este número é mais que suficiente para representar as diferentes tonalidades da escala de cinza, uma vez que, como já foi explicado, o olho humano não é capaz de distinguir tantas tonalidades.

3) *Imagen de uma banda do espectro com N níveis de intensidade:* Assim como se obtém uma escala de cinza, é possível obter uma escala de tonalidades de outras bandas espectrais (faixas de comprimentos de onda abrangidas). Exemplos de diferentes bandas espectrais são: escala de vermelho, verde, azul, infra-vermelho, etc. É interessante ressaltar que uma escala de tonalidades não precisa ficar restrita apenas às freqüências visíveis, podendo atingir outras faixas de comprimentos de onda, como por exemplo o infra-vermelho. Esta banda é muito usada em satélites, que possuem sensores especiais para esta faixa, podendo detectar por exemplo, a emissão de calor e consequentemente identificar queimadas.

4) *Imagenes multi-espectrais (coloridas):* A quantificação das intensidades de mais de uma banda espectral por vez fornecerá uma informação mais completa a respeito da cena adquirida. Este é o caso da obtenção de imagens coloridas, pois estas são obtidas através de sensores específicos para a determinação de cada uma das intensidades de vermelho, verde e azul - RGB.

A mistura destas três bandas ou cores, permite que se tenha a sensação de enxergar imagens coloridas, uma vez que o olho humano possui sensores que atuam justamente nestas três faixas de comprimento de onda. As imagens coloridas são usualmente compostas por um conjunto de 24 bits: 8 bits para representar as intensidades de vermelho, 8 bits para o verde, e 8 bits para o azul. Com a composição destas três cores básicas, utilizando 24 bits/pixel, pode-se chegar a um número de até 16 milhões de cores e tonalidades distintas. Este número é perfeitamente adequado para a representação da realidade, em cenas digitalizadas, sem a menor perda de detalhes e qualidade em relação às cores. Isto é devido ao fato deste número de cores ser em muito superior à capacidade do olho humano em distinguir cores e tonalidades.

5) *Imagens coloridas com uso de Look Up Table:* A obtenção de imagens coloridas não está restrita apenas às imagens com representação multi-espectral para cada ponto. Pode-se gerar uma imagem onde cada pixel é associado a um valor referente a uma tabela de palette (LUT - look up table), que descreve a cor real deste ponto da imagem. Desta forma não é preciso associar a cada ponto da imagem as informações a respeito das intensidades RGB, podendo então armazenar imagens de cenas coloridas sem a necessidade de reservar uma memória de 24 bits/pixel. Para este tipo de imagens, uma cena é composta pelo mapa de pixels, ou seja, a matriz $M \times N$ de pontos com os índices de acesso às cores da LUT, adicionada a uma tabela de acesso indireto às cores reais (LUT), a chamada tabela de palette.

Além de existirem imagens com diferentes formas de quantificação, ainda existem imagens nas quais a amostragem resulta em formas de descrição das imagens diferentes da matricial. Até o momento foram abordadas somente as imagens com descrição matricial, as quais serão o alvo principal de estudos deste trabalho; porém é importante que sejam citados os outros tipos de imagens. Este tipo de imagens compõem a próxima classe a ser analisada.

6) *Imagens com descrição vetorial:* São imagens cuja descrição não será feita através de uma representação matricial como as demais, mas sim através de algum código especial como por exemplo, "chain codes" [PAV 82]. Estas imagens podem ser resultado da captura de cenas através de dispositivos como o "flying spot scanner", onde o resultado não é uma imagem matricial, mas o contorno que foi "seguido" pelo sensor. Outro tipo de imagem vetorial é a obtida na digitalização com o uso de mesas digitalizadoras, onde a imagem é composta pelo conjunto de coordenadas dos pontos obtidos pela movimentação do cursor sobre a mesa.

Quando se faz o uso de um equipamento para a digitalização de imagens, como uma câmera ou scanner, o resultado desta digitalização será uma imagem do tipo 1, 2, ou 4. No caso do processamento de imagens visando o reconhecimento de textos, as imagens dos tipos 2 e 4 tem que sofrer um processo denominado binarização, para converter este tipo de imagens em uma imagem de apenas dois níveis.

2.2.2 Armazenamento

O resultado do processo de digitalização é a obtenção de uma imagem digital armazenada na memória do computador. Esta imagem pode então ser visualizada ou armazenada em um dispositivo de armazenamento secundário, para posterior processamento. Neste item será analisado o armazenamento de imagens digitais em arquivos.

Uma imagem digital possui a informação referente à luminosidade de cada ponto, e esta pode estar representada de diferentes formas como já foi apresentado na seção 2.2.1.4. De acordo com o apresentado, tem-se imagens de 1 a 24 bits por pixel, com diferentes resoluções de amostragem espacial e representações das faixas espectrais. Uma vez que as imagens possuem uma série de características próprias a cada uma, como estas serão descritas em um arquivo?

Existem basicamente dois problemas a serem tratados: a especificação do formato de armazenamento destas imagens e a forma de codificação das mesmas. Estes itens tem que ser analisados cuidadosamente, pois tomando-se uma imagem que represente a realidade, esta pode chegar a ocupar 3 Mbytes de memória (1024 x 1024 pixels, com 24 bits/pixel) e neste caso o emprego de algum tipo de técnica de codificação visando a compactação dos dados para o armazenamento, é extremamente aconselhável.

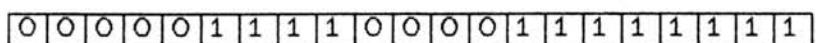
2.2.2.1 Compactação

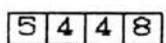
A primeira maneira de armazenar uma imagem é através da gravação linha a linha, coluna a coluna, sem o uso de qualquer tipo de compactação. Neste caso é possível armazenar cada pixel em um byte para imagens monocromáticas, ou em três bytes por pixel para imagens coloridas. Além da imagem propriamente dita, também terá que ser armazenado o número de linhas e colunas da matriz imagem, de forma que esta possa ser novamente reconstituída. Este sem dúvida não é o melhor método, mas é o mais simples.

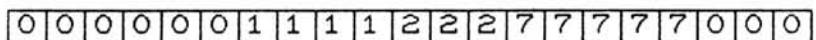
Um método um pouco mais sofisticado, consiste em realizar a compactação a nível de bytes, ou seja, caso a imagem possua um número de bits/pixel que não seja múltiplo exato do número de bits de um byte, compacta-se a imagem de forma que todos os bits de cada byte sejam utilizados. Nesta técnica, alinham-se as informações a respeito de cada pixel, uma após a outra, sem considerar a organização da memória orientada a bytes, ocupando-se todos os bits.

O primeiro método real de compactação das informações que será analisado aqui é o emprego da técnica RLE (*run length encoding*) [TIN 80][YAS 80]. Através desta técnica, faz-se uma codificação visando aproveitar o grande número de pontos vizinhos que possuem a mesma informação de cor em uma imagem (figura 2.5).

Existem diferentes formas de executar uma compactação através do método RLE. O processo geral consiste basicamente em descrever uma linha da imagem, não através dos valores de cada ponto, mas através do valor da cor seguido de um indicador do número de ocorrências consecutivas de pontos desta cor. A compactação por este processo é extremamente simples, mas em compensação obtém-se discretos índices de compressão dos dados para certos tipos de imagens [GOM 90].

 Dois níveis

 Codificação RLE (2 níveis - Alternância)

 Tons de Cinza

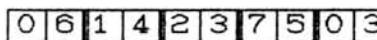
 Codificação RLE (Vários níveis)

Figura 2.5 - Exemplo de compactação RLE

Atualmente existe uma série de algoritmos de compressão de dados: código de Huffmann [HUF 72], RLE [TIN 80], BTC [GOM 90], RAC [YAS 80], etc. Mas entre os mais usados com imagens, destacam-se [GRA 89]: o RLE e principalmente o algoritmo LZW (*Lempel-Ziv & Welch Algorithm*) [WEL 84][REG 90]. Este algoritmo possui a característica de obter taxas de compressão de imagens em torno de 50% e ocasionalmente, atinge taxas de compressão de 90% da imagem. É um algoritmo que se destaca pela rapidez, tanto na fase de compactação, como na descompactação, sendo adotado nos mais diferentes padrões de armazenamento de imagens como forma de compactação [GRA 89].

2.2.2.2 Formato de Arquivos

Existe uma série de especificações de formatos de arquivos utilizados na descrição de imagens. Cada fabricante de um equipamento ou software para aquisição e manipulação de imagens cria um novo formato adequado a sua aplicação.

A solução para este número crescente de formatos diferentes e incompatíveis entre si, é a adoção de um formato padrão e a criação de conversores deste formato para os outros formatos de arquivos (e vice-versa) que possam ser necessários.

Através da adoção de um formato padrão, pode-se desenvolver uma série de ferramentas que sejam capazes de manipular imagens descritas neste padrão. Caso seja necessário compatibilizar as ferramentas com um novo formato de arquivo, basta criar um conversor de arquivos do novo formato para o adotado, realizando uma prévia conversão dos arquivos. A vantagem é que não será necessário alterar todas as ferramentas já desenvolvidas para adicionar novas rotinas de acesso aos dados descritos em um novo formato, pois o formato adotado continuará sendo o mesmo.

Atualmente alguns dos formatos mais utilizados para a descrição e armazenamento de imagens (usados principalmente junto a softwares de *desktop publishing*) são o GIF, TIFF, CUT, PCX e NAPLPS [GRA 89].

Um grupo formado por pesquisadores, professores e alunos que atuam na área de Processamento Gráfico no CPGCC/UFRGS, após debater o problema da incompatibilidade de diferentes formatos de imagens, decidiu especificar e adotar um formato padrão próprio. Este formato de arquivos deverá ser utilizado no armazenamento de imagens geradas por aplicações desenvolvidas no CPGCC. O formato do CPGCC/UFRGS é descrito no anexo A.1, juntamente com os outros formatos adotados na implementação do sistema N²OCR. O objetivo da adoção deste padrão é tornar as diferentes ferramentas desenvolvidas no CPGCC compatíveis entre si a nível de troca de arquivos. Além da adoção deste padrão, já está previsto o desenvolvimento de conversores de formato, possibilitando o acesso das referidas ferramentas a diversos outros padrões. Alguns destes conversores já foram implementados e atualmente está sendo desenvolvida uma biblioteca de funções para manipulação de arquivos descritos no padrão do CPGCC.

2.2.3 Exibição

Uma imagem digital pode ser exibida através do uso de um dispositivo de saída do tipo matricial, ou seja, um dispositivo cuja representação e geração da imagem seja feita através do uso de uma matriz de pontos. Assim como existem diferentes tipos de imagens digitalizadas, onde varia a resolução e o tipo de informação associada a cada pixel, os dispositivos de saída também possuem características próprias referentes a estes dois tipos de propriedades.

Quando uma imagem vai ser exibida em um dispositivo de saída, este dispositivo pode ou não possuir a capacidade para a exibição "imediata" da imagem. Em alguns casos pode ser necessário alterar a resolução espacial da imagem ou a quantificação dos valores de cada pixel, para que então esta possa ser exibida. A alteração da resolução espacial pode ser obtida por truncamento ou preenchimento de espaços, ou, por expansão ou compressão em termos de escala da imagem. Estas técnicas serão analisadas posteriormente, no item referente à implementação do sistema de pré-processamento de imagens do N²OCR, que é objeto desta dissertação. Serão abordados agora casos nos quais as imagens possuem um número de cores superior ao número de cores disponíveis no dispositivo de saída, onde é necessário o emprego de um algoritmo especial como o *halftoning* ou *dithering* [FOL 82][GOM 90].

2.2.3.1 Halftoning

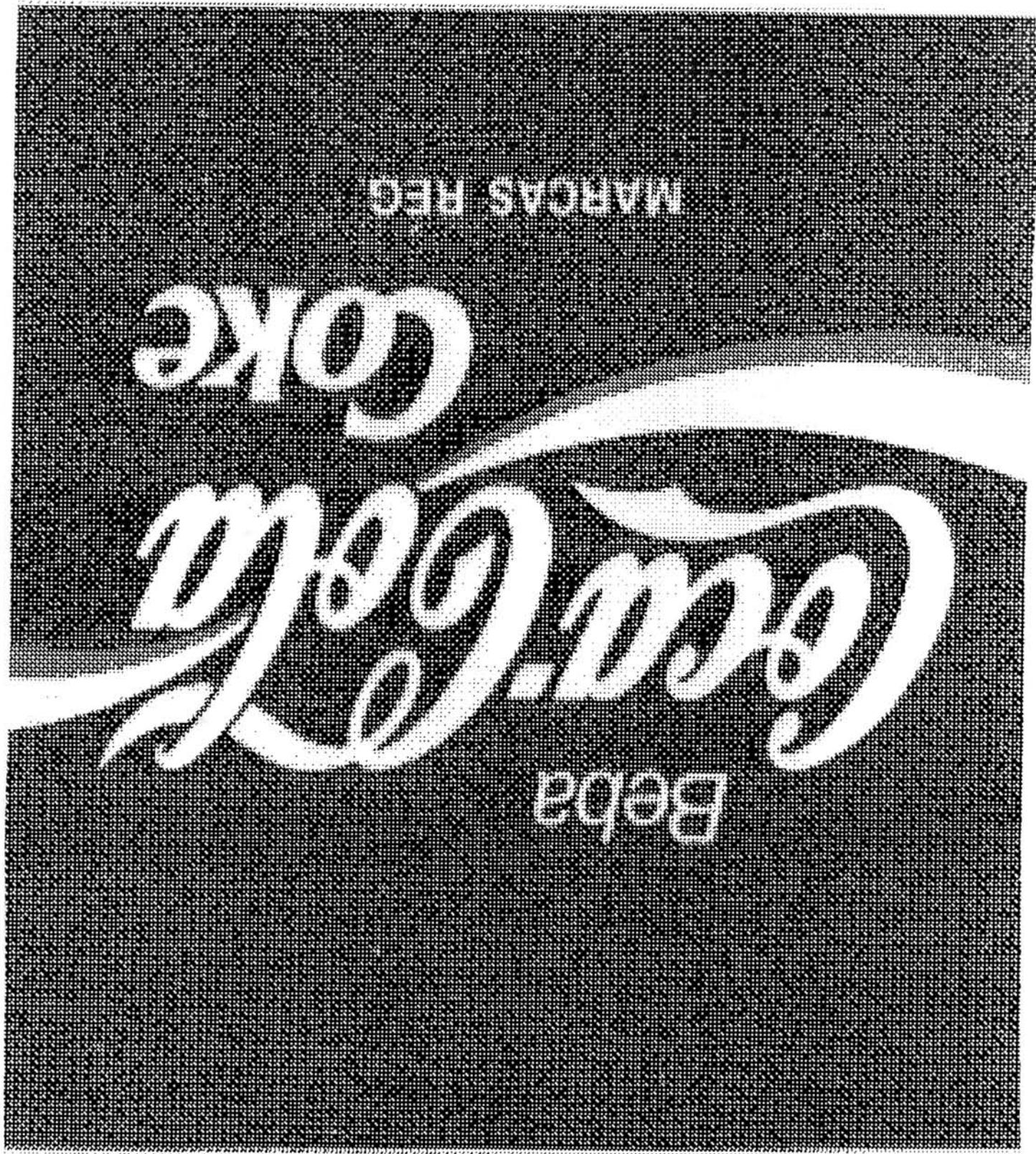
A aproximação por *halftoning* (meios-tons) é obtida quando se usa apenas dois níveis de intensidade para apresentar imagens com um número superior de níveis. Como é feita esta expansão da faixa de intensidades disponíveis? A resposta vem da capacidade do próprio olho humano de realizar uma "integração espacial" de cores e intensidades.

Olhando-se uma área onde os elementos observados possuem uma resolução espacial superior à capacidade do olho humano em distinguir elementos, o olho integrará os mínimos detalhes contidos nesta área e guardará apenas a intensidade média de tons contidos em toda a área.

Este fenômeno é usado na impressão de fotos preto e branco em jornais, revistas e livros; usando a chamada técnica de *halftoning*. Cada pequena unidade de resolução é impressa com um círculo de tinta preta, cuja área é proporcional ao inverso do nível de intensidade da cor (sistema subtrativo) da área correspondente da foto original. Esta técnica pode ser empregada em dispositivos de saída com capacidade gráfica limitada em termos de cores, onde a resolução do dispositivo é, preferencialmente, superior à resolução original da imagem.

A resolução final da imagem, gerada por este processo, é alterada. Supondo que seja utilizada uma imagem de $M \times N$ pixels, e que esta imagem será recodificada através do uso de um padrão de *halftoning* (pequena matriz de pontos - figura 2.7) de $K \times K$; a imagem resultante da aplicação do padrão de $K \times K$ elementos na imagem de $M \times N$ pontos terá uma dimensão de $K.M \times K.N$ pontos.

Esta técnica é importante devido à sua larga utilização, não somente para a visualização de imagens de múltiplos tons em dispositivos de capacidade limitada, mas também devido à sua utilização junto a dispositivos de digitalização do tipo *scanners*. Muitos destes dispositivos de aquisição de imagens, por serem orientados principalmente ao uso em aplicações de *desktop publishing* onde a imagem final é impressa em dispositivos de apenas dois níveis de intensidade, possuem implementada em hardware esta técnica. Este fato implica que muitas vezes a imagem resultante de uma digitalização possua apenas dois níveis de intensidade real, mas onde tem-se indiretamente um número superior de tonalidades representadas através do *halftoning*. A figura 2.6 apresenta exemplos do emprego desta técnica.





(b)

Figura 2.6 - Exemplos de aproximação de intensidades através do uso de *halftoning*

Como, no caso do processamento de imagens de textos, não se deseja a utilização deste tipo de imagens, torna-se necessário desfazer o *halftoning*. Para se obter novamente uma imagem com N níveis de intensidade, basta que seja definida a dimensão da matriz $K \times K$ utilizada na geração do *halftoning*. A partir desta informação, cada conjunto de $K \times K$ pontos é substituído por um único ponto onde a intensidade deste é igual ao valor da soma total de pontos acesos (soma das intensidades dos pixels). Através do uso deste procedimento, a imagem será reduzida em suas dimensões por um fator igual a K . Somente após desfeito o *halftoning* é que a imagem estará pronta para sofrer o processamento adequado, visando o reconhecimento de caracteres.



Figura 2.7 - Matrizes de *halftoning* usadas na geração de cinco tons

O método do *halftoning* é apropriado somente se a resolução da imagem a ser exibida é inferior à resolução do dispositivo de saída, permitindo o uso de múltiplos pixels do dispositivo de saída para exibir uma informação equivalente a de apenas um pixel da imagem. Esta resolução do dispositivo de saída também deve ser razoavelmente alta, de forma que o observador não perceba os elementos constituintes da matriz de *halftoning*, como elementos individuais.

2.2.3.2 Dithering

Quando a resolução da imagem deve ser mantida, é empregada uma outra técnica denominada de *dithering* ou *order dither*. Através do uso desta técnica pode ser exibida uma imagem de $M \times N$ pontos com múltiplos níveis de intensidade, em um dispositivo com $M \times N$ pontos de dois níveis.

No *dithering*, a decisão de intensificar ou não o pixel (x, y) , depende da intensidade desejada $I(x, y)$ neste ponto, e de uma matriz de *dither* $K \times K = D^{(K)}$. A matriz *dither* é indexada de 0 à $K-1$ ao longo de suas linhas e colunas. Cada um dos inteiros de 0 à K^2-1 aparece uma vez em uma posição da matriz. Quando $K=2$, a matriz de *dither* será $D^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$.

Para processar um ponto em (x, y) é computada a seguinte equação:

$$\begin{array}{ll} i = x \text{ módulo } K & \text{Se } I(x, y) > D_{ij}^{(K)} \\ j = y \text{ módulo } K & \text{Então o ponto } (x, y) \text{ é intensificado} \\ & \text{Senão o ponto } (x, y) \text{ é apagado} \end{array}$$

Matrizes de vários tamanhos foram desenvolvidas por Bayer [BAY 73] para minimizar a quantidade de textura que pode ser introduzida na imagem exibida. A matriz de *dither* permite a especificação de números diferentes de níveis de intensidade da imagem, onde esta deve possuir uma dimensão de K , sendo K o números de níveis representados.

Estas técnicas de *dithering* e *halftoning* são muito importantes para o estudo junto ao processamento digital de imagens, devido a sua larga utilização. Isto implica o fato de existirem arquivos com imagens digitalizadas com apenas dois níveis, que na realidade representam imagens de outra classe, ou seja, imagens com um número de níveis superior a dois, mas onde estes estão representados implicitamente através da utilização de técnicas como o *halftoning* e o *dithering*. Obviamente, as imagens deste tipo sofrerão um tratamento diferenciado, apesar de possuírem apenas dois níveis.

2.3 Tratamento de Imagens

As imagens digitais podem sofrer algumas modificações, através de um tratamento visando a alteração de suas características, tanto de resolução quanto de quantização. Uma das primeiras técnicas de tratamento de imagens já foi abordada no item 2.2.3.1 e consiste em fazer ou desfazer o *halftoning* em uma imagem, alterando as características de representação das intensidades luminosas dos seus pontos. A seguir serão abordadas algumas das principais técnicas empregadas no tratamento de imagens, e que estão relacionadas com as aplicações de reconhecimento de textos. Neste item e nos seguintes será abordado apenas o tratamento das imagens de dois níveis ou de uma escala de cinza. As imagens coloridas são consideradas como um caso especial, podendo ser tratadas tomando-se apenas uma escala de intensidades de uma dada cor por vez.

As operações de tratamento de imagens podem ser divididas em dois grupos principais: operações pontuais e operações locais. As operações pontuais são aquelas que consideram apenas a informação contida anteriormente em um dado elemento da imagem que está sendo processado, sem levar em consideração o valor dos outros pontos da imagem.

As operações locais levam em consideração os aspectos referentes à vizinhança de um dado ponto, ou seja, um ponto da imagem recebe contribuições de outros pontos vizinhos no cálculo de seu novo valor. Serão apresentadas nas seções seguintes técnicas baseadas em operações pontuais (modificação da escala de cinza) e em operações locais (filtros).

2.3.1 Histograma

Quando se trabalha com imagens do tipo escala de cinza, uma das primeiras operações comumente realizadas é o levantamento do histograma da imagem. Um histograma é um gráfico das freqüências de ocorrência de cada nível de intensidade. Um "pico" (ponto elevado) no histograma representa a ocorrência de um certo nível de intensidade luminosa em grande número, destacando-se dos demais.

Através da visualização do histograma da imagem, pode-se obter uma série de informações, que guiarão o usuário na aplicação de transformações da imagem através de operações pontuais.

O histograma é obtido a partir da aplicação do algoritmo apresentado a seguir, e tem como resultado um gráfico semelhante ao da figura 2.8.

→ Algoritmo:

MaxNiveis - Número máximo de níveis de intensidade

MaxLinhas - Número 'M' de linhas de uma imagem $M \times N$

MaxColunas - Número 'N' de colunas de uma imagem $M \times N$

$i(u,v)$ - Devolve o valor do pixel de coordenadas (u,v)

HCI - Freqüências de ocorrência relativas de cada nível de intensidade

```

  Inicio.
    Inicializa HCID com 0;           0 ≤ I ≤ MaxNiveis
    Para u = 0 até MaxLinhas
      Para v = 0 até MaxColunas
        Incrementa HCi(u,v))
    Plota o gráfico de HCID
  Fim.

```

Uma vez gerado o histograma fica mais fácil de se determinar os tipos de operações a serem realizadas na imagem. Estas operações se caracterizam justamente pela realização de transformações que afetam diretamente o histograma da imagem (operações pontuais).

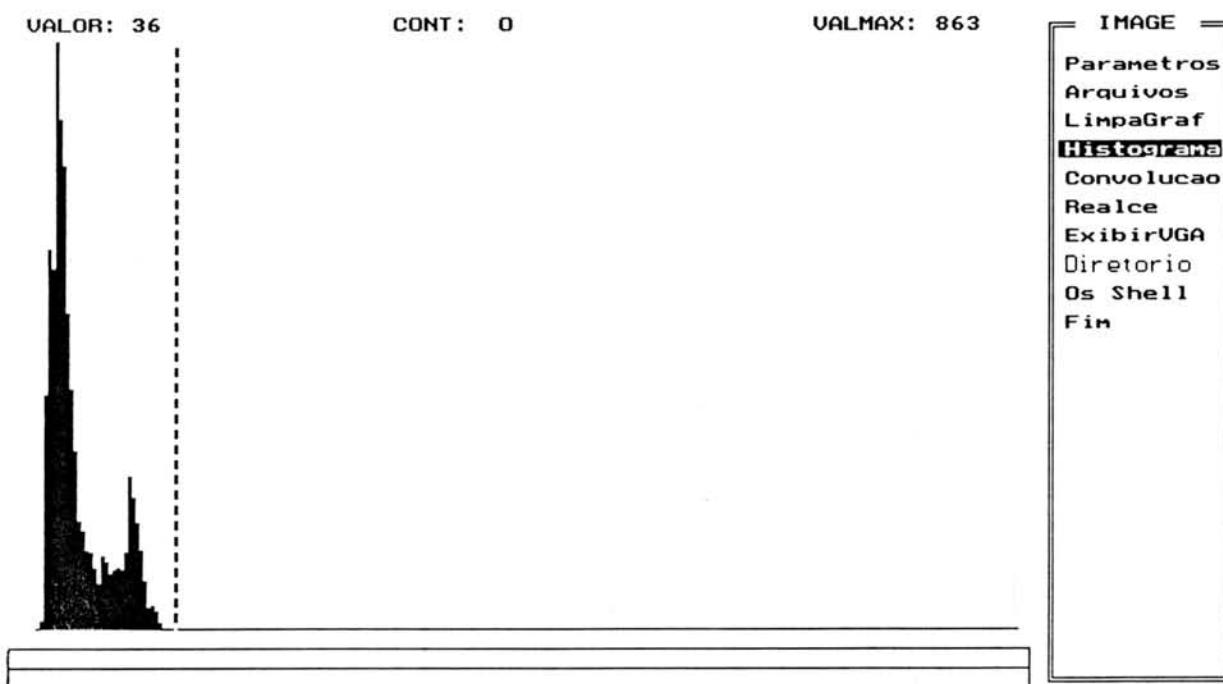


Figura 2.8 - Histograma da figura 2.6 (b) com o halftoning desfeito (matriz 6 × 6)

2.3.2 Modificação da Escala de Cinza

A modificação da escala de cinza é o tipo de tratamento de imagens referenciado no item anterior. Dentro desta categoria de operações destacam-se: equalização de histograma e limiarização. Outras operações também podem ser realizadas visando o realce das imagens através da alteração da escala de cinza, citando-se, a inversão, compressão e expansão dos níveis de cinza [SCH 87, PAZ 88 e MAS 89].

2.3.2.1 Equalização de Histograma

Para uma dada imagem, a equalização tem por objetivo reduzir a freqüência de ocorrência de valores de intensidade muito comuns e aumentar a freqüência de ocorrência dos valores relativamente raros. Desta forma o histograma da imagem terá uma aparência mais equilibrada, e possibilitará a obtenção de imagens com um número maior de detalhes visíveis, facilitando a sua visualização. Através da exibição da imagem e de seu respectivo histograma equalizado, fica mais fácil para o usuário a determinação exata do limiar a ser utilizado na binarização da imagem, assunto este que será abordado no item 2.3.2.2 próximo.

A equalização do histograma pode ser conseguida particionando a Distribuição Cumulativa de Freqüências (DCF= Soma dos valores do histograma em cada nível de intensidade), em um maior número de intervalos para os níveis com maior freqüência e em menor número para os níveis mais raros. O algoritmo de equalização de histogramas pode ser encontrado na literatura [PAV 82 e SCH 87].

2.3.2.2 Limiarização

A limiarização de uma imagem, do tipo escala de cinza, consiste em se determinar um valor limiar (*threshold*) de forma que todos os valores inferiores a este limiar sejam levados para um mesmo valor base, usualmente 0. Todos os valores superiores a este limiar serão levados para um mesmo valor de topo, usualmente 1. Desta forma é realizada a chamada binarização da imagem, onde, partindo de uma imagem do tipo escala de cinza, obtém-se uma imagem com apenas dois níveis, como é demonstrado na figura 2.9.

$$\text{Limiarização : } i(u,v) = \begin{cases} 1 & \text{se } i(u,v) > L \text{ (Limiar)} \\ 0 & \text{se } i(u,v) \leq L \end{cases}$$

A binarização das imagens é de grande uso e de destacada importância quando são utilizadas imagens digitalizadas de textos. Uma vez que um texto deve possuir apenas duas tonalidade, a cor do fundo e a cor do caractere, toda a imagem digitalizada deverá ser convertida em uma imagem de apenas dois níveis. Imagens com mais de dois níveis de intensidade devem sofrer o processo de binarização. Como foi descrito acima, neste processo é necessária a definição de um valor para o limiar (*threshold*), que será utilizado na binarização. Este limiar pode ser estabelecido de forma interativa durante o processo de digitalização, de forma a tornar o ajuste do mesmo mais perfeito.

Ao digitalizar um texto, pode-se levantar o seu histograma, o qual deve apresentar duas faixas de altas freqüências de ocorrência de intensidades. Estas duas faixas com picos no histograma definem normalmente as probabilidades mais altas de ocorrências das duas cores básicas da imagem (cor do fundo e cor do caractere). Neste caso, o limiar adequado pode ser estabelecido visualmente, determinando-se um ponto localizado no "vale" formado entre os dois picos do histograma.



Figura 2.9 - Binarização da imagem do texto da figura 2.6(a)

Uma vez que esteja sendo considerada a digitalização de diversas folhas de texto, sob as mesmas condições de iluminação e com os textos possuindo a mesma qualidade de reprodução, é aconselhável que se faça um primeiro ajuste interativo do limiar. Analisando interativamente o histograma, a imagem original e a imagem binarizada, ajusta-se o limiar até se obter a melhor imagem possível, sem a inserção acentuada de ruído ou de artefatos adicionais, e sem perder um grande número de pontos referentes aos caracteres. Uma vez estabelecido o valor ideal do limiar, as folhas restantes podem então ser digitalizadas fazendo uso do mesmo limiar pré-estabelecido. O limiar deve ser alterado somente quando forem alteradas as condições de digitalização: qualidade do texto impresso e iluminação.

Uma outra possibilidade é o ajuste automático do valor do limiar [NAK 87], mas este método não é muito aconselhável devido à dificuldade de se criar um algoritmo rápido e preciso, que ache o ponto ótimo onde ocorre a menor perda de informação e a menor inserção de ruído na imagem. Surge o problema: como determinar o que é ruído e o que é informação ?

Como o processo de digitalização já envolve uma calibração e ajustes manuais, o método do ajuste interativo do limiar torna-se uma opção prática, viável e de bons resultados. O ser humano, quando bem informado, possui a capacidade de identificar com uma boa precisão qual é o limiar ideal. Após definido o limiar, o processo de binarização pode ser executado por um software ou fazer uso de um hardware desenvolvido para este fim, o que possibilitaria o processamento da imagem em tempo real.

O processo de limiarização pode não ficar restrito apenas à binarização através da aplicação de um limiar único. É possível definir diversas faixas de valores sobre os quais será realizada a limiarização. Neste trabalho foi abordada apenas a binarização por ser este um processo de destacada importância para os sistemas de reconhecimento de caracteres.

2.3.3 Filtros

As imagens digitais podem sofrer uma série de operações denominadas operações com filtros. Estas operações realizam tarefas tais como: ressaltar elementos, suavizar ou aumentar o contraste, detectar bordas, remover ruído, etc. Os filtros podem ser implementados para realizar as transformações das imagens nos assim denominados domínio espaço ou no domínio freqüência.

2.3.3.1 Operações no Domínio Espaço

As operações no domínio espaço são realizadas diretamente com os pixels da imagem, o que é uma vantagem, pois a imagem não sofre transformações prévias e posteriores para poder ser processada, ao contrário do que ocorre com as operações realizadas no domínio freqüência onde a imagem tem que ser transformada do domínio espaço para o domínio freqüência para poder ser tratada e, então, transformada novamente para o domínio espaço.

Um exemplo de filtro de destaqueada importância que realiza operações locais no domínio espaço é a convolução, a qual será abordada no próximo item.

2.3.3.1.1 Convolução

A convolução é um exemplo de operação local, pois esta leva em consideração os aspectos referentes à vizinhança de um dado ponto, ou seja, a imagem é tratada considerando-se o fato dos elementos desta possuirem relações entre si e estarem dispostos em uma organização bidimensional. Como exemplo deste tipo de operação local, tem-se a detecção da presença de uma linha horizontal, onde é necessário se levar em consideração, em relação a um dado ponto a ser analisado, os pontos laterais esquerdo e direito.

A operação de convolução pode ser realizada diretamente sobre a imagem no domínio espaço. Dadas duas matrizes de pontos, a convolução é realizada através da aplicação de uma destas matrizes, denominada de operador ou máscara de convolução, sobre a outra matriz que é a imagem. A descrição matemática desta operação é a seguinte:

Máscara de Convolução: $H(j,k)$ onde $0 \leq j \leq J$ e $0 \leq k \leq K$

Imagen a Processar : $i(u,v)$ onde $0 \leq u \leq M$ e $0 \leq v \leq N$

Imagen já Processada : $i'(u,v) \rightarrow$ Imagem $M \times N$

Fator de Normalização: F - Pode ser aplicado após ou já estar inserido diretamente na matriz $H(j,k)$

$$i'(u,v) = \frac{1}{F} \cdot \sum_{j=0}^J \sum_{k=0}^K i(u+j, v+k) \cdot H(j,k) \quad \text{Para cada } i(u,v)$$

As máscaras de convolução possuem uma dimensão normalmente igual a 3x3 pontos, sendo que esta dimensão já é suficiente para a maioria das operações. Isto implica que para cada ponto calculado, estão sendo levados em consideração o próprio ponto mais os seus oito vizinhos imediatos. Para matrizes desta dimensão, obtém-se:

$$\begin{aligned}
 i'(u,v) = & \dots \\
 & (i(u-1, v-1).H(0,0) + i(u-1, v).H(0,1) + i(u-1, v+1).H(0,2) + \\
 & i(u , v-1).H(1,0) + i(u , v).H(1,1) + i(u , v+1).H(1,2) + \\
 & i(u+1, v-1).H(2,0) + i(u+1, v).H(2,1) + i(u+1, v+1).H(2,2)).1/F
 \end{aligned}$$

O valor "1/F" é usado para normalizar o resultado, porque o somatório aplicado sobre cada ponto pode resultar em um acréscimo indesejado no valor da intensidade do mesmo. Por exemplo, uma máscara de convolução para cálculo da média simples [PAZ 88] contém os seguintes valores:

$$H(j,k) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

MATRIZ ORIGINAL (F=9)

$$H'(j,k) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

MATRIZ COM O FATOR F EMBUTIDO

A máscara $H(j,k)$ pode resultar em um acréscimo do valor de um pixel de até 9 unidades, logo o valor para a constante de normalização F deve ser igual a 9. No caso da máscara $H'(j,k)$ este fator já se encontra aplicado sobre cada elemento da matriz.

As máscaras de convolução são extremamente úteis, pois permitem realizar operações de filtragem, eliminação de ruído, detecção de pontos ou linhas e ressaltar bordas ou contornos (figura 2.10). Algumas destas máscaras usualmente empregadas em Processamento Digital de Imagens são [KIS 90 e MAS 89]:

*Detecção de
Pontos*

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

*Detecção de
Linhas Horizontais*

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

*Detecção de
Linhas Verticais*

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

*Detecção da
Diagonal Esquerda*

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

*Detecção da
Diagonal Direita*

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

*Detecção de
Bordas*

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Figura 2.10 - Imagem original e imagem após convolução para detecção de contornos

A convolução torna-se uma opção interessante como forma de pré-processamento das imagens de textos digitalizados, uma vez que esta possibilita a remoção de ruídos ou ressaltar elementos. É interessante ressaltar que também é possível a realização de convoluções em tempo real, através do uso de um hardware dedicado, como o proposto por Netto [NET 89] - o *CV2D*.

2.3.3.2 Operações no Domínio Freqüência

Como exemplo de operações realizáveis no domínio freqüência tem-se os filtros "passa baixa" e "passa alta". Os filtros "passa baixa" atenuam as altas freqüências e servem para realizar uma suavização (*smoothing*) da imagem. Os filtros "passa alta" atenuam as baixas freqüências e são utilizados para realçar as bordas (*sharpening*).

A convolução também pode ser realizada no domínio freqüência. Para tanto, a imagem descrita no domínio espaço é primeiramente transformada para o domínio freqüência quando, então, é realizada a filtragem. Após, a imagem sofre uma nova transformação para retornar ao domínio espaço. Uma transformada muito empregada é a Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*) que é aplicada sobre dados discretos [TRI 90 e PAV 82].

O uso de transformadas implica altos custos em computacionais e tempos demorados de resposta do sistema. Neste trabalho, não será considerada a sua utilização.

2.4 Reconhecimento de Padrões

Conforme o dicionário Webster apud Hall [HAL 79], o reconhecimento de padrões consiste em perceber ou identificar um elemento a partir de uma experiência passada ou conhecimento, ou por algum detalhe deste padrão.

O elemento a ser reconhecido é denominado de padrão, o qual é definido como sendo uma amostra representativa de uma classe ou tipo.

Segundo o "Novo Dicionário Aurélio da Língua Portuguesa" [FER 86], reconhecer também pode ser definido como: conhecer novamente quem ou o que se conheceu noutro tempo. Estas definições, a partir de um conceito mais amplo do que aquele relacionado apenas com o Processamento de Imagens, demonstram algumas das características inerentes a todo o sistema de reconhecimento de padrões.

O objetivo do reconhecimento de padrões junto ao Processamento de Imagens é portanto o de extrair, detectar e identificar elementos presentes em uma cena. Desta maneira, procura-se imitar o ser humano e suas habilidades, tentando criar dispositivos e algoritmos capazes de realizar as mesmas funções realizadas pelo homem como: localizar objetos, classificar padrões e detectar relações entre estes. Quais são as vantagens de se realizar o reconhecimento de padrões através dos computadores? Como já foi citado, a principal vantagem é a de capacitar máquinas a realizem tarefas até então só possíveis de serem realizadas por seres humanos. As aplicações do reconhecimento de padrões incluem funções como: compactação de dados [PRA 80], aumento da capacidade de aquisição de dados pelo computador [HAR 72], sensoriamento remoto [MAS 89], controle de qualidade, classificação de objetos e uma série de outras aplicações científicas e comerciais.

O reconhecimento de padrões, *in latu sensu*, abrange uma gama muito extensa de aplicações e tipos diferentes de objetivos, estando subdividido em duas principais áreas, de acordo com o tipo de objetivo a ser alcançado: obtenção de descrições (segmentação, vetorização e detecção) e obtenção de classificações (casamento de padrões, identificação de um elemento pertencente a uma classe).

2.4.1 Obtenção de Descrições

Nesta categoria se encontram os procedimentos que visam apenas a detecção de algum tipo de elemento em uma imagem digital. Como exemplo de obtenção de descrições encontram-se os algoritmos para a vetorização [PAR 88 e JIM 82], que tem por objetivo transformar uma descrição de uma imagem a partir de uma matriz bidimensional de pontos em uma descrição através de um conjunto de vetores. Fica bem claro que, neste tipo de procedimentos, não interessa realizar uma classificação dos elementos presentes em uma dada cena, o objetivo é apenas a detecção de algumas características e relações que fazem com que certos elementos da imagem estejam relacionados entre si.

Existem dois termos que estão intimamente relacionados com o reconhecimento de padrões visando a obtenção de descrições, são eles: segmentação e detecção. A segmentação consiste na extração de zonas da cena que possuem uma mesma cor ou textura e que consequentemente possuem elementos relacionados entre si. Normalmente este "agrupamento" de elementos relacionados constitui um objeto. A detecção pode ser empregada para a extração de vetores, ou até para a identificação de presença de um padrão mais complexo em uma cena. Ambos os métodos de reconhecimento de padrões para a obtenção de descrições resultam usualmente em uma descrição dos objetos em um nível mais elevado, e não em uma classificação destes. Tais processos podem implicar algumas vezes na categorização dos objetos presentes na cena, mas neste caso já estaria sendo implementada uma técnica de obtenção de classificações.

2.4.2 Obtenção de Classificações

Esta forma de reconhecimento consiste da atribuição de uma dada imagem (padrão teste) a um dos conjuntos de classes de imagens pré-definidas por algum método [KOV 80].

Este reconhecimento é obtido através de um processo de abstração, onde descarta-se os detalhes não essenciais e destaca-se as propriedades que são relevantes consideradas como sendo as características definidoras da classe. Esta abstração leva à identificação de uma classe de elementos com atributos semelhantes na qual o padrão de testes será incluído, como sendo mais um exemplar desta. O objetivo principal neste tipo de reconhecimento é a maximização da função de verossimilhança (*maximum likelihood*) e a minimização da probabilidade de se fazer uma decisão incorreta.

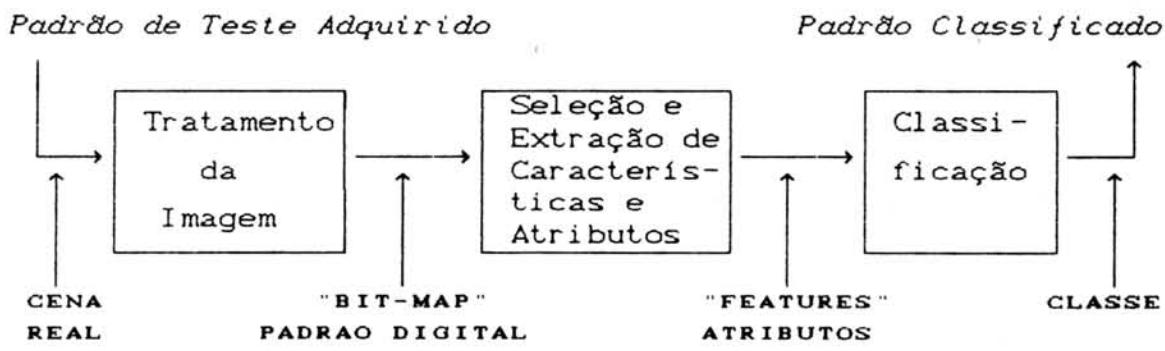
Uma classe de padrões se caracteriza por possuir um conjunto de propriedades essenciais comuns a seus componentes. O objetivo da classificação é agrupar os padrões em suas respectivas classes, de acordo com suas propriedades. Na classificação será empregada uma função de decisão, a qual é responsável pela decisão ótima, em relação aos atributos conhecidos, determinando o(s) conjunto(s) possível(eis) de classificação ao(s) qual(is) pertence o padrão analisado. É importante ressaltar que, na classificação, cada padrão é usualmente atribuído a apenas uma classe, aquela que melhor representa o padrão. No entanto, é possível que um mesmo padrão possa ser atribuído a diversas classes ou que ao invés de se identificar apenas a melhor classe representante do padrão, seja identificado um conjunto de "N" classes com as quais o padrão está relacionado dentro de certos limites.

O reconhecimento de padrões para a obtenção de uma classificação é um procedimento que pode ter uma grande complexidade, envolvendo diversos tipos de processamento de imagem. Os passos a serem realizados para o reconhecimento de imagens são [HAL 79]:

- 1) Observar os atributos e características dos objetos;
- 2) Selecionar deste conjunto de características aquelas que são úteis para a classificação;

- 3) Obter uma função de decisão ótima para a classificação, baseada na análise de atributos;
- 4) Avaliar a eficácia da classificação.

É possível se visualizar estas etapas, como sendo processos de caracterização, abstração e generalização. O processo de classificação de padrões pode envolver inclusive etapas referentes ao tratamento da imagem ou à obtenção de descrições, como forma de garantir uma melhor caracterização do padrão. A figura 2.11 apresenta um esquema global que é utilizado usualmente no processo de reconhecimento e classificação de padrões, apresentando as diversas etapas do processamento para este fim.



ETAPAS:

1. TRATAMENTO DE IMAGENS
Realce → Restauração → Reconstrução
2. ANALISE ESTRUTURAL DE PROPRIEDADES
Segmentação → Detecção/Descrição → Classificação

Figura 2.11 - Esquema geral de um processo de reconhecimento e classificação de imagens

A classificação pode ocorrer em diferentes níveis, em relação ao algoritmo de definição das classes ou quanto ao tipo de algoritmo de classificação dos padrões.

O algoritmo de definição das classes servirá para criar o conjunto de classes possíveis ao qual será atribuído um padrão de teste. A criação do conjunto de classes, ou seja, a definição de grupos que representam conjuntos de atributos similares, pode ser realizada de duas formas: supervisionada ou por aglomeramento (*clustering*).

A criação de classes de uma forma supervisionada (*supervised*), consiste da apresentação de amostras padrões (*sample patterns*) para cada classe. Estas amostras possuem um rótulo de identificação da classe à qual pertencem. Estes padrões rotulados podem ser uma descrição de baixo nível como uma matriz de pixels, ou uma descrição de nível superior, onde está representado um conjunto de atributos do padrão, ao invés do próprio padrão. A partir destes padrões rotulados, é possível se identificar as diferentes classes existentes e durante o processo de classificação busca-se relacionar os padrões de teste com alguma das classes pré-definidas.

Os algoritmos de *clustering* visam, de uma forma automática, a agrupar os padrões com características similares em conjunto de classes, sem no entanto haver uma identificação prévia das classes existentes para a classificação. Os algoritmos de *clustering* possuem a capacidade de determinar automaticamente as classes, sem a necessidade de uma etapa de apresentação de padrões previamente rotulados.

Quanto aos tipos de algoritmos de classificação, em relação aos padrões, tem-se [KOV 80]: análise de padrões, análise de elementos, análise de contorno; ou, se for tomado como parâmetro a função de decisão, tem-se [SCH 87]: casamento de padrões, abordagem do discriminante ou abordagem sintática.

2.4.2.1 Análise de Padrões

O próprio padrão é utilizado neste processo como forma de identificação das classes. O conjunto de pontos que formam o padrão é utilizado diretamente como sendo um conjunto de atributos de classificação que descrevem uma classe.

2.4.2.2 Análise de Elementos

O padrão é transformado de forma a obter uma nova descrição que representa os elementos constituintes deste padrão. Nesta abordagem, pode-se, por exemplo, vetorializar um dado padrão e obter uma nova descrição deste. A classificação se dará não mais pela análise de cada elemento do padrão, mas sim pela análise dos atributos extraídos deste padrão.

2.4.2.3 Análise de Contorno

Neste caso, o padrão é descrito pela varredura de seus contornos. Este processo é usado no caso da digitalização obtida através de dispositivos do tipo *flying spot scanners* ou mesas digitalizadoras. É conveniente considerar-se a manipulação deste tipo de descrição a parte, pois diferem consideravelmente dos outros tipos de descrição, tanto em relação aos procedimentos de obtenção dos atributos de classificação, como em relação ao procedimento de classificação (usualmente utiliza uma abordagem sintática).

2.4.2.4 Casamento de Padrões

Este tipo de procedimento é também denominado de casamento de gabaritos (*template matching*). O casamento de padrões consiste da comparação de um padrão de teste (o próprio ou o conjunto de atributos extraídos deste) contra outro conjunto de padrões que representam as diferentes classes. O padrão usado como gabarito de comparação será o representante de toda uma classe, onde este pode ser um único padrão, no caso de termos um exemplar perfeito que represente toda uma classe, ou pode ser um exemplar obtido a partir de um conjunto de exemplares da classe. No caso de termos N gabaritos padrões para uma mesma classe, é necessário que o algoritmo usado na criação do gabarito de comparação seja um algoritmo do tipo adaptativo, ou seja, permita uma adaptação destes N padrões de forma a obter um gabarito de comparação médio que represente ao mesmo tempo todos os padrões representantes da classe.

No casamento de padrões, há um valor de controle de similaridade, usado na função de decisão. Este valor de controle indica o quanto um padrão de teste necessita "casar" exatamente com o gabarito de comparação. Este tipo de abordagem implica três tipos de respostas: reconhecimento correto, reconhecimento incorreto ou não reconhecimento, caso o índice mínimo de similaridade não tenha sido atingido para nenhuma das classes pré-definidas. Outra função de decisão pode ser a de se tomar sempre o padrão com o mais alto índice de similaridade, independente do valor deste índice; neste caso sempre haverá uma classificação do padrão (correta ou incorreta).

O casamento de padrões pode ocorrer ao nível de padrão inteiro (total) ou em relação a pequenas partes deste padrão (parcial) ou ainda através de descrições mais flexíveis, como as *rubber masks* [SCH 87] ou *correlation techniques* [HAL 79].

2.4.2.5 Abordagem do Discriminante

A tomada de decisão a respeito da classificação de um padrão é baseada neste caso em uma regra de decisão estatística, onde é levada em consideração uma medida de similaridade, de distância ou uma função discriminante. O caso do *template matching* pode ser considerado como um caso particular da abordagem do discriminante.

2.4.2.6 Abordagem Sintática

Nesta abordagem, o padrão é representado por um *string*, uma árvore ou um grafo de padrões primitivos e seus relacionamentos. A tomada de decisão é feita em geral a partir de uma análise sintática ou de um procedimento de *parsing*. Pode-se fazer uso de uma medida de similaridade (distância) entre dois *strings*, árvores ou grafos, de forma a melhorar a flexibilidade de classificação.

2.4.2.7 Avaliação do Processo de Classificação

Todos os diferentes processos de classificação de padrões citados acima devem sofrer uma avaliação, visando determinar parâmetros relativos ao seu desempenho, os quais caracterizam tais algoritmos e possibilitam análises que podem levar ao seu aperfeiçoamento. Na avaliação de um processo de classificação, serão levados em consideração três itens principais:

1) Reconhecimento Correto de Padrões: Índice representativo da percentagem do número total de padrões que foram corretamente classificados (*right match*).

2) Reconhecimento Incorreto de Padrões: Índice representativo da percentagem do número total de padrões que foram incorretamente classificados (*substitution/wrong match*).

3) Não Reconhecimento de Padrões: Índice representativo da percentagem do número total de padrões que foram rejeitados, indicando-se que provavelmente não pertencem a nenhuma das classes definidas (*not match*).

A redução da probabilidade de reconhecimento incorreto, deixando-se o sistema de classificação com uma função de decisão mais rígida, implica no aumento da probabilidade de rejeição, sendo igualmente válido o inverso. O objetivo principal de um sistema de reconhecimento de padrões deve ser reduzir ao mínimo as probabilidades de reconhecimento incorreto e de rejeição, sendo preferível, na maioria dos casos, uma rejeição de um dado padrão ao seu reconhecimento incorreto [KOV 80].

Este processo de avaliação é extremamente difícil de ser realizado, principalmente no que diz respeito a comparação de diferentes algoritmos. Isto se deve a esta avaliação ser muito dependente dos dados de teste utilizados, do conjunto de padrões de treinamento e teste. Sempre há um conjunto de dados para os quais um algoritmo terá uma eficácia superior a todos os outros algoritmos rivais; devido a isto deve-se tomar muito cuidado na avaliação e comparação dos resultados obtidos por diferentes técnicas [NAG 83]. Quanto a avaliação de sistemas de reconhecimento de padrões, o assunto será apresentado em maiores detalhes no item referente a avaliação de sistemas de OCR.

3 SISTEMAS DE RECONHECIMENTO DE CARACTERES

[Os sistemas de reconhecimento ótico de caracteres - OCR (Optical Character Recognition) são sistemas desenvolvidos para, de uma certa forma, reproduzir a capacidade humana de ler textos.] Este capítulo abordará os diferentes tópicos relacionados com o assunto, apresentando uma classificação dos sistemas de reconhecimento de caracteres em geral (não somente os óticos) e descrevendo as principais características e algoritmos empregados em tais sistemas.

Mas afinal, o que é "reconhecer um texto"? Qual o tipo de classificação que é realizado? Um OCR recebe um texto sob a forma de uma imagem digital e através de um processo de reconhecimento e classificação dos padrões encontrados, que representam os caracteres, obtém como resultado um código que representa cada caractere. Podem ser usados como códigos de identificação dos caracteres o ASCII, EBCDIC ou outro código similar.

3.1 Origens e Evolução

[As origens da tentativa de simular a leitura humana datam de 1870, quando foi inventado por Carey o scanner de retina. A partir da evolução dos dispositivos capazes de captar e traduzir imagens em sinais que podiam ser medidos, novos horizontes foram descobertos. Entretanto, a primeira tentativa bem sucedida de reconhecimento de caracteres foi realizada somente em 1900 pelo cientista russo Tyutin, que desenvolveu um sistema que visava o auxílio de deficientes visuais [MAN 86].

A partir do desenvolvimento do computador digital nos anos 40, esta área de estudos ganhou um grande impulso, pois até o momento os dispositivos usados eram muito complexos e tinham que ser desenvolvidos unicamente com o propósito de reconhecer caracteres.

Os computadores digitais tornaram mais viáveis os estudos a respeito dos sistemas de reconhecimento de caracteres. Já na década de 50, são encontradas uma série de publicações e relatos a respeito de tais sistemas, como em Dineen (1955) e Bomba (1959), conforme citação em [HAR 72]. Nos anos 60, grandes avanços continuaram sendo realizados e esta área de estudos da computação começou a se consolidar [STE 61].

A partir desta época o crescimento da área de estudos do reconhecimento de caracteres foi muito veloz, principalmente devido às inovações tecnológicas que permitiram a utilização de dispositivos mais sofisticados, precisos e acessíveis em termos de custos, tanto para a aquisição como para o processamento dos dados. Um grande número de pesquisas e trabalhos foram desenvolvidos e publicados a respeito do assunto, possibilitando uma maior difusão e evolução da área.

Atualmente, uma grande quantidade de patentes e publicações específicas sobre sistemas OCR tem sido feitas, sendo que tais sistemas se tornaram disponíveis, tanto para aplicações comerciais de pequeno, médio ou grande porte, como para aplicações domésticas. Os objetivos atuais continuam sendo a busca de uma redução dos custos e o aumento da eficiência, bem como da flexibilidade de tais sistemas, sendo que ainda há muito a ser feito nesta área no futuro.

Mas por que este objetivo de desenvolver sistemas de reconhecimento de caracteres tem sido tão duramente buscado e gasto tantos esforços dos cientistas a procura de melhores soluções? Basicamente isto é devido às inúmeras aplicações possíveis de tais sistemas, tais como:

- └ - Aquisição de dados numéricos comerciais [SUE 80];
- Compactação de dados [PRA 80];
- Leitura automática de formulários [DAI 90];
- Identificação de endereçamento postal [SCH 78];
- Reconhecimento de partituras musicais [SUE 80];
- Leitura de cheques bancários [HAR 72];
- Sistemas de aquisição de textos para a tradução automática [HAR 72];

- Interface homem x máquina mais natural e sem necessidade de recodificação dos dados (sujeitos a um menor número de erros, pois não é necessário redigitar os textos, introduzindo novos erros) [HAR 72] e

- Auxílio a deficientes visuais, na leitura de textos (tradução e impressão automática em Braille) [HAR 72];

Esta lista de aplicações poderia ser estendida por muitos e muitos outros itens ainda. Como pode ser visto, o número de aplicações é bem amplo, mas além de todas estas possibilidades de utilização dos sistemas de reconhecimento de caracteres, ainda há um outro fator muito importante: o grande desafio de se conseguir replicar em uma máquina as funções humanas.

3.2 Tipos de Sistemas de Reconhecimento de Caracteres

Os sistemas de reconhecimento de caracteres podem ser desenvolvidos utilizando-se diferentes procedimentos tanto de aquisição de dados como de processamento de informações além de serem orientados para diferentes tipos de aplicações. O esquema da figura 3.1 apresenta uma classificação dos sistemas de reconhecimento de caracteres, em relação aos itens citados acima. Este esquema é baseado em [MAN 86], com o acréscimo de mais alguns itens.

[Os diferentes sistemas de reconhecimento de caracteres podem ser classificados primeiramente quanto ao tipo de mecanismo utilizado na aquisição dos textos a serem reconhecidos. Nesta categoria, encontram-se os sistemas magnéticos (onde o texto é escrito com uma tinta magnética especial), os sistemas mecânicos e os sistemas óticos. Este último será o alvo dos estudos deste trabalho. Todos estes sistemas podem utilizar um processamento realizado a nível de sinais analógicos ou digitais. Os sistemas mais antigos, devido à tecnologia disponível, usavam mecanismos magnéticos, mecânicos ou óticos, baseados em dispositivos de tratamento de informações analógicas.]

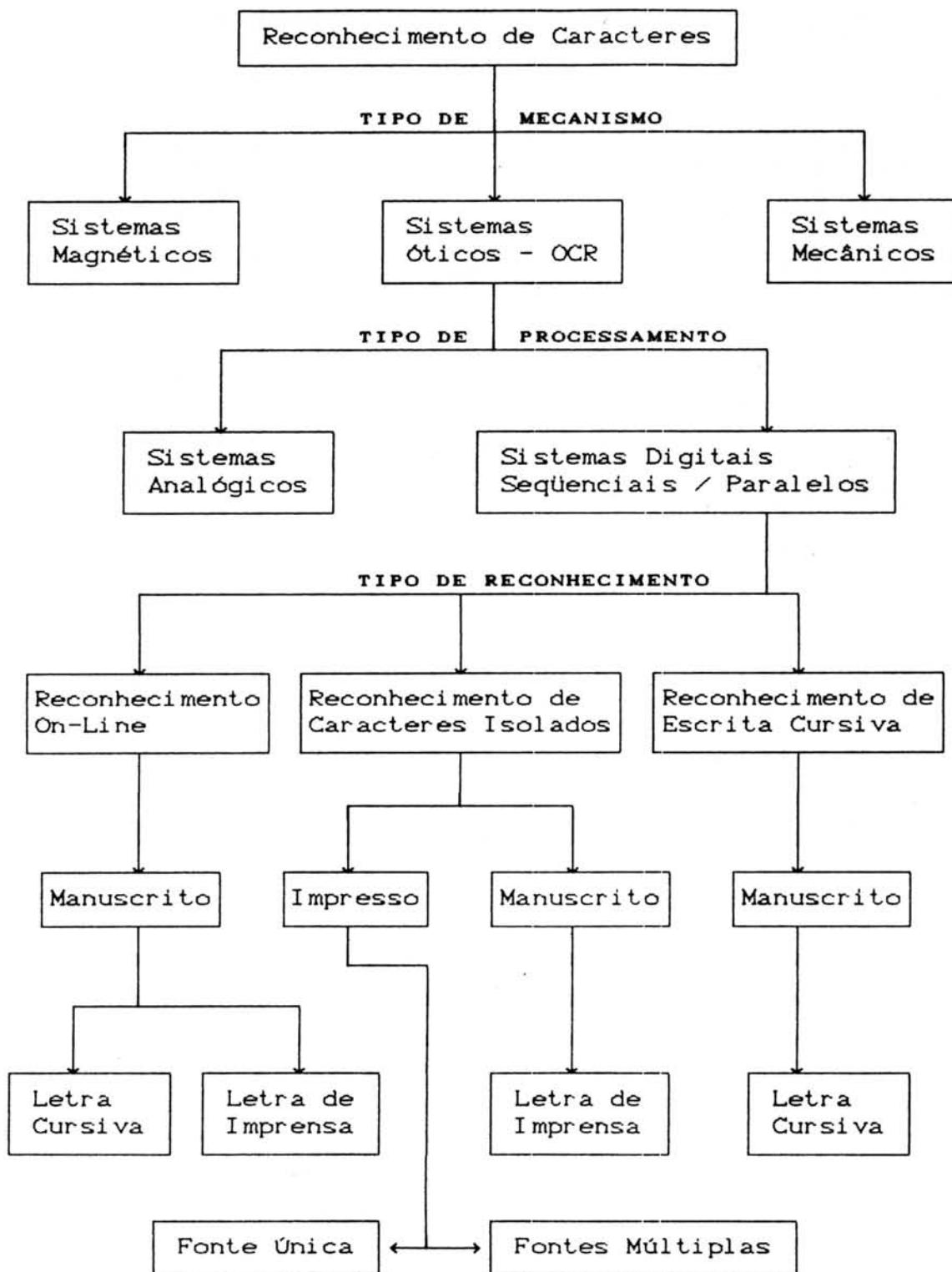


Figura 3.1 - Classificação dos sistemas de reconhecimento de caracteres

Um exemplo destes sistemas analógicos é descrito em [HAR 72], onde foi empregada uma técnica que trabalha com os campos de distribuição potencial. Nesta técnica, uma representação elétrica do caractere é colocada em uma matriz de resistores na qual um campo de potencial uniforme foi aplicado. A sobreposição da representação do padrão distorce o campo, de tal forma que a medida da distribuição potencial resultante nas extremidades da matriz, permite a obtenção de uma classificação única do padrão.

