

**Universidade de São Paulo
Instituto de Física de São Carlos**

**AVALIAÇÃO DA QUALIDADE DE PLACAS DE MADEIRA
ATRAVÉS DE UM SISTEMA DE INFERÊNCIA NEBULOSO
BASEADO EM REDES ADAPTATIVAS**

CELSO APARECIDO DE FRANÇA

Tese apresentada ao Instituto de Física de
São Carlos, da Universidade de São Paulo,
para a obtenção do título de Doutor
em Ciências: Física Aplicada sub-área
Física Computacional.

Orientador: Prof.Dr. Adilson Gonzaga

ou

São Carlos
1999

USP/IFSC/SBI



8-2-001282

IFSC-USP SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

de França, Celso Aparecido
Avaliação da qualidade de placas de madeira através
de um sistema de inferência nebuloso baseado em
redes adaptativas. São Carlos, 1999.
208p.
Tese(Doutorado) – Instituto de Física de São
Carlos, 1999.
Orientador: Prof. Dr. Adilson Gonzaga
1. Redes neurais. 2. Lógica fuzzy. 3. Lógica
nebulosa
I. Título

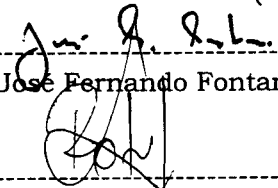


MEMBROS DA COMISSÃO JULGADORA DA TESE DE DOUTORADO DE CELSO APARECIDO FRANÇA APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 12 DE AGOSTO DE 1999.

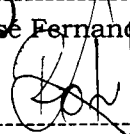
COMISSÃO JULGADORA:



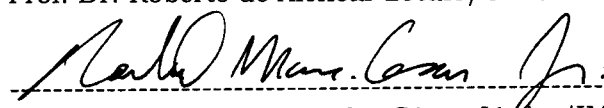
Prof. Dr. Adilson Gonzaga/EESC-USP



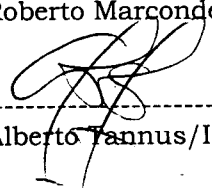
Prof. Dr. José Fernando Fontanari/IFSC-USP



Prof. Dr. Roberto de Alencar Lotufo/UNICAMP



Prof. Dr. Roberto Marcondes César Júnior/IME-USP



Prof. Dr. Alberto Yannus/IFSC-USP

“Na medida em que as leis da matemática se referem à realidade, elas não estão certas e, na medida em que estão certas, elas não dizem respeito à realidade” Albert Einstein

Dedico esta tese a Deus, o grande
criador do universo

AGRADECIMENTOS

Agradeço às pessoas que de uma forma especial ajudaram na realização deste trabalho:

- Ao Prof. Dr. Adilson Gonzaga, por sua excelente orientação e correção do texto que tornou possível a realização deste trabalho.
- À Profa. Annie F.F. Slaets, por sua inestimável ajuda.
- Ao CNPQ, pela bolsa de estudo concedida.
- Aos professores doutores Evandro e Maria Stela, pela amizade.
- À Wladerez, que sempre atendeu-me com boa vontade.
- À Evelina, Anderson, Mônaco e José Luís, pela amizade, pelos “cafés” e pelas discussões.
- Aos vários amigos (as) do predinho:
Cazuza, Carlos, Ciro, Cristina, Elias, Fabiano, Fábio, Hilário, Jefferson, João, Juliana, Ju2, Luciana, Luís Antônio (Potó), Maurício (Penico), Nagui, Orides, Vivian, Renata, Ricardo, Taira, Vitória.
pelos churrascos, futebol e bagunças nesses anos todos.
- E, claro, à minha mãe por tudo.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	iii
LISTA DE ALGORITMOS	iv
RESUMO	v
ABSTRACT	vi
1 – INTRODUÇÃO	1
1.1 - Organização da tese	2
2 - RECONHECIMENTO DE PADRÕES	4
2.1 - Reconhecimento de Padrões	4
2.2 - Vetor de Características	5
2.3 - Características radiométricas de uma imagem	7
2.4 – Características de textura	7
2.4.1 - Métodos estatísticos	9
2.4.2 - Métodos estruturais	14
2.4.2.1 - Seleção das primitivas enfatizando bordas e esqueletos	15
2.4.2.2 - Primitivas em termos de região	16
2.4.3 - Métodos espectrais	17
2.5 - Classificação	18
2.5.1 - Classificação do vizinho mais próximo	18
2.5.2- Classificador linear	19
2.5.3- Classificador Bayesiano	20
2.5.3.1 – Regra de Bayes para o mínimo erro	21
2.5.3.2 – Regra de Bayes para a máxima verossimilhança	22
2.6 - Conclusão	24
3 – LÓGICA “FUZZY”	25
3.1 - Introdução	25
3.2 - Operações com conjuntos “fuzzy”	28
3.3 - Normas triangular (t-norms)	31
3.4 - Co-normas triangulares (t-conorms)	32
3.5 - Implicação “fuzzy”	33
3.6 - Raciocínio “fuzzy”	35
3.7 - Modelos “fuzzy” (Sistema de Inferência “Fuzzy”)	37
3.7.1 - Saída do modelo “fuzzy”	37
3.7.1.1 - Criação da regra “fuzzy”	38
3.7.1.2 - “Fuzzificação”	38
3.7.1.3 - Combinação das entradas nas regras “fuzzy”	40

3.7.1.4 - Parte consequente de cada regra	40
3.7.1.5 - Combinação de todas as partes consequentes	40
3.7.1.6 - “Desfuzzificação”	40
3.7.2 - Alguns modelos “fuzzy”	41
3.7.2.1 - Modelo “fuzzy” de Mamdani	42
3.7.2.2 - Modelo “fuzzy” de Larsen	43
3.7.3.3 - Modelo de Sugeno	44
3.7.3.4 - Modelo “fuzzy” de Tsukamoto	46
3.8 - Conclusão	46
4 – REDES NEURAIS	48
4.1 - Introdução	48
4.2 - Neurônio	49
4.3 - Perceptrons	50
4.3.1 - Aprendizagem dos perceptrons	52
4.4 - Classificação de redes neurais	53
4.4.1 - Operação das unidades de processamento	53
4.4.2 - Topologia das redes	54
4.4.3 - Relação entre entrada e saída de dados	55
4.4.4 - Algoritmo de aprendizagem	56
4.4.5 - Taxonomia das redes neurais	57
4.5 - Escolha do modelo de rede neural	58
4.6 - Perceptrons de múltiplas camadas	59
4.6.1 - Classificador	61
4.6.2 - Dificuldade no aprendizado	63
4.7 - Conclusão	64
5 – REDE NEURO-FUZZY	66
5.1 - Introdução	66
5.2 - Neurônio “fuzzy”	67
5.2.1 - Neurônio AND	68
5.2.2 - Neurônio “fuzzy” mínimo (MIN FN)	69
5.2.3 - Neurônio OR	69
5.2.4 - Neurônio “fuzzy” máximo (MAX FN)	69
5.2.5 - Neurônio “fuzzy” competitivo	70
5.3 - Camadas das redes neuro-fuzzy	70
5.4 - Construção das regras SE-ENTÃO de uma rede neuro-fuzzy	71
5.5 - Arquitetura ANFIS	72
5.6 - Classificação neuro-fuzzy	75
5.7 - Conclusão	76
6 – INSPEÇÃO VISUAL EM PLACAS DE MADEIRAS	78
6.1 – Introdução	78
6.2 – Método de classificação de madeiras	79
6.3 – Madeiras	82

