Campus de São Carlos

**SISTEMA DE RECONHECIMENTO DE PADRÕES VISUAIS
INVARIANTE A TRANSFORMAÇÕES GEOMÉTRICAS UTILIZANDO
REDES NEURAIS ARTIFICIAIS DE MÚLTIPLAS CAMADAS**

Autor

JOSÉ ALFREDO FERREIRA COSTA

Orientador

Prof. Dr. Adilson Gonzaga

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

SISTEMA DE RECONHECIMENTO DE PADRÕES VISUAIS INVARIANTE A TRANSFORMAÇÕES GEOMÉTRICAS UTILIZANDO REDES NEURAIAS ARTIFICIAIS DE MÚLTIPLAS CAMADAS

JOSÉ ALFREDO FERREIRA COSTA

Dissertação apresentada à Escola de
Engenharia de São Carlos da
Universidade de São Paulo, como parte
dos requisitos para obtenção do título de
Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Adilson Gonzaga

DEDALUS - Acervo - EESC

31100017641

São Carlos
Estado de São Paulo - Brasil
Janeiro de 1996

Class. Tese - EESC
Curr. 5690
Tombo 018/96

Eng. Elétrica

11 0744070

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC-USP

C837s Costa, José Alfredo Ferreira
Sistema de reconhecimento de padrões visuais
invariante a transformações geométricas utilizando
redes neurais artificiais de múltiplas camadas /
José Alfredo Ferreira Costa. -- São Carlos, 1996.
171p.

Dissertação (Mestrado) -- Escola de Engenharia
de São Carlos-Universidade de São Paulo, 1996.
Orientador: Prof. Dr. Adilson Gonzaga

1. Reconhecimento automático de objetos. 2.
Visão computacional. 3. Redes neurais artificiais.
I. Título

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em 15-1-1996
pela Comissão Julgadora:

Prof. Doutor ADILSON GONZAGA - Orientador
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Prof. Doutor NELSON DELFINO D'AVILA MASCARENHAS
(Universidade Federal de São Carlos - UFSCar)

Prof. Doutor LUCIANO FONTOURA COSTA
(Instituto de Física de São Carlos - Universidade de São Paulo)

Presidente da Comissão de Pós-Graduação
Prof. Dr. MOUNIR KHALIL EL DEBS

Coordenador da Área Engenharia Elétrica
Prof. Dr. GERALDO ROBERTO M. DA COSTA

*Dedicado a meus pais,
José Américo e Gisélia Costa,
e a minha esposa, Bianca Costa*

Agradecimentos

Desejo apresentar minha gratidão e reconhecer o esforço de muitas pessoas que até hoje me ajudaram, sobretudo aos meus pais, José Américo e Gisélia Costa, meus irmãos e familiares, e especialmente minha esposa Bianca, pela paciência, compreensão e amor.

Agradeço o suporte financeiro do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, e ao Departamento e à Pós-Graduação de Engenharia Elétrica da EESC, aos professores e funcionários da USP, como meu orientador, Dr. Adilson Gonzaga, e o Prof. Amilcar Cesar, entre outros, que muito me incentivaram durante este trabalho.

Muitos pesquisadores foram gentis, enviando contribuições recentes e mantendo contato através de endereços eletrônicos, entre os quais posso citar: Prof. K. Fukushima (Osaka University - Japan), Prof. S. Hashimoto (Waseda University - Japan), Prof. H. Barrow (Sussex - UK), Prof. W. Pedrycz (U. Manitoba - Canada), Prof. J. Ponce (University of Illinois-USA), Prof. R. M. Rangayyan (U. Calgary - Canada), e outros no Brasil, como: Prof. L. da F. Costa (IFSC / USP), Dr. R. Machado (IBM - Brasil), Prof. N. Mascarenhas (UFSCar), Prof. A. de Carvalho (ICMSC - USP), entre tantos outros.

Por último, porém não menos importante, aos colegas mestrandos do Departamento de Engenharia Elétrica da EESC / USP, do IEEE Seção Sul Brasil, do ramo estudantil IEEE São Carlos e da UFSCar.

Sumário

Sumário	v
Lista de figuras	ix
Lista de tabelas	xi
Lista de abreviaturas	xii
Resumo	xiii
Abstract	xiv
1. Introdução	1
1.1 Uma visão geral dos objetivos deste trabalho	4
1.2 Organização deste trabalho	6
2. Reconhecimento de Padrões Visuais	7
2.1 Introdução	7
2.2 Estrutura básica de um sistema de RPV	8
2.2.1 Aquisição da imagem	9
2.2.2 Pré-processamento	11
2.2.3 Extração de atributos	11
2.2.4 Classificação	13
2.3 Fases dos sistemas de reconhecimento de padrões	16
2.4 Tecnologias para o reconhecimento determinístico de padrões	17
2.4.1 Reconhecimento Estatístico de Padrões	18
2.4.2 Redes Neurais Artificiais	19
2.4.3 Método “Vizinhos Mais Próximos”	20

2.4.4	Outras técnicas	21
2.5	Alguns outros fatores no problema de reconhecimento de padrões visuais	21
2.6	Áreas de aplicação	23
3.	Redes Neurais Artificiais	24
3.1	Introdução	24
3.2	Definição de RNA	26
3.3	Neurônio biológico	26
3.4	Estruturas de redes neurais	27
3.5	Características de redes neurais	31
3.6	Rede neural de múltiplas camadas	32
3.7	Treinamento de RNAs do tipo MLFN	34
3.7.1	Método de treinamento “Gradiente Conjugado”	37
3.7.2	Método de treinamento “Levenberg-Marquardt”	39
3.7.3	Algoritmos híbridos: uso de “simulated annealing” com os métodos anteriores.	40
3.8	Outros tópicos para otimização das redes MLFN	41
4.	Reconhecimento invariante de padrões visuais utilizando redes neurais artificiais	43
4.1	Espaço de Atributos Invariantes	43
4.1.1	Momentos e Invariantes de Momentos	45
4.1.2	Momentos de Zernike	49
4.1.3	Descritores de Fourier	51
4.1.4	Outros atributos invariantes	52
4.1.5	Abordagens de RNAs usando EAI	55
4.1.5.1	Perceptron de múltiplas camadas	55
4.1.5.2	Redes Neurais Booleanas	58
4.1.5.3	Rede de Funções de Base Radial	59
4.1.5.4	Outras redes que utilizam EAI	59

4.2 Invariância pela Estrutura	60
4.2.1 Redes Madaline	60
4.2.2. Redes Neurais de Alta-Ordem (HONN)	61
4.2.3. Neocognitron	63
4.2.4. Arquitetura de ligações dinâmicas	64
4.2.5 Outras abordagens de obtenção de invariância pela estrutura	66
4.3 Invariância por treinamento	67
4.4 Normalização em escala, posição e orientação dos padrões	68
4.5 Outras abordagens envolvendo RNAs.	71
4.5.1 Memórias Associativas	72
4.5.2 Teoria da Ressonância Adaptativa (ART)	73
4.6. Situação Brasileira	74
5. Sistema de reconhecimento automático de padrões visuais	78
5.1 Recursos utilizados	78
5.2 Forma geral do sistema	80
5.2.1 Pré-processamento.	82
5.2.1.1 Binarização	82
5.2.1.2 Filtragem de ruído	83
5.2.1.2.1 Motivação para uso de filtros	85
5.2.1.2.2 Esquema utilizado para filtrar a imagem	86
5.2.1.3 Codificação de regiões conectadas	87
5.2.1.3.1 Tempo de codificação de regiões conectadas e eliminação por área	91
5.2.1.3.1 Utilização de mais de uma filtragem antes da CRC e $ERC_{A<L}$	91
5.2.2 Extração de atributos	95
5.2.3 Classificador: RNAs MLFN.	97
5.2.4 Estágio de interpretação de dados	98
6. Resultados	99
6.1 Bancos de Imagens	99

6.1.1	Comentários sobre as distribuições numéricas de atributos	100
6.2	Classificação	101
6.2.1	Imagens sintéticas	101
6.2.2	Imagens reais	109
6.2.3	Partição do espaço de atributos com o aumento do número de classes	115
6.3	Tempo total de execução.	116
7.	Conclusões	118
	Referências bibliográficas	121
	Apêndice A - Performance da filtragem de ruído e eliminação de regiões de área pequena	143
	Apêndice B - Distribuições dos atributos em relação a transformações geométricas	145

Lista de Figuras

1. Etapas de um sistema de reconhecimento de padrões utilizando EA.	8
2. Efeitos da quantização espacial na imagem	11
3. Espaço Euclidiano bidimensional de atributos e contornos de decisão	15
4. Diagrama do treinamento supervisionado	16
5. Diagrama do treinamento não supervisionado	17
6. Um mesmo padrão transformado geometricamente pode pertencer a diversas classes	22
7. Neurônio artificial	28
8. Função sigmoid ($\lambda=5$)	29
9. Arquitetura básica de uma rede MLFN	34
10. (a) Objeto quadrado e (b) magnitude do espectro de Fourier e (c) representação tridimensional da FFT.	68
11. Hardware básico do sistema	78
12. Diagrama do sistema de reconhecimento de objetos	80
13. Código utilizado para geração de ruído	83
14. Geração de ruído: (a) Imagem original; (b) e a mesma imagem corrompida por 10% de ruído	83
15. Variação dos IMs em relação ao ruído: (a) sem filtragem; (b) uma filtragem.	84
16. Definição da estrutura em linguagem C para cada objeto conectado que satisfaz as condições	86
17. Pseudo código para Codificação de região na direção descendente ($i, j=0 \rightarrow N, M$) a partir de semente P_{xy}	87
18. Filtragem e codificação de regiões em imagem ruidosa e com múltiplos objetos	89
19. Tempo de codificação versus a variação de ruído para uma filtragem	90
20. Imagem ruidosa de objeto e suavização pela mediana	91
21. Resultado da CRC e $ERC_{A \leq L}$ sobre as imagens da figura 20	92
22. Variação percentual dos IMs com o aumento do número de iterações de filtragens	93
23. Imagens de objetos e seus respectivos espectros (módulo) da transformada de Fourier	94
24. Exemplos de objetos utilizados: (a) imagens sintéticas; (b) imagens reais	99
25. Resultado da classificação para GC, erro = 0.1, invariantes de momentos (eq. 23), imagens sintéticas	103
26. Resultado da classificação para GC, erro = 0.01, invariantes de momentos (eq. 23), imagens sintéticas	104
27. Resultado da classificação para GC, erro = 0.001, invariantes de momentos (eq. 23), imagens sintéticas	104
28. Resultado da classificação para LM, erro = 0.1, invariantes de momentos (eq. 23), imagens sintéticas	105
29. Resultado da classificação para LM, erro = 0.01, invariantes de momentos (eq. 23), imagens sintéticas	106
30. Resultado da classificação para LM, erro = 0.1, invariantes de momentos (eq. 23), imagens sintéticas	107

31. Resultado da classificação para LM, erro = 0.7, invariantes de momentos (eq. 33), imagens sintéticas	108
A-1. Imagens de objetos degradadas por ruído, após 1 filtragem do tipo mediana, e após CRC e EA.	144
B-1. Distribuição de valores dos atributos (eq. 23) para objetos de imagens sintéticas	146
B-2. Distribuição de valores dos atributos (eq. 33) para objetos de imagens sintéticas	149
B-3. Distribuição de valores dos atributos (eq. 23) para objetos de imagens reais	155
B-4. Distribuição de valores dos atributos (eq. 33) para objetos de imagens reais	150

Lista de Tabelas

- 1 - Tempo de treinamento das RNAs para diferentes valores de erros. IMs (eq. 23) em img. sintéticas
- 2 - Resultado da classificação para GC, erro = 0.1, IMs (eq. 23), imagens sintéticas
- 3 - Resultado da classificação para GC, erro = 0.01, IMs (eq. 23), imagens sintéticas
- 4 - Resultado da classificação para GC, erro = 0.001, IMs (eq. 23), imag. sintéticas
- 5 - Resultado da classificação para LM, erro = 0.1, IMs (eq. 23), imagens sintéticas
- 6 - Resultado da classificação para LM, erro = 0.01, IMs (eq. 23), imagens sintéticas
- 7 - Resultado da classificação para LM, erro = 0.001, IMs (eq. 23), imag. sintéticas
- 8 - Resultado da classificação k-vizinho mais próximo, IMs (eq. 23), imag. sintéticas
- 9 - Resultado da classificação para LM e vizinho mais próximo, erro = 0.7, IMs (eq. 33), imag. sint.
- 10 - Resultado da classificação para imagens reais: algoritmos LM e GC, IMs (eq. 23).
- 11 - Confusion-matrix para o método vizinho mais próximo, LM e GC.
- 12 - Resultado da classificação para imagens reais: algoritmos LM e GC, IMs (eq. 33).
- 13 - Resultados da partição do espaço de atributos para uma dada configuração
- B1.a - Média dos valores dos IMs (eq. 23) para imagens sintéticas
- B1.b - Desvio padrão dos valores dos IMs (eq. 23) para imagens sintéticas
- B1.c - Coeficiente de variação dos valores dos IMs (eq. 23) para imagens sintéticas
- B2.a - Média dos valores dos IMs (eq. 33) para imagens sintéticas
- B2.b - Desvio padrão dos valores dos IMs (eq. 33) para imagens sintéticas
- B2.c - Coeficiente de variação dos valores dos IMs (eq. 33) para imagens sintéticas
- B3.a - Média dos valores dos IMs (eq. 23) para imagens reais
- B3.b - Desvio padrão dos valores dos IMs (eq. 23) para imagens reais
- B3.c - Coeficiente de variação dos valores dos IMs (eq. 23) para imagens reais
- B4.a - Média dos valores dos IMs (eq. 33) para imagens reais
- B4.b - Desvio padrão dos valores dos IMs (eq. 33) para imagens reais
- B4.c - Coeficiente de variação dos valores dos IMs (eq. 33) para imagens reais

Lista de Abreviaturas

ART	- Adaptive ressonance theory
BP	- Back propagation
CRC	- Codificação de regiões conectadas
CCD	- Coupled charge device
EA	- Extração de atributos
EAI	- Espaço de atributos invariantes
EP	- Elemento processador (neurônio)
ERC_{A<L}	- Eliminação de região conectada com área menor que limiar
DF	- Descritores de Fourier
FFT	- Fast Fourier transform
GC	- Gradiente conjugado
GD	- Gradiente descendente
HONN	- Higher-order neural network
IA	- Inteligência artificial
IM	- Invariantes de momentos
IVA	- Inspeção visual automática
LMS	- Erro médio quadrático
LM	- Levenberg-Marquardt
MLFN	- Multi-layer feedforward network
MLP	- Multi-layer percetron
MR	- Módulo de reconhecimento
MSO	- Métodos de segunda ordem
MZ	- Momentos de Zernike
PNN	- Probabilistic neural network
RNA	- Redes neurais artificiais
RNB	- Redes neurais booleanas
RP	- Reconhecimento de padrões
RT	- Regra de treinamento
RSP	- Reconhecimento sintático de padrões
SA	- Simulated annealing
SOM	- Self organizing map
VC	- Visão computacional

Resumo

As áreas de visão computacional e redes neurais artificiais (RNAs) e suas aplicações, tiveram um enorme progresso em pesquisa e aplicações práticas nos últimos anos. Sistemas de inspeção visual automática têm despertado muita atenção na indústria pois provêem meios econômicos, eficientes e precisos de obtenção de controle de qualidade. Porém, apesar do grande avanço tecnológico, a maioria dos sistemas existentes, com exceção de alguns poucos experimentais, são especializados e foram projetados para inspecionar um único objeto ou peça, de tipo previamente conhecido, e em posição, orientação e distância em relação à câmara altamente restritas. Este trabalho descreve um sistema de reconhecimento de imagens contendo múltiplos objetos de classes aleatórias e tolerante a ruído. Um estágio de pré-processamento filtra parte do ruído e segmenta regiões conectadas da imagem (RCI). A classificação dos padrões é feita com redes neurais de múltiplas camadas a partir de atributos invariantes calculados sobre as RCIs. No final do processo temos uma listagem dos objetos contidos na cena, suas posições e orientações, os quais podem servir de entrada a um sistema de entendimento da cena, de mais alto nível, ou para outras máquinas, como um manipulador automático. Outros parâmetros podem ser utilizados para normalizar, em escala, orientação e posição, os padrões contidos na imagem, para efeito de comparações com imagens e parâmetros dos objetos previamente armazenados em bancos de dados. Dois métodos de treinamento de RNAs foram testados, o gradiente conjugado e o Levenberg-Marquardt, em conjunção com simulated annealing, para diferentes condições de erro e conjuntos de atributos. Imagens reais e sintéticas foram utilizadas para efeitos de testes de classificação correta e rejeição de padrões espúrios. Resultados são apresentados e comentados, destacando a capacidade de generalização do sistema mesmo com elevada degradação da imagem pelo ruído. Uma das vantagens do tipo de RNA empregado é a velocidade de execução, que permite ao sistema ser integrado a uma linha de montagem industrial. O sistema foi projetado com a utilização de recursos acessíveis e de baixo custo, sendo executado em computadores pessoais, e podendo ser facilmente adaptado para o uso em pequenas e médias empresas.

Palavras chaves: Reconhecimento automático de objetos; Sistemas de classificação; Visão computacional; Redes neurais artificiais; Sistemas inteligentes.

Abstract

Computer vision (CV) and artificial neural networks (ANN) are important research fields of artificial intelligence. Visual pattern recognition (VPR) and object recognition (2 or 3-D) are central tasks in a high level computer vision system. Despite the great development in the recent years, most of the current automatic visual inspection systems work with only one kind of pattern at time which has pose highly restricted. This dissertation describes a system designed to recognize patterns and objects in a digital image which have unknown number object types and poses. Such image, which is also degraded by noise, serve as input for the system. After gray level change and filtering, the pixel connected regions (CR) are codified, and the remained noise is eliminated. Invariant features, i.e., moment invariants, serve as inputs for artificial neural networks that perform pattern classification. An interpretation module decode the net's outputs and increases the correct assignment by testing the net's higher outputs values. After all identified patterns were classified, we have an object listing of the scene, their positions and other information, which can be the input for a higher level scene understanding system, that may check for objects relations and could send information for humans or for other machines. Two ANN learning methods were adopted for training the networks, the conjugate gradient and the Levenberg-Marquardt Algorithms, both in conjunction with simulated annealing, for different error conditions and feature sets. Synthetic and real images were utilized for testing the net's correct class assignments and rejections. Results are presented as well as comments focusing the system's generalization capability despite noise, geometrical transformations, object shadows and other degradations over the images. One of the advantages of the ANN employed is the low execution time allowing the system to be integrated to an assembly industry line. The system runs on low cost personal computers, therefore it can be easily adapted for the Brazilian reality and can even be used by little companies and industries.

Keywords: Pattern classification; Visual pattern recognition; Computer vision; Artificial neural networks; Automation.

Capítulo 1

Introdução

A visão é o sentido mais desenvolvido e a ela é dedicado boa parte do “poder neuro-computacional” dos organismos vivos que a possuem. A visão natural, que é um processo complexo e dos mais importantes da natureza, é realizado através de redes neurais biológicas e é responsável pela maior parte do volume da informação que absorvemos diariamente (Zeki, 1993).

Visão computacional¹ (VC) e suas aplicações tiveram um enorme progresso em pesquisa e aplicações práticas nos últimos anos. Entendemos por visão computacional a área de inteligência artificial envolvida na extração de informações de imagens, com a intenção de identificar e classificar objetos e padrões nela contidos, obtendo representações de níveis mais altos para uso do homem ou de máquinas. Sistemas que possuem esta tecnologia estão sendo utilizados em diversas áreas como a robótica, automação industrial, sensoriamento remoto, aplicações militares, e até em produtos comerciais, como reconhecimento óptico de caracteres, que em alguns casos já vêm com editores de texto. Além disso, podemos comprovar o crescimento de VC em pesquisas também pelo aumento anual de congressos internacionais, revistas e outras publicações.

De acordo com Spirovská e Reid (1994), os sistemas robustos de visão computacional são compostos tipicamente de quatro tarefas principais: (1) pré-processamento, para remoção de ruído e acentuação de bordas; (2) segmentação, para delimitar contornos de objetos ou padrões contidos em uma cena; (3) reconhecimento de padrões, para determinar os padrões presentes na imagem; e (4) compreensão da

¹ Alguns autores chamam “visão por computador”.

imagem, para interpretar os relacionamentos entre os padrões contidos na cena. Apesar da maioria dos sistemas existentes utilizarem, em quase todas as etapas, programação convencional, poderíamos implementar as três primeiras tarefas utilizando técnicas de inteligência artificial (IA) conexionista (redes neurais artificiais), (Pal e Pal, 1993), (Costa e Gonzaga, 1995), porém a quarta tarefa deveria ser implementada com técnicas de IA simbólica.

Reconhecimento de padrões (RP) e de objetos (2 ou 3-D), uma tarefa difícil e importante para sistemas de alto nível de VC, objetiva a extração de informações de uma imagem ou classificação de seu conteúdo (Yüceer e Oflazer, 1993). Os avanços nas tecnologias computacionais têm possibilitado a execução de algoritmos complexos de análise de imagens, antes puramente teóricos em máquinas cujo poder computacional cresce exponencialmente. Fatores como processadores mais rápidos e paralelos, grandes quantidades de memória disponíveis a custo baixo, e desenvolvimento de técnicas de processamento e reconhecimento de imagens foram os principais agentes desse progresso. Ainda assim, temos que reconhecer as limitações dos computadores atuais para tarefas complexas, como o reconhecimento de padrões visuais invariante a transformações geométricas no espaço Euclidiano, e buscar soluções para um espectro cada vez mais denso de aplicações, como por exemplo a identificação e inspeção visual automática (IVA) de objetos em tempo real para aplicações na indústria.

Em uma linha de produção industrial podemos utilizar RP como etapa de um sistema de IVA, para identificar e classificar peças manufaturadas. Sistemas de IVA visam a otimização do processo de identificação de padrões defeituosos, podendo checar automaticamente diversos parâmetros medidos dos objetos produzidos, assegurando produtos que não apresentem características com elevada similaridade, em relação aos padrões ótimos requeridos, sejam rejeitados. Diversas vantagens podem ser enumeradas de IVA em relação à inspeção feita por seres humanos. As mais importantes são precisão e velocidade, pois os sistemas podem ser projetados para executar tarefas repetitivas de forma dedicada. Sistemas de IVA têm despertado muita atenção na indústria pois provêem meios econômicos, eficientes e precisos de obtenção de controle de qualidade. Especificações dimensionais podem ser facilmente

asseguradas em determinados sistemas de IVA. Porém, apesar do grande número de vantagens, diversos problemas ainda persistem para diversos sistemas, como a incapacidade de trabalhar com peças diferentes e em posições aleatórias em relação ao sistema de captação de imagem. Em recente revisão, Newman e Jain (1995) analisaram diversos sistemas de IVA e descreveram que a grande maioria desses sistemas, com exceção de alguns poucos experimentais, são especializados e foram projetados para inspecionar um único objeto ou peça, de tipo previamente conhecido, e em posição, orientação e distância em relação à câmara altamente restritas. A possibilidade de tornar o sistema invariante a mudanças de posição e orientação e também do sistema poder trabalhar com vários tipos de objetos diferentes é altamente desejável, visto que isto aumenta a flexibilidade e a tolerância a falhas do sistema, permitindo a adaptação do sistema a diversos tipos de ambientes (uso menos específico).

Outra área da inteligência artificial que teve seu interesse renovado, principalmente, após a segunda metade da década de 1980 é o campo das redes neurais artificiais (RNAs). RNAs são apropriadas para tarefas de percepção, como o reconhecimento, classificação e auto-associação de padrões. Apesar da maioria dos modelos atuais serem baseados mais em métodos numéricos que biológicos, a inspiração para diversas RNAs provém de áreas como a neurofisiologia. Atualmente a pesquisa em RNAs envolve diversas áreas do conhecimento, envolvendo pessoas com propósitos e formações diferentes. Uma vantagem dessa multidisciplinaridade é a riqueza e a exploração heterogênea de tópicos da área. Por outro lado, estes benefícios tornam-se complexos devido a diferenças utilizadas desde a linguagem empregada pelos diferentes pesquisadores, até ao imenso número de contribuições, sendo impossível acompanhar efetivamente o desenvolvimento da área. Para o projeto de classificadores, RNAs apresentam uma série de características desejáveis, tais como a tolerância ao ruído, a capacidade de generalização, o aprendizado adaptativo a partir de exemplos e processamento paralelo. Em ambientes industriais, diversas fontes de ruído, como interferências elétricas, variação da temperatura nos componentes dos equipamento de captação e processamento das imagens, erros de quantização ou binarização, etc., degradam a imagem, implicando em uma diminuição da taxa de reconhecimento dos sistemas de classificação de objetos. Muitos desses fenômenos não são simples de equacionarmos, e dessa forma, a utilização de RNAs contribui para a

robustez do sistema, pois todo treinamento é feito com exemplos de forma que a estrutura da rede se molda às características do problema.

1.1 Uma visão geral dos objetivos deste trabalho

O principal objetivo desse trabalho é o desenvolvimento e implementação de um sistema para reconhecer padrões em imagens contendo múltiplos objetos, invariante a transformações geométricas, e tolerante a certo grau de ruído.

Uma imagem contendo um número de padrões desconhecido serve de entrada ao sistema. Após a binarização e filtragem da imagem, os padrões são identificados e codificados, ao mesmo tempo em que o ruído residual é eliminado. Após a identificação e codificação das áreas conectadas da imagem, i.e., possíveis padrões, extraímos atributos de cada região, invariantes de momentos, os quais servem de entrada para uma rede neural de múltiplas camadas, que classifica os objetos. Um estágio de interpretação do classificador foi adicionado para aumentar a confiabilidade das decisões da rede.

Testes com imagens reais e sintéticas foram efetuados para averiguar a capacidade de tolerância do sistema a fatores como o ruído e outros problemas de captação da imagem, como iluminação deficiente. Dois conjuntos de atributos são testados, os invariantes de momentos de Hu (Kothanzad e Lu, 1990) e outros invariantes de momentos derivados da técnica de Wong et al. (1995). Após a classificação de todos os padrões identificados na imagem, temos uma listagem dos objetos contidos na cena, suas posições e orientações. Estas informações podem servir de entrada a um sistema de entendimento da cena, de mais alto nível, que obtenha relações e gere inferências a partir desses dados e possa prover informação para outras máquinas, como para um manipulador automático que deva realizar operações sobre as peças independente de prévios ajustes ou calibrações, ou para o homem. Outros parâmetros, como o ângulo do eixo principal do objeto, podem ser utilizados para normalizar, em escala, orientação e posição, os padrões contidos na imagem, para

efeito de comparações com imagens e parâmetros dos objetos previamente armazenados em bancos de dados.

Dessa forma, neste trabalho foram abordados os seguintes tópicos: (i) identificar formas de representar padrões visuais, diminuindo a dimensionalidade da informação, mantendo, porém, informações essenciais que permitam sua discriminação; (ii) estudo e implementação de classificadores baseados em redes neurais artificiais, que tenham velocidade elevada de resposta e apresente boas tolerância e capacidade de generalização; (iii) também foi abordada, brevemente, uma metodologia de utilização desse sistema em um outro, maior, de inspeção visual automática voltada para aplicações industriais. Uma das premissas do projeto é a utilização de recursos acessíveis e de baixo custo, de forma que o sistema possa ser finalizado com poucos recursos e que tenha boa performance, podendo ser facilmente adaptado à realidade nacional e sirva para pequenas e médias empresas.

A adição de um sistema de reconhecimento de padrões invariante a transformações geométricas e capaz de trabalhar com múltiplos objetos em uma imagem tem por finalidade principal aumentar a flexibilidade do sistema de IVA, capacitando-o a trabalhar com diversos tipos de classes diferentes, aumentando a robustez e possibilidades, seja para aplicações industriais, militares, segurança, médica, etc. Exemplos de aplicações dessas técnicas são sistemas de classificação e análise de imagens, reconhecimento óptico de caracteres, identificação de alvos, inspeção de objetos e peças, reconhecimento automático de impressões digitais e da íris, entre outros.

1.2 Organização deste trabalho

Este trabalho, além da introdução, contém outros seis capítulos. O capítulo 2 foi dedicado ao reconhecimento de padrões visuais e problemas relacionados. As redes neurais artificiais, mais especificamente a rede de múltiplas camadas, que é a arquitetura escolhida para o projeto do classificador do sistema, e as razões da escolha desse tipo de classificador estão descritas no capítulo 3. Diversos tipos de redes e esquemas tem sido propostos para o reconhecimento de padrões visuais e o capítulo 4 é dedicado a uma breve revisão dos métodos de obtenção de invariância sobre os padrões visuais utilizando redes neurais. Os tipos de transformações geométricas considerados neste trabalho são mudança de escala (dilatações ou contrações), de orientação (rotação em superfície) e de posição (translações horizontais e/ou verticais) do objeto no campo visual². O capítulo 5 apresenta o sistema proposto, a metodologia empregada, os recursos utilizados e alguns resultados preliminares. O capítulo 6 é dedicado a apresentação de resultados e comentários sobre classificação, tempo de treinamento para as RNAs, etc. Finalmente, o capítulo 7 é dedicado a considerações finais, conclusões e futuras otimizações e adições ao sistema com o objetivo de torná-lo mais robusto.

² Campo visual: espaço capaz de enviar estímulos à nossa retina quando mantemos a cabeça parada e o olhar fixo em um ponto determinado. Para nosso propósito, é o tamanho NxM da imagem a ser processada.

Capítulo 2

Reconhecimento de Padrões Visuais

“Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data.” Bob Duin (Delft University).

2.1 Introdução

Reconhecimento de padrões¹ (RP) visuais, uma tarefa difícil e importante para sistemas de alto nível de VC, objetiva a extração de informações de uma imagem ou classificação de seu conteúdo (Yüceer & Oflazer, 1993). Para obtermos uma boa generalização, i.e., a habilidade para reconhecer corretamente padrões não treinados, o sistema deveria ser insensível a transformações geométricas, tais como mudanças na escala (contrações ou dilatações), na orientação (rotação em superfície), e na posição (translações verticais e/ou horizontais) dos objetos no campo visual.

Segundo Mascarenhas e Velasco (1989), o reconhecimento de padrões tem-se desenvolvido, em geral, ao redor de duas grandes linhas de abordagens. A primeira é a classificação de padrões, onde o objetivo é atribuir um objeto desconhecido a uma das possíveis classes. Geralmente são empregadas técnicas estatísticas para o projeto dos classificadores. A segunda abordagem é o reconhecimento sintático (ou estrutural) de padrões (RSP), onde o objeto passa a ser descrito em termos de relações entre suas partes. RSP utiliza geralmente a teoria das linguagens formais. Neste trabalho não descreveremos RSP, o que pode ser visto em Fu (1974) e em Rothman (1992).

¹ Padrões podem ser figuras, sons, aliás, todas as coisas podem ser consideradas padrões. No presente estudo restringimos ao reconhecimento de padrões bidimensionais estáticos e com G níveis de cinza, como por exemplo, imagens de caracteres impressos ou manuscritos ou imagens planares de peças industriais.

De acordo com Parker (1994) os objetivos de um sistema de reconhecimento e/ou classificação são: reconhecer um objeto específico em uma imagem; discriminar objetos com atributos semelhantes em classes; ou rejeitar objetos desconhecidos. Uma classe de objetos é um conjunto de instâncias de objetos similares. O objetivo de um sistema de classificação é atribuir instâncias de objetos aos seus nomes de classes correspondentes.

2.2 Estrutura básica de um sistema de RPV

Sistemas tradicionais de reconhecimento de padrões (RP) dividem a tarefa de reconhecimento em quatro etapas (ver figura 1): (i) aquisição de dados, (ii) pré-processamento de informações, (iii) extração de atributos (EA) e (iv) classificação (ver seção 4.1). Algumas abordagens usando RNAs para RP combinam as duas últimas tarefas em uma única etapa, com ambas extração de atributos e classificação embutidas na própria RNA, ex. o Neocognitron (Fukushima, 1982). O processo de treinamento geralmente é realizado antes do sistema entrar em operação. Porém, em alguns casos, o treinamento do sistema pode ser contínuo, e dessa forma, ele deve ser considerado como uma das etapas de execução do sistema.

Figura 1: etapas de um sistema de reconhecimento de padrões utilizando EA.

Há sistemas em que há possibilidade de comunicação (*feedback*) entre os blocos básicos constituintes. Dessa forma, pode-se utilizar alguma informação de um

processamento de nível mais elevado para melhorar o desempenho de uma tarefa de nível menor, por exemplo, uma alteração de um parâmetro.

2.2.1 Aquisição da imagem

Na primeira etapa, os objetos a serem classificados são captados por sensores de imagens. Os modelos de imagens a serem tratados computacionalmente possuem várias simplificações em relação às imagens reais²: são discretos, bidimensionais, limitados em extensão e em número de cores ou níveis de cinza possíveis. Esta representação numérica (*raster*) da imagem torna possível a armazenagem e processamento por computadores digitais. As imagens são normalmente representadas por funções bidimensionais de intensidade de luz, $f(x, y)$, onde x e y são as coordenadas espaciais, cartesianas, e o valor de f em qualquer ponto (x, y) é proporcional ao brilho da imagem naquele ponto, luminância. Sendo N_x e N_y o valor máximo das coordenadas x e y , respectivamente, temos que para imagens digitais, normalmente f, x, y, N_x e $N_y \in \mathbb{N}$, onde $\mathbb{N}$ é o conjunto dos números naturais³. Tamanhos típicos de imagens são 2^N , contendo G níveis de cinza, onde $N, G \in \mathbb{N}$. Os principais motivos desses valores estão relacionados com o armazenamento de dados e pelo fato que alguns algoritmos, como a transformada rápida de Fourier, FFT, processam sinais e imagens com estas dimensões. Exemplos de tamanhos de imagens comuns são 128×128 , 512×512 , etc. Em relação ao número máximo de tonalidades de cinza utilizados, geralmente escolhe-se 256, também por razões de armazenagem nos computadores: cada elemento de imagem ocupa 1 byte de memória.

Neste trabalho nos concentraremos em imagens monocromáticas, daqui em diante denominadas simplesmente imagens. O processamento de imagens policromáticas envolve o armazenamento de imagens captados em S espectros (tipicamente 3) e envolvendo S imagens da mesma cena, uma para cada espectro. Outra característica das imagens neste trabalho é que elas apresentam um contraste entre os

² Uma imagem real é contínua, possui extensão arbitrária, número arbitrário de cores, e tridimensional.

³ Eventualmente temos alguns algoritmos que processam imagens normalizadas, por exemplo $f \in [0, 1]$, em outros casos como momentos de Zernike exigem que x e y estejam no círculo unitário, i.e., $x, y \leq 1$.

objetos a serem identificados e o fundo da imagem. Isto nos permite efetuar com mais facilidade a segmentação do objeto para posterior processamento.