Os sistemas mecânicos, magnéticos e óticos que atuam sobre dados analógicos continuam a ser investigados até hoje, obtendo inclusive resultados muito promissores devido à possibilidade de exploração do paralelismo inherente a alguns destes processos. Infelizmente, tais sistemas são do tipo dedicado e exigem um custo muito alto em seu desenvolvimento.

Este estudo será restrito aos sistemas de OCR baseados no processamento digital, seja em sistemas sequenciais ou paralelos. Um dos motivos desta escolha é o fato de tais sistemas serem de uso geral, e desta forma seu custo pode ser reduzido em função da utilização do equipamento para outros fins.

Dentro dos sistemas OCR, são encontradas duas classes importantes, aqueles que são baseados no processamento sequencial das informações e os baseados no processamento paralelo. Este tipo de aspecto será abordado em maiores detalhes posteriormente. Entretanto, a principal distinção a ser feita em relação aos sistemas de OCR é em relação ao tipo de texto utilizado no reconhecimento. Existem três categorias principais: sistemas de reconhecimento on-line, sistemas de reconhecimento de caracteres isolados e sistemas de reconhecimento de escrita cursiva, onde estes dois últimos são realizados off-line.

Os sistemas de reconhecimento on-line, possuem características próprias que os distinguem dos demais, entre elas, a forma como os caracteres são entrados no sistema.

Nestes sistemas, utiliza-se usualmente uma mesa digitalizadora ou uma light-pen para fazer a aquisição unicamente de textos manuscritos. Devido à maneira como os caracteres são fornecidos, os dados disponíveis possuem características próprias. Estes dados são referentes a trajetória do traçado e não mais uma matriz de pontos, estando sujeito apenas a erros de precisão de coordenadas sem sofrer de problemas típicos da digitalização de imagens (inversão de pontos devido a interferências no processo de aquisição). Como resultado desta forma de entrada de dados, a descrição dos caracteres já possuirá uma representação "esqueletizada", apresentando apenas o contorno do caractere. Além da informação da forma do caractere que é fornecida por estes dispositivos de entrada, tais dispositivos também fornecem um dado extra referente à ordem do traçado (descrição temporal), podendo também contribuir para a classificação do mesmo.

Outra característica importante destes sistemas é o fato de ser inviável um treinamento prévio para uma grande quantidade de padrões exemplos, uma vez que o usuário tem que entrar manualmente com cada um destes padrões. Portanto os sistemas de OCR on-line devem normalmente já terem um conhecimento prévio embutido, não necessitando de uma fase de treinamento.

Os sistemas de reconhecimento on-line podem aceitar taxas de reconhecimento inferiores aos demais sistemas OCR. Isto se deve ao fato de tais sistemas serem interativos, permitindo um feedback de/para o usuário, de forma que ele se adapte, acostumando-se com as características e limitações do sistema, e desta forma incrementando posteriormente a taxa de reconhecimento. A taxa inicial de classificação correta de caracteres aceitável para sistemas on-line é de cerca de 95% de acertos. Este tipo de sistemas vem sofrendo um crescimento na sua utilização na atualidade e hoje já se encontram alguns equipamentos e aplicações nas quais o teclado foi totalmente substituído por entradas textuais realizadas através de um tablet [PCW 91].

Os sistemas de reconhecimento de caracteres isolados são os de maior utilização na atualidade, o que é devido provavelmente: à implementação mais simples, às taxas de reconhecimento correto mais altas e também devido à sua grande aplicabilidade. Este tipo de sistemas visa o reconhecimento de caracteres que estão representados no texto isoladamente, ou seja, textos impressos com letras não contíguas ou manuscritos com uso de letras de imprensa.

O último tipo de sistema OCR é o de reconhecimento de escrita cursiva, o qual é considerado o mais complexo dos três tipos de sistemas, em termos de dificuldade de implementação e de obtenção de taxas razoáveis de reconhecimento correto. Neste tipo de sistema, um dos maiores problemas (que não é tão grave nos outros) é o da separação dos caracteres, de forma a identificar cada um separadamente. Esta tarefa pode se tornar extremamente complexa e sujeita a muitos problemas. Um exemplo deste tipo de aplicação pode ser encontrado em [TAP 82], onde é aplicada uma técnica de "*elastic match*". Os sistemas de reconhecimento de caracteres escritos com letra cursiva não serão abordados neste trabalho, pois possuem características completamente diferentes dos outros sistemas, o que iria requerer todo um estudo específico para este caso.

Em relação a nomenclatura, costuma-se denominar de sistema OPCR (*Optical Printed Character Recognition*), os sistemas para reconhecimento de caracteres impressos, e, sistemas OHCR (*Optical Handwritten Character Recognition*), os sistemas para reconhecimento de caracteres manuscritos. No caso do termo OHCR, este é utilizado tanto para sistemas do tipo *handprint* (letra de imprensa), como para sistemas do tipo *handscript* ou *cursive script* (letra cursiva). Neste trabalho serão utilizados seguidamente os termos OPCR e OHCR, mas como não serão abordados os sistemas de reconhecimento de escrita cursiva, subentende-se que os sistemas OHCR referenciados são aqueles onde a escrita é do tipo *handprint* apenas.

Tem-se como exemplos de sistemas de reconhecimento de escrita do tipo OPCR, os sistemas descritos em [MAR 88, MAR 90b, SIL 90, WON 82, BRI 82 e HAR 72], e de sistemas OHCR, em [TAP 82, DAI 90, GON 90, SCH 78 e SUE 80].

Na classificação dos sistemas OCR, ainda podem ser levados em consideração mais alguns aspectos tais como:

1) Limitações quanto à abrangência do conjunto de classificação: São encontrados diferentes tipos de sistemas OPCR/OHCR, onde pode ser limitada a quantidade de símbolos (caracteres) diferentes a serem reconhecidos, criando assim reconhecedores para classes como: números, caracteres alfanuméricos (alfabeto latino, grego, katakana, etc), conjunto de símbolos completo, como por exemplo o utilizado na linguagem FORTRAN, ou, os caracteres imprimíveis dos códigos ASCII, BRASCI, EBCDIC, etc.

2) Limitações quanto ao número de estilos de escrita: Os sistemas OPCR podem ser limitados em relação ao número de fontes distintas de caracteres que o reconhecedor identifica simultaneamente. Pode-se ter reconhecedores apenas para um dos seguintes estilos: *sans-serif*, *pica*, *elite*, *itálico*, *negrito* (*boldface*); ou para um conjunto formado pela união de um grupo destes estilos. Neste tipo de classificação, é possível distinguir principalmente entre textos datilografados (com uma única fonte usualmente), ou textos estilizados impressos (com várias fontes usualmente).

3) Limitações quanto à forma de escrita: Esta limitação em sistemas OHCR equivale à limitação em termos de estilos para os sistemas OPCR. É muito comum se encontrar na literatura referências a sistemas de OHCR onde os usuários são previamente treinados para escrever de um certo modo, alterando a sua forma natural de escrita. Desta maneira, consegue-se restringir o número de estilos diferentes de escrita existentes entre os diversos candidatos ao uso do sistema. Uma outra limitação ao estilo é o uso de formulários, onde o usuário fica restrito ao preenchimento dos caracteres dentro de campos de tamanho e posições fixas.

As restrições quanto à forma do caractere, que podem ser impostas de forma a facilitar e melhorar a eficácia do reconhecimento nos sistemas OCR, têm sido muito aconselhadas, principalmente pelos órgãos responsáveis pela normatização de diferentes países: OCR Committe of ANSI, Japanese OCR Committe e OCR Committe of Canada (figura 3.2). Baseado neste tipo de restrições, foram definidos diversos padrões de definição de conjuntos de caracteres, tanto para sistemas OPCR (OCR-A - Americano, OCR-B - Europeu), como para sistemas de OHCR (Conjunto de caracteres para escrita manual ANSI 1974) [SUE 80].

0 1 2 3 4 5 6 7 8 9

A B C D E F G H I J

K L M N Ø P Q R S T

U V W X Y Z

(A) Conjunto Padrão de Caracteres Manuscritos - OCR ANSI

0 1 2 3 4 5 6 7 8 9

A B C D E F G H I

J K L M N Ø P Q R

S T U V W X Y Z

(B) Conjunto Padrão de Caracteres Manuscritos - OCR Japão

0123456789

(C) Conjunto Padrão de Numerais para Sistemas OCR - Canadá

Figura 3.2 - Exemplos de padronização de conjuntos de caracteres para sistemas de OCR

O sistema N²OCR, que é a base do estudo deste trabalho, está orientado principalmente ao reconhecimento de caracteres impressos isolados, onde pode ser considerada a utilização de múltiplas fontes de texto. Os conceitos apresentados em relação ao sistema N²OCR podem ser facilmente estendidos para sistemas OHCR (orientados também ao reconhecimento de caracteres isolados), independentemente da entrada ser *on-line* ou *off-line*. Esta possibilidade de extensão do sistema será abordada no penúltimo capítulo. O único tipo de reconhecimento que não será abordado é o de escrita cursiva.

3.3 Descrição dos Sistemas OCR

Existem diferentes etapas de processamento das informações visando o reconhecimento de textos por um sistema de OCR. Estas etapas se estendem desde a captura da imagem do texto até a obtenção final de uma identificação deste. Em um sistema de OCR não se deve considerar apenas a parte referente às técnicas empregadas na resolução do problema de classificação, mas um item de extrema importância para sistemas de reconhecimento de caracteres é também a medição e avaliação dos resultados obtidos por estes.

3.3.1 Etapas de Processamento

Em geral, os sistemas de OCR possuem implementadas as seguintes funções [SUE 80]: aquisição da imagem do texto, tratamento da imagem, localização e separação dos caracteres, pré-processamento dos padrões, extração de atributos, reconhecimento/classificação dos padrões e pós-processamento. A figura 3.3 apresenta um esquema representando as etapas citadas acima.

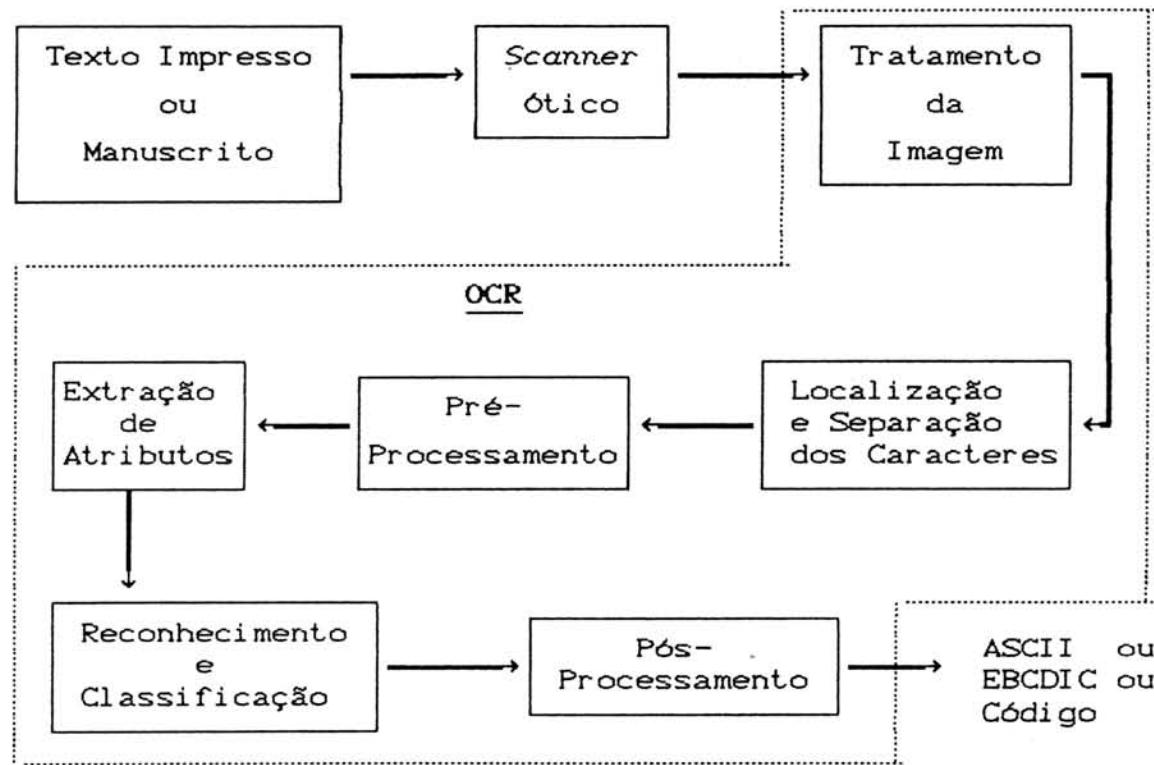


Figura 3.3 - Esquema das etapas de processamento de um sistema OCR

As etapas de tratamento da imagem, separação dos caracteres (segmentação), pré-processamento e extração de atributos, representadas na figura 3.3, são denominadas por alguns autores simplesmente de pré-processamento. Neste capítulo serão abordados os itens relacionados a partir da separação dos caracteres, visto que o capítulo 2 já abordou os itens referentes à aquisição e tratamento das imagens.

A separação dos caracteres, ou segmentação, é uma importante etapa do trabalho de reconhecimento, envolvendo a separação individual dos caracteres do resto do conteúdo da imagem digitalizada. Numa primeira etapa da segmentação torna-se importante realizar a diferenciação e separação entre imagens gráficas (desenhos, fotografias, etc) das imagens textuais. Após, as áreas que foram identificadas como sendo áreas cujo conteúdo são textos sofrem um processamento visando isolar cada caractere em uma pequena matriz de pontos. Somente a partir do momento em que for obtida uma matriz representando apenas um caractere é que este poderá ser então processado visando a sua classificação.

Para a separação dos caracteres, o algoritmo tem que levar em consideração diversos fatores, entre eles: o espaçamento entre caracteres e tamanho do caractere (fixo ou variável), a utilização de uma grade auxiliar (no caso de formulários com espaçamento bem definido), e o tipo de texto a ser reconhecido, onde podem haver caracteres que se tocam ou caracteres com descontinuidades (características muito importantes para o módulo de segmentação). Esta etapa de processamento é até hoje uma das etapas mais críticas, sendo um problema ainda não completamente solucionado [CAS 82]. Portanto, a etapa de segmentação merece uma destacada importância dentro do processo global de reconhecimento. O pré-processamento realizado após a segmentação dos caracteres consiste basicamente em realizar alguma correção em termos de distorções do caractere. O principal objetivo é normalmente o de ajustar o tamanho da matriz resultante da segmentação dos caracteres, de forma que esta tenha um tamanho compatível com o tipo de "padrões exemplos" das classes a serem identificadas. Logo, o pré-processamento envolve em primeiro lugar o ajuste de escala (normalização de tamanho), podendo incluir outros procedimentos como: correção de distorções, ajuste de posição, correção de inclinações e rotações, e até mesmo um segundo nível de tratamento de imagens. A diferença do pré-processamento para o tratamento de imagens é que neste o tratamento é realizado a nível de um único padrão e não mais sobre a imagem inteira.]

Após terem sido isolados os caracteres, e preferencialmente com a realização de um ajuste de tamanho e posição pré-definidos, pode-se então realizar uma etapa de extração de atributos. Esta etapa é opcional e dependerá exclusivamente do tipo de técnica empregada no reconhecimento e classificação dos padrões. A recodificação de um padrão através de seus atributos visa ressaltar algumas características dos padrões, de forma a facilitar a sua identificação. Por exemplo, pode-se processar um padrão correspondente a um caractere, de forma a extrair segmentos de retas e curvas presentes neste. Usando esta descrição do caractere através deste novo conjunto de atributos, pode ser implementado um algoritmo de classificação cuja função de decisão leve em consideração a presença, ausência ou inter-relações entre estes novos atributos.

O reconhecimento consiste na aplicação de uma técnica para reconhecimento de padrões (como as citadas no item 2.4) do qual resulta uma classificação deste em uma das classes pré-definidas. O reconhecimento dependerá muito da técnica empregada e dos atributos levados em consideração, sendo que no próximo item serão analisadas as principais técnicas empregadas no caso específico do reconhecimento de padrões, onde estes padrões são caracteres.

Uma etapa final a ser realizada no reconhecimento de caracteres é o emprego de técnicas de validação do texto reconhecido. Nesta etapa, realiza-se um pós-processamento, baseado no contexto, visando corrigir alguns erros realizados durante a fase de reconhecimento. Este assunto será abordado posteriormente.

Todas as etapas citadas acima, compreendem um sistema de OCR o mais completo possível, sendo que, dependendo do tipo de sistema de OCR, algumas etapas podem ser omitidas ou podem sofrer algumas modificações.

3.3.2 Técnicas de Reconhecimento

Existem diferentes técnicas empregadas no reconhecimento de padrões, mas estas podem ser classificadas em três grandes categorias, técnicas baseadas em: atributos globais, distribuição de pontos e atributos geométricos e topológicos [SUE 80].

3.3.2.1 Atributos Globais

Nesta categoria, os atributos (*features*) são extraídos de cada ponto interior ao retângulo que circunscreve a matriz do caractere, a qual também pode ser denominada de *frame*. Os atributos empregados não refletem nenhuma propriedade local, geométrica ou topológica do próprio traçado. Algumas destas técnicas foram desenvolvidas originalmente para reconhecer apenas caracteres impressos.

3.3.2.1.1 Casamento de Padrões e Correlações

Esta técnica toma os atributos do estado (preto ou branco) de todos os pontos pertencentes ao *frame*. Ela mede simplesmente a similaridade entre o caractere de entrada e as referências armazenadas, pelo casamento e obtenção da correlação de pontos ou grupos de pontos do *frame*. Exemplos de emprego desta técnica são encontrados em [MAR 90a e KOZ 71].

3.3.2.1.2 Transformações e Expansões em Séries

Um problema vindo do uso dos próprios pontos como atributos é a alta dimensão do vetor de atributos usado no casamento de padrões. Para reduzir esta dimensão e extrair atributos invariantes a deformações globais (como por exemplo, rotação e translação), faz-se transformações ou expansões em séries. Transformações como de Fourier, Walsh, Haar e Hadamard, bem como expansões em séries de Karhunen-Loeve tem sido exploradas. No capítulo 2, já foi abordado um exemplo de uso da transformada de Fourier para tratamento de padrões/sinais.

3.3.2.2 Distribuição de Pontos

É uma outra forma de reduzir a dimensionalidade dos conjuntos de atributos, onde estes são derivados a partir das distribuições estatísticas dos pontos. Diferentes tipos de distribuições tem sido utilizadas, correspondendo a diferentes técnicas de reconhecimento.

3.3.2.2.1 Zoneamento

O frame que contém o caractere é subdividido em várias zonas sobrepostas ou não, e a densidade de pontos nestas diferentes regiões formarão os atributos. Este método é usado comumente em reconhecimento de caracteres manuscritos em letra de imprensa. Este tipo de procedimento é abordado em [SIL 90 e HUS 72].

3.3.2.2.2 Momentos

Os momentos dos pontos do caractere em relação a um centro escolhido, por exemplo, o centro de gravidade do caractere, ou de um sistema de coordenadas escolhido, são usados como atributos.

3.3.2.2.3 N-Uplas

A ocorrência de elementos pretos ou brancos ou a ocorrência da junção destes pontos são usadas como atributos. Um exemplo desta técnica é encontrado em [ULL 69].

3.3.2.2.4 Characteristic Loci

Para cada ponto pertencente ao interior do caractere, geram-se vetores horizontais e verticais. O número de vezes (um máximo de dois é usado na ordem, para limitar a dimensão do atributo) que um segmento de linha é intersecionado por estes vetores é usado como atributo. Um exemplo do emprego desta técnica é encontrado em [SIL 90].

3.3.2.5 Cruzamentos e Distâncias

Atributos são obtidos medindo-se o número de vezes que segmentos de linha são atravessados por estes vetores em direções específicas ou medindo-se as distâncias dos elementos ou segmentos de linha em relação a uma dada borda, tal como as bordas do *frame* que contém o caractere.

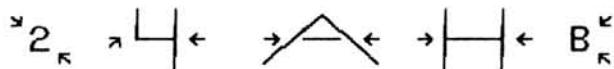
3.3.2.3 Atributos Geométricos e Topológicos

Esta técnica é baseada na extração de atributos que descrevem a geometria ou topologia de interesse no traçado do caractere. Estes atributos podem representar propriedades locais ou globais do caractere. Esta é de longe a mais popular técnica estudada pelos pesquisadores, e alguns exemplos de atributos extraídos são descritos abaixo:

1) Retas e Curvas em Várias Direções: É realizada a identificação de elementos como

-, |, \, /, <, >, U, ∩

2) Pontos Finais, Intersecção de Segmentos de Reta e Laços: É realizada a identificação de elementos como



Todas as técnicas acima visam facilitar e melhorar a eficácia do reconhecimento de caracteres, através de uma melhor seleção dos atributos que representam este. Estas técnicas listadas acima servem para compor o vetor de atributos de um padrão, podendo ser a própria matriz de pontos do caractere, ou uma nova codificação obtida a partir desta, ou ainda podendo ser resultante da composição de dois ou mais conjuntos de atributos diferentes.

Estes variados métodos influenciarão diretamente a capacidade do sistema em ser imune a problemas de deformação, ruído e outras perturbações. Este aspecto será analisado no item seguinte.

3.3.3 Avaliação do Reconhecimento

A comparação do desempenho entre os diversos sistemas de OCR é uma tarefa muito difícil e provavelmente os resultados obtidos nesta comparação não sejam muito confiáveis. Para cada algoritmo candidato a avaliação, sempre haverá provavelmente um conjunto de padrões de teste para o qual a avaliação será superior a de um outro algoritmo rival [NAG 83]. Baseado nestas constatações, foi desenvolvida uma série de métodos e parâmetros a serem utilizados, visando no mínimo uma melhor caracterização dos algoritmos e sistemas de OCR em geral.

Uma das formas de realizar a avaliação destes sistemas foi através da criação de bancos de dados padrões para análise de desempenho, contendo amostras variadas de padrões a serem reconhecidos por um novo algoritmo submetido à avaliação. Alguns destes bancos de amostras de padrões de caracteres se tornaram muito difundidos e utilizados na avaliação de diferentes sistemas. Dois destes bancos de dados muito conhecidos são: o de Highleyman e o de Munson. Estes bancos de dados estão disponíveis junto ao IEEE e fornecem um conjunto de caracteres manuscritos em letra de imprensa. Os dados de Munson possuem 12760 amostras de 46 tipos diferentes de caracteres, com uma resolução de 24 x 24 pixels, e os dados de Highleyman possuem 1800 amostras de letras e 500 de numerais, com um total de 36 tipos diferentes de caracteres, com uma resolução de 12 x 12 pixels. Esta resolução (12x12) infelizmente não é a mais adequada, uma vez que para o reconhecimento de caracteres manuscritos é aconselhada uma matriz com dimensões em torno de 20 x 25 pixels [SUE 80]. O maior problema referente a tais bancos de dados é a divulgação de sua existência e as formas de acesso aos mesmos. Infelizmente, para a realização deste trabalho não foi possível a utilização destas bases de dados de padrões de caracteres, devido primeiramente ao fato de serem orientados a sistemas de reconhecimento de caracteres manuscritos e, em segundo, porque não foi possível até o momento ter acesso a eles.

Espera-se, num futuro próximo, que seja possível o acesso a estas e a outras bases de dados de testes, visando uma melhor validação do trabalho desenvolvido.

Os principais parâmetros a serem tomados como referência são os índices de avaliação de desempenho de sistemas de OCR. Estes índices consistem de três indicadores: taxa de caracteres reconhecidos corretamente, taxa de caracteres substituídos (classificações incorretas) e taxa de rejeição (caracteres não identificados) [MAR 90c]. Estas taxas são as mesmas que já haviam sido relacionadas no item 2.4 referente à avaliação de sistemas de reconhecimento de padrões.

O objetivo deste texto não é o de estabelecer valores fixos a serem atingidos por estas taxas de avaliação de desempenho. Isto não é aconselhável, pois estes índices de desempenho dependem de uma série de fatores próprios de cada aplicação. O que será feito é uma apresentação com o objetivo de orientar o leitor em termos de parâmetros e itens a serem considerados, de forma a se ter uma noção do desempenho esperado para um dado sistema alvo.

Quando se trabalha com sistemas OCR, é importante se considerar as taxas referentes ao seu desempenho para que seja possível determinar a viabilidade, em termos econômicos, da aplicação prática do sistema e em relação ao objetivo de uso destes sistemas como forma alternativa de aquisição de informações pelo computador. Para que seja possível fazer este tipo de análise é importante se tomar conhecimento de alguns dados de referência muito importantes. O primeiro se refere a capacidade do ser humano em reconhecer caracteres, onde a taxa de erros na identificação dos caracteres, na ausência de contexto, é de 4% aproximadamente. Desta forma, um sistema OCR, que não se utiliza de um pós-processamento para avaliação de contexto, onde é buscado um desempenho similar ao humano, deve atingir taxas de 96% de reconhecimento correto de caracteres. Obviamente que taxas superiores a esta devem ser sempre buscadas.

Um outro dado importante é a taxa de substituição de caracteres em textos datilografados que, antes de sofrerem revisão, se situa em torno de 0.1%, para dados que possuem um certo significado contextual [MAR 90c]. Esta última taxa indica que sistemas OCR com uma eficácia igual ou superior a esta são plenamente indicados para substituir o ser humano em tarefas de transcrição de dados por digitação.

Mas então quais são as taxas aconselháveis de desempenho para um sistema OCR? É muito difícil de estimar, e inclusive são encontradas contradições entre diferentes autores. Segundo [MAR 90c], os sistemas comerciais possuem tradicionalmente taxas de rejeição mais substituição em torno de 10:1 ou 100:1. Como pode ser visto, esta faixa é muito ampla, e isto é devido ao fato de que se deve levar em conta o tipo de aplicação do sistema OCR. Um sistema de OPCR terá uma certa taxa de reconhecimento aceitável, e um sistema de OHCR terá outra, dependendo ainda do tipo de dados originais.

A seguir, ver-se-á um exemplo de avaliação a ser seguido. Uma vez determinado o tipo de aplicação, como por exemplo, um sistema de entrada automática de dados para o computador através da aquisição de textos impressos; passa-se para a análise das taxas de erros e rejeições admissíveis. Neste sistema exemplo, uma taxa de substituições e rejeições em torno de 10:1 é inaceitável, pois em cada uma ou duas palavras de tamanho médio, haverá uma letra incorreta e o trabalho de correção será muito demorado. Neste caso seria preferível digitar todo o texto diretamente no computador.

Para dar uma noção das taxas de desempenho, foi reproduzida uma tabela de valores destas representando diferentes tipos de sistemas analisados em [SUE 80]. Esta tabela apresenta o desempenho de algoritmos referenciados na literatura, onde para cada caso estão listados um exemplo de baixa, mediana e alta eficácia. Todos os algoritmos listados na tabela 3.1 são referentes a sistemas OHCR.

Tabela 3.1 - Taxas de reconhecimento obtidas por diferentes sistemas OCR

Tipo de Reconhecimento	Autor do Algoritmo	Desempenho		
		TA	TE	TR
Numerais	Neill, 1969 Spantersberg, 1974 Masterson, 1962	85 % 92,80% 99,79%	2,7 %	4,5%
Alfanumérico	Dydyk, 1970 Naylor, 1971 Krause, 1973	65 % 80,40% 97,10%	2,9 %	0 %
Símbolos	Caskey, 1973	91,80%		
Fortran	Mori, 1975 Fujimoto, 1976	96 % 99,74%	0,5 % 0,06%	3,5% 0,2%
Caracteres	Masuda, 1972	94,4 %	0,2 %	5,4%
Katakana	Tomita, 1967 NEC Lab., 1974 ETL Group, 1976	98,1 % 98,4 % 98 %	0,6 % 0,2 %	1 % 1,8%

Obs.: TA - Taxa de Acertos TE - Taxa de Erros
 TR - Taxa de Rejeições

É interessante salientar que, entre os algoritmos relacionados na tabela 3.1, todos aqueles que obtiveram o melhor desempenho em seu respectivo grupo (Masterson, Krause, Fujimoto) fizeram uso de restrições na escrita dos caracteres. Foram utilizados nestes testes um conjunto de caracteres OCR padronizado, com o treinamento prévio dos usuários para escrever da forma solicitada! Isto demonstra como uma tabela comparativa pode ocultar certos detalhes muito importantes. Outros comparativos podem ser também encontrados em [NAG 82].

Além das taxas de avaliação de desempenho, descritas anteriormente, existem outros fatores muito importantes:

- Sensibilidade à deformação da imagem (será abordada no item 3.3.4);
- Implementação prática da técnica;
- Custo do sistema e

- Características da aplicação: tipo de caracteres, restrições em relação aos documentos digitalizados, volume de dados envolvidos e qualidade do texto ou do meio utilizado.

Quanto à implementação prática do sistema, alguns fatores importantes são: a possibilidade de obtenção automática de máscaras para casamento de padrões, a velocidade de reconhecimento em termos do número de caracteres reconhecidos por segundo, a complexidade e viabilidade da implementação prática do sistema e a independência da técnica em relação à utilização de técnicas suplementares. Outro aspecto muito importante em relação à implementação, é o grau de paralelismo empregado no sistema, o qual influí drasticamente na velocidade de reconhecimento do sistema.

Em relação ao custo de sistemas OCR comerciais, é possível encontrá-los com preços que variam de US\$100 (sistemas usados junto a *handy scanners*, com custo aproximado de US\$250) até sistemas de US\$4000 (sistemas acoplados a leitores automáticos de páginas, com custo aproximado de US\$1500). As velocidade de reconhecimento destes sistemas variam de 0,1 a 2,0 caracteres por segundo, operando com um conjunto de caracteres para reconhecimento de 50 a 3.000 símbolos diferentes [MAR 90c]. Existem também sistemas dedicados de alto desempenho que possuem taxas de até 40.000 caracteres reconhecidos por segundo e com custos bem mais elevados, em torno de US\$1.000.000 [HAR 72].

3.3.4 Principais Dificuldades

Existem diversos elementos que dificultam a tarefa de reconhecimento de caracteres, influenciando diretamente no desempenho dos sistemas OCR. Os principais problemas enfrentados no reconhecimento de caracteres são:

- Deformações dos caracteres a serem reconhecidos;
- Problemas de segmentação e separação dos caracteres;
- Abrangência e tipo dos caracteres a reconhecer;
- Desempenho do sistema independente da quantidade e qualidade da massa de dados manipulada e
- Capacidade de adaptação a novos caracteres.

As deformações possíveis de ocorrer em caracteres são as seguintes: ruído, distorção, translação, rotação, mudança de escala e variação de estilo. O ruído consiste da inversão de pontos do padrão da imagem, ocorrendo normalmente na etapa de digitalização (possivelmente associado ao dispositivo de aquisição), gerando linhas desconectadas, espaçamentos inexistentes, preenchimento de espaços, união de caracteres e surgimento de pontos esparsos. Um outro tipo de "ruído" muito comum é o proveniente do meio onde está escrito o texto a ser digitalizado. Isto pode ser facilmente reparado em reproduções de um original através de uma fotocopiadora. Quanto maior o número de reproduções do original, quanto mais for manipulado este material, e quanto pior for a qualidade do papel, mais a imagem digital capturada estará sujeita a conter *spots* (pontos de sujeira) no texto, que dificultarão o processo de reconhecimento.

As distorções são variações locais, arredondamento de bordas, dilatação ou compressão do padrão, torções, e todo o tipo de deformação resultante de uma perturbação do processo de digitalização. Estas distorções ocorrem muito em dispositivos tipo *handy scanner*, onde o usuário é o responsável pelo deslocamento do dispositivo de varredura. Pode ocorrer que ao realizar esta tarefa, o usuário faça pequenos deslocamentos indevidos na direção de varredura, causando assim o surgimento de distorções. Outra causa de distorções pode ser a digitalização de uma folha com dobras ou ondulações ou ainda a ocorrência de uma variação indevida da velocidade de varredura para aquisição da imagem.

Ao fazer a aquisição do texto, é difícil a obtenção de um referencial extremamente preciso, de forma a que o texto seja posicionado exatamente no ponto desejado e com a orientação de forma que as linhas do texto estejam colocadas perfeitamente na horizontal em relação ao dispositivo de aquisição. Devido aos problemas de posicionamento e orientação, os caracteres muitas vezes se encontrarão levemente rotacionados ou transladados, dificultando a sua identificação.

Além da translação e rotação, os caracteres estão sujeitos a problemas de distorções que podem provocar uma alteração no tamanho destes. A alteração de tamanho também pode ser resultante do afastamento do dispositivo de aquisição das imagens em relação ao objeto alvo, resultando uma redução ou ampliação dos padrões contidos na cena, e consequentemente, necessitando um ajuste de escala. Em certos casos, como no casamento de padrões onde são levados em consideração os próprios pontos do padrão, este tipo de perturbação terá que ser corrigido.

A variação de estilo pode ser considerada como uma distorção, pois causa a deformação do padrão, que assume uma nova forma, diferente daquela para a qual o sistema foi treinado. Quando ocorrem variações de estilo em um mesmo texto, o sistema de reconhecimento tem que ser capaz de identificar diferentes representações para um mesmo caractere.

Cada uma das técnicas apresentadas no item 3.3.2 terá um comportamento diferente diante da presença das várias deformações possíveis dos caracteres. A tabela 3.2 apresenta um comparativo destas técnicas, em relação ao tipo de distorção que pode ser enfrentado [SUE 80].

Tabela 3.2 - Comportamento das técnicas de reconhecimento

CRITÉRIO →	Tolerância a deformação da imagem				
	Ruído	Distorção	Variação de Estilo	Translação	Rotação
MÉTODOS ↓					
1. Atributos Globais					
Casamento de Padrões	+	□	-	-	-
Transform.	-	+	+	+	+
2. Distrib. de Pontos					
Zoneamento	-	□	-	-	□
Momentos	□	□	-	+	+
N-Uplas	□	-	□	-	□
Characteris. Loci	-	+	+	+	□
Cruzamento e distância	-	+	+	+	□
3. Atributos Geométric. e Topológ.	-	+	+	+	□

+ Alta Tolerância □ Média - Baixa Tolerância

Os problemas de segmentação típicos são a separação de textos e gráficos e o problema dos caracteres que se tocam - *touching characters*. A separação de textos e gráficos pode ser resolvida de uma maneira eficiente, como será demonstrado no item 3.5.1.1, o que não ocorre com os caracteres que se tocam. A união de dois caracteres pode ser causada por deformações dos caracteres ou inclusive ser resultante de características inerentes ao estilo da letra. Esta união é bastante problemática, sendo inclusive um dos motivos porque a segmentação de caracteres ainda é um problema não completamente resolvido até hoje. Uma solução adotada em relação ao problema dos *touching characters* consiste em simplesmente considerar o par de caracteres unidos, como uma classe a mais a ser reconhecida, e desta forma contornar o problema, sem na realidade resolvê-lo.

A abrangência e o tipo de caracteres a serem reconhecidos são pontos que podem aumentar bastante a complexidade de um sistema OCR. Um sistema de reconhecimento pode ser orientado apenas para o reconhecimento de numerais, o que facilita muito o processamento, uma vez que o conjunto de classes é bastante reduzido. Imagine-se agora um sistema para o reconhecimento de múltiplas fontes, para todos os caracteres ASCII imprimíveis... O número de classes é extremamente grande, possibilitando confusões entre uma classe e outra e, como se isto não bastasse, ainda há o problema das múltiplas fontes implicando vários padrões de representação para uma mesma classe. A figura 3.4 apresenta uma classificação em termos de complexidade dos sistemas OCR, em relação à abrangência do reconhecimento realizado por estes sistemas.

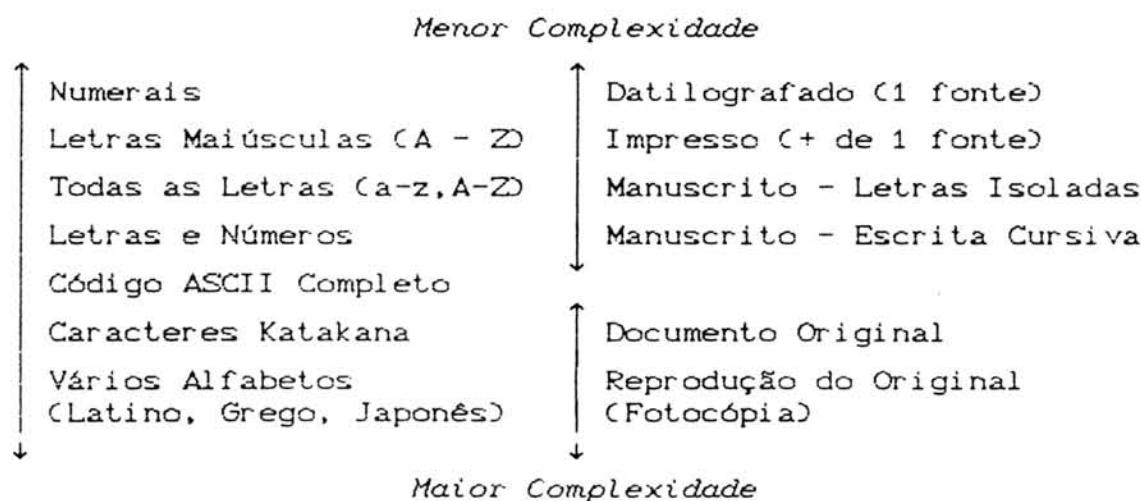


Figura 3.4 - Complexidade dos sistemas OCR

Em relação ao número de caracteres diferentes a serem reconhecidos, fica fácil demonstrar os problemas resultantes deste aumento, bastando listar as substituições de caracteres mais comuns em sistemas de OCR. Nota-se que só o fato de trabalhar apenas com letras ou apenas com números já possibilita uma grande redução neste tipo de erros.

Mesmo em sistemas restritos a conjuntos menores, certos erros comuns podem ocorrer, como é o caso da confusão entre letras minúsculas e maiúsculas (s, S; c, C; u, U; etc), e outros erros como os apresentados na tabela 3.3, onde a melhor solução para uma classificação correta é o emprego de correções baseadas no contexto.

Tabela 3.3 - Erros de substituição mais comuns

Troca	Motivo da confusão / O que fazer
4 ↔ A	4 é similar o A com uma falha na base Solução : Escrita do 4 com topo aberto
2 ↔ Z	Podem possuir a mesma forma Solução : Escrita do Z com um risco central
1 ↔ I	Diferença muito pequena apenas em cima Solução : Escrever 1 sem risco em baixo e o I com o uso de 2 riscos nas extremidades
0 ↔ O	Não há nenhuma diferença (apenas o contexto) Solução : Escrever o zero cortado 0
7 ↔ Z	O 7 e Z só se diferenciam pelo risco inferior Solução : Escrever o 7 sem risco central e o Z com o uso deste risco
Outros Erros Comuns	2 ↔ 7 ↔ Z ; U ↔ V ; C ↔ L ; a ↔ d n ↔ h ; L ↔ I ↔ 1 ; S ↔ 5 ; G ↔ 6
SOLUCAO GERAL	ACONSELHA-SE CONSIDERAR O CONTEXTO

Quanto ao desempenho dos sistemas de OCR, este é um problema difícil de ser resolvido, pois para um melhor reconhecimento há um grande acréscimo no número de informações manipuladas. No caso de sistemas on-line, existe uma restrição em relação ao tempo de processamento um pouco mais rígida, uma vez que não deve ser prejudicada a "taxa natural" de entrada de caracteres. Em sistemas off-line esta taxa de aquisição de dados pode ser controlada, mas mesmo assim o sistema deve ter um desempenho razoável, de forma a satisfazer o usuário no tipo de aplicação para o qual está voltado. Uma das formas de acelerar o processamento é através da execução de tarefas em paralelo, otimizando o processamento.

Este trabalho apresenta uma proposta que permite a aceleração do reconhecimento através do emprego de paralelismo. Este assunto será abordado em maiores detalhes nos capítulos 5 e 6.

Por fim, uma última dificuldade dos sistemas de reconhecimento de caracteres é a implementação de uma capacidade de adaptação ou aprendizado. Conforme foi apresentado, existe um conjunto enorme de tipos e estilos de caracteres e, mesmo se fosse feita uma restrição apenas a caracteres impressos, ainda assim o número de fontes para impressão seria muito grande. Uma solução para este problema, que gera uma dificuldade a mais, é a criação de um sistema com capacidade de adaptação, onde cada usuário realiza um treinamento prévio específico para o tipo de caracteres que ele deseja reconhecer. Esta fase inicial de adaptação permite que o sistema aprenda a reconhecer melhor os caracteres que lhe serão posteriormente apresentados, obtendo assim taxas melhores de reconhecimento. Um sistema adaptativo também fornece uma maior flexibilidade ao OCR, uma vez que este não precisa mais ser desenvolvido para o reconhecimento de um conjunto específico de caracteres. Este conjunto, que deveria ser o mais amplo possível para que o sistema tivesse um bom desempenho de reconhecimento, aumenta muito a complexidade do sistema, o tempo de resposta e o número de erros. Sistemas adaptativos podem aprender um conjunto mais restrito de caracteres a serem reconhecidos, específicos para cada aplicação, e desta forma possibilitam ainda uma redução quanto ao número de erros que podem ocorrer.

Uma vez que as características e dificuldades de implementação de sistemas OCR já foram apresentadas, serão abordados agora os algoritmos desenvolvidos para solucionar os principais problemas referentes ao assunto, bem como as técnicas necessárias para que o objetivo final de reconhecer caracteres seja alcançado.

3.4 Análise de Algoritmos

Este item irá resumir as técnicas consideradas de maior importância, na implementação de um sistema OCR, e mais especificamente aquelas que foram utilizadas no desenvolvimento do sistema N²OCR.

3.4.1 Segmentação

Em relação à segmentação, serão abordados dois aspectos, separação entre textos e gráficos, e separação de caracteres entre si.

3.4.1.1 Separação de Textos e Gráficos

Um documento impresso normalmente é constituído de duas classes bem distintas de informações: os gráficos e os textos. Para realizar o reconhecimento de caracteres é importante que o sistema de OCR tenha a capacidade de distinguir entre as regiões de texto e as regiões de gráficos. O método que será apresentado é o *Run-Length Smoothing*, o qual permite a realização da tarefa descrita acima [WON 82].

Os documentos digitalizados, utilizados junto a esta técnica, são imagens do tipo B&W com figuras, desenhos e fotografias. As fotografias são representadas através de halftoning. O procedimento empregado na separação consiste basicamente de dois passos: o primeiro é um procedimento que subdivide a área global do documento em regiões (blocos), cada uma delas podendo conter apenas um tipo de dados (texto, figuras, fotografias, etc). A seguir, alguns atributos básicos destes blocos são calculados, permitindo que seja realizada uma classificação destes. Cada bloco será classificado de acordo com uma avaliação a respeito de dados estatísticos sobre seus atributos.

O algoritmo básico utilizado, o RLSA - *Run Length Smoothing Algorithm*, tem como objetivo original a detecção de longas linhas horizontais e verticais. Este algoritmo foi estendido para obter um *bit-map* com áreas bem definidas, representando os blocos que contêm os diversos tipos de dados. O RLSA básico é aplicado sobre uma seqüência binária, na qual pixels brancos são representados por 0's e pixels pretos são representados por 1's. O algoritmo transforma uma seqüência binária X em uma seqüência de saída Y de acordo com as seguintes regras:

- 0's em X são trocados para 1's em Y, se o número de 0's adjacentes for menor ou igual a um limiar C pré-definido.
- 1's em X são mantidos inalterados em Y.

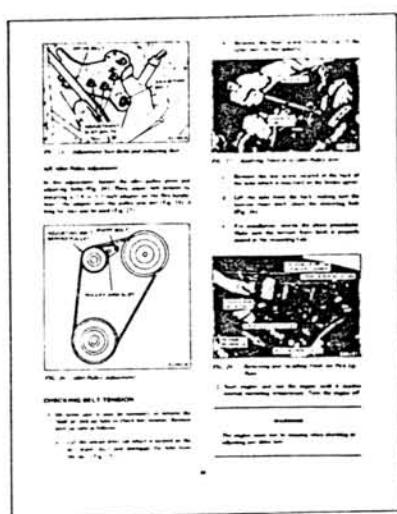
Por exemplo, com C=4, a seqüência X é transformada em uma seqüência Y como segue:

X: 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0
Y: 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1

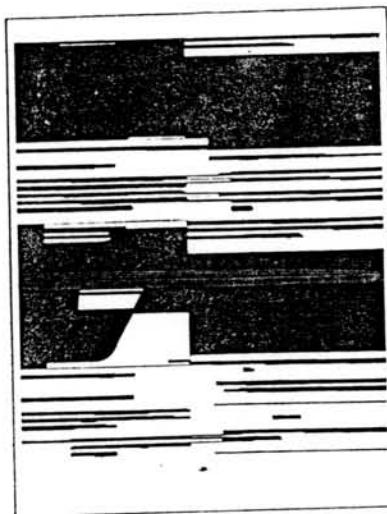
Quando o RLSA é aplicado sobre matrizes de pontos, ele possui o efeito de juntar as áreas pretas, que são separadas por menos que C pixels. A escolha apropriada de C faz com que sejam unidos todos os pontos de regiões de um tipo de dados único. O grau de união dos pontos dependerá do valor de C, da distribuição dos pontos pretos e brancos no documento (característica para cada tipo de dado) e da resolução de digitalização. O valor de C pode ser estimado experimentalmente, de acordo com as características de resolução e do tipo de dados presentes nos documentos.

O RLSA é aplicado linha a linha, coluna a coluna em um documento, obtendo dois distintos *bit-maps*. Por causa do fato de se ter uma variação independente do espaçamento horizontal e vertical dos diferentes componentes presentes nos documentos, dois valores diferentes de C podem ser usados (C_H e C_V).

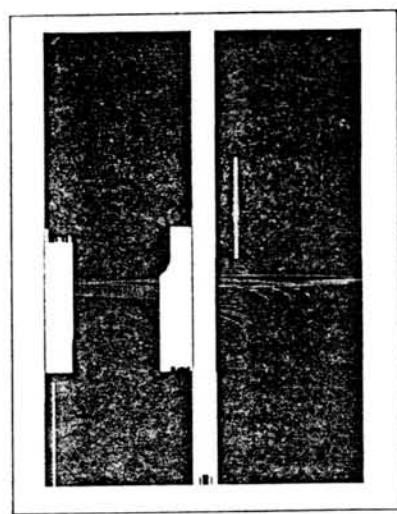
A figura 3.5 [WON 82] apresenta um documento (a), e o resultado da aplicação do RLSA na direção horizontal (b) e na vertical (c). Os dois bit-maps são combinados através de uma operação lógica AND, realizada ponto a ponto, obtendo a imagem representada na figura 3.5 (d).



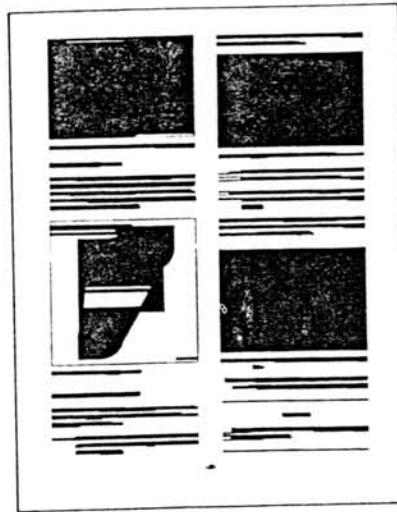
(a)



(b)



(c)



(d)

Figura 3.5 - Etapas do processamento do algoritmo RLSA

Os blocos apresentados na figura 3.5 (d) podem ser localizados e classificados de acordo com o seu conteúdo. Os blocos são isolados através de um algoritmo de identificação de regiões contínuas de pontos pretos, atuando de forma similar a do algoritmo que será descrito na seção 3.5.1.2. Para que seja possível a identificação realizada no passo seguinte, é necessário obter as seguintes medidas para cada um dos blocos:

- Número total de pontos pretos no bloco segmentado;
- Número total de pontos pretos da imagem original referentes aos pontos contidos no bloco obtido;
- Tamanho do bloco, comprimento na horizontal e vertical e
- Transições de pontos brancos para pretos da imagem original para a resultante do processamento.

De posse destes dados, pode-se então caracterizar o tipo de conteúdo de cada bloco. Segundo [WON 82], cada tipo de dado da imagem (texto, figuras, fotografias, etc), possui um tipo específico de formação, que resulta em valores característico destas medidas. Este valores podem ser determinados através da realização de experimentos onde são recolhidos dados estatísticos a respeito de cada tipo de dado. Por exemplo, é possível determinar a presença de textos com caracteres de tamanho entre 10×10 e 20×20 pixels, examinando a altura dos blocos. Uma altura entre 10 e 20 pixels indica uma boa possibilidade de que o conteúdo do bloco seja um texto. Analisando o conjunto de todas as informações extraídas a respeito dos blocos, pode ser obtida uma classificação perfeita de cada tipo de bloco. A figura 3.6 apresenta o resultado da aplicação desta técnica, para o caso dos dados obtidos a partir do documento da figura 3.5.

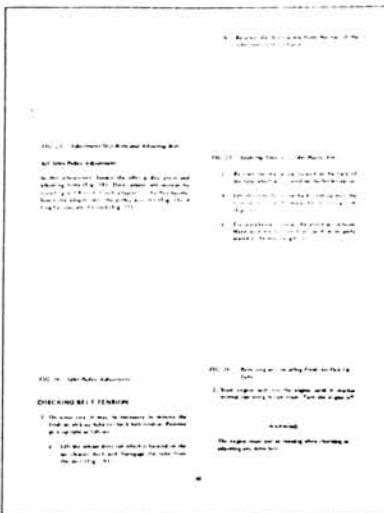


Figura 3.6 - Separação dos textos do documento da figura 3.5

Além desta técnica, existem outras que são normalmente baseadas na identificação prévia da altura e largura dos caracteres [WON 82]. Estas outras técnicas são bastante simples e podem levar a alguns erros na classificação dos elementos que compõem o documento, devido ao fato de considerarem apenas informações como a altura e a largura dos elementos extraídos. O algoritmo RLSA certamente obtém melhores resultados que estes métodos, pois este utiliza uma técnica que considera a altura e largura dos elementos constituintes de cada bloco extraído, além de outras informações complementares.

3.4.1.2 Separação de Caracteres

Depois da separação dos gráficos e textos, pode-se então realizar a separação dos caracteres um a um, de forma a obter pequenas matrizes de pontos que serão fornecidas para o processo de classificação. Existem basicamente dois tipos de processos para a separação de caracteres: separação de caracteres escritos com espaçamento fixo e separação de caracteres escritos com espaçamento variável.

Um texto pode ser escrito usando-se um espaçamento fixo entre caracteres, ou seja, todos os caracteres terão um mesmo tamanho. Este tipo de textos pode ser encontrado em impressoras matriciais, onde a impressão é realizada em modo textual sem uso de recursos gráficos.

Outra possibilidade de se ter um espaçamento fixo é através do preenchimento de um formulário, onde a pessoa que escreve está restrita a pequenos quadros, e em cada quadro deve ser escrito apenas um caractere. A separação dos caracteres neste tipo de textos é simples. Basta que seja localizado um primeiro caractere de uma linha e, a partir deste caractere, quebra-se toda a imagem em pequenos blocos de $M \times N$ pontos, sendo os valores de M e N indicados pelo usuário, ou estimados automaticamente pelo sistema a partir do primeiro caractere identificado. Os valores M e N indicam o tamanho exato de um caractere, em relação a sua altura e largura. Estes valores devem ser considerados com uma pequena margem de erro, de um ou dois pixels em ambas as direções, devido a problemas referentes a possíveis deslocamentos indevidos na escrita do texto ou até a pequenos deslocamento ocorridos na digitalização da imagem.

No caso da digitalização de textos obtidos através da impressão em uma impressora matricial, há o problema dos caracteres não serem formados por linhas e curvas contínuas, mas sim por um conjunto de pequenos pontos, que normalmente se encontram espaçados entre um e o outro. Para estes caracteres serem isolados, pode ser feito uso do fato de terem um espaçamento fixo, ou também através do uso de uma operação de *smoothing*, que permita a união entre os pontos adjacentes. O algoritmo de *smoothing* (suavização) permitirá uma aplicação posterior do algoritmo de separação por continuidade, como será demonstrado logo a seguir. O problema da aplicação de um algoritmo de *smoothing* é que pode ser causada não somente a união dos pontos internos de um caractere, mas também dos pontos pertencentes a dois caracteres. Esta união entre dois caracteres (*touching characters*) irá provocar um erro mais difícil ainda de ser tratado. Como pode ser visto, tais algoritmos exigem que se tenha a possibilidade de realizar ajustes mínimos em relação a parâmetros de funcionamento, tais como o tamanho estimado do caractere, o grau de *smoothing* aplicado na imagem, o tipo de espaçamento, etc.

Isto leva à utilização de sistemas parametrizáveis, o que é feito normalmente através do uso de um arquivo de configuração, a exemplo do utilizado no sistema N²OCR.

Será abordado em maiores detalhes, neste trabalho, o reconhecimento de caracteres com espaçamento variável, pois estes constituem a maioria dos textos impressos ou manuscritos que se manipula. Além disto, o algoritmo de separação de caracteres com espaçamento fixo é trivial, bastando quebrar a imagem em pequenos pedaços, a cada $M \times N$ pixels.

A primeira etapa na separação de caracteres com espaçamento variável é a localização do início da linha de texto. O caractere é localizado buscando-se um pixel com a cor de escrita deste, a partir do topo (canto superior esquerdo) da matriz de pontos que compõe a imagem. Ao ser localizado um ponto que pertence a um caractere, pode então ser realizada a sua extração. A extração de um caractere segue o princípio de que cada caractere é composto por um conjunto de pontos unidos entre si (contínuos). Após serem extraídos os pontos que estão unidos entre si e que formam um caractere, segue-se na direção horizontal em busca de mais um ponto da cor dos caracteres, que ao ser localizado é extraído da mesma forma que o anterior. Este processo se repetirá até que se chegue ao limite extremo direito da imagem, sendo após iniciada uma busca de uma nova linha inferior a que foi identificada e que sofreu a separação. O algoritmo da figura 3.7 apresenta de uma forma global o processo de separação de caracteres com espaçamento variável.

```

    Início
        X=0; Y=0;
        Enquanto Não Terminar a Imagem (X < Xmax, Y < Ymax)
            [
                Acha_Início_da_Linha (X,Y);
                Enquanto Não Terminar a Linha
                    [
                        Isola_Letra (X,Y);
                        Se Em_Treinamento Então Rotula_Letra;
                        Salva_Letra;
                        Avança_Próxima_Letra (X,Y);
                    ]
                Avança_Próxima_Linha (X,Y);
            ]
        Fim
    
```

Figura 3.7 - Algoritmo geral de separação de caracteres

A rotina de "Acha_Início_da_Linha" consiste em dada uma coordenada Y de base para o início da procura, e sendo a coordenada X considerada igual a zero, faz-se uma varredura da imagem, da esquerda para a direita, de cima para baixo em busca de um pixel com a cor de escrita dos caracteres. Como pode ocorrer de uma letra do meio da linha aparecer com pontos mais altos em relação às demais, a varredura ocorre simultaneamente na horizontal e na vertical de acordo com valores de deslocamento previamente especificados.

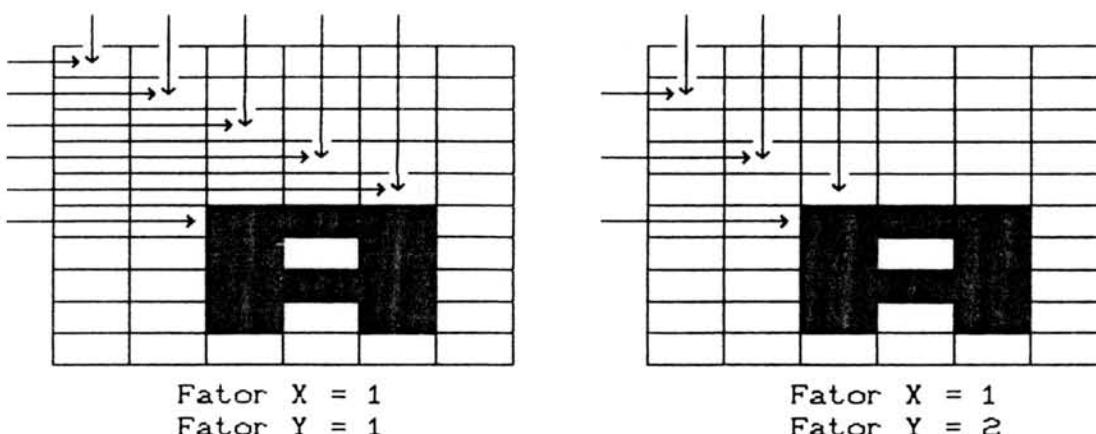


Figura 3.8 - Localização do início de uma linha

A tarefa de identificar o início de uma linha leva em consideração dois fatores configuráveis (fator x e fator y, conforme a figura 3.8) que se encaixam dentro do conceito de um sistema totalmente configurável, de acordo com o tipo de documento analisado. Isto é necessário devido ao fato de os documentos apresentarem diferentes organizações e distribuições do texto em seu interior. Além destes dois fatores podem ser definidos outros, como por exemplo, um fator máximo para o incremento em X e Y. Caso seja importante manter a formatação do texto (o que normalmente é necessário), deve ser possível ao sistema detectar também quando há a presença de espaços em branco ou linhas em branco no texto. Novamente será necessária a definição de valores máximos configuráveis para que se considere X pixels em branco, como sendo um espaço em branco, e Y pixels em branco, como sendo uma linha em branco. O valor de X e Y deve ser um pouco superior ao maior tamanho de um caractere. Todos estes fatores que foram apresentados aqui serão detalhados posteriormente no capítulo que descreve o sistema N²OCR.

Uma vez identificada uma posição na qual se encontra um ponto pertencente a um caractere, é necessário realizar uma operação de extração do mesmo. Esta tarefa é realizada pela função "Isola_Letra", que recebe como parâmetro uma coordenada de um ponto pertencente à letra e devolve uma matriz de pixels contendo o caractere isolado. O algoritmo empregado para isolar todo o caractere se encontra listado na figura 3.9.

Início - Isola_Letra

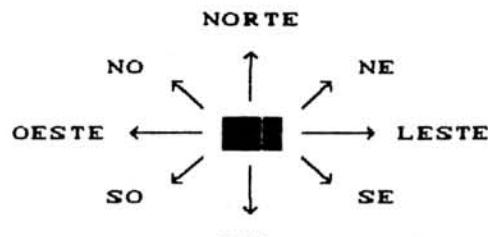
Se Pixel (X,Y) é da Cor do Caractere

Então Faz

```

    Copia Pixel (X,Y) para o Buffer
    Apaga Pixel (X,Y) da Imagem
    Isola_Letra (X+1,Y)
    Isola_Letra (X,Y+1)
    Isola_Letra (X+1,Y+1)
    Isola_Letra (X-1,Y)
    Isola_Letra (X,Y-1)
    Isola_Letra (X-1,Y-1)
    Isola_Letra (X+1,Y-1)
    Isola_Letra (X-1,Y+1)
  
```

Fim



Pesquisa Recursiva

Figura 3.9 - Algoritmo para isolar um caractere

Existem dois problemas básicos que podem ocorrer neste processo de extração de um caractere: caracteres unidos e caracteres compostos por porções isoladas. O caso dos caracteres unidos ou *touching characters*, conforme já foi descrito, será tratado como um único padrão, ou seja, a extração considera ambos os caracteres unidos como sendo um único caractere. No caso de caracteres com porções isoladas, é necessário realizar algumas alterações no algoritmo básico de extração de forma que este consiga extrair corretamente este tipo de caracteres. Um exemplo típico deste tipo de problema são os caracteres com acentuação, onde o acento faz parte do caractere mas está separado do mesmo, ou casos em que, devido a um ruído na digitalização, o caractere ficou com porções desconexas (figura 3.10). A alteração do algoritmo consiste em se determinar um MBB (Menor retângulo envolvente - *Minimum Bounded Box*) do caractere extraído pelo processo descrito anteriormente, adicionar uma margem extra de pontos superiores e inferiores, e varrer todo o interior desta região extraíndo qualquer pixel que ainda não tenha sido lido. A implementação desta função deve ser muito cuidadosa pois pode acabar provocando erros indesejáveis como os *touching characters*.