6.3.1 – Estrutura geral da madeira	83
6.3.2 – Madeira de coníferas	84
6.3 – Definição das classes	87
6.4 – Conclusão	89
7 – MÉTODOS PARA CLASSIFICAÇÃO	92
7.1 – Introdução	92
7.2 – Metodologia	94
7.2.1 – Aquisição de imagem	94
7.2.2 – Pré-processamento	95
7.2.3 – Extração de características	98
7.2.4 – Definição da rede neural a ser empregada	101
7.2.5 – Utilização de lógica “fuzzy”	104
7.2.5.1 – Resultados obtidos com os sistemas propostos	106
7.3 – Outros Classificadores	107
7.3.1 – Classificador neural	108
7.3.2 – Classificador Estatístico	109
7.3.3 – K-vizinhos mais próximos	113
7.4 – Conclusão	115
8 – RESULTADOS E CONCLUSÕES	116
8.1 – Introdução	116
8.2 – Resultados	117
8.3 – Comparações entre os métodos	117
8.4 – Contribuições	121
8.5 – Conclusão final	121
8.6 – Sugestões para futuros trabalhos	122
ANEXO 1 – Tabela de classificação do conjunto de 199 placas	124
ANEXO 2 – Conteúdo do CD-ROM	129
REFERÊNCIAS BIBLIOGRÁFICAS	130

LISTA DE FIGURAS

Fig. 2.1 - Esquema geral de reconhecimento de padrões	4
Fig. 2.2 - Escolha das características de entrada	6
Fig. 2.3 - Modelo binário local	12
Fig. 2.4 - Código cadeia	15
Fig. 2.5 - Exemplo da aplicação do código cadeia	15
Fig. 2.6 - Figura aproximada por polígonos	16
Fig. 2.7 - Classificação pelo método do vizinho mais próximo	18
Fig. 2.8 - Exemplo de classificador linear	20
Fig. 2.1 - Classificador Bayesiano	23
Fig. 3.1 - Função de pertinência para temperatura ambiente	26
Fig. 3.2 - Funções de pertinência para velocidade	27
Fig. 3.3 - Operação de união e intersecção nos conjuntos “fuzzy”	29
Fig. 3.4 - Exemplos de implicação “fuzzy”	34
Fig. 3.5 - Funções utilizadas para “fuzzificação”	39
Fig. 3.6 - Estratégias de “desfuzzificação”	41
Fig. 3.7 - Raciocínio “fuzzy” de Mamdani	43
Fig. 3.8 - Modelo “fuzzy” de Larsen	44
Fig. 3.9 - Modelo “fuzzy” de Sugeno	45
Fig. 3.10 - Modelo “fuzzy” de Tsukamoto	46
Fig. 4.1 - Neurônio	49
Fig. 4.2 - Modelo básico do neurônio	51
Fig. 4.3 - Função Heaviside	51
Fig. 4.4 - Problema OR exclusivo	53
Fig. 4.5 - Estruturas de redes neurais	55
Fig. 4.6 - Perceptrons de múltiplas camadas	60
Fig. 4.7 - Função Sigmoidal	60
Fig. 4.8 - Rede de duas camadas e classificação do espaço	61
Fig. 4.9 - Região de decisão	62
Fig. 4.10 - Combinações das regiões na segunda camada	62
Fig. 4.11 - Função de energia de uma rede neural	63
Fig. 5.1 - Neurônio “fuzzy”	68
Fig. 5.2 - Neurônio AND	68
Fig. 5.3 - Neurônio OR	69
Fig. 5.4 - Funções de pertinências para duas entradas da rede neuro-fuzzy	71
Fig. 5.5 - Estrutura da rede neuro-fuzzy do exemplo	72
Fig. 5.6 - Raciocínio “fuzzy” de Tsukamoto para três entradas	73
Fig. 5.7 - Estrutura ANFIS do raciocínio de Tsukamoto	74
Fig. 5.8 - Partição “fuzzy” do espaço	76
Fig. 6.1 - Seção transversal do tronco de uma árvore	83
Fig. 6.2 - Desenho do corte transversal do tronco de uma árvore	85
Fig. 6.3 - Plano de orientação da madeira mostrando superfície transversal (X) , superfície tangencial (T) e superfície radial (R)	86
Fig. 6.4 - Métodos de corte da madeira	87
Fig. 6.5 - Placa A	88
Fig. 6.6 - Placa B	88
Fig. 6.7 - Placa C	88

Fig. 6.8 - Placa D	89
Fig. 6.9 - Placa S	89
Fig. 7.1 - Sistema mecânico para classificação das placas	92
Fig. 7.2 - Sistema montado I	93
Fig. 7.3 - Sistema montado II	93
Fig. 7.4 - Sistema montado III	93
Fig. 7.5 - Sistema montado IV	93
Fig. 7.6 - Sistema em blocos para a classificação das placas	94
Fig. 7.7 - Placa sem enriquecimento	96
Fig. 7.8 - Placa após enriquecimento	96
Fig. 7.9 - Placa antes da binarização	97
Fig. 7.10 - Placa após a binarização	97
Fig. 7.11 - Placa dividida em 16 regiões	98
Fig. 7.12 - Tipos de arquitetura híbrida	101
Fig. 7.13 - Regras “fuzzy” MIN-MAX	104
Fig. 7.14 - Variáveis linguísticas para cada classe	106
Fig. 7.15 - Diagrama em blocos do classificador neural	108
Fig. 7.16 - Distribuição dos valores de média	109
Fig. 7.17 - Distribuição espacial dos valores de média	109
Fig. 7.18 - Distribuição dos valores de variância	110
Fig. 7.19 - Distribuição espacial dos valores de variância	110
Fig. 7.20 - Distribuição dos valores de entropia	110
Fig. 7.21 - Distribuição espacial dos valores de entropia	110
Fig. 7.22 - Distribuição dos valores de IDM	110
Fig. 7.23 - Distribuição espacial dos valores de IDM	110
Fig. 7.24 - Distribuição dos valores de “pixels” escuros	111
Fig. 7.25 - Distribuição espacial dos valores de “pixels” escuros	111

LISTA DE TABELAS

Tabela 3.1 - Normas triangulares básicas (t-norms)	32
Tabela 3.2 - Co-normas triangulares básicas (t-conorms)	32
Tabela 3.3 - Operadores de implicação “fuzzy”	34
Tabela 4.1 - Taxonomia de redes neurais	57
Tabela 7.1 - Número de placas coincidentes entre os inspetores humanos e as características das redes neurais combinadas com lógica “fuzzy” de um total de 199 placas, utilizando diretamente as saídas das redes neurais nas regras “fuzzy”	106
Tabela 7.2 - Número de placas coincidentes entre os inspetores humanos e as características das redes neurais combinadas com lógica “fuzzy” de um total de 199 placas, com o uso de três variáveis linguísticas nas saídas das redes neurais.	106
Tabela 7.3 - Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador de redes neurais com lógica “fuzzy” (IDM e número de “pixels” escuros), utilizando diretamente as saídas das redes neurais nas regras “fuzzy”.	107
Tabela 7.4 - Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador de redes neurais com lógica “fuzzy” (IDM e número de “pixels” escuros), utilizando a etapa de fuzzificação nas saídas das redes neurais.	107
Tabela 7.5 - Número de placas coincidentes entre os inspetores humanos as características da rede neural de um total de 199 placas	109
Tabela 7.6 - Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e a rede neural (IDM e número de “pixels” escuros).	109
Tabela 7.7 - Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador estatístico.	112
Tabela 7.8 - Número de placas coincidentes entre os inspetores humanos e as características do classificador KNN da placa inteira de um total de 199 placas.	114
Tabela 7.9 - Número de placas coincidentes entre os inspetores humanos e as características do classificador KNN da placa com janelas de um total de 199 placas.	114
Tabela 7.9 - Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador KNN da placa toda (variância e número de “pixels” escuros).	114
Tabela 8.1 - Frequências de ocorrências de classes nos métodos analisados	118
Tabela 8.2 - Comparação entre os métodos estudados	119