A forma de obtenção de imagens é diversa. Pode-se utilizar sensores analógicos, ex. câmara de vídeo *Plumbicom*, ou digitais, ex. CCDs, Scanners, etc. Os sistemas de visão que utilizam a primeira classe de sensores, os analógicos, necessitam de um equipamento adicional para conversão do sinal para a forma digital, que é a forma convencional de representação de dados para tratamento e processamento de informações em computadores digitais. Dois tipos de erros principais estão relacionados com a obtenção da imagem dos objetos: a quantização espacial e a quantização de tonalidades (espectrais) da imagem. Como neste trabalho os atributos são medidos de imagens binarizadas não detalharemos o segundo tipo de erro, que pode ser visto em Gonzales e Wintz (1987). Em relação ao primeiro tipo de erro, quantização espacial, diversos tipos de problemas podem ocorrer para um sistema de inspeção visual decorrentes de uma quantização fraca, por exemplo, o cálculo de atributos e subsequente passagem em cascata de valores errados para o classificador. Outros tipos de erros que produzem variação nas medidas são os erros de posicionamento da câmara e da iluminação sobre os objetos. A minimização das fontes de erros aumentam a probabilidade de melhores resultados do sistema. Para uma análise mais detalhada dos efeitos de quantização espacial em uma e duas dimensões ver (Yang e Marefat, 1994).

Podemos ver os efeitos da quantização espacial na figura 2. Um objeto apresentado na figura 2.a é submetido ao processo de digitalização da imagem e resulta na representação mostrada na figura 2.b. Vemos claramente uma perda de definição nas bordas do objeto. Para uma versão do objeto em outra orientação a quantização pode produzir resultados diferentes. A única forma de minimizar o erro de quantização espacial é aumentar o número de elementos de imagem para a mesma área, o que geralmente significa maiores custos.

Figura 2: efeitos da quantização espacial. (a) objeto antes da digitalização. (b) representação digitalizada do objeto em (a).

2.2.2 Pré-processamento

Geralmente utiliza-se uma fase de pré-processamento para executar operações como filtragem de ruído, escalonamento de dimensões ou níveis de cinza, segmentação de padrões, etc., de forma a transformar a imagem ou sinal original em uma imagem (ou sinal) com melhor qualidade. No capítulo 5 voltaremos a discutir esta fase e como o pré-processamento pode melhorar sensivelmente a performance do sistema de RP.

2.2.3 Extração de atributos

Uma das principais etapas do sistema de RPV é a extração de atributos dos padrões a serem classificados. Representar padrões por atributos que são invariantes a transformações e em seguida apresentá-los ao classificador é um método tradicional para reconhecimento de padrões em sistemas de visão computacional (Duda e Hart, 1973). Atributos são propriedades mensuráveis dos objetos a serem reconhecidos na imagem ou em parte dela. Para figuras planas como caracteres impressos, exemplos típicos de atributos são a área, altura, largura, número de vértices e buracos, invariantes de momentos, etc. (Parker, 1994), (Di Zenzo et al., 1992). A tarefa de extração de atributos (EA) funciona como uma forma de compressão de dados⁴, reduzindo a dimensionalidade da informação, eliminando redundância e

⁴ O tamanho do vetor Y é muito menor que $N \times M$, onde N e M são as dimensões da imagem.

aumentando o poder discriminador entre as classes. EA tem um papel muito importante no processo de reconhecimento, influenciando, muitas vezes, em proporção direta as taxas de reconhecimento.

O projeto de um sistema de reconhecimento visual usando EA envolve, inicialmente, a escolha de um conjunto de atributos, o que é na maioria das vezes *ad hoc*. Entretanto, de acordo com Parker (1994) e Di Zenzo et al. (1992), podemos efetuar esta escolha baseando-se em algumas propriedades desejáveis para os atributos, como (lista não exaustiva):

- 1) Possuir uma boa capacidade discriminante de classes: as curvas de distribuição para todas as classes não deveriam se sobrepor (ver apêndice B);
- 2) Facilidade e velocidade de computação: é muito melhor utilizar muitos atributos que são fáceis de calcular do que outros complexos e com possibilidade de baixa precisão;
- 3) Pequena variância em cada classe: bons atributos deveriam gerar valores semelhantes para instâncias diferentes dos mesmos objetos. Fatores como instabilidades numéricas, arredondamentos nos cálculos dos atributos, ruído e oclusão podem modificar estes vetores. Atributos deveriam ter boa aproximação numérica.
- 4) Não ser fortemente afetada pela taxa de amostragem: a natureza discreta das representações das imagens nos computadores pode causar erros de arredondamentos e instabilidades numéricas.

Ghorbel (1994) ainda cita outros critérios como: *Compleitude da descrição*, que garante que se duas imagens apresentam as mesmas representações invariantes então os objetos que estas representam devem ser também similares; Uma *distância* entre formatos (objetos) que é invariante a uma classe de transformações geométricas; e o critério de estabilidade, que garante que pequenas variações no espaço de imagem devem corresponder a pequenas variações no espaço de atributos.

EA para representação de padrões parece ser uma estratégia similar à empregada pela natureza. Por exemplo, temos cerca de 126 milhões de fotorreceptores

em cada olho e o nervo óptico possui apenas cerca de 800 mil fibras nervosas (Marr, 1982), havendo dessa forma uma compressão de dados para cerca de 0.635% do espaço de informações original.

Podemos considerar EA como uma função descrita em (1), onde $f(x, y)$ é a imagem sob consideração e $\bar{Y}$ é o vetor de atributos que será utilizado pelo classificador,

$$\bar{Y} = F\{f(x, y)\} \quad (1)$$

Um problema associado com EA é a necessidade de um estágio de pré-processamento para o cálculo dos atributos, o que eleva o tempo total de resposta do sistema de RP (Breuel, 1992). Outro problema com EA é que informações importantes podem ser perdidas em uma redução extensiva de dados (Yüceer e Oflazer, 1993).

2.2.4 Classificação

A etapa central do sistema de RPV é o classificador, que efetua uma transformação ou mapeamento entre os atributos e as classes dos padrões. Baseado nos atributos calculados $Y_1, Y_2, \dots, Y_n$ o classificador deriva, baseado em uma função de decisão de um modelo de classificador, a classe $C_0, C_1, \dots, C_m$, onde $C_x, x = 1, 2, \dots, m$, representa os tipos de objetos a serem discriminados. A classe C_0 é utilizada quando o classificador é incapaz de reconhecer a instância do objeto, classe de rejeição. A tarefa de classificação pode ser descrita pela equação 2:

$$C_x = D(\bar{Y}) = D[F\{f(x, y)\}] \quad (2)$$

Os atributos $Y_1, Y_2, \dots, Y_n$, podem ser vistos como coordenadas em um espaço n-dimensional de atributos (EA), e os objetos a serem identificados podem ser representados por pontos neste espaço (P). As m classes particionam o EA formando

m subconjuntos, durante a fase de treinamento do sistema. Estes subconjuntos são as regiões de decisão. Um objeto será atribuído à classe C_x se seus pontos de atributos representativos caírem em apenas uma região correspondente. Caso contrário, se P 's caírem em mais de uma região, ou em nenhuma, o classificador deverá rejeitar o objeto (Di Zenzo et al., 1992). Diversos fatores podem contribuir para o aumento de variações e da incerteza sobre os vetores de atributos, ex., ruídos nos sensores de medidas, erros numéricos de arredondamento e truncamento nos cálculos desses atributos, a representação digital das imagens, etc.

A figura 3 apresenta um gráfico da distribuição do espaço Euclidiano bidimensional de atributos, para o caso de duas classes e dois tipos de atributos (caso extremamente simplificado em relação aos casos reais). Este espaço é definido por regiões de decisão, as quais são separadas por fronteiras ou contornos de decisão. Nesta figura também estão apresentadas duas curvas chamadas contornos de decisão. As funções que geram estes contornos são chamadas de funções discriminantes e podem ser lineares, ex. o contorno de decisão 1, ou não lineares, ex. o contorno de decisão 2. Estas funções formam a base de diversas técnicas de reconhecimento de padrões. Considerando n como a dimensão do vetor de atributos e m a dimensão do vetor de classes, as funções discriminantes mapeiam o espaço de atributos no espaço de classificação ($\mathcal{R}^n \rightarrow \mathcal{R}^m$). Para uma dada amostra (x_1, x_2) de classe desconhecida, podemos determinar a classe pertencente submetendo os valores à função e comparando os resultados, por exemplo, através do método do vizinho mais próximo (ver seção 2.4.3).

Para o caso ideal de imagens sintéticas e uma transformação invariante ideal, os valores dos atributos seriam concentrados na média dos valores dos atributos, com desvio padrão zero. Pelo fato que os sistemas de captação de imagens introduzem distorções, ruídos, etc., além da representação digital introduzindo erros de quantizações, os valores de cada atributo concentram-se em determinada média e sua distribuição é similar a uma distribuição Gaussiana. No caso real temos, em geral, diversos fatores complicadores como superposições de instâncias de atributos, cruzamentos de agrupamentos, etc., o que torna a decisão por uma classe mais difícil.

Na figura 3, por exemplo, a instância marcada pelo símbolo “ * ” seria classificada como pertencente à classe ‘B’ caso utilizássemos o contorno de decisão 1. Caso utilizássemos o contorno de decisão 2, a classe escolhida seria a ‘A’. No capítulo 3 veremos como RNAs podem ser úteis na geração de mapeamentos complexos onde não podemos controlar as regras de decisão, i.e., os contornos de decisão, para um dado problema de classificação. Dependendo do tipo de RNAs utilizado, podemos formar diversos tipos de contornos dividindo as regiões de decisão, como os hiperplanos, funções quadráticas, radiais, etc. (Vasconcelos, 1995).

Figura 3: Espaço Euclidiano bidimensional de atributos e contornos de decisão

2.3 Fases dos sistemas de reconhecimento de padrões

Sistemas de reconhecimento apresentam tipicamente três fases:

(1) *Treinamento*. Nesta fase o sistema aprende a discriminar as classes através da apresentação de instâncias, exemplos, representativas das classes de padrões. O treinamento pode ser supervisionado ou não supervisionado. Dizemos que para o primeiro caso o sistema aprende com um instrutor, i.e., para cada entrada apresentada ao sistema é fornecida a resposta desejada. Portanto, cada instância submetida a treinamento possui um vetor da forma $P_i = \{ Y, C_i \}$, onde $Y = \{ Y_1, Y_2, \dots, Y_n \}$ é o vetor de atributos e C_i é a classe correspondente da amostra. No segundo caso, o sistema é autônomo e se auto-organiza a partir dos dados de entrada, extraíndo propriedades desses e refletindo essas propriedades na saída. Usualmente, utiliza-se uma distância entre o vetor de atributos da instância cuja classe é desconhecida e um vetor rotulado. Caso esta distância seja menor que um valor preestabelecido a instância é atribuída àquela classe. Caso contrário, uma nova classe é criada (*clusters*). As figuras 4 e 5 ilustram estas duas formas de treinamento. O classificador no diagrama é uma rede neural artificial (RNA), porém, outros tipos de classificadores utilizam este mesmo esquema, trocando-se apenas o bloco RNA pelo outro tipo. Exemplos de algoritmos de aprendizado supervisionados para RNAs são “backpropagation”, gradiente conjugado, “Cascor”, etc. RT não supervisionadas para RNAs geralmente são ligadas diretamente a uma arquitetura de rede, por exemplo, a rede SOM (Mapas auto-organizantes) de Kohonen, a ART (Teoria da Ressonância Adaptativa) de Grossberg, o Neocognitron (Fukushima, 1982), etc.

Figura 4: Diagrama do treinamento supervisionado

Figura 5: Diagrama do treinamento não supervisionado

(2) *Teste*. Nesta fase apresenta-se um conjunto de padrões não vistos na etapa anterior (i.e., não utilizado para o treinamento). Se o sistema consegue derivar as classes dos objetos corretamente dizemos que o sistema possui capacidade de generalização. Caso o sistema não atinja os níveis desejados, podemos reestruturar a arquitetura do sistema, modificando parâmetros, ou aumentar a base de dados de treinamento.

(3) *Execução*. Se durante a fase de teste o sistema apresentou taxas de reconhecimento aceitáveis para a aplicação desejada, o sistema está pronto para execução prática. Geralmente a taxa de reconhecimento correta de padrões é inversamente proporcional ao número de classes que o sistema pode discriminar, i.e., a diversidade de padrões, para uma estrutura fixa do sistema.

2.4 Tecnologias para o reconhecimento de padrões

Diversas tecnologias tem sido reportadas na literatura para o reconhecimento de padrões. Nesta seção abordaremos brevemente métodos de classificação numéricos, como os métodos estatísticos, redes neurais artificiais, o método da menor distância a outra classe (vizinho mais próximo), entre outros. Apesar de técnicas como o “*pattern matching*”, i.e., casamento de padrões, serem sensíveis a ruído, transformações geométricas, etc., em algumas aplicações estas e outras técnicas simples podem ser as

escolhidas, por exemplo, se tomarmos em consideração os fatores tempo de implementação, complexidade, e até mesmo, o grau de precisão requerido pelo sistema, que em alguns casos não é tão elevado.

2.4.1 Reconhecimento Estatístico de Padrões

Reconhecimento estatístico de padrões está fortemente ligado a teoremas como o de Bayes, calculando-se a probabilidade a posteriori das classes dos objetos (Duda e Hart, 1973), (Mascarenhas e Velasco, 1989), e ainda hoje é bastante utilizado na solução de diversos problemas práticos (Schalkoff, 1992). Grande parte dos algoritmos de classificação estatísticos assumem possuir algum conhecimento sobre a distribuição das variáveis randômicas utilizadas para classificar. Frequentemente utiliza-se a distribuição normal como modelo, e a base de dados de treinamento é utilizada para estimar o vetor de médias e a matriz de covariância das populações (Masters, 1995).

Para i classes (C_i), e tendo as funções densidade de probabilidade associada a cada classe $P(C_i)$, onde $0 \leq P(C_i) \leq 1$, podemos utilizar a regra da probabilidade condicional (Bayes) para estimar a classe de um padrão. Sendo o vetor de atributos $Y = (y_1, y_2, \dots, y_n)$, que é o conjunto de medidas da instância que desejamos determinar a classe, a probabilidade condicional $P(C_i/Y)$ nos fornece uma confiança de quanto correlacionado nossas medidas estão dos valores padrões para a classe. Pode-se decidir por uma classe I através da comparação dessas probabilidades para cada classe e tomando o maior valor como a classe resultante: se $P(C_i/Y) > P(C_j/Y)$ para $i = 1, 2, \dots, n$, para $i \neq j$, (Beale e Jackson, 1990). Infelizmente a função densidade de probabilidade condicional, $P(Y/C_i)$, requerida pela regra de Bayes, equação 3, não é, em geral de fácil obtenção, nos casos reais, devendo portanto ser estimada através de alguma forma, o que influenciará diretamente na qualidade final do classificador. Na prática, geralmente, obtemos uma média calculada a partir de exemplos e assumimos que os valores apresentam uma distribuição normal.

$$P(C_i / Y) = \frac{P(Y / C_i) \cdot P(C_i)}{\sum_j P(Y / C_j) \cdot P(C_j)} \quad (3)$$

Seguindo Vasconcelos (1995), o RP estatístico fornece uma abordagem formal e precisa onde regras de decisão podem ser formuladas levando em consideração a minimização de uma medida de risco. A desvantagem de RP estatístico é que erros podem ocorrer quando os desvios padrões das distribuições de classes são elevados, podendo ocorrer o problema de sobreposição de classes em determinada região do espaço de atributos.

2.4.2 Redes Neurais Artificiais

Após a segunda metade da década de 1980 o interesse em RNAs para RP foi restabelecido, após o trabalho de Rumelhart et al. (1986), que estabeleceu o algoritmo “*back propagation*” (BP) para treinamento de redes de múltiplas camadas. Atualmente existem centenas de arquiteturas e algoritmos de treinamento de RNAs (ver capítulo 4). A maioria das redes tem sua arquitetura ligada diretamente a função, i.e., poucas são as redes de uso geral, que servem por exemplo para problemas de classificação, otimização, mapeamento, etc. Diversas características presentes nesta tecnologia fazem dela uma das mais promissoras na área de RP.

A idéia básica é que unidades de processamento simples computando certas funções matemáticas, dispostas em uma ou mais camadas, interagindo umas com as outras e com o ambiente, apresentam elevado poder computacional (Vasconcelos, 1995). Uma das vantagens em relação aos métodos estatísticos é a não necessidade de ter as densidades de probabilidade para cada classe de objetos. Embora não se saiba profundamente, que comportamento e operação, do ponto de vista matemático, teoricamente esperado para a maioria das classes de RNAs, tem se constatado experimentalmente que muitas RNAs projetadas para servir como classificadores fornecem respostas na camada de saída que estimam a probabilidade a posteriori Bayesiana. (Richard & Lippman, 1991). Para uma revisão descrevendo métodos de

classificação envolvendo RNAs e classificadores estatísticos, ver Ripley (1994). Um modelo de rede fortemente baseado em estatística é a PNN (probabilistic neural networks) (Masters, 1995). O capítulo 3 desse trabalho é dedicado a esse tipo de tecnologia para classificação de padrões.

2.4.3 Método “Vizinhos Mais Próximos”

Classificação baseada no método *vizinho mais próximo* é um método não-paramétrico e efetua a decisão baseada na distância entre a instância de classe desconhecida e os padrões previamente rotulados (Cover e Hart, 1967). Este método requer uma base de dados e uma extensiva comparação entre valores armazenados e os valores do padrão a ser rotulado. Diversos tipos de métrica podem ser utilizados para calcular a distância entre os atributos do objeto e dos padrões armazenados. A distância de *Hamming* $\{ H = \sum (|x_i - y_i|) \}$ proporciona um número não negativo para a variação entre vetores. Este tipo de medida é usualmente empregado para vetores binários, alterando-se $|x_i - y_i|$ para $x_i \text{ XOR } y_i$. Outras medidas de distância são a Euclideana, que é a mais comum e está definida em (4), a *distância City block*, que é semelhante à Euclidiana com exceção do cálculo da raiz e do quadrado da diferença entre componentes dos vetores, a *distância quadrada*, que é simplesmente o máximo valor das diferenças dos componentes dos vetores, etc.

$$D_{\text{euc}}(X, Y) = \sqrt{\left(\sum_{i=1}^n (X_i - Y_i)^2 \right)} \quad (4)$$

Atualmente utiliza-se não o vizinho mais próximo, mas um número, i.e., K vizinhos mais próximos. A decisão é feita baseada no máximo número de elementos de uma mesma classe dentre estes K elementos.

2.4.4 Outras técnicas

Técnicas não determinísticas como o RSP e lógica nebulosa (ou *fuzzy*) tem sido aplicada a diversas áreas de IA e classificadores fuzzy tem sido propostos (Pedricz, 1990). Lógica nebulosa tem sido também estendida a RNAs com as *redes neurais fuzzy* com a idéia que padrões passam a ter graus de pertinência a todos os tipos de classe, e não o método tradicional de critério mutuamente exclusivo: “pertence ou não”. Além disto, classificadores fuzzy se assemelham a forma em que classificamos objetos pois representam estes em uma forma lingüística, e não numérica. Outras tecnologias têm sido sugeridas e propostas para RP como métodos baseados em regras e bases de conhecimento (Di Zenzo et al., 1992), uso de estruturas do tipo fractais para objetos naturais, algoritmos genéticos para otimização do treinamento de classificadores, arvores de decisão (Al-Emani e Usher, 1990), etc. Além de sistemas digitais de processamento, sistemas ópticos de processamento para RP tem sido bastante estudados principalmente em situações onde o tempo de resposta deve ser o mínimo possível. Grande parte desses sistemas utilizam filtros especiais para convolução de mascaras com a imagem, processam a transformada de Fourier em duas dimensões, etc. Maiores dados podem ser vistos em Reichel et al. (1994).

2.5 Alguns outros fatores no problema de reconhecimento de padrões visuais

Além da mudança da representação numérica (matricial) dos padrões visuais causada por transformações geométricas, a impossibilidade na maioria das vezes de utilização de informação contextual na identificação de padrões visuais, como peças manufaturadas, é um mais um dado a ser levado em conta no problema. Para sistemas de RPV de caracteres impressos ou manuscritos, muitas vezes podemos inferir a classe do padrão através da informação contextual. As vezes o mesmo padrão rotacionado de alguns graus pode ser classificado como pertencente a outras classes. Por exemplo, a figura 4 mostra quatro padrões idênticos cada um com orientação diferente do outro de $N \cdot 90$ graus, onde N é um número inteiro e diferente de zero.

Figura 6: um mesmo padrão transformado geometricamente pode pertencer a diversas classes

Vemos na figura 6 que o padrão (a) poderia ser classificado como o dígito “3”. O padrão (b) poderia ser classificado como a letra (símbolo) “E”, o padrão (c) poderia ser interpretado como a letra “M”, e finalmente o padrão (d) poderia ser a letra “W”. Outros exemplos são os dígitos 6 e 9; as letras d e p, e as letras b e q. Em tais casos, a solução para escolher qual é a classe a qual o padrão pertence é tentar utilizar alguma informação contextual, por exemplo, se o símbolo faz parte de uma *string* (conjunto de símbolos), é possível identificá-lo tendo um dicionário de palavras em banco de dados. Por exemplo, fica evidente que a falta ou mal classificação de um padrão representado pelo caractere coringa “*” na *string* AUTOMOV*L é o símbolo “E”. Dessa forma, a informação contextual poderia ser uma aliada ao processo de reconhecimento, averiguando o sentido da resposta fornecido pelo classificador.

Um outro problema que pode ocorrer nos sistemas de RPV é a oclusão de objetos. Neste caso rotinas de segmentação podem identificar e separar partes de objetos diferentes. A extração de atributos de uma parte do objeto leva, em grande maioria dos casos, a números diferentes dos esperados, o que em cascata produz uma classificação errada. Para casos de RPV com oclusão pode-se utilizar memórias associativas, por ex. a rede de Hopfield, onde fornecendo-se apenas uma parte do objeto a rede itera até o restabelecimento completo do padrão. Um outro problema no RPV é quando não existe a separação entre símbolos ou objetos, i.e., os padrões não estão isolados. Neste trabalho não detalharemos estes dois últimos tipos de problemas, oclusão e não-separação de padrões.

2.6 Áreas de aplicação

Atualmente diversas áreas tem utilizado procedimentos de reconhecimento de padrões e análise de imagens. Áreas como o processamento de informações escritas e documentação tem utilizado análise de imagens para reconhecimento de caracteres e documentos. Na área médica, temos a identificação e diagnóstico automático de tumores, contagem de células do sangue, sistemas de visualização, etc. Na indústria a inspeção automática é uma das principais aplicações. Na área de segurança temos a identificação de pessoas através de impressões digitais, reconhecimento de face, da Íris, e do formato da mão. Em aplicações militares, a identificação de alvos táticos, interpretação de imagens de satélites, etc. Na robótica diversos sistemas de visão e reconhecimento de padrões tem sido desenvolvidos, seja para aplicações de exploração em lugares perigosos a seres humanos, como em outros planetas ou em usinas nucleares, seja para auxílio a deficientes físicos, sistemas de visão de auxílio a dirigibilidade de automóveis, aviões, etc. Outras áreas de aplicação incluem realidade virtual, entretenimento, astronomia, etc.

Capítulo 3

Redes Neurais Artificiais

"When the only tool you have is a hammer, Everything looks like a nail ..." L. A. Zadeh

3.1 Introdução

A inteligência sempre despertou interesse para a humanidade. A possibilidade de construir máquinas inteligentes tem vislumbrado cientistas há centenas de anos. Apesar de grande desenvolvimento tecnológico nas últimas décadas, tarefas como interpretação automática de imagens são tarefas difíceis de serem resolvidas utilizando-se o paradigma tradicional da computação, i.e., a arquitetura de von Neumann.

Diferentemente do processamento serial, os sistemas visuais biológicos empregam estratégias massivamente paralelas (redes de neurônios interconectadas) para efetuar tarefas visuais em tempo real. A eficiência de arquiteturas paralelas sobre o processamento serial é refletido pela eficiência na qual processamos melhor imagens do que a linguagem (Dony e Haykin, 1995).

As pesquisas na área de redes neurais artificiais (RNAs) iniciaram-se com o trabalho pioneiro de McCulloch e Pitts (1943) e sua aplicação para reconhecimento de padrões (RP) foram alvos de grandes pesquisas na década de 1960's. O trabalho de Minsky e Papert (1969) levou a área de RNAs a um estado de estagnação por sugerir que RNAs não eram capazes de solucionar problemas simples como o OU-EXCLUSIVO para duas ou mais variáveis. Somente após o aparecimento do algoritmo *back propagation* (BP) (Rumelhart et al., 1986) para treinamento de redes de múltiplas camadas, o interesse por RNAs como uma poderosa ferramenta para RP foi renovado.

Computação neural¹ é um conceito computacional alternativo para a abordagem convencional baseada na seqüência de instruções programadas. Atributos do cérebro humano como robustez, tolerância a falhas, manipulação com informações incompletas, ruidosas e/ou inconsistentes, paralelismo, dissipação de pouca potência, etc., são características desejáveis nos sistemas de inteligência artificial. Apesar de alguns modelos de RNAs serem baseados em conceitos biológicos (O'Reilly e Johnson, 1994), (Peres e Hochstein, 1994), muitas abordagens não tentam ser realistas, em detalhes, do ponto de vista biológico. De fato, muitas RNAs são modelos extremamente simplificados comparados às redes neurais biológicas. O estudo em conjunto de teorias biológicas e os modelos artificiais podem propiciar avanços para as duas áreas. O estudo de RNAs visando o modelamento e implementação prática de teorias sobre os sistemas biológicos aliado à compreensão de fenômenos complexos do cérebro podem nos conduzir a novas gerações de processadores com características como maior robustez, paralelismo, eficiência, etc. (Fukushima, 1993), enquanto que a implementação de modelos em computadores pode comprovar a veracidade ou falha dos modelos computacionais idealizados. Muito do trabalho efetuado até o presente envolve a simulação de programas em software em computadores de uso geral. Apesar de hardware de uso específico já ser disponível para alguns tipos de redes, espera-se no futuro que computadores e processadores paralelos implementem, de forma mais eficiente, diversos tipos de RNAs.

A área de pesquisa de RNAs está envolvida em quatro grandes atividades (Meech e Kumar, 1994): (i) algoritmos de aprendizado; (ii) configurações ou padrões diferentes de conexões entre camadas e neurônios; (iii) implementação em hardware: processamento paralelo, óptico, etc.; e (iv) operação e estrutura de redes neurais biológicas.

Atualmente as áreas de aplicação de RNAs são inúmeras. Teoricamente RNAs podem realizar qualquer tipo de computação. Na prática as RNAs são especialmente aplicáveis em problemas que requerem mapeamentos complexos, onde temos grande volume de dados disponíveis porém não temos ou é difícil descrever regras para o comportamento do sistema. Além disso, RNAs têm a capacidade de serem tolerantes a

¹Do inglês "Neural Computation".

ruído ou erros (variações) nos dados. Algumas das aplicações de RNAs são: processamento de linguagem, conversão de texto para voz (Ex. o NETTalk), e processamento de linguagem natural; análise de imagem, reconhecimento de caracteres manuscritos e impressos, classificação de alvos táticos, inspeção industrial; compressão de imagens; processamento de sinais; previsão mercadológica, modelos econômicos e financeiros; sistemas de controle inteligentes e adaptativos; memórias associativas, etc.

3.2 Definição de RNA

Podemos definir² uma rede neural artificial como um dispositivo de processamento, algoritmo ou equipamento de hardware, cujo projeto foi motivado pela estrutura e funcionamento das redes neurais biológicas ou sistemas nervosos. Uma RNA é uma rede de diversos processadores simples (neurônios), cada um possuindo uma pequena quantidade local de memória, que são conectados por canais de comunicações (conexões) por onde são transportados dados numéricos. Os neurônios operam apenas sobre seus dados locais e sobre as entradas que recebem via conexões.

3.3 Neurônios biológicos

Os neurônios são os blocos básicos da construção do sistema nervoso e do cérebro humano (Meech e Kumar, 1994). O cérebro humano possui cerca de 10^{11} neurônios, cada um conectado a cerca de outros 10^4 , em média. Um neurônio recebe sinais através de uma rede semelhante a uma árvore de fibras nervosas chamados dendritos, em um processo de transmissão bastante complexo, baseado em fenômenos químicos. Neste processo, substâncias transmissoras são liberadas da célula pré-sináptica e absorvidas pela célula pós-sináptica, resultando na elevação ou diminuição do potencial elétrico interno da célula receptora. O efeito deste processo é semelhante a uma integração de sinais e se o potencial atingido supera um determinado limiar, um pulso, ou potencial de ação de largura e duração fixos, é enviado através do axônio e

² Adaptado do Neural Networks frequent answered questions - Internet.

dizemos que o neurônio disparou. Um sinal elétrico pode ser medido como resultado do processo de transmissão de informação. A quantidade de sinal transmitido depende da quantidade de neuro-transmissores liberados pelos axônios e recebidos pelos dendritos. A base da memória do cérebro é formada pelos neurônios e pelas sinapses. Durante o processo de aprendizado a intensidade das sinapses é modificada. Outra característica dos neurônios biológicos é que após um disparo, a célula possui um tempo chamado 'período refratário' em que não pode disparar.

3.4 Estruturas de RNAs

Apesar da inspiração biológica para modelos de neurônios e suas redes interconectadas para processamento de informação, atualmente grande número de modelos são fracamente baseados nos conhecimentos recentes de ciências como neurologia.

RNAs são especificadas pela arquitetura da rede, características dos neurônios (elementos processadores - EP), dinâmica, e as regras de treinamento ou aprendizado (RT). A RT especifica como os pesos podem ser adaptados durante o aprendizado para melhorar sua performance (Hinton, 1992) e a maioria delas possui um relacionamento próximo ao tipo de topologia da rede. Os métodos de aprendizado são a pré-programação, regras de treinamento supervisionadas ou não supervisionadas. Geralmente a alteração dos parâmetros de uma rede só é permitida durante a fase de treinamento, permanecendo estáticos durante as fases de teste e execução. Exemplos de RT supervisionada são o BP, a regra *Delta*, etc. Exemplos de RT não supervisionada são SOM (Kohonen) e o Neocognitron (Fukushima, 1989).

A arquitetura de uma rede está, geralmente, relacionada com suas funções. Seguindo Maren et al. (1991), pode-se descrever as estruturas em diversos níveis: micro-estrutura, a menor estrutura de uma rede, i.e., o neurônio ou elemento processador (EP); meso-estrutura, a rede, no qual o projeto está relacionado com sua função; e a macro-estrutura, duas ou mais meso-estruturas trabalhando em conjunto para efetuar tarefas mais complexas.

Os neurônios ou EPs tipicamente funcionam como integradores não lineares de sinais. Um neurônio coleta sinais de outros neurônios, integra os dados, adiciona um limiar e passa o resultado através de uma função de ativação (ver figura 6). Como resultado da operação é chamada ativação do neurônio e este valor será ponderado e coletado pelos neurônios da próxima camada. As características que diferem os neurônios em uma RNA são: (i) o tipo da função de ativação, que pode ser a sigmoid, linear, base radial (Gaussiana) ou função degrau; (ii) a natureza dos sinais utilizados para comunicação entre EPs, que podem ser analógicos ou booleanos; (iii) a dinâmica do EP, determinístico ou estocástico; (iv) a adição de outros parâmetros ao EP, como limiares adaptativos, ganhos, a possibilidade de condução do potencial de ação por mais de um ciclo, etc. A equação (5) descreve o processamento efetuado por um neurônio simples como o mostrado na figura (7).

Figura 7: o neurônio artificial

$$Y_j = f\left(\sum_{i=1}^n w_{ij} \cdot x_i - \theta_j\right) \quad (5)$$

onde x_i é o estado do neurônio i da camada precedente, w_{ij} é o peso da conexão entre o neurônio i de uma camada k ao neurônio j da camada $k+1$, e θ_j é limiar do neurônio j da camada $k+1$.

Matematicamente, o resultado do somatório das componentes do produto interno entre o vetor de entrada ($\mathbf{x}$) e o vetor de pesos do neurônio ($\mathbf{w}$) é mapeado através da função de ativação, produzindo a resposta do neurônio Y_j . A função sigmoid é uma das mais utilizadas pois o cálculo de sua derivada de primeira ordem é

numericamente fácil de se determinar, o que implica em uma simplificação do esforço computacional do método de minimização do erro da rede, como no algoritmo *back propagation*. Outro exemplo de função utilizada é a tangente hiperbólica. A função sigmoid é dada por (6), onde λ é um parâmetro que controla a inclinação da curva. Para $\lambda=5$, a função f tem a forma apresentada na figura 8.

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (6)$$

Figura 8: função sigmoide ($\lambda=5$)

Meso-estruturas de RNAs estão relacionadas com a organização física - arranjo dos neurônios (Simpson, 1990), (Maren et al. 1991). As características das meso-estruturas permitem a discriminação das redes em classes. As redes são compostas de diversos EPs e podem possuir uma, duas ou várias camadas (multi-camadas). Os neurônios podem ser organizados em diversos padrões de conexão. Algumas redes permitem apenas o fluxo de sinais para frente (feed-forward). Outras podem possuir conexões retroativas (feedback), laterais, conexões que pulam camadas, competitividade entre neurônios e normalizações de valores numéricos. A estrutura de uma RNA está, geralmente, relacionada com sua função. A saída da rede requer a ativação de um ou mais neurônios de saídas. Estes resultados podem ser interpretados como uma classificação de padrões (heteroassociação) ou como uma versão completa do padrão de entrada (auto-associação), livre de ruído e distorções.

Uma meso-estrutura consiste de uma série de camadas de EPs com interconexões entre camadas diferentes. Existem três tipos de camadas: a camada de entrada, a escondida (ou interna), e a camada de saída.

A camada de entrada, formada por neurônios sensores, às vezes chamada de retina, serve para armazenar o vetor de entrada, i.e., o padrão a ser classificado. O número de EPs da camada de entrada é, desta forma, facilmente obtido pelo número de componentes do vetor de entrada do sistema.

Entre as camadas de entrada e de saída podem existir uma ou mais camadas internas (ou escondidas), assim chamadas por não comunicar-se com o mundo exterior. As camadas internas proporcionam complexidade e não linearidade para a RNA. Infelizmente não há método até o presente momento para determinação, a priori, do número ótimo de camadas internas e do número de EPs em cada camada, para um dado problema. Porém, como o número de conexões e portanto do tempo envolvido com o cálculo da resposta e treinamento aumentam significativamente com o aumento do número dos EPs das camadas internas, a escolha destes parâmetros é importante e influencia diretamente a performance do sistema. Este número deve ser fixado de acordo com um compromisso entre a taxa de reconhecimento e a velocidade requerida de treinamento e execução do sistema. Redes sem camadas internas do tipo Perceptron não conseguem resolver problemas não lineares, o que foi uma das razões do descrédito da área no final da década de 1960 (Minsky e Papert, 1969). Por outro lado se o número de neurônios nas camadas internas é muito grande a rede passa a funcionar como uma memória, perdendo em geral, a capacidade de generalização.