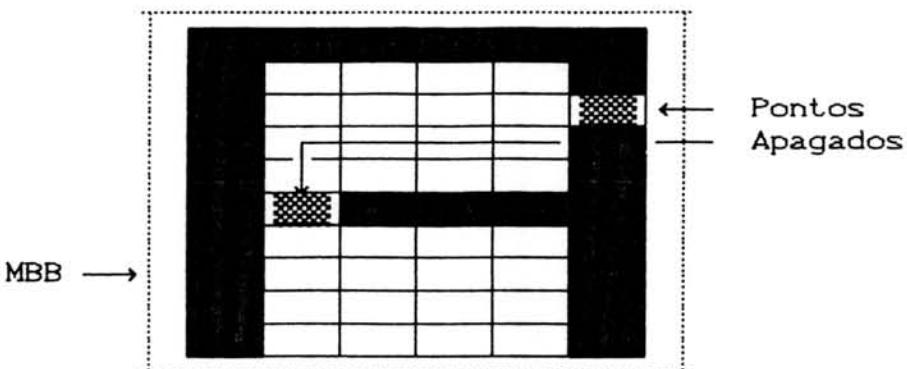


Figura 3.10 - Extração de um caractere desconexo

Ao isolar o caractere, alguns dados são guardados, como por exemplo, as coordenadas Y inferior e superior máximas que este ocupava na imagem. Baseado nestes valores, segue-se em direção à direita, verificando os valores dos pixels que se encontram dentro desta faixa de linhas, até que se encontre um novo pixel da cor do caractere. Neste ponto, realiza-se novamente a extração do caractere, repetindo o processo de busca do próximo e extração até que se chegue ao canto direito final da imagem (ou do bloco extraído do documento de acordo com o algoritmo descrito na seção 3.5.5.1).

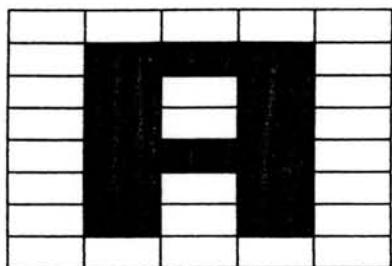
Do processo de separação de caracteres, resulta portanto um conjunto de pequenas matrizes contendo apenas um caractere cada uma (ou dois, caso haja caracteres que se tocam). Estas matrizes servirão como dados de entrada para o processo seguinte do reconhecimento.

3.4.2 Extração de Atributos

Serão apresentados nesta seção alguns dos métodos de extração de atributos de interesse para este trabalho descrevendo-se a sua implementação em maiores detalhes. Estas técnicas apresentadas, recebem denominações próprias dada pelo autor, mas todas elas podem ser classificadas (com nomes iguais ou não) de acordo com a classificação apresentada na seção 3.3.2.

3.4.2.1 Vizinhança Global

Consiste na contagem dos pixels do caractere que são vizinhos a outros pontos do mesmo, em relação a uma dada direção. A vizinhança pode ser imediata ou pode também ser considerada uma vizinhança em segundo grau (distância maior que 1).



Vizinhança Global Imediata

Norte	=	10
Noroeste	=	3
Nordeste	=	3
Leste	=	4

Sul, Oeste, Sudeste e Sudoeste
são redundantes a estes dados

Figura 3.11 - Exemplo de vizinhança global

3.4.2.2 Vizinhança Local

Realização de uma codificação para cada tipo de subconjunto de pontos do padrão (usualmente 3 x 3 pontos). Após definidos os tipos de padrões possíveis, é realizada uma contagem da ocorrência de cada um destes padrões no caractere.

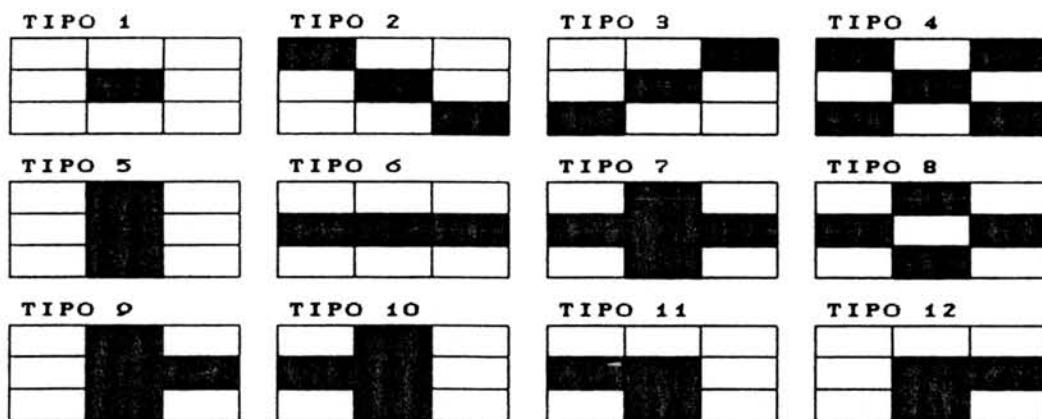


Figura 3.12 - Exemplos de padrões de vizinhança local

Para o caractere representado na figura 3.11, os valores obtidos utilizando-se as matrizes representadas na figura 3.12 são:

- Tipos 1 a 8 : 0 ocorrências
- Tipos 9 a 12 : 1 ocorrência

3.4.2.3 Projeção

Consiste em fazer a contagem do número de pixels pertencentes ao caractere, encontrados em uma certa direção (horizontal, vertical, diagonal esquerda e diagonal direita) de análise do padrão.

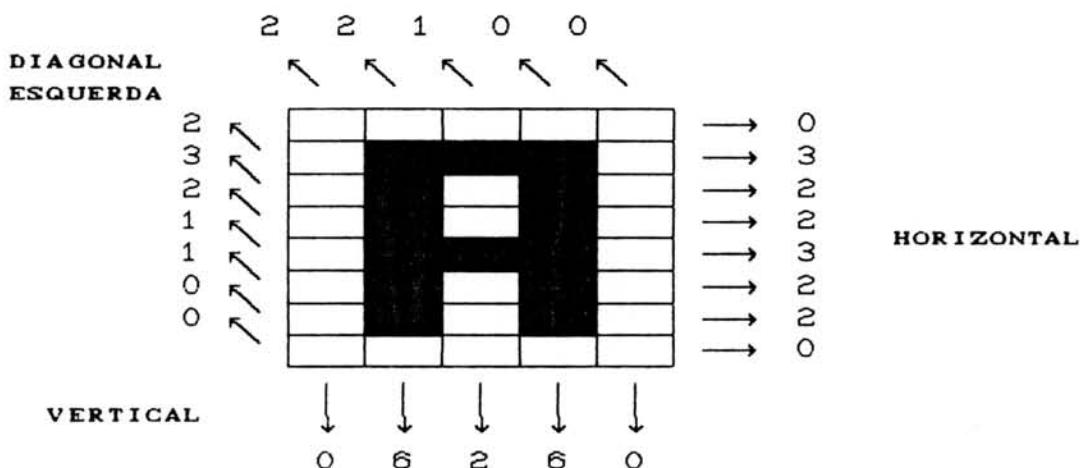


Figura 3.13 - Exemplo de cálculo de projeções

3.4.2.4 Alternância

É similar à projeção mas, ao invés de contar os pixels, conta-se o número de alternâncias de cores, ou seja, conta-se o número de cruzamentos por linhas. Para o caso do caractere representado na figura 3.13 serão obtidos os seguintes valores:

- Alternância Horizontal : 0, 2, 4, 4, 2, 4, 4, 0
- Alternância Vertical : 0, 2, 4, 4, 2, 0
- Alternância Diagonal Esquerda: 0, 0, 2, 2, 4, 2, 4, 4, 2, 0, 0
- Alternância Diagonal Direita : 0, 0, 2, 2, 4, 2, 4, 4, 2, 2, 0, 0

3.4.2.5 Simetria

A partir da definição das projeções é possível definir se o padrão possui alguma simetria. A simetria tanto pode ser relativa, se for tomada em relação a uma linha imaginária central horizontal, vertical ou diagonal, como também pode ser absoluta, se for considerado o padrão como um todo. A figura 3.14 demonstra estas possibilidades.

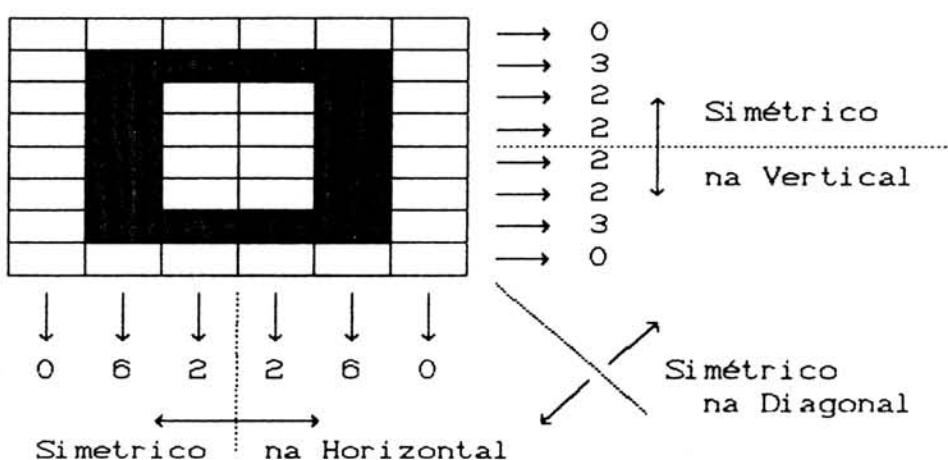


Figura 3.14 - Exemplo de cálculo de simetria

3.4.2.6 Densidade

Uma técnica muito empregada que obtém ótimos resultados consiste em calcular a densidade de pixels em sub-regiões do padrão (figura 3.15). Através desta técnica, é detectada a presença ou ausência de pontos em posições de relevada importância para a identificação dos caracteres.

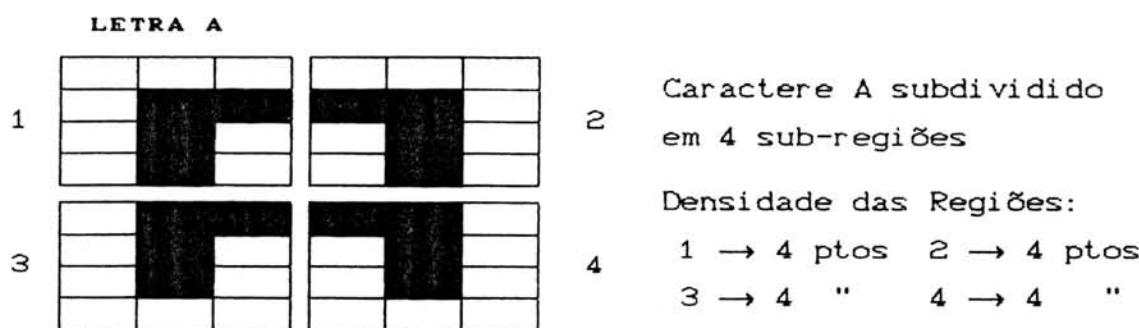


Figura 3.15 - Exemplo de cálculo da densidade de sub-regiões

3.4.3 Alteração de Escala e Posição

Os caracteres que foram isolados através da aplicação do algoritmo de segmentação apresentado na seção 3.5.1 tem diferentes tamanhos. Para que seja possível a utilização destes padrões junto a um sistema de classificação baseado no casamento de padrões, é necessária a implementação de um módulo para padronização de tamanho e posição dos caracteres.

Quanto ao ajuste de posição, todos os caracteres extraídos da imagem (com algumas poucas exceções) devem ser ajustados pelo seu canto superior esquerdo, o qual é tomado como referencial. Existem outras formas de ajuste de posição, como por exemplo, em relação ao centro deste, mas que não serão abordadas neste trabalho. Antes de realizar o ajuste de posição, é realizado um teste em relação ao tamanho do padrão extraído:

1) Se este tamanho for superior ao tamanho utilizado na classificação, então o ajuste será realizado pela extremidade superior esquerda seguindo-se uma alteração de redução do padrão.

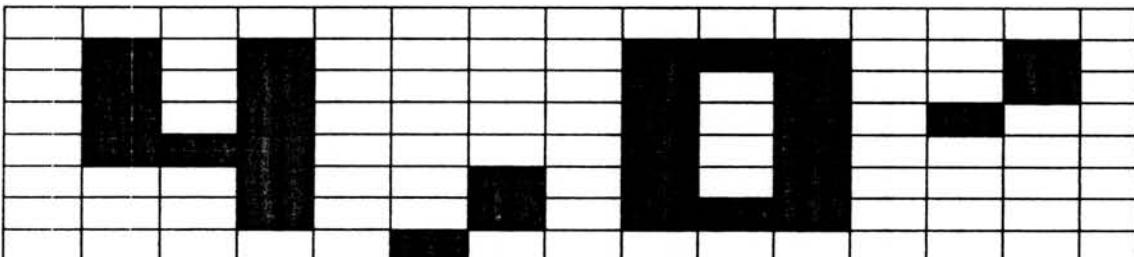
2) Se este tamanho for igual ao tamanho utilizado na classificação, nenhuma alteração é feita;

3) Se o tamanho for menor que o tamanho utilizado na classificação, é feito um "ajuste especial" de posição, sem realizar uma alteração do tamanho real do caractere extraído. Este caso ocorre em caracteres como : ponto, vírgula, aspas, etc.

O "ajuste especial" de posição consiste em manter as características essenciais do caractere extraído, de forma a possibilitar a sua identificação. Caso seja feito sempre o ajuste pelo canto superior esquerdo, tanto a vírgula (",") como o apóstrofo ("'") serão identificados como um único caractere, não possuindo nada que os diferencie.

Nestes casos, o padrão possui um posicionamento onde deve ser mantida uma relação com a linha original em que ele estava inserido, para que seja feita uma classificação correta. Este processo deve ser realizado junto à separação dos caracteres, pois é nesta etapa que será possível identificar o posicionamento do caractere em relação à linha em que está contido. O processo é bem simples, em vez de ajustar a dimensão do caractere, cujo tamanho é sensivelmente inferior ao tamanho médio dos outros caracteres obtendo-se assim uma matriz de dimensões reduzidas, faz-se a extração do mesmo, mantendo a sua posição relativa a linha (altura), e preenchendo com pixels da cor de fundo o restante da matriz, de forma que esta tenha o mesmo tamanho dos caracteres "normais" da linha. Este tipo de procedimento está demonstrado na figura 3.16.

IMAGEM ORIGINAL



CARACTERES ISOLADOS

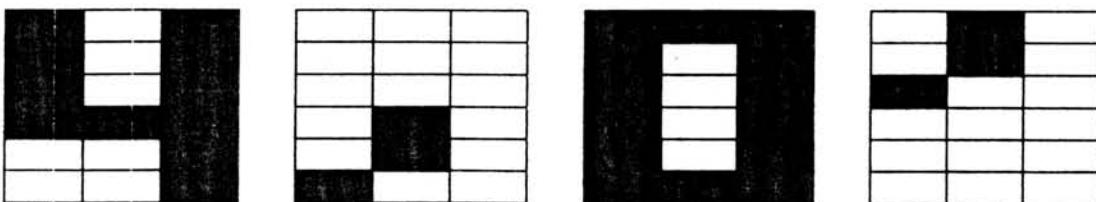
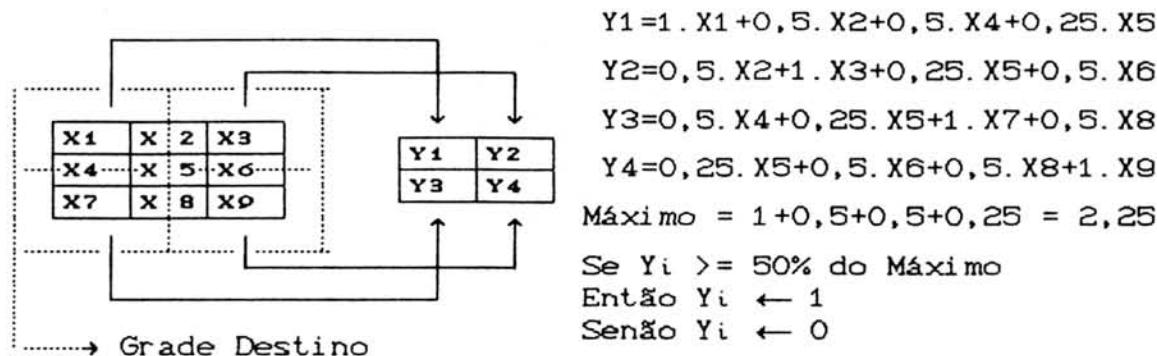


Figura 3.16 - Separação e posicionamento de caracteres

O ajuste de escala deve realizar uma conversão de um padrão de tamanho $M \times N$ para outro padrão $J \times K$ equivalente, onde M , N , J e K são números inteiros quaisquer. É interessante notar que J e K não precisam ser múltiplos exatos de M e N , o que implica na criação de um algoritmo um pouco mais complexo. A figura 3.17 apresenta um exemplo de mapeamento de um padrão 3×3 para outro 2×2 .

Figura 3.17 - Redução de um padrão 3×3 para 2×2

O algoritmo, apresentado na figura 3.18, que realiza a alteração da escala é uma generalização do processo descrito através da figura 3.17. O primeiro passo é a determinação de duas grades, uma original (grade da matriz do padrão) e uma destino (grade do padrão com a escala alterada). A sobreposição destas duas grades determinará a influência (ponderação) que cada pixel da grade origem terá sobre um pixel da grade destino. No caso de uma redução, existe um ou mais pixels da grade origem a influenciar a cor resultante do pixel correspondente na grade destino e, no caso de uma ampliação, existe um pixel da grade original a influenciar um ou mais pixels da grade destino. O cálculo da cor final dos pixels na grade destino ocorre da seguinte forma: se o somatório da multiplicação dos valores dos pixels da grade origem (referentes a um pixel da grade destino) pela sua influência for maior que 50% do valor máximo possível de ser atingido neste somatório, então o pixel terá o valor 1 (aceso) e, em caso contrário, terá o valor 0 (apagado). Esta técnica, na forma em que está descrita, deve ser empregada apenas em imagens binarizadas.

```

SE  $\sum (\text{Pixel\_Origem} \cdot \text{Influencia\_no\_Pixel\_Destino})$ 
     $\geq (0,5 \cdot \sum (1 \cdot \text{Influencia\_no\_Pixel\_Destino}))$ 
ENTÃO Pixel_Destino = 1
SENÃO Pixel_Destino = 0

```

Figura 3.18 - Algoritmo de alteração de escala

Este algoritmo produz resultados razoáveis que satisfazem o tipo de aplicação que está sendo abordado. É claro que algumas deformações podem ocorrer na aparência do caractere, mas uma vez que para um mesmo padrão, a mesma deformação será gerada, então este problema não afetará o reconhecimento de caracteres. Existem outros algoritmos de mudança de escala, como o apresentado em [CAS 82], que não provocam deformações, mas a sua implementação implica tempos de processamento muito altos, incompatíveis com o desempenho esperado de um sistema OCR.

3.5 Perspectivas

Os sistemas de OCR desenvolvidos até o presente momento possuem um bom desempenho, mas ainda há muito a ser feito, pesquisado e melhorado, para que se tenha sistemas com uma capacidade de reconhecimento igual ou superior a dos seres humanos. Dentre os diferentes aspectos abordados neste capítulo alguns se destacam, indicando tendências das pesquisas atuais em sistemas OCR. São eles: a capacidade de adaptação, o processamento paralelo e o uso de contexto.

3.5.1 Adaptação

O ser humano possui uma grande capacidade de adaptação ao meio em que está inserido. Através do aprendizado, ele consegue se adaptar a novas situações. No caso específico do reconhecimento de caracteres, o ser humano possui uma grande facilidade para o aprendizado de diferentes conjuntos de símbolos, podendo desta forma ampliar o conjunto de elementos que ele reconhece. Além de aprender novos padrões, o ser humano também possui a capacidade de se "ajustar" a um tipo de escrita, acostumando-se a interpretá-la, mesmo que esta a princípio pareça ilegível.

Esta capacidade de aprendizado e adaptação tem levado muitos pesquisadores a buscar formas de incorporá-las aos sistemas de OCR. Com isto, tais sistemas se tornam mais flexíveis e possivelmente terão seu desempenho melhorado. Afinal nada mais lógico do que, na busca de um sistema que simule o comportamento humano de ler textos, se fazer a implementação de recursos similares aos que o ser humano utiliza para realizar este reconhecimento.

Estes sistemas dotados de aprendizado e adaptação podem ser implementados de diferentes formas, como por exemplo, a descrita em [MAR 89]. Uma das maneiras muito interessantes de dotar um sistema OCR com estas características é através do emprego de redes neurais, as quais serão apresentadas no capítulo 4. Um dos motivos da adoção das redes neurais como forma de implementação do sistema N²OCR foi justamente pelo fato delas possibilitarem que o sistema tenha esta característica de aprendizado e adaptação.

3.5.2 Processamento Paralelo

Como já foi citado, o processamento paralelo possibilita o aumento do desempenho dos sistemas, através da execução simultânea de tarefas. No caso do casamento de padrões, a possibilidade de acelerar o processo de classificação através do uso de processamento paralelo fica muito clara. Uma vez que, neste processo, é necessário comparar um padrão não identificado com N outros padrões modelos, a fim de identificar qual é o mais parecido, pode-se realizar em paralelo a tarefa de comparação entre padrão desconhecido versus padrões modelo, já que não há uma dependência entre uma comparação e outra.

O processamento paralelo tem se tornado uma alternativa viável devido ao desenvolvimento atual da tecnologia, que possibilitou o surgimento de novos equipamentos baseados no processamento paralelo.

Em vista deste fato, nesta última década, muitas aplicações começaram a fazer uso de equipamentos que possibilitassem este tipo de processamento. Os sistemas OCR não são uma exceção, podendo também ser implementados em equipamentos que possuam uma arquitetura paralela, e que possibilitem uma aceleração do processamento.

As redes neurais são uma proposta de um novo modelo de computação, baseado no processamento paralelo. Este é mais um dos motivos que levaram à sua utilização junto ao sistema N²OCR.

3.5.3 Uso de Contexto

A tarefa de reconhecimento pode ser desenvolvida de forma a permitir um pós-processamento. A idéia básica consiste de, na classificação, indicar não apenas a classe mais adequada à classificação do padrão, mas sim um conjunto de classes mais prováveis às quais o padrão pode pertencer. Este conjunto de classes possíveis serão avaliadas, conforme o contexto em que estão inseridas, de forma a selecionar a melhor destas classes.

O contexto também pode ser utilizado em uma fase de pré-processamento onde, antes mesmo de iniciar o reconhecimento do padrão, já se possui uma lista de prováveis candidatos ao reconhecimento. Com isto, é reduzido o espaço de soluções possíveis, facilitando o reconhecimento e diminuindo a possibilidade de ocorrência de erros. No entanto, a forma mais empregada em relação ao uso de contexto, é o pós-processamento.

Em resumo, o emprego do contexto para a correção do texto reconhecido em uma etapa de pós-processamento é uma importante etapa do reconhecimento, que atua em um nível mais alto do processamento das informações. A idéia de contexto consiste em se observar características gramaticais, sintáticas ou semânticas do texto em reconhecimento. Com o uso do contexto pode-se corrigir muitos erros, melhorando sensivelmente o índice de reconhecimentos corretos [HAR 72].

Sabe-se que o contexto é utilizado naturalmente pelo ser humano ao realizar uma tarefa como a leitura de um texto, sendo que já foi demonstrado que o ser humano causa cerca de 4% de erros quando realiza a leitura de um texto na ausência de contexto [SUE 80]. O homem é também capaz de remontar um texto, sem numerais e inserido em um contexto, mesmo que este possua 25% dos caracteres removidos randomicamente [HAR 72]. Deve-se salientar que a noção de contexto está intimamente ligada à língua utilizada na representação do texto (inglês, português, alemão, etc), e portanto estudos desta natureza devem levar em consideração aspectos lingüísticos.

Assim como o homem utiliza o contexto no reconhecimento de caracteres, é portanto interessante que um sistema de OCR siga este exemplo e também se utilize deste tipo de informações para melhorar o seu desempenho. O problema da implementação de um sistema OCR com uso de regras de validação do texto reconhecido através do contexto consiste em se determinar este conjunto de regras. Como já foi citado, estas regras estão intimamente ligadas a aspectos lingüísticos e, portanto, sua determinação requer amplos conhecimentos em relação a esta área de estudos. Um trabalho de estudo de aspectos lingüísticos visando a aplicação do contexto na correção de textos resulta em um estudo extenso e que escapa do escopo deste trabalho. Desta maneira, será considerada apenas a implementação do sistema N²OCR de tal forma que este possa ser estendido por um módulo complementar para a validação por contexto. Este módulo não será estudado nem implementado na versão atual do sistema, sendo apenas descrita, no capítulo 5, a forma como pode ser adicionado ao sistema N²OCR este módulo para validação do texto reconhecido através do contexto.

4 REDES NEURAIS

Neste capítulo será apresentada uma visão global sobre as redes neurais e a sua utilização junto ao reconhecimento de padrões. O objetivo é apresentar, de uma maneira simples e resumida, os conceitos desta área de conhecimento, de modo a possibilitar ao leitor uma introdução ao assunto. São dadas desta forma condições para um maior entendimento do que são e para que servem as redes neurais, bem como são fornecidos subsídios para um maior aprofundamento no assunto. Após serem apresentados os principais tópicos a respeito das redes neurais, poderá ser justificada a escolha do modelo adotado na implementação do sistema N²OCR, bem como ficará mais clara a forma de funcionamento e características deste sistema.

Serão apresentados um histórico, a evolução e os modelos de redes neurais de grande importância na atualidade, onde para cada modelo serão analisadas as suas características principais, vantagens e desvantagens, no que diz respeito ao reconhecimento de padrões. Os modelos abordados são: Perceptron, Adaline, Madaline, rede multinível, modelo de Hopfield, máquina de Boltzmann, BAM, modelo de Kohonen e ART.

O estudo das redes neurais é uma área de pesquisas multidisciplinar, envolvendo conhecimentos de diversas áreas como: Medicina, Biologia, Física, Computação, e outras. Isto gera certos "conflitos" a cerca da nomenclatura e dos diferentes enfoques na abordagem do assunto. Neste capítulo será abordado o assunto orientado à Ciência da Computação.

4.1 Conceitos Básicos

As redes neurais foram desenvolvidas a partir de uma tentativa de criar um modelo que descrevesse a estrutura e o funcionamento dos neurônios do cérebro humano.

Os modelos de redes neurais buscam definir novos computadores ou novos modelos de processamento de dados. Estes devem apresentar um comportamento baseado em modelos neurobiológicos ao invés de modelos baseados em "circuitos de silício" (portas lógicas, circuitos combinacionais, biestáveis, etc).

O sistema nervoso humano é o responsável pelas tarefas de automatismo, decisões, integração de idéias ou sensações, e de adaptação do organismo e do próprio ser, sendo esta última realizada através do aprendizado. O centro do sistema nervoso é constituído por células que são responsáveis pelo seu funcionamento, onde estas recebem, geram e transmitem os estímulos que chegam ou partem de nosso cérebro. Estas células são denominadas de neurônios e estão presentes em nosso cérebro em uma quantidade da ordem de 10^{11} com cerca de 10^{15} conexões entre elas [KOH 88a]. A figura 4.1 apresenta a estrutura biológica de um neurônio, a qual foi utilizada como base para o desenvolvimento da teoria sobre redes neurais.

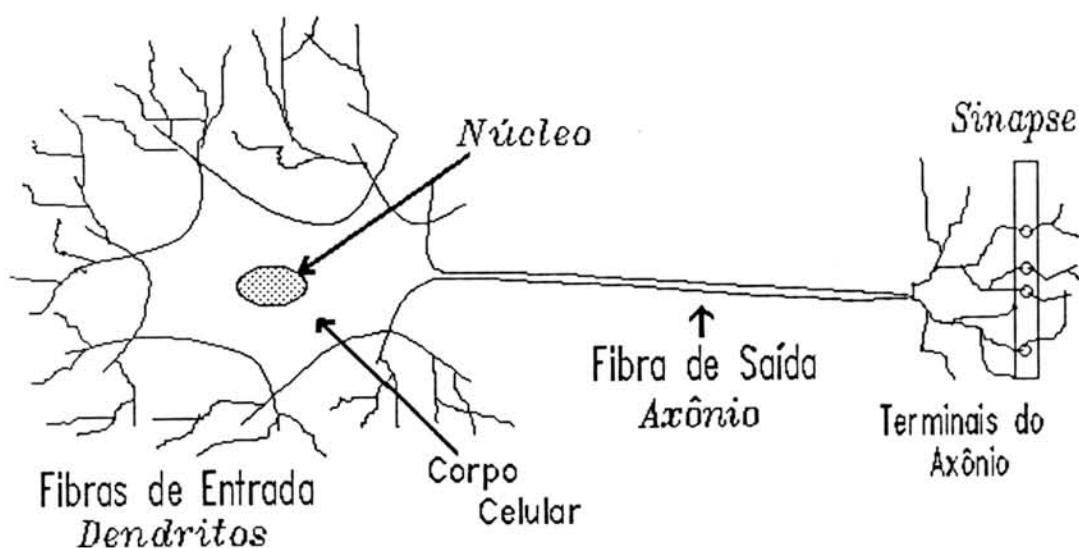


Figura 4.1 - Aparência externa de um Neurônio

A partir da observação do comportamento de sistemas reais (redes neurais de seres humanos), foi possível descrever de uma maneira muito simplificada o comportamento do neurônio. Este comportamento, de uma forma bem abstrata, será descrito a seguir. Os sinais (estímulos) de entrada, recebidos pelo neurônio através de seus dendritos, fazem com que este passe a gerar ou não um sinal de saída, o qual é enviado através do axônio. Cada neurônio é uma unidade independente de processamento de informações que está conectada a diversos outros neurônios através dos terminais do axônio. Estes terminais se conectam aos dentritos de outros neurônios, através de uma estrutura biológica denominada sinapse que possibilita a passagem dos sinais. A sinapse existente entre dois neurônios irá influenciar sobre o caráter do sinal recebido, em função da atuação dos neurotransmissores, fazendo com que para um dado neurônio um sinal de entrada venha a funcionar como sinal excitador e para outro neurônio este mesmo sinal poderá atuar como inibidor.

Mas então surge a pergunta: o que tem a ver uma ciência exata, como é a computação, com estes conceitos biológicos? A resposta é simples, a partir do conhecimento da estrutura e do comportamento destes elementos básicos, é possível extrair suas características fundamentais e partir para a criação de modelos que imitem os sistemas reais. Estes modelos são utilizados na criação das chamadas *redes de neurônios artificiais*, também conhecidas como : *redes neurais*, *neuroniais* ou *neuronais*, *modelos conectivistas* (*connectionist models*), *conectivismo* (*connectionism*), *inteligência natural* ou *processamento paralelo distribuído* (PDP). O nome *rede de neurônios artificial* é o mais adequado, mas o nome de uso corrente, adotado na atualidade para denominar estes sistemas, é *redes neurais*.

A partir da definição de um modelo específico pode-se implementar programas de simulação, ou então implementar um hardware dedicado que implemente funcionalmente o modelo, utilizando-se de uma arquitetura de processamento paralelo.

Para se obter uma definição mais exata das redes neurais, é preciso determinar antes como funciona o neurônio (modelo comportamental). Ele possui várias entradas, as quais recebem estímulos de entrada (entrada ativa → valor positivo, entrada inativa → 0 ou valor negativo). De acordo com uma ponderação - pesos sinápticos ou pesos de atuação - sobre os sinais de entrada, estes terão um caráter excitatório ou inibitório, fazendo com que o neurônio se "ative", enviando assim um sinal de saída, ou não se ative. Este sinal de saída será propagado de acordo com a topologia de interconexões da rede de neurônios.

Este é portanto um primeiro modelo de neurônio, formado por um integrador de sinais que avalia cada estímulo de entrada com o respectivo peso de atuação. Conforme a avaliação total das entradas, poderá ser gerado ou não um sinal de saída único, que será propagado pela rede através de suas interconexões. Existe um maciço paralelismo entre os elementos processadores (neurônios), todos avaliam em paralelo os seus estímulos de entrada.

O modelo descrito acima é uma descrição superficial do *Psychon* (figura 4.2), que foi o resultado do trabalho precursor de McCulloch e Pitts [MCC 43] na definição de modelos baseados no estudo do sistema nervoso. O *Psychon* é um dispositivo lógico de dois estados que, quando as entradas somadas ultrapassam um valor limite de excitação, gera um sinal binário de saída. Baseado apenas nestas informações já seria possível especificar a implementação de um simulador para o mesmo.

O modelo do *Psychon*, apesar de representar razoavelmente bem o funcionamento de um neurônio, não previa a capacidade de adaptação. Não se pode formar novas memórias

com este modelo, uma vez que a ponderação dos estímulos é determinada externamente. Um trabalho muito importante que seguiu-se ao do *Psychon*, foi o do neurofisiologista D. O. Hebb [HEB 49], que influenciou a criação de muitos outros modelos.

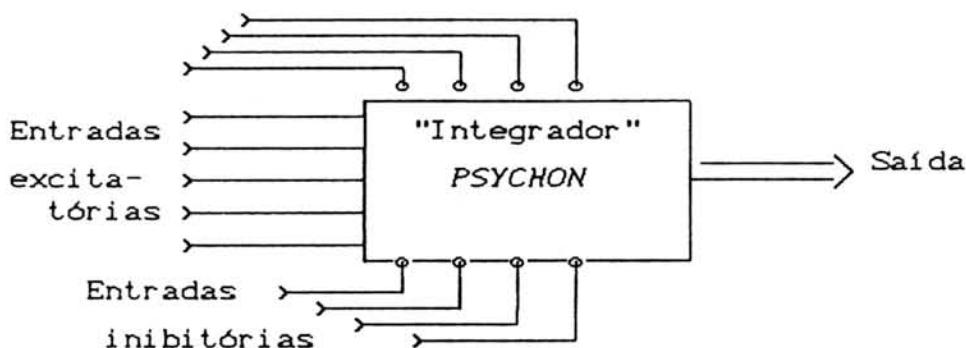


Figura 4.2 - Modelo do *Psychon* de McCulloch & Pitts

O trabalho de Hebb não resultou diretamente em um modelo específico e bem formalizado de uma rede neural, mas sim em uma análise das características e formas de comportamento que uma rede neural deveria possuir. Hebb propôs que a capacidade de aprendizado de uma rede neural reside na auto-organização das suas ligações sinápticas, ou seja, o aprendizado ou memória reside na distribuição de intensidades das ligações entre os neurônios. Afirma Hebb que pares de neurônios que são ativados simultaneamente tornam-se mais fortes através de alterações em suas sinapses (pesos de atuação). Ele estabeleceu que as sinapses mais freqüentemente ativadas devem ter maior chance de se tornarem ativas novamente.

Baseando-se nos conceitos vistos anteriormente, é possível agora se determinar quais os itens que devem ser detalhados, visando a definição de um modelo específico de uma rede neural. Através do detalhamento destes itens, diferentes modelos de redes neurais são definidos, sendo que alguns destes modelos se destacam, e serão analisados

posteriormente neste capítulo. Os itens a serem especificados na definição de um modelo de rede neural são os seguintes [CAU 90]:

1) *Topologia da Rede* : Definição da forma de interconexão entre os neurônios, destacando-se as organizações de um nível linear, retro-alimentada, totalmente conectada, multinível, matricial ou livre (grafo). A figura 4.3 apresenta o esquema de algumas destas topologias de interconexão entre os neurônios.

2) *Definição do Neurônio Formal* : Especificação comportamental de um neurônio isolado. O que diferencia basicamente os neurônios é a denominada função de transferência. Esta função de transferência pode ser linear ou não, e determina o disparo (ativação da saída) ou não de um neurônio. A figura 4.6 apresenta as principais funções de transferência empregadas junto às redes neurais.

3) *Regra de aprendizado* : Determinação das formas pelas quais os pesos de atuação da rede serão modificados. Alterar os pesos de entrada dos neurônios implica alterar seu comportamento diante de um conjunto específico de estímulos de entrada.

Esta alteração de um comportamento diante uma dada situação (conjunto de estímulos) consiste de um aprendizado. A alteração dos pesos segue certas regras específicas, podendo ser um aprendizado do tipo supervisionado ou auto-aprendizado (*supervised* ou *self-learning*). Maiores detalhes a respeito dos itens vistos acima serão dados ao serem apresentados os modelos específico de redes neurais.

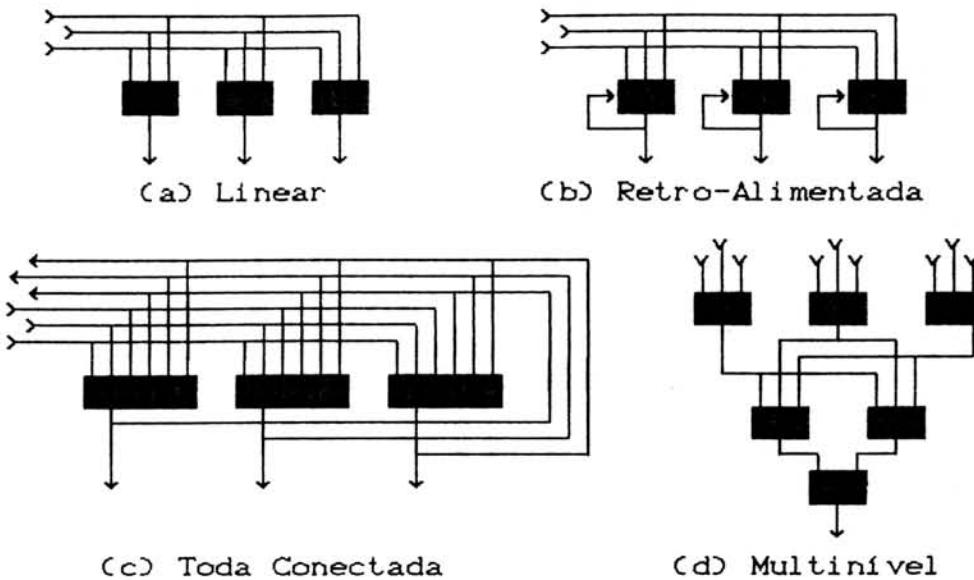


Figura 4.3 - Topologias de interconexões em redes neurais

4.2 Origem e Evolução

A origem e evolução dos estudos sobre redes neurais estão baseadas nos modelos que surgiram a partir dos anos 40s até a atualidade. Estes modelos, que se destacaram e tornaram-se famosos, foram sendo desenvolvidos ao longo deste período e a ordem cronológica dos fatos ajuda a compreender melhor esta evolução. Todos os modelos abordados neste item serão posteriormente descritos em maiores detalhes.

O estudo das redes neurais começou basicamente com a criação do *Psychon* em 1943 por McCulloch e Pitts [MCC 43], sendo que, em 1949, D. O. Hebb publicava a sua importante obra "*The organization of Behavior*" [HEB 49], que veio no futuro a influenciar vários modelos de redes neurais.

Em 1959, Frank Rosenblatt criou o *Perceptron* [ROS 59] que, como será visto, tem até hoje uma grande influência nos estudos sobre as redes neurais. Alguns outros modelos similares ao Perceptron foram também desenvolvidos nesta época, como é o caso do *Adaline*, criado por Bernard Widrow em 1962 [WID 62]. Estes modelos são baseados na correção de erros e formam uma importante classe de redes neurais.

Foi em 1969 que os modelos baseados no Perceptron receberam uma dura crítica feita por Minsky e Papert, através de sua obra "Perceptrons: An introduction to computational geometry" [MIN 69]. Através deste livro, ficou provado matematicamente que os modelos de redes neurais de apenas um nível baseados no Perceptron (aqueles que eram utilizados até o momento), não eram capazes de aprender uma simples função lógica "ou-exclusivo" (XOR - Exclusive OR). A função XOR possui um padrão de valores de entrada e saída que não podiam ser aprendidos pelos modelos de redes neurais propostos até aquela época.

O Madaline, criado também por Widrow, podia de certa forma resolver o problema, mas o aprendizado não podia ser realizado de uma forma muito "natural", como será visto posteriormente. Desta forma as redes neurais ficaram esquecidas por um certo tempo ...

Somente em 1982, surgia um novo modelo de rede neural de destacada importância. Criado por J. Hopfield [HOP 82], este novo modelo fez os estudos sobre as redes neurais ganharem um novo impulso. O modelo de Hopfield propunha um tipo de rede neural da classe das redes baseadas em sistemas competitivos, utilizando-se dos conceitos definidos por Hebb. Outros modelos similares ao de Hopfield surgiram logo após, todos pertencentes à classe de modelos com aprendizado competitivo. Estes novos modelos eram: a máquina de Boltzmann [HIN 84] e a BAM [KOS 87].

A década de 80 ficou marcada pelo aparecimento de um grande número de modelos importantes de redes neurais. Isto se deve principalmente à disponibilidade de novos computadores, mais velozes e com maiores potencialidades, o que permitiu a criação de um número maior de novos simuladores para as redes neurais, e também devido ao pouco sucesso da Inteligência Artificial em relação a uma série de aplicações (percepção visual, reconhecimento de fala, etc). Esta década foi muito importante, pois além do modelo de Hopfield foi desenvolvido o modelo multinível, que se utiliza da

regra de aprendizado de *Back-Propagation*. Este modelo foi desenvolvido por D. Parker [PAR 82] e D. Rumelhart [RUM 85] sendo baseado no Perceptron com correção de erros, mas implementando redes de múltiplos níveis. Estas redes podem inclusive resolver o problema do aprendizado da função XOR.

Além dos modelos de Hopfield e do modelo multinível, outro importante modelo também surgiu na década de 80, o modelo de Teuvo Kohonen, criado também em 1982 [KOH 82]. O modelo de Kohonen, é muito interessante pois permite um aprendizado competitivo e uma auto-organização da rede neural, criando os chamado "mapas de atributos de auto-organização".

Por fim o último modelo de destaque que será abordado é o modelo do ART, criado em 1983 por G. Carpenter e S. Grossberg [CAR 83]. Este modelo possui um aprendizado do tipo não-supervisionado. O modelo do ART teve várias versões posteriores, sendo que só será abordada neste trabalho a primeira versão.

Os estudos sobre as redes neurais sofreram uma grande evolução a partir dos anos 80, conforme foi demonstrado acima. E, a partir desta época, cada vez mais, esta área de estudos tem se destacado, seja pelas promissoras características apresentadas pelos modelos de redes neurais propostos ou pelas condições tecnológicas atuais de implementação de redes neurais em arquiteturas paralelas dedicadas.

4.3 Características e Aplicações das Redes Neurais

As redes neurais possuem a característica de serem muito apropriadas ao reconhecimento de padrões. Como já foi visto, as redes neurais podem sofrer um aprendizado, modificando seu comportamento frente a um conjunto de estímulos de entrada (padrão de entrada). Portanto, a rede pode aprender a dar uma resposta específica para um determinado conjunto de estímulos fornecidos. Isto será obtido através da alteração dos pesos de atuação das entradas.

Logo as redes neurais são muito adequadas para o reconhecimento e classificação de padrões, pois podem se adaptar para responder a um padrão específico.

As redes neurais possuem também uma alta velocidade de processamento, devido ao seu maciço paralelismo interno, possibilitando o desenvolvimento de certos tipos de aplicações complexas que necessitam operar em tempo real.

Devido às características inerentes às redes neurais, estas podem realizar alguns tipos de tarefas que não são executadas de uma forma satisfatória em sistemas computacionais tradicionais, mas que para o ser humano são tarefas triviais. Elas possuem a característica de se adequar perfeitamente às seguintes aplicações :

- Reconhecimento e síntese contínua da fala [KOH 88, WAI 89 e SEJ 86];
- Reconhecimento visual de padrões, OCR [LIP 87, BRO 87 e CAR 80] e padrões em geral [FUK 88 e REI 89]. Reconhecimento e classificação de imagens como, por exemplo: textos, assinaturas, impressões digitais, objetos, etc;
- Processamento adaptativo de sinais [WID 88] e eliminação de ruídos [DOW 89 e KLI 89];
- Aplicações onde os dados fornecidos são incompletos e os resultados produzidos são aproximados (*fuzzy*).

Além destas aplicações, também encontram-se referências de uso das redes neurais em outras aplicações como: memórias associativas, veículos autônomos, previsão do tempo, análise de finanças, gerenciamento de bases de dados e sistemas especialistas.

Apesar desta capacidade de realizar o reconhecimento de padrões de uma forma eficiente, as redes neurais, assim como os seres humanos, já não são tão rápidas e precisas no tratamento de dados numéricos.

O homem ao realizar cálculos aritméticos está sujeito a muitos erros possíveis, sendo também muito lento na realização de cálculos complexos e que necessitam de uma grande precisão. Estes cálculos exigem a capacidade de manipulação simbólica, sendo difícil a realização do mapeamento deste tipo de funções para uma rede de neurônios artificiais.

As redes neurais, devido as suas características, são utilizadas para trabalhar com dados aproximados, tanto de entrada como de saída. Um sistema baseado em redes neurais não pode ser considerado muito eficiente para aplicações como uma folha de pagamento. Por isso, o uso das redes neurais deve ser considerado como uma extensão para os sistemas tradicionais, por exemplo, com o uso de placas de redes neurais co-processadoras acopladas a estes. Estas placas podem servir para executar as tarefas que não são facilmente implementáveis em sistemas convencionais, como por exemplo, reconhecimento e síntese de fala; enquanto os sistemas convencionais continuam a realizar aquelas tarefas típicas onde obtém um bom desempenho.

4.4 Modelos de Redes Neurais

As redes neurais trabalham basicamente sobre uma estrutura como a apresentada na figura 4.4. Este modelo de neurônio serve de base para os demais que serão apresentados nesta seção. Serão vistos apenas oito modelos, mas estes são os que possuem uma destacada importância em termos de histórico, evolução e aplicações na atualidade [LIP 87 e WID 90].

Existe uma série de outros modelos possíveis de serem implementados, mas que não serão abordados aqui porque isto tornaria este trabalho excessivamente extenso. A maioria destes modelos não abordados são variações dos modelos básicos apresentados neste capítulo ou modelos que foram desenvolvidos e não possuem uma grande repercussão junto aos principais grupos de pesquisa sobre redes neurais.

Alguns destes modelos levam em consideração, por exemplo, o tempo de propagação dos sinais ou tentam modelar curvas de resposta o mais similares possíveis do comportamento real dos neurônios. Na opinião do autor, alguns fatores, como os citados acima, podem ser desprezados no caso dos sistemas de reconhecimento visual, lembrando-se que um modelo de simulação é uma simplificação da realidade que permite, em termos funcionais, uma reprodução de um comportamento de um sistema real. A reprodução exata da realidade pode não ser a solução mais adequada para uma certa aplicação.

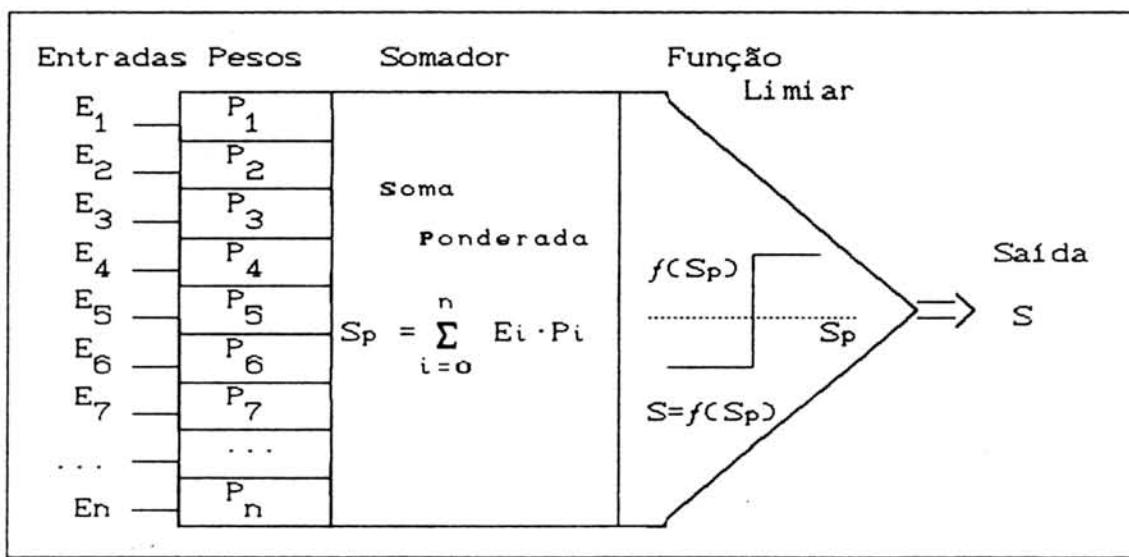


Figura 4.4 - Estrutura de um neurônio artificial

4.4.1 Classificação dos Modelos

Os diferentes modelos de redes neurais podem ser classificados em diversos grupos, conforme as suas características e propriedades. Estes grupos dividem as redes neurais em categorias de acordo com:

- Treinamento: auto-aprendizado (*self-learning, unsupervised learning*) ou aprendizado supervisionado (*supervised learning*) [LIP 87];

- Regra de Aprendizado: algoritmo competitivo (*competitive learning*) ou algoritmo adaptativo por correção de erros (*adaptive error correction*) [WID 90];
- Interconexões dos Neurônios: redes com neurônios sem realimentação (*feed-forward*) ou com realimentação (*feedback*) [LAW 90];
- Organização da Rede: um nível único ou em diversos níveis (*multi-layer*) [LIP 87];
- Distribuição das Memórias: localizada (*grand-mothering*) ou distribuída (*generalization*) [CAU 90];
- Classificação: separador linear ou separador não linear [LAW 90];

Os diferentes tipos de treinamento (auto-aprendizado ou aprendizado supervisionado) estão diretamente relacionados com estes mesmos dois tipos de treinamento utilizados em sistemas de reconhecimento de padrões, abordados no capítulo 2. Em resumo, o treinamento supervisionado consiste em apresentar à rede um padrão a ser reconhecido, juntamente com a resposta que a rede deve fornecer ao reconhecer novamente este mesmo padrão. O treinamento com auto-organização ou auto-aprendizado, consiste em apresentar apenas os padrões para a rede e esta se encarrega de agrupar aqueles que possuem propriedades similares (*clusterization*).

Uma importante categorização das redes neurais é em relação à regra de aprendizado, onde existem dois tipos bem distintos de redes neurais: as redes com regras de aprendizado competitivas e as redes baseadas na correção de erros. As redes neurais que se utilizam de regras de aprendizado do tipo competitivo são as redes de Hopfield e Kohonen, caracterizando-se pelas conexões laterais dos neurônios com os seus vizinhos e estabelecendo assim uma "competição entre os neurônios", que levará a rede a um estado estável. Os algoritmos de agrupamento (*clustering*) como o de Kohonen são exemplos típicos de redes com regra de aprendizado do tipo competitivo.

As redes neurais com aprendizado do tipo correção de erros são baseadas no princípio de adaptação e correção dos pesos de atuação de cada neurônio, até que este responda da maneira desejada. Este tipo de aprendizado é utilizado em redes baseadas em neurônios como o Adaline, utilizando-se de regras como a *Delta Rule*, que será apresentada posteriormente. A correção de erros está diretamente ligada ao aprendizado do tipo supervisionado ou tutorado.

Quanto à interconexão dos neurônios, costuma-se separar os modelos de redes neurais em redes do tipo *feed-forward* ou *feedback*. As redes do tipo *feed-forward* possuem neurônios cujas saídas conectam-se apenas com os níveis inferiores da rede, não havendo realimentação de suas entradas. Neste tipo de redes, a saída de um neurônio não influenciará a sua própria resposta. Em uma rede do tipo *feedback*, um neurônio pode se conectar a uma de suas próprias entradas ou à entrada de um neurônio em um nível superior da rede. Desta forma, a rede do tipo *feedback* possui neurônios cuja saída é capaz de influenciar de maneira direta ou indireta o seu próprio comportamento. Um exemplo típico desta rede é o modelo de Hopfield.

Uma rede pode ter uma topologia de interconexões, cuja organização seja em uma única camada (nível) ou em múltiplas camadas. As figuras 4.3 (a), (b) e (c) apresentam redes de um único nível e a figura 4.3 (d) apresenta uma rede do tipo multinível.

Uma rede neural forma memórias, ou seja, é capaz de guardar informações ou, no que se refere a este trabalho, a rede neural pode memorizar padrões. A memória de uma rede neural fica localizada nos pesos de atuação de suas entradas, que são responsáveis pelo seu aprendizado. Um único neurônio pode ser treinado para aprender um dado padrão, e desta forma, todos os pesos de suas entradas estão ligados a um único padrão, ocorrendo assim a criação de uma memória localizada (*grandmothering*). Um único neurônio pode não ser capaz de memorizar um padrão único, mas este

neurônio associado a vários outros neurônios pode ser capaz. Desta forma, a informação não estará localizada em um único neurônio, estando distribuída entre os diversos neurônios da rede neural. Sendo assim, os pesos de diferentes neurônios são responsáveis pela formação da memória e não se pode dizer que a informação está localizada em um único ponto da rede neural.

Um último tipo de distinção entre diferentes tipos de modelos de redes neurais é feito em relação à capacidade de classificação destes modelos. Os modelos que implementam um classificador linear são modelos onde basicamente se tem apenas um tipo de padrão associado a cada classe e desta forma ocorre uma associação linear de padrões. Um classificador não linear permite que tipos bastante distintos sejam associados a uma mesma classe. Este tipo de classificador será bem discutido posteriormente, uma vez que envolve o problema clássico do reconhecimento de um padrão que represente uma função XOR. Os classificadores lineares não são capazes de aprender uma função XOR, enquanto um classificador não linear terá a capacidade de aprender este tipo de função.

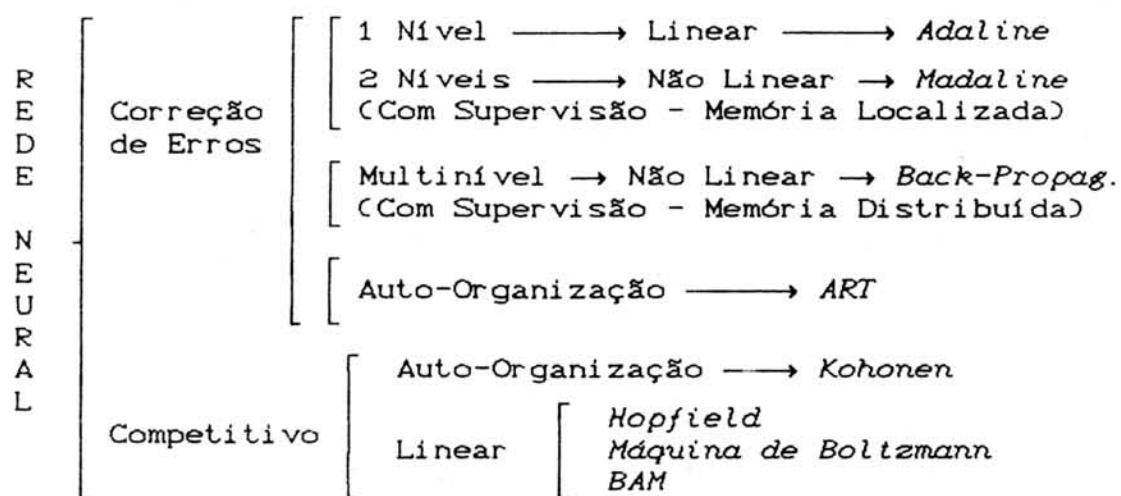


Figura 4.5 - Taxonomia das redes artificiais de neurônios

As categorias citadas acima servem para classificar diferentes tipos de modelos de redes neurais, mas estes modelos estão distribuídos de uma forma irregular nestas categorias. Devido a este fato, o esquema apresentado na figura 4.5 aborda apenas os itens de destaque na classificação de cada um dos modelos de redes neurais, não estando relacionadas todas as categorias.

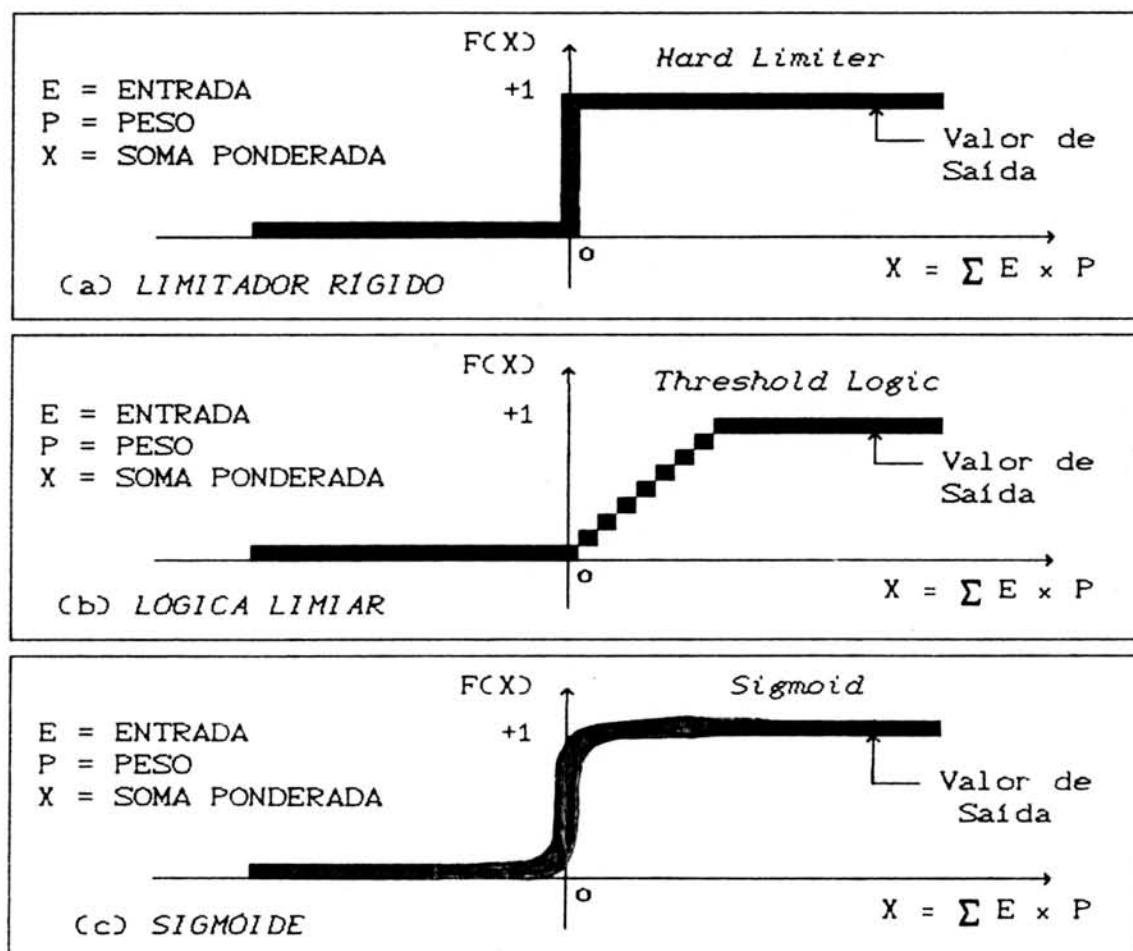


Figura 4.6 - Funções de transferência

4.4.2 Modelos Adaptativos

Os modelos de redes neurais adaptativos são modelos baseados no Perceptron. Estes modelos possuem em geral um aprendizado do tipo supervisionado, onde é aplicado um algoritmo de correção de erros.

4.4.2.1 Perceptron

O modelo básico do Perceptron foi idealizado por Frank Rosenblatt [ROS 59]. Este modelo possui muitas variações, onde será analisado aqui o modelo de apenas um nível, o mais simples possível. A estrutura dos neurônios se apresenta exatamente igual a da figura 4.4, sendo que sua função de transferência típica, é uma função do tipo *hard limiter* - figura 4.6 (a).

Este modelo consiste de uma rede linear de neurônios, onde os valores de entrada alimentam simultaneamente todos os neurônios. O neurônio que, após a sua ativação, possuir o maior valor de saída ou que atingir o limiar para a ativação indicará a resposta correta. Neste modelo cada neurônio é treinado para "responder" (identificar) a um padrão específico.

Será dada uma demonstração de como funciona uma rede de Perceptrons, através de um exemplo prático. Tomando-se um padrão de aprendizado o qual se deseja ensinar à rede, altera-se os pesos de um determinado neurônio que aprenderá este padrão. Os pesos são alterados de modo que, ao ativar a rede com um padrão de entrada similar ao do aprendizado, se obtenha um valor máximo na soma ponderada.

No exemplo da figura 4.7, tem-se uma entrada que é o padrão a ser reconhecido (linha superior), a qual ao ser fornecida a rede é multiplicada pelos pesos de cada um dos neurônios (linhas 1 a 5). Estes pesos refletem um treinamento previamente realizado. Para este padrão de entrada, o conjunto de pesos correspondentes a linha três fará com que a saída deste respectivo neurônio alcance um valor máximo e será portanto aquele que reconheceu o padrão de entrada apresentado. O padrão reconhecido é aquele que se parece mais com o padrão original para o qual este neurônio foi treinado. A função de transferência pode fazer com que apenas um neurônio seja ativado, aceitando um baixo grau de distorção no padrão de teste. Caso a função de transferência

seja menos rígida ou se for utilizada uma função do tipo *threshold logic*, pode ocorrer de mais de um neurônio ter sua saída ativa. Neste caso, é necessário selecionar o maior valor de ativação, onde no final só um neurônio deve se manter ativo. Isto pode ser feito usando-se uma rede tipo MAXNET [LIP 87] que seleciona o neurônio com a maior saída.