LISTA DE ALGORITMOS

Alg. 7.1 – Método da transformação linear nos níveis de cinza para 64 níveis	96
Alg. 7.2 – Procura do “threshold” usando a seleção interativa	97
Alg. 7.3 – Cálculo do histograma	99
Alg. 7.4 – Cálculo da média	99
Alg. 7.5 – Cálculo da variância	100
Alg. 7.6 – Cálculo da entropia	100
Alg. 7.7 – Cálculo do momento da diferença inversa (IDM)	100
Alg. 7.8 – Cálculo do número de “pixel” escuro	100
Alg. 7.9 – Treinamento e teste de redes neurais MLP com “backpropagation”	103
Alg. 7.10 – Raciocínio de decisão “fuzzy”	105
Alg. 7.11 – Classificador estatístico	112
Alg. 7.12 – Classificador do K-vizinhos mais próximos	113

RESUMO

A inspeção visual automática é uma tarefa importante para a produtividade industrial. Ela pode ser aplicada em controle de qualidade para substituir operadores humanos em trabalhos perigosos ou repetitivos.

O estágio de classificação em controle de qualidade da produção industrial é frequentemente baseado no conhecimento humano. Portanto, torna-se importante alimentar um sistema visual automático com dados nebulosos ou ambíguos. Um sistema “neuro-fuzzy” é uma forma adequada de implementar isto.

O trabalho contribui na área tecnológica de inspeção visual com o desenvolvimento de uma nova abordagem para avaliação da qualidade de placas de madeira utilizadas na fabricação de lápis. Outra contribuição foi a divisão do vetor de características, fazendo com que cada característica específica seja tratada em uma rede neural própria.

O método é baseado em duas redes neurais, cada uma tratando com apenas uma característica de entrada. Os resultados das redes neurais são combinados através de lógica nebulosa (“fuzzy”) fornecendo um sistema com maior poder discriminante do que aqueles que utilizam métodos tradicionais.

O sistema se caracteriza por ser ágil, repetitivo, com um padrão de classificação definido e por possuir baixo custo.

ABSTRACT

Automatic visual inspection is an important task for industrial productivity. It could be applied for quality control or for replacing manual work under dangerous or repetitive activity.

The classification stage in quality control of the industrial production is often based on the human knowledge. It seems, therefore, to be a great concern to supply an automated visual inspection system with fuzzy or ambiguous data. The Neuro-Fuzzy system is a good way to do this.

The objective of this work is to develop a new approach for the classification of wooden plates used in the pencil production. This new method is based on two neural networks, each one working with just an input feature. The results of neural networks are combined through fuzzy logic giving the system a greater discriminating power than those that use traditional methods.

The proposed method is characterized by being agile, repetitive, with a defined classification pattern and having low cost.

1 - INTRODUÇÃO

A automação industrial se faz cada vez mais presente na produção, com o objetivo de obter uma melhor qualidade nos produtos. Durante um processo, os produtos são inspecionados visualmente em várias etapas por operadores humanos.

Como essas etapas são repetitivas, as falhas acontecem devido ao cansaço, humor e às condições físicas e emocionais das pessoas envolvidas no processo.

Os métodos tradicionais de inspeção automática, por sua vez, são pouco flexíveis e devido ao grande volume de computação exigido, tornam-se inviáveis para aplicações em linha de montagem industrial.

O objetivo deste trabalho, portanto, é conceber um sistema de inspeção visual automático mais flexível levando-se em conta o fator humano e compatível com a velocidade requerida em um processo de produção na indústria. Para isso, será pesquisada uma arquitetura de rede neural artificial e, também, procedimentos baseados em um sistema de inferência nebuloso (“fuzzy”)¹, como meio de aproximar da flexibilidade humana em processos de classificação.

¹ Como o termo “fuzzy” é amplamente utilizado, mesmo em artigos em português, faremos seu uso “entre aspas” durante o desenvolvimento dos próximos capítulos.

A aplicação deste trabalho será dirigida para a classificação da placas de madeiras, quanto à qualidade, utilizadas para confecção de lápis levando-se em consideração a velocidade da inspeção, além do desempenho do sistema.

1.1 – Organização da tese

Esta tese é dividida em sete capítulos, além desta introdução. Os capítulos iniciais (2, 3, 4 e 5) expõe alguns conceitos que são as bases teóricas utilizadas para o desenvolvimento do método proposto para a classificação visual das placas de madeira.

O capítulo 2 trata do reconhecimento de padrões onde são abordados os métodos de extração de características de textura e os classificadores tradicionais.

O capítulo 3 discute a lógica “fuzzy” e seus principais modelos de inferência, apresentando um breve resumo de alguns conceitos relevantes na teoria dos conjuntos “fuzzy”.

O capítulo 4 trata as redes neurais e suas classificações quanto a operação de suas unidades de processamento, a topologia, a relação entre entrada e saída de dados e ao algoritmo de aprendizagem. Também explica o motivo da escolha da rede de “perceptrons” de múltiplas camadas (MLP) que é utilizada na tese.

O capítulo 5 apresenta as redes neuro-fuzzy através dos neurônios “fuzzy” e da construção das regras SE-ENTÃO (IF...THEN...).

No capítulo 6 são discutidas algumas publicações relacionadas à inspeção visual de placas de madeiras. Neste capítulo são definidas as classes das placas baseadas nas propriedades estruturais da madeira.

O capítulo 7 descreve os métodos de classificação estudados nesta tese (estatístico, K-NN, rede neural e o sistema neuro-fuzzy). São apresentados os algoritmos utilizados para cada método, além dos algoritmos de pré-processamento e de extração de características das imagens das placas.

No capítulo 8 são mostrados os resultados alcançados com os métodos de classificação estudados e são descritas as conclusões finais, a contribuição da tese e os trabalhos futuros.

2 – RECONHECIMENTO DE PADRÕES

2.1 - Introdução

O termo reconhecimento de padrões abrange uma larga faixa de problemas de processamento de informações de grandes significados práticos, de reconhecimento de voz e classificação de caracteres manuais até detecção de falha em máquinas e diagnósticos médicos [Bishop, 1995].

O processo de reconhecimento de padrões é visualizado como uma seqüência de passos: (i) aquisição de dados, (ii) seleção de características e (iii) procedimentos de classificação, conforme mostra a figura 2.1 [Pedrycz, 1990].

A *aquisição de dados* do ambiente para o reconhecimento de padrões visuais, é realizada através de sensores que transformam a energia luminosa de uma cena em sinais elétricos, e a partir destes uma

cena digitalizada pode ser montada

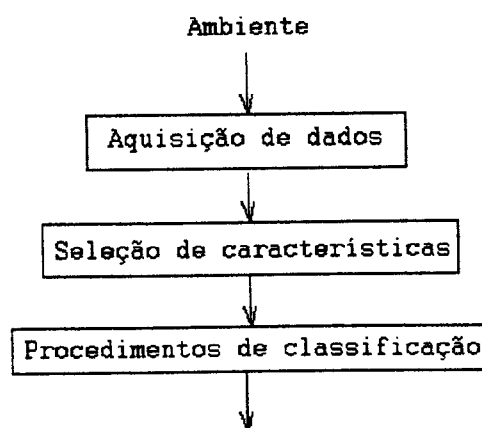


Figura 2. 1 - Esquema geral de reconhecimento de padrões

num “buffer” da memória. Este passo é muito importante, pois o desempenho de qualquer programa de análise de cena está intimamente limitado pela natureza e qualidade dos dados de entrada [Duda et al, 1979]. Uma cena bem iluminada é necessária, pois a luz ambiente geralmente não é aceitável porque ela pode resultar em imagens de baixo contraste, sombras e outros dados errôneos.