A camada de saída serve como um *buffer* para armazenar as respostas da rede ao padrão inserido na camada de entrada. Para problemas de classificação, como é o propósito deste trabalho, a camada de saída tem número de EPs igual ao número de classes do problema. No sistema desenvolvido foi adicionado um pequeno estágio de interpretação dos resultados da RNA. A finalidade deste estágio é aumentar a confiabilidade das respostas apresentadas na camada de saída, testando-se condições limiares. Para mais detalhes ver capítulo 5.

Outro parâmetro da operação da RNA é o modo de operação: síncrono ou assíncrono. Para RNAs que necessitam funcionar em tempo real ou que possuam tamanho relativamente grande, este parâmetro se torna mais representativo.

Macro-estruturas são tópicos de pesquisa atual. Poderíamos utilizar uma rede especialista para extração de atributos e outra específica para classificação. Dividir tarefas complexas em específicas, utilizando módulos especializados de processamento parece ser uma estratégia utilizada pelos sistemas neurais biológicos (Zeki, 1993). O maior problema por enquanto é a integração das meso-estruturas que são geralmente treinadas separadamente em *off-line* para adquirirem habilidades específicas.

3.5 Características das RNAs

As características das RNAs foram os principais motivos da escolha deste tipo de abordagem para implementação do classificador deste sistema.

RNAs são apropriadas para tarefas perceptivas e memórias associativas. A informação é armazenada em um conjunto de pesos ao invés de um programa. O processo de restauração de dados é mais semelhante ao processamento de sinais do que com o processamento de símbolos. Nas palavras de *Kevin Gurney (1994)*,

“The combining of signals and producing new ones is to be contrasted with the execution of instructions stored in a memory”

RNAs são robustas na presença de ruído e falha de hardware: pequenas mudanças no sinal de entrada ou nos pesos da rede não irão afetar a saída da rede significativamente. Em diversas outras abordagens, o ruído e a degradação de informação é tratada, na teoria, como fenômenos “controláveis”, seguindo um padrão ou distribuição particular, por exemplo, com erro Gaussiano. A perda de elementos processadores ou de conexões destes, i.e., diluição da rede, geralmente não

compromete o resultado, significativamente, em termos globais, do processamento como ocorre em outros métodos, por exemplo, para sistemas especialistas.

RNAs são de fácil inserção na tecnologia atual. Apesar da maioria das implementações de RNAs serem em *softwares*, muitos dispositivos de *hardware* eletrônicos e/ou óticos têm sido propostos para a obtenção de operações em tempo real. RNAs são intrinsecamente paralelas, pois a computação de um dado componente é independente de outro, o que torna sua performance em máquinas paralelas ainda mais apropriada.

O aprendizado adaptativo é uma das chaves do sucesso das RNAs: elas aprendem a partir de exemplos e podem se adaptar e auto-organizar aos padrões de entrada. Não há necessidade de se obter distribuições de probabilidades para todas as classes de padrões, ou determinar a priori regras de agrupamento de classes ou de comportamento para o sistema.

A habilidade de generalização é outra característica importante: é praticamente impossível treinar uma rede com todas as versões deformadas dos padrões. Com capacidade de generalização podemos treinar a rede com apenas algumas poucas instâncias de cada classe de padrões e a rede será capaz de reconhecer corretamente um padrão deformado não visto durante a etapa de treinamento.

3.6 Rede neural de múltiplas camadas

Diversas abordagens de RNAs para reconhecimento de padrões visuais têm sido propostas (ver capítulo 4). Cerca de 90% das aplicações existentes de RNAs (Masters, 1995) é baseada na rede de múltiplas camadas do tipo *feed-forward* (MLFN)³, inspirada nos Perceptrons de múltiplas camadas (MLP) introduzidos por Rosenblatt (1962). Este tipo de rede assemelha-se a um grafo direcionado onde os arcos, unidirecionais, ligam os EPs, os quais estão organizados em camadas.

³Do inglês "MultiLayer Feed-Forward Network".

A falta de uma regra de treinamento para Perceptrons multi-camadas foi um dos principais fatores que levou a pesquisa em RNAs a um estágio de paralisação no final da década de 1960's. O algoritmo BP desenvolvido por Werbos (1974)⁴, redescoberto independentemente por Rumelhart et al. (1986) renovou o interesse da utilização de RNAs para reconhecimento de padrões. BP é atualmente o algoritmo mais popular para o treinamento de redes multi-camadas, apesar de diversos problemas como o tempo extensivo necessário para convergência e a escolha adequada de parâmetros. MLFN são redes em que o fluxo dos sinais é permitido apenas de neurônios da camada (L-1) para neurônios da camada L. Não existem conexões recursivas, laterais ou auto-conexões.

As razões que levaram a escolha desta meso-estrutura neural para ser o classificador do nosso sistema, além das características apresentadas na seção anterior, são:

(1) A base de treinamento disponível: nosso conjunto de padrões para treinamento da RNA é rotulado, i.e., temos acesso fácil aos tipos de classes de todos os objetos que estamos lidando, o que nos permite trabalhar com um algoritmo de treinamento supervisionado.

(2) Velocidade de execução: redes MLFN são extremamente velozes na fase de execução, o que facilita sua implementação em sistemas que necessitam respostas em tempo real, como nas aplicações industriais.

(3) Além disto, MLFN tem sido utilizado com sucesso como classificadores (Zurada, 1992) e MLFN de três camadas podem formar regiões de decisões complexas arbitrárias (Lippmann, 1987).

A figura 9 apresenta a arquitetura básica de uma rede MLFN com N elementos na camada de entrada, uma camada escondida, e M elementos na camada de saída.

⁴ Chamado na época de "feed-back derivatives"

Figura 9: Arquitetura básica de uma rede MLFN

3.7 Treinamento de RNAs do tipo MLFN

Atualmente existem centenas de regras de treinamento para as redes de múltiplas camadas. Para a classe de regras de treinamento supervisionado, o aprendizado é, geralmente, um processo de duas etapas, que utiliza pares de entrada e saída desejada para cada instância dos padrões que queremos ensinar à rede. Inicialmente inicializamos os pesos sinápticos em valores pequenos de distribuição randômica. A primeira etapa é a propagação do sinal da camada de entrada até a saída. Este procedimento computa a resposta R_i para uma instância do padrão P_i , inserido na camada de entrada da rede, utilizando-se a equação (7), reescrita da equação (5), para computar a saída do neurônio i , na camada $k+1$.

$$n^{k+1}(i) = f\left(\sum_{j=1}^{N_k} w^{k+1}(i, j) \cdot n^k(j) + b^{k+1}(i)\right) \quad (7)$$

Para uma RNA de M camadas o sistema de equações em forma matricial é dado por

$$\bar{n}^{k+1} = f\left(\sum W^{k+1} \cdot \bar{n}^k + \bar{b}^{k+1}\right), \quad \text{para } k = 0, 1, \dots, M-1. \quad (8)$$

A segunda etapa modifica o conjunto de pesos sinápticos utilizando uma função de erro baseado na diferença entre a resposta desejada D_i e a computada, R_i . O

aprendizado é feito em geral através da minimização de uma função erro ou energia do sistema, E , por exemplo definida como o quadrado da média das diferenças (LMS) entre o D_i e R_i (equação 9). Atualmente diversas formulações de funções de erros tem sido utilizadas, como o logaritmo da média das diferenças, Kalman-Kwasny, etc. (Masters, 1994). A modificação dos pesos é feita computando-se o gradiente e retropropagando o erro pelas camadas da rede. Geralmente usa-se uma função do tipo sigmoid (equação 6), ou TanH (tangente hiperbólica), como função de ativação dos neurônios, por ter características de fácil diferenciação e ser limitada tanto superior quanto inferiormente. Para um dado vetor de pesos da rede w , o erro LMS é definido como

$$E(w) = \frac{1}{2} \sum_{j=1}^N (R_j(w) - D_j)^2 \quad (9)$$

Na equação (10), $R_j(w)$ e D_j são, respectivamente, o estado obtido e o desejado do j -ésimo EP na camada de saída, e $j = 1, 2, \dots, N$, são os neurônios de saída.

O BP é o método de treinamento mais utilizado atualmente, apesar de ter diversas desvantagens. Alguns de seus méritos são os fatos que BP tem obtido bastante sucesso em aplicações práticas e tem fácil implementação. BP utiliza a informação do gradiente descendente (GD) do erro, provido pelo instrutor do sistema, para adaptar os pesos sinápticos. BP é uma técnica de regressão não linear de propósitos gerais que objetiva a minimização do erro global. Qualquer função multidimensional pode, pelo menos teoricamente, ser sintetizada por uma rede MLFN treinada com BP. Apesar do método GD tornar a fase de treinamento bastante lenta, em alguns casos, na fase de execução a rede *feed-forward* é bastante rápida pois não há iterações ou estados de mínima energia a serem atingidos, como ocorre em outros algoritmos, ex. a rede de Hopfield.

Para o caso do BP, os pesos são adaptados de acordo com uma quantidade

$$\Delta w_{ij}^k(t) = -\varepsilon \frac{\partial E}{\partial w_{ij}} + \alpha \cdot \Delta w_{ij}^k(t-1) - \beta \cdot w_{ij}^k(t-1) \quad (10)$$

onde uma constante positiva ε controla o gradiente descendente, $0 < \alpha < 1$ é o momentum, β é o coeficiente de decaimento (%), e t é o índice da iteração em progresso. Utilizando $0.01 > \beta > 0$ apenas os pesos efetivamente contribuindo para a redução do erro sobrevivem e ainda aumentam a generalização da rede (Pal & Mitra, 1992).

O gradiente do erro pode ser escrito como

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dx_j} \cdot \frac{dx_j}{dw_{ij}} = \frac{\partial E}{\partial y_j} \cdot y_j^k (1 - y_j^k) \cdot y_i^{k-1} \quad (11)$$

Para a camada de saída, temos

$$\frac{\partial E}{\partial y_j} = R_j - D_j \quad (12)$$

Para as camadas internas,

$$\frac{\partial E}{\partial y_j} = \sum_h \frac{\partial E}{\partial y_h} \cdot \frac{dy_h}{dx_h} \cdot \frac{dx_h}{dy_j} = \sum_h \frac{\partial E}{\partial y_h} \cdot \frac{dy_h}{dx_h} w_{hj}^k \quad (13)$$

onde os EPs j e h estão nas camadas k e $k+1$.

Algumas limitações de MLFN treinadas com o BP são: (i) longo tempo requerido para treinamento. Problemas complexos podem necessitar de milhares de ciclos de treinamento; (ii) escolha de parâmetros de projeto. Pode ser difícil escolher um conjunto de parâmetros ótimo (momentos, taxa de aprendizado, número de EPs e camadas internas, etc.). Algoritmos como o QuickProp definem alguns parâmetros automaticamente; (iii) a falta de conhecimento do poder classificador a priori de determinada estrutura; (iv) mínimo local. BP pode convergir, durante o aprendizado,

para um mínimo local e desta forma não minimizar o erro global para a RNA; (v) outro problema dos MLFNs é a plasticidade: o processo BP de treinamento é um mapeamento vetorial dos pesos sinápticos, e a rede uma vez treinada com um conjunto de padrões X, o processo de adicionar mais alguns padrões Y ao sistema implica em retreinamento de todos os padrões W, $W = X + Y$. Neste tipo de rede não há possibilidade de um incremento do número de padrões para uma RNA treinada, como uma rede do tipo *Adaptive Resonance Theory* (ART) descrita por Carpenter et al. (1989).

Muito esforço tem sido feito na tentativa de otimizar redes MLFN e eliminar suas deficiências. Para mais informações veja a seção 3.8. Nas seções seguintes, descreveremos os métodos de treinamento que realmente trataremos no sistema proposto: os algoritmos gradiente conjugado e o Levenberg-Marquardt. Os dois métodos serão ainda combinados com o método *simulated annealing*, formando uma regra de treinamento híbrida, para melhor escolha da mínima energia do sistema.

3.7.1 Método de treinamento “Gradiente Conjugado”

O método de minimização do erro utilizado pelo algoritmo BP, o gradiente descendente, é bastante simples e fácil de se implementar computacionalmente, porém, não é uma técnica boa de minimização. Por outro lado, técnicas poderosas de minimização muitas vezes não são viáveis de se implementar em uma RNA. Por exemplo, o método de Newton requer inversão da matriz Hessiano $n \times n$, para um conjunto de n pesos sinápticos, tendo custo da ordem de n^3 passos cada vez, além da necessidade de computar segundas derivadas. Estes fatores o tornam um método não prático, além dos problemas de instabilidade numérica (Hertz et al., 1991).

O método do gradiente conjugado tem o poder e velocidade de convergência dos métodos de segunda ordem (MSO), porém, não requerem a computação nem a necessidade de armazenamento dos MSO. (Masters, 1995). A principal causa das dificuldades associadas com o gradiente descendente é que quando minimizamos em

uma direção, perdemos parte do progresso nas outras direções em iterações anteriores. Utilizando uma minimização do tipo zig-zag, i.e., minimizar na direção que é conjugada a direção anterior, se chegarmos a um mínimo em uma direção, permaneceremos no mínimo naquela direção. Não há perda dos ganhos em iterações anteriores. (Press et al., 1993), Masters (1995).

Matematicamente, seguindo Hertz et al. (1991), temos

$$\mathbf{W}_{\text{new}} = -\nabla E_{\text{new}} + \beta \cdot \mathbf{W}_{\text{old}} \quad (14)$$

onde os vetores $\mathbf{W}_{\text{old}}$ e $\mathbf{W}_{\text{new}}$ são conjugados e β é dada pela regra de Polak-Ribiere:

$$\beta = \frac{(\nabla E_{\text{new}} - \nabla E_{\text{old}}) \cdot \nabla E_{\text{new}}}{(\nabla E_{\text{old}})^2} \quad (15)$$

que satisfaz a condição

$$\mathbf{W}_{\text{old}} \cdot \mathbf{H} \cdot \mathbf{W}_{\text{new}} = 0 \quad (16)$$

onde $\mathbf{H}$ é o Hessiano (equação 17), ∇E_{new} é o gradiente na busca pelo mínimo, $\mathbf{W}_{\text{new}}$ é a direção em que efetuaremos a busca, e $\mathbf{W}_{\text{old}}$ é a direção da busca na iteração anterior.

$$H_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_N \partial x_1} & \dots & \frac{\partial^2 f(\mathbf{x})}{\partial x_N \partial x_N} \end{bmatrix} \quad (17)$$

Portanto, o método gradiente conjugado consiste de uma sucessão de minimizações na direção dada por (14) e (15). Um resultado notável do método é que ele minimiza funções multidimensionais, procurando minimizar o gradiente

descendente perpendicular a última iteração sem a necessidade de cálculo do Hessiano. (Hertz, 1991), (Press et al. 1993).

No BP o momentum é dado por uma constante. No GC o momentum, dado por (15) é ajustado de forma ótima para cada novo passo. Este é um dos fatores da melhor performance do CG em relação ao BP.

3.7.2 Método de treinamento “Levenberg-Marquardt”

O método Levenberg-Marquardt (LM) combina os métodos de minimização de Newton e o gradiente descendente e utiliza uma aproximação do Hessiano que pode ser obtida facilmente durante a retropropagação do erro (BP). Sendo necessário armazenar esta matriz, o Hessiano, o LM é útil para redes pequenas, que tenham um pequeno número de pesos sinápticos. Além disso, há necessidade de resolver um sistema de equações lineares para cada iteração (Masters, 1995). Em contrapartida, este método geralmente é mais veloz que o CG, e dessa forma, também testaremos sua eficácia para utilização em nosso sistema.

Diferentemente de outros métodos que utilizam a informação do erro para ajustar os pesos, a estratégia utilizada pelo método LM é reduzir o gradiente a um vetor nulo. Os pontos que apresentam gradiente nulo são os pontos de máximo, mínimo, e pontos de sela da função. Desta forma, utilizando uma tática para eliminar a primeira e a terceira possibilidades, se o gradiente em um ponto é nulo, este ponto é o mínimo global da função.

A regra de atualização de pesos é dada por

$$\mathbf{W}(t) = \mathbf{W}(t-1) - (\mathbf{H} + \alpha \mathbf{I}) \cdot \nabla E \quad (18)$$

onde $\mathbf{H}$ é o Hessiano, $\mathbf{I}$ é a matriz identidade, e ∇E pode ser dado por (13-15). Espera-se que $\mathbf{W}(t) < \mathbf{W}(t-1)$. O parâmetro α , o momentum, controla a influência do

gradiente descendente sobre o método de Newton. Para valores altos de α , a contribuição do método de Newton é pequena para a resposta obtida.

3.7.3 Algoritmos híbridos: uso de *simulated annealing* com os métodos anteriores

Diferentemente dos métodos determinísticos, os métodos estocásticos de minimização utilizam de estratégias de busca por mínimos baseados em números aleatórios, procurando obter uma visão mais global do trecho da função que está sendo otimizada. A principal vantagem é a fuga de mínimos locais, i.e., os estados espúrios, o que é um dos principais problemas dos métodos determinísticos como o BP.

Podemos utilizar estratégias híbridas para otimizar a procura por mínimos, como por exemplo, utilizando o método *simulated annealing* (SA), para encontrar regiões de menor energia, e refinar a busca através de um método determinístico como o GC.

O termo “*simulated annealing*” é derivado do termo “*annealing*”, i.e., *recozimento*, utilizado na metalurgia para obtenção de uma liga priorizando uma melhor organização entre os átomos do material. Para a área de redes neurais, SA está ligada a um processo de busca de pontos mínimos, através de números aleatórios, na qual a faixa de busca (temperatura) é gradativamente diminuída. SA é uma técnica geralmente utilizada em conjunto com outro método, como por exemplo o GC. A principal razão deste fato é que SA é muito lenta, sendo assim, não prático para utilização isolada. Neste trabalho seguimos a abordagem descrita em Masters (1995), p. 138: SA é utilizada no início da rotina e após a convergência do algoritmo determinístico onde o propósito é sair de eventual mínimo local.

3.8 Outros tópicos para otimização das redes MLFN

Algumas técnicas e otimizações tem sido sugeridos recentemente para superar algumas das limitações das RNAs MLPs. Hartono e Hashimoto (1993) apresentaram um método de geração de pares {entrada, resposta desejada} durante o aprendizado, que foi chamado de “*active learning*”. O principal objetivo é a diminuição da tarefa de coleta de padrões para treinamento, reduzindo-se, assim, a necessidade de uma grande base de dados para treinamento.

Outros métodos visam o aumento da velocidade de convergência do treinamento (Ergezinger e Thomsem, 1995). Kak (1994) descreve métodos para redução do número de iterações durante o aprendizado para redes *feed-forward* de três camadas com entradas discretas. Os algoritmos “corner classification” foram testados e comparados com versões rápidas do algoritmo BP. Um dos métodos, o CC3, “corner classification” usando correlações randomizado, foi superior ao BP tanto em generalização quanto em tempo, neste último, com otimização na performance em mais de 50 vezes.

A escolha de parâmetros, como o número de neurônios para camadas internas, a priori, para determinado problema tem sido alvo de pesquisas. Khaw et al. (1995) propuseram uma abordagem para a escolha ótima de parâmetros da rede através dos métodos Taguchi.

Em termos de velocidade de processamento, diversos fabricantes tem lançado processadores neurais, como o C.I. 170 NDX da Intel, o qual podemos associar a outros módulos de forma que podemos alocar o número de estruturas requerido pelo problema, e que pode trabalhar de forma dedicada em determinado equipamento ou processo. A maioria dos processadores requerem um computador e um software para treinamento, e quando os pesos da rede atingiram o estado desejado, estes valores (configuração da rede) são copiados para memórias não voláteis internas aos C.I.’s. Além disto, computadores paralelos de uso geral, ou seqüenciais mais rápidos, implementações de RNAs em circuitos e dispositivos opto-eletrônicos VLSI e ULSI

tem conduzido a melhores performances e permitindo aplicações em tempo real (Shima et al., 1994).

Capítulo 4

Reconhecimento de padrões visuais usando RNAs

Diversas abordagens tem sido propostas para obtenção de RPV invariante a transformações geométricas utilizando os modelos de RNAs. Este capítulo trata de uma revisão, abreviada, dessas abordagens. Seria impossível descrever a maioria dos paradigmas existentes. Dessa forma, serão consideradas apenas as mais usadas e não descreveremos cada meso-estrutura em detalhes, o que pode ser visto nas referências. Existem pelo menos quatro classes de abordagens para reconhecimento invariante usando RNAs. A primeira é através da seleção de um conjunto apropriado de atributos invariantes para representar os padrões e assim inserir estas representações invariantes nas RNAs. Segunda, a estrutura da rede pode ser projetada de forma que sua saída seja sempre invariante a certas transformações. Terceira, podemos apresentar para a rede diversas instâncias de cada classe durante o treinamento de forma que a rede extraia características relevantes dos padrões de cada classe e aprende quais transformações são equivalentes. A outra opção é normalizar os padrões em posição, escala e orientação (PSR)¹ de forma que a rede seja treinada e testada usando padrões em representações normalizadas.

4.1 Espaço de Atributos Invariantes (EAI)

Representar padrões por atributos que são invariantes a transformações geométricas e em seguida apresentá-los ao classificador é um método tradicional para reconhecimento de padrões (ver capítulo 2). Atributos são propriedades mensuráveis

¹ do inglês *Position, Scale & Rotation*.

dos objetos a serem reconhecidos na imagem ou em parte dela. Atributos podem ser classificados em pelo menos três categorias: globais, locais ou relacionais (Chin e Dyer, 1986). Exemplos de atributos globais são: centróide, perímetro, área, momentos, etc. Exemplos de atributos locais são: segmentos de retas, segmento de arco com curvatura constante, pontos terminais, i.e., vértices de um objeto, etc. Exemplos de atributos relacionais incluem uma variedade de medidas de distância e orientação relativa entre as subestruturas ou regiões de um objeto (Chin e Dyer, 1986). O projeto de um sistema de reconhecimento visual usando EAI envolve, inicialmente, a escolha de um conjunto proveitoso de atributos, o que é na maioria das vezes *ad hoc*. Por exemplo, para uma aplicação de reconhecimento de dígitos manuscritos usando invariantes de momentos obtiveram performance diferente quando usados para outra base de imagens, e.g., imagens de aviões (Belkasim et al., 1991).

Idealmente um atributo invariante teria, de fato, valor constante, independentemente do grau de transformação sofrido pela imagem do objeto. Porém, na prática, diversas fontes de incerteza são acrescentadas ao modelo ideal, resultando em uma variabilidade dos valores medidos. Uma referência para medir o quão estável ou instável é o atributo que estamos trabalhando é o coeficiente de variação, definido pela divisão do desvio padrão pela média da amostra. Observando este parâmetro e a tolerância aceitável do sistema para determinada característica, poderíamos utilizar, para um dado problema, até determinados atributos que em algumas aplicações seriam inviáveis, como por exemplo a área de um objeto. Portanto, a escolha dos tipos de atributos a serem utilizados pelo sistema de reconhecimento é, na maioria dos casos, dependente dos tipos de objetos ou padrões que desejamos reconhecer. A escolha deve ser efetuada baseando-se nos critérios apresentados no capítulo 2.

Diversos tipos de atributos e arquiteturas de RNAs tem sido propostas na literatura para solucionar o problema de RPV invariante a transformações geométricas. A seguir vemos uma pequena revisão sobre alguns tipos de momentos e invariantes de momentos, que são os tipos de atributos escolhidos para este trabalho. Em seguida outros tipos de atributos são comentados.

4.1.1 Momentos e Invariantes de Momentos

Hu (1961, 1962) propôs a utilização de invariantes de momentos (IM) para representar imagens de caracteres e obter reconhecimento invariante de padrões visuais. Desde sua publicação, IMs tem sido bastante aplicados em reconhecimento de objetos e classificação de imagens (Teague, 1980), (Khotanzad e Lu, 1990), Sugisaka e Teshnehlab (1993), Wong et al. (1995). Podemos utilizar momentos para caracterizar uma imagem, binária ou em níveis de cinza, considerando esta como uma função de distribuição bidimensional de densidade. Neste trabalho o termo momento se referirá aos momentos regulares.

Momentos descrevem quantidades numéricas em alguma distância de um ponto de referência ou eixo Prokop e Reeves (1992). Momentos são bastante utilizados em estatística e em mecânica, nesta última para caracterizar corpos a partir de sua distribuição de massa. O teorema da unicidade (Hu, 1962) descreve que os momentos de todas as ordens existem se $f(x, y)$ possui valores não nulos em apenas uma região do plano cartesiano (x, y) . O conjunto dos momentos $\{m_{p,q}\}$ são determinados de forma única por $f(x, y)$, e alternativamente, $f(x, y)$ é determinada de forma única por $\{m_{p,q}\}$. Para representar toda a informação contida em uma região de uma imagem seria necessário um número infinito de valores de momento. Como esta possibilidade não é prática, um dos objetivos no projeto de um sistema de reconhecimento que utilize momentos é a determinação da ordem n na qual os momentos contenham informações suficientes para caracterizar as imagens de forma única e atendendo os critérios de velocidade e poder discriminante dos padrões a serem classificados.

A definição de momentos geométricos regulares tem a forma de uma projeção da função $f(x, y)$, que representa a imagem, em uma função monomial do tipo $x^p y^q$. O momento $(p+q)$ é definido de acordo com (19)

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \cdot f(x, y) \cdot dx dy \quad (19)$$

$(p, q = 0, 1, 2, \dots)$

Para utilização em computadores digitais, o momento bidimensional para uma imagem discretizada ($N \times M$), $f(x, y)$ é

$$m_{p,q} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q \cdot f(x, y) \quad (20)$$

($p, q = 0, 1, 2, \dots$)

Para trabalhar com imagens de tamanhos diferentes, podemos normalizar as coordenadas x e y entre os valores $-1 \leq x, y \leq 1$.

Estes momentos não são invariantes a qualquer distorção. Podemos normalizar (20) de forma que possamos obter momentos invariantes a translação e a mudanças de escala, de acordo com

$$\mu_{p,q} = \frac{1}{m_{0,0}^{1+(p+q)/2}} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q \cdot f\left(x + \frac{m_{1,0}}{m_{0,0}}, y + \frac{m_{0,1}}{m_{0,0}}\right) \quad (21)$$

($p, q = 0, 1, 2, \dots$)

onde $m_{0,0}$ representa a área da região da imagem,

$$m_{0,0} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} f(x, y) \quad (22)$$

($p, q = 0$)

As quantidades $(m_{1,0} / m_{0,0})$ e $(m_{0,1} / m_{0,0})$ representam as coordenadas do centróide, respectivamente x_c e y_c , do objeto na imagem. O conjunto completo de momentos de ordem n consistem em todos os momentos, $m_{p,q}$, com $p+q \leq n$, i.e., de $\frac{1}{2}(n+1)(n+2)$ elementos.

Pela combinação dos momentos de segunda e terceira-ordem, podemos obter atributos que são invariantes também à rotação dos padrões na imagem (Khotanzad e Lu, 1990):

$$\phi_1 = \mu_{2,0} + \mu_{0,2} \quad (23-a)$$

$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \quad (23-b)$$

$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \quad (23-c)$$

$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} - \mu_{0,3})^2 \quad (23-d)$$

$$\begin{aligned} \phi_5 = & (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2}) \cdot [(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2] + \\ & + (3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3}) \cdot [3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] \end{aligned} \quad (23-e)$$

$$\phi_6 = (\mu_{2,0} - \mu_{0,2})[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2] + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}) \quad (23-f)$$

As funções acima são alguns dos invariantes de momentos de Hu, os quais provêem discriminação suficiente para os tipos de objetos (classes) que estamos trabalhando. Os valores numéricos de ϕ_1 a ϕ_6 são muito pequenos. Khotanzad e Lu (1990) propuseram o uso do logaritmo dos módulos dessas funções, i.e., $\log |\phi_i|$, $i=1, \dots, 6$, como atributos para representar as imagens.

Recentemente, Wong et al. (1995) apresentaram uma nova forma de derivar automaticamente IMs de uma dada ordem baseando-se na teoria dos números para encontrar todas as possíveis combinações dos IMs. Diversos novos IMs foram derivados que ainda não tinham sido reportados na literatura. Além disso, fatores de ponderação para melhorar o poder discriminante dos atributos foram estudados e definidos. Taxas de reconhecimento de até 98.8 % foram relatadas, para uma aplicação de OCR.

Existem alguns problemas do uso de momentos para representar imagens. Primeiro, momentos são bastante sensíveis a ruído presente na imagem, o que ocorre freqüentemente em ambientes industriais (ver capítulo 1). Segundo, pelo fato da função base utilizada, o produto monomial $x^p y^q$ não ser ortogonal, existe um certo grau de redundância entre os termos dos momentos. A reconstrução da imagem (importante numa aplicação para a determinação do número de momentos a serem utilizados) a

partir de momentos de Hu é uma tarefa bastante complicada, em certos casos impossível, sendo mais fácil se utilizamos funções bases ortogonais (Teague, 1980). Terceiro, momentos são atributos globais, e se calcularmos os momentos sobre uma imagem contendo mais de um padrão o resultado pode ser interpretado como valores da cena como um todo, e não como momentos de um dos padrões. Um problema descrito em Raveendran et al. (1993) é que os invariantes de momentos regulares não permanecem invariantes quando a imagem apresenta mudança de escala diferente para os eixos x e y . inalmente, o custo computacional envolvido para o cálculo de momentos é, também, alvo de pesquisas tanto na área de *software* quanto em *hardware*. No capítulo 5 discutiremos nosso sistema de reconhecimento de padrões visuais que utiliza invariantes de momentos como atributos e a forma escolhida para contornar alguns desses problemas.

Li (1991) descreve que a combinação de invariantes de momentos (IM) com RNAs permite o reconhecimento invariante à posição, escala e rotação de padrões distorcidos em imagens. O autor descreve que RNAs não são capazes de reconhecimento invariante sem o uso de treinamento excessivo de padrões. Li propôs a combinação dos momentos, que garante correspondência de um para um entre a sequência $m_{p,q}$ a uma distribuição de irradiação na imagem, com RNAs, que possui características como generalização e tolerância a falhas. Neste trabalho foi utilizada a RNA tipo Hopfield discreto, onde os padrões apresentados a rede são IMs calculados a partir do padrão apresentado, no caso instâncias de caracteres em imagens de 8×8 pixels, e posteriormente codificados em números binários para então serem apresentados à rede. O sistema foi capaz de reconhecer padrões rotacionados e com certo grau de ruído. Como o modelo usado é uma memória associativa, após diversas iterações a rede restaurou o padrão correto codificado em binário do padrão apresentado.

Um dos problemas da utilização de IMs é o esforço computacional requerido (ECR). Para otimizar o cálculo de IMs, diversas abordagens tem sido propostas. Li e Shen (1991) propuseram um algoritmo iterativo para reduzir ECR ao primeiro grau de adições mais um pequeno número de multiplicações. Entretanto, para uma imagem binária de um objeto simples conectado com contorno de M pixels este algoritmo

requer $32 \cdot M$ adições para calcular os momentos. u et al. (1993) propuseram o cálculo de IMs através da transformada de Hadamard. Após o estabelecimento de uma relação entre momentos e os coeficientes de uma imagem no espaço transformado de Hadamard, os autores descrevem as operações necessárias para o cálculo dos momentos a partir destes coeficientes. Estas operações reduzem-se a adições e deslocamentos, além de um pequeno número de multiplicações. Para imagens binárias com objetos conectados, o esforço computacional pode ser ainda mais reduzido através da projeção da imagem original em dois vetores unidimensionais e traçando o contorno dos objetos para obtenção dos valores. A complexidade computacional do algoritmo é da ordem de $N \log_2 N$ para uma imagem de $N \cdot N$ pixels. Resultados de simulações indicam uma redução de 50% do tempo necessário ao algoritmo apresentado por Li e Shen (1991), para imagens com $N < 1024$.

Yazdi et al. (1993) apresentou uma arquitetura para implementação VLSI de uma rede neural de múltiplas camadas e usando pesos sinápticos digitais em memórias externas. A arquitetura foi aplicada a implementação de um classificador que usa invariantes de momento calculados a partir dos contornos dos objetos em imagens. O treinamento controlado por um computador digital externo.

4.1.2 Momentos de Zernike

Momentos de Zernike (MZ) são baseados em funções ortogonais tendo vantagem em relação a momentos regulares por terem menos redundância em informação e pela possibilidade de restauração da imagem a partir destes valores. Teague (1980) propôs a utilização de bases ortogonais como o conjunto de polinômios complexos introduzidos por Zernike (1934), que formam um conjunto ortogonal sobre o interior de um círculo unitário. As magnitudes dos momentos de Zernike não mudam após uma mudança de orientação do padrão da imagem, e, dessa forma, são utilizados como atributos invariantes a rotação. Uma outra característica que torna atraente a utilização de MZ é que seus momentos de alta ordem são mais fáceis de serem calculados do que para os momentos regulares. Kothanzad e Lu (1990) compararam

dois conjuntos de atributos derivados de momentos regulares e de momentos de Zernike, comprovando a melhor performance do segundo caso.

O momento complexo de Zernike Z_{nl} é definido como (Prokop e Reeves, 1992)

$$Z_{nl} = \frac{(n+1)}{\pi} \int_0^{2\pi} \int_0^1 V_{nl}(r, \theta) * f(r, \theta) r dr d\theta \quad (24)$$

onde * representa o conjugado complexo e $V_{nl}(r, \theta)$ são os polinômios de Zernike de ordem n , que são definidos por

$$V_{nl}(r, \theta) = R_{nl}(r) e^{il\theta} \quad (25)$$

onde $0 \leq l \leq n$, $n-l$ par, e o polinômio radial é dado por

$$R_{nl}(r) = \sum_{k=l}^n B_{nlk} \cdot r^k \quad (26)$$

Os coeficientes de Zernike para $n-k$ par, B_{nlk} , são

$$B_{nlk} = \frac{(-1)^{(n-k)/2} [(n+k)/2]!}{[(n-k)/2]! [(k+l)/2]! [(k-l)/2]!} \quad (27)$$

Belkasim et al. (1991) apresentaram um estudo detalhado da efetividade de diferentes invariantes de momentos em aplicações de reconhecimento de padrões. Seis tipos de momentos, acrescentados de mais dois derivados de normalizações: invariantes de momento de Hu (HMI), regulares (RMI), de Zernike (ZMI) e pseudo Zernike (PZMI), de Zernike (NZMI) e pseudo Zernike normalizados (NPZMI), de Bamieh (BMI), e de Teague-Zernike (TZMI). Dois conjuntos de dados foram utilizados para testar a performance dos invariantes de momentos: imagens de dígitos manuscritos e figuras de aviões, ambos binários. A técnica de classificação utilizada foi

a *múltiplos membros*, onde cada amostra foi classificada baseada em um método ponderado do K-vizinho mais próximo. A performance com a presença de ruído também foi considerada. Os autores relataram que as rotinas de normalização melhoraram a performance de classificação nos dois conjuntos de dados. O NZMI provou ser o conjunto de atributos mais efetivo.