Padrão →		-1 +1 -1 +1 +1 +1 -1 -1 +1 -1 +1	Saída = $\sum E.P$
C o n d j e s u n t o	→ 1 2 3 4 5	+1 -1 +1 -1 +1 +1 -1 +1 -1 +1 -1 -1 -1 -1 -1 +1 +1 +1 +1 +1 +1 -1 +1 -1 +1 +1 +1 -1 -1 +1 -1 +1 +1 +1 +1 +1 +1 -1 -1 -1 -1 -1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1	Saída = -5 Saída = -1 Saída = +11 Saída = +1 Saída = +1

Figura 4.7 - Exemplo de uso do Perceptron

Outro exemplo prático de aplicação é o emprego de uma rede de Perceptrons com a saída normalizada [KIN 89], o que facilita bastante a definição dos limiares e até o ajuste dos pesos de atuação pode ser feito de forma bastante simples. Nesta rede, os pesos são determinados da seguinte forma:

NE = Número total de elementos (pixels) do padrão,
 P_{xK} = Peso da entrada x do neurônio K da rede,
 PA_{ix} = Elemento x do padrão i de aprendizado e
 Valores de entrada = -1 (apagado) ou +1 (ligado)

Para o neurônio K que será responsável pelo reconhecimento do padrão i , faz-se:

Se $PA_{ix} = +1$ então $P_{xK} = +1/NE$
 senão $P_{xK} = -1/NE$

O ajuste de pesos descrito acima é realizado uma única vez, através de uma inicialização (não existe o procedimento de adaptação). Com o algoritmo acima o valor máximo de saída de um neurônio será +1, caso o padrão de teste seja exatamente igual ao padrão aprendido; e -1 caso o padrão de teste seja totalmente diferente do padrão para o qual este dado neurônio foi treinado.

O exemplo acima é interessante pois ele possui embutida, diretamente sobre o neurônio, a noção de similaridade de um padrão. Pode-se dizer que a saída da rede deste modelo, indica de uma forma imediata o número de pontos do padrão de teste que são exatamente iguais ao padrão aprendido. Outra vantagem é o fato da faixa de valores de saída ser extremamente bem definida, variando apenas entre -1 e +1, e dos pesos serem sempre iguais a uma constante com uma precisão pré-estabelecida, podendo valer apenas: -1/NE ou +1/NE.

Este modelo de rede tem uma arquitetura extremamente simples, como a da figura 4.3 (a), não correspondendo à organização real dos neurônios do sistema nervoso, o qual possui uma alta interconectividade entre seus componentes. Este modelo é excessivamente simples e possui uma grande limitação: não possui a capacidade de generalização! Logo, estas redes neurais ficarão enormes pois cada neurônio aprende apenas um único padrão.

Partindo-se do modelo básico do Perceptron, pode-se chegar a outros modelos mais completos, que possibilitem um melhor aprendizado. Um destes modelos é o do Adaline.

4.4.2.2 Adaline

Este modelo é uma alteração do Perceptron, onde utiliza-se a Regra Delta de aprendizado (*Delta Rule*), ou como também é conhecido, algoritmo de aprendizado LMS (*Least Mean Square*). O Adaline foi criado por Widrow & Hoff

[WID 62] e seu nome vem da abreviação do nome dado aos elementos processadores : ADaptive LINear Element. A arquitetura é de uma rede de uma única camada disposta linearmente com cada neurônio possuindo uma retro-alimentação de sua própria saída. A figura 4.8 apresenta o esquema do modelo de neurônio utilizado no Adaline.

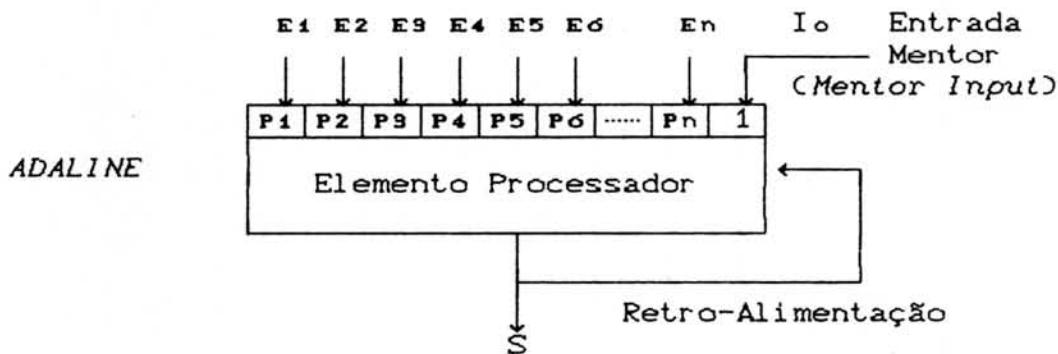


Figura 4.8 - Esquema do neurônio tipo Adaline

O Adaline é um neurônio que utiliza uma regra de aprendizado que permite uma adaptação, fazendo com que este se ajuste a novos padrões. O aprendizado é do tipo supervisionado, por correção de erro, através do uso de uma rede com retro-alimentação de apenas um nível. Este tipo de rede realiza uma classificação linear dos padrões.

O Adaline tem uma entrada denominada de *mentor input*, a qual possui uma ponderação sempre igual a 1. Esta entrada apresenta a verdadeira resposta que deveria ser dada por um determinado neurônio diante de um determinado conjunto de estímulos (padrão de teste).

A Regra Delta altera o vetor de pesos de atuação de forma que, para um determinado conjunto de entrada, a saída se aproxime da resposta desejada, indicada através da *mentor input*. Isto é obtido através da minimização do erro médio quadrático (erro LMS), que com o uso da Regra Delta é feito da maneira mais eficiente possível [CAU 90]. A Regra Delta nada mais é do que a regra de otimização de funções empregando-se o método do gradiente [CAR 90].

A figura 4.9 descreve em detalhes o funcionamento da Regra Delta de aprendizado que é utilizada pelo Adaline. Esta regra é muito importante, sendo empregada inclusive em modelos sucessores do Adaline.

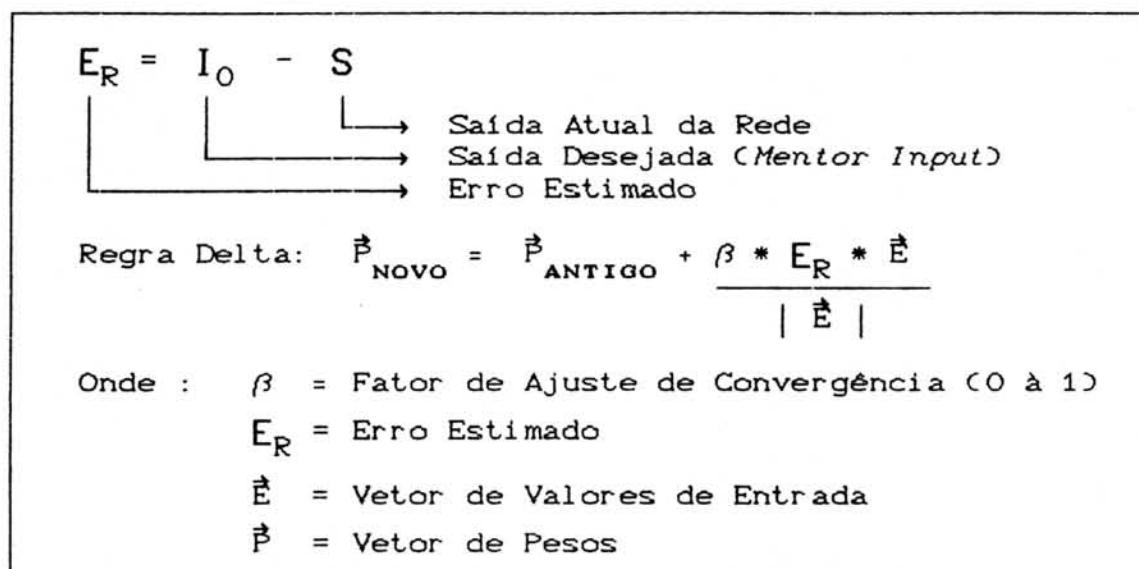


Figura 4.9 - Algoritmo de aprendizado *Delta Rule*

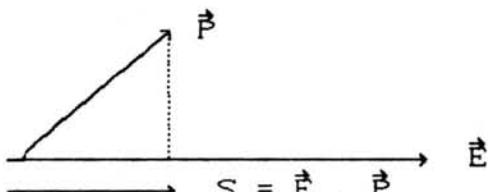
A representação dos valores de entradas e pesos de atuação através de vetores possibilita obter uma interpretação geométrica e matemática mais clara do processo de aprendizado. Abaixo é apresentada esta interpretação para um vetor de duas componentes apenas. Estes conceitos são também válidos para N componentes, mas neste caso se terá um espaço N -dimensional ao invés de um plano.

Neurônio : 2 entradas

Vetor de Entrada : $E_1, E_2 (\vec{E})$

Vetor de Pesos : $P_1, P_2 (\vec{P})$

Soma Ponderada : $S = \vec{E} \cdot \vec{P}$



- A soma ponderada é o produto interno entre os vetores \vec{E} e \vec{P} .

- O produto interno entre dois vetores tem como resultado a projeção de um vetor sobre o outro. Quanto maior for a projeção (valor de saída da soma ponderada), mais próximos estão os dois vetores. Caso os vetores sejam ortogonais, a projeção será igual a 0 (zero).

- O aprendizado deve deslocar o vetor de pesos de forma a aproximá-lo do vetor de entradas que representa o padrão a ser ensinado a um dado neurônio.

- O reconhecimento consiste em identificar qual o vetor de pesos, de um dado neurônio, que está mais próximo do vetor de entrada de teste.

Pode-se considerar também os vetores de pesos como variáveis de uma equação da reta do tipo:

$$S = E_1 \cdot P_1 + E_2 \cdot P_2 \longleftrightarrow C = A \cdot X + B \cdot Y$$

Esta equação se transforma em uma inequação devido ao uso da função de transferência. Desta forma o plano, denominado de espaço de decisão, é dividido em dois semi-planos no caso de equações a duas variáveis, ou em dois semi-hiperplanos para N variáveis.

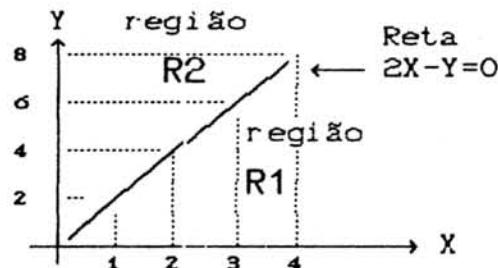
Se for considerado o exemplo dado abaixo, pode-se visualizar claramente os conceitos que foram descritos anteriormente.

$$\begin{array}{l} S \leq P_1 E_1 + P_2 E_2 \rightarrow \text{É uma inequação que divide o plano em duas} \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \text{sub-regiões} \\ C \leq A \cdot X + B \cdot Y \end{array}$$

$A, B \rightarrow$ São os pesos

$X, Y \rightarrow$ São as entradas.

X e Y estão localizados
em uma das duas sub-
regiões formadas



Supondo-se :

$$P_1 = A = 2 \quad \text{Vem} \dots \quad 0 \leq 2 \cdot X - 1 \cdot Y \quad (\text{Região R1})$$

$$P_2 = B = -1 \quad 0 > 2 \cdot X - 1 \cdot Y \quad (\text{Região R2})$$

Observação : o limiar adotado é zero - 0 (hard limiter)

Portanto :

Se a soma ponderada S for: $S \geq 0 \rightarrow$ Padrão na região R1
 $S < 0 \rightarrow$ Padrão na região R2

Na interpretação dada acima, percebe-se que um dado conjunto de pesos de atuação de um neurônio terá como função dividir o espaço de decisão em duas sub-regiões. Cada neurônio se ativará de acordo com a posição do padrão de entrada dentro do espaço de decisão, se ativando ou não, conforme a região onde se encontra este padrão.

Esta divisão do espaço de decisão em duas sub-regiões bem definidas e isoladas é o que determina a característica de classificação linear dos modelos baseados no Adaline. Este tipo de interpretação também serve para que se possa compreender porque o Adaline não é capaz de aprender os padrões que representam a função XOR.

4.4.2.3 Madaline

Uma extensão proposta por Widrow ao modelo do Adaline é a da criação de um novo elemento, chamado de Madaline. O Adaline é capaz de aprender a reconhecer apenas um padrão simples e, a partir deste Adaline, surgiu a idéia de criar um elemento composto, chamado Madaline (Many Adaline). O Madaline apenas agrupa a resposta de dois ou mais Adalines, realizando um "e lógico" (AND), "ou lógico" (OR) ou pela obtenção de maioria (Majority Voter) dos valores obtidos nas saídas de um conjunto de neurônios do tipo Adaline [WID 88]. Com isto, é possível fazer com que

uma saída da rede seja ativada para mais de um tipo de padrão de entrada. Assim, é possível treinar a rede de Madaline para reconhecer, por exemplo, um padrão transladado ou até rotacionado. Para isto, basta treinar cada elemento interno do Madaline para reconhecer o padrão em uma dada posição ou orientação e fazer o "ou" das saídas destes elementos [CAU 90].

Os modelos de um único nível baseados no Perceptron sofreram um grande abalo após a publicação de Minsky [MIN 69] que demonstrou uma grande limitação na sua capacidade de aprendizado.

A pergunta crítica feita por Minsky é: como uma rede de neurônios de um único nível, pode aprender a reconhecer o padrão de uma função lógica "ou exclusivo" (XOR)? O problema das redes de um nível baseadas no Perceptron é que elas dividem o espaço de soluções em duas regiões distintas, mas em uma função XOR não se consegue abranger as duas respostas em uma só região. O diagrama da figura 4.10 apresenta a relação entre diferentes modelos de redes e a formação das regiões de decisão.

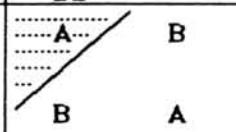
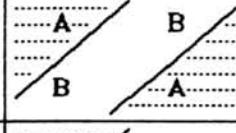
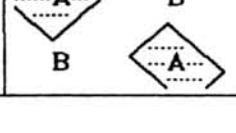
TIPO REDE	TIPOS DE REGIÕES	XOR
1 Nível	Criação de um plano ou hiper-plano de divisão	
2 Níveis	Regiões convexas abertas ou fechadas	
3 Níveis	Regiões arbitrárias de formato livre	

Figura 4.10 - Níveis de uma rede versus regiões de decisão

O modelo anterior, o Adaline, que é uma rede de um nível, não pode resolver o problema de um XOR.

Será dado aqui uma breve demonstração do motivo que levou a procura de uma nova solução, chegando-se então às redes neurais de mais de um nível, como o Madaline. Pode-se comparar uma rede de um nível com um sistema de equações lineares onde, se for considerado um neurônio com duas entradas, obtém-se:

$$\text{Saída} = P_1 \cdot E_1 + P_2 \cdot E_2 \longrightarrow P = \text{Vetor de Pesos}$$

$$E = \text{Vetor de Entradas}$$

Para a função XOR, as equações obtidas serão:

E1	E2	S
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	-1

Equações : $P_1, P_2 = \text{Pesos}$

$$\begin{aligned} -P_1 - P_2 &= -1 \\ -P_1 + P_2 &= 1 \\ P_1 - P_2 &= 1 \\ P_1 + P_2 &= -1 \end{aligned} \quad] \quad \begin{array}{l} \text{SISTEMA} \\ \text{INSOLÚVEL} \end{array}$$

Portanto em uma rede de um único nível não há um conjunto de pesos que possibilite a solução do problema. A idéia é se criar um nível a mais que possibilite a união das respostas do nível superior. A solução do problema do XOR em uma rede de dois níveis é imediata:

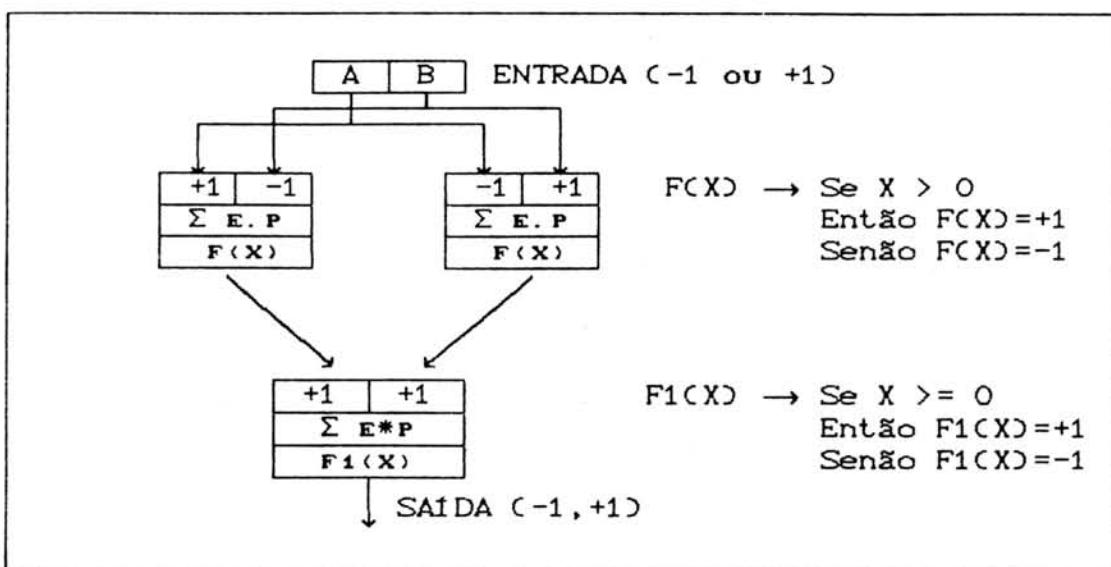


Figura 4.11 - Solução do problema do XOR

Uma rede de mais de um nível poderá aprender inclusive padrões como o da função XOR. O Madaline, que é uma rede neural de dois níveis (como a da figura 4.11), resolve em parte o problema, mas ainda apresenta um inconveniente, o usuário é quem realiza a "junção" dos Adalines.

Até então, não havia um algoritmo que fizesse o aprendizado automaticamente de uma forma completa, faltando uma regra de aprendizado mais genérica para as redes de múltiplos níveis. Além disto, no Adaline e mesmo no Madaline ainda não se tem um conhecimento distribuído na rede, sendo que cada neurônio tem que ser treinado para reconhecer um dado padrão. O Madaline apenas possibilita a junção de dois Adalines de forma a "aparentemente" fazer com que quaisquer dois padrões possam ser identificados como pertencentes a uma mesma classe.

4.4.2.4 Redes Multinível

A técnica de *Back-Propagation* conseguiu dar uma nova vida às redes neurais, pois permite o aprendizado das redes neurais baseadas no Perceptron, com qualquer número de níveis, servindo inclusive para o caso do XOR. Este algoritmo de aprendizado foi desenvolvido por Parker [PAR 85] e Rumelhart [RUM 86].

Uma rede neural que utiliza o modelo de *Back-Propagation* é uma rede com neurônios dispostos em três ou mais níveis: um nível de entrada (*input layer*), um ou mais níveis intermediários ou ocultos (*hidden layers*) e um nível de saída (*output layer*) - vide figura 4.3 (d). A função de transferência utilizada é uma função do tipo função sigmóide (tem a forma de um 'S') e a regra de aprendizado é derivada da Regra Delta, mas aplicável a uma rede multinível. A seguir são apresentadas as funções utilizadas nas redes multinível.

* Ajuste dos Pesos :

Nível de Entrada : Todas as entradas possuem pesos igual a 1.

Nível de Saída : Adaptação dos pesos através da Regra Delta original.

Nível Intermediário :

- Calcula-se o ERRO de cada neurônio do nível intermediário;
- Aplica-se o ERRO (calculado conforme descrição abaixo) na equação principal da Regra Delta;
- Usa-se como função de transferência a função sigmoidal.

* Cálculo do Erro :

$$\text{ERRO}_i = F'(E) \cdot \left\{ \sum P_{ij} \cdot \text{ERRO}_j \right\}$$

→ Onde :

- $F'(E)$ - Derivada da função de ativação para a entrada E
- i - Número do neurônio no nível intermediário
- j - Número do neurônio no nível de saída

* Função de Transferência :

$$F(X) = \frac{1}{(1 + e^{-(x + t)})}$$

$$F'(X) = F(X) \cdot (1 - F(X))$$

→ Onde :

T - Limiar simples da função de transferência
(Constante definida pelo projetista da rede)

X - Valor de entrada da função de transferência

e - Número de Euler (constante)

O aprendizado por *back-propagation* baseia-se na propagação retrógrada do erro para os níveis anteriores da rede, de acordo com o grau de participação que cada neurônio teve no erro do nível posterior. O modelo multinível utilizado normalmente é o composto por três níveis, onde o número de neurônios do nível de entrada e de saída podem ser estimados de acordo com as características do problema. Já o número de neurônios do nível intermediário, assim como o número de níveis intermediários tem que ser estabelecidos de uma maneira intuitiva, não havendo uma técnica precisa para a estimativa destes dados [CAU 80]. Isto se deve ao fato de que neste modelo há uma generalização do conhecimento, onde agora não existe mais um neurônio encarregado de aprender especificamente um padrão e, desta forma, o número de neurônios da rede não pode ser estabelecido de forma exata. Se o nível intermediário tiver um pequeno número de neurônios, a rede pode não ter capacidade de memorização para todos os padrões e, se o nível intermediário tiver muitos neurônios, pode ocorrer o efeito de *grandmothering*, ou seja, a criação de neurônios especializados para reconhecer cada padrão.

As redes neurais do tipo multinível com *back-propagation* são um dos modelos mais estudados e adotados na atualidade pela sua grande capacidade de adaptação a qualquer tipo de padrão. É claro que o custo em termos de implementação é um pouco mais elevado que o Adaline mas, dependendo da aplicação e principalmente do tipo de padrões a serem reconhecidos ou classificados, pode ser vantajosa a utilização deste modelo.

4.4.2.5 ART

O modelo ART ou teoria da ressonância adaptativa (*Adaptive Resonance Theory*) foi desenvolvido por Carpenter e Grossberg [CAR 83]. Este modelo implementa uma rede neural, a qual é treinada sem uma supervisão, formando agrupamentos

(clusters) e podendo ser completamente descrita usando equações diferenciais não lineares [LIP 87].

O modelo ART é basicamente um modelo onde é desenvolvida uma técnica que permite uma evolução da rede, fazendo com que esta possa "crescer" à medida que novos padrões lhe são ensinados. O algoritmo principal seleciona a primeira entrada como um exemplar para o primeiro cluster. A próxima entrada é comparada com este primeiro cluster onde, caso a distância do novo exemplar seja menor que um certo limite, faz-se apenas uma adaptação de forma que o cluster represente ambos os padrões. Caso a distância entre o novo exemplar e o exemplar já armazenado seja superior ao limite estabelecido, é criado um novo cluster que servirá para representar este novo exemplar. Este processo é repetido para todas as entradas seguintes. O número de clusters crescerá com o tempo e dependerá tanto do limite como da métrica usado no cálculo da distância na comparação das entradas com os exemplares armazenados na rede.

O modelo ART possui uma propriedade muito interessante, que é a possibilidade de expansão do número de padrões aprendidos pela rede. Através da criação de novos clusters, este modelo de rede neural pode ir se expandindo à medida que seja necessário acrescentar novos exemplares. É interessante notar que o limite de comparação de distância entre o exemplar de entrada e os exemplares armazenados pode ser fixado de acordo com cada aplicação. Este limite determina a flexibilidade do reconhecimento, ou seja, um valor muito pequeno indica que os exemplares serão muito parecidos para serem colocados juntos, representando um único cluster. Mas, quando o limite é um valor maior, haverá uma maior flexibilização e os exemplares de entrada serão agrupados em um conjunto menor de clusters.

O algoritmo completo do comportamento de um rede do tipo ART é descrito por Lippmann [LIP 87]. A figura 4.12 apresenta um exemplo do comportamento de uma rede do modelo ART para o algoritmo descrito por Lippmann [LIP 87].

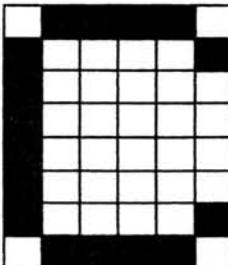
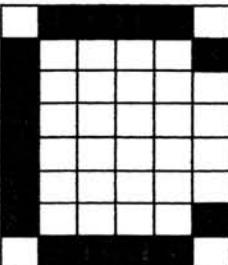
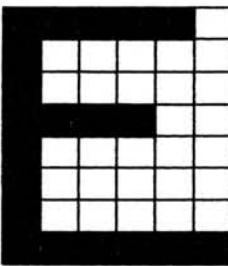
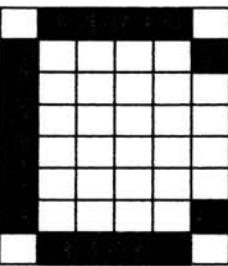
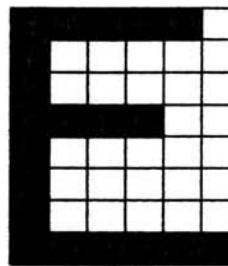
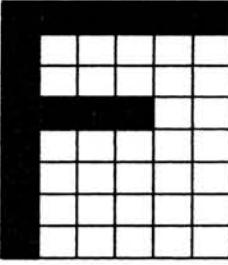
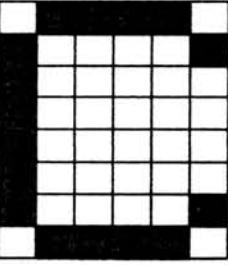
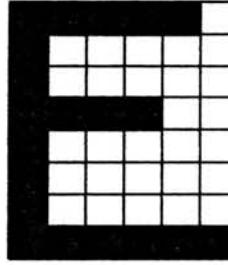
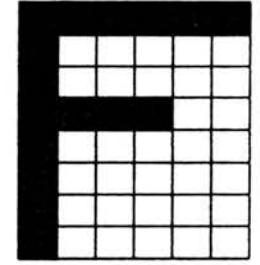
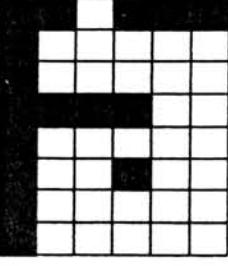
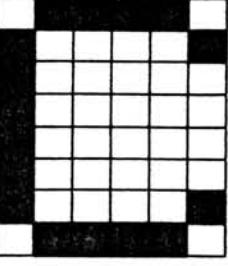
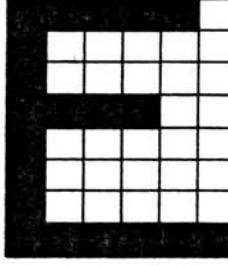
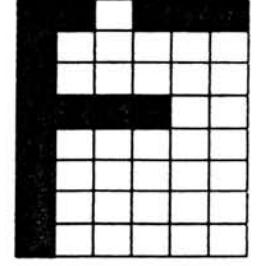
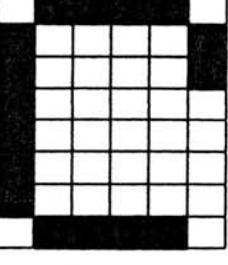
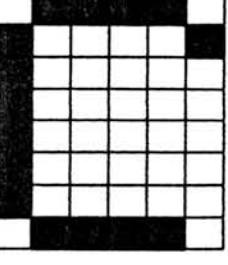
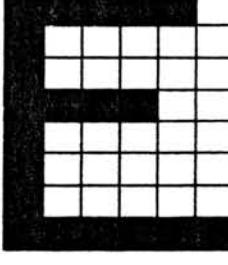
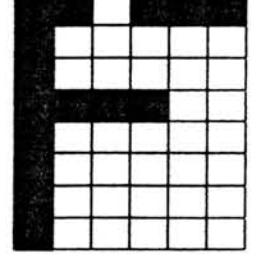
ENTRADA	EXEMPLARES APÓS CADA ENTRADA		
			
			
			
			
			

Figura 4.12 - Funcionamento do modelo ART

O modelo ART sofreu várias evoluções, sendo que existem diversos modelos, denominados de ART-1, ART-2 e ART-3 [WID 90]. Estes modelos no entanto implicam uma sobrecarga de trabalho muito pesada em simulações, sendo considerados mais como elementos de pesquisa do que modelos de aplicação prática [CAU 90].

4.4.3 Modelos Competitivos

Os modelos competitivos baseiam-se na competição entre os neurônios, onde existem modelos com alguma das seguintes características: conexões laterais entre neurônios vizinhos, formação de clusters ou auto-organização. Nesta classe de modelos de rede neurais, encontra-se os modelos de Hopfield, máquina de Boltzmann, BAM e modelo de Kohonen.

4.4.3.1 Modelo de Hopfield

O modelo original de Hopfield [HOP 82] surge utilizando uma rede onde se dá destaque ao comportamento coletivo. O conjunto possui mais importância do que os elementos, estando os neurônios totalmente interconectados entre si. Um neurônio só não conecta a sua saída às suas próprias entradas.

O modelo de Hopfield é utilizado como uma memória associativa ou para resolver problemas de otimização. Este modelo é estudado em Física, onde se reduz ao problema do vidro de spin infinito numa rede de interconexão total [PEN 90]. O modelo de Hopfield é muito empregado na restauração de padrões, sendo que ele trabalha com valores de entrada e saída binários (+1, -1). A figura 4.13 apresenta a topologia de uma rede de Hopfield, a qual pode utilizar a regra de aprendizado somente uma vez para inicializar a rede - figura 4.16. A função de transferência empregada é do tipo *hard limiter*, conforme é apresentada na figura 4.6 (a).

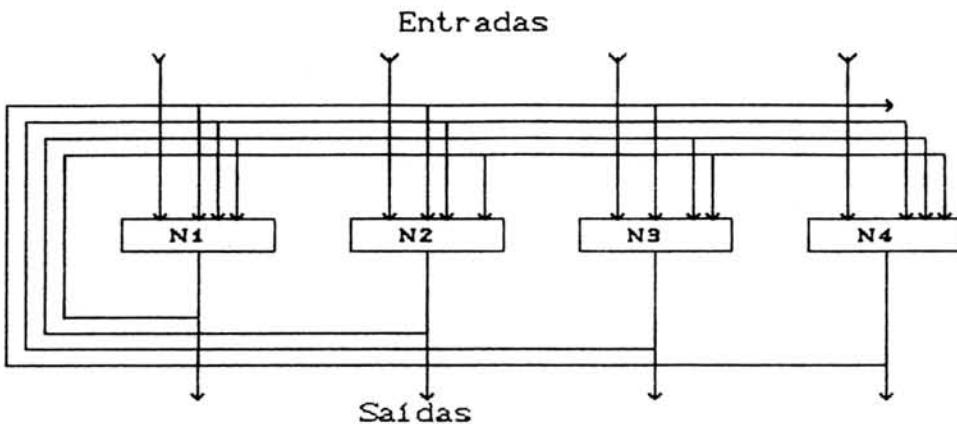


Figura 4.13 - Organização de uma rede de Hopfield

A rede de Hopfield é utilizada para o reconhecimento de padrões como : caracteres, objetos e voz. A figura 4.14 apresenta o exemplo de uso do modelo de Hopfield.

Padrão Original	Padrão de Entrada (Distorcido)	Padrão de Saída (Reconstituído)
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □
□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □

Figura 4.14 - Utilização da rede de Hopfield

O funcionamento do modelo proposto por Hopfield é o seguinte:

- Cada neurônio é responsável por um elemento (pixel) do padrão de entrada, podendo assumir somente dois estados (+1 = Ligado) ou (-1 = Desligado);

- Ao aprender um padrão, o neurônio se comportará de forma que sua saída ajude a ativar algum outro neurônio que faça parte do padrão aprendido e que deveria estar ativo; ou inibir algum outro neurônio que faça parte do padrão e que deveria estar inativo (figura 4.15);

- A rede será ativada e a medida que os neurônios vão interagindo uns com os outros, a saída da rede vai convergindo para um estado estável, que deve ser um dos padrões aprendidos;

- Para adicionar novos padrões (novas memórias - estados estáveis) para os quais a rede deve convergir, basta adicionar os pesos já existentes aos valores novos obtidos através do mesmo cálculo utilizado na inicialização.

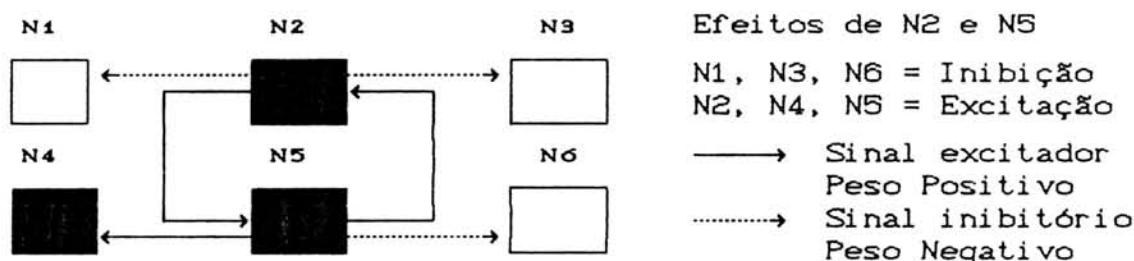


Figura 4.15 - Interação entre os neurônios no modelo de Hopfield

O modelo de Hopfield possui algumas limitações. Uma delas é quanto ao tamanho da rede em termos do número de neurônios, o qual deve respeitar a equação [HOP 82]:

$$\text{Nro. de neurônios} \times 0,15 > \text{Nro. de padrões armazenados}$$

Devido a esta restrição, as redes deste modelo tem um tamanho muito grande. Por exemplo, para armazenar 10 padrões é preciso de mais de 70 neurônios com um total de aproximadamente 5000 conexões (pesos de atuação). Caso esta limitação não seja respeitada, a rede pode convergir para os chamados "estados espúrios", que são estados imperfeitos, uma mistura entre os padrões aprendidos.

Outro problema é o aprendizado de padrões muito similares que também podem levar à obtenção de um estado espúrio ou podem convergir para um padrão incorreto. A solução neste caso é realizar alguns procedimentos de ortogonalização [WAL 86], visando diferenciar mais os padrões.

Além do modelo de Hopfield, existem dois outros modelos que são muito parecidos a este, atuando também como memórias associativas, são eles: a máquina de Boltzmann e o modelo BAM.

Aprendizado/Inicialização
$P_{ij} = \begin{cases} M-1 & p \\ \sum_{p=0}^M E_i E_j & \\ 0 & se i=j \end{cases}$ <p>Onde :</p> <p> P_{ij} = Peso p/entrada i do neurônio j p = Padrões Existem M padrões E_i^p = Elemento de entrada i do padrão p </p>
Reconhecimento
<p>1) $\mu(0)$ = Padrão a ser reconhecido</p> $\mu_i(0) = E_i^{teste} \quad \text{para } 0 \leq i \leq N-1$ <p>2) $\mu_j(t+1) = F \left(\sum_{i=0}^{N-1} P_{ij} \mu_i(t) \right) \quad \text{para } 0 \leq j \leq N-1$</p> <p>3) Repetir o passo 2 até que o padrão de saída $\mu(t)$ fique estável</p> <p>Onde :</p> <p> N = Nro. de elementos do padrão de teste (entradas) $\mu_i(t)$ = Saída do neurônio i no tempo t $t \rightarrow$ Número da iteração $F(x)$ = Função de Transferência </p>

Figura 4.16 - Algoritmos da rede de Hopfield

4.4.3.2 Máquina de Boltzmann

O modelo da máquina de Boltzmann foi desenvolvido por Hinton, Sejnowski e Ackley [HIN 84, HIN 86]. Este modelo é composto por uma rede neural totalmente conectada, que possui uma representação distribuída do conhecimento, sendo formado por uma rede de elementos de processamento estocástico. A máquina de Boltzmann pode ser considerada, de uma maneira simplificada, como uma rede do modelo de Hopfield, que estabelece as soluções por um processo de *simulated annealing* governado pela estatística de Boltzmann [WID 90].

O *simulated annealing* (resfriamento simulado) [KIR 83] é uma técnica de pesquisa que tem sido aplicada a diversos problemas de otimização. Este tipo de mecanismo é empregado, de forma a tentar evitar os chamados estados espúrios do modelo de Hopfield. O princípio para o uso desta técnica é a adição de um componente aleatório ao processo de decisão de cada neurônio, de forma a tentar escapar dos estados espúrios, também denominados de mínimos locais de energia [BAL 89].

Este modelo segue, de uma forma geral, os princípios de funcionamento do modelo de Hopfield. Por este motivo e, devido ao fato deste tipo de rede sofrer problemas de perda de precisão, não serão apresentados maiores detalhes sobre o mesmo. Para maiores informações, sugere-se a leitura de [HIN 84, HIN 85, ACK 83 e BAL 89].

4.4.3.3 BAM

A BAM - memória associativa bidirecional (*Bi-directional Associative Memory*), foi desenvolvida por Kosko [KOS 87], baseada em algumas idéias de Hopfield e Grossberg. Esta rede serve como uma memória associativa capaz de associar pares de padrões. Ao ser apresentado um dos componentes de algum destes pares de padrões, a rede gerará automaticamente na saída a representação do outro padrão associado ao padrão apresentado.

A rede BAM, assim como a rede de Hopfield, se utiliza de uma técnica de minimização de energia. A diferença é que, no modelo de Hopfield, um padrão é associado a ele mesmo e, portanto, na saída, o padrão é apresentado apenas reconstituído. Na BAM, um padrão pode ser associado a um outro padrão e, desta forma, ao apresentar algum destes dois padrões associados, mesmo distorcidos, a rede apresentará na saída o padrão associado ao apresentado na entrada da rede.

O modelo BAM associa pares de padrões (A_i, B_i) , onde $A_i \in \{-1, +1\}^N$ e $B_i \in \{-1, +1\}^P$. Para estes padrões A e B, compostos por vetores de dimensões N e P respectivamente, será necessária uma matriz de pesos M de dimensões N x P para codificar tais padrões.

A codificação, que é a forma como se denomina o aprendizado em uma rede BAM, é realizada simplesmente através do somatório das matrizes de correlação de cada um dos pares de padrões [BLU 90]. Desta forma, a matriz M que codifica K pares de padrões é simplesmente:

$$M = \sum_{i=1}^K A_i^T B_i$$

K = Nro. de pares de padrões
 M = Matriz de pesos
 A_i^T = Transposta do vetor A de padrões
 B_i = Vetor B de padrões

O reconhecimento é denominado de *recall* ou decodificação. Ele consiste apenas da multiplicação do vetor de teste (padrão de teste - codificado em valores +1, -1) pela matriz M, seguida da aplicação de uma função de transferência, resultando um outro vetor. Este vetor resultante será o padrão associado ao vetor de teste. A rede é dita bidirecional, pois $A.M$ resultará no padrão B, assim como $B.M^T$ resultará no padrão A. Outra característica interessante deste modelo é a possibilidade de retirar um par de padrões da memória, "esquecendo-os". A figura 4.17 apresenta os processos de codificação e decodificação.

A BAM, assim como o modelo de Hopfield, também possui problemas de aprendizado em relação ao número máximo de padrões aprendidos e de reconhecimento incorreto de padrões similares (necessidade de ortogonalização da entradas). O limite de memorização em uma BAM será da ordem de M onde M é o menor valor entre as dimensões N e P dos pares de padrões codificados na rede [KOS 87a].

Codificar:

$$X_1 = (101010) \text{ e seu par } Y_1 = (1100)$$

$$X_2 = (111000) \text{ e seu par } Y_2 = (1010)$$

Recodificação dos Padrões: $(0,1 \rightarrow -1,+1)$

$$A_1 = (1 \ -1 \ 1 \ -1 \ 1 \ -1) \quad B_1 = (1 \ 1 \ -1 \ -1)$$

$$A_2 = (1 \ 1 \ 1 \ -1 \ -1 \ -1) \quad B_2 = (1 \ -1 \ 1 \ -1)$$

Matrizes:

$$A_1^T B_1 = \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

$$A_2^T B_2 = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix}$$

Função de Transferência:

$$F(X) = -1 \text{ se } X < 0$$

$$F(X) = 1 \text{ se } X \geq 0$$

$$A_1 \cdot M = (8 \ 4 \ -4 \ -8) \rightarrow F(X) \rightarrow (1 \ 1 \ -1 \ -1)$$

$$A_2 \cdot M = (8 \ -4 \ 4 \ -8) \rightarrow F(X) \rightarrow (1 \ -1 \ 1 \ -1)$$

$$B_1^T \cdot M^T = (4 \ -4 \ 4 \ -4 \ 4 \ -4) \rightarrow (1 \ -1 \ 1 \ -1 \ 1 \ -1)$$

$$B_2^T \cdot M^T = (4 \ 4 \ 4 \ -4 \ -4 \ -4) \rightarrow (1 \ 1 \ 1 \ -1 \ -1 \ -1)$$

Figura 4.17 - Processo de codificação em uma BAM

Uma solução para problemas de limitação de aprendizado em uma BAM, é a criação dos sistemas BAM [BLU 90].

Um sistema BAM é composto por várias redes BAM, mas a sua grande vantagem é a possibilidade, ao se determinar que uma dada rede não consegue aprender um novo padrão, de codificar-se este padrão em outra rede. Para determinar se a rede pode ou não aprender um padrão, basta codificá-lo na matriz M_e , em seguida, testar para ver se a rede responde corretamente à apresentação deste padrão. Caso a rede não se comporte da maneira esperada, o padrão é removido da rede e codificado em uma outra rede, que tenha a capacidade de armazená-lo corretamente.

4.4.3.4 Modelo de Kohonen

Este modelo recebe o nome de seu criador Teuvo Kohonen [KOH 84]. Sendo um modelo baseado no auto-aprendizado, ou seja, a rede neural do modelo de Kohonen possui a capacidade de auto-organização. Este modelo também é conhecido pela criação dos mapas de atributos de auto-organização (*Self-Organization Feature Maps*). A rede neural utilizada no modelo básico de Kohonen é uma rede de um único nível, onde os neurônios possuem conexões com as entradas e conexões laterais com outros neurônios, até um certo grau de vizinhança pré-estabelecido. As entradas externas estão conectadas em paralelo a todos os neurônios simultaneamente. A figura 4.18 apresenta um esquema de organização do modelo de Kohonen para neurônios conectados apenas aos seus oito vizinhos mais imediatos.

A propriedade de auto-aprendizado é muito interessante, pois reproduz o aprendizado humano de certas funções, de uma maneira muito natural. Considerando-se um recém-nascido com exemplo, encontra-se um auto-aprendizado na função de foco dos olhos, a qual é adquirida de forma automática, sem a necessidade de uma "supervisão do aprendizado". Outra característica muito interessante do modelo de Kohonen é a formação de regiões da rede que tem seus neurônios ativados quando determinado padrão é apresentado.

Cada padrão provocará a ativação de uma região determinada da rede, onde a identificação de um dado padrão se dá através da determinação de qual área de rede foi ativada. Estudos neurofisiológicos demonstram que o cérebro humano atua de uma forma similar a esta, onde regiões diferentes do cérebro são ativadas para cada tipo de estímulo externo recebido.

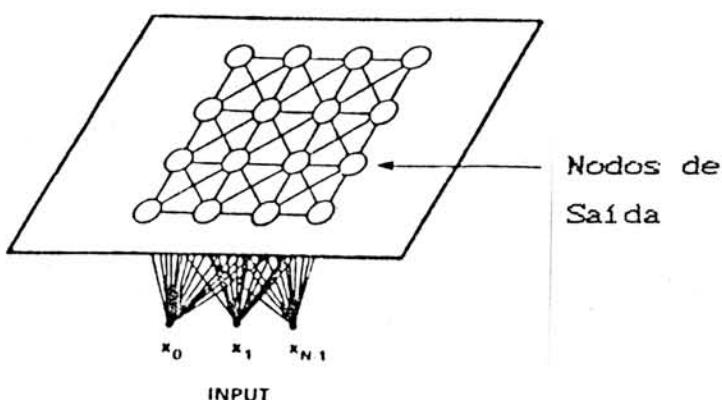


Figura 4.18 - Organização de uma rede neural de Kohonen

Esse fenômeno torna a rede um mapa de classificação, ou seja, para um determinado vetor de entrada, uma determinada região do mapa bidimensional será ativada. Além disso, os pesos serão organizados de forma que neurônios topologicamente próximos sejam sensíveis a entradas fisicamente similares. A figura 4.19 apresenta um exemplo de saída de ativação de uma rede de Kohonen, onde os pontos correspondem aos neurônios e seus tamanhos correspondem à sua atividade (valor de ativação na saída). Na figura da direita é apresentado o resultado da lenta e gradual mudança das entradas, e o movimento do aglomerado é indicado pela sua "cauda".

O tamanho do aglomerado também pode ser ajustado através da determinação do grau de interconexões de um neurônio, em relação aos seus vizinhos. Pode-se dizer então que o raio do aglomerado é proporcional à extensão da vizinhança dos neurônios. Também foi descoberto

empiricamente que, para bons resultados de auto-organização da rede neural, o raio do aglomerado deve decrescer monotonicamente com o decorrer do tempo. A figura 4.20 demonstra exemplos de vizinhança possíveis de serem adotados, bem como a possível evolução da rede em termos de grau de vizinhança das conexões.

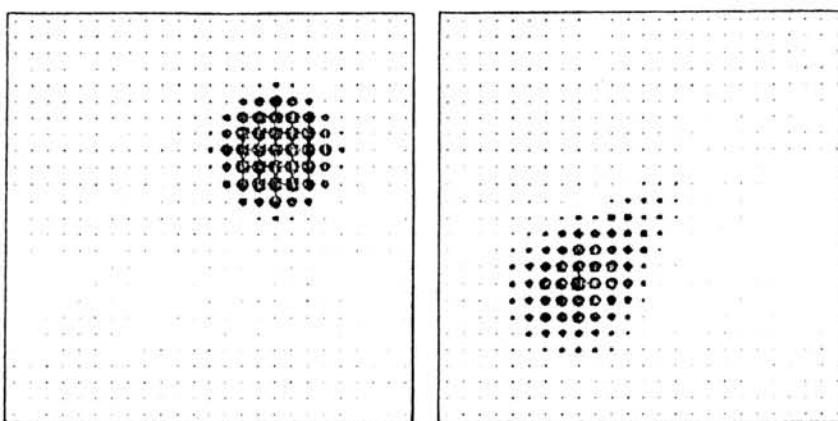


Figura 4.19 - Agrupamento da atividade das saídas de uma matriz bidimensional de neurônios

O algoritmo de ajuste dos pesos e adaptação da rede (algoritmo de *clustering*) pode ser encontrado na literatura [LIP 87, BAL 89 e KOH 84]. A rede de Kohonen constitui-se uma interessante ferramenta para reconhecimento de padrões, sendo inclusive muito empregada em sistemas de reconhecimento de fala, como o citado por Kohonen [KOH 88]. Uma única desvantagem da aplicação deste tipo de modelo no reconhecimento de padrões é a necessidade de identificação das regiões de ativação, ou seja, existe a necessidade de posteriormente associar cada região de ativação a um dado padrão.

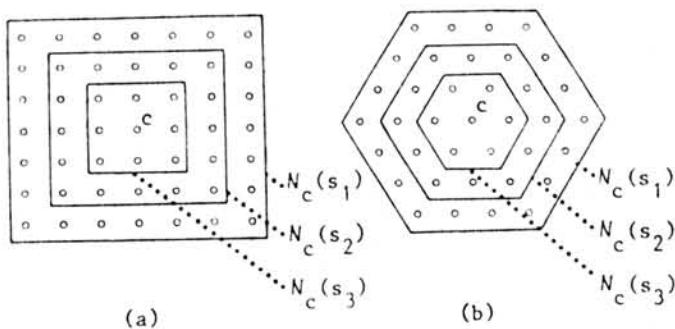


Figura 4.20 - Exemplos de topologias de vizinhança

4.5 Aplicações das Redes Neurais Junto aos Sistemas OCR

As redes neurais tem sido muito empregadas junto ao reconhecimento de padrões e principalmente em sistemas OCR. Uma vez que a entrada de uma rede já se constitui de um padrão, do tipo de um *bit-map*, e sua saída representa a identificação, classificação ou associação de um padrão a outro, logo, é natural que as redes neurais sejam utilizadas em sistemas OCR. Em quase todos os artigos e exemplos de aplicações sobre as redes neurais encontra-se sempre uma demonstração de seu comportamento junto ao reconhecimento de caracteres.

Outro motivo que leva a sua utilização junto aos sistemas de OCR é devido ao maciço paralelismo embutido nas redes neurais. Como os sistemas de OCR são sistemas que precisam processar muitas informações, é interessante que seja buscada uma solução que permita a obtenção de um desempenho satisfatório (sistema eficiente).

Além destes motivos, a principal característica inerente às redes neurais que as tornam especiais é o fato destas poderem sofrer uma adaptação. A adaptação e o aprendizado são fortes razões para que se implemente um sistema de OCR através da utilização de redes neurais. Como já foi discutido no capítulo 3, esta capacidade de adaptação/aprendizado é extremamente desejável em sistemas OCR, permitindo a criação de sistemas mais flexíveis.

Por último, as redes neurais são uma tentativa de recriar funcionalmente o comportamento do cérebro humano. Se o objetivo final de um sistema OCR é o de reconhecer caracteres e textos, de uma forma a imitar o ser humano em sua tarefa de leitura, nada mais óbvio que as redes neurais sejam utilizadas como uma forma de recriar estes mecanismos que permitem ao homem realizar este tipo de tarefas.

Devido a estes fatores e, inserido no contexto que foi apresentado nos capítulos 2, 3 e 4, deste trabalho, resultou o desenvolvimento de um sistema de OCR, o sistema N²OCR, que se utiliza de um simulador de uma rede neural para executar a tarefa de reconhecimento. O capítulo 5 descreverá a proposta do sistema N²OCR, a implementação de um protótipo deste sistema, uma análise dos modelos de rede neural passíveis de utilização e o modelo que foi adotado.

5 SISTEMA N²OCR

Este capítulo apresentará a proposta de um sistema completo de reconhecimento de caracteres, denominado de sistema N²OCR - Neural Network for Optical Character Recognition [OSO 90a]. O sistema N²OCR é orientado ao reconhecimento de caracteres impressos de múltiplas fontes, possibilitando o reconhecimento de documentos datilografados, impressos por computador ou textos de livros e revistas. O sistema N²OCR, conforme será apresentado posteriormente, também prevê uma extensão que possibilitará o reconhecimento de caracteres manuscritos em letras de imprensa (escrita não cursiva).

Baseado nesta proposta de implementação de um sistema completo de reconhecimento de caracteres, foi implementado um protótipo do mesmo, abrangendo as principais funções necessárias à validação das técnicas e algoritmos utilizados junto ao sistema N²OCR. Deixou-se para um trabalho posterior a implementação das demais funções que compõem o sistema N²OCR.

As funções, que foram implementadas no protótipo, permitem a utilização deste para o reconhecimento de caracteres, mas de uma forma bem simplificada e sem a flexibilidade necessária para um bom desempenho em sistemas comerciais. Algumas rotinas que permitiriam uma maior flexibilidade ao sistema, assim como rotinas que possibilissem ao mesmo sua utilização para o processamento de grandes volumes de dados (aumento do automatismo de suas tarefas) não foram implementadas completamente. Desta maneira o protótipo do sistema, como diz o nome, é apenas um protótipo de testes, que permite a avaliação geral da proposta deste trabalho. Logo, o protótipo do N²OCR não é uma ferramenta desenvolvida para fins de uso corrente, e sim para validação da proposta desenvolvida.

A tabela 5.1 apresenta uma lista de funções típicas de sistemas OCR, onde são assinaladas aquelas funções estudadas e previstas no sistema N²OCR e as funções que foram implementadas no protótipo.

Tabela 5.1 - Funções previstas no sistema N²OCR e funções implementadas no protótipo

« Funções »	N ² OCR	Protótipo
Aquisição de imagens por: câmera de vídeo ou scanner	●	* ¹
Aquisição de textos através de mesa digitalizadora	X	X
Geração de imagens para teste com distorção controlada	●	●
Armazenamento da imagem: (lé e escr.) formatos CUT, BTM, FIM e IMF	●	●
Armazenamento da imagem: (lé e escr.) formatos TIFF, GIF, PCX e pelo LZW	●	X
Manipulação e visualização de imagens P&B, halftoning, tons de cinza	●	●
Manipulação de imagens coloridas: componentes RGB	X	X
Manipulação de imagens: edição, recorte, junção e justaposição	●	* ¹
Processamento de Imagens: histograma, limiarização, binarização, convolução	●	●
Processamento de Imagens: operações no domínio freqüência, transformadas	X	X
Processamento de Imagens: realce, restauração, filtragem e vetorização	●	X
Automação do processamento com configuração de tarefas (uso off-line)	●	X
Manipulação de imagens de tamanho superior a capacidade de memória	●	* ²

Tabela 5.1 - Funções previstas no sistema N²OCR e funções implementadas no protótipo (Continuação)

« Funções »	N ² OCR	Protótipo
Reconhecimento de caracteres isolados de múltiplas fontes impresso → manuscrito (letra de imprensa) →	●	●
	●	* ³
Reconhecimento de caracteres em letra cursiva ou com escrita on-line	X	X
Reconhecimento de caracteres em tempo real (uso de paralelismo)	●	X
Reconhecimento de caracteres que se tocam (touching characters)	●	* ²
Conjunto de caracteres sem limitações OCR ANSI, alfabeto latino, grego, etc	●	●
Resultado do reconhecimento: ASCII, BRASCI, EBCDIC, outras codificações	●	●
Sem restrições quanto a escrita: tamanho, formulários especiais, estilo	●	●
Pré-processamento: separação de textos e gráficos	●	* ¹
Pré-processamento: isolar caracteres espaçamento variável (por contorno) → espaçamento fixo (pelo tamanho) →	●	●
Pré-processamento: ajuste de escala e posição dos caracteres	●	●
Pré-processamento: extração de novos atributos e recodificação dos padrões	●	X
Pós-processamento: geração de lista de candidatos para análise de contexto	●	●
Pós-processamento: reconhecimento com uso de contexto na correção de erros	●	X
Capacidade de avaliação do reconhecimento: grau de acerto, erro, não id.	●	●
Sist. de reconhecimento com capacidade de aprendizado tutoriado interativo	●	●

Tabela 5.1 - Funções previstas no sistema N²OCR e funções implementadas no protótipo (Continuação)

« Funções »	N ² OCR	Protótipo
Sistema de reconhecimento com capacidade de adaptação do aprendizado	•	•
Sistema de reconhecimento com capacidade de emprego de paralelismo no processamento	•	X
Rede neural: reconhecimento do XOR, expansível, aprendizado supervisionado por correção de erros	•	•

Notação:

• - Função prevista (N²OCR) ou implementada (protótipo)

X - Função não prevista e não implementada

*¹ - Possível através da utilização de programa externo (uso de ferramenta que não faz parte do sistema)

*² - Possível de ser realizado, mas de maneira "restrita"

*³ - Não foi testada/avaliada esta possibilidade de uso

O problema da falta de flexibilidade encontra-se, por exemplo, no algoritmo de segmentação de caracteres, que possui implementado no protótipo apenas a segmentação de caracteres com espaçamento variável por varredura do contorno. Este algoritmo funciona bem com textos datilografados ou impressos (se for no computador, através de impressora laser) de espaçamento variável. O mesmo algoritmo não tem um bom desempenho com textos impressos através de uma impressora matricial, e para este caso é necessária a implementação de um novo algoritmo de segmentação para caracteres com espaçamento fixo.

As seções seguintes deste capítulo descreverão o sistema N²OCR em termos de suas características gerais, e a implementação de seu protótipo.

5.1 Organização Geral do Sistema

O sistema N²OCR é composto por vários módulos que implementam as funções da aquisição das imagens até a saída com os caracteres reconhecidos. Os principais módulos que compõem o sistema são:

- Módulo de aquisição e geração de imagens;
- Módulo de tratamento das imagens;
- Módulo de preparação dos dados para a rede neural e
- Módulo simulador da rede neural.

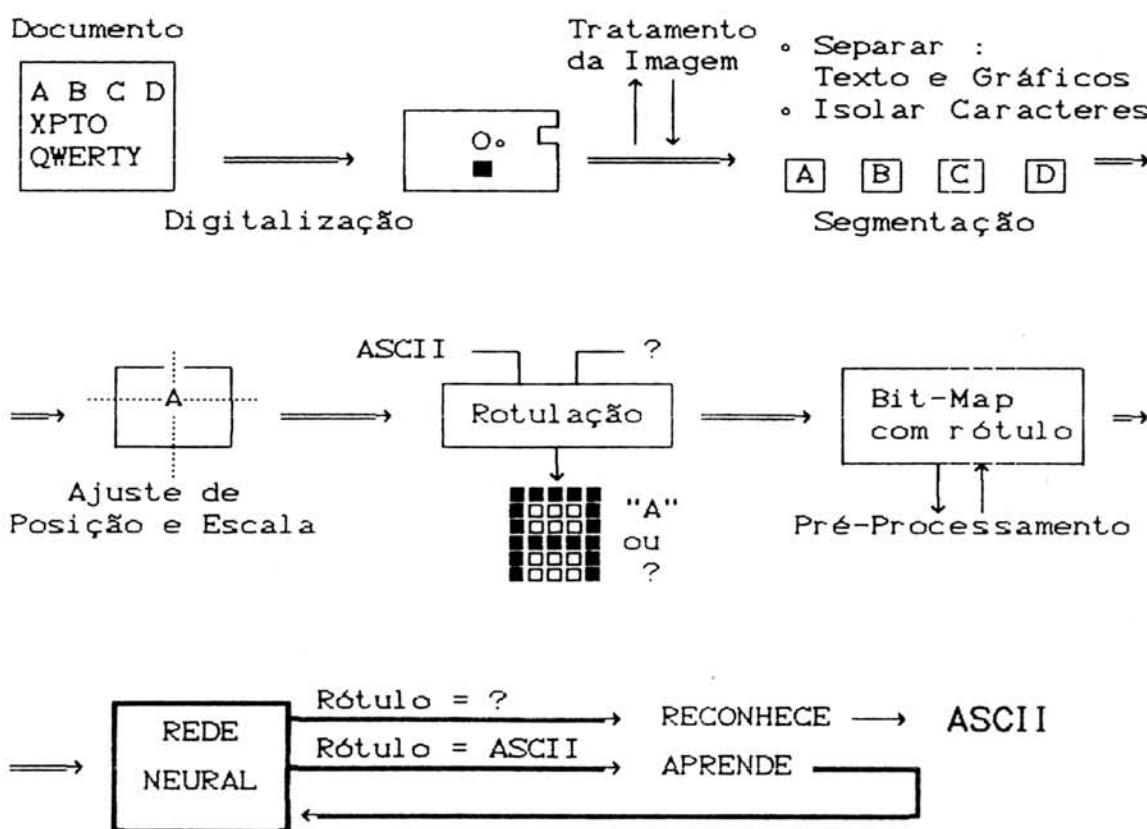


Figura 5.1 - Esquema de funcionamento do sistema N²OCR

A figura 5.1 apresenta um esquema geral da organização do sistema N^2 OCR, onde estão representadas as diversas etapas de processamento realizada pelo sistema através de cada um destes módulos que o compõem. Nas próximas seções serão abordados em detalhes cada módulo que forma o sistema. Na figura 5.2 são apresentados os diferentes módulos que foram implementados no protótipo do sistema N^2 OCR, juntamente com as funções disponíveis de cada módulo e a interação entre estes.