A *seleção de características* procura por aquelas que melhor definam um modelo de entrada [Beale and Jackson, 1990]. Algumas são naturais sendo definidas pela aparência visual de uma imagem enquanto outras são chamadas artificiais porque resultam de manipulação específicas de uma imagem [Pratt, 1991]. Como características naturais tem-se luminância de uma região de “pixels” e escala de cinza de regiões de textura. Para características artificiais cita-se a amplitude de histograma e o espectro de frequência espacial.

Finalmente o *procedimentos de classificação* decide a que classe pertence um dado vetor de características extraídas da entrada.

2.2 - Vetor de características

Raramente a classificação é realizada usando uma simples medida, ou característica, do modelo de entrada. Usualmente, várias medidas são necessárias para ser capaz de distinguir adequadamente entradas que pertencem a diferentes categorias (ou classes) [Beale and Jackson, 1990]. O vetor de características reúne todas as medidas utilizadas num classificador e o número de elementos do vetor (n) cria um espaço de características n -dimensional.

A escolha das características é fundamental para se obter um bom (ou mau) classificador. A figura 2.2(a) mostra boas características, das quais pode-se definir um limite de decisão (ou função discriminante) entre duas classes. Como exemplo, se fosse necessário separar uma classe que possuísse modelos de passarela e outra de pessoas que frequentam "spas", duas boas características seriam altura e peso. Na figura 2.2(b) é difícil definir um classificador por causa das escolhas ruins das características; no exemplo dado se fossem escolhidas as características de idade e raça, seria difícil separar as classes.

Uma vez escolhidas as características, o problema de reconhecimento de padrões reside em como obter essa função discriminante para mapear o espaço de características.

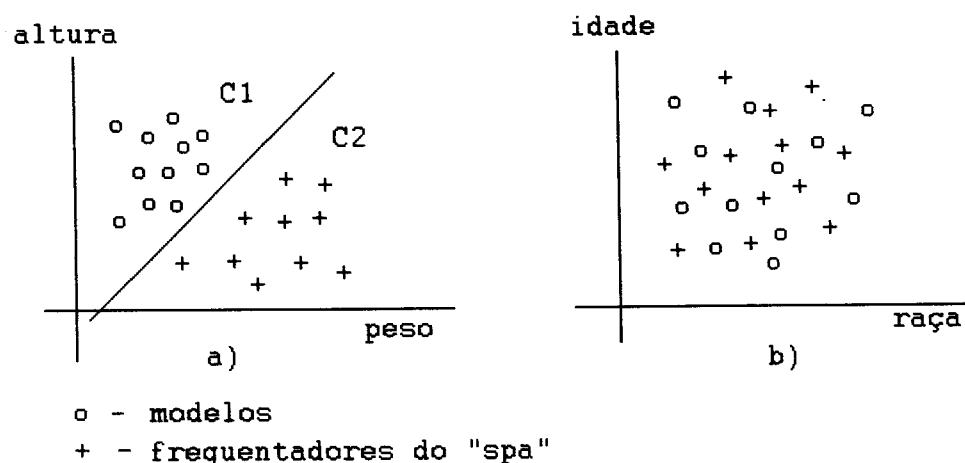


Figura 2. 2 - Escolha das características de entrada. (a) características linearmente separáveis, (b) características mal escolhidas.

2.3 - Características radiométricas de uma imagem

A característica básica de amplitude de uma imagem é o valor de seu “pixel” em análise. Esta medida pode ser feita em um ponto específico ou sobre uma vizinhança (média), o que traz como vantagem a redução de ruído. Seja uma imagem $F(i,j)$, sendo (i,j) a coordenada do ponto central de uma janela $W \times W$. A imagem da média ($M(i,j)$) é dada por:

$$M(i, j) = \frac{1}{W^2} \sum_{m=-W/2}^{W/2} \sum_{n=-W/2}^{W/2} F(i+m, j+n)$$

A mediana dos “pixels” em uma vizinhança $W \times W$ é definida como sendo a amplitude do “pixel” da janela $W \times W$ no qual metade dos “pixels” são menores ou iguais a esta amplitude, e metade são iguais ou maiores [Pratt, 1991].

Outra característica importante da imagem é seu desvio padrão (S), ou como é também conhecido dispersão de imagem. O desvio padrão pode ser calculado por:

$$S(i, j) = \frac{1}{W^2} \left[\sum_{m=-W/2}^{W/2} \sum_{n=-W/2}^{W/2} [F(i+m, j+n) - M(i, j)]^2 \right]^{1/2}$$

2.4 – Características de textura

Textura é uma das mais importantes características usadas para identificar objetos ou regiões de interesse em uma imagem [Haralick et al, 1973]. Em uma

classificação ou segmentação, é importante selecionar um conjunto de propriedades a ser usado para identificar duas ou mais regiões de textura [Tamura et al, 1978]. Propriedades textural são definidas para uma região ou sub-imagem, não para um ponto.

A textura de uma imagem é descrita pelas primitivas que compõe a imagem e pela dependência espacial (ou interação) entre as primitivas. Uma primitiva tonal pode ser descrita em termos da média dos tons de cinza, ou dos valores de tons máximo e mínimo de uma região [Haralick, 1979].

Tom e textura não são conceitos independentes, eles sempre existirão em uma imagem, embora algumas vezes uma propriedade pode dominar a outra e há uma tendência de se falar neste caso somente em tom ou somente em textura [Haralick, 1979]. Quando uma área de imagem tem pequena variação de primitiva tonal, a propriedade dominante desta área é o tom; caso contrário se houver uma grande variação de primitiva tonal, a propriedade dominante será a textura.

Os três métodos usados em processamento para descrever textura de regiões são: método estatístico, método estrutural e método espectral [Gonzalez & Wintz, 1987].

O modelo estatístico descreve textura por regras estatísticas que governam a distribuição e a relação dos níveis de cinza [Ballard & Brown, 1982]. O modelo estrutural considera as primitivas que formam um padrão repetitivo e descreve tais padrões em termos das regras que o geram [Ballard & Brown, 1982]. As técnicas espectrais são baseadas no espectro de Fourier e são usadas para detectar o período de uma imagem pela identificação de alta energia (ou picos estreitos no espectro de frequência)[Gonzalez & Wintz, 1987].

2.4.1 - Métodos estatísticos

Muitas texturas não tem uma regularidade geométrica, exibindo variação que não pode ser descrita por sua forma, mas pode ser descrita por modelos estatísticos [Ballard & Brown, 1982]. Esse método trabalha melhor com as texturas naturais, uma vez que essas texturas possuem variações nas formas geométricas que compõe a imagem.

Além das características de amplitude (média e mediana) para medir textura, Haralick [Haralick, 1979] cita: textura de bordas, elementos estruturais e matriz de dependência espacial dos níveis de cinza (SGLDM).

Textura de bordas considera a textura sendo a quantidade de bordas existente em uma área. Uma borda local é uma área pequena na imagem onde os níveis de cinza mudam rapidamente [Ballard & Brown, 1982]. Para detectar essas bordas são usados operadores de bordas conhecidos, tais como: Sobel, Prewitt, Roberts, Laplaciano entre outros.