Apesar de algumas vantagens, MZ são invariantes apenas à rotação, o que implica em uma rotina de pré-processamento que deve normalizar a imagem em posição e escala, antes de serem calculados. Além disso, o cálculo dos MZ é mais complexo do que IMs e para caracterizar padrões há necessidade de um número maior de momentos, o que torna maior a RNA e conseqüentemente implica em maiores tempos de treinamento, execução, uso de memória e outros requerimentos computacionais. Dessa forma, em nosso sistema, em que a velocidade de execução é uma das metas principais, não utilizaremos MZ.

4.1.3 Descritores de Fourier

Descritores de Fourier (D) são métodos de representar padrões a partir do contorno dos objetos e tem sido bastante utilizados em aplicações como reconhecimento de caracteres. A partir do contorno do objeto podemos traçar duas curvas descrevendo as variações do contorno nos eixos x e y: $x(n)$ e $y(n)$, onde n , $n = 0, 1, \dots, N-1$, são os pontos do contorno do objeto.

Segundo Jain (1989), para um contorno fechado temos

$$u(n) \equiv x(n) + jy(n) \tag{28}$$

que é periódico com período N. Representando $u(n)$ através da série discreta de Fourier, temos:

$$\begin{aligned} u(n) &\equiv \frac{1}{N} \sum_{k=0}^{N-1} a(k) \cdot \exp\left(\frac{j2\pi kn}{N}\right), & 0 \leq n \leq N-1 \\ a(k) &\equiv \sum_{n=0}^{N-1} u(n) \cdot \exp\left(\frac{-j2\pi kn}{N}\right), & 0 \leq k \leq N-1 \end{aligned} \quad (29)$$

onde os coeficientes complexos $a(k)$ são os descritores de Fourier do contorno. Os efeitos das transformações geométricas sobre os descritores de Fourier como rotação do objeto são contornáveis. O maior problema ocorre quando temos ruído presente na imagem, o que geralmente deteriora as bordas do objeto e dessa forma altera sensivelmente os valores destes atributos. Esta é uma das principais razões da não utilização deste método no nosso projeto, além do número de atributos serem maior que se utilizarmos IMs.

Dobnikar et al. (1992) estudou o reconhecimento invariante à escala e rotação usando descritores do espectro da transformada de Fourier e um modelo de rede neural combinando propriedades dinâmicas e associativas. Foi pressuposta uma posição standard para obtenção das imagens dos padrões, desprezando-se, assim, invariância translacional. Os autores descreveram que o modelo de RNA obteve boa performance, além do sistema apresentar ainda um potencial capaz de processamento altamente paralelo considerável.

4.1.4 Outros atributos invariantes

Schmitt e Mattioli (1991) utilizaram morfologia matemática para transformar uma imagem planar binária de padrões em curvas que são invariantes a PSR. Classificadores baseados em RNAs e estatísticos foram utilizados para discriminar os atributos.

Maruno et al. (1993) descreveram um sistema de reconhecimento invariante a rotação através da representação dos objetos pela transformação phi-s. Os atributos usados são as séries de ângulos como funções de circunferências obtidos dos objeto

(coeficientes ϕ -s). O sistema de reconhecimento utiliza uma rede de camadas multifuncional com neurônios quantizadores em Chip (NC) e uma camada de neurônios com retroação (self feedback layer). Resultados apontaram taxas de reconhecimento correto de 100% de padrões com ruído, obtidos por uma câmara CCD, a partir de 50 épocas de treinamento.

Lim et al. (1992) desenvolveram um sistema de reconhecimento de faces usando lógica nebulosa (fuzzy) e RNAs. O método de extração de atributos foi através das distâncias entre pontos como olhos, nariz, boca, calculadas sobre os contornos do rosto. Para tornar o sistema invariante a mudanças de escala, foi usada normalização dessas distâncias. Variáveis lingüísticas fuzzy foram então usadas para representar a distância aproximada entre os pontos dos atributos. Os vetores de atributos 'fuzificados' foram treinados por uma RNA e usados para reconhecer imagens de rostos na fase de reconhecimento. Os autores descrevem que todas as faces foram reconhecidas mesmo havendo variações nos padrões. Não foi descrito se problemas de captação da imagem, por exemplo, uma variação do rosto no eixo z (espinha dorsal da pessoa), causaria mudanças das distâncias entre os pontos chaves, olhos, etc. Esta abordagem pode ser aplicada com sucesso se considerada a restrição de que obtenção da imagem de todos os padrões (rostos) tenha o mesmo ângulo do eixo z.

Wang e Cohen (1994) descrevem um sistema de reconhecimento tridimensional de objetos no qual os objetos são identificados a partir dos contornos e usando curvas B-spline e um sistema de captação de imagens stereo.

Lejeune e Sheng (1993) descreveram um sistema neural óptico para RPV, consistindo de um correlator óptico e uma RNA. O correlator foi usado para extração de atributos através de filtros espaciais Fourier-Mellin, o qual gera padrões de saída únicos para cada classe de objetos. A RNA utilizada foi um MLP treinado com BP, de tamanho reduzido devido a grande redução da dimensionalidade do uso de EAI, sendo implementada em micro-computadores. Por outro lado, o sistema apresenta limitações, como a impossibilidade de utilização de mais de um padrão na imagem.

Ghorbel (1994) derivou um conjunto completo de descritores de figuras invariantes a transformações de similaridade, para imagens em níveis de cinza. Esta abordagem é baseada na transformada generalizada de Fourier. Também foi proposto uma nova abordagem definindo a transformada analítica de Fourier-Mellin (A MT) sobre a qual uma descrição completa e invariante foi proposta

Hwang e Tseng (1993) apresentaram um modelo de RNA robusta para resolver o problema de invariância no reconhecimento de oclusão, ruído ou vista parcial de objeto 3-D. Os objetos são representados parametricamente por uma rede neural de reconstrução de superfície (SRNN) treinada pelos pontos de contorno amostrados da instância do objeto. Na etapa de execução, se os pontos amostrados do contorno do objeto distorcido não apresentam correspondência, a representação paramétrica permite a informação propagar de forma retrograda pela SRNN para determinação gradual da melhor transformação de similaridade do objeto distorcido.

Em alguns casos, a extração de atributos foi efetuada após a normalização dos padrões (seção 4.4).

Zhou et al. (1991) apresentaram um modelo de atenção seletiva para reduzir o número de entradas da rede e reconhecer imagens contendo múltiplos objetos distorcidos por qualquer translação ou orientação. A invariância à posição e rotação é conseguida através de representação da imagem por atributos invariantes e codificação, após normalização em posição e orientação standard. Foi utilizada uma rede recorrente e testada no problema TC (ver Rumelhart et. al. 1986).

Barnard e Casasent (1990b) descrevem RPV através de uma abordagem de trajetórias e baseada em restrições. O conceito de trajetória é usado para descrever como os dados variam com distorções. As restrições são empregadas para localizar transformações que reduzem objetos ou vetores a um ponto característico para cada objeto. Para classificação, diversos destes pontos característicos são apresentados à RNA.

Tsang et al. (1992) estudaram o reconhecimento de objetos parcialmente ocluídos. A idéia básica é a detecção de atributos essenciais dos contornos dos objetos a serem identificados. A seleção desses atributos para representação das figuras na etapa de treinamento e na etapa de reconhecimento é o principal problema. O sistema deve reconhecer objetos baseado em um conjunto de atributos incompleto, e ao mesmo tempo invariante a PSR. Neste trabalho os contornos mais externos de um objeto foram transformados no domínio theta-S, e filtros com alta capacidade de rejeição de ruído foram empregados para extrair pontos descontínuos que designam as posições de segmentos de atributos simples pertencendo a família de arcos circulares e terminações. Uma RNA MLP foi utilizada para classificar baseada no vetor de atributos. Os autores descrevem resultados satisfatórios.

4.1.5 Abordagens de RNAs usando EAI

Após a extração de atributos, uma rede classificadora é utilizada para discriminar, baseado nestas medidas, a classe da instância apresentada ao sistema. O tipo de rede mais utilizado tem sido o Perceptron de múltiplas camadas, porém, diversas outras arquiteturas como a RB (radial basis function), PNN (probabilistic neural network), a BNN (boolean neural network), etc., tem sido utilizadas para efetuar esta tarefa.

4.1.5.1 Perceptron de múltiplas camadas

Perceptrons foram introduzidos por Rosenblatt (1962). A falta de uma regra de treinamento para Perceptrons multi-camadas foi um dos principais fatores que levou a pesquisa em RNAs a um estágio de paralisação no final da década de 1960's, após a publicação do livro "Perceptrons" de Minsky & Papert (1969). O algoritmo back-propagation (BP) desenvolvido por Werbos (1974) na forma de "feed-back derivatives" e redescoberto independentemente por Rumelhart et al. (1986) renovou o interesse da utilização de RNAs para reconhecimento de padrões. BP é atualmente o algoritmo mais popular para o treinamento de redes multi-camadas, apesar de possuir diversos problemas inerentes ao ineficiente método de redução de erro utilizado.

Masters (1995) apresenta diversas otimizações em relação aos métodos de redução de erro e algoritmos como o gradiente conjugado, que permitem treinamento mais rápido e eficiente. Como a rede ML N, detalhada no capítulo 3, apresenta características semelhantes aos MLPs, não comentaremos muito a respeito da meso-estrutura neural, detalhando mais nas descrições das implementações reportadas na literatura.

MLPs tem sido utilizado com sucesso como classificadores (Zurada, 1992). Perceptrons de três camadas podem formar regiões de decisões complexas arbitrárias (Lippmann, 1987). A utilização de atributos invariantes em conjunção a MLPs tem sido reportados na literatura (Sugisaka e Teshnehlal, 1993). Diversos tipos de momentos e invariantes de momentos tem sido propostos para representação de padrões e tem provado ser uma abordagem com sucesso (Kothanzad e Lu, 1990).

Rogers et al. (1988) apresentaram duas abordagens utilizando EAI e MLPs para a detecção e classificação de alvos táticos a partir de imagens de radar laser de múltiplas funções. A primeira RNA processou momentos de Zernike, enquanto que a segunda RNA foi treinada com picos de correlação no espaço transformado de ourier-Mellin.

Kothanzad e Lu (1989, 1990) demonstraram que um Perceptron de duas camadas treinados com BP, utilizando invariantes de momentos e momentos de Zernike (MZ) para representar imagens binárias de caracteres manuscritos, foram superiores a três outros classificadores tradicionais: "*nearest neighbor*", "*Bayes*", e "*minimum-mean distance*", especialmente quando ruído estava presente. Além disso, eles descreveram que os atributos baseados nos momentos de Zernike possuem um grande poder de separabilidade entre classes e são mais poderosos que os invariantes de momentos de Hu.

Agui et al. (1991) descreveram um sistema composto de RNA MLP de três camadas para reconhecimento de caracteres Katakana do alfabeto japonês. Dois tipos de momentos foram usados para representar as imagens, os momentos regulares e os momentos de Zernike. Também foi abordado o problema da estrutura da RNA e a

relação entre a taxa de reconhecimento e as bases de dados para padrões similares e diferentes.

Troxel et al. (1988) descrevem o uso de MLP para classificação invariante a TG. Objetos a serem classificados eram dados de tanques e caminhões (alvos táticos) captados com radar laser de profundidade. Uma imagem Doppler segmentada foi usada para mascarar a imagem em alvos candidatos. Cada alvo é comparado com máscaras armazenadas que representam as classes diferentes. A máscara e a imagem foram transformadas na magnitude do espectro da transformada de Fourier com coordenadas polares ($\ln r$, θ). A classificação foi obtida através da correlação de picos da transformada da imagem e de uma máscara. Apesar de afirmar taxa de reconhecimento de cerca de 100%, os autores não descreveram o número de classes e quantidade de padrões usados para cada classe. Em trabalho posterior, Troxel et al. (1992) utilizaram MLP para reconhecimento de objetos com oclusão.

Gupta et al. (1990) descrevem uma abordagem usando MLP treinada com BP para classificação de figuras planas com contorno fechado. Esta abordagem pode ser considerada uma composição de extração de atributos (EA) e invariância por treinamento (IT), ver seção 4.3. EAI porque há uma redução da dimensionalidade da representação a ser apresentada ao classificador: seqüências de contorno dos objetos. IT porque o sistema é gradualmente treinado, inicialmente sem ruído e passo a passo inserindo dados ruidosos para obter uma melhor robustez do sistema.

Kim et al. (1991) descrevem uma aplicação de MLP utilizando um algoritmo de aprendizado BP modificado aplicado a reconhecimento de imagens de aeronaves. As imagens são representadas (espaço de atributos) de valores (L , Φ) invariantes a orientações 3-D e a distorção 2-D. A estrutura ótima da RNA foi estudada para obtenção de uma alta performance do classificador, e foi introduzida uma medida de confiança para o projeto de classificadores baseados em RNAs.

Diversas outras aplicações usando MLP tem sido reportadas. O reconhecimento de peças mecânicas para inspeção foi estudada por McAulay et al. (1990), Ventura e Chen (1993) e Costa e Gonzaga (1995). McAulay et al. (1990)

descreve um sistema flexível para inspeção de peças em uma linha de montagens que aprende a discriminar as peças boas e as defeituosas. O sistema extrai atributos numa região de interesse circular da imagem e as processa através da transformada rápida de Fourier, atingindo invariância pela utilização de coordenadas polares, e usa esta representação como entrada para uma RNA treinada com BP. Duas formas diferentes de iluminação dos objetos foram consideradas: iluminação frontal e a traseira, este último ignorando os efeitos de superfície dos objetos. Os autores descrevem taxa de reconhecimento de 100% para os dois casos de iluminação e para um número limitado de padrões disponível. Ventura e Chen (1993) utilizaram um conjunto de primitivas extraído do contorno dos objetos alimentar o classificador MLP. Assume-se que os padrões sejam planares. O autor descreve que o modelo para representar os dados é invariante a transformações e oferece flexibilidade para implementação em tempo real.

Apesar de extensivamente utilizados, os MLPs treinados com BP apresentam diversas limitações, como o tempo requerido para treinamento, a escolha de parâmetros de projeto, a falta de conhecimento do poder classificador a priori de determinada estrutura, e o mínimo local durante o aprendizado. Recentemente, métodos tem sido sugeridos para superar algumas dessas limitações (ver capítulo 3).

4.1.5.2 Redes Neurais Booleanas

Redes neurais booleanas (RNB) tem sido usadas para reconhecimento de padrões oferecendo algumas vantagens sobre outras abordagens: facilidade de implementação em hardware, simplicidade, e velocidade de treinamento e execução. Carvalho et al. (1993, 1994) descreveram uma rede auto-organizante para extração de atributos para utilização em conjunto com redes neurais do tipo “goal-seeking”. O sistema utiliza aprendizado não-supervisionado e neurônios tipo RAM - “Random Access Memory”. A utilização de atributos em relação ao uso da própria imagem para a representação de padrões aumenta as taxas de reconhecimento. As maiores desvantagens de RNB são a saturação, a performance obtida para tarefas de RP declina rapidamente quando a diversidade de padrões aumenta, implicando na capacidade da generalização, e as entradas devem ser sinais booleanos.

4.1.5.3 Rede de Funções de Base Radial

Uma variante da arquitetura “*feedforward*” bastante utilizada é a rede de unções de Base Radial (*Radial Basis Function*), na qual uma camada escondida com Gaussianas de variâncias fixas são usadas no lugar de sigmóides. De acordo com Vasconcelos (1995), enquanto que a definição de um MLP através de um algoritmo de aprendizagem pode ser vista como um processo de otimização estatístico, o projeto de uma RB é realizada através da perspectiva de um problema de aproximação de funções onde uma superfície num espaço multidimensional (função f) é definida por uma combinação linear de funções de base. Geralmente, estas redes utilizam apenas uma camada de Gaussianas e a camada de saída é estritamente linear. Existem várias estratégias para treinamento para os centróides, funções de base e pesos da camada de saída, sendo em geral mais rápido que BP. A regra de propagação da rede é a aplicação de uma função radial, ex. a distância Euclideana entre o vetor centróide e o vetor apresentado à rede ponderado pelos pesos sinápticos. A este valor é aplicado a função de ativação, uma Gaussiana. Quanto menor a distância entre os vetores, maior será a resposta, pela característica das curvas Gaussianas, que apresentam maior valor na média. A configuração ótima de pesos da camada de saída pode ser determinada analiticamente com uma inversão de matriz.

As desvantagens da RB são a necessidade de memória, relativamente alta, e o tempo de classificação, um pouco superior ao MLP (Vasconcelos, 1995), além da performance de classificação ser inferior ao MLP treinado extensivamente com o BP (Morgan e Boulard, 1995).

4.1.5.4 Outras redes que utilizam EAI

Masters (1995) descreve a rede PNN (*probabilistic neural network*) que é em sua essência uma rede neural para classificação. Apesar de vantagens como aprendizado quase instantâneo e capacidade de deduzir as razões de uma decisão da rede, PNN são bastante lentas para trabalho em tempo real, nos computadores que

utilizamos, e exige considerável volume de memória, o que inviabiliza seu uso no nosso sistema.

Li e Nasrabadi (1990) descreveram um sistema que utiliza informações geométricas dos objetos e diversas redes RCE (restricted Coulomb energy) em cascata para reconhecimento de objetos. Se existe regiões de decisão sobrepostas em uma RNA a próxima RNA a classificação é deixada para a próxima rede. Também descreveram um método de aprendizado *coarse-to-fine* para aumentar a generalização do sistema quando o número de padrões de treinamento é pequeno. Foi descrito taxa de reconhecimento de 100%.

A maioria dos sistemas de RP utilizando RNAs foram implementados em software. Ultimamente, sistemas tem sido implementados em hardware, o que está possibilitando a operação dedicada e em tempo real de equipamentos contendo estas tecnologias. Horan et al. (1990) descreve um classificador de segunda ordem invariante a translação, explorando características como tolerância a falhas e ao ruído. Os autores descrevem um sistema de operação baseado numa arquitetura óptica que extrai atributos particulares para o modelo apropriado para ótica. Implementações em hardware eletrônico, como VLSI, pode ser visto em Shima et al. (1994).

4.2. Invariância pela estrutura

Redes neurais invariantes podem ser projetadas criando conexões entre neurônios que forcem versões transformadas do mesmo objeto a serem classificadas como a mesma classe.

4.2.1 Redes Madaline

Widrow et al. (1988) apresentaram um sistema composto por rede invariante para obtenção de saídas insensíveis a transformações geométricas seguida de uma rede neural Madaline. Invariância translacional foi obtida através da utilização de diversas

estruturas de camadas (slabs) de neurônios ADALINEs (ADaptive LINEar Elements). A regra de treinamento do classificador foi a MRII (Winter e Widrow, 1988). O número de conexões necessário é muito alto para imagens para aplicações do mundo real: N^4 conexões são necessárias para uma imagem contendo N pixels com apenas um objeto presente (Barnard e Casasent, 1991). Esta abordagem requer uma replicação massiva de elementos de hardware, o que torna o custo computacional muito elevado, sendo inviável para imagens de dimensões utilizadas na prática.

Hosokawa et al. (1989) seguiram a mesma estrutura básica do sistema de Widrow et al. (1988), mudando a função de ativação dos neurônios de hard-limiter para sigmoidal e usaram a regra de treinamento BP. Eles afirmaram ter obtido melhor performance que o método anterior. Cruz et al. (1989) descrevem um sistema composto de duas redes seguindo a mesma abordagem de Hosakawa et al. (1989).

Fukumi et al. (1991, 1992) também descrevem a mesma estrutura básica de Widrow et al. (1988), onde o sistema é composto de duas etapas, uma rede estática invariante de muitos slabs seguida de uma rede de múltiplas camadas adaptável. Os autores descreveram a implementação de reconhecimento invariante a rotação aplicado a reconhecimento de moedas.

4.2.2 Redes Neurais de Alta-Ordem (Higher-order neural network - HONN)²

Giles e Maxell (1987) descrevem que as redes neurais de alta ordem (HONN) possuem boas capacidades computacionais de armazenamento e aprendizado, podendo projetar a ordem da estrutura de uma HONN para a ordem ou a estrutura do problema. Dessa forma, HONN pode ser projetado para ser invariante a transformações geométricas.

Spirovskia e Reid (1994) estudaram o reconhecimento invariante de padrões em duas ou três dimensões, como figuras de aeronaves, utilizando HONN. Os autores destacam que as invariâncias são construídas diretamente na arquitetura da HONN e

² do inglês *Higher-Order Neural Networks*

não necessitam serem treinadas. Assim, o treinamento de um sistema, que pode ser o BP, para reconhecimento de objetos 2D requer apenas uma instância do objeto para cada classe e um pequeno número de iterações, não diversas instâncias escalonadas, transladadas ou rotacionadas. Isto é devido aos pesos de alta-ordem de uma HONN que aprendem (capturam) as correlações de alta-ordem dos padrões.

Uma HONN com uma camada apenas pode ser representada pela equação

$$Y_i(x) = f \left[W_0(i) + \sum_j W_1(i, j) \cdot x(j) + \sum_j \sum_k W_2(i, j, k) \cdot x(j) \cdot x(k) + \dots \right] \quad (30)$$

onde $f(\cdot)$ é a função de ativação (ex. sigmoid), W é o vetor de pesos sinápticos e x é o vetor de entradas. Diversos algoritmos podem ser utilizados para treinamento de uma HONN, incluindo o BP.

Stavros e Lisboa (1992) introduziram um método para redução e controle do número de pesos de uma RNA de terceira ordem para RPV. A abordagem produz uma rede com alta taxa de reconhecimento invariante a PSR e mais econômica em termos de espaço de armazenamento e processamento. Os autores descrevem a performance da HONN superior a redes MLP usadas com invariantes de momentos, para uma aplicação de dígitos manuscritos.

Zhou et al. (1991) apresentaram um modelo de atenção seletiva para reduzir o número de entradas da HONN. A invariância a posição foi conseguida através de codificação de restrições apropriadas sobre as conexões dos campos receptivos. Os autores implementaram uma HONN de segunda ordem para reconhecer objetos em qualquer posição e rotação de 90 graus. A rede foi testada no problema TC (ver Rumelhart et. al. 1986).

O problema principal de HONN é o tamanho do campo visual, i.e., a retina, limitado devido à memória necessária para o armazenamento do grande número de interconexões numa rede totalmente conectada. Métodos tem sido propostos para otimizar esta desvantagem.

Spirkovska e Reid (1990a) descrevem estratégias para reduzir o número de interconexões necessárias ao armazenamento de uma rede HONN de terceira ordem. De acordo com os autores, o método mais apropriado foi o de conectividade regional, obtendo-se reduções na ordem de 2000 vezes. Mais recentemente, Spirkovska e Reid (1993, 1994)³ apresentaram uma estratégia envolvendo uma combinação de conectividade parcial (regional) e técnicas de *coarse-coding* para superar a limitação de espaço de memória utilizado e poder assim aumentar o campo visual ao tamanho requerido por problemas práticos de reconhecimento de objetos.

Outro problema descrito por Takano et al. (1994) é a não eficiência das HONNs em relação a invariância à escala ou para outras distorções desconhecidas. Dessa forma, o autor apresentou algumas soluções para otimizar a precisão do reconhecimento de padrões escalonados e distorcidos através da mudança da estrutura da HONN.

4.2.3 Neocognitron

Desenvolvido por K. Fukushima (1980), o “Neocognitron”, derivado do “Cognitron”, é uma RNA inspirada na biologia e tem sido utilizado para tarefas como o reconhecimento de caracteres impressos e manuscritos. O Neocognitron é uma RNA hierárquica de múltiplas camadas, auto-organizante para reconhecimento de padrões visuais (RPV) baseada no modelo do sistema visual de Hubel e Wiesel (1962). Diversas restrições que controlam as redes neurais biológicas tem sido investigadas e introduzidas no projeto dessas RNAs, tornando-as capaz de reconhecer padrões independente de deformações. Seguindo Fukushima (1993), estas restrições incluem conexões locais entre camadas numa rede hierárquica, campos receptivos não uniformes, conexões retroativas, conexões pulando camadas, atenção seletiva e conexões inibitórias. O Neocognitron aprende com uma regra não-supervisionada,

³ Em correspondência eletrônica recebida pelo autor, L. Spirkovska descreve que esta técnica ainda é a que produz melhor performance entre taxa de reconhecimento, tempo de treinamento e espaço em armazenamento de informações. Contato: L. Spirkovska, NASA Ames Research Center. E-mail: spirkov@ptolemy.asc.nasa.gov

reforçando os pesos utilizando aprendizado competitivo. O Neocognitron extrai os atributos dos padrões e classifica-os.

Um sistema de reconhecimento de caracteres alfanuméricos tem sido desenvolvido baseado no modelo Neocognitron (Fukushima, 1992). Os princípios da atenção seletiva podem ser estendidos para diversas aplicações, por exemplo, o reconhecimento e segmentação de caracteres conectados em manuscritos de palavras na língua inglesa ou o reconhecimento de caracteres chineses.

Himes e Iñigo (1992) apresentaram uma abordagem para reconhecimento automático de alvos militares utilizando o Neocognitron. Métodos para a redução do tempo de treinamento e uma nova regra para reforço dos pesos sinápticos foram introduzidos. As imagens foram centralizadas em uma grade log-spiral antes de serem inseridas na rede.

Ting & Chuang (1993) estenderam a habilidade do Neocognitron, através de um algoritmo adaptativo, para reconhecer imagens analógicas (com diversos níveis de cinza). Estudos experimentais com diferentes níveis de tonalidades de cinza mostraram que as regras de compensação foram efetivas.

Chao (1992) introduziu uma rede neural óptica baseada no paradigma do Neocognitron. Um novo aspecto do projeto da arquitetura é a invariância a translações usando correlações ópticas multicanais de Fourier em cada camada de processamento. Processamento multi-camadas foi obtido iterativamente retroagindo a saída do correlator de atributos para a entrada do modulador espacial de luz e atualizando os filtros de Fourier.

Li e Wu (1991) descrevem uma RNA multi-camadas para reconhecimento PSR de imagens baseado no módulo hierárquico proposto por Hubel e Wiesel (1962). O sistema apresenta um estágio de invariância a rotação na área de processamento de imagens, a qual rotaciona a imagem para diferentes orientações em coordenadas polares, seguido do Neocognitron e finalmente um módulo RNA com campos de máscaras modificados (MFNN) (*masking fields*) na área de associação. MFNN permite

competição e cooperação com outros componentes para classificação de padrões sem distorções ou mudança de escala significativa, e encontra a orientação do padrão de entrada. Baseado nesta orientação, o segundo módulo, Neocognitron, é usado para resolver mudanças significativas de escala e outras distorções. O sistema foi aplicado a reconhecimento de alvos e de caracteres manuscritos.

Chao e Stoner (1993) descrevem uma implementação ótica de uma RNA baseada no modelo Neocognitron e correlatores de atributos invariantes de múltiplos canais. O sistema foi aplicado a reconhecimento de alvos militares para sistemas de defesa.

Barnard e Casasent (1990) mostraram a performance do Neocognitron não é intrinsecamente invariante a translações, e determinados parâmetros do modelo devem ser escolhidos apropriadamente para a obtenção de uma invariância translacional aproximada.

Seguindo Maren et al. (1991) as desvantagens do Neocognitron são a necessidade de muitos elementos processadores, e a necessidade de camadas e estruturas complexas. Por outro lado, Costa (1993) descreve como contribuições do Neocognitron a diversidade de células, já que a maioria das arquiteturas de RNAs existentes tem apenas um tipo de neurônios.

4.2.4 Arquitetura de ligações dinâmicas

Von der Malsburg (1988) descreve que RNAs convencionais não são estruturas ideais para reconhecimento de objetos invariante e robusto. Imagens são representadas por grafos topológicos rotulados. Imagens de tamanho 128x128 foram processadas e armazenadas utilizando uma pirâmide de neurônios com campos receptivos na forma de funções de Gabor. Lange e von der Malsburg (1988) apresentam uma RNA com ligações dinâmicas capaz de reconhecimento invariante à posição e escala (com limitações). O padrão distorcido é apresentado à rede, *sheet A*, e copiado para outra rede estática, *sheet B*, onde A e B consistem em uma grade de posições (no caso unidimensional, uma lista de posições). A cópia preenche completamente a *sheet B*,

obtendo-se um tamanho e posição invariante. Esta operação de cópia é baseada em um sistema de conexões dinâmicas entre A e B. Após a apresentação da imagem, acoplamentos aferentes da superfície sensorial (retina) selecionam, para cada posição de A, um subconjunto de atributos específicos. Este subconjunto descreve a qualidade local da imagem. Um mecanismo de segmentação restringe a atividade em A para com o segmento contendo o padrão a ser copiado.

Buhmann et al. (1989) apresentaram uma abordagem de reconhecimento de objetos baseado em um processo de casamento de grafos, aplicado a reconhecimento de faces humanas. Imagens em níveis de cinza são representadas por componentes locais de Gabor de forma hierárquica pela resolução, que são todas versões escaladas e rotacionadas de todas as outras. O processo de casamento de pontos foi formulado como uma tarefa de otimização resolvida pela difusão dos pontos a serem confrontados. O algoritmo é uma implementação simples da arquitetura de ligações dinâmicas para RNAs e a difusão é controlada por um potencial determinado pela similaridade de *fluxo de sinais (jet)* e por restrições que preservam a topologia.

Buhmann et al. (1990) apresentaram uma abordagem de reconhecimento de padrões (rostos) baseado na arquitetura de ligações dinâmicas, que é uma modificação das RNAs tradicionais por um novo tipo de variável para expressar os agrupamentos hierárquicos de neurônios. Esta capacidade foi usada para agrupar detetores de atributos primitivos locais (*wavelets* do tipo Gabor) em detetores de atributos compostos (*jets*) e para preservar os relacionamentos de vizinhança entre os *jets*. Os objetos são representados como grafos de atributos, com *jets* servido de atributos. O reconhecimento é formulado como um casamento de grafos e foi implementado como uma difusão topológica de ligações do objeto, contendo restrições.

4.2.5 Outras abordagens de obtenção de invariância pela estrutura

Buhmann e Schulten (1988) propuseram um sistema neural de duas camadas para o reconhecimento invariante a translação. Conexões sinápticas entre as camadas são mudadas através da regra de Hebb. A primeira rede neural extrai atributos de um padrão apresentado e codifica os relacionamentos dos atributos pela atividade

coincidente de neurônios. A segunda rede funciona como uma memória, tendo diversos padrões armazenados.

Rajapakse e Acharya (1990) propuseram uma RNA hierárquica e auto-organizante capaz de reconhecer e reconstruir contornos binários previamente treinados. As propriedades da rede incluem invariâncias a distorções e ruído, sendo invariância à rotação limitada aos padrões originalmente treinados (não generaliza rotação). Os autores descrevem que a rede possui plasticidade, isto é, pode incorporar padrões de treinamento sem perda de informação previamente armazenada.

4.3 Invariância por treinamento

Utilizando a habilidade de generalização das RNAs, poderíamos treinar a rede com diferentes instâncias do mesmo objeto, i.e., vistas transformadas diferentes do objeto. Se o conjunto de padrões utilizado pelo sistema é grande e representativo o bastante para que a rede seja capaz de discriminar objetos apropriadamente, podemos esperar que a rede generalizará corretamente outras transformações não vistas ainda. Um MLP utilizando BP para obter invariância rotacional usando este método foi investigado por Rumelhart et al. (1986).

Zhang et al. (1991) propuseram um modelo de back-propagation modificado que reduz a complexidade da rede como um todo, em vez da remoção de neurônios internos. A medida da complexidade da RNA foi definida como a entropia do padrão de conectividade. O aprendizado da rede leva em conta a minimização dessa medida assim como o erro de saída. Resultados de simulações mostram que a rede conseguiu generalizar e ter comportamento invariante à orientação após ser treinada com alguns ângulos discretos. Os autores atribuíram esta generalização à mudança de treinamento da rede.

A principal desvantagem deste método é o tamanho do conjunto de padrões necessário para treinamento, i.e., a rede deve ser treinada com muitos aspectos de cada objeto. Uma consequência direta é também o longo tempo de treinamento para a rede

e os contornos complexos das regiões de decisão que devem ser criados pela rede. Se existe um número significativo de classes, a rede poderá não convergir (saturação).

Nas palavras de Barnard e Casasent (1991),

“With the present techniques it would be necessary to retain the net on all transformations for every new object to be recognized”,

o que é uma tarefa não prática.

4.4 Normalização em escala, orientação e posição dos padrões

Utilizando um estágio de pré-processamento para normalização dos padrões em escala, posição e rotação nas etapas de treinamento e teste conduzem a uma performance melhor que a última opção: “Invariância por treinamento”. Diversas formas de normalizações foram relatadas. Utilizando este método, o classificador utilizado não necessita ter propriedades invariantes, recebendo uma entrada uma posição fixa (standard) Chan (1991), Nagao e Grimson (1994).

Yüceer e Oflazer (1993) apresentaram um sistema de classificação híbrido baseado em um pré-processador que executa transformações sobre um padrão de entrada para a obtenção de posicionamento, tamanho e orientação standards para representar a imagem, e uma RNA do tipo MLP. Diversos resultados experimentais são apresentados para conjuntos de dados diferentes: letras do alfabeto americano, letras do alfabeto japonês Katakana, e figuras geométricas. Uma vantagem deste método é que ele pode ser projetado para sistemas de propósitos gerais e possui uma boa tolerância ao ruído.

A transformada de Fourier tem sido utilizada como forma de normalizar em relação à posição do objeto (Rosenfeld, 1981). De acordo com a propriedade de deslocamento da transformada de Fourier, i.e.,

$$f(x - x_0, y - y_0) \leftrightarrow F(u, v) \cdot e^{-j2\pi(ux_0 + vy_0)} \quad (31)$$

um objeto transladado horizontalmente de x_0 e verticalmente de y_0 , em uma imagem $f(x, y)$, não sofrerá variações na magnitude da função transformada. Dessa forma, podemos utilizar $F(u, v)$ como um espaço invariante à posição do objeto em relação ao campo visual. Comprovamos experimentalmente a alta estabilidade dessa técnica através de uma experiência simples: fizemos um objeto transladar sobre a imagem e calculamos a magnitude do espectro das imagens. Os resultados descrevem invariância à translação, o que pode ser visto pela variância desprezível dos valores, o que pode ser explicado pela quantização numérica dos computadores e algoritmos. Apresentamos na figura 10 um objeto, quadrado preto em um fundo de imagem branco (a) e seu respectivo espectro de magnitudes de Fourier (b). A figura 10.c representa a figura 10.b em forma tridimensional.