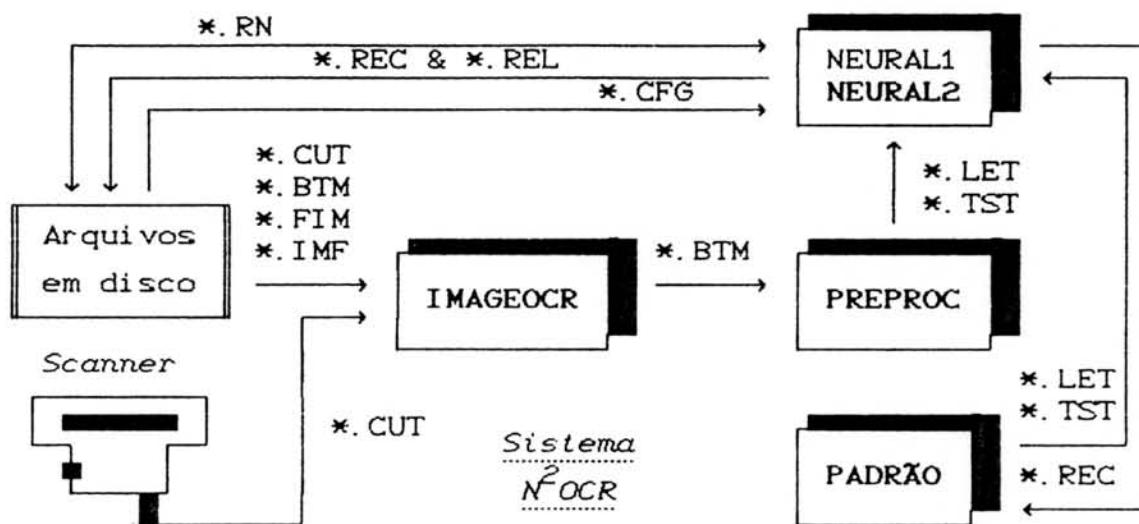


Figura 5.2 - Protótipo do sistema N^2 OCR

Os módulos que compõem o protótipo do sistema N^2 OCR e que aparecem representados na figura 5.2 são os seguintes:

- Padrão : (padrão.exe)
 - Definição de parâmetros de geração e distorção de imagens textuais;
 - Geração de padrões: com ou sem distorção, para aprendizado ou reconhecimento (gera rótulos);
 - Visualização do arquivo de padrões;
 - Avaliação do reconhecimento.

- Imageocr : (imageocr.exe)
 - Visualização de imagens (P&B até 256 níveis);
 - Conversão de formatos de arquivos de imagens;
 - Recorte de imagens;
 - Geração de histograma;
 - Alteração do histograma: compressão e expansão da escala de cinza, limiarização e binarização;
 - Realização de operações de convolução (3x3).
- Preproc : (preproc.exe)
 - Visualização de imagens e arquivo de padrões;
 - Separação dos caracteres;
 - Rotulação dos padrões;
 - Ajuste de posição e escala;
 - Extração de atributos (não implementado).
- Neural2 : (neural2.exe)
 - Rede neural → modelo ADAnLIENE;
 - Configuração do simulador (arquivos *.CFG);
 - Aprendizado de padrões (arquivos *.LET);
 - Reconhecimento de padrões (arquivos *.TST);
 - Salvamento e recuperação de redes neurais já treinadas (arquivos com pesos da rede *.RND);
 - Geração de relatório de acompanhamento da simulação (arquivos *.REL);
 - Geração do texto reconhecido, incluindo ou não os candidatos alternativos para pós-processamento (arquivos *.REC).
- Neural1 : (neural1.exe)
 - Rede neural → modelo ADALINE
 - Similar ao neural2 (1^a versão do simulador, sem a capacidade de expandir a rede);

5.2 Aquisição de Imagens

A aquisição de imagens pode ser realizada através de diferentes dispositivos, como câmeras de vídeo e scanners, conforme foi discutido no capítulo 2. O protótipo do sistema N²OCR permite a utilização de imagens obtidas através de algum destes dispositivos, podendo manipular apenas imagens monocromáticas, mas com um número variável de tonalidades. O módulo de aquisição de imagens baseia-se na utilização de um *handy scanner*, o DFI HS-3000 [FAL 88]. A captura e manipulação de imagens (recorte, junção, edição das imagens) é feita utilizando-se um programa de editoração eletrônica, o *HaloDPE*, fornecido pelo fabricante juntamente com o equipamento.

Devido ao uso de ferramentas prontas, a forma de acesso às imagens é através de arquivos gravados por estes programas. Foi desenvolvida uma série de programas para conversão de arquivos para diferentes formatos, sendo alguns destes utilizados no sistema implementado. O formato dos arquivos utilizados junto ao sistema N²OCR e ao HaloDPE estão descritos no anexo A.1 e A.2.

Para tornar o sistema N²OCR viável, é necessário que se desenvolva uma ferramenta própria para a captura de imagens, pois a manipulação através de arquivos gravados em disco torna o processo muito demorado. Portanto, um módulo, não implementado no protótipo, a ser incorporado ao sistema é um módulo que realizasse as tarefas de aquisição das imagens e sua manipulação diretamente através do sistema N²OCR, sem a necessidade de entrar e sair de aplicações, e onde os dados seriam manipulados preferencialmente sempre na memória, agilizando o processo de reconhecimento.

O sistema N²OCR, além de permitir a captura direta de imagens, também deve permitir o acesso a arquivos de imagens gerados por outros sistemas. Desta forma o N²OCR pode operar acoplado a um sistema específico de aquisição de imagens, mas também pode aceitar o processamento de imagens obtidas por outros dispositivos.

Figura 5.3 - Tela de ajuda do módulo gerador de imagens

Sistema N2OCR - Neural Network Optical Character Recognizer
PADRAO - Geracao de padroes de caracteres para teste

Autor : Fernando Osorio - Data : Abril 1991 - Versao 1.0

1. Parametros
2. LimpaGraf
3. Configura
4. Relatorio
5. GeraFaixa
6. GeraSelec
7. GeraArq
8. Perturba
- 9,10 e 11. Diretorio, Os Shell e Fim

TELA DE AJUDA DO SISTEMA. Entre numero da opcao :



Figura 5.4 - Configuração do módulo gerador de imagens

Configuracao Atual

Formato do arquivo - 0

Tamanho dos Textos - 1

Escala dos Textos - Mult X = 1 Mult Y = 1
Div X = 1 Div Y = 1

Fonte do Texto - Default

Perturbacao - Deslocamento em X = 0

Deslocamento em Y = 0

Deformacao em X = 0.00

Deformacao em Y = 0.00

Angulo de Rotacao = 0

Ruido aleatorio = 0%



A obtenção de imagens para uso junto ao sistema N²OCR foi implementada de duas formas: através do acesso a arquivos de imagens gerados pelo scanner HS-3000, ou através de um programa que permite a geração de um bit-map, simulando a captura de um texto através de um dispositivo como os scanners.

O módulo gerador de imagens (programa padrão.exe - fig. 5.3) é um programa que realiza a função inversa de um sistema de reconhecimento de caracteres, ou seja, dado um texto qualquer, o programa gerará um bit-map com a imagem do texto fornecido. Foi implementado um gerador de imagens textuais que permite a configuração de alguns parâmetros, apresentados na figura 5.4, tais como:

- Seleção de diferentes fontes (tipos de caracteres);
- Seleção do tamanho dos bit-maps gerados para cada caractere e
- Geração simulada de perturbações no texto: inversão aleatória de pontos (figs. 5.5, 5.6 e 5.7), rotação (fig. 5.8) ou translação dos caracteres gerados.

O objetivo deste programa de geração de imagens textuais foi o de possibilitar uma análise do sistema de reconhecimento, independentemente da possibilidade de se utilizar ou não um sistema de aquisição real de imagens. Como o dispositivo de aquisição de imagens só se tornou disponível em uma fase mais adiantada da implementação do sistema, tal programa foi de destaque importância na implementação do N²OCR. Outra característica muito importante deste programa de geração de imagens é a possibilidade de um controle exato dos efeitos de distorção sobre o texto, permitindo desta forma uma melhor análise e acompanhamento do desenvolvimento do sistema de reconhecimento de caracteres, como demonstram as figuras 5.5 a 5.8.

```
$16 16 2 0
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0011111111110000
0011111111110000
0000000000111100
0000000000111100
0011111111111100
0011111111111100
0011111111111100
1111000000111100
1111000000111100
0011111111111100
0011111111111100
0000000000000000
0000000000000000
97
1111000000000000
1111000000000000
1111000000000000
```

MOSTRA - Tecle algo para prosseguir ...

ab.let



Figura 5.5 - Padrão do caractere 'a' gerado sem distorção

\$16 16 2 0

```

1000010010000000
0000000000000000
0000000100000000
0000010000000000
0011111111110000
0011111111110000
0001100000111100
000000000111100
001111111111100
001111111111100
1011000000111100
111100000011100
0001111111111100
0011111111111100
000000000101000
0000000000000000
97
1111000000000000
1111000000000000
1111000000100000

```

MOSTRA - Tecle algo para prosseguir ...

ab.let

The screenshot shows a software window titled 'Sist N2OCR' version 1.0 by F.Osorio91. The menu bar at the top includes 'PADRAO', 'Parametros', 'LimpaGraf', 'Configura', 'Relatorio', 'RELAT', 'Avalia', 'Conta', 'Mostra' (which is highlighted), and 'Edita'. The status bar at the bottom displays the text 'Sist N2OCR Versao 1.0 F.Osorio91'.

Figura 5.6 - Padrão do caractere 'a' gerado com a inversão aleatória de 5% dos pixels

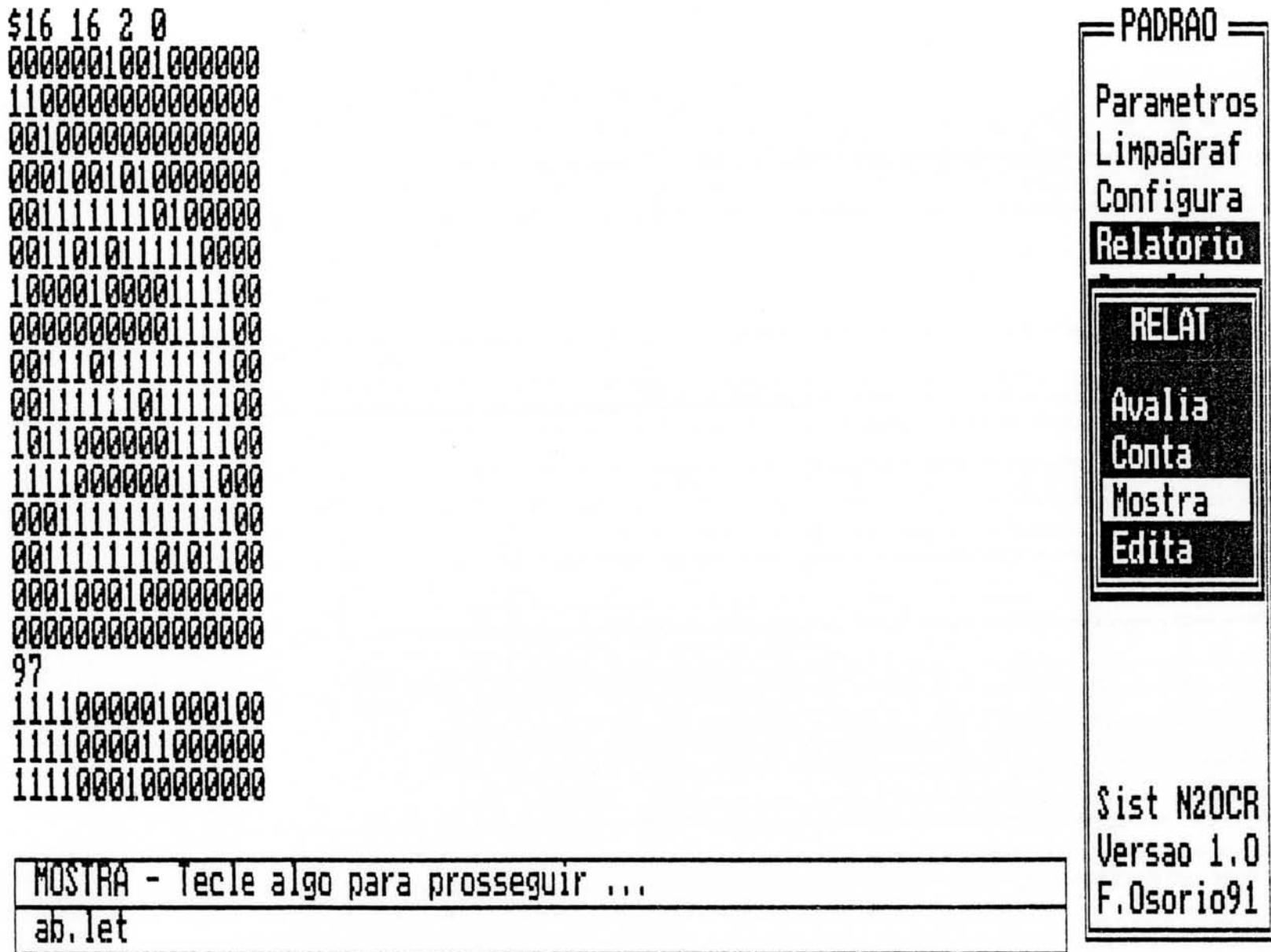


Figura 5.7 – Padrão do caractere 'a' gerado com a inversão aleatória de 10% dos pixels

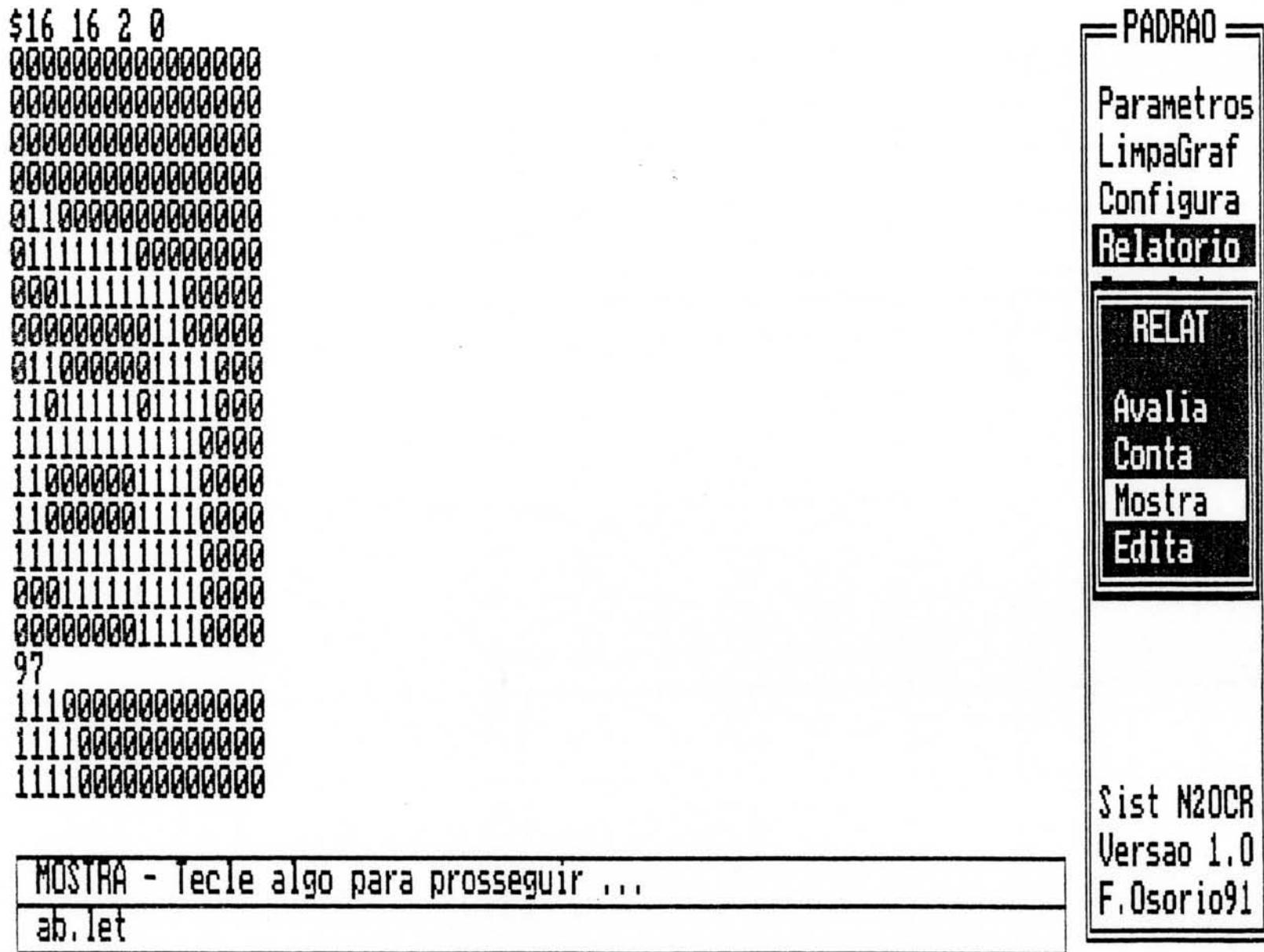


Figura 5.8 – Padrão do caractere 'a' gerado com rotação de 10 graus

5.3 Tratamento da Imagem

O sistema N²OCR possui um módulo dedicado ao processamento das imagens de documentos digitalizados (programa imageocr.exe - fig. 5.9). Este módulo implementa algumas das funções descritas no capítulo 2, tais como:

- Realização de conversões de diferentes tipos de formatos de armazenamento de imagens em arquivos (fig. 5.10);
- Geração de uma imagem com uma escala de tonalidades reais, a partir de uma imagem com halftoning (desfazer o halftoning);
- Manipulação da imagem, possibilitando o recorte de uma área específica para "salvamento" em disco;
- Visualização de imagens monocromáticas de 2 até 256 níveis de intensidade (figs. 5.11 e 5.12);
- Geração do histograma de uma imagem (fig. 5.13);
- Alteração do histograma da imagem: limiarização, binarização (fig. 5.14) e equalização, compressão ou expansão da escala de cinza;
- Realização de operações de convolução para detecção de pontos, linhas horizontais, verticais, diagonais e bordas (fig. 5.15), e
- Aplicação na imagem de um operador de convolução definido pelo usuário.

Todas as operações citadas acima, descritas em detalhes no capítulo 2, foram implementadas no protótipo do sistema N²OCR, facilitando assim a manipulação das imagens digitalizadas. Algumas operações de tratamento de imagens poderiam ainda ser adicionadas a este módulo do sistema, incluindo funções como: aplicação de filtros (*smoothing*, *sharpening*, etc) e realização de transformadas. Podem também ser estendidos os formatos de arquivos manipulados pelo sistema, que se restringem atualmente aos formatos: CUT (usado no HaloDPE), BTM e FIM (formatos internos do sistema N²OCR) e o formato IMF (padrão definido no CPGCC/UFRGS) - Anexo A.1.

Sistema N2OCR - Neural Network Optical Character Recognizer
 IMAGE - Modulo de Manipulacao de Imagens Digitalizadas

Autor : Fernando Osorio - Data : Abril 1991 - Versao 1.0

1. Parametros
2. Arquivos
3. LimpaGraf
4. Histograma
5. Convolucao
6. Realce
7. ExibirVGA
- 8,9 e 10. Diretorio, Os Shell e Fim

= IMAGE =

Parametros

- Arquivos
- LimpaGraf
- Histograma
- Convolucao
- Realce
- ExibirVGA
- Diretorio
- Os Shell
- Fim

Sist N2OCR
 Versao 1.0
 F.Osorio91

TELA DE AJUDA DO SISTEMA. Entre numero da opcao :

Figura 5.9 - Tela de ajuda do módulo de tratamento de imagens

Formato dos Arquivos tipo *.CUT

Imagem = Tamanho Total -> Nro. Colunas / Nro. Linhas (6 bytes)
'N' / Nro. bytes na descr. de 1 linha da imagem
Blocos \ Descricao da linha

Tamanho -> Xl Xh Yl Yh 00 00 (6 bytes -> Xl=LSB, Xh=MSB, ...)

Blocos -> NroBytes (LSB) + NroBytes (MSB) + Descr. dos bytes
Descr. dos bytes tem total de bytes igual a NroBytes
Cada bloco representa uma linha horizontal

Descricao dos bytes -> Total Cor [Cor Cor ...] 00

Se Total tem o bit 8 ligado
Então repete Cor Total vezes (Codificação RLE)
Senão segue Total bytes, apos Total, descrevendo as cores

FORMATO - Descrever arquivos BTM = 1, CUT = 2, FIM = 3 :
2

IMAGE
ARQUIVO
ConvCut
ConvBtm
ConvPcx
ConvImf
Formato
Tamanho
Recorta
Exibe

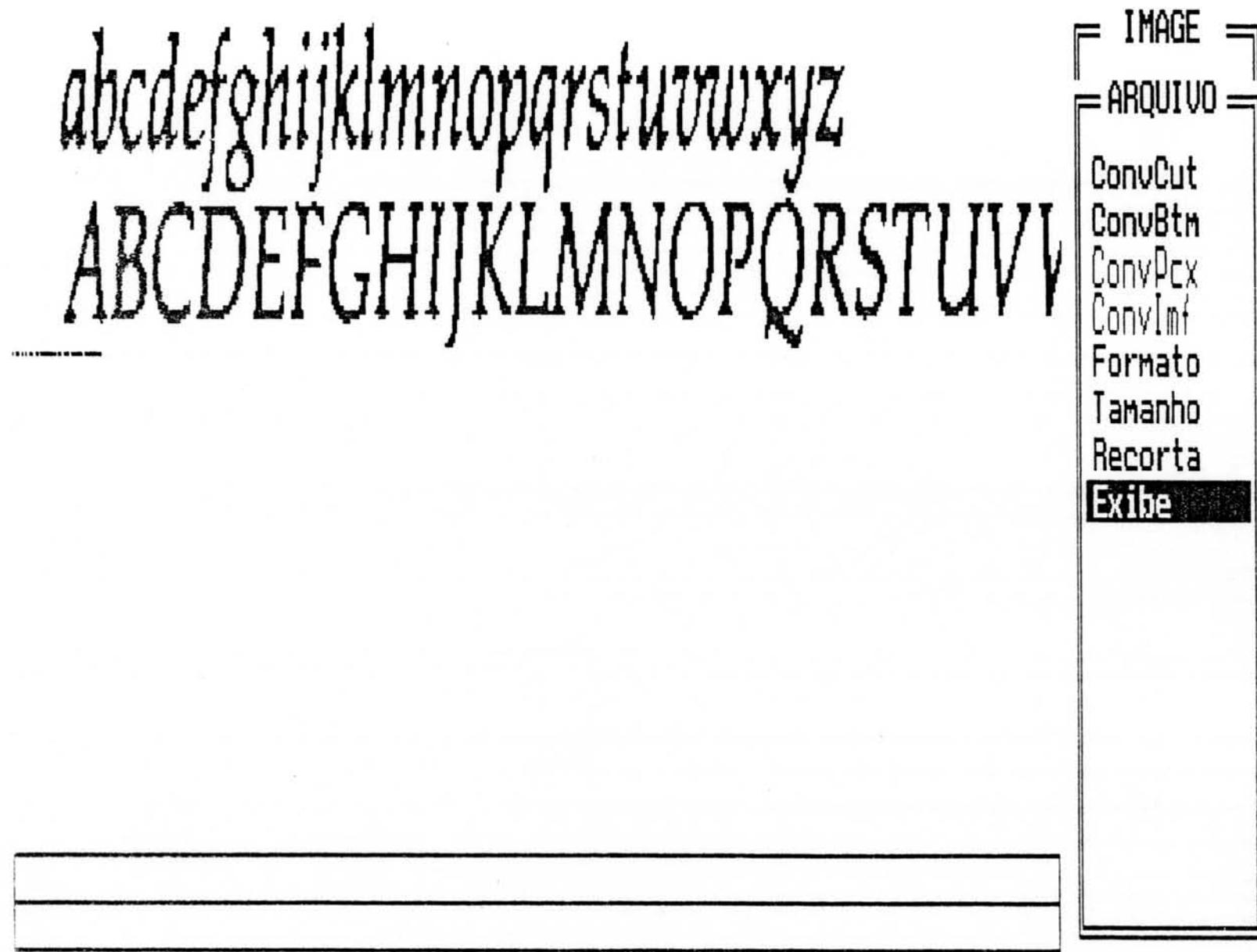


Figura 5.11 - Visualização de uma imagem de um texto

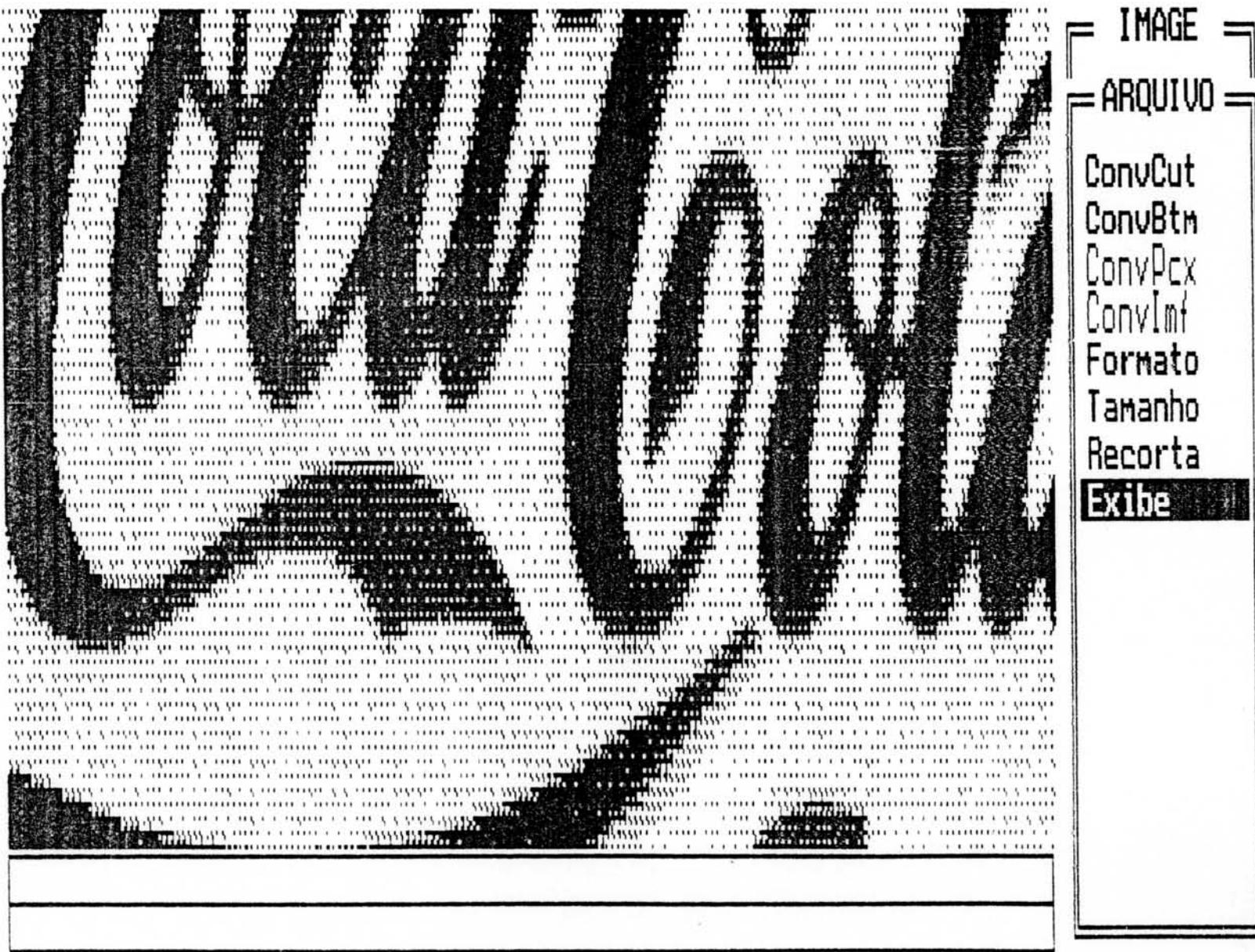


Figura 5.12 – Visualização de uma imagem com halftoning

VALOR: 14

CONT: 166

VALMAX: 11319



Histograma - Nome do arquivo :
coke.fim

= IMAGE =

Parametros
Arquivos
Limpagraf
Histograma

HISTOGR
Gera
Salva N
Equaliza

Sist N2OCR
Versao 1.0
F.Osorio91

Figura 5.13 - Histograma da imagem da fig. 5.12 com half-toning desfeito

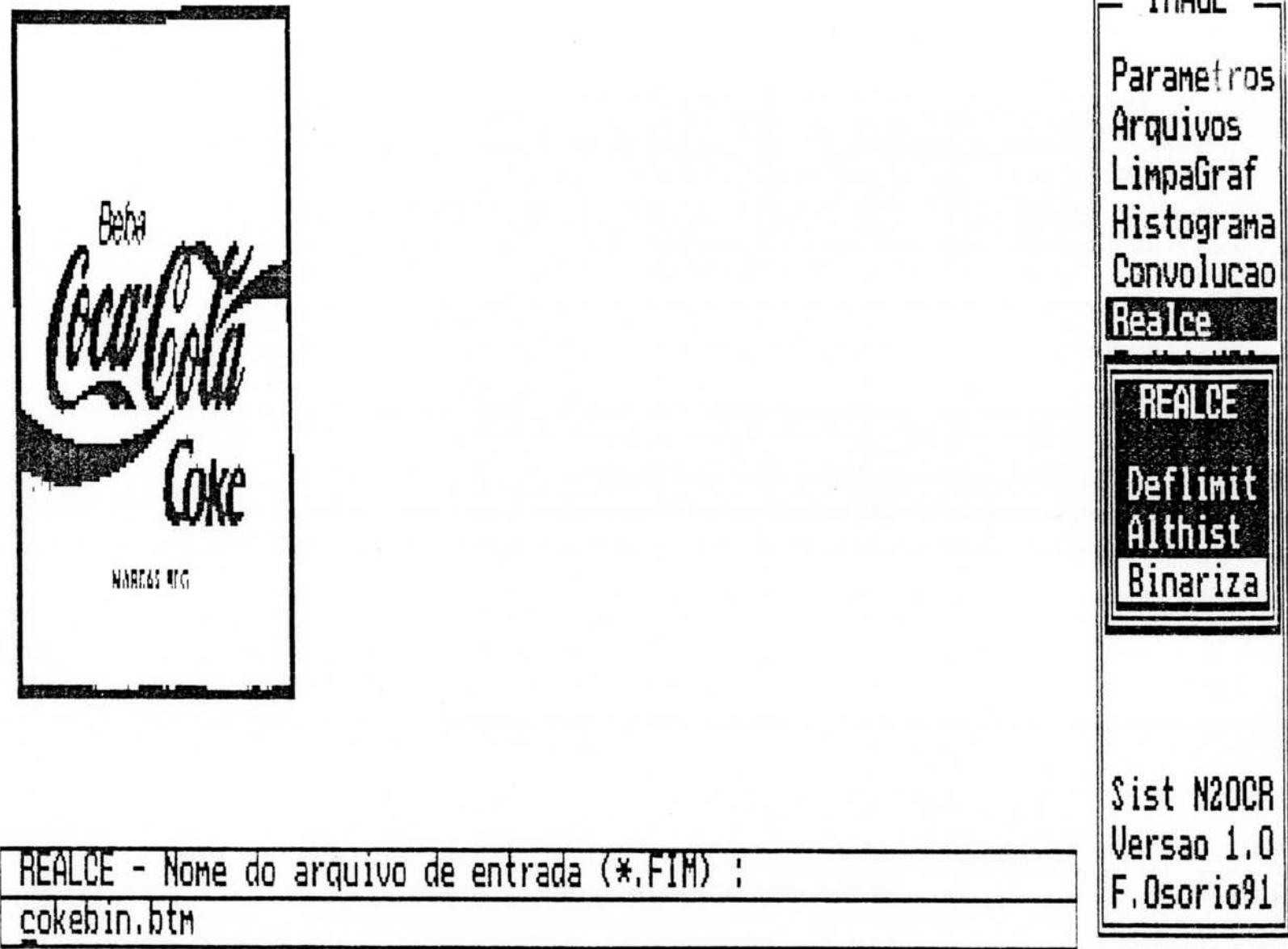


Figura 5.14 - Binarização da imagem cujo histograma aparece na fig. 5.13

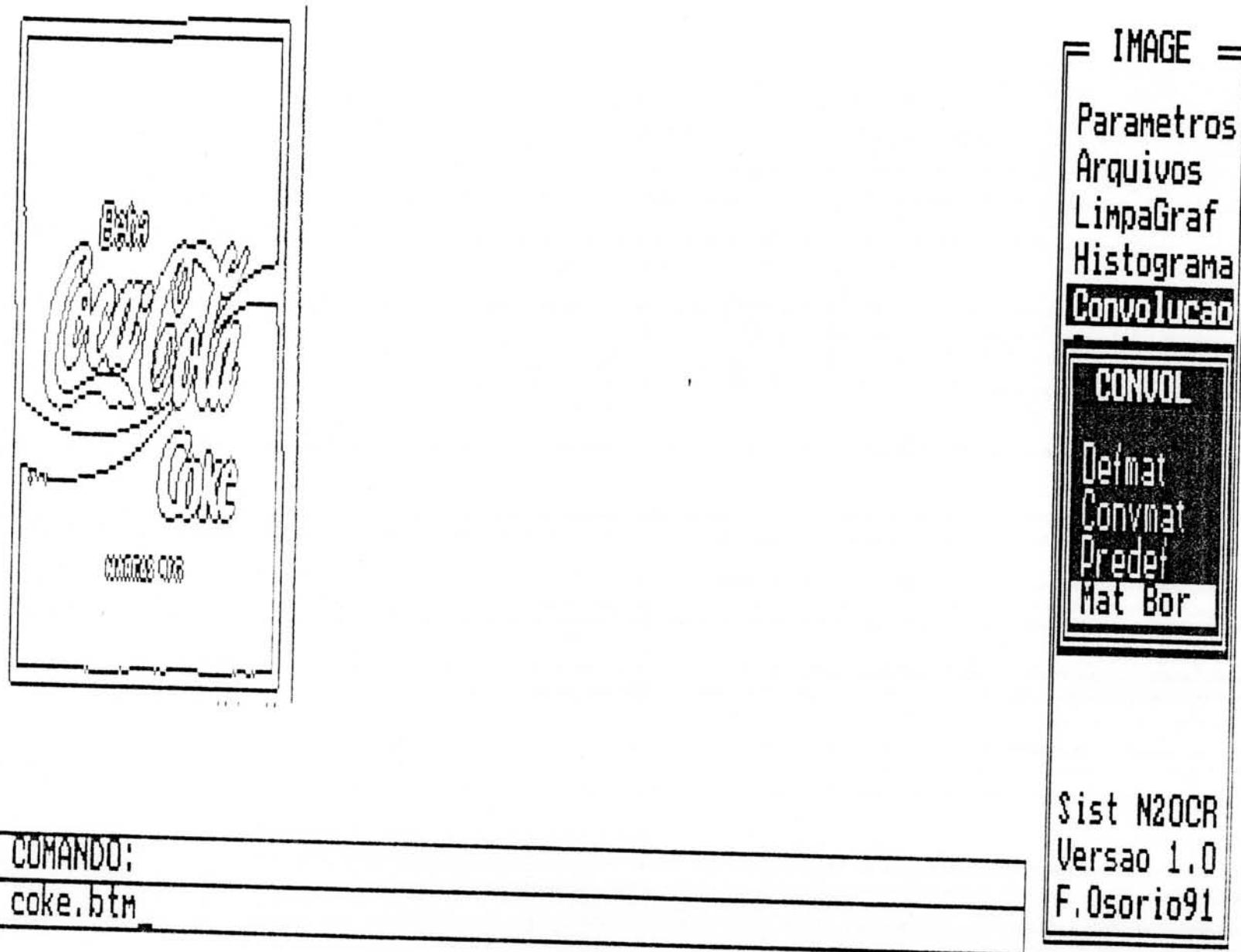


Figura 5.15 - Convolução para detecção de bordas na imagem da fig. 5.14

O módulo de tratamento de imagens consiste de um programa de processamento de imagens essencialmente interativo. Através da visualização da imagem, do seu histograma e do resultado da aplicação de filtros e operações de convolução, o usuário poderá analisar melhor as imagens e optar pelo tipo de processamento que seja mais adequado para um dado documento. Através deste processo interativo de visualização e de processamento da imagem, pode-se, por exemplo, estabelecer o limiar ótimo para a binarização de uma imagem, ou ajustar de uma maneira mais adequada o contraste.

O módulo de tratamento de imagens deve permitir que sejam determinados fatores e procedimentos ideais para um dado tipo de documento a ser digitalizado. Após, este sistema deve permitir a aplicação automática destes procedimentos pré-fixados para toda uma série de documentos a serem processados. Desta forma o usuário pode fixar interativamente um procedimento de tratamento para um certo tipo de documento e posteriormente tornar esta tarefa de processamento automatizada. O protótipo do sistema N²OCR ainda não provê este tipo de facilidades, onde o tratamento da imagem, quando necessário, terá que ser todo realizado com a supervisão/interação com o usuário.

5.4 Preparação dos Dados para a Rede Neural

Depois da digitalização e do tratamento global de um documento, ele é passado para o módulo de preparação dos dados para a rede neural (programa preproc.exe - fig. 5.16). Este módulo extrai os caracteres, um a um, de forma que possam ser identificados pela rede neural. Neste módulo, são implementadas as funções de segmentação e ajuste de posição e escala dos caracteres, além de possibilitar também uma análise visual das imagens por ele manipuladas (fig. 5.17 e 5.18).

=PREPROC=**Parametros**

LimpaGraf
BuscaArq
Segmenta
Escala
Atributos
Texto
exibeImg
Diretorio
Os Shell
Fim

Sist N2OCR
Versao 1.0
F.Osorio91

Sistema N2OCR - Neural Network Optical Character Recognizer
PREPROC - Pre'-Processamento de Imagens Digitalizadas

Autor : Fernando Osorio - Data : Abril 1991 - Versao 1.0

1. Parametros
2. LimpaGraf
3. BuscaArq
4. Segmenta
5. Escala
6. Atributos
7. Texto
8. ExibeImg

9,10 e 11. Diretorio, Os Shell e Fim

Obs: Preproc /c - Configura o sistema para placa CGA !

TELA DE AJUDA DO SISTEMA. Entre numero da opcao :

Figura 5.16 - Tela de ajuda do módulo de preparação dos dados

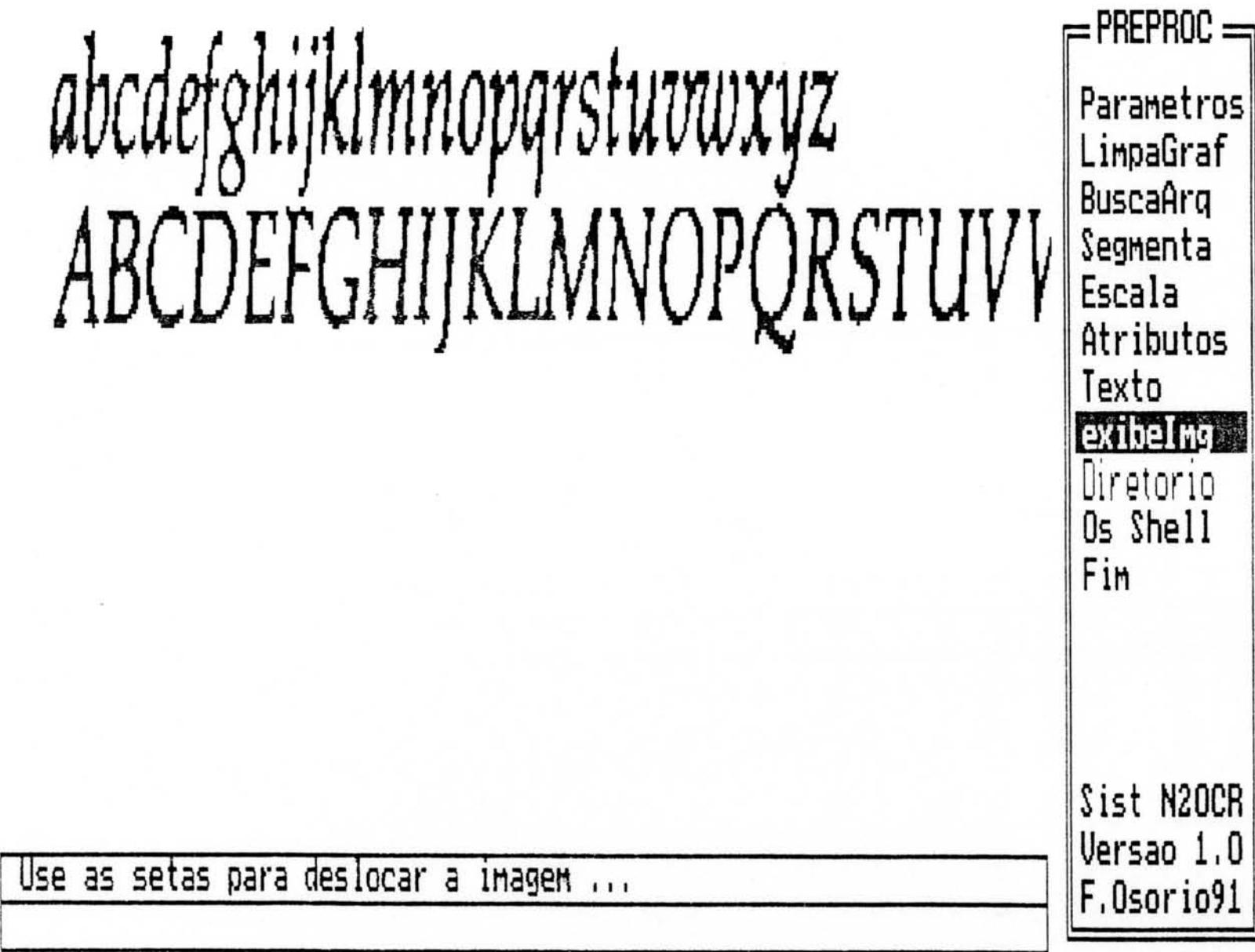


Figura 5.17 – Imagem exibida pelo módulo de preparação dos dados

IJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

=PREPROC=

- Parametros
- LimpGraf
- BuscaArq
- Segmenta
- Escala
- Atributos
- Texto
- exibeImg**
- Diretorio
- Os Shell
- Fim

Sist N2OCR
Versao 1.0
F.Osorio91

Use as setas para deslocar a imagem ...

Figura 5.18 - Exibição de outra área da imagem da fig. 5.17
(deslocamento interativo)

Junto ao módulo de preparação dos dados, poderá ser incluída uma etapa de pré-processamento dos padrões a serem reconhecidos. Este assunto será abordado em maiores detalhes na seção 5.8.

5.4.1 Segmentação

O procedimento de segmentação é dividido em duas etapas: a separação dos textos e gráficos, e a separação dos caracteres entre si.

A separação entre textos e gráficos consiste da aplicação do algoritmo descrito na seção 3.4.1.1 (algoritmo RLSA) resultando uma imagem com os textos apenas, onde todos os gráficos são removidos de forma a não prejudicar o processo de reconhecimento. Cada bloco de texto isolado através do algoritmo RLSA será posteriormente submetido a um algoritmo de separação dos caracteres. Também há a opção de ser realizada a separação de gráficos e textos de forma interativa (o usuário edita a imagem do documento, isolando os textos dos gráficos), mas tal processo é lento e dificulta bastante a utilização prática do sistema de reconhecimento.

A segmentação de caracteres consiste da aplicação de um algoritmo para a separação de caracteres com espaçamento variável, ou através da aplicação de um algoritmo de separação de caracteres com espaçamento fixo. Através de uma configuração prévia é feita a seleção de qual dos dois algoritmos será utilizado na segmentação dos caracteres, definindo-se também os outros parâmetros necessários para a realização destas funções (fig. 5.19). O algoritmo de segmentação para caracteres com espaçamento variável utilizado é aquele descrito na seção 3.4.1.2, sendo baseado na continuidade das linhas que compõem cada caractere. O algoritmo para a separação de caracteres com espaçamento fixo, também descrito na seção 3.4.1.2, quebra a imagem em pequenos blocos com dimensões aproximadas de $M \times N$ pixels, definida pelo usuário.

SISTEMA N20CR - MODULO DE SEGMENTACAO DE IMAGENS

=> Configuracao Atual :

- * Busca inicial - Passo em X = 4
Passo em Y = 1
- * Espacamento entre letras = ?
- * Espacamento entre linhas = ?
- * Numero de 'tiros' horizont.= Maximo
- * Delta superior do 'tiro' = -1
- * Delta inferior do 'tiro' = 1
- * Delta segment. alternativa = 3
- * Inversao da cor da imagem = SIM
- * Execucao pausada = SIM
- * Conjunto de caracteres = ASCII EXTENDIDO
- * Caracter desconhecido = ? - 0x3F
- * Caracter de nova linha = ~ - 0xFE
- * Caracteres do padrao = @ - !

=PREPROC=

=SEGMENT=

CargaCfg

Def_Cfg

ExibeCfg

Graf_Txt

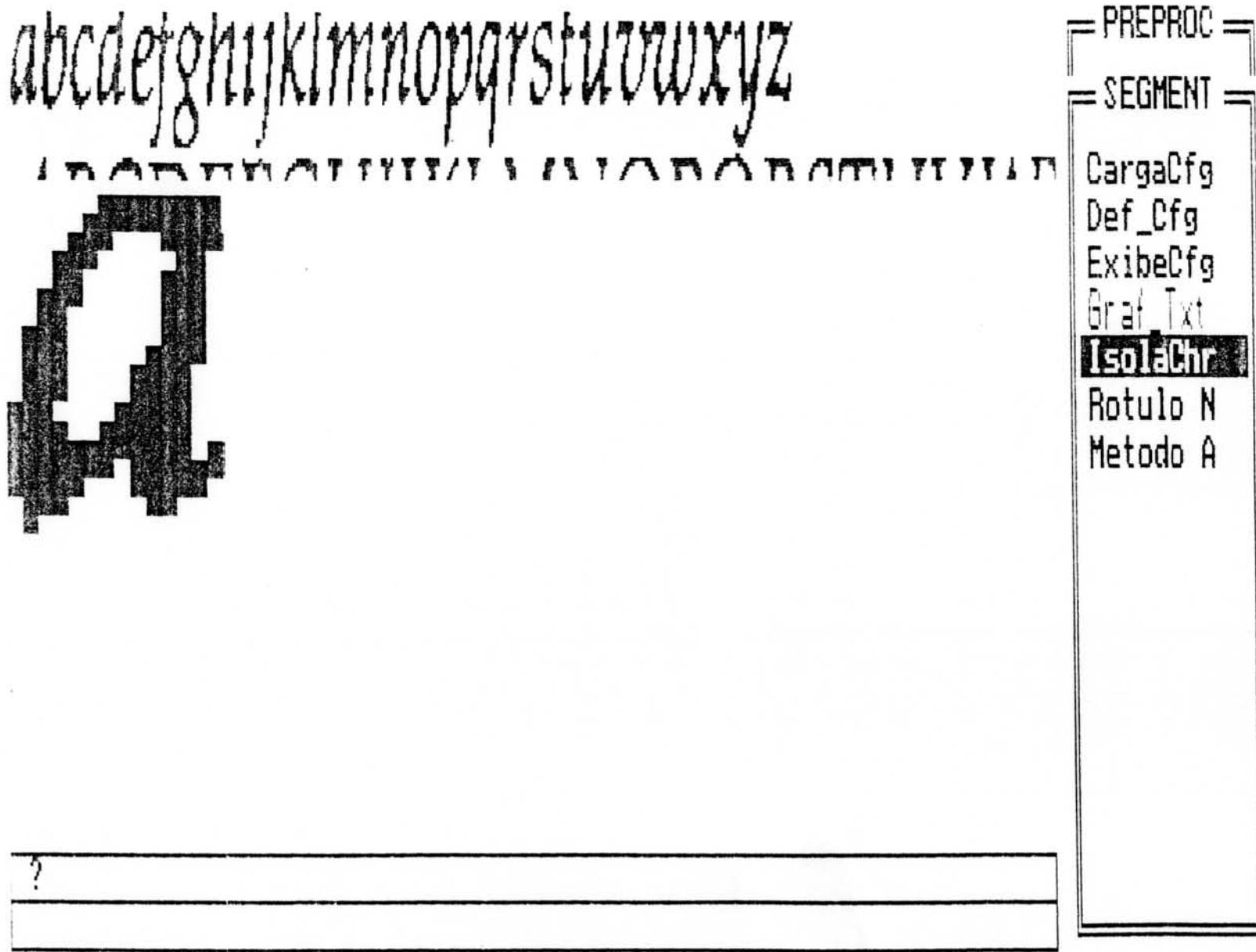
IsolaChr

Rotulo N

Metodo S

Figura 5.19 - Parâmetros de configuração da segmentação

Figura 5.20 - Primeiro caractere isolado da imagem



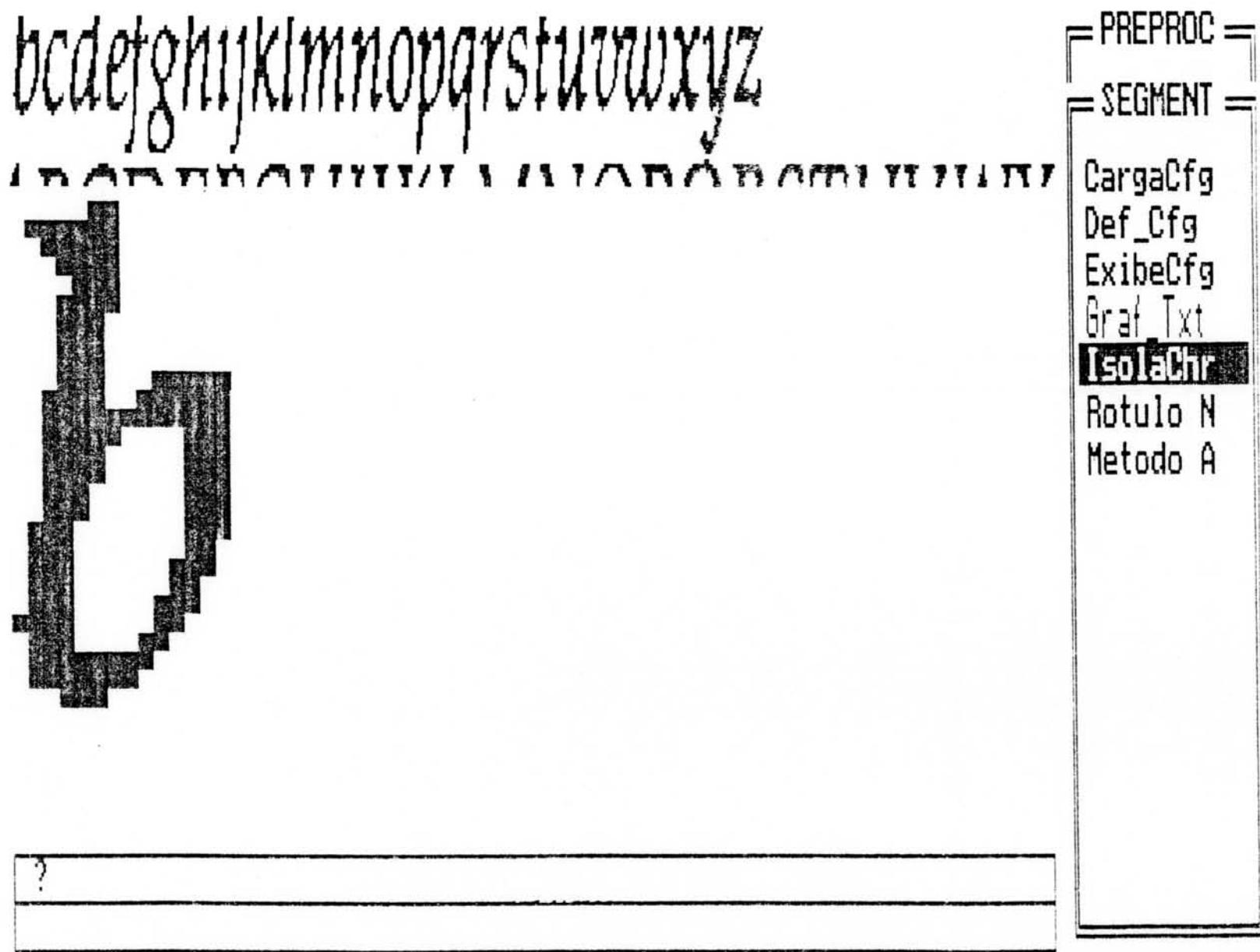


Figura 5.21 - Segundo caractere isolado da imagem

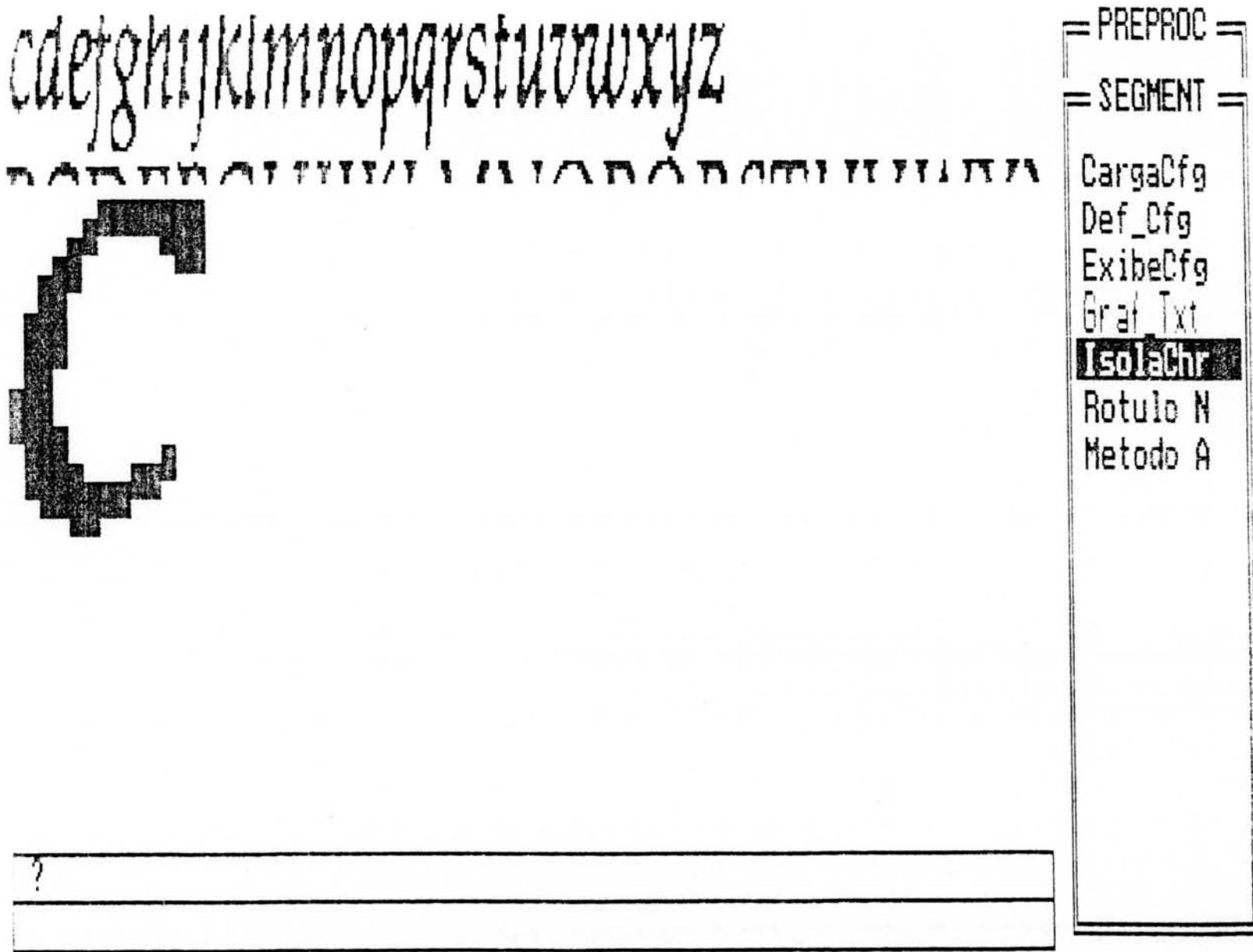


Figura 5.22 - Terceiro caractere isolado da imagem

O protótipo do sistema N²OCR implementa apenas o algoritmo de separação de caracteres com espaçamento variável. Foi implementado somente este algoritmo porque pode ser utilizado na maioria dos casos de reconhecimento de caracteres em textos impressos. A forma como foi implementado o algoritmo de segmentação permite que o usuário tenha uma realimentação do andamento do procedimento de segmentação, ajudando ao usuário no estabelecimento dos parâmetros de configuração da função e, consequentemente, permitindo a obtenção de melhores resultados. As figuras 5.17 e 5.18 apresentam os recursos implementados no módulo de preparação dos dados que facilitam a visualização da imagem, e as figuras 5.20, 5.21 e 5.22 apresentam o desenvolvimento passo a passo da segmentação de caracteres.

O processo de segmentação de caracteres resulta em um conjunto de padrões, com tamanhos diferentes, representando os caracteres isolados. O protótipo do sistema N²OCR gera em disco um arquivo que irá conter os caracteres que foram isolados, onde a descrição deste arquivo é apresentada no Anexo A.2.

A separação de caracteres com espaçamento fixo não está disponível na atual versão do sistema e a separação entre textos e gráficos deve ser realizada de forma interativa, ou seja, o usuário deve editar as imagens através de algum programa, como o HaloDPE, eliminando manualmente os gráficos encontrados junto ao texto. No entanto, a falta destes dois algoritmo na implementação do protótipo do sistema N²OCR não prejudica a sua funcionalidade para fins de teste e validação do mesmo, uma vez que existem métodos alternativos que permitem a sua utilização mesmo sem tais algoritmos implementados.

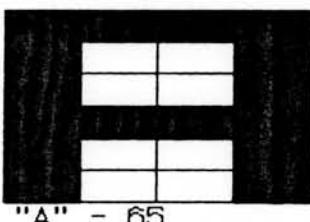
5.4.2 Rotulação

Uma outra fase da preparação dos dados para a rede neural é a rotulação dos padrões já segmentados.

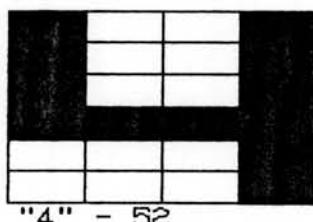
Esta etapa consiste em adicionar aos padrões um rótulo com a sua identificação, onde este rótulo será utilizado pela rede neural na fase de aprendizado dos padrões. O processo de rotulação dos padrões foi implementado no protótipo de forma a ocorrer juntamente com o procedimento de separação dos caracteres.

A rede neural adotada pelo sistema N²OCR é do tipo de rede que recebe um aprendizado supervisionado, ou seja, o usuário precisa apresentar à rede os padrões e a resposta correta que deve ser obtida como saída desta. Para que seja possível este tipo de procedimento, é necessário que, na fase de treinamento, os padrões sigam acompanhados de uma identificação (resposta correta do reconhecimento), permitindo o aprendizado. A identificação ou rótulo nada mais é do que o próprio código ASCII do padrão em questão, conforme demonstra a figura 5.23.

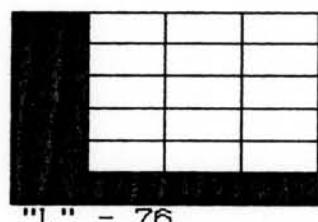
Caracteres Isolados com seus Respectivos Rótulos



"A" - 65

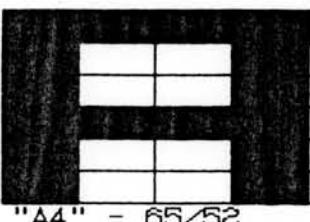


"4" - 52



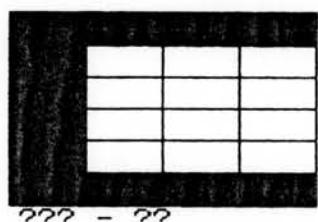
"L" - 76

Caracteres que se Tocam Rotulados



"A4" - 65/52

Desconhecido



??? - ??

Figura 5.23 - Rotulação de Caracteres

O processo de colocação de rótulos é um processo tipicamente interativo, onde, após serem segmentados, os caracteres são apresentados na tela, um a um, de forma que o usuário possa então adicionar os rótulos a cada um destes (fig. 5.20, 5.21 e 5.22).

O usuário pode optar pela realização de uma segmentação dos caracteres, onde a cada caractere isolado será pedido de maneira interativa a indicação do seu código de identificação, ou, no caso de arquivos para reconhecimento, onde os caracteres não são conhecidos, é possível se configurar o sistema para a realização da segmentação sem a necessidade da intervenção do usuário (fig. 5.24). Neste último caso a adição do rótulo que identifica justamente que os códigos de identificação dos caracteres não são conhecidos se dará de forma automática.

A adição de rótulos consiste em observar o caractere isolado com a parte do documento onde este está inserido e, então, indicar qual o código ASCII deste. Poderão ser utilizadas algumas outras formas de codificação, como por exemplo o ASCII Estendido ou o BRASCI. O sistema pode também receber os rótulos a serem associados aos caracteres a partir de um arquivo texto, mas deve se ter um cuidado especial para que haja um casamento do rótulo com o padrão adequado.