Elemento estrutural é definido como uma região de células constituindo uma forma específica, tais como: linhas ou quadrados, então é gerada uma nova imagem binária pela erosão do elemento estrutural com a imagem original [Haralick, 1979]. A característica textural pode ser obtida pela contagem dos “pixels” de valor igual a 1. Elemento estrutural está ligado à matemática morfológica pelo uso da função de erosão. A matemática morfológica está sendo largamente utilizada em análise de imagens como uma aproximação para alisamento, segmentação de imagens, detecção de bordas, afinamento, análise de formas e codificação de imagens [Lee & Wong, 1996].

A *matriz de dependência espacial do nível de cinza (SGLDM)* é o método estatístico mais utilizado. Consiste da construção de matrizes contando o número de ocorrência de pares de “pixels” com um dado nível de cinza em um dado deslocamento [Chen et al, 1995]. Em outras palavras, o nível de cinza co-ocorrente pode ser especificado em uma matriz de frequências relativas P_{ij} com a qual duas células vizinhas separadas pela distância d e com uma orientação “ θ ” ocorrem na imagem, uma com nível de cinza i e outra com nível de cinza j [Haralick, 1979]. Haralick [Haralick et al, 1973] sugere 28 (vinte e oito) características de textura que podem ser extraídas da matriz de dependência espacial dos níveis de cinza. Entre essas características encontram-se:

Segundo momento angular, que indica o quão uniforme é a imagem:

$$f_1 = \sum_i \sum_j \{p(i,j)\}^2$$

Contraste, indica a quantidade de variações locais presente na imagem:

$$f_2 = \sum_{n=0}^{Nq-1} n^2 \left\{ \sum_i \sum_{\substack{j \\ |i-j|=n}}^{Nq} p(i,j) \right\}$$

Entropia, mede a não uniformidade da imagem:

$$f_3 = - \sum_i \sum_j p(i,j) \cdot \log(p(i,j))$$

Momento da diferença inversa, medida de homogeneidade da imagem:

$$f_5 = \sum_i \sum_j \frac{1}{1+(i-j)^2} \cdot p(i,j)$$

A *seqüência do nível de cinza* caracteriza textura grossa como tendo muitos “pixels” com tom constante na seqüência e textura fina com poucos “pixels” com nível constante na seqüência [Haralick, 1979].

Ojala [Ojala et al, 1996] realiza um estudo comparativo de medidas de textura considerando as seguintes medidas: método da diferença do nível de cinza, medida de textura de Law e medidas de covariância de centro simétrico e modelo binário local.

No *método da diferença do nível de cinza*, o conjunto de características é dado por: DIFFX e DIFFY, que são histograma das diferenças absolutas dos níveis de cinza entre “pixels” vizinhos computados na direção horizontal e vertical, respectivamente. DIFF2 acumula a diferença absoluta nas direções vertical e horizontal e DIFF4 acumula a diferença nas quatro direções principais.

A *medida de textura de Law* é gerada pelos seguintes conjuntos de vetores.

$$L5 = [1 \ 4 \ 6 \ 4 \ 1]$$

$$E5 = [-1 \ -2 \ 0 \ 2 \ 1]$$

$$S5 = [-1 \ 0 \ 2 \ 0 \ -1]$$

$$W5 = [-1 \ 2 \ 0 \ -2 \ 1]$$

$$R5 = [1 \ -4 \ 6 \ -4 \ 1]$$

Sendo que os mnemônicos significam: L-nível (level), E-borda (edge), S-ponto (spot), W-onda (wave) e R-ondulação (ripple). Através das convoluções desses vetores são geradas 25 (vinte e cinco) matrizes bi-dimensionais (L5L5, E5L5....). Essas matrizes são aplicadas na imagem gerando 25 (vinte e cinco) novas imagens. Após isso, essas imagens são substituídas pela medida de energia de textura (TEM) para cada “pixel”, isso é realizado utilizando uma região de 15x15 ao redor de cada “pixel”, através do seguinte filtro:

$$Inova(x, y) = \sum_{i=-7}^7 \sum_{j=-7}^7 |Ivelha(x+1, y+j)|$$

As novas imagens são rotuladas por L5L5T, E5L5T...(onde T indica a medida de energia de textura (TEM) e são normalizadas por L5L5T. Após isso, as características podem ser combinadas para serem independentes da direção. Por exemplo L5E5T é sensível a bordas verticais e E5L5T é sensível a bordas horizontais, a combinação E5L5TR = E5L5T + L5E5T se torna sensível ao conteúdo da borda.

A *medida de covariância* de centro simétrico somente computa a autocovariância local ou a auto-correlação dos valores dos "pixels" centrados simetricamente em vizinhança de tamanho adequado [Ojala et al, 1996].

No *modelo binário local* em uma vizinhança 3x3 o "threshold" é dado pelo valor do "pixel" central da janela (fig. 2.3(b)). Os valores resultante são multiplicados pelo peso dos "pixels" correspondente (fig. 2.3(c)). O resultado (fig. 2.3(d)) é somado, obtendo-se o valor da unidade de textura.

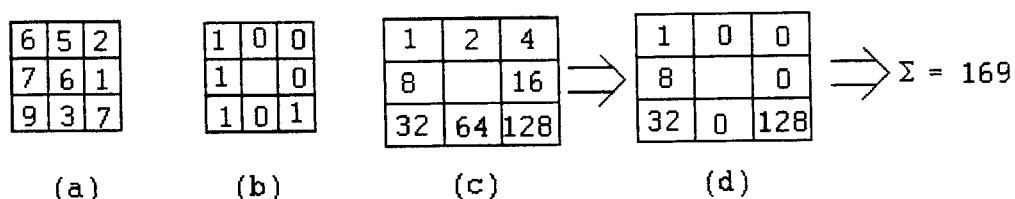


Figura 2. 3 - Modelo binário local. (a) representa uma distribuição de níveis em uma imagem; (b) é a mesma distribuição depois de ser aplicado o "threshold" pelo valor do "pixel" do meio; (c) é o peso de cada "pixel" na janela; (d) é a multiplicação do resultado do "threshold" pelo seu peso correspondente, o qual somando, fornece o valor do Modelo binário local.

Ojala {Ojala et al, 1996} conclui que o método da diferença de nível de cinza tem um melhor desempenho discriminante sobre as texturas analisadas, enquanto o método de Law foi o mais pobre.

Os dados estatísticos podem ser computados indiretamente através do histograma da imagem. A distribuição de probabilidade de 1ª. ordem da amplitude do histograma pode ser definido como [Pratt, 1991]:

$$P(b) = P\{I(i, j) = b\}$$

onde $P(b)$ é a quantidade de “pixels” no nível b ($0 \leq b \leq L-1$).

O histograma estimado de 1ª. ordem é dado por [Pratt, 1991]:

$$P(b) = \frac{N(b)}{M}$$

onde M é o número total de “pixels” numa janela e $N(b)$ é o quantidade de “pixels” de nível b na mesma janela.

O histograma estimado de 2ª. ordem, também denominado de matriz de co-ocorrência, é definido por:

$$P(a, b) = P\{F(i, j) = a, I(m, n) = b\}$$

onde a e b representa valores da amplitude dos “pixels” e o histograma de 2ª ordem estimado será:

$$P(a, b) = \frac{N(a, b)}{M}$$

onde M representa o número total de “pixels” numa janela e $N(a, b)$ denota o número de ocorrência na qual $F(i, j) = a$ e $F(m, n) = b$.