Figura 10: (a) Objeto quadrado; (b) magnitude do espectro de Fourier e (c) representação tridimensional da figura 10.b

Sullivan e Forman (1989) descrevem uma implementação paralela de algoritmos para a obtenção de invariância em reconhecimento de alvos usando sensores laser CO₂. A invariância translacional foi obtida através da transformada de Fourier. A imagem transformada foi usada como entrada numa rede neural treinada para reconhecer vistas representativas de alvos de interesse. O sistema foi implementado em um processador aritmético geométrico paralelo (GAPP), composto de cerca uma matriz de 83 mil processadores.

Yu et al. (1993) descrevem uma abordagem que usa o módulo do espectro da transformada de Fourier seguida de um conjunto de funções de expansões harmônicas circulares (CHE) para a obtenção de invariância em relação à translação e à rotação. Uma matriz de pesos interconectados foi construída usando o modelo hetero-associação interpadrões (IHA).

Zhang e Chen (1993) descrevem uma RNA de múltiplas camadas com feed-back, chamada rede de memória de correlação associativa (CAMN). A imagem a ser inserida na rede passa por transformações para obtenção de invariâncias. Inicialmente é transformada para o espectro de Fourier, a qual a magnitude dos termos é invariante a translações. Em seguida esta imagem (o módulo do espectro de Fourier) é transformado novamente através do mapeamento complex-log e seguido da transformada de Fourier em uma imagem invariante à escala e rotação. A resposta obtida (associação obtida) realimenta a rede até a convergência da imagem correta.

Outras transformações foram reportadas. Minnix et al. (1992) utilizaram a transformação de Walsh-Haddamard modificada para gerar representações invariantes de uma imagem e após uma Neocognitron modificada para reconhecer os padrões normalizados em posição.

Hartmann et al. (1993) normalizaram os objetos através de uma retina especial e mapeamentos paramétricos, obtendo representações invariantes à posição, escala e orientação, que serviram de entrada a uma rede auto-associativa. A orientação do objeto foi obtida com alta precisão. Um sistema consistindo de um robô com manipulador e câmara foi capaz de reconhecer objetos a partir de posições arbitrárias com suficiente precisão para manipulá-los.

Diversas transformadas podem ser combinadas, por exemplo em cascata, a fim de se obter a característica desejada. Por exemplo, a transformada de Radon transforma uma rotação em um deslocamento no eixo ϕ (translação) (Jain, 1989), (You e Ford, 1994). Dessa forma, poderíamos obter invariância a rotação adicionando

após esta transformada uma outra invariante a translação, como a transformada de Fourier.

Finalizando este tipo de abordagem, temos as desvantagens associadas ao método, que neste caso é que o número de conexões das redes é grande devido ao vetor de entrada da rede (tamanho da imagem), gerando como conseqüências os problemas de tempo de execução e treinamento, memória requerida, etc. Além disso, a normalização envolve às vezes interpolação de dados, o que pode mudar, em certo grau, os padrões a serem inseridos na rede, e o tempo envolvido no cálculo das transformadas, que pode ser elevado.

4.5 Outras abordagens envolvendo RNAs

Mundkur e Desai (1991) descrevem uma implementação de uma macro-estrutura neural (ver capítulo 3) na qual o primeiro estágio é um MLP e em seguida uma RNA Maxnet modificada. O sistema foi aplicado ao reconhecimento automático de alvos. Os autores descrevem que a composição das duas redes proporcionou a invariância a translação.

You et al. (1993) apresentaram uma arquitetura de rede integrada a um sistema de quatro estágios. O primeiro é uma implementação da transformada de Radon, usada para transformar rotações e translações do objeto na imagem em translações nas coordenadas θ e s . O segundo estágio provê atributos invariantes à translação pelo uso de correlações e uma rede *maximum-pick-up*. As saídas deste estágio são usadas tanto para reconhecimento de objetos quanto para estimação de ângulos de rotação. O estágio de reconhecimento emprega uma transformada rápida para invariância a rotação e uma rede MLP para classificação dos objetos. Para estimação do ângulo de rotação diversas máscaras são utilizadas, e o ângulo é determinado pela melhor adaptação (casamento) dessas máscaras com o padrão inserido. O sistema apresenta performance de reconhecimento comparada a outros que utilizam atributos como

momentos de Zernike, e a estimação de ângulo comparada a sistemas utilizando momentos regulares de segunda ordem.

4.5.1 Memórias Associativas

Memórias associativas tem sido mais utilizadas em sistemas de restauração de imagens, porém, alguns trabalhos descrevem seu uso para RPV.

As redes de Hopfield (HN) (Hopfield, 1982) são modelos de RNAs cujos neurônios são conectados lateralmente e são usadas tipicamente para auto-associação dos padrões e para problemas de otimização. HN são redes de camada única de neurônios discretos ou analógicos. As conexões laterais, às vezes recorrentes, podem causar o aparecimento de padrões diferentes em um único ciclo de operação. Cada EP é conectado a todos os outros e os pesos sinápticos são simétricos, i.e., $w_{ij} = w_{ji}$, $w_{ii} = 0$.

Dada uma imagem distorcida, ruidosa, corrompida por algum processo, ou até mesmo uma região da imagem, a dinâmica da rede itera em passos discretos até convergir para um estado (padrão) pré-armazenado estável, o que é interpretado fisicamente como um mínimo de energia. Yu e Li (1990) descreve o uso de invariantes de momentos em códigos binários como entradas da HN para a obtenção de reconhecimento de padrões invariante a PSR. A rede utilizada foi um modelo de Hopfield.

HN pode ser usado como uma ferramenta de otimização. Gee et al. (1991) utilizaram um método de projeção de um sub-espço para comparar uma máscara de n pontos com um padrão desconhecido. O encontro do ponto correto foi formulado como um problema de otimização combinatorial e então mapeado em uma rede de Hopfield modificada. O sistema foi aplicado a reconhecimento de dígitos manuscritos (10 classes). Os autores descrevem resultados que confirmam que o método proporciona invariância a transformações lineares e ordenação de pontos.

Suganthan et al. (1993) apresentaram uma aplicação industrial de HN no reconhecimento de membranas transparentes e flexíveis de circuitos impressos e um sub-gráfico isomórfico de uma linha de padrões sintéticos. Diversas funções de energia foram formuladas e exploradas experimentalmente. A HN descrita possui restrições para obtenção de invariâncias. O sistema foi capaz de reconhecer padrões parcialmente ocluídos e a também possui estrutura para implementação altamente paralela.

As limitações com HN são a baixa capacidade de armazenamento (a capacidade máxima tolerável para armazenagem de padrões é de cerca de 15% do número total de neurônios), e a instabilidade do sistema se um padrão compartilha muitos bits com outro padrão.

4.5.2 Teoria da Ressonância Adaptativa (ART)

As RNAs Adaptive resonance theory (ART) são modelos inspirados na biologia. Estas redes implementam processamento cooperativo e competitivo e permitem a realização de processos complexos.

ARTs formam agrupamentos e são treinadas sem supervisão (self-organizing). O algoritmo de agrupamento seleciona a primeira entrada como exemplar para o primeiro agrupamento. A próxima entrada é comparada com o primeiro exemplo do agrupamento. Se a distância para o primeiro agrupamento é menor que um limiar esta entrada é agrupada ao primeiro *agrupamento (cluster)*. Caso contrário, um novo agrupamento é criado. Resultados são computados usando conexões *feed-forward* e o máximo valor é acentuado utilizando inibição lateral entre os outros EP de saída. No ART1 apenas padrões binários são permitidos. ART2 é similar ao ART1 com a possibilidade de usar padrões analógicos. ART3 apresenta princípios organizacionais que permitem o projeto de camadas hierárquicas.

Rak e kolodzy (1988) demonstraram classificação robusta entre objetos apresentando contornos semelhantes, utilizando pré-processamento clássico e ART. Durante o treinamento seqüências de oito classes, inicialmente sem ruído e posteriormente com ruído de 10 a 60%, foram apresentadas ao sistema. Classificação

correta foi obtida até ruído de 50%. O sistema também foi capaz de reconhecer corretamente padrões com até 20% de oclusão.

Srinivasa e Jouaneh (1992) demonstraram que o ART1 possui plasticidade para entradas não-familiares e não requer treinamento explícito das entradas. Rajapakse e Acharya (1990) descrevem uma arquitetura ART multi-camadas auto-organizante altamente estável e plástica capaz de reconhecimento, reconstrução, e segmentação de traços de padrões binários previamente aprendidos.

4.6 Situação Brasileira

No Brasil a pesquisa em sistemas de visão computacional utilizando RNAs se encontra ainda no início, tendo poucos pesquisadores e nenhum jornal ou veículo de comunicação permanente. O ano de 1994 foi importante para a área de RNAs no Brasil pela realização de dois eventos nacionais, o “I Simpósio Brasileiro de Redes Neurais” (Caxambú-MG) e o “I Congresso Brasileiro de Redes Neurais” (Itajubá-MG). Outros eventos também abordaram o tema de reconhecimento de padrões em imagens e redes neurais artificiais como o “Workshop em Visão Cibernética” (agosto/1994 - São Carlos-SP), e o SIBGRAPI’94. Os três primeiros encontros são praticamente toda a fonte de publicações nacionais a respeito de reconhecimento de padrões visuais (e visão computacional) e redes neurais artificiais.

Em relação ao I SBRN e ao I CBRN, o espectro de aplicações foi muito abrangente, desde ambientes de desenvolvimento a sistemas elétricos de potência, passando por processamento de sinais, visão computacional, algoritmos genéticos e lógica fuzzy. Alguns trabalhos focalizaram reconhecimento de objetos e de símbolos (caracteres, dígitos). A maioria dos trabalhos em que houve implementações o tipo de rede neural escolhido foi o MLP utilizando o algoritmo de treinamento *back-propagation*, provando ser o mais popular. Em escala de utilização bem menor apareceram as redes de Kohonen, as Booleanas e uma implementação da

counterpropagation. Alguns artigos dedicaram grande espaço a explicar o funcionamento das redes, coisa que a literatura está repleta e não justifica o grande detalhamento em congressos para público especializado. Particularmente acho que as implementações que fazem uso de redes com características bastante conhecidas como a MLP e algoritmos como o BP deveriam ser dada ênfase a novas formas de extração de atributos, otimização do aprendizado, e outras características da arquitetura do sistema.

Gomes e Carvalho Filho (1994) apresentaram um estudo comparativo entre o paradigma da rede neural MLP e da técnica de momentum no processo de reconhecimento de assinaturas em *off-line*. O reconhecimento de assinaturas em *off-line* é difícil pois as características mais importantes muitas vezes são as dinâmicas, como o tempo gasto para assinar, a pressão exercida da caneta sobre o papel, etc., que não são disponíveis nas imagens estáticas. Os autores utilizaram momentos para descrever as assinaturas, armazenadas em um banco de imagens. Em três configurações de experimentos, os autores descrevem melhores resultados na combinação de momentum e MLP, sendo as outras combinações momentum e classificador mínima distância, e invariância por treinamento direta das assinaturas.

Flores et al. (1994) descreve o reconhecimento de dígitos manuscritos através de redes neurais. A técnica utilizada para obtenção de invariância foi a normalização dos padrões seguindo a técnica de Yüceer e Oflazer (1993). O classificador utilizado foi o MLP utilizando o algoritmo back-propagation. Resultados são comentados, como a diminuição da taxa de reconhecimento para um número crescente de tipos diferentes de letras.

Varella et al. (1994) utilizou a extração de invariantes de momentos seguido de uma rede MLP-backpropagation para o reconhecimento de caracteres. Os autores inicialmente utilizaram apenas uma RNA. O número foi aumentado para trinta e seis (alfabeto americano + dígitos), sendo que cada rede tornou-se especialista em cada padrão. Em terceiro experimento, o número de redes aumentou para 466, 36 padrões com inclinações variando de 15 em 15 graus. Estas últimas apresentam 6 entradas,

quatro células na camada intermediária e duas saídas. Os autores descrevem resultados de aproximadamente 97% para esta última configuração.

Sepeda Filho & Stemmer (1994) apresentaram um sistema de reconhecimento de peças utilizando RNAs do tipo counterpropagation. O tipo de atributo utilizado é um histograma de intensidades que foram obtidos para os eixos x e y. Apesar dos autores descreverem taxa de reconhecimento entre 72% e 100%, para 13 classes de peças, a técnica de extração de atributos é deficiente pois não é insensível a translações, rotações e variações de intensidades luminosas.

Carvalho (1995) descreve o modelo integrado para reconhecimento óptico de caracteres utilizando o SOFT, que é um extrator de características auto-organizante projetado para trabalhar em conjunto com uma rede neural classificadora Booleana do tipo GSN. O artigo é similar ao anterior (Carvalho et al., 1994).

Marar e Carvalho Filho (1995) discutiram sobre classificação de assinaturas *off-line* através da extração de características por Wavelets. O tipo de RNA empregado foi o MLP e experimentos foram realizados sobre os mesmos dados de Gomes e Carvalho Filho (1994).

Tronco et al. (1995) descrevem o desenvolvimento de um sistema de visão para um robô de montagem baseada na RNA do tipo GSN. O sistema irá trabalhar com imagens de peças mecânicas, porém não foram descritos atributos específicos para facilitar o reconhecimento. Além disso, não foram apresentados resultados, limitando-se, os autores, a descrever uma proposta de projeto.

Perelmuter et al. (1995) descrevem um sistema para reconhecimento de imagens bidimensionais utilizando MLPs e coeficientes da transformada de Mellin. A RNA necessitou de 16k neurônios de entrada, 100 na camada escondida, e cinco na camada de saída, o que corresponde a um total de 1638900 sinapses, para reconhecer cinco tipos de objetos, que são bastante diferentes: moeda, disquete, mão humana, chave e caneta. Os autores não descrevem os experimentos de forma estatística, tendo apresentado tabelas de ativações máximas dos neurônios para os objetos transladados

e/ou rotacionados. Vale salientar que só utilizaram três imagens de cada tipo de objeto para testes e concluíram que o sistema é "... capaz de ser aplicado a virtualmente qualquer conjunto de objetos bidimensionais ...". Nenhuma referência quanto a utilização de outro método, como o k-vizinho mais próximo foi mencionado.

Capítulo 5

Sistema de reconhecimento automático de padrões visuais

Este capítulo é dedicado à descrição da forma geral do sistema implementado para contornar os problemas de invariância a transformações geométricas e ruído discutidos nos capítulos anteriores. Este sistema emprega técnicas convencionais de processamento de imagem e utiliza um classificador baseado em redes neurais artificiais.

5.1 Recursos utilizados

Dois tipos de imagens estão sendo utilizadas para testar o sistema: imagens reais e imagens sintéticas. A utilização de imagens reais de peças mecânicas é importante por considerarmos diversas distorções na imagem, como sombras e outros problemas de aquisição da imagem. Por outro lado, a utilização de imagens sintéticas permite-nos controlar alguns parâmetros como a taxa de ruído presente na imagem. Dessa forma podemos analisar o comportamento do sistema quando variamos um determinado parâmetro.

O hardware básico é apresentado na figura 11. As imagens são captadas por uma câmara CCD que foi colocada acima de uma esteira, por onde deslizam objetos. Estas imagens são transferidas para a memória de um computador através de uma placa I/O conversora A/D. O computador utilizado foi um AT-486 DX2. A escolha pelo tipo de computador deve-se ao fato de ser bastante utilizado em nosso país, de

baixo custo e de fácil utilização. Em geral, o hardware utilizado é de custo relativamente baixo, facilitando a implantação até mesmo em sistemas de manufatura simples. O sistema é capaz de reconhecer objetos em imagens compostas de múltiplos padrões de classes diferentes e é tolerante a taxas de ruído na imagem (ver figura 18). Em um ambiente industrial diversas fontes de ruído podem contribuir para uma degradação da imagem, o que diminui sensivelmente a precisão dos algoritmos de classificação e reconhecimento de objetos. Podem ser acrescentados ao hardware do sistema um braço mecânico que efetue manipulações sobre os objetos através do comando do computador.

Figura 11: Hardware básico do sistema

O software foi desenvolvido em linguagem C e C++, utilizando-se o compilador Borland C++ 3.1. Dentre as razões da escolha para esta linguagem estão a portabilidade, facilidade de acesso a recursos do hardware, modularidade, facilidade para manutenção e desenvolvimento de sistemas, suporte a programação orientada a objetos, entre outras características desejáveis. As imagens tratadas possuem dimensão de 128 por 128 elementos de imagem, com 256 níveis na escala de cinza. As imagens sintéticas foram geradas a partir de imagens padrões das classes e foram rotacionadas, transladadas e escalonadas de forma que todas as imagens utilizadas no banco de treinamento e testes são diferentes em pelo menos uma transformação geométrica. Além disto, pelo fato de estarmos trabalhando no espaço discreto de coordenadas, os algoritmos de rotação e escalonamento não preservam, fielmente, na maioria dos casos, a forma original do objeto. Para o caso da rotação existe exceções quando a mesma é feita com ângulos múltiplos de 90 graus. Portanto, o sistema deve ser também robusto para tolerar uma distorção associada a estas operações, que é mais

evidenciada nos contornos do objeto transformado. Na seção seguinte descrevemos a estrutura do sistema, seus blocos funcionais, a interação entre os blocos e posteriormente resultados.

5.2 Forma geral do sistema

Algumas premissas básicas fundamentaram o desenvolvimento do sistema. Inicialmente, o sistema deverá ser capaz de eliminar ruído sem degradar significativamente as bordas e/ou regiões das imagens correspondentes aos objetos. Segundo, o sistema deve reconhecer padrões de forma invariante a transformações geométricas do objeto no campo visual. Esta capacidade é importante por motivos simples, por exemplo, a distância da câmara para a esteira pode variar, conseqüentemente variando o tamanho dos objetos na imagem. Além disto, os objetos são livres, ou seja, podem apresentar orientações e posições diferentes daquelas em que foram utilizadas para treinar o módulo de reconhecimento (MR). O sistema deve ser robusto ainda a outras distorções não previstas no modelo, como distorções geométricas não lineares devido a problemas ópticos de lentes das câmaras, etc., e ruído residual na imagem após o pré-processamento. Para assegurar esta robustez, será utilizado um classificador baseado em redes neurais artificiais do tipo MLFN treinado com o algoritmo gradiente conjugado, e que apresenta boa capacidade de generalização.

Durante a fase de treinamento, apresentamos instâncias de objetos isolados ao sistema. Após a extração de atributos o classificador recebe, além destes valores, a classe do objeto. Durante a fase de execução, o sistema pré-processa as imagens antes de calcular os atributos. A figura 12 mostra o diagrama geral do sistema. Seguindo um procedimento que deve ser aplicado anteriormente ao cálculo dos invariantes de momentos, podemos reconhecer imagens que contenham múltiplos objetos como a figura 18.

Figura 12: diagrama do sistema de reconhecimento de objetos

5.2.1 Pré-processamento

Dada uma imagem $f(x, y)$ de dimensão $N \times M$ e contendo G níveis de cinza¹, o primeiro estágio do sistema, o pré-processamento, realiza quatro procedimentos principais: binarização da imagem, filtragem, codificação de regiões conectadas e eliminação do ruído residual. O pré-processamento é uma etapa de grande importância e em cada etapa devem ser tomadas as devidas precauções para minimizar o erro que é passado em cascata para os próximos estágios.

5.2.1.1 Binarização

A binarização é um processo de segmentação dos objetos contidos na imagem e é baseada em uma mudança dos níveis de cinza de acordo com uma função na qual seu conjunto imagem possui apenas dois valores possíveis. É uma das técnicas mais simples de segmentar uma imagem, e pode ser feita baseada em informações globais (ex. o histograma da imagem) ou pode ser efetuada utilizando-se informações locais (ex. matriz de co-ocorrência). (Pal e Pal, 1993).

Um dos fatores que afetam diretamente na precisão dos atributos extraídos dos objetos da imagem é a qualidade da segmentação. Em recente publicação Zhang (1995) descreve a variação de um conjunto de atributos em relação a fatores, como influência do ruído, do tamanho e da forma dos objetos, para diferentes valores de limiares, resultando em curvas com comportamentos diferentes para diferentes atributos.

A binarização separa a informação (objetos, padrões) do resto da imagem. Após esta operação, teremos uma imagem que possui apenas dois níveis de cinza, o "0", preto, que indica a presença de objetos, e o "G-1", que indica o fundo da imagem (*background*), a qual não nos interessa.

¹ Escala de 0 a G-1.

A operação pode ser descrita pelo algoritmo abaixo:

Parâmetro: Valor_limiar

```
Para i = 1 ... N
  Para j = 1 ... M
    Se  $imagem[i][j] < Valor\_limiar$ 
       $imagem[i][j] = 0$ ;
    Senão
       $imagem[i][j] = G-1$ ;
  Fim-para
Fim-para
```

Podemos utilizar diversas formas na obtenção do valor do parâmetro que serve para o limiar, como o método de vale do histograma, ou utilizar alguma heurística na escolha deste parâmetro. No nosso caso utilizamos um valor pré-determinado que foi obtido a partir da experiência nas diversas classes que estamos trabalhando.

5.2.1.2 Filtragem de ruído

Para testar a influência do ruído sobre os resultados dos valores dos atributos e da classificação dos objetos, foi criada uma rotina com o objetivo de degradar uma dada imagem com uma quantidade desejada de pixels. Variando-se a quantidade de pixels de ruído sobre uma imagem padrão podemos analisar diversos comportamentos e, assim, tentar obter esquemas que melhorem a qualidade da classificação final.

A geração do ruído sobre a imagem sintética pode ser visto como a passagem da imagem por um canal binário e simétrico com probabilidade de erro de $X\%$. A rotina de geração de ruído é apresentada na figura 13. O parâmetro *ruído_desejado* controla a quantidade percentual de inversões de pixels na imagem. Para cada pixel (i, j) da imagem é gerado um valor aleatório entre zero e 100. Caso este valor esteja abaixo do parâmetro *ruído_desejado* o pixel é invertido. Caso contrário nada acontece. Na média teremos um número de inversões de pixels igual a *ruído_desejado* %. Para o caso desse parâmetro ser igual a 100% teremos uma total inversão dos pixels da imagem.


```
float ruído_desejado;  
int i,j;  
|  
for(i=0; i < Nx; i++)  
    for(j=0; j < Ny; j++)  
        if( rand()%100 < ruído_desejado )  
            inverter_pixel(i, j, imagem);  
|
```

Figura 13: código utilizado para geração de ruído

A figura 14-a ilustra uma imagem sem ruído e a figura 14-b, a mesma imagem corrompida por 15% de ruído. Como a probabilidade de ocorrência é igual para todos os pontos dos eixos X e Y e para os valores da escala de níveis de cinza, é de se esperar que o resultado tenha comportamento semelhante, por exemplo, o número de pontos gerados para uma linha seja igual para todas as linhas da imagem. Na prática vemos que apesar da função que gera os números randômicos ter distribuição uniforme, o resultado nem sempre também é, pois trata-se de eventos discretos. A importância dessa experiência é que poderemos analisar como se comportam (variação das respostas) os diversos estágios do sistema, como os filtros, o extrator de características, o classificador, etc., para situações sempre diferentes, o que ocorre na prática.

Figura 14 - Geração de ruído: (a) imagem original. (b) imagem original corrompida por 15% de ruído.

5.2.1.2.1 Motivação para uso de filtros

Apesar dos IMs serem invariantes à escala, posição e rotação, eles apresentam grande sensibilidade ao ruído. A figura 15-a apresenta um gráfico mostrando a variação percentual dos IMs em relação ao caso sem ruído. Para uma dada imagem de um objeto, IMs foram calculados para uma taxa crescente de ruído. Os atributos A_i , $i = 1, 2, \dots, 6$, são os invariantes apresentados na equação 23. Vemos que os IMs não permanecem invariantes com o aumento de ruído e dessa forma, a imagem deve ser tratada previamente, de forma a aumentar a constância desses atributos. Também vemos que um dos atributos apresenta variação acima de 160% para uma imagem com taxa de ruído de 45%, em relação ao caso sem ruído.

Figura 15-a: variação dos IMs em relação a imagem sem ruído. Caso 1: sem filtragem.

Figura 15-b: variação dos IMs em relação a imagem sem ruído. Caso 1: uma filtragem.

Vemos na figura 15-b que após uma filtragem os atributos comportam-se de forma mais estável que para o caso anterior (sem filtragem) na figura 15-a. Para mais de uma filtragem os atributos calculados são, conseqüentemente, mais próximos dos valores originais, devido a melhor restauração da imagem. Fatores como o custo desta operação serão discutidos mais adiante.

5.2.1.2.2 Esquema utilizado para filtrar a imagem

Para eliminar o ruído, propomos uma conjunção de dois métodos, a utilização de um filtro convencional, como filtragem pela mediana de uma janela de 5 por 5 pixels, e em seguida a eliminação de objetos pela área. Este último é efetuado da seguinte forma: para cada objeto conectado com área, ou momento de inércia, abaixo de um determinado valor limiar Γ , consideramos como ruído, sendo eliminado igualando-se seus códigos ao código do fundo da imagem, e deixando assim, de influenciar no resultado dos objetos remanescentes da operação.

A importância da filtragem por eliminação de áreas pequenas reside no fato de que mesmo se utilizarmos um tipo de filtro diversas vezes sobre a imagem, a partir de um certo número de iterações o resultado permanece constante, e dessa forma, muitos pequenos objetos, derivados do ruído, permanecem influenciando nos valores dos

atributos que serão extraídos dos objetos. Utilizando-se uma heurística, i.e., algum conhecimento sobre que valor de área acima do qual podemos considerar como um provável candidato a ser objeto na imagem, podemos eliminar grande parte do ruído residual das filtragens convencionais. Maiores detalhes estão expostos na seção seguinte.

5.2.1.3 Codificação de regiões conectadas e eliminação residual do ruído

A codificação de regiões conectadas e a eliminação do ruído residual são duas etapas que podem ser efetuadas concomitantemente. A necessidade dessa etapa ocorre principalmente na fase de execução do sistema, pois o mesmo pode receber como entrada uma imagem contendo diversos objetos e portanto deve tratá-los de maneira diferente. Para cada objeto conectado na imagem, o sistema aloca, dinamicamente, uma estrutura, ver figura 16, que o descreverá até o final do procedimento. Inicialmente grande parte dos campos da estrutura alocada para um determinado objeto estão vazios, e são preenchidos no decorrer da execução do sistema.

```
struct objeto {  
    int x_inic, y_inic;  
    float x_centro, y_centro;  
    float area, ..., momentos[N][M];  
    float Invariantes_de_momentos[1][6];  
    :  
    :  
    int classe_do_objeto;  
}
```

Figura 16: definição da estrutura em C para cada objeto conectado que satisfaz as condições.

A rotina de codificação de regiões tem por finalidade marcar regiões conectadas com intensidades luminosas semelhantes, i.e., para cada seqüência de pixels conectados é atribuído um nível de cinza diferente. Um pixel (P) pertencente ao objeto é marcado com o código desejado e a partir dele expande-se a marcação para os outros pixels. Esta técnica também é denominada rotulação ou marcação de regiões conectadas. A figura 17 ilustra, simplificada, os passos na direção descendente ($i, j = 0 \rightarrow N, M$) a partir da semente (P). No procedimento real as duas direções são utilizadas.


```

Parâmetros: x_inic, y_inic, Código_desejado
f(x_inic, y_inic) = Código_desejado
Para x = 0, ... N
  Para y = 0, ... M
    Se f(x, y) = Código_desejado
      para i = x - 1 ... x + 1
        para j = y - 1 ... y + 1
          Se f(x, y) = 0
            f(x, y) = Código_desejado
          fim-se
        j = j + 1
      i = i + 1
    fim-se
  fim-se

```

Figura 17: Pseudo-código para Codificação de região na direção descendente ($i, j = 0 \rightarrow N, M$) a partir da semente $P(x_inic, y_inic)$

O tipo de vizinhança escolhido foi 8 vizinhos. A razão principal foi que utilizando-se apenas 4 vizinhos o resultado da codificação dos objetos foram piores que para 8 vizinhos quando o ruído é elevado.

A eliminação de ruído residual foi efetuada considerando-se que regiões conectadas com área menor que determinado limiar (Γ) foram consideradas ruído, sendo então eliminada. O cálculo da área da região conectada foi efetuado durante sua codificação, de forma que seu cálculo não implica em um aumento sensível de processamento. Regiões com área maior que Γ são considerados objetos e a estas regiões são atribuídos códigos (ver figura 18.c). Esta eliminação de regiões pequenas baseado em um valor limiar pode ser efetuada logo após a codificação da área do objeto, ou após a conclusão da marcação de todas as regiões conectadas da imagem. O primeiro caso tem a vantagem de ocupação menor de memória: objetos são eliminados assim que não satisfazem o critério mínimo de área. Por outro lado, o valor do limiar pode ser automaticamente selecionado por um método como o “método do vale” sobre o histograma de áreas de todas as regiões conectadas.

Como sabemos a princípio um pouco sobre a distribuição de alguns atributos como a área para os tipos de objetos que estamos lidando, utilizamos a primeira

estratégia: o valor limiar da área mínima foi estabelecido antes de iniciar o processamento.

A figura 18 apresenta seqüências de filtragem e codificação para uma imagem livre de ruído (a), imagem com 10 % de ruído (d) e imagem com 25% de ruído (g, j). Uma filtragem do tipo mediana foi efetuada nas imagens 18-a, -d e -g, resultando nas imagens 18-b, -e e -h, respectivamente. Cinco filtrações do tipo mediana foram efetuadas na imagem 18-j, resultando na imagem apresentada na figura 18-l. Vemos que quando o ruído é elevado, a utilização de mais de uma filtragem melhora sensivelmente a qualidade final dos objetos, além de reduzir o tempo de codificação, como veremos mais adiante. O tipo de filtro utilizado foi a mediana em uma janela de 3x3 pixels.

Uma vez eliminado grande parte do ruído, teremos uma imagem como a figura 16-m, onde os objetos são codificados com valores entre 1 e (G-2). Tipicamente utiliza-se $G = 256$, o que faz com que após binarização teremos nível '0' (zero) para os objetos, e '255' (branco) para o fundo da imagem. Todos os valores entre 1 e 254 podem ser utilizados para codificar áreas da imagem (Parker, 1994). Esquemas de codificações mais complexas podem ser adotadas caso o número de objetos seja maior que 254, o que é bastante improvável para o nosso caso, em específico.

Figura 18: (a) imagem original após binarização; (b) resultado da filtragem de (a); (c) resultado da codificação de objetos sobre a imagem em (b); (d) a mesma imagem de (a) porém com 10% de ruído; (e) resultado da filtragem de (d); (f) resultado da codificação de objetos sobre (e). (g) mesma imagem de (a) porém com 25% de ruído; (h) resultado da filtragem de imagem (g); (i) resultado da codificação de objetos sobre (h); (j) mesma imagem de (a) porém com 25% de ruído; (l) resultado do uso de cinco passagens de filtragem de imagem apresentada na figura (j); (m) resultado da codificação de objetos sobre a imagem em (l).

Após a operação de codificação sobre seqüências de pixels 8-conectados da imagem com área acima do limiar Γ , teremos um vetor de K estruturas semelhantes a da figura 16, que servirá para armazenar dados dos K objetos conectados identificados.

5.2.1.3.1 Tempo de codificação de regiões conectadas e eliminação por área

Nesta seção será discutido a questão do tempo envolvido na codificação de regiões conectadas (CRC) e a eliminação de regiões de área menor que o limiar ($ERC_{A<L}$). Utilizando apenas uma filtragem do tipo mediana ou suavização por janela de 5x5 pixels, obtemos os dados plotados na figura (19). Este gráfico mostra o tempo de codificação versus a variação de ruído. Para as experiências foi utilizado um limiar de área mínima igual a 100 pixels.

Figura 19: tempo de codificação versus a variação de ruído para uma filtragem

A filtragem apenas por $ERC_{A<L}$ mostrou-se ineficaz, tanto pelos aspectos de resultados de atributos como pelo tempo necessário de execução. Para uma taxa de 30% de ruído foram necessários mais de 25 segundos para tal filtragem. Ainda mais, os resultados dos atributos calculados foram insatisfatórios, o que demonstra a necessidade de uma filtragem tipo mediana antes da $ERC_{A<L}$.

5.2.1.3.1 Utilização de mais de uma filtragem mediana ou suavização antes da CRC e $ERC_{A<L}$

Para imagens sintéticas contendo uma taxa de ruído elevada podemos utilizar mais de uma filtragem, na tentativa de melhorar ao máximo a imagem, antes da eliminação por área e codificação de regiões. Esta melhora tem um custo relativamente

elevado: o tempo. Dessa forma, temos que encontrar um número de iterações que atinja uma qualidade de imagem admissível levando em consideração que o tempo de processamento deve ser o mínimo possível. Porém a partir de um determinado número de iterações, o resultado não muda sensivelmente, portanto, o número de vezes em que efetuaremos o procedimento de filtragem tem, de certa forma, um limite superior.

Por outro lado, foi constatado experimentalmente para imagem com ruído elevado (por exemplo 40%) que o tempo perdido com uma ou mais filtragens adicionais (por exemplo, mediana) antes da filtragem $ERC_{A<L}$ é menor que o tempo dessa última operação, além da vantagem de que os atributos medidos apresentaram valores mais próximos aos padrões do que apenas uma filtragem.

Para o caso de uma imagem com taxa de ruído igual a 35% e utilizando a filtragem tipo mediana, antes da codificação, temos que a partir de 5 filtragens o resultado dos atributos não varia sensivelmente. Portanto, pode-se utilizar de uma a cinco repetições deste tipo de filtragem antes da CRC. As figuras 20 e 21 ilustram este fato. A figura 20-a é a imagem original de um dos objetos com as condições descritas no início do parágrafo. As figuras 20-b, -c, -d, -e, -f, -g e -h são os resultados de 1, 2, 3, 4, 5, 9, e 25 iterações da filtragem do tipo mediana sobre a imagem da figura 20-a.

Figura 20: (a) imagem do objeto com taxa de 35% de ruído.; (b) suavização pela mediana da figura em (a); (c), (d), (e), (f), (g) e (h), suavização pela mediana da figura em (a) por 2, 3, 4, 5, 9 e 25 vezes, respectivamente.