Um caso particular a ser tratado são os caracteres que se tocam (*touching characters*), onde, conforme já foi descrito, estes serão tratados como apenas um único padrão, mas cujo rótulo é composto pelos códigos de ambos os caracteres. Desta forma, dois caracteres que não puderam ser isolados serão reconhecidos como um elemento único, mas onde o seu código de reconhecimento é um rótulo de mais de um caractere.

Assim, na fase de aprendizado a rede neural recebe caracteres com o seu respectivo rótulo de identificação, possibilitando o aprendizado. Para a fase de reconhecimento, utiliza-se os padrões que receberam automaticamente um rótulo indicando justamente que este é um padrão desconhecido. A rede neural, ao receber um padrão com o rótulo de desconhecido, então, realizará o reconhecimento, devolvendo o código referente ao padrão apresentado.

Figura 5.24 - Configuração do processo de rotulação

4. Segmentacao : Extrai caracteres da imagem, um a um
- 4.1.CargaCfg - Carrega um arquivo com a configuracao
4.2.Def_Cfg - Define item a item a configuracao
4.3.ExibeCfg - Exibe a configuracao atual do sistema
4.4.Graf_Txt - Realiza a separacao de textos e graficos
4.5.IsolaChr - Realiza a separacao dos caracteres
4.6.Rotulo N - Sem rotulo. Caracteres sao 'desconhecidos'
 S - Com rotulo. O usuario indica interativamente
 o rotulo de cada caracte
4.7.Metodo S - Metodo de separacao de caracteres simplificado
 A - Metodo de separacao de caracteres avancado

=PREPROC=	
=SEGMENT=	
CargaCfg	
Def_Cfg	
ExibeCfg	
Graf_Txt	
IsolaChr	
Rotulo N	
Metodo A	

TELA DE AJUDA DO SISTEMA. Entre numero da opcao :

4

```
#97 14 18
0000001111111
0000011111111
00001110001111
00011100000110
00011000001110
00111000001110
00110000001110
01110000001110
01110000011110
01110000111100
01110000111100
11100001111100
11110011111100
11111111111101
11111110111111
11111000111110
01110000011000
01000000000000
#98 14 27
00000110000000
```

MOSTRA - Tecle algo para prosseguir ...
texto.let



Figura 5.25 - Caractere isolado com código de identificação

```
#63 14 18
00000011111111
00000111111111
0001110001111
00011100000110
00011000001110
0011000001110
00110000001110
01110000001110
01110000011110
01110000111100
01110000111100
11100001111100
11110011111100
11111111111101
11111110111111
11111000111110
01110000011000
01000000000000
#63 14 27
00000110000000
```

MOSTRA - Tecle algo para prosseguir ...

texto.tst



Figura 5.26 - Caractere isolado com rótulo de desconhecido

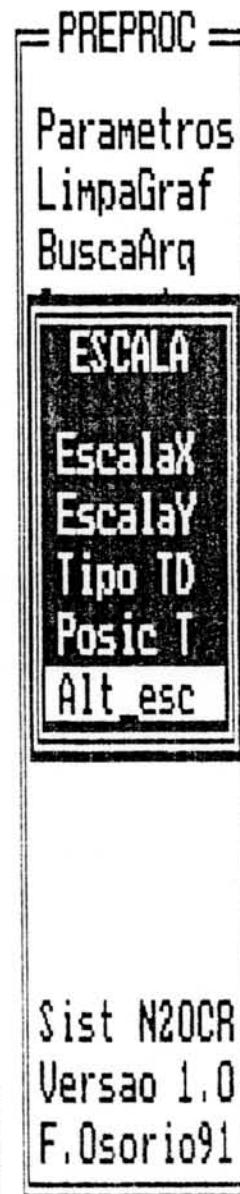
A figura 5.25 apresenta um caractere isolado com o seu respectivo código de identificação ('a' = ASCII 97), e a figura 5.26 apresenta um caractere com o rótulo que identifica que este caractere é desconhecido ('?' = ASCII 63). O código de identificação de caracteres desconhecidos é um parâmetro configurável do sistema.

5.4.3 Ajuste de Escala e Posição

A etapa final da preparação dos dados para a rede neural é o ajuste de escala e posição, que foi implementado seguindo exatamente a descrição feita na seção 3.4.3. Este algoritmo é de fundamental importância no processo de reconhecimento, porque a rede neural necessita de uma entrada com um número fixo de elementos, para poder efetuar o reconhecimento. Devido a este fato, o ajuste de escala é indispensável, uma vez que permite a padronização do tamanho dos caracteres segmentados pelo processo realizado na etapa anterior. O ajuste de posição também possui uma importância destacada, pois o processo de reconhecimento empregado, na versão atual do sistema, baseia-se no casamento direto de padrões. A figura 5.27 apresenta as opções implementadas na função de ajuste de escala e posição, e as figuras 5.28, 5.29 e 5.30 apresentam exemplos de ajuste de escala em diferentes tamanhos.

5.5 Rede Neural

O módulo de reconhecimento é implementado através de uma rede neural. Esta rede neural adota um modelo específico, criado especialmente para esta aplicação e denominado de: ADAn-LIENE - *ADaptive Non-Linear Expandible NEtwork*. Esta seção descreverá a estrutura, o funcionamento e a forma de atuação do modelo proposto para atuar junto ao sistema N²OCR.



- 5.Escala : Ajuste de escala e posicao do padrao extraido
- 5.1.EscalaX - Tamanho padrao de escala no eixo X (colunas)
- 5.2.EscalaY - Tamanho padrao de escala no eixo Y (linhas)
- 5.3.Tipo TD = Todos os padroes sofreem o ajuste
- GT = Todos padroes maiores que o tamanho padrao sao reduzidos, os outros permanecem inalterados
- GP = Ajuste parcial. Somente a dimensao (X ou Y) que exceder o tamanho maximo sera' ajustada
- 5.4.Posic T = Ajuste de posicao pelo topo (esq, sup.)
- C = Ajuste de posicao pelo centro
- 5.5.AltEsc - Funcao que efetiva o ajuste de escala e posicao

Figura 5.27 - Ajuste de escala e posição

TELA DE AJUDA DO SISTEMA. Entre numero da opcao :

5

```
$10 10 4
0000111111
0001100111
0011000110
0110000110
0110000110
0110001110
1100011110
1111111111
1111001110
0100000000
97
0111100000
0011100000
0011000000
0011001111
0111100111
0111000011
0110000011
0110000110
1110001100
```

MOSTRA - Tecle algo para prosseguir ...
textol.let



Figura 5.28 – Padrão com ajuste de escala 10 x 10

```
$16 16 4
0000000111111111
0000001111111111
0000111100001111
0001110000001110
0001110000001110
0011100000011110
0111100000011110
0111100000111110
0111100001111100
0111000011111100
1111000011111100
1111101111111100
1111111011111111
1111110001111110
0111100000111000
0100000000000000
97
0000011100000000
0011111100000000
0000111100000000
```

MOSTRA - Tecle algo para prosseguir ...
texto2.let



Figura 5.29 - Padrão com ajuste de escala 16 x 16

\$20 20 4
 000000001111111111
 000000011111111111
 00000011110000111111
 00000111100000011110
 00001111000000011110
 000111000000011110
 0001110000000011110
 011111000000011110
 0111110000000111110
 0111110000011111000
 0111110000011111000
 111100000111111000
 1111100001111111000
 111111111111111001
 111111111111111111
 1111111110111111111
 1111111000001111100
 01111100000001110000
 01100000000000000000

Figura 5.30 - Padrão com ajuste de escala 20 x 20

MOSTRA - Tecle algo para prosseguir ...

texto3.let



5.5.1 Modelo Adotado

O modelo de rede neural adotado - ADAn-LIENE, como o próprio nome já indica, é baseado no ADALINE de Widrow e Hoff. Este modelo resultou da análise dos diferentes modelos de redes neurais adotados na atualidade, onde se buscava um modelo específico para um sistema OCR. O modelo adotado deveria suprir as necessidades de tais aplicações, de maneira simples e eficiente.

5.5.1.1 Escolha do Modelo

Será apresentada aqui uma rápida análise dos modelos já apresentados no capítulo 4, justificando a escolha e desenvolvimento do modelo ADAn-LIENE.

Os modelos competitivos como o modelo de Hopfield e outros similares, como a máquina de Boltzmann e a BAM, foram os primeiros a serem eliminados da lista de possíveis modelos a serem adotados. Tais modelos possuem alguns inconvenientes como: o tamanho excessivamente grande das redes, a convergência para estados espúrios e a dificuldade de implementação direta em *hardware*. Analisando o modelo de Hopfield, como sendo o representante desta classe de modelos, nota-se que para este modelo ainda seria necessário uma complementação da rede neural, uma vez que esta apenas "corrigir" as distorções, devolvendo o padrão restaurado. A identificação do padrão ainda tem que ser feita numa etapa posterior. Outro problema é referente à própria estrutura da rede, onde o seu tamanho é demasiadamente grande e o número de conexões, que interligam todos os neurônios entre si, praticamente inviabiliza uma implementação em *hardware*, onde cada neurônio seria considerado como um elemento individual de processamento. Como um dos objetivos finais deste trabalho é justamente a proposta de um modelo de rede neural que possa ser implementado em *hardware*, estes modelos foram descartados.

Apesar do modelo de Kohonen ser um dos modelos mais interessantes, pela forma similar à realidade com que este representa o comportamento do cérebro humano, este modelo também foi descartado. O principal motivo foi o fato deste modelo gerar os mapas de atributos através da criação de zonas de ativação na rede neural, mas para identificar estas zonas de ativação seria necessário um novo sistema de reconhecimento de padrões. A complexidade de tal operação não justificava a utilização de tal modelo.

Restaram os modelos adaptativos por correção de erros, que somente pela sua capacidade de adaptação já satisfazem a uma das principais necessidades de implementação de sistemas OCR. Dentro destas classe de modelos, há um grupo com aprendizado supervisionado e um modelo de auto-organização, o ART.

Os sistemas de OCR possuem a característica de seu aprendizado ser realizado através de um processo supervisionado. Isto é devido ao fato de que cada novo padrão a ser ensinado ao sistema já possui uma identificação prévia, que é fornecida ao sistema. De nada adianta um sistema onde haja uma auto-organização, e onde os padrões são agrupados (*clusterization*) de acordo com suas características, pois este agrupamento podem não se dar da forma desejada. Um algoritmo de agrupamento pode vir a colocar em uma mesma classe os caracteres "8" e "B", por exemplo, apesar destes não pertencerem a uma mesma classe. Em resumo, em um sistema de OCR, as classes já são bem definidas (conjunto de caracteres), o que implica no uso de algoritmos com aprendizado supervisionado.

Isto quer dizer que o modelo ART deve ser completamente desconsiderado? Não, existe uma propriedade muito importante no modelo ART que será adotada no modelo do ADAn-LIENE. Esta propriedade é a da expansão da rede, do aumento do número de clusters no modelo ART. Cada vez que a rede é apresentada a um novo padrão que não se enquadra, dentro de certos limites, aos padrões já conhecidos, a rede

cria um novo nodo responsável pelo aprendizado deste novo padrão. Conforme será apresentado a seguir, o modelo do ADAn-LIeNE também possuirá esta capacidade de expansão, que torná-lo-á mais eficiente.

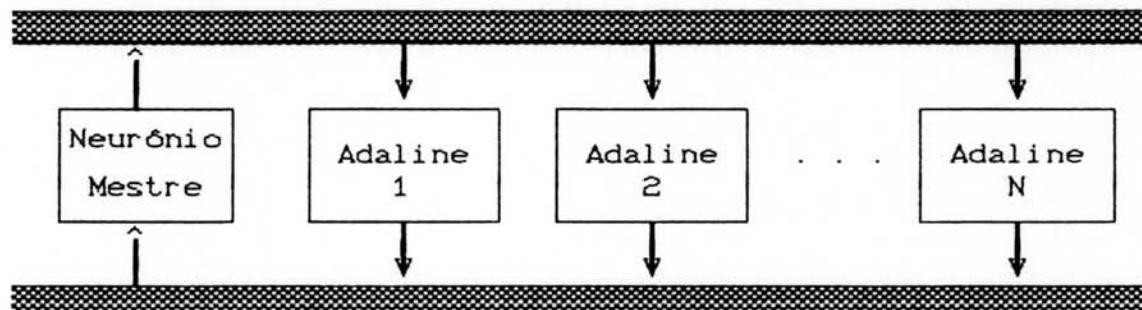
Do modelo ART foi extraída a capacidade de expansão da rede, durante o processo de aprendizado e adaptação a novos padrões. E do modelo do Adaline foram extraídas as características dos neurônios utilizados na rede. O modelo ADAn-LIeNE utiliza neurônios exatamente iguais aos propostos no modelo do Adaline, com aprendizado através da Regra Delta. Este algoritmo de aprendizado garante à rede uma capacidade de adaptação muito visada em sistemas OCR. A rede neural que se utiliza de neurônios do tipo Adaline possuirá a capacidade de se ajustar aos padrões.

E quanto ao problema da linearidade das decisões tomadas pelo Adaline (rede de apenas uma camada), que levavam a impossibilidade de reconhecer padrões como o XOR? É exatamente devido a este fato que a rede implementada faz uso de uma abordagem de expansão tipo ART, usando também os conceitos definidos por Widrow na criação da rede Madaline. Desta forma as redes são dotadas de decisões não lineares. É justamente destas características que surge o nome adotado para este modelo: *Adaptive Non-Linear Expandible Network*.

Resta apenas a análise de um último modelo, o modelo multinível baseado na regra de *back-propagation*. Este modelo poderia ter sido adotado, sendo inclusive um modelo muito utilizado em diferentes aplicações das redes neurais na atualidade. Mas como a sua grande vantagem está em implementar uma rede neural com decisões não lineares, e isto já foi obtido através do emprego de uma rede similar ao modelo Madaline, torna-se desnecessária a sua utilização. Outros motivos de não ter sido empregado o modelo multinível foram: a sua complexidade, a dificuldade de implementação deste modelo diretamente em hardware e o problema da estimativa do número de neurônios e camadas necessárias para uma dada aplicação.

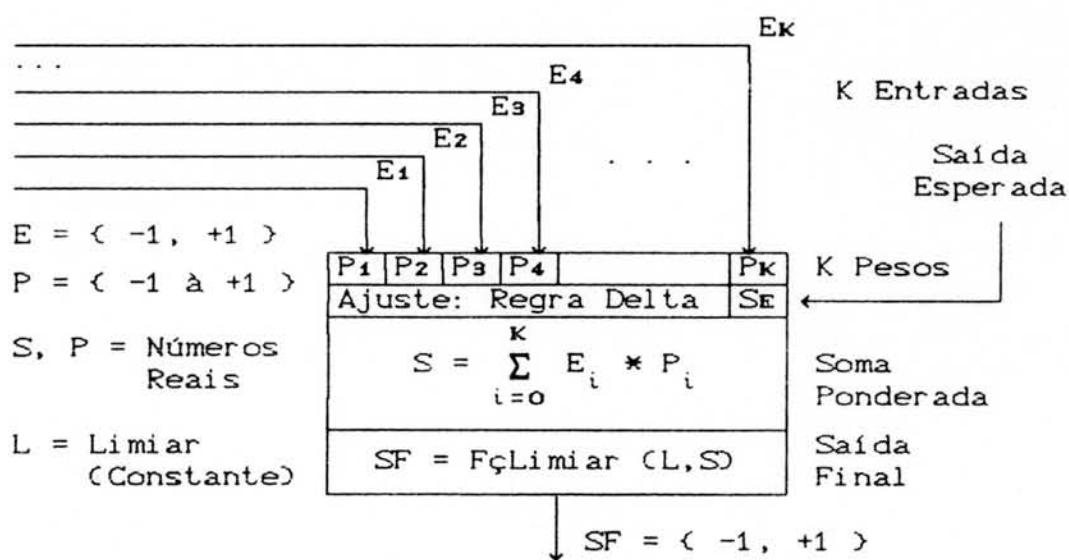
O modelo do ADAn-LIeNE, conforme será demonstrado logo a seguir, é extremamente simples e possibilita uma fácil implementação em hardware.

Barramento de Dados de Entrada



Barramento de Dados de Saída

(a) Visão Global da Rede Neural



(b) Elemento Processador Individual - Neurônio Adaline

Figura 5.31 - Organização da rede neural do modelo ADAn-LIeNE

5.5.1.2 Descrição do Modelo

O modelo ADAn-LIeNE é composto por um conjunto de neurônios do tipo Adaline, onde cada neurônio recebe como entrada todos os elementos que constituem um determinado padrão. Cada um dos neurônios será responsável pelo aprendizado de uma determinada classe, ou seja, cada neurônio tem por função o reconhecimento de um caractere específico. A figura 5.31 apresenta a organização da rede neural do modelo ADAn-LIeNE. A rede neural deste modelo, conforme pode ser visto na figura 5.31, possuirá um elemento especial, o neurônio mestre (*Master Neuron*), sendo este responsável pela coordenação do aprendizado e reconhecimento na rede neural, bem como do controle da expansão desta.

A rede proposta se utiliza apenas de valores binários de entrada, representando os elementos de uma imagem de apenas dois níveis. Todos os modelos de redes apresentados e analisados eram orientados ou permitiam a sua utilização para este fim específico. É importante salientar que existem outros modelos de redes neurais que aceitam como entrada valores com mais de dois estados, mas estes modelos, voltados ao processamento de imagens em geral, não são de interesse neste trabalho e não foram objeto de estudo, apesar de sua destacada importância.

O modelo proposto permite que os padrões binários, representando os caracteres, sejam apresentados à rede de forma que apenas um neurônio se ative identificando o caractere. Para que isto seja possível, é necessária a realização de uma fase de aprendizado, para somente após realizar o reconhecimento.

5.5.2 Aprendizado

A fase de aprendizado é realizada recebendo-se, um a um, os padrões juntamente com o código ASCII associado a estes. O neurônio mestre é quem gerenciará estes padrões, determinando qual o neurônio que será treinado para reconhecer cada um dos padrões.

Inicialmente a rede começa a operar, na fase de aprendizado, com todos os neurônios inativos, exceto o neurônio mestre que coordena todo o funcionamento da rede. O neurônio mestre seleciona então o primeiro padrão e "ensina" este a um primeiro neurônio selecionado. O mestre, então, cria uma tabela de controle de aprendizado, onde será descrito o número de cada neurônio em uso na rede (cada um é identificado por um número ou endereço único) e o código do padrão (ASCII) ensinado a este. Para cada novo padrão a ser ensinado à rede, o mestre consulta esta tabela para ver se o código do padrão já está definido ou não. Caso o código do novo padrão a ser aprendido não esteja na tabela, o mestre seleciona um novo neurônio, ensinando a este o novo padrão, e adicionando mais uma entrada na tabela de controle. Caso o código deste novo padrão já exista na tabela, o mestre apenas testa a rede, verificando se o neurônio que foi treinado para identificar este caractere se comportará da maneira correta, ativando-se. Pode ser necessária uma adaptação (ajuste dos pesos) no caso de se ter no aprendizado dois ou mais padrões diferentes representando uma mesma classe (caractere).

O aprendizado consiste em enviar o padrão de treinamento a cada neurônio em uso na rede, juntamente com a resposta que este deve fornecer em sua saída (saída esperada - mentor *input*). Ao serem ativadas as entradas da rede, apenas um neurônio deve ter sua saída ativada (+1), isto implica que apenas um dos neurônios deve ter ativada (+1) sua entrada mentora. Através da aplicação da Regra Delta, os neurônios que não obtiverem em suas saídas o valor de ativação correto ajustam os seus pesos de atuação de forma a obter a resposta desejada. Assim, ao ser ativada a rede pelo neurônio mestre, esta se adapta de forma a responder conforme o indicado por ele, ajustando seus pesos pela Regra Delta.

Cada novo caractere a ser ensinado à rede resulta na adição de um novo neurônio, conforme demonstra a figura 5.32. Assim a rede vai se expandindo à medida que novos padrões lhe são ensinados, colocando em uso neurônios que estavam inativos. De uma certa forma, isto equivale a criar novos neurônios na rede à medida que estes são necessários. Este tipo de expansão da rede foi baseado no princípio de funcionamento do modelo ART.

Devido a estas características do modelo ADAn-LIENE, fica fácil a adição ou remoção de padrões a serem reconhecidos pela rede. Para ampliar o número de classes (caracteres) reconhecidas basta que sejam adicionados tantos neurônios quantas são as novas classes adicionadas à rede, treinando-os para reconhecer as novas classes de padrões. E no caso de não ser mais necessário o reconhecimento de uma dada classe, então é necessário apenas "desligar" o neurônio que era responsável pelo seu reconhecimento.

Aprendidos : caractere "A" e "B"

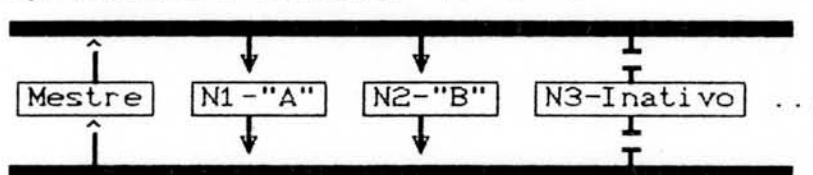


Tabela de Controle de Aprendizado

Neur.	Caract.
N1	"A"
N2	"B"
N3	Inativo
...	Inativo

Aprendidos : caractere "A", "B" e "C"

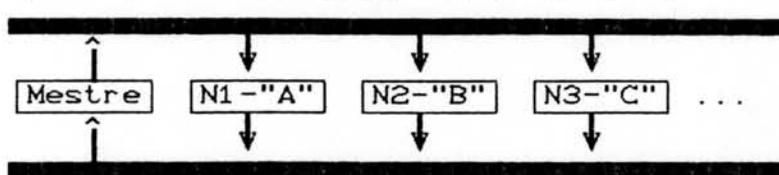


Tabela de Controle de Aprendizado

Neur.	Caract.
N1	"A"
N2	"B"
N3	"C"
...	Inativo

Figura 5.32 - Expansão da rede neural

Ao sofrer o ajuste dos pesos, durante a fase de aprendizado, um neurônio pode "esquecer" o padrão para o qual foi treinado. Quando são alterados os pesos de um

neurônio para este se adaptar a um novo padrão ou para este dar uma determinada resposta, pode ocorrer deste neurônio ter seus pesos modificados de tal forma que não responda mais da maneira correta aos padrões ensinados anteriormente. Devido a este fato, para cada novo padrão ensinado à rede, deve se testar novamente todos os outros padrões ensinados anteriormente, de forma a comprovar que a rede continua a apresentar um comportamento correto diante de cada um dos padrões ensinados a ela. Logo o aprendizado segue a seguinte ordem :

Ensina	Padrão 1	Padrão 2	Padrão 3	Padrão 4
Testa	Padrão 1	Padrão 2,1	Padrão 3,1,2	Padrão 4,1,2,3

O aprendizado segue o algoritmo que é descrito abaixo:

1. Ativa a rede para o padrão N
2. A resposta foi correta ?
 - 2.1 Não - Adapta os pesos, volta ao passo 1
 - 2.2 Sim - Segue para o próximo passo
3. Para K igual a 1 até N incrementado de 1
 - 3.1 Ativa a rede para o padrão K
 - 3.2 A resposta foi correta ?
 - 3.2.1 Não - Adapta os pesos e volta para 3
 - 3.2.2 Sim - Passa para o próximo valor de K
 - 3.3 Repete o passo 3.1 até acabar todos os padrões de treinamento (até K ser igual a N)
4. Fim

É importante salientar que a rede só terminará de realizar a adaptação a um novo padrão no momento em que esta for ativada para todos os padrões já ensinados, fornecendo a resposta correta para todos eles. Caso ocorra um erro, corrigir-se os pesos e volta-se a testar os padrões a partir do primeiro.

O simulador implementado, deste primeiro modelo, realizava a execução exata do algoritmo descrito acima. O ajuste dos pesos era feito através da Regra Delta. Sabia-se que para o caso do padrão da função XOR este procedimento resultaria na execução de um laço sem saída, onde o passo 3 do algoritmo nunca chegaria a ser completado. No caso da XOR, um mesmo neurônio não conseguiria nunca ajustar os seus pesos de forma a dar a resposta correta para todos os padrões de entrada. Este fato é devido ao modelo implementado realizar apenas uma função de decisão linear. Mesmo ciente deste fato, acreditava-se que um sistema de reconhecimento de caracteres não deveria nunca ficar sujeito ao aprendizado de um padrão como o da função XOR. Esta linha de pensamento mostrou-se incorreta com o decorrer dos testes para com a rede neural implementada.

Ao ser realizada uma simulação da rede neural para um dado conjunto de caracteres específicos, ocorreu da rede entrar em um laço infinito, durante a fase de aprendizado. Após uma análise mais detalhada do problema, chegou-se à conclusão que este só poderia ser resultante do aprendizado de padrões que não podiam ser aprendidos com a utilização de uma função de decisão linear: os padrões não eram linearmente separáveis...

Foi então que decidiu-se pela implementação de uma nova técnica, a qual, de certa forma, segue os conceitos propostos por Widrow no modelo do Madaline. A técnica adotada consiste do seguinte: ao ser detectado um laço infinito, caracterizado pela alternância de ajuste de pesos em dois neurônios específicos que nunca conseguem chegar ao resultado desejado, o neurônio mestre pára o processo de aprendizado e cria um novo neurônio na rede. Este novo neurônio especializar-se-á em reconhecer o último padrão que estava sendo ensinado à rede e que causou o problema do laço infinito.

Este caso do laço infinito só ocorre quando um novo padrão, pertencente a uma classe já definida anteriormente, é ensinado à rede. Este novo padrão possui a característica de, juntamente com outro padrão pertencente a mesma classe deste, formar uma situação similar ao do aprendizado da função XOR, ou seja, estes dois padrões pertencentes a uma mesma classe não podem ser classificados corretamente por uma função de decisão linear.

Como o caso descrito acima só ocorre quando é feita a tentativa de ensinar um novo padrão à rede, que pertence a uma classe já definida (o padrão é uma nova amostra de um caractere que já foi ensinado a rede), é necessário apenas que este caso particular seja resolvido. A solução deste problema é simples, como já foi descrito anteriormente. Adiciona-se um novo neurônio que se dedicará a reconhecer exclusivamente este novo padrão causador do problema. Surge então uma nova questão, como fazer para que todos os padrões representantes de uma mesma classe sejam reconhecidos como pertencentes a uma classe única? É simples, na tabela de controle de aprendizado é adicionado um novo campo, onde este campo indica se um determinado neurônio foi treinado para a primeira versão de uma classe ou para alguma versão alternativa desta mesma classe. Esta "versão alternativa" é justamente o padrão problema que gerou a necessidade de criação de um novo neurônio para reconhecer uma classe já ensinada à rede. A tabela de controle de aprendizado terá a organização demonstrada na figura 5.33.

Através do uso da tabela de controle, que permite a identificação de versões diferentes de um mesmo caractere, é possível se implementar uma rede similar ao Madaline. Nesta rede pode se ter, assim como no Madaline, dois neurônios tipo Adaline que identificam dois padrões diferentes pertencentes a uma mesma classe. A saída final da rede é uma função "ou lógico" das saídas dos neurônios associados. Este modelo de simulação permite, então, que se

tenha diferentes versões para uma mesma classe, onde a saída final da rede, gerada pelo neurônio mestre, indica qual a classe reconhecida, independente de ter sido o neurônio X ou Y o que reconheceu uma das versões desta classe.

Nro. do Neurônio	Classe (Caráter)	Versão
0 0 1	"A"	0
0 0 2	"B"	0
0 0 3	"FT"	0
0 0 4	"A"	1
0 0 5	"C"	0
0 0 6	"D"	0
0 0 7	"B"	1
0 0 8	"F"	0

Figura 5.33 - Tabela de controle de aprendizado

Um fator importante a ser considerado, que é reflexo do modelo de rede proposto, é quanto ao tamanho da rede neural. Como a rede neural permite a expansão através do acréscimo de novos neurônios, uma preocupação natural é para que esta rede não cresça demasiadamente. Obviamente, o número mínimo de neurônios será igual ao total de caracteres diferentes a serem reconhecidos (número total de classes). Quanto ao número máximo de neurônios, este dependerá de alguns fatores relacionados com o processo de aprendizado, estes fatores são:

- O limiar (L) usado na função de transferência;
- O fator de ajuste (β) da Regra Delta;
- O número de padrões exemplares de uma mesma classe e
- A distorção destes padrões, ou seja, o grau de diferença entre os padrões pertencentes a uma mesma classe.

Em relação ao limiar da função de transferência, quanto maior for, mais rígido será o processo de classificação. E quanto mais rígida a classificação, mais difícil será adaptar um mesmo neurônio para reconhecer diferentes versões de uma mesma classe. Como consequência deste fato, será criado um número maior de neurônios.

O fator β controla a magnitude da alteração dos pesos através da Regra Delta. Caso o ajuste dos pesos seja realizado com variações muito grandes, ficará difícil a implementação de um "ajuste fino" dos neurônios durante a fase de aprendizado. Portanto, quanto maior o valor de β , mais difícil ficará o ajuste dos pesos, e consequentemente a rede crescerá pela adição de diferentes versões para uma mesma classe.

O limiar da função de transferência, assim como o valor do fator de ajuste da Regra Delta, são definidos através do arquivo de configuração da rede neural, descrito posteriormente na seção 5.6. Estes valores são aplicados a todos os neurônios da rede, ou seja, todos os neurônios fazem uso do mesmo valor de limiar da função de transferência e do fator de ajuste da regra de aprendizado. É importante salientar que no caso do limiar da função de transferência, este valor pode ser programado para um certo valor a ser utilizado apenas na fase de aprendizado, com a função de fazer a rede neural se comportar conforme a descrição acima, e, na fase de reconhecimento este valor poderá ser redefinido, passando a atuar como um controle do grau de semelhança mínimo entre o padrão de teste e os padrões aprendidos. O fator de ajuste não terá influência sobre o reconhecimento, uma vez que a rede neural desativa a função de adaptação dos pesos durante esta fase.

Tanto o valor do limiar quanto o do fator β devem levar em consideração o número de representantes de uma mesma classe, assim como o grau de distorção destes padrões, pois todos estes elementos juntos determinarão o comportamento da rede em relação a sua expansão.

Um estudo sobre as relações diretas entre estes componentes torna-se muito interessante e produtivo. Este tipo de análise constitui-se em uma possível extensão dos trabalhos desenvolvidos até o presente.

5.5.3 Reconhecimento

A fase de reconhecimento é simples. Primeiramente é desabilitado o ajuste dos pesos, uma vez que a rede só pode realizar este ajuste se for fornecida a resposta correta junto com os padrões e, no reconhecimento, esta resposta não é conhecida.

O reconhecimento consiste do neurônio mestre fornecer o padrão de teste simultaneamente para todos os neurônios em uso da rede. O neurônio mestre receberá as saídas de todos os outros neurônios, verificando qual o neurônio que se ativou. Caso nenhum neurônio tenha se ativado, a resposta será "não reconheceu" (*not match*). Em caso contrário, o neurônio mestre consultará a tabela de controle de aprendizado para poder indicar a resposta final da rede. O mestre, sabendo qual foi o neurônio que se ativou, pode então obter através da tabela de controle o código ASCII do caractere identificado (o número da versão é desprezado).

E se dois neurônios ativarem suas saídas ao mesmo tempo? Neste caso o mestre selecionará um destes como sendo a resposta correta. O processo de seleção pode ser aleatório ou baseado na ordem em que estes aparecem na tabela de controle. É claro que este tipo de metodologia não é a mais indicada para a solução do problema pois pode levar a erros de reconhecimento. Por isto, na próxima seção, são apresentadas algumas sugestões para a melhoria do modelo descrito.

O elemento responsável pela determinação de uma maior ocorrência de casos como "não reconheceu" ou "múltiplos reconhecimentos" é o limiar da função de transferência aplicado na fase de reconhecimento.

5.5.4 Extensão do Modelo

Conforme foi demonstrado no item anterior, pode ocorrer de mais de um neurônio ativar sua saída na presença de um determinado padrão de entrada. Isto pode constituir tanto um problema como também um auxílio no funcionamento do sistema OCR.

No capítulo 3, foi citada a possibilidade de utilização de métodos para a correção do texto reconhecido através do uso de um pós-processamento baseado no contexto. Neste mesmo capítulo foi indicado que o sistema N²OCR, apesar de não implementar tal função, já possuia prevista a extensão do mesmo de forma a possibilitar este tipo de operação.

Existem basicamente duas possibilidades de pós-processamento para a correção dos textos reconhecidos através do sistema N²OCR. A primeira é baseada no modelo que foi proposto anteriormente, sem nenhuma alteração. Como neste modelo é possível a rede ativar a saída de mais de um neurônio, para um dado padrão de entrada, basta que o neurônio mestre, ao invés de "sortear" um dos neurônios como sendo o da resposta correta, passe ao módulo de pós-processamento todos os valores obtidos como possíveis respostas do reconhecimento realizado pela rede. O módulo de pós-processamento terá então mais condições para determinar, entre um conjunto de dados possíveis, qual resposta que possui a maior probabilidade de ser o caractere correto, dentro do contexto em que este está inserido.

Através da alteração do valor do limiar da função de transferência, é possível aumentar ou diminuir a incidência de respostas simultâneas de reconhecimento de uma classe pela rede. Um valor elevado para o limiar da função de transferência implica um maior número de respostas tipo "não reconheceu", e um valor mais baixo para este limiar implica um maior número de saídas dos neurônios ativadas ao mesmo tempo.

A segunda possibilidade de utilização de um pós-processamento junto ao sistema N²OCR requer uma modificação do modelo de rede neural proposto. A modificação ocorre apenas na conexão da saída dos neurônios com o neurônio mestre. Ao invés de enviar apenas valores +1 ou -1 para o mestre, cada neurônio pode passar a enviar diretamente o valor correspondente ao resultado da soma ponderada, sem a aplicação da função de transferência, ou, com a aplicação uma função de transferência do tipo "lógica limiar" ou "sigmóide". Esta alteração no modelo proposto faz com que a saída dos neurônios não forneça apenas valores binários, mas sim uma faixa de valores, a qual deve ser normalizada (valores entre -1 e +1) para fins de simplificação dos algoritmos, permitindo inclusive um melhor acompanhamento do funcionamento da rede neural.

Para determinar qual o neurônio que melhor identificou o padrão, o mestre achará o maior valor de ativação de saída entre os valores recebidos. O neurônio que enviou o maior valor de saída será considerado como aquele que mais provavelmente identificou corretamente o padrão presente em suas entradas. A determinação do maior valor é feita de uma maneira imediata, não afetando a eficiência do sistema. Uma vez que o mestre tem que ler um a um os valores de saída dos neurônios, a comparação pode ser feita imediatamente entre o maior valor lido até o momento e o novo valor recebido, guardando sempre o maior deles.

Nesta segunda proposta, a correção através do pós-processamento é realizada da seguinte forma: o neurônio mestre agora tem a possibilidade de gerar não apenas o código de um único caractere reconhecido, mas também a lista dos N melhores candidatos ao reconhecimento, classificados em ordem ascendente de similaridade. É necessário apenas definir o número de elementos que constituirão esta lista de candidatos (adotou-se três) e enviá-los ao módulo de pós-processamento. Este módulo então avaliará entre os candidatos qual o melhor, de acordo com o contexto em que estão inseridos.

Com a eliminação da aplicação da função de transferência, sempre existirá um candidato ao reconhecimento, não ocorrendo a situação onde a resposta é "não reconheceu". A resposta "não reconheceu" poderá voltar a ser obtida adicionando-se um controle, como o realizado pela função de transferência dos neurônios tipo Adaline, ao neurônio mestre. Isto equivale a postergar a aplicação da função de transferência, deixando esta tarefa para o neurônio mestre. A vantagem desta abordagem é o fato do neurônio mestre ter condições de determinar entre dois candidatos qual o que obteve o maior valor de ativação, o que não era possível quando a aplicação da função de transferência ocorria a um nível interno nos neurônios.

Outro ponto interessante de ser ressaltado é o efeito destas alterações realizadas a nível do comportamento da rede neural no processo de aprendizado. O problema surge na aplicação da regra delta, uma vez que modificamos a saída dos neurônios e não se tem mais o conceito de saída igual a +1 significando o reconhecimento do padrão e saída igual a -1 significando o não reconhecimento. Como saber se devemos ou não realizar a correção dos pesos de atuação de um neurônio? Qual é a saída desejada para cada neurônio?

Existem duas soluções principais, a primeira é a de modificar o comportamento dos neurônios, desativando a aplicação da função de transferência, apenas na fase de reconhecimento, mantendo-a da maneira como vinha sendo utilizada durante a fase de aprendizado. A segunda solução consiste em transferir para o neurônio mestre a decisão de realização ou não de uma correção dos pesos da rede. Neste tipo de procedimento, durante a fase de aprendizado, o neurônio mestre irá receber os valores de ativação de cada neurônio, determinando aquele que obteve o maior grau de reconhecimento, ou seja, o maior valor de saída. Se o neurônio que obteve a maior saída é o que deveria identificar o padrão, não precisamos adaptar os pesos.

No outro caso, em que o neurônio que obteve a maior saída não é aquele que deveria reconhecer o caractere, o neurônio mestre ordena que seja aplicada a Regra Delta, no neurônio que teve a maior saída, mas que não deveria ter reconhecido o padrão (reconhecimento incorreto - saída desejada = -1), e no neurônio que não teve a maior saída, mas que era aquele que deveria reconhecer o padrão (erro de não reconhecimento - saída desejada = +1).

Estes diferentes modelos e variações foram todos desenvolvidos através da implementação de um simulador, onde tem-se a possibilidade de configurar uma série de parâmetros, fazendo com que este atue de uma ou de outra forma. O simulador implementado permite a configuração do tipo da função de transferência utilizada na fase de aprendizado, do tipo desta função utilizada na fase de reconhecimento, dos valores de limiar para estas funções e inclusive o fator de ajuste da regra de aprendizado que pode ser diferenciado para o caso de reconhecimento incorreto (fator β de erro) e para o caso de erro de não reconhecimento (fator β de acerto).

Todos estes parâmetros constituem uma série muito ampla de possibilidades de investigação quanto ao desempenho da rede neural junto ao reconhecimento de caracteres. Os parâmetros adotados, tipos de funções de transferência e exemplos de configurações possíveis para o simulador do modelo ADAn-LIene apresentam-se listados no Anexo A.5, sendo também especificados junto a cada gráfico de avaliação do desempenho do sistema (Anexo A.4), qual a configuração empregada. O autor utilizou-se de um conjunto de parâmetros nos testes, os quais resultaram em um desempenho muito bom da rede neural no reconhecimento de caracteres, mas a estimativa de quais são os valores ótimos empregados nestas funções constitui-se de um trabalho extenso e novo, o qual ainda não foi realizado.

5.5.5 Implementação em Hardware

Nesta seção, será apresentada apenas uma visão superficial de alguns aspectos relevantes para a especificação de uma arquitetura especial que implemente o modelo de rede neural descrito neste trabalho. O objetivo é dar ao leitor uma visão geral da possibilidade de implementação em hardware do modelo que foi simulado por software.

A arquitetura básica do sistema é baseada no esquema apresentado na figura 5.31. O sistema é composto por um conjunto homogêneo de processadores que operam em paralelo e realizam individualmente as tarefas relativas aos neurônios tipo Adaline. Estes processadores estão interconectados por um barramento a um processador mestre. O processador mestre é um processador mais sofisticado (um microprocessador, por exemplo) que executa tarefas de controle da rede neural.

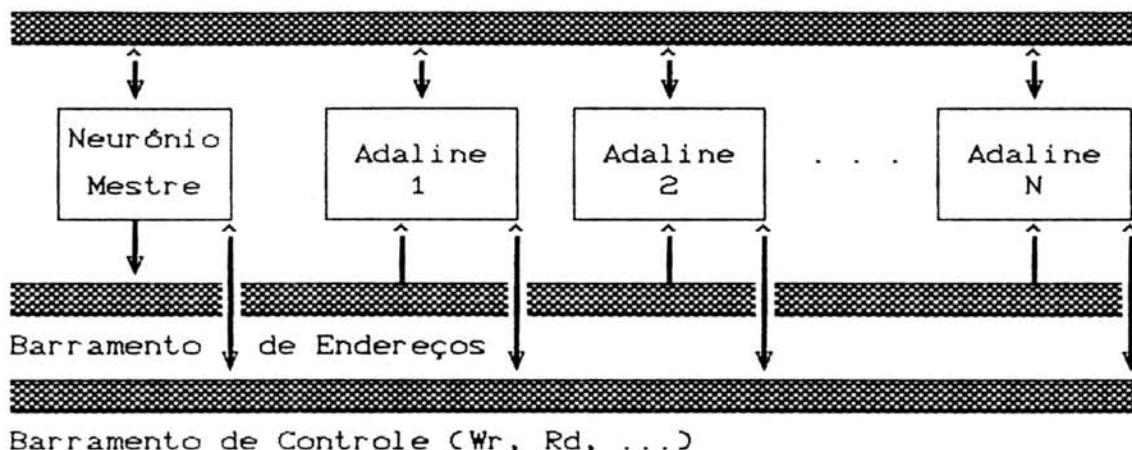
Os dados são eviados através do barramento de dados, onde utiliza-se uma técnica de difusão simultânea (*broadcast*) das informações para todos os elementos processadores (neurônios). Os dados podem ser enviados de uma só vez, caso o barramento possua uma largura que suporte todo o padrão (em torno de 256 bits - 16x16), ou o padrão pode ser particionado enviando-se, por exemplo, de 16 em 16 bits até compor todo o padrão.

O neurônio mestre terá que possuir alguma forma de seleção individual dos neurônios, seja para selecionar o neurônio que enviará seu valor de saída, seja para selecionar qual o neurônio que receberá o valor de saída desejado (*mentor input*). Uma possibilidade de implementação é através do uso de barramento de endereços, onde cada neurônio recebe um endereço específico que será utilizado para o seu acesso. Através desta técnica, o envio do valor de saída desejado pode ser feito selecionando o endereço de um certo neurônio e enviando este dado.

Para a leitura do valor de saída de cada neurônio, o mestre pode realizar um processo de *pooling*, fazendo o acesso a cada um dos neurônios seqüencialmente.

A figura 5.34 apresenta um esquema simplificado de uma arquitetura para implementação do modelo analisado neste trabalho, juntamente com as funções que este sistema deve oferecer.

Barramento de Dados



Funções do Mestre

Envia → Dado : Padrão (*bit-map*) – Mestre envia

Padrão Endereço: Endereço do neurônio que deve ser
ativar (Único com saída = +1)

Controle: Sinal de escrita (Wr)
 Se endereço = 0 → Reconhece
 Se endereço \neq 0 → Aprende

Recebe → Dado : Valor de ativação – Mestre recebe

Saídas Endereço: End. do neurônio a ser acessado

Controle: Sinal de leitura (Rd)

Outras funções: Inicializa – Pesos, Limiar, Fator β

Carrega Pesos – Valores pré-gravados

Lê Pesos – Grava os pesos atuais

Figura 5.34 – Proposta simplificada de uma arquitetura

Uma rede neural pode ser implementada em hardware, através do projeto de uma arquitetura dedicada como a descrita acima, ou pode ser desenvolvido um projeto visando o aproveitamento apenas do paralelismo inherente ao processo, mas utilizando-se de uma arquitetura para o processamento paralelo, que não seja necessariamente dedicada apenas a implementação de uma rede neural.

O paralelismo nas redes neurais pode ser explorado de diferentes formas. A primeira, que é a mais clara destas formas, é obtida através de uma implementação onde cada neurônio é um elemento individual de processamento. Este enfoque foi o apresentado através do esquema da figura 5.34. Uma outra possibilidade é o aproveitamento apenas do paralelismo da multiplicação do vetor de entradas pelo vetor de pesos, onde esta multiplicação pode ser realizada em processadores vetoriais, acelerando o processo. Neste enfoque já não se tem mais o conceito de neurônio como sendo um elemento individual. O aproveitamento do paralelismo existente na multiplicação das entradas e pesos pode ser realizado de diferentes maneiras. Um exemplo de implementação de simuladores de redes neurais em arquiteturas paralelas não dedicadas é descrito em [OSO 90b].

5.6 Arquivo de Configuração

Para que o sistema N^2 OCR possa se tornar um sistema realmente prático no reconhecimento de caracteres, é necessária a possibilidade de configuração de alguns parâmetros. A configuração do sistema permite que este se adapte de uma forma mais perfeita ao tipo de texto e necessidades de uma dada aplicação do usuário.

A implementação desta configuração do sistema é feita através da utilização de um arquivo de configuração. O arquivo de configuração descreverá os diversos itens referentes ao sistema N^2 OCR que podem ser definidos, programados ou selecionados pelo usuário.

Este arquivo de configuração pode ser alterado livremente pelo usuário (arquivo em formato texto) e deve ser consultado pelo sistema na sua inicialização ou quando for solicitada uma reconfiguração do sistema.

O protótipo do sistema N²OCR não possui ainda implementadas todas as funções de configuração descritas nesta seção. Como nem todos os algoritmos previstos para o sistema N²OCR estão presentes no protótipo, alguns parâmetros de configuração não foram considerados. As funções implementadas no protótipo, em alguns casos, foram simplificadas e não possuem todas as possibilidades de configuração que poderiam fazer uso. Outro detalhe a respeito da implementação realizada no protótipo é o fato de cada módulo possuir a sua própria configuração, sendo que no sistema N²OCR está prevista a utilização de um arquivo único para a configuração geral de todas as ferramentas.

Uma função muito importante dos arquivos de configuração é a definição de parâmetros para o processamento automático do sistema, com o mínimo de intervenções realizadas pelo usuário. Este tipo de processamento, na implementação disponível atualmente, não é um procedimento comum, onde a configuração e processamento no sistema se dá através da configuração e seleção de opções através de uma interface amigável orientada a menus.

Os parâmetros configuráveis do sistema N²OCR são os seguintes:

1. Parâmetros para aquisição de imagens:

- Definição da fonte da imagem digitalizada, indicando se é obtida diretamente do dispositivo de aquisição de imagens, ou se esta deve ser carregada de um arquivo específico;
- Nome do arquivo que contém a imagem e
- Formato do arquivo de imagens, podendo ser do tipo CUT, TIF, PCX, BTM ou IMF.

2. Parâmetros para tratamento das imagens:

- Tipo de tratamento da imagem, indicando o tipo de filtragem a ser realizada sobre esta;
- Indicação se é ou não necessária a realização de uma operação para desfazer o halftoning e
- Indicação se é necessária a realização da binarização da imagem e, no caso da realização desta, fornecimento do valor do limiar utilizado na binarização.

3. Parâmetros para a preparação dos dados para a rede neural:

- Indicação do método de separação de textos e gráficos (automática ou interativa);
- Fornecimento dos fatores usados no controle da separação entre textos e gráficos, conforme descrição feita no capítulo 3;
- Tipo de espaçamento dos caracteres, podendo ser espaçamento fixo ou variável (*fixed pitch/proportional*). Indica qual algoritmo de segmentação será empregado;
- Fornecimento dos fatores usados para a determinação de espaços ou linhas em branco na separação dos caracteres entre si, conforme descrição feita no capítulo 3;
- Fornecimento dos fatores para segmentação de caracteres com espaçamento fixo: altura e largura média de um caractere;
- Tipo de detecção inicial dos textos, podendo ser automática ou interativa;
- Indicação dos fatores usados para a detecção automática de textos, conforme descrição do capítulo 3, e
- Indicação da necessidade ou não da extração de atributos, fornecendo qual o processo a ser empregado;

4. Parâmetros para a simulação:

- Tipo de código usado na representação dos caracteres: ASCII, EBCDIC, BRASCI, ASCII Estendido ou definível pelo usuário;

- Código de saída da rede para um caractere não reconhecido;
- Especificação do número máximo de neurônios utilizáveis pela rede neural (gera aviso no caso de exceder este número);
- Tipo de procedimento a ser executado: aprendizado ou reconhecimento;
- Se for efetuado um aprendizado, pode ser feita a indicação de que se deseja "salvar" os pesos da rede após realizar este.
- Se for realizado um reconhecimento, pode ser feita a indicação de um arquivo para a carga dos pesos previamente salvos, realizando o reconhecimento baseado em um conhecimento prévio;
- Determinação do comportamento das saídas dos neurônios: saídas com valores binários ou não;
- Definição do tipo de função de transferência empregada na fase de treinamento e de reconhecimento;
- Fixação dos valores do limiar "L" e do fator de ajuste " β " usados pelos neurônios;
- Determinação do limite de retentativas realizadas no aprendizado, antes de realizar uma expansão da rede, e
- Indicação da necessidade de geração ou não de um relatório da simulação. Este relatório descreverá o número total de neurônios utilizados, o número total de ativações da rede, o número de caracteres com mais de uma versão (e quais são estes caracteres), o número médio de ajustes dos pesos por neurônio, etc.

Todos os fatores citados acima permitem um melhor desempenho do sistema frente a diferentes tipos de textos a serem reconhecidos. A obtenção de uma boa taxa de reconhecimento correto de caracteres dependerá da correta configuração e do ajuste mais adequado dos valores de controle do sistema.

5.7 Avaliação do Desempenho do Sistema

Baseando-se nos resultados obtidos a partir da utilização do protótipo do sistema N²OCR, chegou-se à conclusão que este possui um ótimo desempenho no que se refere ao reconhecimento correto dos padrões. Quando o sistema é treinado para um certo conjunto de padrões e, após, estes mesmos padrões são então reintroduzidos na rede para o reconhecimento, a taxa de reconhecimento correto é sempre de 100%. Esta propriedade é inerente ao sistema, devido ao tipo de modelo adotado, onde o algoritmo de aprendizado só termina a sua execução quando todos os padrões de aprendizado tiverem sido testados e reconhecidos corretamente.

Os testes realizados com o sistema N²OCR foram realizados com três conjuntos diferentes de padrões de aprendizado/reconhecimento:

- Imagens de caracteres obtidas através da digitalização com um scanner manual de textos de uma impressora laser → Denominado de *Padrão de Teste A*;

- Padrões gerados através da utilização de um programa desenvolvido para este fim, com a possibilidade de adicionar distorções → Denominado de *Padrão de Teste B*, e

- Padrões obtidos por um sistema CCD de aquisição de imagens a partir de uma conjunto de dados fornecido pelo correio inglês → Denominado de *Padrão de Teste C*.

Estes três tipos de padrões de teste possuem alguns exemplares que são apresentados no Anexo A.3, onde tem-se alguns padrões típicos de cada amostra que foram empregados na avaliação do sistema N²OCR. A seguir serão descritos cada um dos padrões de teste utilizados, em relação as características de obtenção dos mesmos (processo de aquisição) e em relação ao tipo de caracteres contidos nestes (fontes, distorção, etc).

O padrão de teste A foi obtido a partir da digitalização com um scanner operado manualmente (*handy scanner HS-3000*) produzido pela empresa DFI Inc. A digitalização foi realizada com a obtenção de imagens P&B (2 níveis somente, sem *halftoning*) e com o ajuste de contraste do dispositivo em sua posição média. O software de aquisição de imagens utilizado foi o programa SCAN.EXE que acompanha o scanner, sendo gravadas as imagens em arquivos do tipo *.CUT. As imagens digitalizadas foram feitas a partir de textos impressos em uma impressora laser de 300 DPI, onde os textos foram editados em uma estação de trabalho SUN, e impressos com uma única fonte, a fonte *default* para impressão na laser. Todo a manipulação posterior das imagens obtidas pelo scanner foi realizado utilizando-se o protótipo do sistema N²OCR.

O padrão de teste B foi produzido com o programa PADRÃO.EXE que é um dos módulos do protótipo do sistema N²OCR. Os padrões de caracteres obtidos já são gerados de forma isolada, não havendo a necessidade de empregar um algoritmo de segmentação. Os padrões foram gerados com a utilização de uma única fonte, equivalente aos caracteres *default* do TurboC (Compilador C - Borland Inc.) para o modo gráfico. Para efeitos de teste do reconhecimento foram geradas distorções apenas com a inversão de pixels do padrão. Nos testes com o padrão B não foram adicionadas outras distorções como rotação ou deslocamento do padrão.

O padrão de teste C foi fornecido pelo prof. Edson Filho da Universidade Federal do Pernambuco, sendo esta base de dados utilizada por este professor nos testes realizados em sua tese de doutorado [FIL 90 e FIL 91]. O conjunto de dados é formado por caracteres alfanuméricos impressos extraídos de envelopes postais e fornecidos pelo Correio Britânico. A digitalização foi realizada através de um dispositivo CCD de aquisição de imagens. A base de dados utilizada é composta por 34 classes de caracteres, obtidos a partir de múltiplas fontes (diferentes tipos de

impressão), onde tem-se as letras de A até Z (excluindo-se a letra 'O' e a letra 'I' - 24 classes), e os dígitos de 0 a 9 (onde os números '0' e o '1' não são distinguidos das letras 'O' e 'I' - 10 classes). Na base de dados, cada classe contém 300 amostras de padrões digitalizados em 384 pixels P&B, organizados em uma matriz de 16 x 24 pontos. Os padrões já se encontram isolados, tendo sofrido um processo anterior de segmentação e ajuste de posição e escala. Para a utilização desta base de dados foi realizada apenas a conversão do formato de arquivo dos padrões para os formatos utilizados pela rede neural (conversão para arquivos do formato tipo *.LET e *.TST), adicionando-se os respectivos rótulos para permitir o aprendizado e reconhecimento.

Os testes realizados sobre o sistema N²OCR foram concentrados principalmente sobre os padrões de teste A e C, por estes reproduzirem mais a realidade de um sistema OCR, utilizando dados extraídos de casos reais. Os testes e resultados obtidos com os diferentes tipos de padrões são apresentados no Anexo A.4. Estes testes fizeram uso de informações geradas nos arquivos *.REC e *.REL (vide Anexo A.5), sendo também utilizada a função de avaliação implementada no módulo PADRÃO (fig. 5.35).

Os teste com o padrão A obtiveram resultados excelentes, onde mesmo com um pequeno treinamento a rede atinge um alto grau de reconhecimento correto dos caracteres. Nestes testes o grau de reconhecimento correto ficou entre 98 e 100%, demonstrando que para este tipo de aplicação o sistema N²OCR está plenamente apto a ser utilizado. Outra característica importante deste teste é o fato de envolver todas as etapas de processamento realizadas pelo sistema N²OCR, da aquisição até o reconhecimento.

Resultados da Comparacao :

IGUAIS = 527 - 99.06%
 ERROS = 5 - 0.94%
 DUVIDAS = 0 - 0.00%
 TOTAL -> 532

Erros Ocorridos :

Original -> Resultado [Posicao]
 57 - 9 48 - 0 [194]



Figura 5.35 - Avaliação automática do reconhecimento

Os resultados obtidos nos testes com o padrão C são muito bons e devem ser analisados cuidadosamente, uma vez que este padrão apresenta um alto grau de distorções e onde o texto está escrito com a utilização de múltiplas fontes, o que, mesmo em sistemas profissionais de OCR, reduz em muito a eficácia destes sistemas. O grau de acertos no reconhecimento é considerado bom, sendo que um treinamento mais cuidadoso pode certamente melhorar a eficácia do sistema atingindo níveis como os obtidos nos testes com o padrão A.

Quanto ao reconhecimento de padrões com distorções (padrão de teste C), onde o conjunto de reconhecimento possui diferenças significativas em relação ao conjunto de aprendizado (padrões perfeitos), a rede neural também teve um ótimo comportamento. No caso de padrões distorcidos por inversão aleatória de pixels, os resultados obtidos demonstram uma exatidão de 90 a 100% de acertos quando a rede é submetida a padrões com distorções de 5 a 20% do conjunto total de pixels deste. É claro que todos estes resultados variam conforme ajustes da configuração do simulador da rede. O Anexo A.4 - Padrão B, apresenta a avaliação realizada com este tipo de padrões gerados com uma distorção controlada dos pixels.

O sistema, na fase de reconhecimento, se mostrou pouco imune a problemas como os caracteres transladados ou rotacionados, que elevam consideravelmente as taxas de reconhecimento incorreto ou de não reconhecimento de caracteres. Este comportamento já era esperado dadas as características do sistema, que realiza um casamento direto dos padrões. No entanto, basta a realização de um aprendizado coerente com o tipo de caracteres que serão reconhecidos, e feito para um número de padrões representativo em relação aos diferentes tipos de padrões de cada classe, que o sistema terá a sua eficácia aumentada significativamente.

Uma avaliação ainda mais detalhada do comportamento do sistema, torna-se muito difícil por uma série de fatores, entre estes estão alguns já citados no capítulo 3. Um dos outros fatores que dificultou bastante a avaliação do desempenho do sistema foi o problema referente ao alto tempo consumido na simulação da rede neural, durante a fase de aprendizado. Como o simulador implementado utiliza-se de um processador único (IBM-PC com processamento seqüencial) na sua execução, e onde a rede neural simulada possui um alto grau de paralelismo (que terá que ser simulado de maneira seqüencial) isto implica um alto custo em termos do tempo gasto para o de processamento de informações. Os programas de simulação de redes neurais usualmente gastam muito tempo de processamento, e, no caso do simulador do modelo ADAn-LIENE, em simulações de complexidade média, eram consumidas em torno de 12 horas de CPU em um equipamento do tipo IBM-PC 386 compatível com co-processador aritmético e operando a 25 Mhz. Este alto tempo gasto é relativo a fase de aprendizado apenas, que é extremamente demorada devido aos ajustes e repetição contínua de ativações da rede até o aprendizado perfeito. A fase de reconhecimento pode ser executada sem problemas de demora excessiva do processamento dos dados. Devido a este fato, não foi possível a realização de testes exaustivos sobre o modelo implementado. Como tem-se um grande número de variáveis envolvidas neste processo, seriam necessárias muitas rodadas de aprendizagem para estabelecer os valores ótimos dos fatores programáveis do sistema. Estes valores ainda tem que ser determinados para cada conjunto diferente de padrões de aprendizado e reconhecimento, onde se obtém comportamentos diferentes da rede.

A avaliação do sistema, segundo o autor, deve ser realizada através do uso de uma base de dados padrão, onde se possa comparar os resultados obtidos com os resultados de outros sistemas OCR, que também foram submetidos aos mesmos testes. Somente assim será possível se ter uma noção mais exata do desempenho real do sistema.

Observando-se o comportamento geral do sistema, foi possível notar algumas de suas características que comprovam a sua versatilidade:

- Capacidade de adaptação: ajuste a novos padrões;
- Aprendizado do reconhecimento de textos com múltiplas fontes de caracteres: o número de caracteres diferentes aprendidos só é limitado pelo tamanho máximo da rede neural;
- Possibilidade de implementação em *hardware* paralelo, obtendo-se uma alta *performance* no reconhecimento;
- A utilização de simuladores permite uma maior flexibilidade no estudo, depuração e ajustes do modelo adotado, antes de sua implementação em *hardware*;
- O modelo empregado ressalta o paralelismo existente no processo de reconhecimento, facilitando a sua implementação em máquinas que permitem o processamento paralelo;
- O "salvamento" dos pesos da rede permite a criação de arquivos com o treinamento (fase crítica em termos de tempo de processamento) já realizado para determinados tipos de textos, e
- Possibilidade de extensão do sistema para adição de pós-processamento, sem maiores alterações deste.

Devido a todas estas características o sistema N^2 OCR mostra-se como uma ferramenta de alta flexibilidade, possuindo recursos que tornam este muito adequado ao reconhecimento de caracteres.

5.8 Evolução do Sistema

O sistema N^2 OCR foi implementado de maneira que este possa sofrer adaptações e expansões visando a sua melhoria. Um exemplo disto é o caso da possibilidade de adição de um módulo de pós-processamento para a correção do texto reconhecido através do emprego do contexto.

Uma outra possibilidade de expansão está relacionada com a preparação dos dados para a rede neural. Conforme foi apresentado na seção 3.4.2, os padrões utilizados pelos sistemas OCR podem sofrer um pré-processamento, visando a extração de algumas de suas características. O sistema N²OCR é um sistema que realiza o casamento de padrões, determinando, para um dado padrão desconhecido, qual o padrão de aprendizado que mais se parece com este. A implementação atual do sistema utiliza como padrão de entrada da rede neural o próprio bit-map do caractere extraído da imagem digitalizada. O sistema não possui nenhuma restrição que impeça este padrão de entrada de sofrer uma recodificação, visando ressaltar algum tipo de atributo específico ou, até mesmo, o bit-map do caractere pode ser estendido pela adição de novas informações complementares.