Através do histograma várias características podem ser extraídas, entre elas:

Histograma de 1º ordem**média:**

$$S_M = \bar{b} = \sum_{b=0}^{L-1} b.P(b)$$

desvio padrão:

$$S_D = \left[\sum_{b=0}^{L-1} (b - \bar{b})^2.P(b) \right]^{1/2}$$

energia:

$$S_N = \sum_{b=0}^{L-1} [P(b)]^2$$

entropia:

$$S_E = - \sum_{b=0}^{L-1} P(b) \cdot \log_2 [P(b)]$$

Histograma de 2º ordem**auto correlação:**

$$S_A = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} a.b.P(a,b)$$

covariância:

$$S_C = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} (a - \bar{a}) \cdot (b - \bar{b}) \cdot P(a,b)$$

onde:

$$\bar{a} = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} a.P(a,b) \quad \bar{b} = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} b.P(a,b)$$

energia:

$$S_G = \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} [P(a,b)]^2$$

entropia:

$$S_T = - \sum_{a=0}^{L-1} \sum_{b=0}^{L-1} P(a,b) \cdot \log_2 [P(a,b)]$$

2.4.2 - Métodos Estruturais

A seleção de primitivas dos métodos estruturais pode ser agrupada em dois tipos, os que enfatizam as bordas e os que enfatizam a região [Fu, 1974].

2.4.2.1- Seleção de primitivas enfatizando bordas e esqueletos

O conjunto de primitivas mais comumente usadas para descrever bordas e esqueletos é o código cadeia [Fu, 1974]. O código cadeia consiste de segmentos de linhas que devem pertencer a uma grade fixa com um conjunto fixo de orientação possíveis [Ballard & Brown, 1982]. A grade de orientação a ser usada pode ser 4 ou 8 conectada (fig. 2.4). Cada segmento de linha da imagem é assinalado com dígito referente à orientação (fig. 2.5).

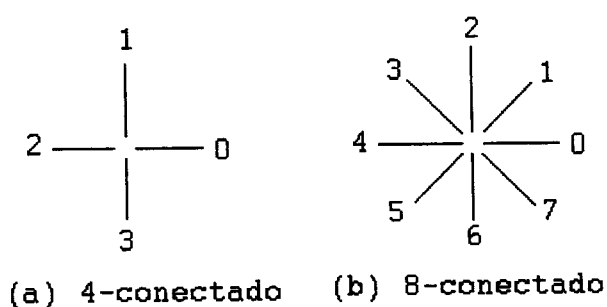


Figura 2. 4 - Código cadeia. (a) orientação para 4-conectado. (b) orientação para 8-conectado.

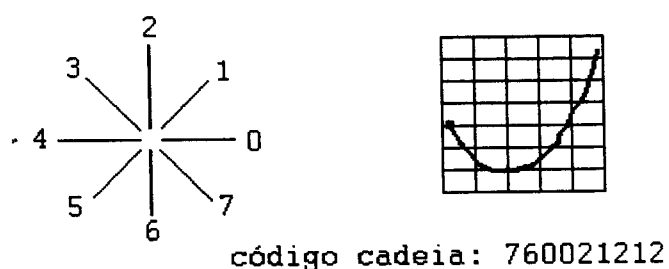


Figura 2. 5 - Exemplo da aplicação do código cadeia.

Esse código é independente de posição e sua derivada é útil porque o torna invariante à rotação de borda [Ballard & Brown, 1982]. Através do código pode-se calcular a área de uma figura fechada através do seguinte algoritmo:

Computa o primeiro ponto (x,y) da figura:

- 1) área = 0
pos_y = y
- 2) para cada elemento do código cadeia:
 - se 0 -> área = área - pos_y
 - se 1 -> pos_y = pos_y + 1
 - se 2 -> área = área + pos_y
 - se 3 -> pos_y = pos_y - 1

2.4.2.2- Primitivas em termos de região

Neste caso as primitivas básicas são semi-planos no campo de observação. Pode ser mostrado que qualquer figura pode ser aproximada por um polígono arbitrário o qual pode ser expresso como a união de um número finito de polígonos convexos [Fu, 1974]. Cada polígono convexo pode ser representado como a intersecção de um número finito de meio planos. A definição de uma ordem

adequada do polígono convexo compõe o polígono arbitrário. É possível, então, determinar o conjunto mínimo de polígonos que através da união representa o polígono arbitrário (fig. 2.6).

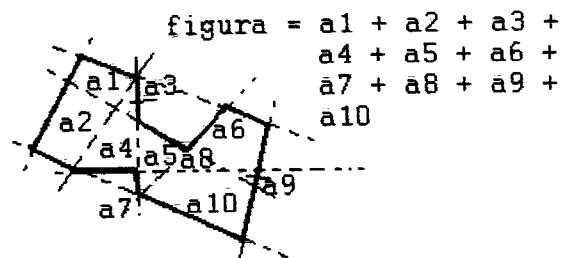


Figura 2. 6 - Figura aproximada por polígonos.

2.4.3 - Métodos Espectrais

O espectro de potência de Fourier bi-dimensional de uma imagem de textura frequentemente revela a periodicidade e a direção da textura [Chen et al, 1995]. Texturas grossas tendem a gerar componentes de baixas frequências enquanto as texturas finas geram componentes de altas frequências [Pratt, 1991]. A transformada de Fourier é definida como [Schalkoff, 1989]:

$$F(u, v) = \iint \exp[-j.2\pi(ux - vy)].f(x, y).dx.dy$$

onde $F(u,v)$ é a forma geral. O espectro de potência é dado por:

$$|F(u, v)|^2 = F(u, v).F^*(u, v)$$

onde F^* denota o complexo conjugado.

A transformada de Fourier geralmente trabalha bem com texturas que possuem uma periodicidade dominante e seu desempenho cai quando a periodicidade enfraquece [Chen et al, 1995].

Um dos métodos muito usados para extração de característica para textura é o filtro multi-canal utilizando o filtro de Gabor [Picher et al, 1996]. Esta técnica extrai as características da imagem com um subconjunto selecionado de um banco de filtro de Gabor fixo. O principal problema ocorre caso a frequência central do filtro de Gabor selecionado não casar com qualquer harmônica principal, produzindo somente informações ruidosas [Picher et al, 1996].

2.5 - Classificação

As técnicas de classificação de padrões podem ser divididas em duas categorias: as numéricas e as não numéricas. Técnicas numéricas incluem medidas determinísticas e estatísticas, as quais podem ser consideradas como uma medida feita no espaço geométrico do modelo.

Técnicas não numéricas são aquelas que possuem processamento de símbolos e são tratadas com métodos dos conjuntos fuzzy [Beale & Jackson, 1990]. Os métodos de classificação do vizinho mais próximo, o classificador linear e o classificador Bayesiano são os mais utilizados na prática, sendo que os dois primeiros são métodos determinísticos e o último estatístico.

2.5.1 - Classificação do vizinho mais próximo

É baseado na distância que um modelo desconhecido possui em relação às classes definidas em um espaço de modelos (fig. 2.7).

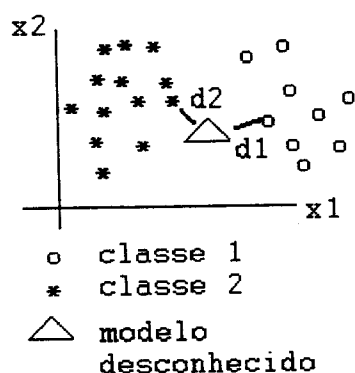


Figura 2. 7 - Classificação pelo método do vizinho mais próximo. d1 representa a distância do modelo desconhecido à classe 1 e d2 a distância até à classe 2.