A figura 21, apresenta o resultado da $ERC_{A \leq L}$ e CRC para as figuras 20 (b-h), respectivamente. Vemos na primeira filtragem (figura 21-b) três objetos foram codificados, sendo dois deles gerados pelos pixels provenientes do ruído. Nas três filtrações subseqüentes, dois objetos foram codificados, e o objeto principal apresenta uma melhor definição a cada iteração. A partir da quinta iteração apenas um objeto foi codificado, e os valores de seus IMs permaneceram constantes mesmo elevando-se o número de filtrações até 25 (ver figura 22).

Olhando a figura 21-(b-h) percebe-se facilmente que se tivéssemos utilizado uma característica derivada do contorno do objeto como descritores de Fourier, provavelmente incorreríamos em erros de classificação.

Figura 21: (a) imagem do objeto com 35% de ruído; (b - h) resultado da codificação e eliminação por área das figuras 24 (b-h), respectivamente.

A figura 22 mostra um gráfico comparativo da variação dos atributos em relação ao valor padrão incrementando-se o número de iterações, vemos que após uma filtragem, o atributo A_1 apresenta variação de 41% em relação ao caso sem ruído. Após 5 iterações, esta diferença cai para 19%. Outros atributos como o A_4 e A_6 tem, após, 5 iterações diferença de apenas 7% em relação aos atributos extraídos de uma imagem livre de ruído.

Figura 22: variação percentual dos IMs com o aumento do número de iterações de filtragens (em relação a um padrão livre de ruído).

A escolha do número de filtragens pode ser decidido automaticamente, se olharmos para o espectro da transformada de Fourier. Por exemplo, A figura 23-a apresenta a imagem de um objeto livre de ruído e em 23-b o espectro da transformada de Fourier de 23-a. A figura 23-c ilustra a mesma imagem de (a) corrompida por 15% de ruído. Vemos na figura 23-d, que representa o espectro de 23-c, uma maior concentração em altas frequências em relação a figura 23-b. A figura (e) representa a imagem após filtragem da mediana e (f) seu espectro. Finalmente a figura (g) representa a imagem final após eliminação por área e codificação, e (h) seu espectro. Notar semelhança entre (a) e (g) e entre (b) e (h). Podemos utilizar, por exemplo, uma medida de concentração de pontos acima de determinado valor em uma região de frequências, e assim, enquanto esta concentração for acima que um limiar, deve ser feita nova filtragem.

Figura 23: (a) imagem original; (b) Espectro da FT de (a); (c) imagem de (a) com 15% de ruído; (d) Espectro da FT de (c); (e) Imagem de (c) após filtragem pela mediana. (f) espectro da FT de (e). (g) resultado da CRC. (h) Espectro da FT de (g).

A utilização deste procedimento não é viável para cada objeto em cada imagem que se deseje classificar. Como este sistema é projetado para trabalhar, por exemplo, em conjunto com uma linha de montagem, o operador pode, antes de iniciar a produção, testar em uma imagem padrão as condições locais, e uma vez extraído número de filtragens ideal (N_{FI}), inicia-se a fase normal de reconhecimento utilizando N_{FI} como número de filtragens na etapa de pré-processamento.

5.2.2 Extração de atributos

A escolha das características para representar os padrões é uma tarefa difícil. Os invariantes de momentos regulares foram escolhidos para representar os objetos por apresentarem as características de invariância a escala, posição e orientação. Através da utilização de um esquema para filtrar e codificar os objetos de uma imagem, eliminamos grande parte dos problemas dos IMs.

Uma vez o objeto codificado (ver seção anterior) as rotinas de extração de atributos devem receber como parâmetro este código, desprezando assim áreas

marcadas com outras codificações². Dessa forma, o cálculo de IM para um objeto despreza totalmente os outros previamente marcados com outros códigos. Esta operação pode ser descrita por

$$m_{p,q}^i = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q \cdot f(x,y), \quad (32)$$

$$f(x,y) = i, \quad i = 1, 2, \dots, K$$

(p, q = 0, 1, 2, ...)

Para o caso de imagens com apenas um objeto ($i=1$), a equação (32) se reduz a equação (20).

De posse dos momentos $\{m_{p,q}^i\}$ o sistema calcula os invariantes $\{\phi_n^i, i=1, 2, \dots, 6\}$ (equação 23). Wong et al. (1995) apresentaram uma técnica de derivação de novos invariantes de momentos de uma dada ordem. Abaixo (equação 33) está uma relação desses novos invariantes de momentos de terceira ordem que serão testados nas experiências.

$$\phi_1' = -2.(\mu_{20} + \mu_{02}).(\mu_{12} + \mu_{30}).(\mu_{30} + \mu_{21}) + 2.\mu_{11}.[\mu_{30}^2 - \mu_{21}^2 + \mu_{12}^2 - \mu_{03}^2 + 2.(\mu_{30}.\mu_{12} - \mu_{21}.\mu_{03})] \quad (33.a)$$

$$\phi_2' = (\mu_{20} + \mu_{02}).[\mu_{30}^2 - \mu_{03}^2 + 3.(\mu_{21}^2 - \mu_{12}^2) + 2.(\mu_{21}.\mu_{03} - \mu_{30}.\mu_{12})] + 4.\mu_{11}.[\mu_{30}(\mu_{21} - \mu_{03}) + \mu_{12}(\mu_{03} + 3\mu_{21})] \quad (33.b)$$

$$\phi_3' = 2\mu_{11}*(\mu_{30}^2 - 3\mu_{12}^2 + 3\mu_{21}^2 - \mu_{03}^2) + 4\mu_{11}.(\mu_{30}.\mu_{12} + \mu_{21}.\mu_{03}) + 2.(\mu_{20} + \mu_{02}).[\mu_{30}(\mu_{03} - \mu_{21}) - \mu_{12}(\mu_{03} + 3\mu_{21})] \quad (33.c)$$

$$\phi_4' = [(\mu_{20}^2 + \mu_{02}^2 - 4\mu_{11}^2) + 2\mu_{20}^2].(\mu_{30}^2 + \mu_{03}^2) - (\mu_{12}^2 + \mu_{21}^2).[3.(\mu_{20}^2 + \mu_{02}^2 + 4\mu_{11}^2) + 2.(\mu_{20} + \mu_{02})] + 2(\mu_{30}.\mu_{12} + \mu_{21}.\mu_{03}).[(2.\mu_{11}^2 - \mu_{20}^2) - (\mu_{20}^2 + \mu_{02}^2)] - 16\mu_{11}.(\mu_{20} + \mu_{02}).(\mu_{30}.\mu_{21} + \mu_{03}.\mu_{12}) \quad (33.d)$$

$$\phi_5' = 4\mu_{11}.(\mu_{20} + \mu_{02}).[(\mu_{30}^2 + \mu_{03}^2) - 2.(\mu_{30}.\mu_{12} + \mu_{21}.\mu_{03}) - 3.(\mu_{12}^2 + \mu_{21}^2)] + 4.(\mu_{03}.\mu_{12} - \mu_{30}.\mu_{21}).[(\mu_{20} + \mu_{02})^2 - 4\mu_{11}^2] \quad (33.e)$$

² Se não tivéssemos utilizado o código da região como parâmetro da função e tivéssemos calculado os invariantes de momento para uma imagem contendo mais de um objeto, resultados como as coordenadas do centro de massa encontrados seriam da imagem inteira, e não o de determinado objeto.

$$\begin{aligned}
 \phi_6' = & 3\mu_{30}^2\mu_{20}\mu_{02}^2 + 3.(\mu_{02}\mu_{20}^2 + \mu_{20}\mu_{02}^2).[6.(\mu_{21}\mu_{03}-\mu_{30}\mu_{12}) + 9.(\mu_{12}^2 - \mu_{21}^2) - \\
 & \mu_{03}^2] + 12\mu_{11}^2(\mu_{20}+\mu_{02}).[6.(\mu_{30}\mu_{12}-\mu_{21}\mu_{03}) + 9.(\mu_{21}^2 - \mu_{12}^2) - \mu_{30}^2] + \\
 & 12\mu_{11}.(\mu_{02} + \mu_{20})^2 .[3.(\mu_{30}\mu_{21}+\mu_{12}\mu_{03})+\mu_{03}^2\mu_{11}-\mu_{30}^2-9\mu_{12}\mu_{21}] + \\
 & (\mu_{20}^3 + \mu_{02}^3).[9.(\mu_{12}^2 - \mu_{21}^2) + 6.(\mu_{21}\mu_{03}-\mu_{30}\mu_{12}) + \mu_{30}^2 - \mu_{03}^2] + \\
 & 16.\mu_{11}^3.[3.\mu_{12}.(3.\mu_{12}-\mu_{03})+\mu_{30}.(\mu_{03}-3\mu_{21})]
 \end{aligned} \tag{33.f}$$

Ao final do processamento, teremos uma tabela com descrição da classe do objeto, a posição de seu centro de massa (X_C , Y_C), sua área, a orientação do eixo principal mais próximo ao eixo x (equação 34), etc. Estas informações podem servir tanto para um manipulador, como um braço mecânico, que irá manipular os objetos, ou para outro estágio de nível mais elevado, onde as relações entre os diferentes objetos contidos na imagem proporcionam a interpretação da cena.

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \tag{34}$$

5.2.3 Classificador: RNAs MLFN

A deterioração do(s) objeto(s) contido(s) na imagem devido ao ruído implica diretamente nos valores dos atributos. Dessa forma, o classificador deve ser robusto e capaz de generalizar e tolerar esta margem de ruído sem grandes conseqüências à taxa de reconhecimento. Por estas e outras razões (ver capítulo 3) que escolhemos a abordagem de RNAs para o projeto do classificador.

Para testar a eficiência de RNAs do tipo MLFN utilizaremos dois tipos de algoritmos de treinamento foram utilizados: gradiente conjugado e o Levenberg-Marquardt (LM). Ambos algoritmos apresentam diversas vantagens em relação ao tradicional *back propagation* (Masters, 1995).

Apesar da facilidade da escolha do número de neurônios das camadas de entrada e saída, iguais ao número de atributos e número de classes, respectivamente, a escolha de quantas camadas intermediárias e quantos neurônios para solucionar um dado problema ainda é uma tarefa empírica. Uma abordagem descrita por vários autores na literatura é iniciar a rede com apenas uma camada intermediária e poucos neurônios. Adicionar neurônios às camadas intermediárias e observar o comportamento global da RNA, tempo de treinamento, capacidade de generalização, etc., é uma tarefa cansativa, pois muitos problemas podem levar dezenas, até centenas de horas para convergir, ou podem, até mesmo, estabilizar em um erro e deste ponto não conseguir mais minimizar. A maioria das redes implementadas neste trabalho tem tamanho de seis neurônios na camada de entrada, dez neurônios na camada intermediária, e seis na camada de saída. No caso de imagens reais o número de neurônios na camada de saída foi nove. Em alguns casos a arquitetura da rede foi modificada de modo a atingir o resultado desejado. Nestes casos, i.e., as exceções, os parâmetros, como número de neurônios, são comentados.

Um procedimento de pós-classificação (interpretação dos dados) testa e valida as respostas das RNAs, comparando a ativação dos neurônios de saída com valores mínimos aceitáveis para se considerar uma classe. O método k-vizinho mais próximo será utilizado para compararmos a eficiência das RNAs em relação a classificação dos dados (ver capítulo 6).

5.2.4 Estágio de interpretação de dados

Para validar a resposta da(s) RNAs empregamos o seguinte método: caso a diferença entre a maior resposta da rede e um valor mínimo de ativação seja positiva, esta resposta é aceita como a classe do padrão desconhecido (decisão da RNA). Se a diferença for negativa, i.e., os neurônios da camada de saída não estão suficientemente ativados, e neste caso o padrão é tido como não reconhecido. Esta informação de rejeição pode ser retroagida pelo sistema de forma que seja verificada se há necessidade de se fazer novamente as etapas anteriores à classificação, podendo-se variar certos parâmetros das rotinas, de forma a aumentar a precisão nas etapas de pré-processamento.

Capítulo 6

Resultados e Comentários sobre Classificação de Padrões

6.1 Bancos de imagens

Os bancos de imagens sintéticas e reais contêm, cada um, seis e nove classes de objetos (figura 24), respectivamente. Para cada classe foram utilizadas quinze imagens para treinamento e quinze para testes, totalizando cento e oitenta imagens para o banco de imagens sintéticas e duzentos e setenta para o banco de imagens reais, testados nas duas configurações de classificadores. Todas as imagens de cada classe apresentam tamanho, ângulo de orientação, e posição diferentes.

Exemplos das peças sintéticas e reais utilizadas no trabalho estão mostradas nas figuras 24-a e 24-b, respectivamente.

Figura 24-a: Exemplos dos tipos de objetos utilizados (sintéticas)

Figura 24-b: exemplos de tipos de objetos (imagens reais)

A figura 24-b apresenta imagens de objetos em níveis de cinza (a, b, ... i-1), da forma que foi obtida pela câmara CCD, e suas imagens após binarização (a, b, ... i-2).

6.1.1 Comentários sobre as distribuições numéricas de atributos

No apêndice A está apresentada uma sequência de imagens para testes da fase de filtragem da imagem antes da extração de atributos. No apêndice B diversas curvas relativas a distribuições dos atributos entre as classes são apresentadas. Observa-se que em alguns atributos existe grande sobreposição das classes, o que dificulta bastante o processo de classificação. A principal explicação deste fato é que para objetos semelhantes (por exemplo ver figura 24-b.d e a figura 24-b.i) os atributos apresentam valores bastante próximos. Idealmente os atributos deveriam se comportar com a mínima variância, porém diversos fatores como: (1) a segmentação gera distorções nos objetos por problemas de iluminação como sombras, e outros de aquisição da imagem. Dependendo do valor do limiar pode-se obter uma melhor ou pior representação em binário do objeto original. Estudos da influência do valor de limiar sobre atributos de objetos podem ser vistos em (Zhang, 1995); (2) os objetos metálicos utilizados refletem luz e algumas regiões destes apresentam regiões claras que quando são binarizadas se transformam em níveis de cinza iguais ao do fundo da imagem. Dessa

forma temos uma distorção não linear na imagem dos objetos relativa a reflexão da superfície metálica dos mesmos; (3) discretização numérica das imagens e procedimentos numéricos, entre outros.

6.2 Classificação

Para testar a capacidade de classificação e generalização do sistema dois algoritmos de treinamento, o gradiente conjugado (GC) e o Levenberg-Maquardt (LM) foram testados (ver capítulo 3). Para os dois algoritmos foram testadas as capacidades de generalização para exemplos não vistos na etapa de treinamento e o tempo de convergência para diferentes valores de erro aceitável. Estas características são importantes para comparar os métodos utilizados. Para comparar os resultados de classificação das RNAs foi utilizado o método k-vizinho mais próximo, com $k = 5$.

Todos os experimentos foram realizados em um computador pessoal do tipo AT 486 DX2 66 MHz, e para as imagens sintéticas foi utilizado uma filtragem do tipo mediana com janela de 5×5 antes da CRC. O tipo de erro utilizado para treinamento das RNAs foi o LMS (erro médio quadrático).

6.2.1 Imagens sintéticas

6.2.1.1 Invariantes de momentos (equação 23)

a) Tempo de Treinamento

A tabela 1 mostra os tempos de convergência dos algoritmos de treinamento utilizados. Vemos a partir de tal tabela que o tempo de convergência do LM é bem menor que o GC. Para a condição de erro de 0.1 os tempos são semelhantes pois o mesmo é atendido logo de início.

Algoritmo	Erro aceitável (LMS)		
	0.1	0.01	0.001
GC	3s	1 h	6 h
LM	3s	6min	23min

Tabela 1: tempo de treinamento das RNAs para diferentes valores de erro. Caso 1: Invariantes de momentos (eq. 23).

b) Resultados de Classificação

As colunas com percentuais das tabelas mostradas em seguida, relativas à classificação, indicam os valores de ativação mínimos aceitáveis. Dessa forma, quanto menor esse valor, menos rigoroso com as respostas da rede. Como uma RNA é um aproximador universal de funções, os mapeamentos entre o conjunto domínio (entradas) - conjunto imagem (respostas) podem ser interpretados de forma mais ou menos severa. Por exemplo, para 100% da ativação, só aceitamos as máximas respostas da RNA que foram superiores às máximas ativações nominais dos neurônios de saída, que no caso é 1.0. Para outro valor, por exemplo 80%, todas as saídas máximas maior que 0.6 foram aceitas como a decisão da classe. Além disso, quando mais de uma saída apresentou valor máximo igual em mais de um neurônio de saída o padrão foi tido como não reconhecido, i.e., rejeição. Assim, a rejeição de um determinado padrão tem duas razões principais: mais de uma saída apresentou valores máximos iguais ou o valor máximo apresentado por um neurônio de saída não superou o valor mínimo aceitável.

1) Algoritmo de treinamento: Gradiente Conjugado

Para um erro aceitável de 0.1, temos a tabela 2 descrevendo o resultado da classificação da RNA para a configuração 1. A figura 25 ilustra os dados da tabela 2.

Ruído	100%	95%	90%	80%	70%	60%	50%
0	34.4%	94.4%	98.9%	98.9%	98.9%	98.9%	98.9%
5	32.2%	93.3%	97.8%	97.8%	97.8%	97.8%	97.8%
10	33.3%	91.1%	98.9%	98.9%	98.9%	98.9%	98.9%
15	36.7%	83.3%	88.9%	96.7%	96.7%	96.7%	96.7%
20	32.2%	80.0%	85.6%	95.6%	96.7%	96.7%	96.7%
25	36.7%	63.3%	80.0%	91.1%	92.2%	93.3%	93.3%
30	35.6%	60.0%	65.6%	74.4%	77.8%	78.9%	80.0%
35	28.9%	45.6%	51.1%	57.8%	62.2%	62.2%	62.2%
40	18.9%	23.3%	25.6%	28.9%	30.0%	32.2%	32.2%
45	15.6%	22.2%	23.3%	23.3%	23.3%	23.3%	23.3%

Tabela 2: Resultado da classificação para GC, erro = 0.1, invariantes de momentos (eq. 23).

Figura 25: Resultado da classificação para GC, erro = 0.1, invariantes de momentos (eq. 23).

A tabela 3 e a figura 26 ilustram os resultados para o erro igual a 0.01.

Ruído	100%	95%	90%	80%	70%	60%	50%
0	43.3%	98.9%	98.9%	98.9%	98.9%	98.9%	98.9%
5	38.9%	97.8%	97.8%	97.8%	97.8%	97.8%	97.8%
10	44.4%	94.4%	98.9%	98.9%	98.9%	98.9%	98.9%
15	44.4%	85.6%	93.3%	97.8%	97.8%	97.8%	97.8%
20	36.7%	76.7%	86.7%	96.7%	96.7%	96.7%	96.7%
25	42.2%	60.0%	78.9%	90.0%	95.6%	95.6%	95.6%
30	35.6%	61.1%	66.7%	74.4%	82.2%	84.4%	85.6%
35	27.8%	44.4%	54.4%	62.2%	66.7%	67.8%	70.0%
40	12.2%	18.9%	20.0%	24.4%	25.6%	25.6%	26.7%
45	12.2%	14.4%	15.6%	20.0%	21.1%	21.1%	22.2%

Tabela 3: Resultado da classificação para GC, erro = 0.01, Invariantes de Hu.

Figura 26: Resultado da classificação para GC, erro = 0.01, invariantes de momentos (eq. 23).

A tabela 4 e a figura 27 ilustram os resultados para o erro igual a 0.001.

Ruído	100%	95%	90%	80%	70%	60%	50%
0	55.6%	98.9%	98.9%	98.9%	98.9%	98.9%	98.9%
5	47.8%	97.8%	97.8%	97.8%	97.8%	97.8%	97.8%
10	46.7%	98.9%	98.9%	98.9%	98.9%	98.9%	98.9%
15	46.7%	91.1%	97.8%	97.8%	97.8%	97.8%	97.8%
20	51.1%	80.0%	93.3%	96.7%	96.7%	97.8%	97.8%
25	44.4%	67.8%	82.2%	92.2%	95.6%	95.6%	95.6%
30	44.4%	55.6%	65.6%	71.1%	80.0%	82.2%	82.2%
35	23.3%	41.1%	45.6%	48.9%	52.2%	57.8%	60.0%
40	7.8%	16.7%	21.1%	22.2%	22.2%	24.4%	27.8%
45	11.1%	13.3%	14.4%	14.4%	14.4%	15.6%	15.6%

Tabela 4: Resultado da classificação para GC, erro = 0.001, Invariantes de Hu.

Figura 27: Resultado da classificação para GC, erro = 0.001, invariantes de momentos (eq. 23).

2) Algoritmo de treinamento: Levenberg-Maquardt

De forma semelhante ao método anterior, as tabelas 5, 6 e 7, e as figuras 28, 29 e 30 apresentam os resultados de classificação do algoritmo de treinamento: Levenberg-Maquardt para as imagens sintéticas.

Ruído	100%	95%	90%	80%	70%	60%	50%
0	32.2%	87.8%	94.4%	98.9%	98.9%	98.9%	98.9%
5	32.2%	86.7%	90.0%	97.8%	97.8%	97.8%	97.8%
10	33.3%	85.6%	93.3%	98.9%	98.9%	98.9%	98.9%
15	42.2%	83.3%	93.3%	97.8%	97.8%	97.8%	97.8%
20	38.9%	76.7%	85.6%	92.2%	92.2%	92.2%	92.2%
25	45.6%	66.7%	73.3%	80.0%	82.2%	84.4%	84.4%
30	44.4%	63.3%	67.8%	72.2%	76.7%	77.8%	78.9%
35	37.8%	47.8%	50.0%	54.4%	57.8%	57.8%	60.0%
40	23.3%	25.6%	27.8%	30.0%	31.1%	31.1%	31.1%
45	12.2%	14.4%	15.6%	15.6%	15.6%	15.6%	15.6%

Tabela 5 Resultado da classificação para LM, erro = 0.1, invariantes de momentos (eq. 23)

Figura 28: Resultado da classificação para LM, erro = 0.1, invariantes de momentos (eq. 23).

Ruído	100%	95%	90%	80%	70%	60%	50%
0	55.6%	95.6%	98.9%	98.9%	98.9%	98.9%	98.9%
5	48.9%	94.4%	98.9%	98.9%	98.9%	98.9%	98.9%
10	48.9%	94.4%	98.9%	98.9%	98.9%	98.9%	98.9%
15	51.1%	90.0%	95.6%	97.8%	97.8%	97.8%	97.8%
20	52.2%	78.9%	86.7%	92.2%	96.7%	96.7%	96.7%
25	38.9%	58.9%	71.1%	85.6%	87.8%	91.1%	91.1%
30	41.1%	44.4%	52.2%	60.0%	68.9%	72.2%	73.3%
35	24.4%	32.2%	32.2%	34.4%	40.0%	42.2%	43.3%
40	10.0%	15.6%	17.8%	18.9%	18.9%	20.0%	20.0%
45	13.3%	15.6%	16.7%	16.7%	16.7%	16.7%	16.7%

Tabela 6: Resultado da classificação para LM, erro = 0.01, invariantes de momentos (eq. 23)

Figura 29 Resultado da classificação para LM, erro = 0.01, invariantes de momentos (eq. 23).

Ruído	100%	95%	90%	80%	70%	60%	50%
0	44.4%	98.9%	98.9%	98.9%	98.9%	98.9%	98.9%
5	35.6%	97.8%	98.9%	98.9%	98.9%	98.9%	98.9%
10	44.4%	97.8%	98.9%	98.9%	98.9%	98.9%	98.9%
15	47.8%	91.1%	95.6%	97.8%	97.8%	97.8%	97.8%
20	37.8%	70.0%	77.8%	88.9%	93.3%	94.4%	94.4%
25	33.3%	55.6%	61.1%	71.1%	80.0%	86.7%	87.8%
30	27.8%	41.1%	48.9%	53.3%	58.9%	62.2%	64.4%
35	21.1%	31.1%	33.3%	36.7%	41.1%	44.4%	47.8%
40	8.9%	14.4%	17.8%	20.0%	20.0%	20.0%	20.0%
45	14.4%	16.7%	16.7%	16.7%	16.7%	16.7%	16.7%

Tabela 7: Resultado da classificação para LM, erro = 0.001, invariantes de momentos (eq. 23)

Figura 30: Resultado da classificação para LM, erro = 0.001, invariantes de momentos (eq. 23).

Para os mesmos dados utilizados o método vizinho mais próximo apresentou os seguintes resultados:

Ruído	Tx.Reconh.
0	98.9%
5	97.8%
10	97.8%
15	94.4%
20	92.2%
25	70.0%
30	61.1%
35	43.3%
40	25.6%
45	18.9%

Tabela 8: Resultado da classificação para o método vizinho mais próximo.

6.2.1.2 Novos invariantes de momentos (equação 33)

Os novos invariantes de momentos apresentaram maiores dificuldades em relação ao treinamento, e em algumas arquiteturas, o erro não convergiu como esperado, apesar de diversas configurações testadas. A rede que apresentou melhor performance *tempo de treinamento x erro atingido* foi a configuração com seis neurônios nas camadas de entrada e saída, e dez neurônios nas duas camadas intermediárias, sendo treinada com o algoritmo LM. O tempo total de treinamento foi

de 27 horas e três minutos e o erro mínimo atingido foi de 0.71. Os resultados de classificação estão mostrados na tabela 9 e o gráfico da figura 31.

Ruído	100%	95%	90%	80%	70%	60%	50%	NN
0	65.6%	86.7%	91.1%	91.1%	91.1%	91.1%	91.1%	76.1%
5	56.7%	82.2%	84.4%	87.8%	90.0%	90.0%	90.0%	73.3%
10	58.9%	81.1%	83.3%	84.4%	85.6%	85.6%	85.6%	75.6%
15	57.8%	77.8%	78.9%	78.9%	80.0%	82.2%	85.6%	76.7%
20	51.1%	64.4%	68.9%	70.0%	71.1%	72.2%	72.2%	65.6%
25	42.2%	53.3%	53.3%	58.9%	60.0%	61.1%	62.2%	57.8%
30	25.6%	43.3%	43.3%	46.7%	46.7%	50.0%	50.0%	43.3%
35	28.9%	32.2%	32.2%	36.7%	36.7%	36.7%	36.7%	34.4%
40	15.6%	24.4%	26.7%	28.9%	30.0%	31.1%	31.1%	32.2%
45	7.8%	10.0%	11.1%	13.3%	13.3%	13.3%	13.3%	12.2%

Tabela 9: Resultado da classificação para LM e vizinho mais próximo, erro = 0.7, IM (eq. 33)

Figura 31: Resultado da classificação para LM, erro = 0.7, NIM (eq. 33).

6.2.1.3 Comentários sobre classificação em imagens sintéticas

Vemos, pelas tabelas 1-7 e pelos gráficos das figuras 25-30, que em média, para taxas de ruído abaixo de 25% o sistema apresenta performance muito boa, principalmente quanto menor o limiar mínimo aceitável de ativação dos neurônios da última camada (ϕ).

Comparando os resultados das RNAs com o método vizinho mais próximo (tabela 8), vemos que as redes apresentaram resultados de classificação superiores a este último, principalmente para taxas de ruído acima de 20 %. Comparando os dois algoritmos, o GC apresentou tempo de convergência bem superior ao LM, porém, os resultados de classificação foram superiores a este último. Assim, havendo necessidade de treinamento rápido, pode-se utilizar o LM, e se há necessidade de maximizar a taxa de reconhecimento, deve-se utilizar o GC.

Em relação aos novos IMs, derivados da equação 33, apresentaram resultados inferiores aos IMs da equação 23. Ainda assim, os resultados das RNAs foram superiores ao do vizinho mais próximo. Uma das explicações para esta menor taxa de reconhecimento em relação ao outro conjunto de atributos é a maior interseção entre as curvas de distribuição dos atributos, ver apêndice B. Wong et al. (1995) descreveram performance superior a 98% para estes atributos, porém informações estatísticas foram utilizadas para aumentar a separabilidade entre classes: foi utilizado o método k-vizinho mais próximo ponderado, com pesos do tipo $1/\sigma_i$, de σ_i é o desvio padrão dos atributos de uma classe. Assim, para valores baixos de σ_i a importância do atributo é elevada, e caso contrário é minimizada.

6.2.2 Imagens reais

Diversos testes foram efetuados, variando-se os parâmetros das redes, como por exemplo, o número de camadas e o número de neurônios em cada uma das camadas. Os resultados não foram satisfatórios pois para os critérios desejados de erro mínimo e uma medida tomada foi diminuir os valores de cada atributo (apêndice B). A diminuição dos valores dos atributos é diretamente proporcional ao erro inicial das redes, visto que o processo de aprendizado é um mapeamento vetorial e se diminuirmos a amplitude dos vetores diminuimos o erro. Além disso, a separabilidade entre classes deve ser mantida, e se possível estendida. Porém, não é fácil encontrar transformações que efetuem as duas coisas ao mesmo tempo e continuando ser um processo automático e com o mínimo tempo possível. A saída para diminuir os valores

dos atributos foi verificar, para o banco de imagens de treinamento, os máximos valores de todos os atributos, para todas as classes. Os atributos foram recalculados para todos os objetos utilizando o número inteiro seguinte ao máximo valor dos atributos como denominador. Assim, todos os valores de treinamento passaram a ter valor entre 0 e 1. Em relação aos valores de teste, os mesmos denominadores derivados dos dados de treinamento foram utilizados. Os resultados tanto em termos de velocidade de treinamento quanto em generalização foram melhores que os obtidos anteriormente, i.e., sem a utilização do denominador.

6.2.2.1 Invariantes de momentos (equação 23)

Para as nove classes, o método vizinho mais próximo apresentou uma taxa de classificação de 72.59%. A tabela 10 ilustra os resultados de classificação para duas redes treinadas com algoritmos diferentes, o LM e o GC.

Algoritmo	100%	95%	90%	80%	70%	60%	50%
LM	62.2%	93.3%	97.8%	98.5%	98.5%	98.5%	98.5%
GC	63.7%	89.6%	95.6%	97.0%	97.8%	98.5%	98.5%

Tabela 10: Resultado da classificação para imagens reais: algoritmos LM e GC.

Os resultados na linha LM se referem a uma rede com seis neurônios na camada de entrada, vinte neurônios na camada intermediária e nove neurônios na camada de saída. O erro atingido foi 0.02 para 24 horas e 30 minutos de treinamento.

Os resultados na linha GC se referem a uma rede com seis neurônios na camada de entrada, trinta neurônios na primeira e na segunda camadas intermediárias, e nove neurônios na camada de saída. O tempo de treinamento para atingir o erro de 0.02 foi de 192 horas. O treinamento com o GC para uma RNA do mesmo tamanho que no LM do parágrafo anterior (com apenas dez neurônios na primeira camada intermediária) foi muito lento. Após 11 dias, i.e., 264 horas, o erro apresentado era superior a 0.194. Este e outros motivos levaram a uma mudança na arquitetura da rede.

Para melhor visualizar o processo de decisão das RNAs, podemos gerar uma *confusion-matrix*, i.e., uma matriz que associa cada padrão de entrada à classe derivada pelo sistema. Para o caso ideal teríamos uma matriz diagonal perfeita. Na prática, porém, geralmente, podemos aceitar, dependendo do grau de precisão requerido pelo sistema, uma matriz diagonal quase perfeita.

Para o método do vizinho mais próximo a *confusion-matrix* está apresentada na tabela 11-a. As tabelas 11-b e 11-c apresentam os resultados para RNAs treinadas com GC e LM (referente à tabela 10), onde a ativação mínima dos neurônios da camada de saída (A_m) foi estabelecido em 90% do valor nominal máximo dos neurônios. Diminuindo este valor limiar a taxa de reconhecimento aumenta, pois alguns padrões que foram rejeitados por não terem sido mapeados em suficiente intensidade pela RNA, i.e., que aparecem na última coluna (rejeição), agora passam a contabilizar em prol da taxa de classificação. Nas tabelas 11-d e 11-e, o valor de ativação (A_m) foi estabelecido em 50%, resultando no fato que alguns padrões tido como não reconhecidos nas tabelas 11-b e 11-c foram mapeados corretamente nesta nova condição.

CM	NN REAL - H									
	1	2	3	4	5	6	7	8	9	0
1	8	0	0	0	2	5	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0
3	0	0	10	0	0	2	0	3	0	0
4	0	0	0	10	0	0	0	0	5	0
5	0	0	0	0	15	0	0	0	0	0
6	1	0	0	0	0	14	0	0	0	0
7	0	0	0	0	0	0	15	0	0	0
8	1	0	3	0	0	0	0	11	0	0
9	0	0	0	15	0	0	0	0	0	0
Soma Diag: 98 % Acerto: 72.6% Tot.Padr.: 135										

Tabela 11-a: *Confusion-matrix* para o método vizinho mais próximo

Confusion-Matrix GRH-RUN T = 90%										
	1	2	3	4	5	6	7	8	9	0
1	15	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0
3	0	0	14	0	0	0	0	0	0	1
4	0	0	0	15	0	0	0	0	0	0
5	0	0	0	0	15	0	0	0	0	0
6	0	0	0	0	0	13	0	0	0	2
7	2	0	0	0	0	0	13	0	0	0
8	0	0	0	0	0	0	0	14	0	1
9	0	0	0	0	0	0	0	0	15	0
Soma Diag: 129 % Acerto: 95.6% Tot.Padr.: 135										

Tabela 11-b: *Confusion-matrix* para o RNA treinada com GC, limiar = 90%.

Confusion-Matrix LRH-RUN T = 90%										
	1	2	3	4	5	6	7	8	9	0
1	15	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0
3	0	0	15	0	0	0	0	0	0	0
4	0	0	0	15	0	0	0	0	0	0
5	0	0	0	0	15	0	0	0	0	0
6	0	0	0	0	0	13	0	0	0	2
7	0	0	0	0	0	0	15	0	0	0
8	0	0	0	0	0	0	0	14	0	1
9	0	0	0	0	0	0	0	0	15	0
Soma Diag: 132 % Acerto: 97.8% Tot.Padr.: 135										

Tabela 11-c: *Confusion-matrix* para o RNA treinada com LM, limiar = 90%.

Confusion-Matrix GRH-RUN T = 50%										
	1	2	3	4	5	6	7	8	9	0
1	15	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0
3	0	0	15	0	0	0	0	0	0	0
4	0	0	0	15	0	0	0	0	0	0
5	0	0	0	0	15	0	0	0	0	0
6	0	0	0	0	0	15	0	0	0	0
7	1	0	0	0	0	0	13	0	0	1
8	0	0	0	0	0	0	0	15	0	0
9	0	0	0	0	0	0	0	0	15	0
Soma Diag: 133 % Acerto: 98.5% Tot.Padr.: 135										

Tabela 11-d: *Confusion-matrix* para o RNA treinada com GC, limiar = 50%.