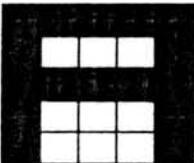
Padrão	Codificação Original	Recodificação (novo padrão)	
	1 1 1 1 1 1 0 0 0 1 =⇒ 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1	(5) 1 1 1 1 1 (2) 0 0 0 1 1 =⇒ (5) 1 1 1 1 1 (2) 0 0 0 1 1 (2) 0 0 0 1 1	1 1 1 1 1 (5) 0 0 0 1 1 (2) 0 0 0 1 1 (2) 0 0 0 1 1 (2) 1 1 1 1 1 (5)
Caracter "A"		Projeção Horizontal	Projeção Vertical

Figura 5.36 - Recodificação de um padrão

A recodificação serve para ressaltar, para a rede neural, algum tipo de característica do bit-map do caractere que pode estar "escondido" neste, e não esteja sendo levado em consideração, ou seja pouco considerado, para a obtenção do resultado do reconhecimento efetuado pela rede neural. Através da recodificação dos padrões, pode-se destacar para a rede algum tipo de propriedade específica destes, de forma que esta propriedade seja levada em consideração no reconhecimento. É necessário se ter muito cuidado ao realizar esta recodificação, pois o reconhecimento realizado

pela rede neural mede, de certa forma, apenas a distância do padrão de teste em relação aos padrões de aprendizado, indicando o mais próximo.

Para o exemplo da figura 5.36, é interessante se notar que o valor 5 foi codificado como "11111" e não em binário, que seria igual a "00101". Isto se deve ao fato do valor "00101" em binário ficar muito parecido a outros valores, como "10101" que equivale a 21, onde, apesar da similaridade entre ambos padrões ("00101" e "10101"), estes representam dois valores extremamente diferentes. Este tipo de metodologia, onde aplica-se um pré-processamento dos padrões, tem sido avaliado como sendo uma forma alternativa de funcionamento do sistema N²OCR, em um futuro próximo. A extensão do sistema N²OCR através desta técnica possibilitará que este se torne mais insensível a certos tipos de deformação como rotações ou translações dos caracteres (conforme o tipo de recodificação empregado). Desta forma, pretende-se dotar o sistema da capacidade de reconhecimento inclusive de caracteres manuscritos em letras de forma. Para isto é necessário um estudo, avaliação e determinação do melhor método de recodificação dos caracteres.

Além da expansão do sistema, outra tarefa que vem sendo realizada é a tentativa de obtenção de uma base de dados de teste padrão, que possa ser utilizada junto ao N²OCR visando a melhor validação do modelo implementado. Também tem sido realizados estudos no sentido de portar o sistema para máquinas onde seja possível a utilização de um processamento paralelo, seja com processadores fricamente acoplados (por exemplo, uma rede de workstations SUN) ou através de processadores fortemente acoplados (por exemplo, um hardware baseado em Transputers). Cria-se desta forma um simulador que executa suas tarefas em paralelo, realizando uma parte do processamento junto a cada processador.

O sistema N²OCR também propiciou, como resultado de seu desenvolvimento, um estudo mais aprofundado das rede neurais. Este estudo resultou na criação de um pro-

jeto denominado SIRENE - Simuladores de Redes Neurais. Este projeto garante a continuidade dos estudos realizados sobre este tema, e visa a implementação de ferramentas para a análise e utilização de modelos de redes neurais de destaque na atualidade, assim como permite investigações com o objetivo de criar novos modelos de redes neurais. O projeto SIRENE está ligado ao grupo de estudos de arquiteturas avançadas de computadores (processamento paralelo), coordenado pelo prof. Philippe O. A. Navaux, junto ao CPGCC da UFRGS. Além deste projeto, que se encontra em pleno desenvolvimento, os estudos realizados para o desenvolvimento deste trabalho também tem sido utilizados para colaborar com o desenvolvimento de um "processador neural", que está sendo feito sob a coordenação do prof. Paulo M. Engel.

Em resumo, o estudo realizado neste trabalho abriu uma série de novas perspectivas de estudo dentro das linhas de pesquisa sobre redes neurais e reconhecimento de caracteres. O sistema N^2 OCR permite também uma evolução, sendo possível a realização de novos estudos para complementar o trabalho que já foi desenvolvido, a nível de avaliação do desempenho, estudo de convergência, estimativa de parâmetros ótimos de configuração, pós-processamento do texto reconhecido, etc.

6 Conclusão

Este trabalho teve por objetivo o estudo do reconhecimento visual de caracteres através da utilização das redes neurais. Para atingir tal objetivo foram desenvolvidos os tópicos referentes ao Processamento Digital de Imagens, aos sistemas de reconhecimento de caracteres (OCR), às redes neurais e por fim foi criada uma proposta de solução e implementação de um sistema OCR, o denominado sistema N²OCR. Para validar esta proposta de implementação foi desenvolvida uma série de ferramentas que implementam as diversas operações e etapas de processamento estudadas neste trabalho, criando assim um protótipo do sistema N²OCR.

Os resultados obtidos na utilização do protótipo do sistema comprovam que o N²OCR não é apenas viável, como também demonstram ser este uma ótima alternativa como sistema para reconhecimento de caracteres impressos. Este sistema demonstra uma série de propriedades que tem sido apresentadas como muito desejáveis, senão essenciais, em sistemas OCR.

O trabalho demonstrou-se muito válido também pelos seus reflexos, como foi o caso da criação do projeto SIRENE no CPGCC da UFRGS. Outro aspecto importante foi o fato deste não ter sido um trabalho limitado, pois permite expansões que podem melhorar ainda mais o seu desempenho, chegando até possivelmente ao reconhecimento de caracteres manuscritos.

Além de incentivar o desenvolvimento de novas frentes de estudos junto ao CPGCC da UFRGS, este trabalho também possibilita que se chegue a resultados com aplicações práticas, não somente para fins científicos. O sistema desenvolvido pode ser aperfeiçoado, através da complementação de alguns módulos e da implementação em hardware de uma arquitetura paralela que realize as funções executadas através do simulador da rede neural, chegando ao nível dos sistemas comerciais de OCR.

É importante salientar que este trabalho, por tratar de uma área pouco estudada em termos de Brasil, e principalmente em relação ao CPGCC da UFRGS, buscou a formação de uma base de conhecimentos sobre o assunto. Desta forma todo o trabalho serve para dar subsídios para novos trabalhos que possam vir a ser desenvolvidos em áreas relacionadas com as que foram abordadas.

ANEXOS

ANEXO A.1 - Padrões de Descrição de Imagens

1. Formato BTM

O formato BTM tem seu nome originário da abreviação de *bit-map*. Este formato decreve a imagem através do armazenamento direto do *bit-map* que compõe a imagem, sendo usado apenas para armazenar imagens P&B (2 cores). Os arquivos no formato BTM tem uma extensão *.BTM no seu nome e seguem a descrição dada abaixo. Estes arquivos são os que mais se adaptam ao sistema N²OCR.

*Formato dos Arquivos tipo *.BTM:*

Imagen → Bloco de bytes com 1 ou 8 pixels por byte. A descrição da imagem começa no canto superior esquerdo, sendo descrita linha à linha, da esquerda para a direita, de cima para baixo. Os últimos 7 bytes do arquivo são o cabeçalho final.

Cabecalho Final → ASSINATURA (1 byte)
 NUMLIN (2 bytes)
 NUMCOL (2 bytes)

ASSINATURA : 0xA5 (165) → 1 bit por pixel (8 pixels/byte)
 0x5A (90) → 8 bits por pixel (1 pixel /byte)

NUMLIN : Nro. inteiro com 2 bytes na ordem invertida, LSB e após o MSB. Indica o número de linhas da imagem.
 LSB = Byte menos significativo (Least Signif.)
 MSB = Byte mais significativo (Most Signif.)

NUMCOL : Nro. inteiro com 2 bytes na ordem invertida LSB e após o LSB. Indica o numero de colunas da imagem.

2. Formato FIM

O formato FIM tem seu nome originário da abreviação de *File of Image*. Este formato decreve a imagem através do armazenamento direto do bit-map que compõe a imagem, sendo usado para armazenar imagens de até 256 níveis (escala de cinza). Os arquivos no formato FIM tem uma extensão *.FIM no seu nome e seguem a descrição dada abaixo.

*Formato dos Arquivos tipo *.FIM:* São idênticos aos arquivos *.BTM em termos estruturais. Possuem mais de dois níveis de cores (escala de cinza). É obtido, geralmente, desfazendo-se o halftoning de um arquivo BTM. Vide a descrição dos arquivos BTM.

3. Formato CUT

O formato CUT tem seu nome originário da palavra "recorte" em inglês (é um pedaço de uma imagem). Este formato decreve a imagem através do armazenamento de uma codificação tipo RLE, sendo usado para armazenar imagens de até 256 níveis (escala de cinza). Os arquivos no formato CUT tem uma extensão *.CUT no seu nome e seguem a descrição dada abaixo. Estes arquivos são utilizados originariamente pelo scanner DFI HS-3000 e podem ser manipulados pelo softwares de editoração eletrônica HaloDPE ou Dr. HALO.

*Formato dos Arquivos tipo *.CUT:*

Imagen → É descrita linha a linha, da esquerda para a direita, de cima para baixo. As linhas são compactadas pelo método RLE. A imagem é composta por um cabeçalho inicial seguido de vários blocos (cada bloco descreve uma linha de varredura da imagem).

Cabeçalho: Tamanho Total → Número total de colunas e número total de linhas (6 bytes)

Xl Xh Yl Yh 00 00 (6 bytes → Xl=LSB, Xh=MSB,
Yl=LSB, Yh=MSB, 00=Valor zero)

Blocos: Descrevem uma linha de varredura da imagem.

NroBytes (LSB) + NroBytes (MSB) → 2 bytes

Descrição dos bytes → Tem um total de bytes igual a 256 . NroBytes (MSB) + NroBytes (LSB)

Descrição dos bytes → Total Cor [Cor Cor ...] 00

Se Total tem o bit 8 ligado

Então repete Cor Total vezes (Codificação RLE)

Senão segue Total bytes, após Total, descrevendo as cores

4. Formato IMF

O formato IMF tem seu nome originário da abreviação de *Image File*. Este formato descreve a imagem através do armazenamento das diferentes informações que compõem esta, o *bit-map*, a tabela de *palette* (se houver) e uma descrição mais detalhada das características da imagem. É usado apenas qualquer tipo de imagem. Os arquivos no formato IMF tem uma extensão *.IMF (Imagem), IMD (Descrição) e IMP (Palette) no seu nome, e seguem a descrição dada abaixo. Este formato de arquivos é um padrão criado pelo grupo de Computação Gráfica do CPGCC da UFRGS para o manuseio de imagens.

*Formato dos Arquivos tipo *.IMF:*

Arquivo: Composto por 2 blocos principais

Cabeçalho + Dados Gráficos (imagem)

Cabeçalho: É todo descrito através de um texto ASCII, sendo terminado por um caracter delimitador, este caracter é o Ctrl Z (^Z - 1Ah) utilizado usualmente para delimitar o fim de arquivos de texto. Desta forma é possível listar o arquivo na tela e ler diretamente a descrição contida no cabeçalho deste. O cabeçalho é composto pelos seguintes campos (separados entre si por um caracter '\$' - 24h):

Tamanho	'\$'	Tipo	'\$'	Compactação	'\$'	NumCol	'X'	NumLin	'\$'
'\$'	'\$'	Bits/Pixel	Palette	'#'	Comentários	'#'	^Z (Ctrl Z)		
'\$'									

1. Tamanho: 1 à N dígitos em ASCII, indicando o tamanho do cabeçalho. Tamanhos aconselhados : 256 ou 512 bytes.
2. Tipo: Campo representado por um caracter ASCII que pode assumir os seguintes valores - '0', '1', '2' ou '?'. Descreve o tipo de descrição da imagem que é utilizado no arquivo. Descrição dos tipos disponíveis:
 - 0 - Ajuste a byte. Os pixels são ajustados de forma a ocuparem sempre um número inteiro de bytes. Neste tipo de arquivo não ocorrerá de 2 pixels ficarem representados total ou parcialmente em apenas um byte (dois pixels não são guardados juntos em um byte).
 - 1 - Não ajusta a byte, faz ajuste por linha. Este tipo de arquivo permite que dois pixels sejam armazenados juntos em um mesmo byte, mas no caso de pixels pertencentes a duas linhas de varredura diferentes, este tipo de caso não será aceito. Isto implica que cada nova linha de varredura da tela começa a ser descrita em um novo byte, sendo que o último byte utilizado na descrição de uma linha pode ficar com alguns bits não usados.

- 2 - Não ajusta a byte, não faz ajuste a linha. Este tipo de arquivo utiliza um descrição da imagem utilizando linearmente todos os bits de um byte, sem restrições ao fato de um pixel ocupar todo ou parte de um byte, juntamente ou não com outros pixels.
- ? - Arquivo compactado. Os arquivos compactados não possuem especificação de ajuste uma vez que a imagem está compactada e sua descrição original foi alterada.
3. Compactação: Indica o método de compactação da imagem (*raster image*) que foi empregado. Pode assumir os seguintes tipos:
- 'SEM' - Não foi realizada nenhuma compactação, o bloco da imagem descreve diretamente cada ponto da imagem.
 - 'RLE' - Método de compactação RLE - *Run Length Encoding*. Este método deverá ser descrito de forma mais detalhada pelo grupo definidor do padrão IMF, em um futuro próximo.
 - 'LZW' - Método de compactação LZW. Este método segue a descrição do algoritmo LZW encontrada na literatura. Uma especificação mais detalhada do mesmo deverá ser apresentada pelo grupo definidor do padrão IMF, em um futuro próximo.
4. NumCol: Número de colunas da imagem do tipo *raster*. É descrito através de um conjunto de dígitos ASCII, indicando o número total de colunas da imagem. A imagem possuirá colunas numeradas de 0 a NumCol-1. O campo NumCol será separado do campo NumLin por um caracter ASCII 'X'.
5. NumLin: Similar a NumCol, mas indicando o numero total de linhas da imagem.

6. Bits/Pixel: Número de bits úteis por pixel. Se for uma imagem P&B teremos 1 bit/pixel, no caso de uma imagem de 256 tons de cinza teremos 8 bits/pixel.
7. Palette: Indica o nome do arquivo de *palette*, caso este seja necessário. Este campo poderá conter o nome do arquivo de *palette* ou as palavras pré-definidas - 'RED' (escala de vermelho), 'GREEN' (escala de verde), 'BLUE' (escala de azul), ou 'GREY' (escala de cinza). Caso o campo seja nulo, ou seja contenha a palavra 'NULO' ou '', é assumida a utilização de uma escala de cinza.
8. Comentários: O campo de comentários é delimitado no seu início e fim pelo caracter ASCII '#'. Este campo é reservado para qualquer tipo de comentários, sendo usado para apresentar uma descrição textual mais informal da imagem. Existem uma série de dados que são indicados para serem colocados nos comentários, são eles:
 - Assinatura: Texto utilizado para a verificação do arquivo, indicando que este realmente é um arquivo de imagem;
 - Origem da imagem: Forma como foi adquirida (câmera, scanner, etc). Local, data e responsável pela obtenção da imagem;
 - Tipo de equipamento empregado para a exibição da imagem: indicar o equipamento, por exemplo, IBM-PC, SUN, etc; e se necessário a placa gráfica empregada, por exemplo, CGA, EGA, VGA, Targa, etc;
 - Tipo de tratamento da imagem: Indicar se esta imagem já sofreu algum tratamento, decretendo o nome do algoritmo empregado, parâmetros e demais informações relevantes;
 - Descrição de informações referentes à imagem: indicar informações importantes, como por exemplo, imagem P&B mas com N tons gerados por *halftoning*, imagem da banda

de infra-vermelho capturada por um satélite, tipo de valores descritos (tons de vermelho, verde, azul ou cinza), etc.

Todos os programas que forem desenvolvidos para a manipulação de arquivos do tipo IMF, devido ao grande número de variações possíveis na combinação dos parâmetros de definição do arquivo, devem oferecer sempre uma rotina de leitura e escrita padrão que aceite a descrição de um arquivo tipo IMF, com os seguintes dados:

- Tipo = 0
- Compactação = 'SEM'
- Bits/Pixel = Máximo de 8 bits/pixel
- Os outros itens são livres.

*Formato dos Arquivos tipo *.IMD:*

Arquivo: Este arquivo permite a descrição completa de uma imagem, associando diferentes descrições isoladas em um único arquivo. Através do arquivo IMD é possível agrupar, por exemplo, descrições isoladas das bandas de vermelho, verde e azul, ou, associar a imagem a um determinado tipo de *palette*. Este arquivo possui o seguinte formato:

#Desc	'\$'	Tamanho Descri.1	'\$'	Nome da Imagen	'\$'	Nome do Palette	'\$'	Comentários	...
-------	------	---------------------	------	-------------------	------	--------------------	------	-------------	-----

1. #Desc: Número de descriptores de imagens.
2. Tamanho Descr.1: Tamanho do descritor 1 da imagem.
3. Nome da Imagem: Nome do arquivo que contém a imagem.
4. Nome do Palette: Nome do arquivo que contém o *palette* associado a uma dada imagem.
5. Comentários: Informações adicionais.

O descritor é composto por uma série de componentes descrevendo o tamanho do descritor, nome da imagem, nome do *palette* e comentários.

No caso de uma imagem colorida, com 24 bits/pixel, onde são utilizados 8 bits por cada banda RGB, o descritor terá 3 blocos, descrevendo o nome dos três arquivos que completam a imagem. Neste caso, cada arquivo descreve uma banda da imagem (8 bits/pixel).

*Formato dos Arquivos tipo *.IMP:*

Arquivo: Este arquivo descreve a tabela de *palette* associada a uma dada imagem. Possui o seguinte formato:

Tamanho	'\$'	Nro. Cores	'\$'	Bits por Componente	'\$'	Nro. de Compon.	'\$'	Palette
---------	------	------------	------	---------------------	------	-----------------	------	---------

1. Tamanho: Tamanho do cabeçalho
2. Nro. Cores: Número total de cores que compõem a tabela de *palette*, equivale ao número de cores que podem ser exibidas simultaneamente no dispositivo.
3. Bits por Componente: Número de bits utilizados por cada componente da tabela de *palette*. Cada componente ocupa sempre 1 byte, podendo sobrar bits não utilizados.
4. Nro. de Componentes: Número total de componentes da tabela de *palette* utilizados para descrever um pixel da imagem.
5. Palette: Descrição byte a byte de cada componente que forma cada uma das entradas da tabela de *palette*.

ANEXO A.2 - Padrões de Descrição de Caracteres

1. Caracteres Isolados

Este formato é gerado como resultado da utilização do módulo de segmentação de caracteres. O programa de segmentação parte de uma imagem tipo *bit-map*, gerando um arquivo conforme o formato descrito abaixo:

```
#<Rótulo> <Nro. Colunas> <Nro. Linhas>
0 e 1's ...
#<Rótulo> <Nro. Colunas> <Nro. Linhas>
0 e 1's ...
...
...
```

Onde:

- O arquivo gerado é um arquivo em formato texto.
- *Rótulo* : É um valor numérico descrevendo o código do caractere no sistema de representação adotado (ASCII, EBCDIC, etc) e que serve para a identificação do caractere, possibilitando o seu aprendizado. O tipo de código de representação adotado se encontra descrito no arquivo de configuração do sistema N²OCR. O rótulo pode conter um valor especial, também descrito no arquivo de configuração, que indica um caractere que não foi identificado (caractere a ser reconhecido posteriormente).
- *Nro. Colunas* : Indica o total de colunas que formam o padrão que será descrito logo abaixo no arquivo.
- *Nro. Linhas* : Indica o total de linhas que formam o padrão que será descrito logo abaixo no arquivo.
- Segue-se a indicação do rótulo, nro. de linhas e colunas de um dado padrão; um conjunto de dígitos '0' e '1' que representam os bits respectivamente desligados e ligados do caractere que foi isolado.
- Cada caractere isolado é descrito conforme o padrão especificado acima.

Exemplos:

#65 6 4	← Caracteres	#0 6 4
111111	Conhecidos	111111
100001		100001
111111	Caracteres	111111
100001	Desconhecidos →	100001
#66 8 5		#0 8 5
11111100		11111100
11001100		11001100
11111111		11111111
11000011		11000011
11111111		11111111

2. Caracteres com Tamanho Padronizado

Os caracteres que foram isolados sofrem um ajuste de escala de forma que todos possuam a mesma dimensão e possam ser fornecidos a rede neural, seja para o aprendizado ou reconhecimento. O programa que faz o ajuste de escala recebe os arquivos no formato descrito no item anterior e produz como saída arquivos com o seguinte formato:

```
$<Nro. Colunas> <Nro. Linhas> <Nro. Padrões> <Tipo>
0 e 1's ...
<Rótulo>
0 e 1's ...
<Rótulo>
...
!
```

Onde:

- O arquivo gerado é um arquivo textual.
- *Nro. Colunas* : Indica o total de colunas que formam todos os padrões que estão descritos no arquivo.
- *Nro. Linhas* : Indica o total de linhas que formam todos os padrões que estão descritos no arquivo.

- *Nro. Padrões* : Indica o número total de caracteres que se encontram descritos no arquivo.
- *Tipo* : É um identificador do tipo de arquivo utilizado, na versão atual do sistema seu valor é usualmente zero. Este item pode ser omitido.
- Cada caracter é descrito através de um conjunto de dígitos '0' e '1' que representam os bits respectivamente desligados e ligados do caracter; seguido de sua identificação.
- *Rótulo* : É um valor numérico descrevendo o código do caracter no sistema de representação adotado (ASCII, EBCDIC, etc) e que serve para a identificação do caracter, possibilitando o seu aprendizado. O tipo de código de representação adotado se encontra descrito no arquivo de configuração do sistema N²OCR. O rótulo pode conter um valor especial ('?') que indica um caracter que não foi identificado (caracter a ser reconhecido posteriormente).
- O caracter '!' indica o fim do arquivo.
- Os arquivos que possuem caracteres que foram identificados e que serão utilizados na fase de aprendizado tem usualmente a extensão *.LET, e os arquivos que possuem caracteres não identificados e que serão utilizados para o reconhecimento tem usualmente a extensão *.TST.

Exemplos:

\$8 5 2 0	← Caracteres	\$8 5 2 0
00000000	Conhecidos	00000000
11111100		11111100
10000100		10000100
11111100	Caracteres	11111100
10000100	Desconhecidos →	10000100
65		?
11111100		11111100
11001100		11001100
11111111		11111111
11000011		11000011
11111111		11111111
66		?
!		!

A.3 - Padrões de Teste Empregados

1. Padrão A - Handy Scanner DFI HS-3000

» Exemplo de caracteres isolados e utilizados nos testes com o padrão A (Anexo A.4 - Padrão A).

#63 13 22	#63 14 22
0000011000000	00000111100000
0001111110000	00011111111000
0011111111000	001111111111000
0111100011100	01111000011100
0111000001100	00100000001110
1110000001110	00000000001110
1110000000110	000000000001110
1110000000110	0000000000001110
1110000000110	00000000000001110
1110000000110	000000000000001110
1110000000110	0000000000000001110
1110000000110	00000000000000001110
1110000000110	000000000000000001110
0111000011111	00000111111100
0111100111111	000001111111100
0011111111111	00000000011110
0001111110111	00000000001110
0000000000110	000000000000110
0000000000110	0000000000000111
0000000000110	0000000000000111
0000000001100	01000000001110
0000000011100	11110000011110
0110011111000	11111101111100
0111111100000	0011111111100
0011110000000	00011111110000
#63 12 20	#63 12 21
1111111111100	000000111111
1111111111111	000011111111
1111111111111	000111100000
110000000111	001110000000
010000000111	011100000000
000000000110	011000000000
000000000110	111000000000
0000000001110	111000000000
0000000001100	110011111100
0000000001100	1111111111100
0000000001100	111110001110
0000000001100	111100000111
0000000001100	111000000011
0000000001100	1110000000011
0000000001100	011000000011
0000000001100	0110000000011
0000000001100	011000000011
0000000001100	001100000011
0000000001100	000111001110
0000000001100	000011111100

» Exemplo de caracteres isolados e utilizados nos testes com o padrão A (Anexo A.4 - Padrão A).

#63 14 21

00001111110000
00111111110000
01111000011100
00100000001100
00000000001110
00000000001110
00000000001110
000000000011100
000001111111000
000001111111000
000000000111100
000000000011100
000000000001110
000000000000110
0000000000000111
0000000000000110
111000000011100
01111100011100
0001111111110000
0000011111000000

#63 13 19

00000001111000
00000001111000
00000111111000
00001111111000
0000110011000
0001110011000
0011100011000
0011000011000
0111000011000
0110000011000
1110000011100
11111111111110
01111111111110
0000000011100
0000000011100
00000000111110
00000011111111
00000011111111

#63 13 20

```
00000000111110  
0000011111110  
0001111000000  
00111000000000  
001100000000000  
011100000000000  
011000000000000  
1110011110000  
11111111111000  
1111110011100  
1111100001110  
0111000000110  
0111000000111  
0110000000011  
01100000000011  
0011000000111  
00110000000110  
0001100001110  
0000111111100  
0000011111100
```

#63 12 21

000011110000
001111111000
011111111100
011100001110
111000000110
111000000110
111000000111
111000000111
111000001110
011110011110
001111111100
011111111100
011100001110
111000000110
110000000111
110000000111
110000000111
111000000110
111100001110
011111111100
001111111100

» Exemplo de caracteres com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão A (Anexo A.4 - Padrão A).

0001111100000000	0000010000000000
0011111110000000	0011111111000000
1110000011100000	0111000111000000
1100000001100000	1110000011100000
11000000001100000	1100000001110000
11000000001100000	11000000001110000
11000000011100000	11000000001110000
11000000011110000	11000000001110000
01100000111100000	01100000111100000
00111111111100000	01110001111100000
00011110011100000	00011110011100000
00000000011000000	00000000001100000
00000000011100000	00000000001110000
00000000111000000	00000000001110000
00000011100000000	00000011110000000
11111110000000000	01111111000000000
00110000000000000	00111100000000000
57	57

0000111100000000	0000011100000000
0001111110000000	0011111111000000
0110000011100000	0111000011100000
01100000011100000	01100000011100000
11100000011100000	01100000011100000
01100000011100000	01100000011100000
0111000011100000	01110000011100000
0111100111000000	0011111111000000
0111111111000000	0011111111000000
0111000011100000	01100000011100000
1100000001110000	1110000000110000
1100000000110000	1100000000110000
11000000001110000	0110000000110000
1110000000110000	0111000000110000
0111111111000000	0011111111000000
0001111111000000	00000110000000000
56	56

» Exemplo de caracteres com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão A (Anexo A.4 - Padrão A).

0000001111110000
 0000011111110000
 0011000000000000
 0011000000000000
 0111000000000000
 1110000000000000
 1110111110000000
 1111111111000000
 1111000000110000
 1110000000110000
 0110000000110000
 0110000000110000
 0110000000110000
 0011000000110000
 0001111111000000
 0000011111000000

54

0000000111110000
 0000011111110000
 0001111000000000
 0011100000000000
 0011100000000000
 0110000000000000
 0110000000000000
 0111111111000000
 1111100001110000
 01110000001110000
 011100000001110000
 011100000001110000
 00110000001110000
 0001111111000000
 0000011111000000

54

0000000111000000
 0000000111100000
 0000011111100000
 0000011011000000
 0000110011000000
 0001110011000000
 0011000011000000
 0011000011000000
 0110000011000000
 1110000011100000
 0111111111110000
 0000000011100000
 0000000011100000
 0000011111110000
 0000001111110000

52

00000001111000000
 00000001111000000
 0000011111000000
 0000111011000000
 0001110011000000
 0011100011000000
 0011100011000000
 0111000011000000
 1110000011100000
 1111111111110000
 1111111111110000
 0000000111000000
 0000000011000000
 0000011111110000
 0000011111110000

52

» Exemplo de caracteres não identificados utilizados no reconhecimento realizado nos testes com o padrão A (Anexo A. 4 - Padrão A).

0000011000000000	0000111100000000
0011111111000000	0111001110000000
0111100011100000	0111000011100000
0111000001100000	1110000001100000
1110000000110000	1100000000110000
1110000000110000	1100000000110000
1110000000110000	1110000001110000
0111000011111000	0111000011110000
0111100111111000	0011111111110000
0001111101111000	0000111100110000
0000000000110000	0000000000110000
0000000000110000	0000000000110000
00000000011100000	00000000011100000
00000000111000000	0000001111000000
0110011111000000	0111111110000000
0011110000000000	0011100000000000
?	?

0000001000000000	0000111100000000
0001111111100000	0111001111000000
0011100011110000	0111000011100000
0111000000110000	1110000001100000
0110000000110000	1100000001100000
0110000000110000	1110000001100000
0011100001110000	0110000011100000
0001111111100000	0011111111000000
0011111111100000	0111111111000000
0111000000110000	1110000001100000
1110000000111000	1100000000110000
0110000000011000	1100000000110000
0110000000111000	1110000000111000
0011100011110000	0111000001100000
0011111111100000	0011111111100000
0000001100000000	0000011100000000
?	?

» Exemplo de caracteres não identificados utilizados no reconhecimento realizado nos testes com o padrão A (Anexo A. 4 - Padrão A).

00000001111000000	0000000111000000
00000001111000000	0000001111000000
0000111111000000	0000011111100000
0000111011000000	0000111111100000
0001110011000000	0001110011000000
0001100011000000	0011100011000000
0111000011000000	0011000011000000
0110000011000000	0110000011000000
1110000011000000	1110000011000000
1110000011000000	1111111111110000
1111111111110000	0011111111110000
0000000111100000	0000000011000000
0000000011000000	0000000011110000
0000000011000000	0000011111110000
000000011111110000	0000011111110000
000000011111110000	0000001111110000
?	?

0000000111100000	0000000111111000
0000111111110000	0000111100011000
000111100000000000	000111000000000000
011100000000000000	011100000000000000
011100000000000000	011100000000000000
111000000000000000	011100000000000000
111111111111000000	111111111111000000
111111111111000000	111111001111000000
1111000000110000	0111000000111000
0110000000110000	0111000000011000
0110000000110000	0111000000111000
0110000000110000	0011000000111000
0111000000110000	0011100000111000
000111111111000000	000111111111000000
000111111111000000	000001111111000000
000001111111000000	000000011100000000
?	?

2. Padrão B - Gerador de Imagens (Programa Padrão)

» Exemplo de caracteres com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão B (Anexo A.4 - Padrão B).

101111111110000	0000111100000000
011111111110000	0000111100000000
1111000000111100	0011111100000000
1111000000111100	0011111100000000
1111100000111100	0000111100000000
1111010000111100	0000111100000000
1111001000111100	0000111100000000
1111000100111100	0000111100000000
1111000010111100	0000111100000000
1111000001111100	0000111100000000
1111000000111100	0000111100000000
1111000000111100	0000111100000000
001111111111000	1111111111110000
0011111111110100	1111111111110000
0000000000000000	0000000000000000
0000000000000000	0000000000000000
48	49

» Exemplo de caracteres não identificados utilizados no reconhecimento realizado nos testes com o padrão B (Anexo A.4 - Padrão B). Os caracteres possuem uma distorção (inversão de pixels) em torno de 15%.

101111111110000	0010101000000000
011111110110000	0000111100000000
1111000001111100	0011111100000000
1111000000111100	0011111100100000
1111100001111100	1000111110000000
1111010000111100	0000011100000000
1111101000101100	1000111101000000
1111000000111100	0000111100010000
1111000001011100	0000111100000000
1110100001111100	0000111100000000
1111000000111100	0000111100000000
1111000000111100	0000111100000000
0011111011111000	1111111111010000
00111111111100100	0111111111110000
0000010000000000	0000000000000000
0000000000000000	0000000000000000
?	?

» Exemplo de caracteres não identificados utilizados no reconhecimento realizado nos testes com o padrão B (Anexo A. 4 - Padrão B). Os caracteres possuem uma distorção (inversão de pixels) em torno de 10%.

0011101101110000	0000110101000100
0111111111110000	1100111100000000
1111000010011100	0001111100000000
11110100001111000	0010110110000000
1111100000111100	0000111101010000
1111010010111100	0000010100000000
1000101000111100	1000101100000000
1111000000111100	0000111100000000
1111000011111100	0000101100000000
1111001011111100	0000111100000000
1111000010111100	0100111100000000
1111000000011100	0000111100000100
0001111111111100	1101111111110000
0011111111110100	1111111110100000
00000000000101000	0001000100000000
00000000000000000	0000000000000000
?	?

» Exemplo de caracteres não identificados utilizados no reconhecimento realizado nos testes com o padrão B (Anexo A. 4 - Padrão B). Os caracteres possuem uma distorção (inversão de pixels) em torno de 5%.

1111101110110000	0000111000000000
0111101111110000	0000011100000000
1101000100011100	0010110000000000
11110100001111000	011111000111010
11011000001111100	0000111001110100
1001010000101100	0000111100000100
1111101000101110	0000101100000100
11010001001111010	0001111000101000
11110000100111100	0100111101000100
11100000001111100	0000011100000010
11100000001111110	0000111100000010
11111000001111100	0000111100100100
0101111001111000	1111111111110010
00101010111110100	1111111111110010
0000001101000000	0000100100000000
00000000000000000	0000000000000000
?	?

3. Padrão C - Correio Inglês

» Exemplo de caracteres numéricos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C).

0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000
000001111100000	111000011110000	0000000110000001
000011111110000	110000111111000	000011111110001
000111111111000	1000011100111000	000011111111000
000111000001100	0000110000011100	001111011111100
0011100000001100	0001110000001110	0111110000111100
0011000000001100	0001100000001100	0111100000001100
00110000000001100	0011100000001110	0111000000000110
001100000000001100	0011100000000110	01110000000000110
001100000000000110	00111000000000110	011100000000000110
0011000000000000110	001110000000000110	0111000000000000110
00110000000000000110	0011100000000000110	01110000000000000110
001100000000000000110	00111000000000000110	011100000000000000110
0011000000000000000110	001110000000000000110	0111000000000000000110
00110000000000000000110	0011100000000000000110	01110000000000000000110
001100000000000000000110	00111000000000000000110	011100000000000000000110
0011000000000000000000110	0001100000001100	00111000000000001100
0001110000001100	00001110000011100	00111000000000001100
0001111000011000	000001110000011100	00111000000000001100
0000111100011000	0000001111000000	000111001111000
0000011111100000	0000000000000000	0000111111110001
00000000010000000	0000000000000000	0000000000000000
00000000000000000	0000000000000000	0000000000000000

» Exemplo de caracteres numéricos utilizados no testes com o padrão C. Estes caracteres são os mesmos apresentados no exemplo anterior, mas estão representados de forma a facilitar a sua visualização e a percepção de pequenos detalhes. Os exemplos seguintes serão todos apresentados desta forma (O sistema N²OCR utiliza a outra codificação).

A diamond-shaped pattern of stars, with a total of 37 stars arranged in a symmetrical design.

A decorative border featuring a grid pattern. The intersections of the grid are marked with small black stars. The grid lines themselves are represented by thin black dots. The pattern is symmetrical and covers the entire page area.

A decorative border consisting of a grid of small black stars and dots. The stars are arranged in a repeating pattern of horizontal rows, with some rows containing only stars and others containing both stars and dots. The dots are placed at the intersections of the star rows, creating a mesh-like effect. The border is contained within a rectangular frame.

A decorative border consisting of a grid of stars and dots. The border is approximately 10 units wide and 10 units high. It features a repeating pattern of a single star surrounded by nine dots, creating a larger star-like shape. The border is centered on a white background.

» Exemplo de caracteres numéricos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C). Continuação ...

A decorative border consisting of a grid of small black stars and dots. The stars are arranged in a repeating pattern of horizontal rows, some containing only stars and others containing both stars and dots. The dots are small black circles placed between the stars. The border is contained within a rectangular frame.

A decorative border consisting of a repeating pattern of stars and dots. The pattern includes various star shapes (solid, outlined, and dashed) and horizontal dotted lines, all enclosed within a rectangular frame.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of single stars, pairs of stars, and groups of three stars. The dots form the background and the boundaries between the star groups.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of horizontal rows, with some rows containing more stars than others. The dots are small circles placed between the stars. The border is contained within a rectangular frame.

A decorative border consisting of a grid of small stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical lines, creating a diamond-like mesh. The dots are smaller and fill the spaces between the stars.

» Exemplo de caracteres numéricos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C). Continuação ...

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of horizontal rows. Each row contains a sequence of stars: a single star followed by a row of five stars, then another single star, and so on. The dots are represented by small circles that form the gaps between the star groups and the outer edges of the border.

A decorative border consisting of a grid of small black stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical lines, creating a scalloped or wavy effect along the edges. The dots are smaller and fill the spaces between the star points.

A decorative border consisting of a grid of small stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical lines, creating a diamond-like mesh. The dots are placed at the intersections of these star lines.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of single stars and pairs of stars. The dots are small circles that form the background of the border.

» Exemplo de caracteres numéricos não identificados utilizados no reconhecimento realizado nos testes com o padrão C (Anexo A.4 - Padrão C).

The image shows a grid of three columns, each consisting of a series of rows of stars and question marks. The first column has 15 rows, the second has 16, and the third has 17. Each row contains either a single star or a single question mark, with the pattern alternating between them. The columns are separated by vertical dotted lines.

» Exemplo de caracteres alfabéticos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C). Os caracteres estão representados em sua codificação original, na forma em que são utilizados junto ao sistema de reconhecimento.

0000000000000000	0010000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000
0000000110000000	0000000000000000	0000000110000000
0000001111000000	0000000000000000	0000000110000000
0000011111000000	0000000010000000	1000000110000000
0000011111100000	0000000011100001	1000000111100000
0000011111100000	00000000111100001	1000000111100000
0000011111100000	00000000111110001	1100000111100000
00000111111100000	10000001111100000	1000000111100000
0000011011110000	10000011111100000	1000011011000000
00000111001110000	10000011111100000	1000011111000000
00000111001110000	10000011111100000	0000111011100000
00000111111100000	100000110001110000	0000111111110000
0001111111110000	000001100001110000	0000111111110000
0011111111111000	00000111111110000	0000111111111000
0011100000011100	00011111111110000	0000110001110000
01111000000011000	0001110011110000	0001110001110000
0111000000011100	0011110001110000	0011110001111000
00111000000011100	1111110001111100	1111111001111100
0011000000011100	1111111011111111	0111111111111100
0010000000011100	1001111001111111	0000100000011100
00000000000000000	00000000000000000	00000000000000000
00000000000000000	00000000000000000	00000000000000000
97	97	97
0000000000000000	0010000000000000	0010000000000000
0000000000000000	0000000000000000	0000000000000000
0011010111100000	0000000000000000	0011101011000000
0111111111110000	0111111111110000	00111111111110011
0010111111111000	0001111111111000	0000011011111011
0000111100011100	0000110000011100	0000001000001100
0000111000011100	0000110000001100	0000011000001100
0000011000011100	00000110000001100	00000011000001100
00000111000011100	000001100000001100	00000011000001100
0000011111111100	0000011111111100	00000011111110000
00000111111111100	00000111111111100	0000001001111000
0000010000011110	00000110000001110	00000010000001100
00000111000001110	000001100000001110	00000011000000110
00000111000000110	000001100000001110	000000110000001100
00000111000000111	000001100000001110	00000011000000110
00000111000000110	000001100000001110	000000110000001100
000001110000001110	000001100000001110	000000110000001100
000001110000001110	000001100000001110	000000110000001100
0111111111111100	0001111111111100	00000110000011100
01111111111111100	0111111111111100	01111111111111111
011111111111111000	00011111111111000	011110111110111
00000000000000000	00000000000000000	00000000000000001
00000000000000000	00000000000000000	00000000000000000
98	98	98

» Exemplo de caracteres alfabéticos utilizados no testes com o padrão C. Estes caracteres são os mesmos apresentados no exemplo anterior, mas estão representados de forma a facilitar a sua visualização e a percepção de pequenos detalhes. Os exemplos seguintes serão todos apresentados desta forma (O sistema N²OCR utiliza a outra codificação).

The image shows a large triangular arrangement of stars, divided into three vertical columns by dotted lines. The stars are represented by small black asterisks. The left column contains the number '97' at its base. The middle column contains the number '97' at its base. The right column contains the number '97' at its base. The top center of the triangle has a single star.

» Exemplo de caracteres alfabéticos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C). Continuação ...

A decorative border consisting of a grid of small stars and dots. The stars are arranged in a repeating pattern of vertical columns and horizontal rows. The dots are smaller circles placed between the stars. The border is oriented vertically on the left side of the page.

A decorative border consisting of a grid of small black stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical rows, with some stars missing to create a mesh-like effect. The border is contained within a rectangular frame.

A decorative border consisting of a grid of small black stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical rows, with some stars being larger than others to create a sense of depth or emphasis. The border is contained within a rectangular frame.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of vertical columns and horizontal rows. The dots are small circles placed between the stars. The border is contained within a rectangular frame.

A decorative border consisting of a grid of stars and dots. The pattern is composed of a central area of stars with a dot in the center, surrounded by concentric layers of stars and dots. The stars are arranged in a staggered, non-overlapping manner.

A grid of stars arranged in a pattern. The stars are placed at the intersections of horizontal and vertical dotted lines. The pattern consists of several rows of stars, with some rows having more stars than others. The stars are small and white.

» Exemplo de caracteres alfabéticos com rótulos de identificação utilizados no aprendizado realizado nos testes com o padrão C (Anexo A.4 - Padrão C). Continuação ...

A decorative border consisting of a grid of small black stars arranged in a repeating pattern. The grid is composed of approximately 10 columns and 15 rows of stars, creating a scalloped or wavy effect along the outer edges.

A decorative border consisting of a grid of small stars and dots. The stars are arranged in a repeating pattern of vertical columns and horizontal rows. The dots are placed between the stars, creating a dense, dotted background.

A decorative border consisting of a grid of small stars and dots. The stars are arranged in a repeating pattern of horizontal rows, with some rows containing only stars and others containing both stars and dots. The dots are placed at regular intervals between the stars. The border is contained within a rectangular frame.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of vertical columns and horizontal rows. The dots are small circles that fill the spaces between the stars.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of horizontal rows. Each row contains a central star flanked by two pairs of smaller stars. The rows are separated by vertical lines of dots. The entire border is enclosed within a rectangular frame of dots.

A decorative border consisting of a grid of stars and dots. The stars are arranged in a repeating pattern of horizontal and vertical rows, with some stars missing to create a mesh-like effect. The border is enclosed by a thin black line.

» Exemplo de caracteres alfabéticos não identificados utilizados no reconhecimento realizado nos testes com o padrão C (Anexo A.4 - Padrão C).

.....	...*	...
.....****
.....******
.....********
.....*********
.....***********
.....**********
.....**********
.....************
.....*************
.....***************
.....***************
.....**********
.....**********
.....*********
.....***************
.....***************
.....***************
.....***************
?	?	?

.....*
.....*****
.....********
*******
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
*********
?	?	?

A.4 - Avaliação do Desempenho do Sistema N²OCR

A avaliação de desempenho do protótipo do sistema N²OCR se baseia no teste de três tipos de exemplos de bit-maps de caracteres, são eles:

- Imagens obtidas a partir da digitalização de textos impressos através de uma impressora laser e capturados através de um *handy scanner*.

- Uso de bit-maps gerados por um programa do sistema N²OCR desenvolvido para este fim. As imagens são geradas de forma a permitir testes do reconhecedor, podendo inclusive simular perturbações sobre os padrões obtidos.

- Imagens obtidas de uma base de dados de padrões de caracteres do correio inglês. Estas imagens apresentam exemplos típicos de caracteres sujeitos a muitas perturbações e de múltiplos fontes, constituindo-se um excelente teste para o sistema implementado.

Antes de apresentar os resultados das simulações é importante que sejam esclarecidos alguns detalhes a respeito dos parâmetros, procedimentos e técnicas empregadas durante o processo de avaliação do sistema N²OCR.

Observações a Respeito da Avaliação do Sistema

- A função de ativação utilizada em todos os teste na fase de aprendizado foi a "altered threshold" com limite superior igual a 1.0 e o limite inferior definido conforme descrição indicada nos dados referentes a fase de aprendizado de cada teste. Estes valores dos limiares são definidos sempre através do arquivo de configuração do simulador (vide Anexo A.5 - Listagens 1 e 3).

- A função de ativação utilizada no reconhecimento é uma função do tipo "threshold logic", onde o limiar superior é igual a 1.0 e o limiar inferior é estabelecido através do parâmetro de "certeza mínima" contido no arquivo de configuração (vide Anexo A.5 - Listagens 1 e 3).

- São empregados dois fatores de convergência (β), o fator de acerto, que é aplicado no neurônio cuja ativação deveria ocorrer e não ocorreu (neurônio responsável pelo reconhecimento), e, o fator de erro, que é aplicado no neurônio cuja ativação não deveria ocorrer, mas que se ativou (não é a resposta correta e está ativado).

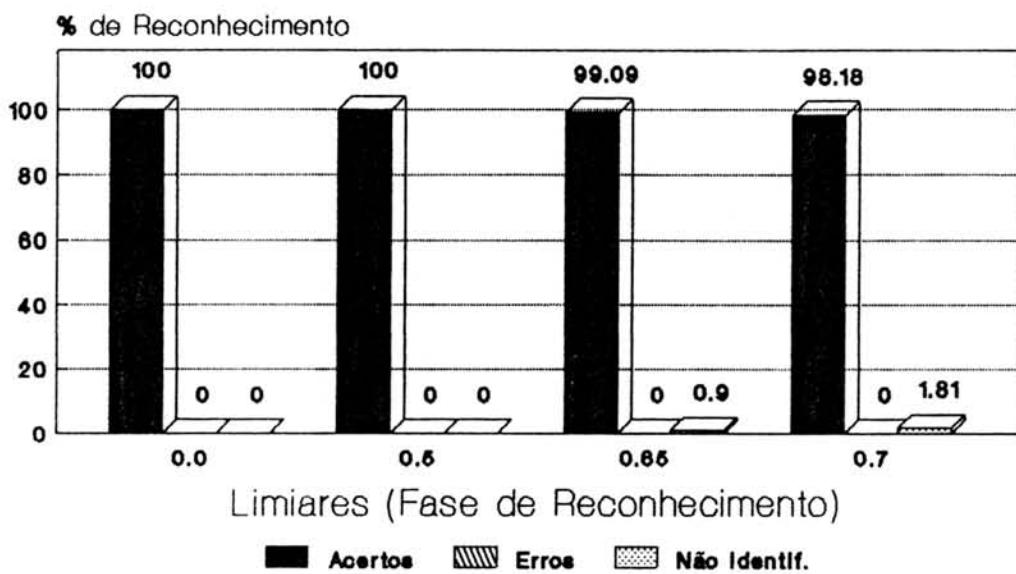
- O total de ativações da rede, obtido nas simulações, controla o número total de vezes que a rede como um todo foi ativada, independente do número de neurônios em uso naquele momento. O total de ativações individuais indica o número de vezes em que foi realizada uma ativação de algum neurônio, indicando de uma maneira mais imediata o número de vezes em que foi realizada uma soma ponderada entre as entradas e os pesos de qualquer um dos neurônios. O número de ativações individuais é aproximadamente igual ao número de ativações totais da rede multiplicado pelo número de neurônios. É certo que o número de ativações individuais é um pouco menor que o resultado deste produto, pois a rede irá se expandir pouco a pouco, aumentando o número de neurônios a serem ativados a medida que a rede cresce.

- A rede é expansível, sendo possível ocorrer em uma simulação de se ter um número superior de neurônios em relação ao número de classes diferentes a serem aprendida. Só é possível a expansão da rede quando temos mais de uma amostra de aprendizado para uma mesma classe. Um outro fator muito importante e que pode influenciar na expansão da rede é o limiar empregado no aprendizado. Quanto maior for este limiar, os padrões aprendidos terão uma "folga" menor e portanto se tivermos dois padrões com diferenças significativas a serem associados a uma mesma classe, isto irá requerer a alocação de um neurônio extra.

- A simulação considera na fase de reconhecimento o neurônio que além de se ativar, possui o valor mais alto de ativação em sua saída. Isto permite identificar apenas um padrão a ser reconhecido, evitando conflitos quando mais de um neurônio se ativa ao mesmo tempo.

1. Padrão de Teste A - Handy Scanner DFI HS-3000

SISTEMA N2OCR
Avaliação de Desempenho
Padrão A - Teste 1



Aprendizado → Nros. 0 a 9 (10 Classes)

- 10 Padrões, 1 Padrão/Classe
- Padrões com 16 x 18 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 1.0
- Fatores de convergência :
- β acerto \rightarrow 0.05 / β erro \rightarrow 0.0
- Neurônios utilizados : 10
- Total de ativações : 285 (rede) - 1740 (individuais)

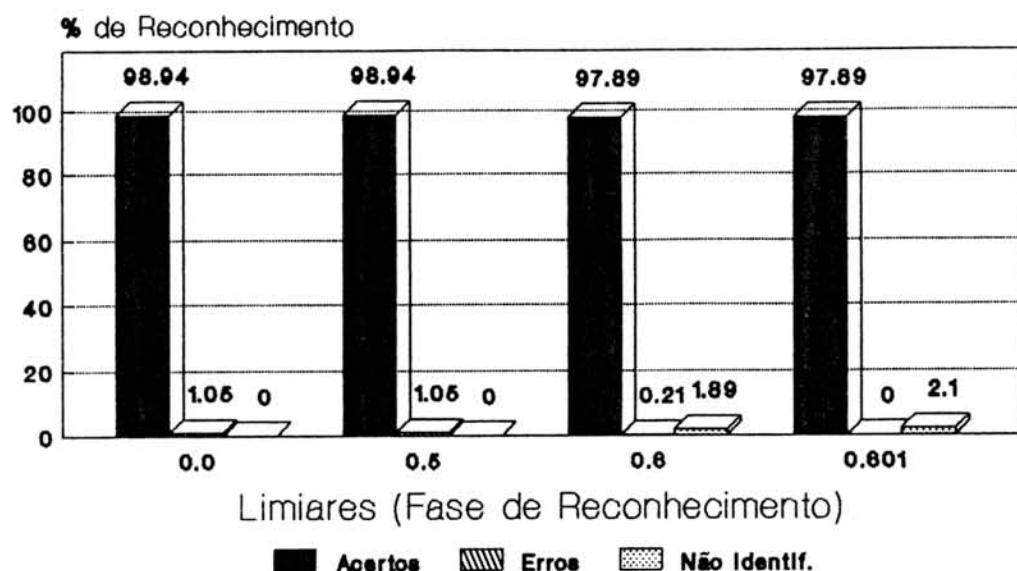
Fase de Reconhecimento:

- Número de padrões de teste : 220
- Classes → Numerais de 0 a 9
- Número de classes : 10 - 22 padrões por classe

SISTEMA N2OCR

Avaliação de Desempenho

Padrão A - Teste 2



Aprendizado → Nros. 0 a 9 (10 Classes)
 • 10 Padrões, 1 Padrão/Classe
 • Padrões com 16 x 16 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.9
- Fatores de convergência :
 β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.01$
- Neurônios utilizados : 10

Fase de Reconhecimento:

- Número de padrões de teste : 476
- Classes \rightarrow Numerais de 0 a 9
- Número de classes : 10
 (Distribuídos de forma irregular entre as classes)

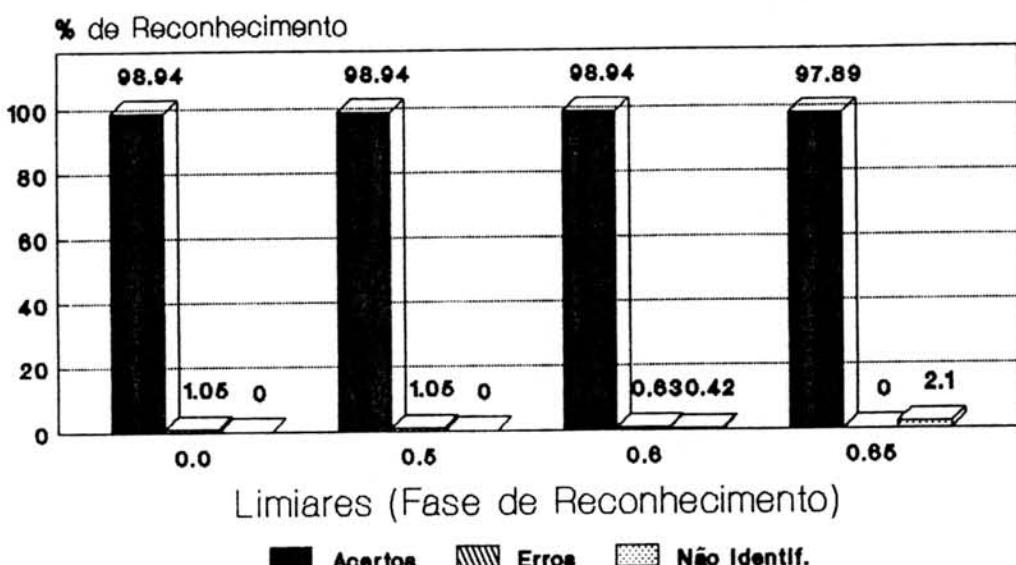
Observações:

- Todos os erros cometidos foram na identificação de '0's em lugar do padrão correto que era o '9'. Isto indica que havia uma "deficiência" no aprendizado do 0 e do 9 dificultando sua diferenciação, ou, que nos padrões de teste haviam padrões de '0' e '9' muito similares.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão A - Teste 3



Aprendizado → Nros. 0 a 9 (10 Classes)
 • 10 Padrões, 1 Padrão/Classe
 • Padrões com 16 x 16 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 1.0
- Fatores de convergência :
 - β acerto → 0.05 / β erro → 0.0
- Neurônios utilizados : 10

Fase de Reconhecimento:

- Número de padrões de teste : 476
- Classes → Numerais de 0 a 9
- Número de classes : 10
 (Distribuídos de forma irregular entre as classes)

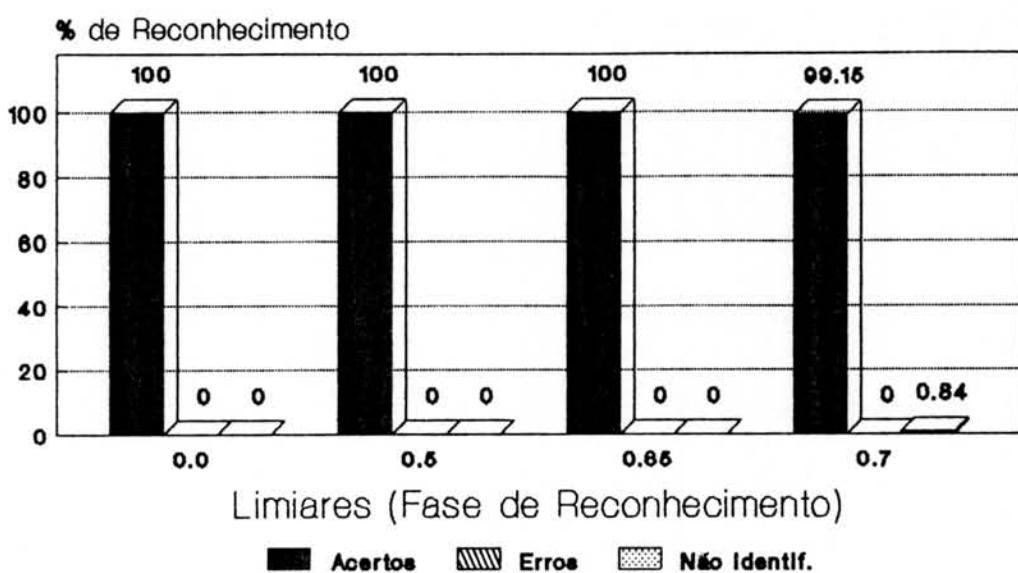
Observações:

- Os padrões de teste foram os mesmos utilizados no teste anterior (Teste 2). Os erros ocorridos também foram do mesmo tipo dos ocorridos no Teste 2.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão A - Teste 4



Aprendizado → Nros. 0 a 9 (10 Classes)
 • 220 Padrões, 22 Padrões/Classe
 • Padrões com 16 x 16 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.8
- Fatores de convergência :
 - β acerto → 0.05 / β erro → 0.0
- Neurônios utilizados : 19
- Total de ativações : 55275 (rede) - 880652 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 476
- Classes → Numerais de 0 a 9
- Número de classes : 10
 (Distribuídos de forma irregular entre as classes)

Observações:

- Os padrões de teste são os mesmos do Teste 2 e 3.

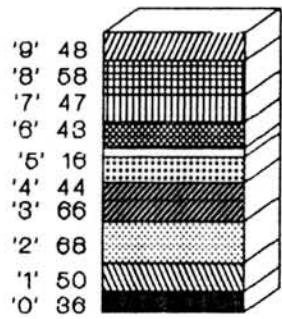
• Durante a fase de reconhecimento, apesar de terem sido treinados 19 neurônios, apenas 13 deles foram utilizados. Isto indica que a rede após se expandir poderia voltar a descartar alguns neurônios (contrair-se), pois aparentemente não são mais necessários. É bastante provável que durante a expansão alguns novos neurônios sejam capazes de "substituir" alguns dos outros neurônios criados.

Tabela de utilização dos neurônios da rede

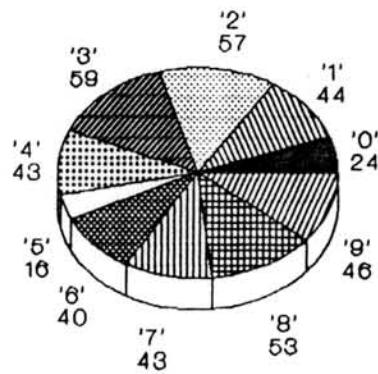
Número do Neurônio	0 a 9	12	13	18
Padrão Aprendido	0 a 9	0	8	9
Número de Usos	X	1	1	1

SISTEMA N2OCR

Avaliação de Desempenho PADRÃO A



TESTE 2



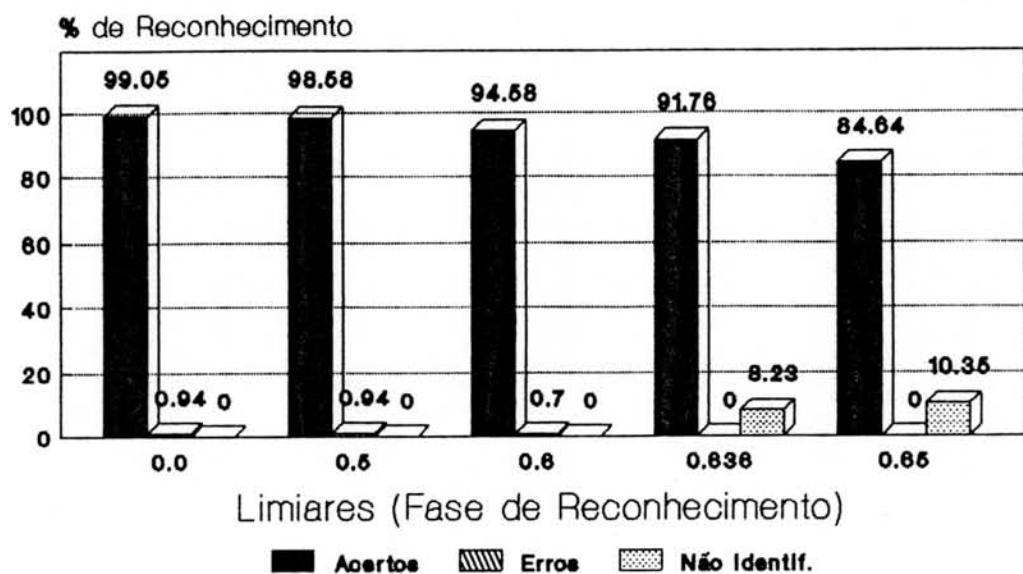
TESTE 5

Quantidade de Padrões por Classe

SISTEMA N2OCR

Avaliação de Desempenho

Padrão A - Teste 5



Aprendizado → Nros. 0 a 9 (10 Classes)
 * 220 Padrões, 22 Padrões/Classe
 * Padrões com 16 x 16 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.8
- Fatores de convergência :
 β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 19
- Total de ativações : 55275 (rede) - 880652 (indiv.)