Essa distância pode ser calculada de várias formas, entre as quais:

$$\text{Distância de Hamming: } H = \sum_n (|x_i - y_i|)$$

$$\text{Distância Euclidiana: } d(x, y) = [\sum_n (x_i - y_i)^2]^{1/2}$$

$$\text{Distância City-Block: } D_{cb} = \sum_n |x_i - y_i|$$

$$\text{Distância Quadrado: } D_{sq} = \max |x_i - y_i|$$

2.5.2 - Classificador Linear

O classificador linear divide o espaço de modelo através de funções discriminantes. Se um novo modelo for apresentado ao classificador a função discriminante se encarrega de classificar o modelo em uma ou outra classe. O limite de decisão entre classes no espaço de amostra define a função discriminante $f(x)$ da seguinte forma [Beale & Jackson, 1990];

$$f_x = \sum_n W_i \cdot X_i$$

onde: X_i = i-ésima componente do vetor de entrada

W_i = i-ésima componente do vetor peso

A figura 2.8 mostra um exemplo de classificador linear:

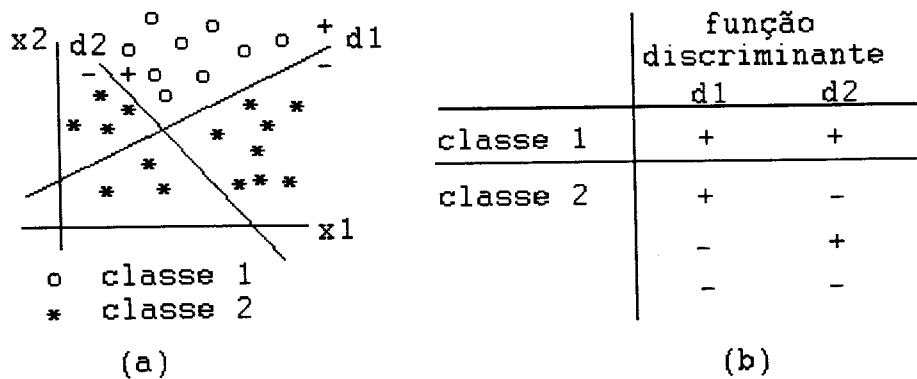


Figura 2. 8 - Exemplo de classificador linear. (a) duas funções discriminantes para separar as duas classes. (b) resultado da classificação através das funções discriminantes.

No exemplo da figura 2.8(a) existem duas classes e para separá-las foram necessárias duas funções discriminantes. A figura 2.8(b) mostra o resultado da classificação com base na saída da função discriminante.

2.5.3 - Classificador Bayesiano

A teoria de decisão de Bayes é uma aproximação estatística fundamental para o problema de reconhecimento de padrões. Esta aproximação é baseada na suposição de que o problema de decisão seja colocado em termos probabilísticos, e que todos os valores de probabilidades relevantes sejam conhecidos [Duda & Hart, 1973].

O problema de classificação é formulado a partir de w_i ($i=1,2,\dots,n$) classes, com a probabilidade de ocorrência (probabilidade a priori) dadas por $P(w_i)$, onde $0 \leq P(w_i) \leq 1$ e $\sum P(w_i) = 1$.

A probabilidade condicional de um modelo pertencer à classe w_i , dado o fato do modelo possuir os valores do vetor de características \mathbf{x} (probabilidade a posteriori), será:

$$P(w_i / \mathbf{x}) = \frac{p(\mathbf{x} / w_i) \cdot P(w_i)}{\sum_i^n p(\mathbf{x} / w_j) \cdot P(w_j)} = \frac{p(\mathbf{x} / w_i) \cdot P(w_i)}{p(\mathbf{x})} \quad (2.1)$$

onde $p(\mathbf{x}/w_i)$ é a função densidade probabilidade condicional do modelo ter os valores do vetor de características \mathbf{x} , dado que o mesmo pertença à classe w_i .

Algumas vezes é mais prudente um sistema de automação que não decida sobre todas as classes [Rauber, 1997]. Isso significa que deve ser introduzida a opção de rejeição através de uma nova classe (w_0). Um exemplo para o uso da opção de rejeição é o diagnóstico médico. Frequentemente é melhor rejeitar a classificação de uma doença (lesão de mama) e deixar a decisão final para um médico especialista que pode realizar outros exames a fim de conseguir um diagnóstico mais preciso.

2.5.3.1 – Regra de Bayes para o mínimo erro

Define-se o erro de um dado vetor de característica \mathbf{x} , se for decidido que o mesmo pertence à classe w_j embora a probabilidade posterior de outra classe w_i ser maior do que a probabilidade da classe w_j . A probabilidade do erro é dado por [Duda & Hart, 1973]:

$$P(\text{erro} / \mathbf{x}) = P(w_i / \mathbf{x}) \quad \text{se for decidido que } \mathbf{x} \text{ pertence à classe } w_j$$

Minimizar a probabilidade de erro para cada vetor de característica \mathbf{x} é equivalente a escolher a classe que possui a maior probabilidade posteriori de \mathbf{x} [Rauber, 1997].

Portanto a decisão de bayes para o mínimo erro sem rejeição é dado por:

$$d(\mathbf{x}) = w_i \quad \text{se } P(w_i / \mathbf{x}) > P(w_j / \mathbf{x}) \quad \text{p/ todo } j \neq i \quad (2.2)$$

Para permitir a rejeição deve ser introduzido um “threshold” de rejeição. Por simplicidade é definido como uma constante para todas as classes λ_r . Neste caso λ_r é o mínimo das probabilidades a posteriori que é necessário para realizar a decisão [Rauber, 1997]. Então a regra de Bayes para o mínimo erro com a opção de rejeição torna-se:

$$\begin{aligned} d(\mathbf{x}) &= w_i \quad \text{se } P(w_i / \mathbf{x}) > P(w_j / \mathbf{x}) \quad \text{p/ todo } j \neq i \text{ E } P(w_i / \mathbf{x}) > \lambda_r \\ d(\mathbf{x}) &= w_0 \quad \text{caso contrário} \end{aligned} \quad (2.3)$$

2.5.3.2 – Regra de Bayes para a máxima verossimilhança

Substituindo a equação 2.2 na equação 2.1 é fácil verificar que a regra de decisão não é afetada pela divisão do fator de normalização ($p(\mathbf{x})$). Portanto a regra de decisão de Bayes pode ser reescrita como:

$$d(\mathbf{x}) = w_i \quad \text{se} \quad p(\mathbf{x}/w_i).P(w_i) > p(\mathbf{x}/w_j).P(w_j) \quad \text{p/ todo } j \neq i$$

ou:

$$d(\mathbf{x}) = w_i \quad \text{se} \quad l(\mathbf{x}) = \frac{p(\mathbf{x}/w_i)}{p(\mathbf{x}/w_j)} = \frac{P(w_j)}{P(w_i)} \quad \text{p/ todo } j \neq i \quad (2.4)$$

onde $l(\mathbf{x})$ é chamado de razão de verossimilhança.

Quando as probabilidades a priori são iguais ($P(w_i)=P(w_j)$), ou quando não se tem informação útil sobre as probabilidades a priori, tem-se um caso particular onde a decisão é realizada pelo maior valor das funções densidades condicionais $p(\mathbf{x}/w_i)$.

Essa decisão é chamada de decisão de Bayes para a máxima verossimilhança:

$$d(\mathbf{x}) = w_i \quad \text{se} \quad p(\mathbf{x}/w_i) > p(\mathbf{x}/w_j) \quad \text{p/ todo } j \neq i \quad (2.5)$$

A representação do classificador Bayesiano é ilustrada na figura 2.9.