Confusion-Matrix LRH-RUN T = 50%										
	1	2	3	4	5	6	7	8	9	0
1	15	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0
3	0	0	15	0	0	0	0	0	0	0
4	0	0	0	15	0	0	0	0	0	0
5	0	0	0	0	15	0	0	0	0	0
6	0	0	0	0	0	15	0	0	0	0
7	0	0	0	0	2	0	13	0	0	0
8	0	0	0	0	0	0	0	15	0	0
9	0	0	0	0	0	0	0	0	15	0
Soma Diag: 133 % Acerto: 98.5% Tot.Padr.: 135										

Tabela 11-e: *Confusion-matrix* para o RNA treinada com LM, limiar = 50%.

O resultado das RNAs foi superior ao do método vizinho mais próximo, que, pela semelhança entre alguns dos objetos utilizados, apresentou decisões erradas, principalmente na classe 9, que foi muitas vezes reconhecida como a classe 4. Os resultados apresentados na tabela 10 mostram que o LM é superior, neste caso, tanto no critério de tempo de treinamento quanto no de generalização (com exceção da primeira coluna).

Diversas outras arquiteturas foram testadas, porém as que obtiveram melhores resultados em termos de tempo de treinamento foram as mencionadas. No geral, o aumento de neurônios nas camadas intermediárias tornou o sistema mais lento, como era de se esperar. Porém, a performance do método LM cai muito mais que o GC com este parâmetro. Para esta implementação, uma rede com 6 neurônios de entrada, vinte e quinze neurônios na primeira e na segunda camadas intermediárias, respectivamente, e nove neurônios na camada de saída levou o sistema a uma falta de memória, sendo abortado. Diminuindo o número de vinte neurônios para quinze, na primeira camada, o sistema iniciou treinamento, i.e., não abortou, porém o tempo de treinamento foi bem maior que o GC. Assim, confirmamos as predições do capítulo 3, em que afirmamos que para redes pequenas o LM apresenta performance superior, em relação a treinamento, ao GC, para redes de igual tamanho.

6.2.2.2 Novos invariantes de momentos (equação 33)

De forma semelhante ao anterior, os dados de treinamento e teste foram ponderados pelo inteiro seguinte ao máximo valor dos atributos da base de dados de treinamento. Porém, as redes treinadas com os atributos derivados da equação 33 não apresentaram bons resultados como os da equação 23, tanto em relação ao tempo de treinamento quanto em relação à generalização. Alguns resultados de classificação para duas configurações e os dois algoritmos são mostrados na tabela 12. Para o mesmo conjunto de dados o método vizinho mais próximo apresentou taxa de acerto de 56.30%.

Algoritmo	100%	95%	90%	80%	70%	60%	50%
LM-1	27.4%	43.7%	48.1%	51.9%	54.8%	57.0%	57.8%
LM-2	36.3%	52.6%	57.0%	58.5%	59.3%	60.0%	60.7%
GC-1	19.3%	25.9%	27.4%	30.4%	32.6%	34.8%	36.3%
GC-2	31.9%	34.8%	40.7%	45.9%	51.9%	57.0%	60.0%

Tabela 12: Resultado da classificação para imagens reais: algoritmos LM e GC.

As duas primeiras linhas da tabela 12 referem-se a redes com seis neurônios na camada de entrada, vinte neurônios na primeira camada intermediária, dez neurônios na segunda camada intermediária, e nove neurônios na camada de saída. A primeira linha mostra o resultado dessa rede para um erro de 0.025, obtido após 156 horas de processamento. A segunda linha mostra o resultado da rede para um erro de 0.25, obtido após 67 horas.

A terceira linha da tabela 12 refere-se a uma rede com seis neurônios na camada de entrada, oitenta neurônios na primeira camada intermediária, cem neurônios na segunda camada intermediária, e nove neurônios na camada de saída. Após 180 horas de treinamento o erro obtido foi 0.38. A última linha da tabela 12 refere-se a uma rede com seis neurônios na camada de entrada, trinta neurônios nas duas camadas intermediárias, e nove neurônios na camada de saída. Após 190 horas de treinamento o erro obtido foi 2.01.

Apesar dos erros da segunda e quarta linhas da tabela serem maiores que os da primeira e terceira, os resultados de classificação correta são melhores que dessas últimas, o que é um resultado não esperado. A utilização dos invariantes de momentos derivados da equação 33 não apresentaram resultados muito superiores ao convencional vizinho mais próximo. Alguns resultados pouco superiores quando o limiar de ativação mínimo da rede é de abaixo de 60% da máxima ativação. Porém, considerando o tempo gasto para treinamento das redes, este pequeno ganho não é significativo. Diversas outras configurações de redes foram testadas, porém, os resultados não foram satisfatórios. Provavelmente os novos invariantes de momentos apresentem uma melhor performance se utilizados em conjunto com uma transformação que aumente a separabilidade entre classes, como foi utilizado em Wong et al. (1995).

6.2.3 Partição do espaço de atributos com o aumento do número de classes

Para avaliar como a partição do espaço de atributos influi no processo de treinamento e classificação, foi realizada uma experiência, com a base de imagens reais, aumentando-se o número de classes para uma dada configuração de neurônios nas camadas de entrada (seis neurônios) e internas (vinte neurônios).

O método de treinamento foi o híbrido LM e SA (Masters, 1995) e foi estabelecido um limiar para a fase de ID de 90% da ativação nominal máxima dos neurônios. A tabela 13 ilustra os resultados:

No. Classes	No. Padrões	Tempo Trein.	Err. Desej.	Err. Inic.	Err. Obtido	Tx. Classif.
2	30	0:00:02	0.050	0.000028	0.000028	100.00%
3	45	0:00:02	0.050	0.000959	0.000959	100.00%
4	60	0:00:02	0.050	0.001316	0.001316	100.00%
5	75	0:00:02	0.050	0.014823	0.014823	97.33%
6	90	0:47:50	0.050	0.077435	0.048745	91.11%
7	105	3:18:02	0.050	0.113869	0.049590	92.38%
8	120	5:10:00	0.050	0.107085	0.047200	94.17%
9	135	4:06:00	0.050	0.471600	0.047520	93.33%

Tabela 13: resultados da partição do espaço de atributos para uma dada configuração

Para cada classe foram utilizados 15 padrões para cada classe durante o treinamento (15 também para testes), assim, a segunda coluna da tabela 13 indica o número total de padrões utilizado para treinamento. Para poucas classes (<6), o tempo de treinamento foi desprezível (2 segundos), pois o erro inicial dado pelo sistema foi menor que o exigido (no caso 0.05). A taxa de classificação é apresentada na última coluna, lembrando que para outro valor mínimo de ativação dos neurônios da camada de saída, os valores podem mudar. Por exemplo, para a penúltima linha, i.e., 8 classes, escolhendo um limiar de 70% do valor nominal, a taxa de classificação passa de 94.17% para 99.17%. Isto reflete que a sensibilidade do mapeamento vetorial realizado pelo processo de treinamento. Um resultado intrigante é o fato de que foi necessário mais tempo para a RNA convergir quando haviam 8 classes do que quando haviam 9. O autor desenvolve atualmente pesquisas para investigar melhor como as superfícies de decisão das RNAs tipo MLFN, i.e., hiperplanos, convergem para uma dada configuração durante o treinamento com o aumento de classes em um determinado número de dimensões (vetor de atributos).

6.3 Tempo total de execução

Uma vez a RNA treinada, o tempo de execução é uma soma de três tempos: o tempo de filtragem, o tempo de codificação de regiões conectadas e extração de atributos destas, e o tempo de classificação pela RNA.

Para os testes com imagens reais desconsideramos a primeira etapa, a filtragem, por economia de tempo. Porém, para uma imagem de 128x128 pixels, o tempo de uma filtragem do tipo mediana com janela de 5x5 foi de 1 segundo, em um computador do tipo 486 DX2 66 MHz. O tempo de codificação da imagem e extração de características é de 0.060 segundos para cada padrão na imagem. O último tempo, relativo a execução da RNA, depende da arquitetura utilizada. Para a arquitetura utilizada no método LM para os invariantes de momentos (eq. 23), i.e., 6, 20 e 9

neurônios nas camadas de entrada, intermediária e de saída, respectivamente, o tempo de execução para cada padrão é cerca de 0.01 segundo, enquanto que para a rede com 30 neurônios nas duas camadas intermediárias, que foi treinada com o método GC, o tempo de execução é de 0.017 segundos. Assim, temos um tempo total de 0.07 segundos na primeira e 0.077 segundos na segunda configuração, o que corresponde a cerca de 14 e 13 padrões por segundo para a primeira e segunda configurações, respectivamente. Para o caso de imagens apresentando apenas um padrão por cena este é o número de imagens que o sistema é capaz de reconhecer por unidade de tempo.

De forma semelhante para imagens sintéticas e ruidosas, o tempo de execução pode ser dado por $T = T_1 + T_2 + T_3$, onde $T = N * TF$, onde N é o número de filtrações do tipo mediana desejado e TF é o tempo para esta operação (1 segundo para imagens de 128x128 pixels); T_2 é o tempo de codificação e eliminação de regiões de área pequena, dado pelo gráfico da figura 19, e T_3 é o tempo de classificação, que para redes pequenas como as utilizadas para esta classe de imagens, fica em torno de 0.01 segundo por padrão. Assim, para uma imagem com 15% de ruído, como a mostrada na figura 14, o tempo de execução do sistema é, para uma filtração, cerca de 2.20 segundos, o que permite o reconhecimento de cerca de 27 objetos por minuto.

O tempo total de processamento é diretamente proporcional ao tamanho das imagens. Para imagens com dimensões maiores, por exemplo, 512x512 pixels, o tempo de 1 filtração do tipo mediana é 16 vezes o tempo para o mesmo procedimento numa imagem de 128x128 pixels. Da mesma forma ocorre em relação à codificação dos padrões e a extração de atributos. Assim, pode-se pensar em utilizar filtros em hardware e esquemas melhores de segmentação de padrões, com o objetivo de minimizar os gargalos de tempo do sistema.

Capítulo 7.

Conclusões

Redes neurais artificiais podem ser utilizadas como uma etapa de um sistema de visão para muitas tarefas como a extração de atributos, classificação, otimização e auto-associação. RNAs possibilitam a geração de regiões de decisão não lineares e dessa forma podem obter melhores resultados que classificadores tradicionais. Diversas outras tarefas da área de visão computacional podem também serem realizadas por RNAs, como extração de bordas (Paik et al., 1992) e de área (Ogata et al., 1992), orientação de objetos (Morris et al., 1990), restauração de imagens (Zhou et al., 1988) e reconhecimento de movimento e textura (Polana, 1994).

O objetivo deste trabalho foi descrever aspectos relacionados ao reconhecimento de padrões visuais, dando um enfoque às técnicas que utilizam redes neurais artificiais, e propor um sistema de reconhecimento de objetos invariante a transformações geométricas para uma aplicação em sistemas de inspeção visual automática. Foi mostrado que o sistema proposto tem capacidade de tratar imagens ruidosas e sujeitas a distorções geométricas como rotações, translações e mudança de escala, e discutimos alguns dos aspectos de implementação e da parte prática.

Dois algoritmos foram utilizados para treinamento das RNAs, o Levenberg-Marquardt e o Gradiente Conjugado, sendo que o primeiro apresenta maior velocidade de convergência para redes pequenas enquanto que o segundo apresentou uma melhor generalização para as condições de erro testadas. Alguns resultados de classificação foram superiores a 98% o que indica a eficiência do mesmo, superior ao método vizinho mais

próximo principalmente quando a taxa de ruído foi acima de 25%. Imagens reais e sintéticas foram utilizadas para testar a capacidade de convergência dos algoritmos e de generalização em relação a padrões não utilizados na etapa de treinamento.

Apesar do sistema ter sido desenvolvido para IVA, podemos facilmente adaptá-lo para outras aplicações, como reconhecimento óptico de caracteres. Além disto, imagens de tamanhos diferentes podem ser utilizadas tanto para treinamento quanto para execução, pois foi adicionada uma rotina de normalização de coordenadas da imagem.

Diversas otimizações podem ser feitas no sentido de aumentar a velocidade de treinamento e execução do sistema, além do aumento da taxa de reconhecimento. Em relação à velocidade, podemos otimizar o cálculo dos atributos através de técnicas matemáticas, como a transformada de Hadamard para o cálculo de momentos, e este objetivo também pode ser contornado pelo uso de computadores paralelos ou seriais de maior poder computacional em relação ao utilizado. Além disso, outras transformações sobre os atributos mais eficientes que o logaritmo do módulo, podem ser obtidas, aumentando a separabilidade entre classes.

A taxa de reconhecimento pode ser otimizada através da minimização de pequenos erros, em cada fase do sistema, como por exemplo o algoritmo de segmentação, mas que em cascata, produzem uma grande diferença na performance global do sistema. Pode-se utilizar métodos para melhorar a qualidade da imagem como filtragem homomórfica, i.e., filtro sequencial alternado sobre a imagem binária, ou equalização do histograma, antes da segmentação. Esta última tarefa também podem ser implementada em RNAs, (Pal e Pal, 1993).

Redes neurais em paralelo, treinadas com conjuntos de atributos diferentes, podem ser implementadas e a correlação das respostas verificada. Esta experiência está sendo efetuada no momento pelo autor, seguindo uma abordagem de sistemas tolerantes a falhas, onde a redundância de processamento, também existente em alguns organismos biológicos, proporciona uma melhor probabilidade de acerto em uma decisão. Espera-se

que as saídas correspondentes tenham valores próximos para um determinado padrão. Podemos utilizar a informação da correlação entre as respostas das redes para aceitar ou rejeitar a decisão das redes.

Outras extensões que podem ser feitas a este trabalho são a escolha de outros atributos para representar os objetos, ou até mesmo uma escolha mais apropriada do conjunto de atributos utilizado, novos filtros no estágio de pré-processamento, etc. As RNAs, apesar de terem sido implementadas em software, podem facilmente ser implementadas em hardware utilizando-se apenas um processador neural como o CI 170 NDX da Intel, entre outros. Tal implementação diminuiria bastante o tempo de resposta da rede e liberaria o computador para outras atividades. Ainda mais, a combinação de RNAs a outras tecnologias como lógica Fuzzy, e algoritmos genéticos para otimização do aprendizado das RNAs, também podem ser adicionados ao sistema, o que poderá conduzir à obtenção de soluções ainda mais robustas.

Referências Bibliográficas

- AGUI, T.; TAKAHASHI, H.; NAKAJIMA, M.; NAGAHASHI, H. (1991). Recognition of handwritten katakana in a frame using moment invariants based on neural network. In: Visual Communication and Image Processing, Proceedings, Boston, v. 1606, p. 188-197.
- AL-EMAMI, S.; USHER, M. (1990). On-line recognition of handwritten arabic characters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v.12, n.7.
- AWWAL, A. A. S.; AHMED, F. (1993). Complex associative memory neural network model for invariant pattern recognition, IEEE National Aerospace and Eletronics Conference, Proceedings, Dayton, OH, p. 892-896.
- BALOGH, A. A.; WAXMAN, A. M. (1991). Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN. *Neural Networks*, v. 4, n. 3, p. 271-302.
- BARNARD, E.; CASASENT, D. (1990). Shift invariance and the neocognitron. *Neural Networks*, v. 3, p. 403-410.
- BARNARD, E.; CASASENT, D. (1990b). Trajectories and constraint-based distortion-invariance. *Proceedings of SPIE*, v. 1295, p. 39-44.
- BARNARD, E.; CASASENT, D. (1991). Invariance and neural nets, *IEEE Trans. on Neural Networks*, v. 2, n. 5, p. 498-508.
- BEALE, R.; JACKSON, T. (1990). Neural computing, an introduction. Adam Hilger, IOP Publishing Ltd: Bristol

- BEBIS, G. N.; PAPADOURAKIS, G.M. (1991). Object recognition using invariant object boundary representations and neural network models. *Pattern Recognition*, v. 25, n. 1, p. 25-44.
- BELKASIM, S. O.; SHRIDHAR, M.; AHMADI, M. (1991). Pattern recognition with moment invariants: a comparative study and new results. *Pattern Recognition*, v. 24, n.12, p. 117-1138.
- BREUEL, T.M. (1992). Geometrical aspects of visual object recognition. *Ph.D. Thesis*, MIT.
- BREUEL, T.M. (1993). Higher-order statistics in visual object recognition, *Memo 93-02*, IDIAP, Suíça.
- BUHMANN, J.; SCHULTEN, K. (1988). Invariant pattern recognition by means of fast synaptic plasticity. In: IEEE Conference on Neural Networks, San Diego, CA. Proceedings, p. 125-132.
- BUHMANN, J.; LANGE, J.; VON DER MALSBURG, C. (1989). Distortion invariant object recognition by matching hierarchically labeled graphs, Intl. Joint Conference on Neural Networks. Proceedings. p. 155-159.
- BUHMANN, J.; LADES, M.; VON DER MALSBURG, C. (1990). Size and distortion invariant object recognition by hierarchical graph matching. In: Intl. Joint Conference on Neural Networks. Proceedings. p. 411-416.
- BURKHARDT, B.; MULLER, X. (1980). On invariant sets of a certain class of fast translation-invariant transforms. *IEEE Trans. on Acoustic, Speech & Signal Processing*, v. 28, p. 517-523.
- CARPENTER, G.A.; GROSSBERG, S.; MEHANIAN, C. (1989). Invariant recognition of cluttered scenes by a self-organizing ART architecture: CORT-X boundary segmentation. *Neural Networks*, v. 2, n. 3, p. 169-181.

- CARVALHO, A.; FAIRUSH, M.C.; BISSET, D.L. (1993). Classifying images using goal-seeking neural network architectures. *IEE Proceedings*, v. 140, n. 1, p. 12-18.
- CARVALHO, A.; FAIRHUST, M.C.; BISSET, D. (1994). Feature extraction using the SOFT boolean architecture. In: I Simpósio Brasileiro de Redes Neurais. Anais, p. 127-132.
- CARVALHO, A.C.P.L.F. (1995). A modular neural architecture for pattern recognition. In: II Simpósio Brasileiro de Redes Neurais, Anais, p. 1-6.
- CASASENT, D.; BARNARD, E. (1990). Adaptive clustering neural net for piecewise nonlinear discriminant surfaces. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 423-428.
- CHAN, L-W. (1991). Translational invariant object recognition using back-propagation network. In: Intl. Joint Conference on Neural Networks. Proceedings.
- CHAO, T-H. (1992). Automatic target recognition using a feature-based optical neural network. In: Optical Pattern Recognition III, Proceedings. Orlando, FL, USA
- CHAO, T-H.; STONER, W.W. (1993). Optical implementation of a feature-based neural network with application to automatic target recognition. *Applied Optics*, v. 32, n. 8, p. 1359-1369.
- CHEN, S.S.; KELLER, J.M.; CROWNOVER, R.M. (1993). On the calculation of fractal features from images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v.15, p. 1087-1090.
- CHIN, R. T.; DYER, C. R. (1986). Model-based recognition in robot vision. *Computing Surveys*, v. 18, n. 1, p. 67-108.
- CLARKSON, M. (1995). Olhos, Ouvidos e Cérebro em um Chip. *Byte Brasil*, v. 4, n. 2, p. 75-80.

COSTA, L. F. (1993a). On the most advanced image processing system: the visual cortex, its modelling and implications for computer vision. In: Workshop sobre Computação de Alto Desempenho. Anais, São Carlos.

COSTA, L. F. (1993b). Redes neurais artificiais para reconhecimento de padrões visuais. In: Workshop em Aplicações de Redes Neurais. Anais, São Paulo.

COSTA, J. A. F.; GONZAGA, A. (1995). A System for object recognition using neural networks and normalized Zernike moment invariants, submitted to SIBGRAPI '95.

COSTA, J. A. F.; GONZAGA, A. (1995). Invariant object recognition using artificial neural networks. In: Simpósio Nipo-Brasileiro de Ciências e Tecnologia. Anais, S. Paulo, p. 100-110.

COVER, T.M.; HART, P.E. (1967). Nearest Neighbor pattern Classification. *IEEE Trans. on Information Theory*, v. IT-13, n.1.

CRUZ, V.; CRISTOBAL, G.; MICHAUX, T.; BARQUIN, S. (1989). Invariant image recognition using a multi-network neural model. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 17-22.

DELOPOULOS, A.; TIRAKIS, A.; KOLLIAS, S. (1994). Invariant image classification using triple-correlation-based neural networks. *IEEE Trans. on Neural Networks*. v. 5, n. 3, p. 392-408.

DI ZENZO, S.; BUONO, M. D.; MEUCCI, M.; SPIRITO, A.(1992). Optical recognition of hand-printed characters of any size, position, and orientation. *IBM Journal of Research and Development*. v. 36, n. 3, p.487-501.

DOBNIKAR, A.; FICZKO, J.; PODBREGAR, D.; REZAR, U. (1992). Invariant pattern classification neural network versus FT approach. *Microprocessing and Microprogramming*, v. 33, n. 3, p. 161-168.

- DONY, R.; HAYKIN, S. (1995). Neural network approaches to image compression. *Proceedings of IEEE*, v. 83, p. 288-303.
- DUDA, R. O.; HART, P. E. (1973). *Pattern classification and scene analysis*. Wiley.
- ELLIMAN, D.G.; R.N. BANKS (1990). Shift invariant neural net for machine vision. *IEE Proceedings*, v. 137, n. 3, p. 183-187.
- ERGEZINGLER, S.; THOMSEN, E. (1995). Accelerated learning algorithm for multilayer perceptrons: optimizations layer by layer. *IEEE Trans. on Neural Networks*, v. 6, n. 1, p. 31-42.
- FISCHBACH, G.D. (1992). Mind and Brain. *Scientific American*, v. 267. n. 3, p. 24-33.
- FLORES, E.L.; REZENDE, E.N.; CARRIJO, G.A.; YABUTI, J.B. (1994). Reconhecimento de caracteres manuscritos utilizando redes neurais. In: I Congresso Brasileiro de Redes Neurais. Anais.
- FRUEHAUF, N.; LUEDER, E.; BADER, G. (1993). Fourier optical realization of cellular neural networks. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, v. 40, n. 3, p. 156-162.
- FU, C-W.; YEN, J-C.; CHANG, S. (1993). Calculation of moments invariants via Haddamard transform. *Pattern Recognition*, v. 26, n. 2, p. 287-294.
- FU, K.S. (1974). *Syntatic methods in pattern recognition*. New York, NY, Academic Press.
- FU, L.M. (1990). Analisis of the Dimensionality of Neural Networks for Pattern Recognition. *Pattern Recognition*, v. 23, n. 10, p. 1131-1140.
- FUKUMI, M.; OMATU, S.; TAKEDA, F.; KOSAKA, T. (1991). Rotation-invariant pattern recognition system with application to coin recognition, In: Intl. Joint Conference on Neural Networks. Anais, p. 1027-1032.

- FUKUMI, M.; OMATU, S.; TAKEDA, F.; KOSAKA, T. (1992). Rotation-invariant neural pattern recognition system with application to coin recognition, *IEEE Trans. on Neural Networks*, v. 3, n. 2, p. 272-279.
- FUKUSHIMA, K. (1970). An eletronic model of the retina. *Proceedings of the IEEE*, v. 58, n. 12, p. 1950-1951.
- FUKUSHIMA, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, v. 36, n. 4, p. 193-202.
- FUKUSHIMA, K.; MIYAKE, S. (1982). Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, v. 15, n. 6, p. 455-469.
- FUKUSHIMA, K. (1988). A neural network model for visual pattern recognition. *Computer*, v. 21, n. 3, p. 65-75.
- FUKUSHIMA, K. (1989). Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, v. 2, n. 6, p. 413-420.
- FUKUSHIMA, K. (1991). Neural network models for visual pattern recognition. *IEICE Transactions*, v. E74, n. 1, p. 179-190.
- FUKUSHIMA, K. (1992). Character recognition with neural networks. *Neurocomputing*, v. 4, n. 5, p. 221-233.
- FUKUSHIMA, K.; IMAGAWA, T. (1993a). Recognition and segmentation of connected characters with selective attention. *Neural Networks*, v. 6, n. 1, p. 33-41.
- FUKUSHIMA, K. (1993b). Improved Generalization Ability Using Constrained Neural Network Architectures. In: Intl. Joint Conference on Neural Networks, Proceedings. Nagoya, p. 2049-2054.

FUKUSHIMA, K.; OKADA, M.; HIROSHIGE, K. (1994). Neocognitron with dual C-cell layers. *Neural Networks*, v. 7, n.1, p. 41-47.

GEE, A.H.; AIYER, S.V.B.; PRAGER, R.W. (1991). A subspace approach to invariant pattern recognition using Hopfield networks. In: Intl. Conference on Neural Networks, Proceedings. Singapore.

GILES, C.L.; MAXWELL, T. (1987). Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, v. 26, p. 4972-4978.

GHORBEL, F. (1994). A complete invariant description for gray-level images by the harmonic analysis approach. *Pattern Recognition Letters*, v. 15, p. 1043-1051

GOMES, H.M.; CARVALHO FILHO, E.C.B. (1994). Reconhecimento off-line de assinaturas utilizando MLP-backpropagation e momentum: um estudo comparativo. In: I Congresso Brasileiro de Redes Neurais. Anais. Itajubá - MG.

GRIESSER, J.W. (1992). Experts among Us. *Business Horizons*, v. 35, n. 3, p. 77-80.

GROSSBERG, S. (1976). Adaptive pattern classification and universal recoding, II: feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, vol. 23.

GUPTA, L.; SAYEH, M. R.; TAMMANA, R. (1989). A neural network approach to robust shape classification. *Pattern Recognition*, v. 23, n. 6, p. 563-568.

GURNEY, K. (1994). *Lecture Notes on Neural Networks*. Anonymous FTP <ftp.brunel.ac.uk/gurney>.

HAMMERSTROM, D. (1993a). Neural Networks at Work. *IEEE Spectrum*, p. 26-32, June.

HAMMERSTROM, D. (1993b). Working with Neural Networks. *IEEE Spectrum*, July, p. 46-53,

- HARTMANN, G.; DRUEE, S.; KRAEUTER, K.O.; SEIDENBERG, E.; WIEMERS, H. (1993). How to provide pattern representations for learning and recognizing differently located objects from arbitrary camera positions. In: Intl. Joint Conf. on Neural Networks. Proceedings, v. 3, p. 2169-2172.
- HARTONO, P.; HASHIMOTO, S. (1993). Active learning algorithm of neural network. In: Intl. Joint Conf. on Neural Networks. Proceedings, p. 2548-255
- HECHT-NIELSEN, R. (1988). Neurocomputing: picking the human brain. *IEEE Spectrum*, v. 25, n. 3, March, p. 36-41.
- HERTZ, J.; KROGH, A.; PALMER, R. (1991). Introduction to the theory of neural computation. Redwood City. Addison-Wesley.
- HIMES, G.S.; I IGO, R.M. (1992). Automatic Target Recognition Using a Neocognitron. *IEEE Trans. Knowledge and Data Engineering*, v. 4, n. 2.
- HINTON, G. E. (1992). How neural networks learn from experience. *Scientific American*, v. 267, n. 3, p. 105-110.
- HOFFMAN, J.; SKRZYPEK, J.; VIDAL, J.J. (1993). Cluster network for recognition of handwritten, cursive script characters. *Neural Networks*, v. 6, n 1, p. 69-78.
- HOLIDAY, D. J.; SAMAL, A. (1995). Object recognition using L-system fractals, *Pattern Recognition Letters*, v. 16, n. 1, p. 33-42.
- HOPFIELD, J. J.; D.W. TANK (1986). Computing with neural circuits: a model. *Science magazine*, v. 233, p. 625-633.

- HOPFIELD, J. (1982). Neural networks and physical systems with emergent collective computational properties. *Proceedings of the National Academy of Sciences of the USA*, v. 79, p. 2554-2588.
- HORAN, P.; UECKER, D.; ARIMOTO, A. (1990). Optical implementation of a second-order neural network discriminator model. *Japanese Journal of Applied Physics, Part 2: Letters*, v. 29, n. 7, p. 1328-1331.
- HOSOKAWA, M.; OMATU, S.; FUKUMI, M. (1989). New approach for pattern recognition by neural networks with scramblers. In: Intl. Joint Conference on Neural Networks. *Proceedings*, Washington.
- HSIN, H.-C.; LI, C.-C.; SUN, M.; SCLABASSI, R. J. (1995). An adaptive training algorithm for back-propagation neural networks. *IEEE Trans. on Systems, Man, and Cybernetics*, v. 25, n. 3, p. 512-514.
- HU, M. K. (1961). Pattern recognition by moment invariants. *Proceedings of the IRE*, v. 49, n. 9, p. 1428.
- HU, M. K. (1962). Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, v. 8, p. 179-187.
- HUBEL, D.H.; WIESEL, T.N. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physics*, v. 160, p. 106-154.
- HUSH, D.R.; B.G. HORNE (1993). Progress in Supervised Neural Networks. *IEEE Signal Processing Magazine*, January, p. 8-39.
- HWANG, J-N.; TSENG, Y-H. (1993). 3D motion estimation using single perspective sparse range data via surface reconstruction neural networks. In: Intl. Conference on Neural Networks. *Proceedings*, p. 1696-1701.

- IMAI, K.; K GOUHARA; Y. UCHIKAWA (1992). Neural Networks Applied to Recognition of CCD Camera Images where a 3-Digit Number is Stamped on the Surface of Chip Resistors. In: IAPR Workshop on Machine Vision Applications. Tokyo. Proceedings, p. 93-96.
- JAIN, A.K. (1989). *undamentals of Digital Image Processing*, Englewood Cliffs, Prentice-Hall.
- JONES, J.G.; THOMASAND, R.W.; EARWICKER, P.G.; ADDISON, S. (1993). Multiresolution statistical analysis of computer generated fractal imagery. *CVGIP: Graphical Models and Image Analysis*, v. 53, p. 349-363.
- KAK, S. (1994). New algoritms for training feedforward neural networks. *Pattern Recognition Letters*, v. 15, p. 295-298.
- KHAW, J.F.C.; LIM, B.S.; LIM, L.E.N. (1995). Optimal design of neural networks using the Taguchi method. *Neurocomputing*, v. 7.
- KHOTANZAD, A.; LU, J.H. (1989). Object recognition using a neural network and invariant Zernike features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Proceedings, p. 200-205.
- KHOTANZAD, A.; LU, J.H. (1990). Classification of invariant image representations using a neural network. *IEEE Trans. on Acoustic Speech and Signal Processing*, v. 38, p. 1028-1038.
- KHOTANZAD, A.; HONG, Y. H. (1990). Invariant image recognition by Zernike moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v. 12, p. 498-497.
- KIM, D-Y.; CHIEN, S-I.; SON, H. (1991). Multiclass 3-D aircraft identification and orientation estimation using multilayer feedforward neural network. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 758-764.

KOREIN, J. U.; ISH-SHALOM, J. (1987). Robotics, IBM Systems Journal, v. 26, n. 1.

KULKARNI, A. D.; YAP, A. C.; BYARS, P. (1990). Neural networks for invariant object recognition. In: Symposium of Applied Computing (SAC), Fayetteville, AR. Proceedings, p. 28-32.

KUROGI, S. (1993). Competitive neural network for affine invariant pattern recognition, In: Intl. Joint Conference on Neural Networks. Proceedings, p. 181-184.

LALANNE, P.; J. TABOURY; P. CHAVEL (1987). A proposed generalization of hopfield's algorithm. *Optics Communications*, v. 63, n. 1, p. 21-25.

LANGE, J.; VON DER MALSBURG, C. (1988). Neural network for position invariant pattern representation. In: Intl. Neural Network Society First Annual Meeting, Boston.

LAWRENCE, J. J. (1990). Untangling Neural Nets. *Dr. Dobbs's Journal*, april, p. 38-44.

LEE, J.C.; SHEU, B. J.; FANG, W.C.; CHELLAPPA, R. (1993). VLSI Neuroprocessors for Video Motion Detection. *IEEE Trans. on Neural Networks*, v. 4, n.2, p.178-193.

LEJEUNE, C.; SHENG, Y. (1993). Optoneural system for invariant pattern recognition, *Journal of Physics*, v. 71, n. 9-10, p. 405-409

LEUNG, W.F.; LEUNG, S.H.; LAU, W.H.; LUK, A. (1991). Fingerprint recognition using neural network, In: Workshop on Neural Networks for Signal Processing, Princeton, NJ. Proceedings.

LI, B.C.; SHEN, J. (1991). Fast computation of moment invariants. *Pattern Recognition*, v. 24, p. 807-813.

LI, C.; WU, C-H. (1991). A biologically motivated model for rotation, shift, scalling and distortion invariant pattern recognition. In: Intl. Joint Conf. on Neural Networks. Proceedings, v. 2, p. 929.

LI, Y. (1991). Applications of moment invariants to neurocomputing for pattern recognition. *Electronics Letters*, v. 27, n. 7, p. 587-588.

LI, W.; NASRABADI, N.M. (1990). Invariant object recognition based on a neural network of cascated RCE nets. Intl. Joint Conference on Neural Networks. Proceedings, p. 845-854.

LIPPMANN, R.P. (1987). An introduction to computing with neural nets. *IEEE Trans. Acoustic, Speech and Signal Processing*. April, p. 4-22.

LU, M-C.; LO, C-H.; DON, H-S. (1991). A neural network approach to 3-D object identification and pose estimation. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 2600-2605.

MARAR, J.F.; CARVALHO FILHO (1995). Classificação de assinaturas off-line baseadas em aproximadores universais de funções com extração de características através de Wavelets. In: II Simpósio Brasileiro de Redes Neurais. Anais. S. Carlos, SP.

MAREN, A.J.; C.T. HARSTON; R.M. PAP (1990). *Handbook of neural computing applications*. San Diego. Academic Press.

MARR, D. (1982). *Vision*. W. H. Freeman.

MARUNO, S.; IMAGAWA, T.; KOHDA, T.; SHIMEKI, Y. (1993). Object recognition system using temporal pattern recognition with quantizer neuron chip. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 1285-1288.

MASCARENHAS, N. D. A.; VELASCO, F. R. D. (1989). *Processamento digital de imagens*. 2a. Ed., I Escola Brasileiro-Argentina de Informática. Buenos Aires: Ed. Kapelusz.

MASTERS, T. (1994). *Signal and Image Processing with Neural Networks*. John Wiley.