Fase de Reconhecimento:

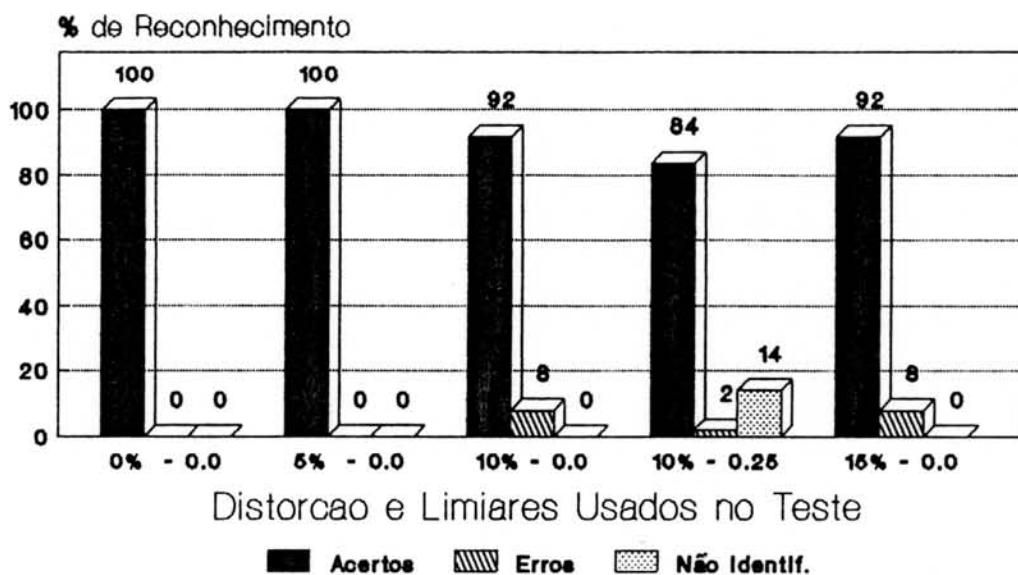
- Número de padrões de teste : 425
- Classes → Numerais de 0 a 9
- Número de classes : 10 (distribuição irregular)

Observações:

- Os padrões de teste foram digitalizados com uma rotação de 90 graus em relação a sua posição normal e posteriormente sua posição foi corrigida por software (introduzindo obviamente algumas distorções).
- Todos os erros ocorridos foram trocas, onde foi reconhecido o número '3' no lugar do '8'.
- Somente 12 neurônios foram realmente utilizados na fase de reconhecimento.

2. Padrão de Teste B - Gerador de Imagens

SISTEMA N2OCR
Avaliação de Desempenho
Padrão B - Teste 1



Aprendizado → Nros. 0 a 9 (10 Classes)
 * 10 Padrões, 1 Padrão/Classe
 * Padrões com 16 x 16 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.3
- Fatores de convergência :
 - β acerto → 0.05 / β erro → 0.0
- Neurônios utilizados : 10
- Total de ativações : 107 (rede) - 698 (indiv.)

Fase de Reconhecimento:

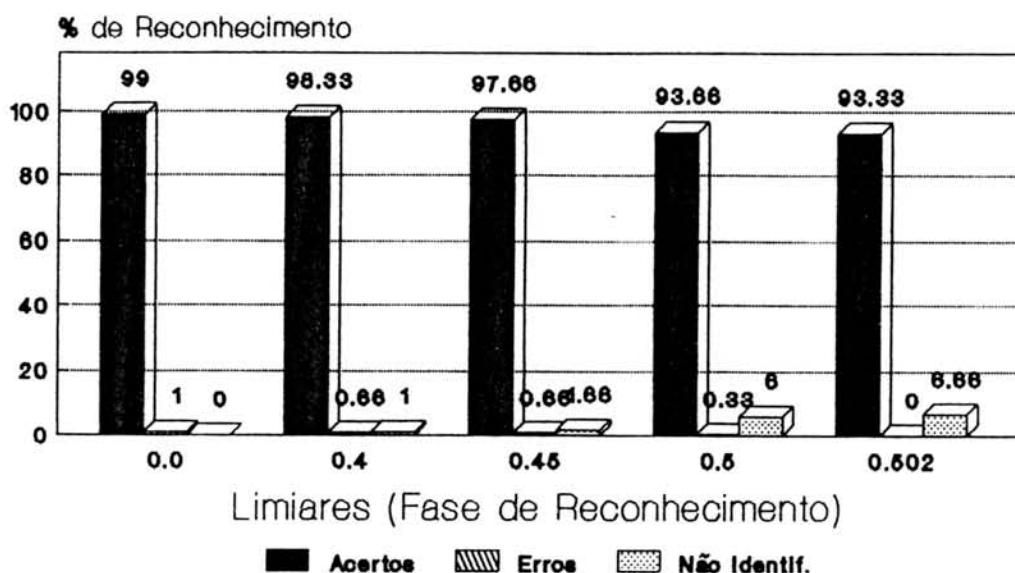
- Número de padrões de teste : 50
- Classes → Numerais de 0 a 9 (distorcidos)
- Número de classes : 10 - 5 padrões por classe

Observações:

- Na fase de reconhecimento foram testados padrões gerados pelo programa gerador de imagens do sistema N²OCR, onde foi inserida uma distorção com a inversão de 0, 5, 10 ou 15% do total de elementos que compõem o padrão.
- Na fase de aprendizado, foram testados os limiares de 0.95, 0.9, 0.6 e 0.5 obtendo-se para todos estes 100% de reconhecimento correto, para qualquer um dos padrões com até 15% de distorção.

3. Padrão de Teste C - Correio Inglês

SISTEMA N2OCR
Avaliação de Desempenho
Padrão C - Teste 1



Aprendizado → Nros. 0 a 9 (10 Classes)

- 50 Padrões, 5 Padrão/Classe
- Padrões com 16×24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.6
- Fatores de convergência :

 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$

- Neurônios utilizados : 10
- Total de ativações : 1555 (rede) - 14999 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 300
- Classes → Apenas o dígito '0' (1 classe apenas)
- Número de classes : 1 - 300 padrões por classe

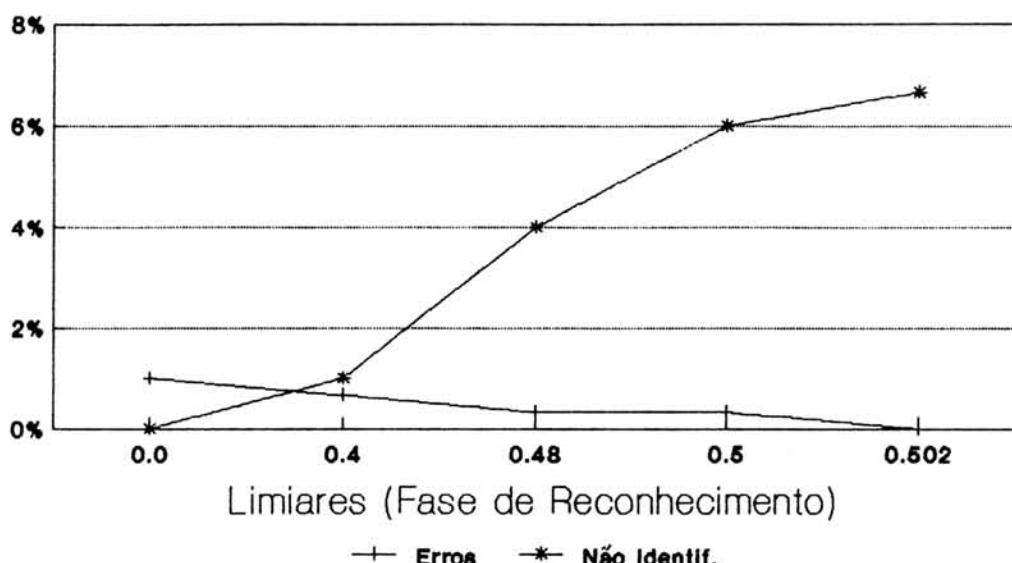
Observações:

- Neste teste foi realizada a geração de candidatos alternativos para um pós-processador. É interessante salientar que apenas três padrões de teste foram identificados incorretamente e que somente para estes três foi indicada a existência de uma segunda alternativa, que era a resposta correta. Para todas as identificações corretas não havia indicação de resultados alternativos.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 1A



Aprendizado → Nros. 0 a 9 (10 Classes)
 • 50 Padrões, 5 Padrão/Classe
 • Padrões com 16×24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.6
- Fatores de convergência :

 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$

- Neurônios utilizados : 10
- Total de ativações : 1555 (rede) - 14999 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 300
- Classes \rightarrow Apenas o dígito '0' (1 classe apenas)
- Número de classes : 1 - 300 padrões por classe

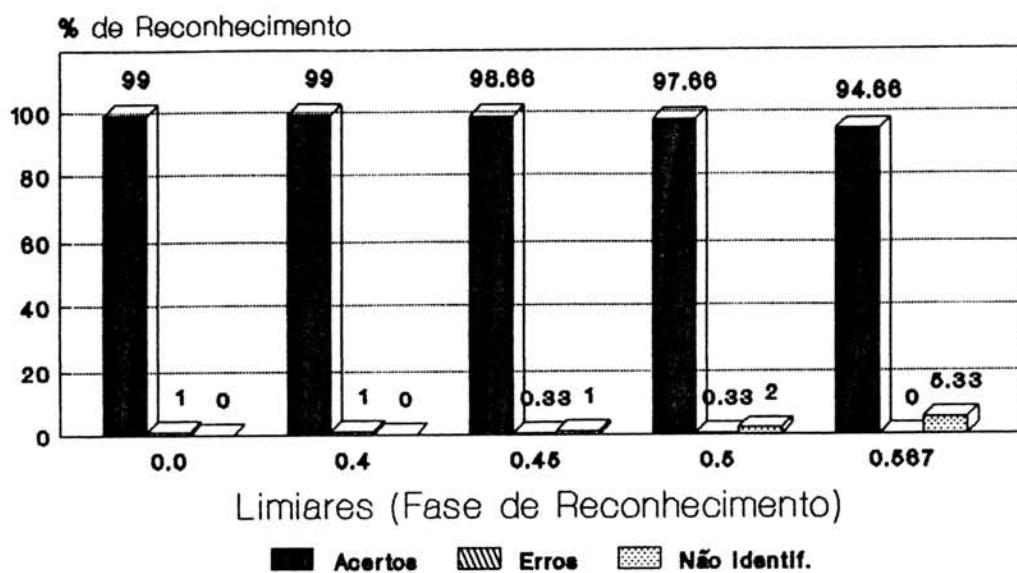
Observação:

- Esta é uma análise complementar do mesmo experimento realizado no Teste 1 (Padrão C).
- Este gráfico permite ver em mais detalhes a influência do limiar de "certeza mínima" empregado na fase de reconhecimento, em relação aos índices de erros e não identificação dos padrões.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 2



Aprendizado → Nros. 0 a 9 (10 Classes)

- 50 Padrões, 5 Padrão/Classe
- Padrões com 16 x 24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.8
- Fatores de convergência :
 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 27
- Total de ativações : 13666 (rede) - 272339 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 300
- Classes → Apenas o dígito '0' (1 classe apenas)
- Número de classes : 1 - 300 padrões por classe

Observações:

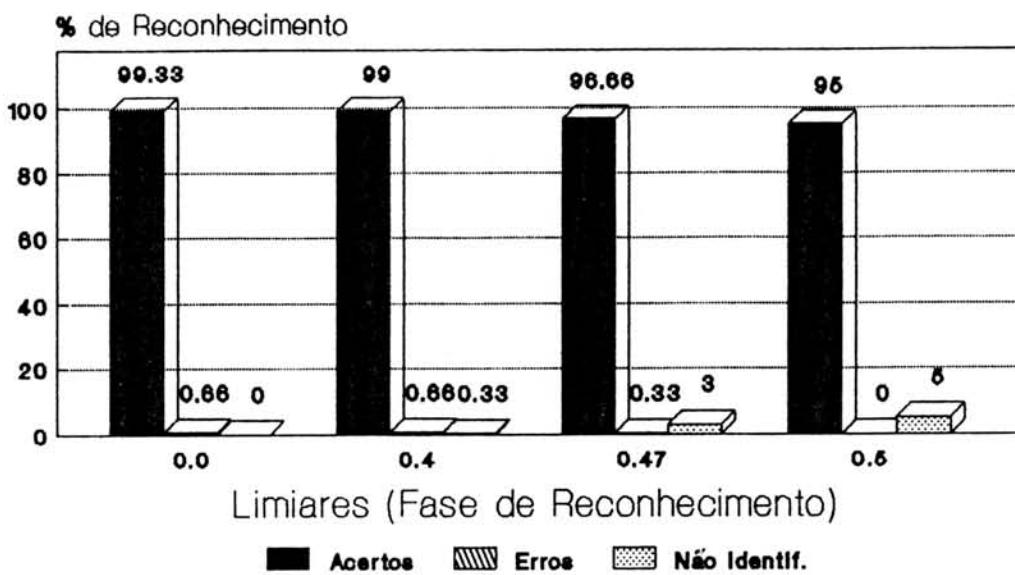
• Este teste é uma repetição do Teste 1 (Padrão C), onde foi alterado apenas o limiar empregado na fase de aprendizado. Como resultado da alteração deste parâmetro, nota-se que:

1. Aumentou do número de neurônios da rede (expansão)
2. Aumentou a eficácia do reconhecimento (a rede está mais "afinada" para o reconhecimento)

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 3



Aprendizado → Letras A a Z (24 Classes)
 • 120 Padrões, 5 Padrão/Classe
 • Padrões com 16 x 24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.7
- Fatores de convergência :
 β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 38
- Total de ativações : 38518 (rede) - 1228767 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 300
- Classes → Apenas a letra 'a' (1 classe apenas)
- Número de classes : 1 - 300 padrões por classe

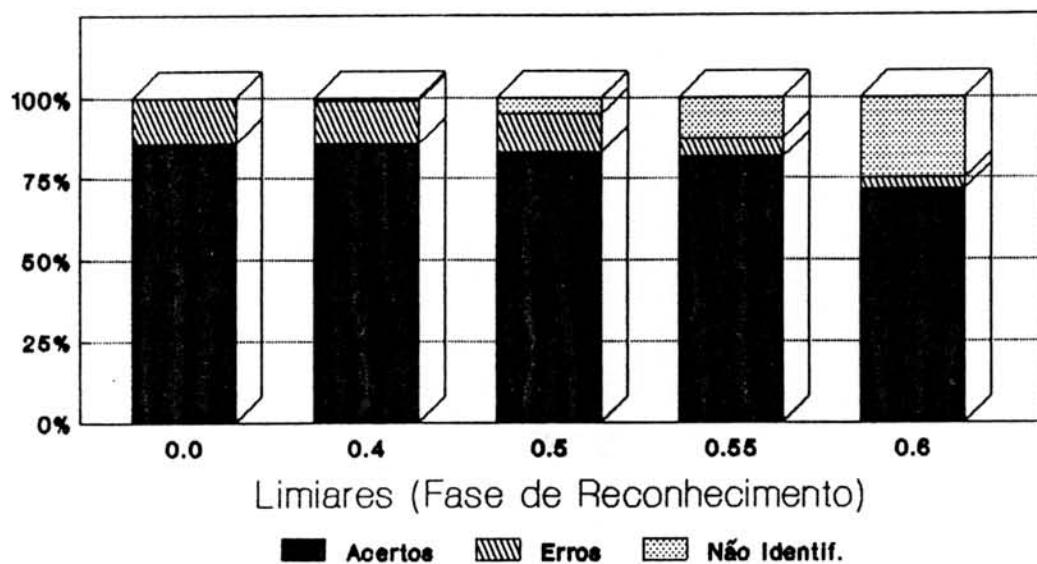
Observações:

- Este experimento obteve ótimos resultados, demonstrando que para este tipo de caso a rede teve um aprendizado muito bom.
- O teste seguinte já possui uma certa queda na eficácia do sistema. Este teste e o seguinte servem para demonstrar como os resultados de uma avaliação podem ser diferentes de acordo com o conjunto de testes utilizados.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 4



Aprendizado → Letras A a Z (24 Classes)

- 120 Padrões, 5 Padrão/Classe
- Padrões com 16×24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.7
- Fatores de convergência :
 - β acerto → 0.05 / β erro → 0.0
- Neurônios utilizados : 38
- Total de ativações : 38518 (rede) - 1228767 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 120
- Classes → Caracteres de 'a' a 'z'
- Número de classes : 24 - 5 padrões por classe

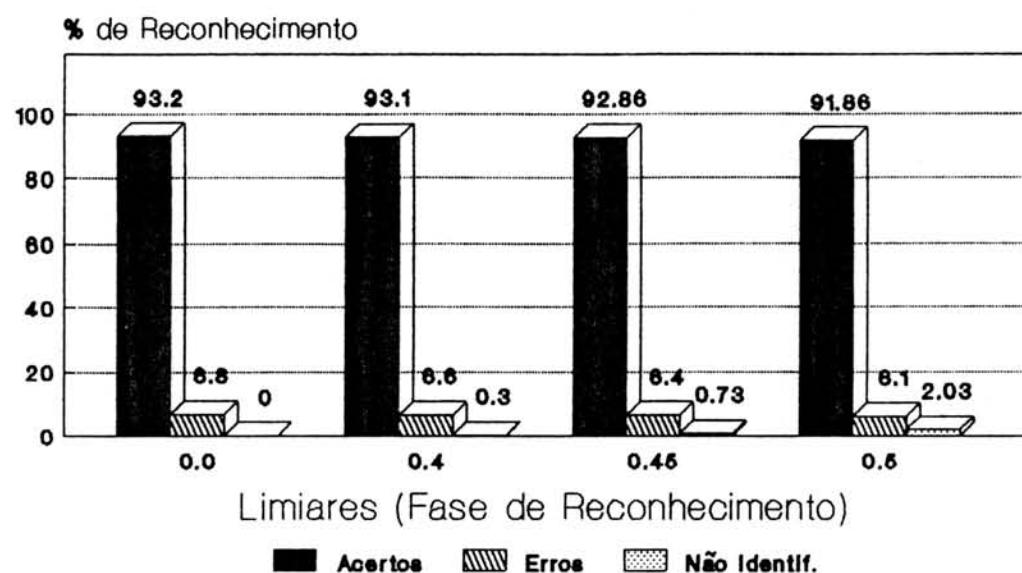
Observações:

- Este teste repete o experimento realizado no Teste 3 (Padrão C), em relação ao aprendizado, mas com outro conjunto de padrões de teste (diferentes dos ensinados a rede).
 - Nota-se uma queda no reconhecimento (com limiar 0.0 tem-se 85.83% acertos), que resulta do fato de termos alguns padrões com o aprendizado muito fraco (reconhecimento ruim) e que prejudicam a média geral.
 - Este teste foi realizado novamente com outro conjunto de padrões e cujos resultados são demonstrados nos gráficos dos Testes 6 e 6A (Padrão C).

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 5



Aprendizado → Nros. 0 a 9 (10 Classes)

- 50 Padrões, 5 Padrão/Classe
- Padrões com 16 x 24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.8
- Fatores de convergência :
 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 27
- Total de ativações : 13666 (rede) - 272339 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 3000
- Classes → Numerais de 0 a 9
- Número de classes : 10 - 300 padrões por classe

Observações:

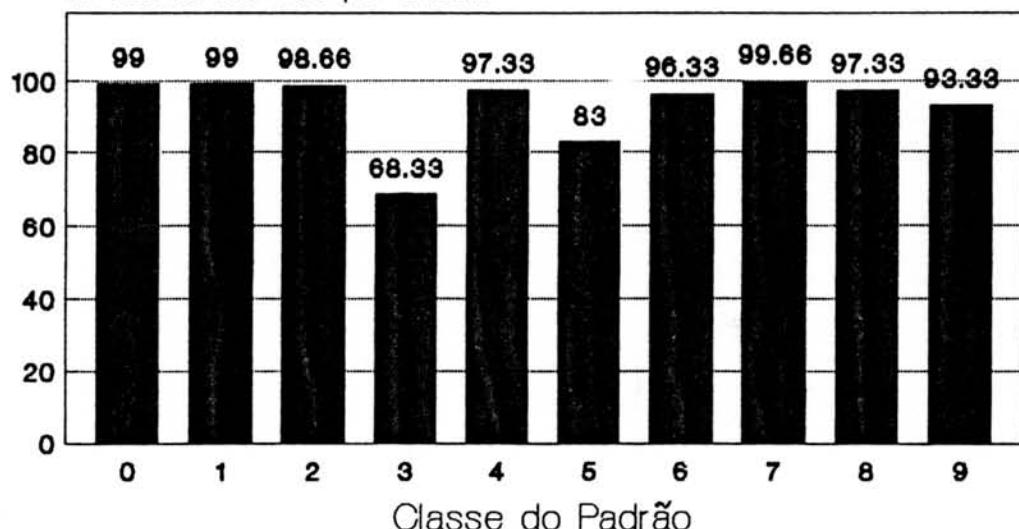
- Este teste repete o experimento realizado no Teste 2 (Padrão C), em relação ao aprendizado, mas com outro conjunto de padrões de teste.
 - O teste seguinte analisa em mais detalhes os resultados obtidos neste experimento.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 5A

% Padrões Corretos por Classe



Aprendizado \Rightarrow Nros. 0 a 9 (10 Classes)

- 50 Padrões, 5 Padrão/Classe
- Padrões com 16×24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.8
- Fatores de convergência :
 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 27
- Total de ativações : 13666 (rede) - 272339 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 3000
- Classes \rightarrow Numerais de 0 a 9
- Número de classes : 10 - 300 padrões por classe

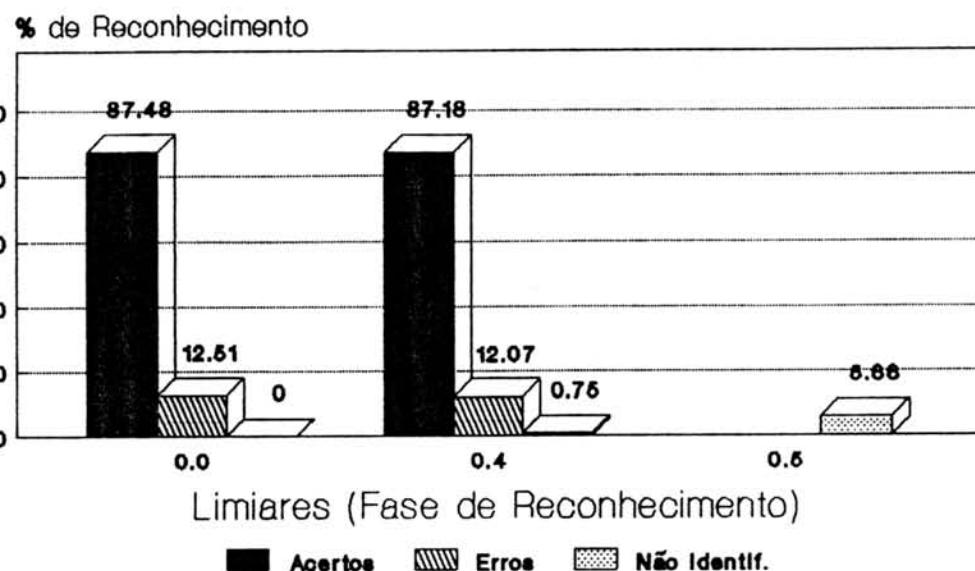
Observações:

- Este teste repete o experimento realizado no Teste 2 (Padrão C), em relação ao aprendizado, mas com outro conjunto de padrões de teste. Este teste também apresenta uma segunda análise para o mesmo experimento realizado no Teste 5 (Padrão C).
- Neste teste é feita uma demonstração dos graus de acerto para cada uma das classes isoladamente (com limiar 0.0). Nota-se algumas destas classes, como por exemplo, o dígito '3' e o '5', onde há uma necessidade de se melhorar o aprendizado.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 6



Aprendizado → Letras A a Z (24 Classes)

- 120 Padrões, 5 Padrão/Classe
- Padrões com 16×24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.7
- Fatores de convergência :
 - β acerto $\rightarrow 0.05$ / β erro $\rightarrow 0.0$
- Neurônios utilizados : 38
- Total de ativações : 38518 (rede) - 1228767 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 7200
- Classes → Caracteres de 'a' a 'z'
- Número de classes : 24 - 300 padrões por classe

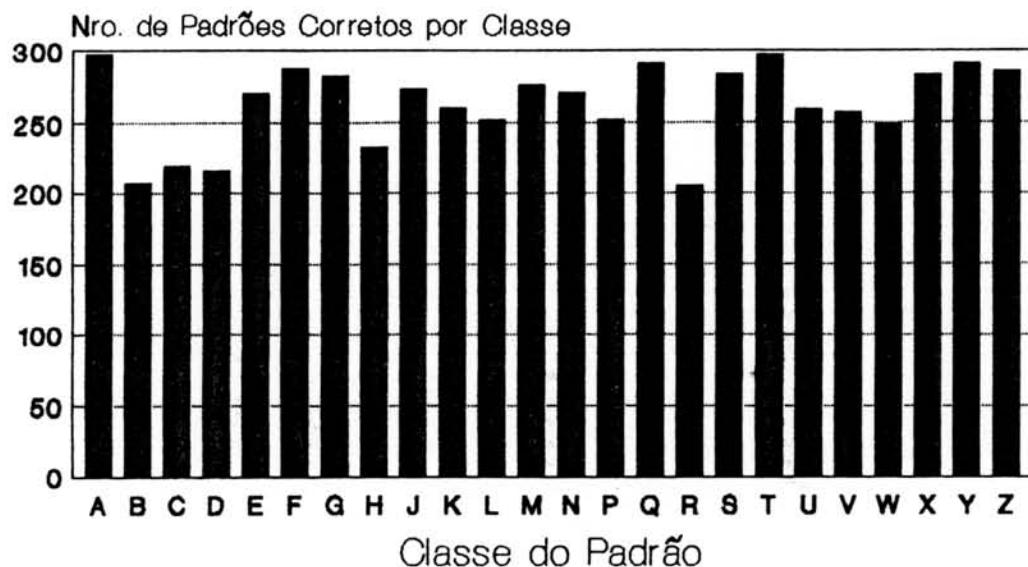
Observações:

- Este teste repete o experimento realizado no Teste 4 (Padrão C), em relação ao aprendizado, mas com outro conjunto de padrões de teste.
 - O teste seguinte analisa em mais detalhes os resultados obtidos neste experimento.

SISTEMA N2OCR

Avaliação de Desempenho

Padrão C - Teste 6A



Aprendizado → Letras A a Z (24 Classes)

- 120 Padrões, 24 Padrão/Classe
- Padrões com 16 x 24 elementos

Fase de Aprendizado:

- Limiar da função de ativação : 0.7
- Fatores de convergência :

 - β acerto → 0.05 / β erro → 0.0

- Neurônios utilizados : 38
- Total de ativações : 38518 (rede) - 1228767 (indiv.)

Fase de Reconhecimento:

- Número de padrões de teste : 7200
- Classes → Caracteres de 'a' a 'z'
- Número de classes : 24 - 300 padrões por classe

Observações:

- Este teste repete o experimento realizado no Teste 4 (Padrão C), em relação ao aprendizado, mas com outro conjunto de padrões de teste. Este teste também apresenta uma segunda análise para o mesmo experimento realizado no Teste 6 (Padrão C).
- Neste teste é feita uma demonstração dos graus de acerto para cada uma das classes isoladamente (com limiar 0.0). Nota-se algumas destas classes, como por exemplo, as letras 'b', 'c', 'd', 'h' e o 'r', onde há uma necessidade de se melhorar o aprendizado.

A.5 Listagens de Arquivos do Sistema N²OCR

1. Listagem de um Arquivo de Configuração Exemplo

```
/***** CONFIGURAÇÃO *****/
```

```
$C RGSAEF
$R 241
$L 0.8 1.0 0.0 1
$F 0.05 0.0 0.0
$N 4
$A 50
```

```
/***** COMENTARIOS *****/
```

=> Comandos:

```
$C = Configuração Geral
$R = Configuração dos Níveis de Relatório
$L = Configuração dos Valores de Limiares
$F = Configuração dos Fatores da Regra Delta
$D = Configuração do Código do Caracter Desconhecido
$N = Configuração do Máximo/Total de Neurônios da Rede
$A = Configuração do Máximo de Retentativas de Aprendizado
      (Somente para o modelo ADAnLIene)
```

=> Configuração Geral :

```
R - Gera arquivo de saída com o texto reconhecido
    (TEXTO.RECD)
L - Lê arquivo com pesos de uma rede neural, sem
    aprendizado (NEURONS.BIN)
S - Grava arquivo com pesos da rede neural após aprendizado
    (NEURONS.BIN)
E - Exibe no vídeo a saída do texto reconhecido
A - Altera a semente de geração de números randômicos
P - Lê todos os padrões originais de aprendizado para a
    memória
N - Não realiza o acompanhamento do aprendizado
G - Gera arquivo com relatório geral da simulação
    (NEURONS.REL)
C - Gera lista de candidatos para o pós-processador
F - Mantém fatores de aprendizado, mesmo ao realizar
    expansão
```

=> Configuração dos Limiares :

Limite_Inf.	Limite_Sup.	Limite_Rígido	Tipo_Fç_Decisão
:	:	:	:
+--> Threshold <--+	+--> Hard Limiter	+--> (0 a 3)	

0 = Threshold Logic	\	
1 = Altered Threshold	\	"Tipo_Função_Decisão"
2 = Hard Limiter	>	Usadas no aprendizado
3 = Sigmoidal Function	/	

=> Configuração dos Fatores :

Fator_Acerto	Fator_Erro	Certeza_Mínima
		+--> Limiar para reconhecimento
		+--> Ajuste neurônio ativado incorretamente
+----->	Ajuste neurônio que não se ativou quando deveria	

=> Flags para Relatório :

REL_APREND	0x01 (Indica padrão em aprendizado) ¹
REL_ACOMP	0x02 (Indica acompanhamento do aprendizado) ²
REL_ERRO	0x04 (Indica erro calculado pela Regra Delta) ³
REL_ESTOURO	0x08 (Indica tentativa de incremento ou decremento dos pesos além da faixa permitida -1 a +1) ⁴
REL_REWIND	0x10 (Indica código escolhido e código correto) ⁵
REL_RESULT	0x20 (Indica o resultado do reconhecimento) ⁶
REL_RECCOMPL	0x40 (Indica todos os graus de reconhecimento) ⁷
REL_EXPAND	0x80 (Indica ocorrência de expansão da rede) ⁸

¹ Aprendizado :

Aprendendo: <nro. ordem padrão> - <código ASCII>

² Acompanhamento Detalhado :

N: <nro. neurônio> # P: <código ASCII> #
O: <atual> - <último> # C: <certeza>

³ Erro Calculado :

#E: <erro>#

⁴ Incremento e Decremento :

Um sinal de '+' ou '-' a cada vez que estourar a faixa

⁵ Escolhido e Correto :

#CE: <escolhido ASCII> <-> CC: <correto ASCII> #

⁶ Resultado do Reconhecimento :

Nr: <nro. neurônio> - S: <caracter ASCII> ((valor ASCII))
C: <certeza>

⁷ Grau de Reconhecimento :

GR: <XX.XX> <XX.XX> ... <XX.XX>

⁸ Expansão da Rede :

<EXP:<novo>> ou <EXP:<erro>-<antigo>>

2. Listagem de um Arquivo de Relatório do Aprendizado

```
>>> RELATÓRIO : Rede Neural - ADAnLIENE  
Padrões : xor.let # Teste : xor.tst  
Neurônios : 4 # Padrões : 4 # Entradas : 2  
Data - 03/10/1991 # Hora Inicial - 10:06
```

```
Congifuração geral : RGSAEF  
Configuração relat.: 241  
Valores dos limites: 0.8 1.0 0.0 1  
Valores das faixas : 0.05 0.0 0.0
```

=> APRENDIZADO

```
Aprendendo: 0 - 0  
Aprendendo: 1 - 1  
Aprendendo: 2 - 1  
#CE: ??? <-> CC: 1 #  
...  
#CE: ??? <-> CC: 1 #  
#<EXP: -1 1>  
Aprendendo: 3 - 0  
#CE: ??? <-> CC: 0 #  
...  
#CE: ??? <-> CC: 0 #  
#<EXP: -1 0>
```

```
Neurônios Utilizados : 4  
Total de Ativações da Rede : 380  
Total de Ativações Individuais : 946
```

Testando ...

```
GR: 85.00 -15.00 4.25 -83.32  
Nr: 0 - S: < > (000) - C: 85.00
```

```
GR: -15.00 85.00 -80.71 16.68  
Nr: 1 - S: < > (001) - C: 85.00
```

```
GR: 15.00 -85.00 80.71 -16.68  
Nr: 2 - S: < > (001) - C: 80.71
```

```
GR: -85.00 15.00 -4.25 83.32  
Nr: 3 - S: < > (000) - C: 83.32
```

Hora Final - 10:06

>>>> FIM DA SIMULAÇÃO <<<<<

3. Listagem de um Arquivo para Aprendizado

Arquivo XOR.LET:

```
$2 1 4 0
00
0
01
1
10
1
11
0
!
```

4. Listagem de um Arquivo para Reconhecimento

Arquivo XOR.TST:

```
$2 1 4 0
00
?
01
?
10
?
11
?
!
```

5. Listagem de Exemplo da Utilização do Sistema N²OCR

ADAnLIENE OCR SYSTEM	Janeiro/91
Por : Fernando Osorio ** Versao 1.0	

Arquivo de padroes : let\xor
 Arquivo de teste : tst\xor
 Arquivo simulacao : result\xornovo
 Nro. total de neuronios : 4
 Nro. total de entradas : 2
 Nro. total de padroes : 4

Aprendendo: 0 - 0
 Aprendendo: 1 - 1
 Aprendendo: 2 - 1
 Aprendendo: 3 - 0

Testando ...
 O caracter e' < > (0)
 Testando ...
 O caracter e' < > (1)
 Testando ...
 O caracter e' < > (1)
 Testando ...
 O caracter e' < > (0)

6. Listagem da Tela de Ajuda do Sistema N²OCR

NEURAL2 - OCR TP-001 => Sistema N2OCR - Versao 1.0
Autor : Fernando Osorio #### UFRGS / CPGCC

USO : NEURAL2 [PADARQ. LET] [IDARQ. TST] [CONFIG. CFG] [SIMUL] [PESOSRN]
 NEURAL2 [/H : -H : ?]

Onde :

- PadArq - Nome do arquivo de padroes (*.LET) - Sem extensao
- IdArq - Nome do arquivo com padrao a identificar (.TST)
- Config - Nome do arquivo de configuracao (*.CFG)
- Simul - Nome do arquivo de simulacao (sem extensao)
- PesosRN - Nome do arquivo de pesos da rede (so' p/reconhec.)
- /H,-H,? - Mostra tela de ajuda com as intrucoes do programa

Saida: Gera um arquivo com o texto reconhecido - SIMUL.REC

7. Listagem das Funções de Transferência Implementadas

/**** FUNCOES DE TRANSFERENCIA ***/

/*-----

Nome : PASSA

Funcao : Retorna o valor de entrada. Ativacao de uma funcao "dummy". Realiza apenas a normalizacao do valor de saida.

*/

```
DOUBLE passa (valor)
DOUBLE valor;
{
    valor /= Rede.nro_entr;      /* Normaliza o valor ! */
    return(valor);
}
```

/*-----

Nome : THRESHOLD

Funcao : Funcao de ativacao do tipo THRESHOLD, so' passa o valor caso este esteja acima de uma faixa limite.

Grafico -->	
-------------	--

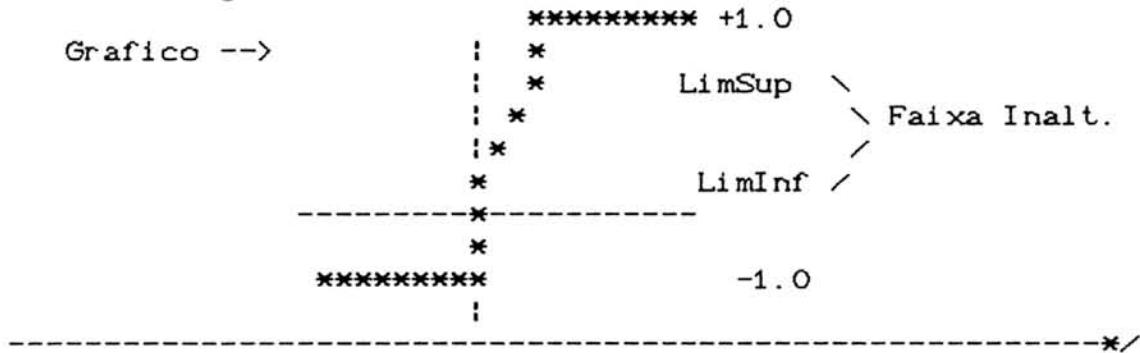
*/

```
DOUBLE threshold (valor)
DOUBLE valor;
{
    valor /= Rede.nro_entr;
    if (valor < LimInf)
        return(LimInf);
    if (valor > LimSup)
        return(LimSup);
    return(valor);
}
```

/*-----

Nome : ALT_THRESHOLD

Funcao : Funcao de ativacao do tipo THRESHOLD, so' passa o valor caso este esteja dentro de uma faixa limite.
 Funcao alternativa: tem um limiar rígido inferior rígido inferior

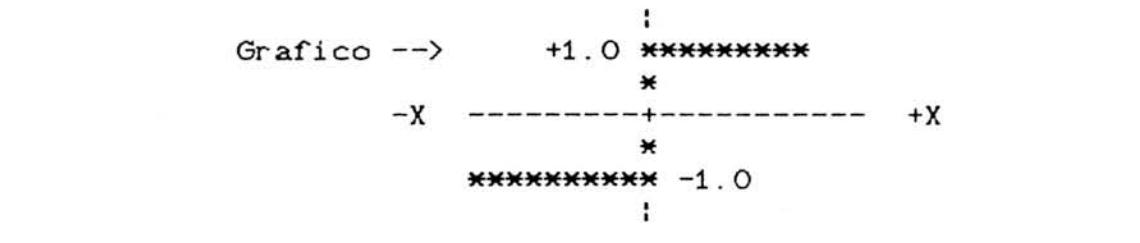


```

DOUBLE alt_threshold (valor)
DOUBLE valor;
{
    valor /= Rede.nro_entr;
    if (valor < LimInf)
        return(-1.0);
    if (valor > LimSup)
        return(1.0);
    return(valor);
}
  
```

Nome : HARD_LIMITER

Funcao : Funcao de ativacao do tipo HARD LIMITER, so' passa o valor caso este esteja acima de uma faixa limite



```

DOUBLE hard_limiter (valor)
DOUBLE valor;
{
    valor /= Rede.nro_entr;
    valor -= LIMIAR;
    if (valor > 0)
        return(1.0);
    else
        return(-1.0);
}
  
```

/*-----

Nome : SIGMOIDAL

Funcao : Funcao de ativacao do tipo SIGMOIDE.
(Tem a aparencia de um "S")

/-----/

```
DOUBLE sigmoidal (valor)
DOUBLE valor;
{
    valor = 1.0 / (1.0 + exp(-1.0 * valor));
    /* Normaliza valor : faixa de -1.0 a +1.0 */
    valor = valor * 2 - 1.0;
    return(valor);
}
```

/**** FIM DAS FUNCOES DE TRANSFERENCIA ***/

BIBLIOGRAFIA CITADA

Capítulo 1

- [DIE 91] DIEHL, Stanford & EGLOWSTEIN, Howard. Tame the Paper Tiger. *Byte*, Hightstown, v.16, n.4, p.220-238, April 1991.
- [FAL 88] FALKNER, Mike. Hand Scanners Inch Forward. *PC Magazine*, New York, v.7, n.20, p.277-308, 29-Nov. 1988.
- [JOC 91] JOCH, Alan & GREHAN, Rick. Voices of Experience. *Byte*, Hightstown, v.16, n.4, p.239-241, April 1991.
- [KAN 89] KANE, R. & PETERSON, T. Scanners Build a Better Image From FAX to Finish. *PC Magazine*, New York, v.8, n.6, p.187-236, 28-March 1989.

Capítulo 2

- [BAY 73] BAYER, B. E. An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, New York, Proceedings (Conference Record)... New York: IEEE, 1973, p.(26-11)-(26-15).
- [FER 86] FERREIRA, Aurélio B. H. *Novo Dicionário Aurélio da Língua Portuguesa*. 2a. Ed. Rio de Janeiro: Editora Nova Fronteira, 1986. 1838p.
- [FOL 82] FOLEY, J. D. & VAN DAM, A. *Fundamentals of Interactive Computer Graphics*. Reading: Addison-Wesley, 1982. 664p.
- [GOM 90] GOMES, J. M. & VELHO, L. *Conceitos Básicos de Computação Gráfica*. São Paulo: VII Escola de Computação - IME/USP, 1990. 311p.
- [GRA 89] GRAEF, Gerald L. Graphics Formats. *Byte*, Hightstown, v.14, n.9, p.305-310, Sept. 1989.
- [HAL 79] HALL, Ernest L. *Computer Image Processing and Recognition*. New York: Academic Press, 1979. 584p.

- [HAR 72] HARMON, Leon D. Automatic Recognition of Print and Script. *Proceedings of the IEEE*, New York, v.60, n.10, p.1165-1176, Oct. 1972.
- [HUF 72] HUFFMAN, D. A. A Method for the Construction of Minimum Redundancy Codes. *Proceedings of IRE*, v. 40, p.1098-1101, Sept. 1972.
- [JIM 82] JIMENEZ, J. & NAVALON, J. Some Experiments in Image Vectorization. *IBM Journal Res. Develop.*, Armonk, v.26, n.6, p.724-734, Nov. 1982.
- [KIS 90] KISHORE, M. S. Image Segmentation. *Elektor Electronics*, Brentford, v.16, n.179, p.46-49, May 1990.
- [KOV 80] KOVALEWSKY, V. A. *Image Pattern Recognition*. New York: Springer-Verlag, 1980. 241p.
- [MAS 89] MASCARENHAS, Nelson A. & VELASCO, Flávio. R. D. *Processamento Digital de Imagens*. 2.ed. Buenos Aires: Kapelusz, 1989. 270p. (Curso ministrado durante a IV Escola Bras.-Argentina de Informática em Termas del Rio Hondo - Argentina - Jan. 1989)
- [NAG 83] NAGY, George. Candide's Practical Principles of Experimental Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, New York, v.PAM-5, n.2, p.199-200, March 1983.
- [NAK 87] NAKANO, T.; OZEKI, O. & YAMAMOTO, S. A New Video-Thresholding Method for Recognition of Low-Quality Stamped Alphanumerals. *Systems and Computers in Japan*, Silver Spring, v.18, n.6, p.60-70, 1987.
- [NET 89] NETTO, João. C. CV2D - Operador de Convolução Para Processamento de Imagens em Tempo Real. Porto Alegre: CPGCC da UFRGS, 1989. 152p. (Dissertação de Mestrado)
- [PAR 88] PARKER, J. R. Extracting Vectors from Raster Images. *Computer & Graphics*, Elmsford, v.12, n.1, p.75-79, 1988.
- [PAV 82] PAVLIDIS, Theo. *Algorithms for Graphics and Image Processing*. Berlin: Springer-Verlag, 1982. 416p.

- [PAZ 88] PAZ, E. P. & CUNHA, T. N. Iniciação ao Processamento Digital de Imagens. Rio de Janeiro: UFRJ/NCE, 1988. 55p. (Curso ministrado no VIII Congresso da SBC - VII JAI - Versão preliminar)
- [PRA 80] PRATT, W. K. et al. Combined Symbol Matching Facsimile Data Compression System. Proceedings of the IEEE, New York, v.68, n.7, p.786-796, July 1980.
- [REG 90] REGAN, Shawn M. LZW Revisited. Dr Dobb's Journal, Redwood City, n.165, p.126-127, June 1990.
- [SCH 87] SCHARCANSKI, Jacob. Um Estudo Sobre os Métodos e Processos Envolvidos na Visão Computacional Bidimensional. Porto Alegre: CPGCC da UFRGS, 1987. 210p. (Dissertação de Mestrado)
- [TIN 80] TING, D. & BIRENDRA, P. Digital Processing Techniques for Encoding of Graphics. Proceedings of the IEEE, New York, v.68, n.7, p.757-769, July 1980.
- [TRI 90] TRINDADE, Ana L. Sinais e sua Representação pelas Transformadas de Fourier. Porto Alegre: CPGCC da UFRGS, 1990. 102p. (T.I. 194)
- [WEL 84] WELCH, T. A. A Technique for High-Performance Data Compression. IEEE Computer, Los Alamitos, v.17, n.6, p.8-19, June 1984.
- [YAS 80] YASUDA, Yasuhiko. Overview of Digital Facsimile Coding Techniques in Japan. Proceedings of the IEEE, New York, v.68, n.7, p.830-845, July 1980.

Capítulo 3

- [BRI 82] BRICKMAN, Norman F. Word Autocorrelation Redundancy Match (WARM) Technology. IBM Journal Res. Develop., Armonk, v.26, n.6, p.681-686, Nov. 1982.
- [CAS 82] CASEY, R. G. & FRIEDMAN, T. D. Automatic Scaling of Digital Print Fonts. IBM Journal Res. Develop., Armonk, v.26, n.6, p.657-666, Nov. 1982.

- [DAI 90] DAINENE, C. A. & SAITO, J. H. Sistema de Reconhecimento de Padrões de Caracteres em Formulários (RECOMPAC). In: JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL, 4-7 Dez., 1990. São Paulo, Anais... São Paulo: USP, 1990. 454p. p. 341-346.
- [GON 90] GONZAGA, A. & RODA, V. O. Método para Reconhecimento de Caracteres Alfabéticos. In: JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL, 4-7 Dez., 1990, São Paulo. Anais... São Paulo: USP, 1990. 454p. p. 237-247.
- [HAR 72] HARMON, Leon D. Automatic Recognition of Print and Script. *Proceedings of the IEEE*, New York, v.60, n.10, p.1165-1176, Oct. 1972.
- [HUS 72] HUSSAIN, A.; TOUSSAINT, G. & DONALDSON, R. Results Obtained Using a Simple Character Recognition Procedure on Munson's Hanprinted Data. *IEEE Transactions on Computers*, New York, v.21, n.2, p.201-205, Feb. 1972.
- [KOZ 71] KOZLAY, D. Feature Extraction in an Optical Character Recognition Machine. *IEEE Transactions on Computers*, New York, v.20, n.9, p.1063-1067, Sept. 1971.
- [MAN 86] Mantas, J. An Overview of Character Recognition Methodologies. *Pattern Recognition*, Oxford, v.19, n.6, p. 425-430, 1986.
- [MAR 88] MARQUES FILHO, Ogé. Character Recognition Using a PAPS. Eindhoven - Netherlands: Computer Vision and Graphics Group of Philips Research Laboratories - Internal Report, 1988. 41p.
- [MAR 89] MARQUES FILHO, Ogé. Automatic Derivation of Template Descriptions for Character Recognition from a Set of Examples. Eindhoven - Netherlands: Computer Vision and Graphics Group of Philips Research Laboratories - Internal Report, 1989. 48p.

- [MAR 90a] MARQUES FILHO, Ogé. Uma Proposta de Reconhecimento de Caracteres Utilizando Correlação com Máscaras Binárias (Binary Template Matching). In: SIBGRAPI'90, 3., 30 Maio - 1 Jul., 1990, Gramado - RS. Anais... Porto Alegre: UFRGS, 1990. 386p. p. 80-89.
- [MAR 90b] MARQUES FILHO, Ogé. Sistema Automático de Reconhecimento Óptico de Caracteres a partir de um Conjunto de Exemplos. In: SIBGRAPI'90, 3., 30 Maio - 1 Jul., 1990, Gramado-RS. Anais... Porto Alegre: UFRGS, 1990. 386p. p. 90-100.
- [MAR 90c] MARQUES FILHO, Ogé. Sistemas de OCR: Fundamentos, Aplicações, Tendências e Perspectivas. In: JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL, 4-7 Dez., 1990, São Paulo. Anais... São Paulo: USP, 1990. 454p. p. 61-80.
- [NAG 82] NAGY, George. Optical Character Recognition - Theory and Practice. In: KRISHNAIAH, P. K. & KANAL, L. N. Handbook of Statistics, Amsterdam: North Holland Publishing Company, 1982, v.2.
- [NAG 83] NAGY, George. Candide's Practical Principles of Experimental Pattern Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, New York, v. PAM-5, n.2, p.199-200, March 1983.
- [PRA 80] PRATT, W. K. et al. Combined Symbol Matching Facsimile Data Compression System. Proceedings of the IEEE, New York, v.68, n.7, p.786-796, July 1980.
- [SCH 78] SCHÜRMANN, Jürgen. A Multifont Word Recognition System for Postal Address Reading. IEEE Transactions on Computers, New York, v.27, n.8, p. 721-732, Aug. 1978.
- [SIL 90] SILVA, Antônio J. F. M. Reconhecimento Automático de Caracteres Impressos. In: SIBGRAPI'90, 3., 30 Maio - 1 Jul., 1990, Gramado-RS. Anais... Porto Alegre: UFRGS, 1990. 386p. p.111-120.

- [STE 61] STEVENS, M. E. Automatic Character Recognition - A State-Of-The-Art Report. National Bureau of Standards, Tech. Note 112, May 1961.
- [SUE 80] SUEN, C. Y.; BERTHOD, M. & MORI, S. Automatic Recognition of Handprinted Characters - The State of the Art. Proceedings of the IEEE, New York, v.68, n.4, p.469-487, April 1980.
- [TAP 82] TAPPERT, C. C. Cursive Script Recognition by Elastic Matching. IBM Journal Res. Develop., Armonk, v.26, n.6, p.765-771, Nov. 1982.
- [ULL 69] ULLMANN, J. R. Experiments With the N-Tuple Method of Pattern Recognition. IEEE Transactions on Computers, New York, v.18, n.12, p.1135-1137, Dec. 1969.
- [WON 82] WONG, K. Y.; CASEY, R. G. & WAHL F. M. Document Analysis System. IBM Journal Res. Develop., Armonk, v.26, n.6, p.647-656, Nov. 1982.
- [PCW 91] PC WORLD MAGAZINE. Put It in Writing - Go Corp's Penpointing, The Notebook That Reads What You Write - Special Issue, PC Magazine, New York, v.10, March 1991.

Capítulo 4

- [ACK 83] ACKLEY, D; HINTON, G. & SEJNOWSKY, T. A Learning Algorithm for Boltzmann Machines. Cognitive Science. Norwood, n.9, p.147-169, 1985.
- [BAL 89] BALANIUK, Remis. Modelos de Redes Neurais: Conceitos Básicos e Evolução Histórica. Porto Alegre: CPGCC da UFRGS, 1990. 122p. (T.I.)
- [BLU 90] BLUM, Adam. Bidirectional Associative Systems in C++. Dr. Dobb's Journal, Redwood City, n.163, p.16-26, April 1990.
- [BRO 87] BROWN, R. J. An Artificial Neural Network Experiment. Dr. Dobb's Journal, Redwood City, p.16-27, April 1987.

- [CAR 83] CARPENTER, G. & GROSSBERG, S. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing*. Duluth, v. 37, p. 54-115, 1983.
- [CAR 90] CARRIJO, Gilberto A. Uso de Redes Artificiais de Neurônios Lineares para o Treinamento e o Reconhecimento de Caracteres com Rotação. In: SIBGRAPI'90, 3., 30 Maio - 1 Jul., 1990, Gramado-RS. Anais... Porto Alegre: UFRGS, 1990. 386p. p.101-110.
- [CAU 90] CAUDILL, M. Neural Networks Primer. *AI Expert - Special Issue*. San Francisco: Freeman Publications, 1990. (Reimpressão de artigos publicados entre Dezembro de 1987 e Dezembro de 1989).
- [DOW 89] DOWLA, F. U. et al. Back-Propagation Neural Networks: Systolic Implementation for Seismic Signal Filtering. *Neural Networks*, Elmsford, v.1, n.3, p.138-153, July 1989.
- [FUK 88] FUKUSHIMA, K. A Neural Network for Visual Pattern Recognition. *IEEE Computer*, New York, v.21, n.3, p. 65-75, March 1988.
- [HEB 49] HEBB, D. O. *The Organization of Behavior*. New York: John Wiley & Sons, 1949.
- [HIN 84] HINTON, G.; SEJNOWSKI, T. & ACKLEY, D. Boltzmann Machines: Constraint Satisfaction Networks that Learn. Carnegie-Mellon University, Dept. of Computer Science, 1984. (Tech. Report CMU-CS-84-119)
- [HIN 86] HINTON, G. & SEJNOWSKI, T. Learning and Relearning in Boltzmann Machines. In: *Parallel Distributed Processing*, Cambridge: MIT Press, 1986, v.1.
- [HIN 85] HINTON, G. Learning in Parallel Networks, *Byte*, Hightstown, v.10, n.4, p.265-276, April 1985.
- [HOP 82] HOPFIELD, J. J. Neural Networks and Physical Systems with Emergent Computational Abilities. In: *Proceedings of the National Academy of Sciences*, Washington, v.79, p.2554-2558, April 1982.

- [KIN 89] KING, Todd. Using Neural Networks for Pattern Recognition. Dr. Dobb's Journal, Redwood City, n.147, p.14-28, Jan. 1989.
- [KIR 83] KIRKPATRICK, S; GELATT, D. & VECCHI, M. Optimization by Simulated Annealing, Science, Washington, v.220, n.4589, p.671-680, 1983.
- [KLI 89] KLIMASAUSKAS, C. Neural Nets and Noise Filtering. Dr Dobb's Journal, Redwood City, n.147, p.32-48, Jan. 1989.
- [KOH 82] KOHONEN, Teuvo. Self-organized formation of topologically correct feature maps, Biological Cybernetics, Berlin, v.43, p.59-69, 1982.
- [KOH 84] KOHONEN, Teuvo. Self-Organization and Associative Memory, 2.ed., Berlin: Springer-Verlag, 1988, 312p.
- [KOH 88] KOHONEN, T. The Neural Phonetic Typewriter. IEEE Computer, New York, v.21, n.3, p.11-22, march 1988.
- [KOH 88a] KOHONEN, T. An Introduction to Neural Computing. Neural Networks, Elmsford, v.1, p.3-16, 1988.
- [KOS 87] KOSKO, Bart. Adaptive Bidirectional Associative Memories, Applied Optics, New York, v.26, p.4947-4960, Dec. 1, 1987.
- [KOS 87a] KOSKO, Bart. Constructing an Associative Memory. Byte, Hightstown, v.12, n.10, p.137-144, Sept. 1987.
- [LAW 90] LAWRENCE, J. Untangling Neural Nets. Dr. Dobb's Journal, Redwood City, n.163, p.38-44, April 1990.
- [LIP 87] LIPPMANN, Richard P. An Introduction to Computing with Neural Networks. IEEE ASSP Magazine, New York, v.3, n.4, 1987, p.4-22, April 1987.
- [MCC 43] McCULLOCH, W. S. & PITTS, W. A Logical Calculus of the Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics, p.115-133, 1943.
- [MIN 69] MINSKY, M. & PAPERT, S. Perceptrons: An Introduction to Computational Geometry. Cambridge: MIT Press, 1969.

- [PAR 82] PARKER, D. Learning-Logic. Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University, Stanford, Oct. 1982.
- [PAR 85] PARKER, D. Learning-Logic. Center for Computational Research in Economics and Management Science, Technical Report, TR-47. MIT, Apr. 1985.
- [PEN 90] PENNA, T. P. & OLIVEIRA, P. M. Redes de Neurônios. [s.l.:s.n.], 1990. Resumo da palestra apresentada na Reunião Anual da SBPC, 42., 1990, Porto Alegre.
- [REI 89] REID, M.; SPIRKOVSKA, L. & OCHOA, E. Simultaneous Position Scale, and Rotation Invariant Pattern Classification Using Third-Order Neural Networks. *Neural Networks*, Elmsford, v.1, n.3, p.154-159, July 1989.
- [ROS 59] ROSENBLATT, R. *Principles of Neurodynamics*. New York: Spartan Books, 1959.
- [RUM 85] RUMELHART, D.; HINTON, G. & WILLIAMS, R. Learning Internal Representations by Error Propagation. La Jolla: Institute for Cognitive Science, Univesity of California at San Diego, Sept. 1985. (ICS Report 8506)
- [RUM 86] RUMELHART, D.; HINTON, G. & WILLIAMS, R. Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press, 1986, v.1.
- [SEJ 86] SEJNOWSKI, T. & ROSENBERG, C. R. NETTalk: A Parallel Network that Learns to Read Aloud. Johns Hopkins University Report, JHU/EECS-86/01, 1986.
- [WAI 89] WAIBEL, A. & HAMPSHIRE, J. Building Blocks for Speech. *Byte*, Hightstown, v.14, n.8, p.235-242, August 1989.
- [WAL 86] WALLACE, D. J. Memory and Learning in a Class of Neural Models. In: Workshop on Lattice Gauge Theory, 1985, Wuppertal, Proceedings... New York: Plenum, 1986.

- [WID 62] WIDROW, B. Generalization and Information Storage in Networks of ADALINE Neurons. In: Self-Organization Systems. Washington: Spartan Books, 1962. p. 435-461.
- [WID 88] WIDROW, B. & WINTER, R. Neural Nets for Adaptative Filtering and Adaptative Pattern Recognition. IEEE Computer, New York, v.21, n.3, p.25-39, March 1988.
- [WID 90] WIDROW, B. & LEHR, M. 30 Years of Adaptative Neural Networks: Perceptron, Madaline, and Back-propagation. Proceedings of the IEEE, New York, v. 78, n.9, p.1415-1441, Sept. 1990.

Capítulo 5

- [FIL 90] FILHO, E. C. D. B. C. Investigation of Boolean Neural Networks based on a Novel Goal-Seeking Neuron. Canterbury: University of Kent, 1990. (PhD Thesis)
- [FIL 91] FILHO, E. C. D. B. C., FAIRHURST, M. & BISSET, D. Adaptative Pattern recognition Using Goal Seeking Neurons. Pattern Recognition Letters, Amsterdam, North-Holland, v.12, n.3, p.131-138, March 1991.
- [OSO 90a] OSORIO, Fernando S. Um Sistema para o Reconhecimento de Caracteres Através de Redes Neurais. In: JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL, 4-7 Dez., 1990, São Paulo. Anais... São Paulo: USP, 1990. 454p. p.336-340.
- [OSO 90b] OSORIO, Fernando S. Utilização dos Transputers em Simuladores de Redes Neurais. Porto Alegre: CPGCC da UFRGS, Nov. 1990. 34p. (Relatório da Disciplina de Arquiteturas Especiais de Computadores)

BIBLIOGRAFIA CONSULTADA

- [BAL 90] BALANIUK, Remis. Uma Rede Neural para o Reconhecimento de Padrões Codificados em Sequências. Porto Alegre: CPGCC da UFRGS, 1990, 204p. (Dissertação de Mestrado)
- [CAR 88] CARPENTER, G. & GROSSBERG, S. The ART of Adaptive Pattern Recognition by a Self-Organization Neural Network. *IEEE Computer*, New York, v.21, n.3, p. 77-90, March 1988.
- [HAY 89] HAYASHI, Y. & OHHASHI, S. Alphanumeric Character Recognition Using a Connectionist Model With the Pocket Algorithm. *Neural Networks*, Elmsford, v.1, n.3, p.175-185, July 1989.
- [HOP 84] HOPFIELD, J. J. Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons. In: *Proceedings of the National Academy of Sciences*, USA, v.81, p.3088-3092, May 1984.
- [HOP 86] HOPFIELD, J. & TANK, D. Computing with Neural Circuits: A Model. *Science*, Washington, v.233, p.625-633, August 1986.
- [JON 87] JONES, W. & HOSKINS, J. Back-Propagation. *Byte*, Hightstown, v.12, n.11, p.155-162, Oct. 1987.
- [KOH 90] KOHONEN, T. The Self-Organizing Map. *Proceedings of the IEEE*, New York, v.78. n.9, p.1464-1480, Sept. 1990.
- [LOT 90] LOTUFO, R. A. & OLIVEIRA, J. R. Sistema de Reconhecimento de Placas de Veículo. In: *JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL*, 4-7 Dez., 1990, São Paulo. Anais... São Paulo: USP, 1990. 454p. p.249-255.
- [OBE 89] OBERMEIER, K. & BARRON, J. Time to Get Fired Up. *Byte*, Hightstown, v.14, n.8, p.217-224, Aug. 1989.

- [OSO 90] OSORIO, Fernando S. Redes Neurais: Modelos de Simulação e Utilização no Reconhecimento de Padrões. In: JORNADA EPUSP/IEEE EM COMPUTAÇÃO VISUAL, 4-7 Dez., 1990, São Paulo. Anais... São Paulo: USP, 1990. 454p. p.37-58.
- [RUM 86a] RUMELHART, D. E. & McCLELLAND, J. L. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press, 1986, v.1.
- [TEI 89] TEIXEIRA, E. P. & GOMIDE, F. C. Condições Extremas para Convergência da Rede Neural de Hopfield. In: Congresso da SBC, 9., 16-21 Julho, 1989, Uberlândia-MG. Anais... Uberlândia: SBC, 1989. p. 481-497.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

"Um Estudo sobre Reconhecimento Visual de Caracteres
através de Redes Neurais".

Dissertação apresentada aos Srs.

Anatólio Laschuk

Prof. Anatólio Laschuk

O Examinador enviou parecer por escrito.

Prof. Dr. Paulo Martins Engel

Philippe Navaux

Prof. Dr. Philippe O. A. Navaux

Ricardo José Machado

Prof. Dr. Ricardo José Machado - Centro Científico IBM-RJ

Visto e permitida a impressão

Porto Alegre, 07/01/92

Anatólio Laschuk

Prof. Anatólio Laschuk
Orientador

Ricardo Augusto da L. Reis
Coordenador do Curso de Pós-Graduação
em Ciência da Computação