- MASTERS, T. (1995). *Advanced Algorithms for Neural Networks*. John Wiley.
- MEECH, J. A.; KUMAR, S. (1994). A hypermanual on expert systems - Version 3.0. Ottawa, Natural Resources Canada - CANMET, 4800 eletronic pages.
- MCAULAY, A.D.; DANSET, P.; WICKER, D. (1990). Assembly line inspection using neural networks. In: Hybrid Image and Signal Processing. Proceedings, v. 1297, p. 339-349.
- MINNIX, J.I.; MCVEY, E.S.; I IGO, R.M. (1992). A multilayered self-organizing artificial neural network for invariant pattern recognition. *IEEE Trans. on Knowledge and Data Engineering*, v. 4, n. 2, p. 162-167.
- MORGAN, N.; BOULARD, H. (1995). Continuous Speech Recognition. *IEEE Signal Processing Magazine*, v. 12, n. 3.
- MORRIS, R.J.T.; L.D. RUBIN; H. TIRRI (1990). Neural network techniques for object orientation detection: solution by optimal feedforward network and learning vector quantization approaches. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v.12, n. 11, p. 1107-1114.
- MOYA, M. M.; FOGLER, R. J.; HOSTELLER, L. D. (1988). Back propagation for perspective-invariant pattern recognition in SAR imagery. In: Intl. Neural Network Society First Annual Meeting. Proceedings. Boston, MA.
- MUNDKUR, P.Y.; DESAI, U.B. (1991). Automatic target recognition using image and network decomposition. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 281-286.
- NAGAO, K.; GRIMSON, E. (1994). Object recognition by alignment using invariant projections of planar surfaces, MIT, MA, Tech. Report AIM-1463.

- NEWMAN, T.S.; JAIN, A.K. (1995). A survey of automated visual inspection. *Computer Vision and Image Understanding*, v. 61, n. 2, p. 231-262.
- OGATA, J.; SASE, M.; KOSUGI, Y. (1992). Neural network approach for attractive area extraction from video images. In: IAPR Workshop on Machine Vision and Applications. Proceedings. Tokyo - Japan p. 159-162.
- O'REILLY, R. C.; JOHNSON, M.H. (1994). Object recognition and sensitive periods: a computational analysis of visual imprinting, *Neural Computation*, v. 6, p. 357-389.
- PAIK, J.K.; J. C. BRAINLEAN; A.K. KATSAGGELOS (1992). An edge detection algorithm using multi-state adalines. *Pattern Recognition*, v. 25, n. 12, p. 1495-1504.
- PAL, N.R.; PAL, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, v. 26, n. 9, p. 1277-1294.
- PARKER, J. R. (1994). *Practical Computer Vision Using C*. Wiley.
- PEDRYCZ, W. (1990). Fuzzy sets in pattern recognition: methodology and methods. *Pattern Recognition*, v. 23, n. 1/2, p. 121-146.
- PEDRYCZ, W. (1990). Neural networks: concepts and architectures. *SBA Controle & Automação*, v. 4, n. 3.
- PERELMUTER, G.; CARRERA, E.V.; VELLASCO, M.; PACHECO, M.A. (1995). Reconhecimento de imagens bidimensionais utilizando redes neurais artificiais. In: VIII SIBGRAPI. Anais, p. 197-203.
- PERES, R.; HOCHSTEIN, S. (1994). Modeling perceptual learning with multiple interacting elements: a neural network model describing early visual perceptual learning. *Journal of Computational Neuroscience*, v. 1, n. 4, p. 323-338.

- POLANA, R. B. (1994). *Temporal texture and activity recognition*. Ph.D. Thesis, University of Rochester.
- PROKOP, R. J.; REEVES, A. P. (1992). A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, v. 54, n. 5, p. 438-460.
- RAJAPAKSE, J.; ACHARYA, R. (1990a). Medical image segmentation with MARA. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 965-972.
- RAJAPAKSE, J.; ACHARYA, R. (1990b). Hierarchical neural architecture for visual pattern recognition and reconstruction. *Proceedings of SPIE*, v. 1246, p. 288-296.
- RAK, S.J.; KOLODZY, P.J. (1988). Invariant object recognition with the adaptive resonance (ART) network. In: Intl. Neural Network Society First Annual Meeting. Proceedings.
- RAVEENDRAN, P.; JEGANNATHAN, S.; OMATU, S. (1993). New regular moment invariants to classify elongated and contracted images. In: Intl. Joint Conf. on Neural Networks. Proceedings, v. 3, p. 2089-2092.
- RAVICHANDRAN, A.; YEGNANARAYANA, B. (1991). A two-stage neural network for translation, rotation and size-invariant visual pattern recognition. In: Intl. Conference on Acoustics, Speech and Signal Processing. Proceedings, v. 4, p. 2393-2396.
- REDDY, G. N.; VAITHILINGHAM, S.; BEAN, W.C. (1993). Neural net based 2D-vision system for real-time applications. In: 3rd Workshop on Neural Networks. Proceedings, v. 1721 (SPIE), p. 345-348.
- REICHEL, F.; LOEFFLER, W.; GAERTNER, E.; GMBH, J. (1994). Using optical space-frequency analysis for real-time pattern recognition. *IEEE Micro*, December, p. 49-60.
- RICHARD, R.D.; LIPPMAN, R.P. (1991). Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation*, v. 1, p. 281-294.

- RIPLEY, B.D. (1994). Neural Networks and Related Methods for Classification. *Journal of the Royal Statistical Society B*, v. 56, n. 3, p. 409-456.
- ROGERS, S. K.; TROXEL, S. E.; RUCK, D. W.; KABRISKY, M. (1988). Classification of tactical targets with neural nets. In: Intl. Neural Network Society First Annual Meeting. Proceedings. Boston, MA.
- ROSENBLATT, F. (1962). *Principles of Neurodynamics*. Spartan Books.
- ROTHER, I.; VOSS, K.; S SSE, H.; ROTHER, J. (1994). A general method to determine invariants. Technical Report, Fakultät für Mathematik und Informatik, Jena, Germany.
- ROTHMAN, P. (1992). Syntactic Pattern Recognition. *AI Expert*, October, 1992.
- RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J. (1986). Learning internal representations by error propagation. In: D.E. Rumelhart e J.L. McClelland, Eds., *Parallel distributed processing: explorations in the microstructure of cognition*, v. 1, Cambridge, MA. MIT Press
- RYBAK, I.A.; GOLOVAN, A.V.; GUSAKOVA, V.I. (1991). Approach to invariant object recognition on gray-level images by exploiting neural network models. In: *Applications of Artificial Neural Networks II*. Proceedings of SPIE, v. 1469, p. 472-482.
- SCHALKOFF, R. (1992). *Pattern Recognition: statistical, structural and neural approaches*. Wiley.
- SCHMITT, M.; MATTIOLI, J. (1991). Shape recognition combining mathematical morphology and neural networks. In: *Applications of Artificial Neural Networks II*. Proceedings of SPIE, v. 1469, p. 392-403.
- SEPEDA FILHO, I.H.; STEMMER, M.R. (1994). Reconhecimento de peças utilizando redes neurais. In: 1. Congresso Brasileiro de Redes Neurais. Anais. Itajubá - MG.

- SHIH, F.Y.; MOH, J.; CHANG, F.-C. (1991). A new ART-based neural architecture for pattern classification and image enhancement without prior knowledge. *Pattern Recognition*, v. 25, n. 5, p. 533-542.
- SHIMA, T.; KIMURA, T.; KAMATANI, Y.; ITAKURA, T.; FUJITA, Y.; IIDA, T. (1994). Neuro chips with on-chip back-propagation and / or hebbian learning. *Toshiba's Selected Papers on Science and Technology*, v. 6, n.1.
- SIMPSON, P. K. (1990). *Artificial neural systems: foundations, paradigms, applications and implementations*. New York. Pergamon Press.
- SIMPSON JR., R. L. (1991). Computer vision, an overview. *IEEE Expert*, August, p. 11-15.
- SOLOMON, E. P.; BERG, L.R.; MARTIN, D.W.; VILLEE, C. (1993). *Biology*. Saunders College Publishing.
- SPIRKOVSKA, L.; REID, M.B. (1990a). Connectivity strategies for higher-order neural networks applied to pattern recognition. In: Intl. Joint Conference on Neural Networks. *Proceedings*, p. 21-26.
- SPIRKOVSKA, L.; REID, M.B. (1990b). An empirical comparison of ID3 and HONNs for distortion invariant object recognition. In: 2nd Intl. IEEE Conference on Tools for Artificial Intelligence. *Proceedings*. Herndon, VA, p. 577-582.
- SPIRKOVSKA, L.; REID, M.B. (1992). Robust position, scale, and rotation invariant object recognition using higher-order neural networks. *Pattern Recognition*, v. 25, n. 9, p. 975-985.
- SPIRKOVSKA, L.; REID, M.B. (1993). Coarse-coded higher-order neural networks for PSRI object recognition. *IEEE Trans. on Neural Networks*, v. 4, n. 2, p. 276-283.

- SPIRKOVSKA, L.; REID, M. B. (1994). Higher-order neural networks applied to 2D and 3D object recognition. *Machine Learning*, v.15, p. 169-199.
- SRINIVASA, N.; JOUANEH, M. (1992). A neural network model for invariant pattern recognition. *IEEE Trans. on Signal Processing*, v. 40, n. 6, p. 1595-1599.
- STAVROS, P.; LISBOA, P.J.G. (1992). Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers. *IEEE Trans. on Neural Networks*, v. 3, n.2, p. 241-251.
- SUGISAKA, M.; TESHNEHLAB, M. (1993). Fast pattern recognition by using moment invariants computation via artificial neural networks. *Control Theory and Advanced Technology*, v. 9, n. 4, p. 877-886.
- SULLIVAN, D. J.; FORMAN, A.V. Jr. (1989). Parallel algorithms for automatic target recognition using CO//2 laser radar images. In: Sixth Multidimensional Signal Processing Workshop. Pacific Grove, CA. Proceedings.
- SUGANTHAN, P.N.; TEOH, E.K.; MITAL, D.P. (1993). Programming Hopfield networks for object recognition. In: Intl. Conference on Systems, Man and Cybernetics. Proceedings, v. 3, p. 114-119.
- TAFTI, M. H. A.; NIKBAKHT, E. (1993). Neural networks and expert systems: new horizons in business and finance applications. *Information Managment & Computer Security*, v. 1, n. 1, p. 22-28.
- TAKANO, M.; KANAOKA, T.; SKRZYPEK, J.; TOMITA, S. (1994). A note on a higher-order neural network for distortion invariant pattern recognition. *Pattern Recognition Letters*, v. 15, p. 631-635.
- TEAGUE, M. R. (1980). Image analysis via the general theory of moments. *Journal of the Optical Society of America*, v. 70, n. 8, p. 920-930.

- TEH, C.; CHIN, R.C. (1988). On image analysis by the method of the moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 10, p. 496-513.
- TELFER, B.; CASASENT, D. (1991). Minimum-cost Ho-Kashyap associative processor for piecewise-hyperspherical classification. In: Intl. Joint Conference on Neural Networks. *Proceedings*, v. 2, p. 89-94.
- TING, C.; CHUANG, K-C. (1993). Adaptive algorithm for neocognitron to recognize analog images. *Neural Networks*, v. 6, n. 2, p. 285-299
- TOWELL, G.G.; SHAVLIK, J.W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, v. 70, p. 119-165.
- TRONCO, M.L.; CARVALHO, A.C.P.L.F.; PORTO, A.J.V. (1995). Sistema de reconhecimento de peças mecânicas para robô de montagem - abordagem utilizando rede neural com arquitetura baseada no modelo GSN. In: II Simpósio Brasileiro de Redes Neurais. *Anais*. S. Carlos, SP.
- TROXEL, S.E.; ROGERS, S.K.; KABRISKY, M. (1988). Use of neural networks in PSRI target recognition. In: Intl. Conference on Neural Networks. *Proceedings*.
- TSANG, P.W.M.; YUEN, P.C.; LAM, F.K. (1992). Recognition of occluded objects. *Pattern Recognition*, v. 25, n. 10, p. 1107-1117.
- VAN CAMP, D. (1992). Neurons for computers. *Scientific American*, vol. 267, n. 3, p.125-127.
- VARELLA, L.E.S.; PASSOS, E.P.L.; SANTOS, M.A.; ARAÚJO, R.L. (1994). Uma solução back-propagation invariante a escala, rotação e translação para o reconhecimento de caracteres. In: I Congresso Brasileiro de Redes Neurais. Itajubá, MG. *Anais*.
- VASCONCELOS, G.C. (1995). Redes Neurais e Reconhecimento de padrões. *Tutorial 2*. In: II Simpósio Brasileiro de Redes Neurais, S. Carlos, SP.

- VENTURA, J.A.; CHEN, J-M. (1993). Shape recognition for machined parts. In: 2nd Industrial Engineering Research Conference. Los Angeles, CA. Proceedings.
- VON DER MALSBERG, C. (1988). Object recognition in dynamical network architecture. *Neural Networks*, v. 1, n. 1, p. 145.
- WALKER, M.; HASLER, P.; AKERS, L. (1989). A CMOS neural network for pattern association. *IEEE Micro Magazine*, October, p. 68-74.
- WANG, J-Y.; COHEN, F. S. (1994). Part II: 3-D object recognition and shape estimation from image countours using B-spline, shape invariant matching, and neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, v. 16, n. 1, p. 13-23.
- WERBOS, P. J. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. dissertation, Harvard University, Cambridge, MA.
- WERBOS, P. J. (1994). *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. New York. Wiley .
- WIDROW, B.; WINTER, R.G.; BAXTER, R. A. (1988). Layered neural nets for pattern recognition. *IEEE Trans. Acoustic, Speech and Signal Processing*, vol. 36, p. 1109-1118.
- WIDROW, B.; WINTER, R. (1988). Neural nets for adaptative filtering and adaptative pattern recognition. *Computer*. March, p. 25-39.
- WINTER, R.; WIDROW, B. (1988). Madaline rule II: A training algorithm for neural networks. In: Intl. Neural Network Society First Annual Meeting. Proceedings. Boston, MA.
- WONG, W.-H.; SIU, W.-C.; LAM, K.-M. (1995). Generation of moment invariants and their uses for character recognition. *Pattern Recognition Letters*, v. 16, n. 2, p. 115-123.

- WU, C.H.; J.F. WANG; W.H. WU (1991). A shunting multilayer perceptron network for confusing/composite pattern recognition. *Pattern Recognition*, v. 24, n. 11, p. 1093-1103.
- YAZDI, N.; AHMADI, M.; SHRIDHAR, M. (1993). Multi-layer neural network architecture with external weight memory. In: 36th Midwest Symposium on Circuits and Systems. Proceedings. V. 2, p. 1288-1291.
- YANG, C.C.; MAREFAT, M.M. (1994). Spatial quantization errors in active vision inspection. In: Intl. Conference on Systems, Man, and Cybernetics. Proceedings. V. 1, p. 67-72.
- YI, J.; KUROGI, S.; MATSUOKA, K. (1991). Back-propagation learning of neural networks for translation invariant pattern recognition. *Systems and Computers in Japan*, v. 22, n. 14, p. 80-89.
- YOU, S. D.; FORD, G. E. (1993). Network model for invariant object recognition and rotation angle estimation. In: Intl. Joint Conference on Neural Networks. Proceedings, p. 2145-2148.
- YOU, S. D.; FORD, G. E. (1994). Network Model for Invariant Object Recognition. *Pattern Recognition Letters*, v. 15, p. 761-767.
- YU, F.T.S.; LI, Y. (1990). Applications of moment invariants to neural computing for pattern recognition. In: Hybrid Image and Signal Processing II. Orlando, FL. Proceedings. SPIE, v. 1297, p. 307-316.
- YU, F.T.; UANG, C-M.; YIN, S. (1993). Shift- and rotation-invariant interpattern hetero-association model. *Proceedings of SPIE*, v. 1959, p. 464-473.
- Y CEER, C.; OFLAZER, K. (1993). A rotation, scaling, and translation invariant pattern classification system. *Pattern Recognition*, v. 26, n. 5, p. 687-710.

- ZAKARIA, M. F.; VROOMEN, L.J.; ZSOMBOR-MURRAY, P. J. A.; VAN KESSEL, J. M.H.M. (1987). Fast algorithm for computation of moment invariants. *Pattern Recognition*, v. 20, p. 639-643.
- ZEKI, S. (1993). *A vision of the brain*. Blackwell Scientific Publications.
- ZEKI, S. (1992). Visual image in mind and brain. *Scientific American*, v. 267, n. 3, p. 24-33.
- ZHANG, W.; HASEGAWA, A.; ITOH, K.; ICHIOKA, Y. (1991). Error back propagation with minimum-entropy weights: a technique for better generalization of 2-D shift-invariants. In: Intl. Joint Conference on Neural Networks. *Proceedings*, p. 645-648.
- ZHANG, Y.; CHEN, Z. (1993). Invariant recognition by feedback network. In: Intl. Joint Conference on Neural Networks. *Proceedings*, p. 2185-2188.
- ZHANG, Y. J., (1995). Influence of segmentation over feature measurement. *Pattern Recognition Letters*, v. 16, n. 2, p. 201-206.
- ZHOU, Y.; CHELLAPPA, R.; VAID, A.; JENKINS, B.K. (1988). Image restoration using a neural network. *IEEE Trans. on Acoustics, Speech and Signal Processing*, v. 36, n.7, p.1141-1151.
- ZHOU, X.; KOCH, M.W.; ROBERTS, M.W. (1991). A selective attention neural network for invariant recognition of distorted objects. In: Intl. Joint Conference on Neural Networks. *Proceedings*. Part. 2, p. 942.
- ZHOU, X.; KOCH, M.W.; ROBERTS, M.W. (1991). Selective attention of high-order neural network for invariant object recognition. In: Intl. Joint Conference on Neural Networks. *Proceedings*. Part. 2, p. 937.
- ZURADA, J. M. (1992). *Introduction to Artificial Neural Systems*. St. Paul - MN, West Publishing Company.

APÊNDICE A

Performance da Filtragem de Ruído e Eliminação de Regiões de Área Pequena

A figura A-1 mostra, para um determinado objeto, a performance dos algoritmos de filtragem e eliminação de regiões conectadas com área menor que um determinado limiar. No presente trabalho, este valor limiar foi fixado em 100 pixels. Todos os objetos tratados neste trabalho apresentam área bem maior que este valor, o que justificou este limiar. A rotina de geração de ruído foi apresentada no capítulo 5.

Observando a figura A-1 vemos o método utilizado apresenta sucesso para um ruído entre 15 e 20 %, podendo ser admissível até cerca de 35%, dependendo do tipo de objeto. Para taxas de ruído acima de 35% a restauração da imagem do objeto degradada pelo ruído é deficiente, pelo fato que muitos pontos ruidosos são integrados ao objeto pela codificação de 8-vizinhos conectados dos pixels, além da inversão dos pixels pertencentes ao objeto.

Os valores dos atributos, derivados de imagens ruidosas, semelhantes à figura A-1, podem ser analisados nos gráficos apresentados na figura 15. Pelo fato que o ruído gerado sobre as imagens sintéticas tem o efeito de um canal com probabilidade X de inversão dos pixels, para um ruído de 100% haveria uma inversão de todos os pixels da imagem, ver figura A-1.

Figura A-1: Imagens de objetos degradadas por ruído (1ª. coluna), filtragem pela mediana (2ª. coluna) e eliminação de regiões conectadas de área menor que 100 pixels (3ª. coluna).

APÊNDICE B

Distribuição dos atributos em relação a transformações geométricas

Para observar a dispersão dos atributos em relação a transformações geométricas e ao ruído (ver figura 15) podemos levantar curvas de distribuição dos atributos em relação aos objetos. Assumindo, hipoteticamente (e levando em consideração o teorema do limite central), que a distribuição dos atributos seja normal, geramos curvas (equação B-1) para cada atributo e cada objeto, a partir da média (μ) e desvio padrão (σ) estimados a partir dos valores extraídos do banco de imagens de teste. Apesar do erro oriundo da assunções, o objetivo aqui é apenas ilustrar a separabilidade entre classes e explicar, parcialmente, a dificuldade dos classificadores em decidir pela classe correta. Quanto menor a abertura da curva, i.e., o desvio padrão, mais invariante o atributo. Quanto mais espaçadas as curvas, maior a separabilidade entre classes relativo àquele atributo. A seguir serão apresentadas curvas das distribuições dos atributos utilizados para os objetos apresentados em 24-a e 24-b (P_i , $i = 1, 2, \dots, 6$). As curvas foram normalizadas e ajustadas para melhor análise.

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty \quad (\text{B-1})$$

onde $-\infty < \mu < \infty$ e $\sigma > 0$.

B-1 Imagens sintéticas

a) Invariantes de Hu.

Os invariantes de momentos (equação 23) foram calculados para um dado conjunto de peças livre de ruídos. Os gráficos mostrados na figura B-1 ilustram as distribuições dos atributos pelos tipos de objetos.

Figura B-1.a: Distribuição de valores do atributo 1 (equação 23-a) para objetos de imagens sintéticas.

Figura B-1.b: Distribuição de valores do atributo 2 (equação 23-b) para objetos de imagens sintéticas.

Figura B-1.c: Distribuição de valores do atributo 3 (equação 23-c) para objetos de imagens sintéticas.

Figura B-1.d: Distribuição de valores do atributo 4 (equação 23-d) para objetos de imagens sintéticas.

Figura B-1.e: Distribuição de valores do atributo 5 (equação 23-e) para objetos de imagens sintéticas.

Figura B-1.f: Distribuição de valores do atributo 6 (equação 23-f) para objetos de imagens sintéticas.

Os dados destes gráficos, média, desvio padrão e coeficiente de variação dos atributos (equação 23) para todas as peças podem ser vistos na tabela 1.

Média	A1	A2	A3	A4	A5	A6
P1	0.4294	1.1319	2.6590	5.4812	9.9195	6.3076
P2	0.4839	1.5510	1.4759	2.6480	4.8567	3.5516
P3	0.4285	5.5605	6.4554	4.6691	10.4349	7.5736
P4	0.2645	1.1468	1.6954	2.1040	4.0110	2.6816
P5	0.5731	1.9673	1.8709	3.9307	6.8339	4.9156
P6	0.2464	0.6674	1.1076	1.6132	2.9737	1.9471
Média:	0.4043	2.0041	2.5440	3.4077	6.5049	4.4962

Tabela B.1.a: média dos valores dos IMs, equação 23, para objetos de imagens sintéticas.

Desv.Pad.	A1	A2	A3	A4	A5	A6
P1	0.00207	0.00589	0.01644	0.31327	0.61784	0.48957
P2	0.00846	0.00429	0.04243	0.02954	0.17244	0.12288
P3	0.00403	0.52065	1.34569	0.45606	0.96186	0.51612
P4	0.00684	0.01620	0.04196	0.02013	0.05300	0.02837
P5	0.00669	0.11041	0.01036	0.32658	0.49538	0.38216
P6	0.01513	0.03521	0.06546	0.08238	0.15430	0.09777

Tabela B-1.b: desvio padrão dos IMs, equação 23, para objetos de imagens sintéticas.

Coef.Var.	A1	A2	A3	A4	A5	A6
P1	0.5%	0.5%	0.6%	5.7%	6.2%	7.8%
P2	1.7%	0.3%	2.9%	1.1%	3.6%	3.5%
P3	0.9%	9.4%	20.8%	9.8%	9.2%	6.8%
P4	2.6%	1.4%	2.5%	1.0%	1.3%	1.1%
P5	1.5%	5.6%	0.6%	8.3%	7.2%	7.8%
P6	6.1%	5.3%	5.9%	5.1%	5.2%	5.0%
Média:	2.2%	3.7%	5.5%	5.2%	5.5%	5.3%

Tabela B-1.c: coeficiente de variação dos IMs, equação 23, para objetos de imagens sintéticas.

b) Novos invariantes de momentos.

Foram calculados, para as mesmas imagens utilizadas para a obtenção dos gráficos apresentados na figura B-1, os novos atributos, equação 33, derivados da técnica de Wong et al. (1995). A figura B-2 apresenta as distribuições dos valores para os objetos das imagens sintéticas.

Figura B-2.a: Distribuição de valores do atributo 1 (equação 33-a) para objetos de imagens sintéticas.

Figura B-2.b: Distribuição de valores do atributo 2 (equação 33-b) para objetos de imagens sintéticas.

Figura B-2.c: Distribuição de valores do atributo 3 (equação 33-c) para objetos de imagens sintéticas.

Figura B-2.d: Distribuição de valores do atributo 4 (equação 33-d) para objetos de imagens sintéticas.

Figura B-2.e: Distribuição de valores do atributo 5 (equação 33-e) para objetos de imagens sintéticas.

Figura B-2.f: Distribuição de valores do atributo 6 (equação 33-f) para objetos de imagens sintéticas.

Os dados destes gráficos, média, desvio padrão e coeficiente de variação dos atributos (equação 33) para todas as peças podem ser vistos na tabela 2.

Média	A1	A2	A3	A4	A5	A6
P1	5.0726	4.5836	4.8059	3.9610	5.0866	4.3721
P2	3.2474	3.0495	2.7808	2.8128	3.1655	3.0819
P3	5.4972	6.7617	6.0162	6.1355	6.9878	6.7726
P4	2.4339	2.5153	2.3788	3.1314	2.8919	2.8894
P5	4.1605	3.7288	3.9831	3.4140	4.6693	3.7099
P6	2.4054	1.8150	2.1799	1.7079	2.5193	1.8498
Média:	3.8028	3.7423	3.6908	3.5271	4.2201	3.7793

Tabela B-2.a: média dos valores dos IMs, equação 33, para objetos de imagens sintéticas.

Desv.Pad.	A1	A2	A3	A4	A5	A6
P1	0.47618	0.23663	0.71810	0.12859	0.30750	0.51801
P2	0.35540	1.02102	0.39967	0.08576	0.24657	0.34056
P3	0.85776	1.66541	0.66469	0.43976	1.57775	0.46142
P4	0.22465	0.38494	0.46494	0.39012	0.66082	0.26303
P5	0.57718	0.30284	0.77074	0.20913	0.56551	0.10236
P6	0.97872	0.29333	0.78889	0.36924	1.02740	0.40108

Tabela B-2.b: desvio padrão dos IMs, equação 33, para objetos de imagens sintéticas.

Coef.Var.	A1	A2	A3	A4	A5	A6
P1	9.4%	5.2%	14.9%	3.2%	6.0%	11.8%
P2	10.9%	33.5%	14.4%	3.0%	7.8%	11.1%
P3	15.6%	24.6%	11.0%	7.2%	22.6%	6.8%
P4	9.2%	15.3%	19.5%	12.5%	22.9%	9.1%
P5	13.9%	8.1%	19.4%	6.1%	12.1%	2.8%
P6	40.7%	16.2%	36.2%	21.6%	40.8%	21.7%
Média:	16.6%	17.1%	19.2%	8.9%	18.7%	10.5%

Tabela B-2.c: coeficiente de variação dos IMs, equação 33, para objetos de imagens sintéticas.

Para avaliar a invariância dos atributos em relação ao ruído, um objeto específico foi tomado como exemplo e a taxa de ruído crescente foi aplicada sobre este objeto. A figura 15 apresenta resultados para imagens sintéticas (equação 23). Imagens de um objeto para uma taxa crescente de ruído pode ser visto no apêndice A. Para os testes foi utilizado uma filtragem do tipo mediana com janela de 5x5 pixels. Outros tipos de ruído e os respectivos filtros que melhor se adaptam foram discutidos anteriormente (Costa, 1995).

B-2 Imagens reais

a) Invariantes de Hu.

Os invariantes de momentos (equação 23) foram calculados para as imagens reais e os gráficos mostrados na figura B-3 ilustram as distribuições dos atributos pelos tipos de objetos.

Figura B-3.a: curvas de distribuição do atributo 1 (eq. 23-a) para as classes de objeto (P_i , $i=1,2,\dots,9$) Figura B-3.b: curvas de distribuição do atributo 2 (eq. 23-b) para as classes de objeto (P_i , $i=1,2,\dots,9$) Figura B-3.c: curvas de distribuição do atributo 3 (eq. 23-c) para as classes de objeto (P_i , $i=1,2,\dots,9$)

Figura B-3.d: curvas de distribuição do atributo 4 (eq. 23-d) para as classes de objeto (P_i , $i=1,2,\dots,9$) Figura B-3.e: curvas de distribuição do atributo 5 (eq. 23-e) para as classes de objeto (P_i , $i=1,2,\dots,9$) Figura B-3.f: curvas de distribuição do atributo 6 (eq. 23-f) para as classes de objeto (P_i , $i=1,2,\dots,9$)

Os dados destes gráficos, média, desvio padrão e coeficiente de variação dos atributos (equação 23) para todas as peças podem ser vistos na tabela 3.

Médias	A1	A2	A3	A4	A5	A6
P1	0.5518	2.6592	5.8781	6.1303	12.4540	7.5826
P2	0.3049	1.9776	4.4097	4.0587	8.6035	5.2719
P3	0.6159	1.7471	5.7404	5.4589	11.2099	6.4594
P4	0.2916	0.6602	1.7120	1.8993	3.7051	2.2303
P5	0.6638	2.6491	4.9329	4.5554	9.7576	6.1024
P6	0.5828	2.4570	5.8715	5.8689	11.9820	7.3613
P7	0.6238	2.5194	2.6029	4.2275	7.7748	5.6686
P8	0.6373	1.7225	5.6917	6.2272	12.3386	7.2911
P9	0.2715	0.6005	2.6869	2.5687	5.1965	2.8706
Média:	0.5048	1.8881	4.3918	4.5550	9.2247	5.6487

Tabela 3-a: média dos valores dos IMs, equação 23, para objetos de imagens reais.

Desv.Padr.	A1	A2	A3	A4	A5	A6
P1	0.01111	0.04594	0.54269	0.69809	1.33532	0.75620
P2	0.00365	0.04061	0.31799	0.46180	0.87703	0.54974
P3	0.04021	0.26696	0.57680	0.35846	0.90369	0.52955
P4	0.07543	0.17984	0.19607	0.26395	0.49384	0.35326
P5	0.00184	0.07742	0.26588	0.10699	0.40599	0.28173
P6	0.00263	0.03724	0.45344	0.15813	0.40715	0.34322
P7	0.01022	0.35500	0.03460	0.55587	0.91272	0.53252
P8	0.04067	0.21293	0.46295	0.54965	1.12441	0.75274
P9	0.03191	0.07267	0.15423	0.12016	0.25705	0.14062

Tabela 3-b: desvio padrão dos IMs, equação 23, para objetos de imagens reais.

Coef.Var.	A1	A2	A3	A4	A5	A6
P1	2.0%	1.7%	9.2%	11.4%	10.7%	10.0%
P2	1.2%	2.1%	7.2%	11.4%	10.2%	10.4%
P3	6.5%	15.3%	10.0%	6.6%	8.1%	8.2%
P4	25.9%	27.2%	11.5%	13.9%	13.3%	15.8%
P5	0.3%	2.9%	5.4%	2.3%	4.2%	4.6%
P6	0.5%	1.5%	7.7%	2.7%	3.4%	4.7%
P7	1.6%	14.1%	1.3%	13.1%	11.7%	9.4%
P8	6.4%	12.4%	8.1%	8.8%	9.1%	10.3%
P9	11.8%	12.1%	5.7%	4.7%	4.9%	4.9%
Média:	6.2%	9.9%	7.4%	8.3%	8.4%	8.7%

Tabela 3-c: coeficiente de variação dos IMs, equação 23, para objetos de imagens reais.

b) Novos invariantes de momentos

Foram calculados, para as mesmas imagens utilizadas para a obtenção dos gráficos apresentados na figura B-3, os novos atributos, equação 33, derivados da técnica de Wong et al. (1995). A figura B-4 apresenta as distribuições dos valores para os objetos das imagens reais.

Figura B-4.a: Distribuição de valores do atributo 1 (equação 33-a) para objetos de imagens reais.

Figura B-4.b: Distribuição de valores do atributo 2 (equação 33-b) para objetos de imagens reais.

Figura B-4.c: Distribuição de valores do atributo 3 (equação 33-c) para objetos de imagens reais.

Figura B-4.d: Distribuição de valores do atributo 4 (equação 33-d) para objetos de imagens reais.

Figura B-4.e: Distribuição de valores do atributo 1 (equação 33-e) para objetos de imagens reais.

Figura B-4.f: Distribuição de valores do atributo 6 (equação 33-f) para objetos de imagens reais.

Os dados destes gráficos, média, desvio padrão e coeficiente de variação dos atributos (equação 33) para todas as peças podem ser vistos na tabela B-4.

Médias	A1	A2	A3	A4	A5	A6
P1	6.5777	7.0222	6.7460	7.0401	7.4893	7.7145
P2	4.2547	4.6085	5.1824	5.0834	5.3832	5.7176
P3	6.0351	6.3779	6.9210	6.8743	7.1513	7.8049
P4	2.3620	2.2102	2.7748	2.4836	2.8184	2.7817
P5	5.2927	5.9373	5.6108	6.2354	6.4262	6.9875
P6	6.5264	6.6041	6.8015	7.0434	7.2039	7.8986
P7	4.8917	4.3521	4.2903	4.2748	4.9362	4.9520
P8	6.9613	6.8133	7.0832	7.1469	7.4572	7.9803
P9	3.3642	2.9760	3.4115	2.7718	3.1913	3.3526
Média:	5.1407	5.2113	5.4246	5.4393	5.7841	6.1322

Tabela B-4.a: média dos valores dos IMs, equação 33, para objetos de imagens reais.

Desv.Padr.	A1	A2	A3	A4	A5	A6
P1	0.80657	0.64011	0.67101	0.59865	0.90575	0.67864
P2	0.65580	0.36260	0.55999	0.54133	0.58844	0.42485
P3	0.61091	0.65055	0.63055	0.61257	0.42935	0.56295
P4	0.70847	0.44700	0.42801	0.66020	0.49863	0.71664
P5	0.61581	0.66619	0.21921	0.58352	0.51491	0.25959
P6	0.46788	0.25806	0.37526	0.46781	0.36438	0.57673
P7	0.76528	0.50099	0.22823	0.32386	0.39402	0.66103
P8	0.67583	0.72439	0.49861	0.36846	0.52099	0.59964
P9	0.75907	0.21207	0.46071	0.42828	0.46565	0.43668

Tabela B-4.b: desvio padrão dos IMs, equação 33, para objetos de imagens reais.

Coef.Var.	A1	A2	A3	A4	A5	A6
P1	12.3%	9.1%	9.9%	8.5%	12.1%	8.8%
P2	15.4%	7.9%	10.8%	10.6%	10.9%	7.4%
P3	10.1%	10.2%	9.1%	8.9%	6.0%	7.2%
P4	30.0%	20.2%	15.4%	26.6%	17.7%	25.8%
P5	11.6%	11.2%	3.9%	9.4%	8.0%	3.7%
P6	7.2%	3.9%	5.5%	6.6%	5.1%	7.3%
P7	15.6%	11.5%	5.3%	7.6%	8.0%	13.3%
P8	9.7%	10.6%	7.0%	5.2%	7.0%	7.5%
P9	22.6%	7.1%	13.5%	15.5%	14.6%	13.0%
Média:	14.9%	10.2%	9.0%	11.0%	9.9%	10.5%

Tabela B-4.c: coeficiente de variação dos IMs, equação 33, para objetos de imagens reais.