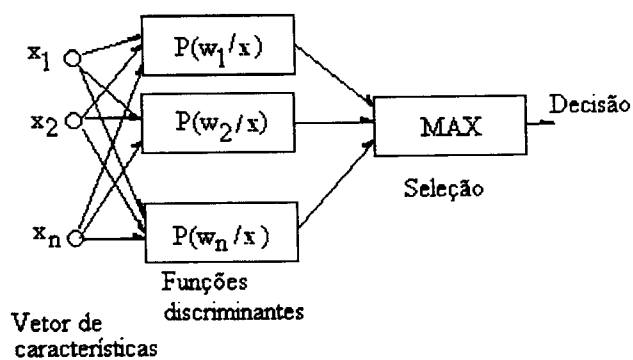


Figura 2.9 - Classificador Bayesiano.

Os classificadores baseados em lógica “fuzzy” e redes neurais serão apresentados nos capítulos sub-sequentes.

2.6 - Conclusão

Em qualquer sistema de reconhecimento de padrão, as características de entrada desempenham um papel fundamental. Não se consegue uma boa classificação (ou segmentação) de uma imagem se as características não forem bem escolhidas.

Existem na literatura vários métodos de extração de características. Esses métodos podem ser classificados em estatísticos, estruturais e espectrais. O uso de um método ou outro, assim como de uma técnica ou outra, está condicionado ao tipo de imagem em análise, ao tempo de processamento e à complexidade computacional.

De uma forma geral os métodos estruturais trabalham melhor quando existe uma regularidade de repetição dos arranjos das primitivas. A textura da pele dos répteis podem ser manipulados por esse método [Ballard & Brown, 1982].

Os métodos estatísticos e espectrais são os mais utilizados, pois trabalham bem com texturas naturais. A informação estatística é, em geral, mais rápida de se obter do que a informação estrutural [Fu, 1974].

3 – LÓGICA “FUZZY”

3.1- Introdução

O conjunto “fuzzy” foi proposto inicialmente por Zadeh [Zadeh, 1965] para ser uma extensão dos conjuntos clássicos. A utilidade desse conjunto reside na sua habilidade de modelar dados incertos ou ambíguos encontrados frequentemente na vida real [Pal & Mitra, 1992].

A principal diferença entre a proposição clássica e a “fuzzy” está na faixa de seus valores verdades [Klir & Yuan, 1995]. Na teoria clássica um elemento pertence ou não a um determinado conjunto. Na teoria “fuzzy” o elemento pode pertencer, não pertencer ou estar parcialmente presente em um determinado conjunto.

A cada elemento do conjunto “fuzzy” é dado um grau de pertinência. Esse valor de pertinência pertence a uma faixa de 0 (elemento não pertencente ao conjunto) até 1 (elemento totalmente pertencente ao conjunto). Uma função de pertinência é a relação entre os valores de um elemento e seu grau de pertinência em um conjunto.

A figura 3.1 mostra uma função de pertinência trapezoidal, considerando a temperatura ambiente em torno de 23° C - 27° C. Nesta faixa a função de pertinência

é igual a 1 indicando que a temperatura é ambiente, fora da faixa a temperatura será: quase ambiente ($21^{\circ}\text{C} - 29^{\circ}\text{C}$), não ambiente ($0^{\circ}\text{C} - 50^{\circ}\text{C}$), e assim por diante dependendo do valor de sua pertinência.

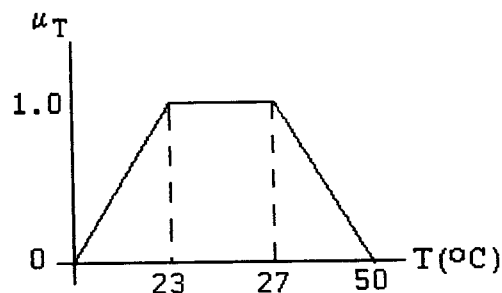


Figura 3. 1 - Função de pertinência para temperatura ambiente.

O conjunto “fuzzy” permite representar conceitos vagos expressos na linguagem natural. A representação do conjunto “fuzzy” depende não somente do conceito, mas também do contexto no qual é utilizado [Klir & Yuan, 1995]. Por exemplo, a temperatura alta no contexto de clima e no contexto do reator nuclear são representadas por conjuntos “fuzzy” diferentes.

Sendo x um elemento genérico de um conjunto X , um conjunto “fuzzy” de A em X é definido como um conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) / x \in X\}$$

onde $\mu_A(x)$ é chamada de função de pertinência de x em A . A função de pertinência mapeia cada elemento de X com um valor de pertinência contínuo entre 0 e 1.

Normalmente os valores de pertinência são definidos para variáveis lingüísticas (ex.: temperatura alta) através da quantificação da percepção humana [Ray & Ghoshal, 1997].

A variável lingüística é caracterizada por uma quintupla $(x, T(x), U, G, M)$ na qual x é o nome da variável, $T(x)$ é o conjunto de termos de x , que é o conjunto de

nomes dos valores lingüísticos de x no qual cada valor é um n.º “fuzzy” definido em U ; G é a regra sintática para gerar o nome dos valores de x ; e M é a regra semântica para associar cada valor com seu significado [Lee, 1990]. Por exemplo, se velocidade for interpretada como uma variável lingüística, então seus conjuntos de termos $T(\text{velocidade})$ poderia ser:

$$T(\text{velocidade}) = \{ \text{lenta, moderada, rápida, muito lenta, mais ou menos rápida...} \}$$

Cada termo em $T(\text{velocidade})$ é caracterizado por um conjunto “fuzzy” no universo de discurso $U = [0, 150]$. O termo lento pode ser interpretado como a velocidade abaixo de 60 Km/h, o termo médio como a velocidade em torno de 80 Km/h e rápido como a velocidade acima de 120 Km/h. Esses termos podem ser caracterizados como o conjunto “fuzzy” no qual as funções de pertinências são mostradas na figura 3.2.

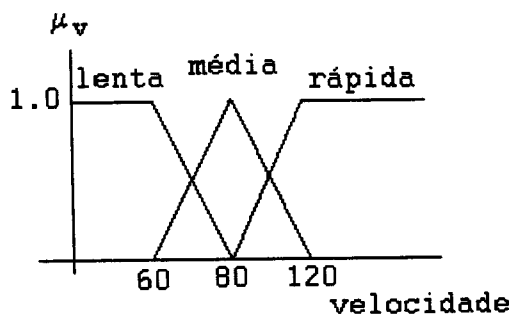


Figura 3. 2 - Funções de pertinência para velocidade.

3.2 - Operações com conjuntos “fuzzy”

Sejam A e B dois conjuntos “fuzzy” em U com suas funções de pertinências μ_A e μ_B , respectivamente. As operações de conjuntos como união, intersecção e complemento para conjuntos “fuzzy” são definidas através de suas funções de pertinências.

União: A função de pertinência $\mu_{A \cup B}$ é definida ponto a ponto para todo $x \in U$ por:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \text{ ou}$$

$$\mu_A(x) \cup \mu_B(x) = \mu_A(x) \vee \mu_B(x)$$

Intersecção: A função de pertinência $\mu_{A \cap B}$ é definida ponto a ponto para todo $x \in U$ por:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \text{ ou}$$

$$\mu_A(x) \cap \mu_B(x) = \mu_A(x) \wedge \mu_B(x)$$

Complemento: A função de pertinência $\mu_{\neg A}$ é definida ponto a ponto para todo $x \in U$ por:

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

Exemplo: Sejam A e B dois conjuntos “fuzzy” de $X = \{-2,-1,0,1,2\}$ e:

$$A = 0.5/-2 + 0.4/-1 + 1.0/0 + 0.3/1 + 0.7/2$$

$$B = 0.1/-2 + 0.4/-1 + 0.9/0 + 0.2/1 + 0.6/2$$

$$A \cup B = 0.5/-2 + 0.4/-1 + 1.0/0 + 0.3/1 + 0.7/2$$

$$A \cap B = 0.1/-2 + 0.4/-1 + 0.9/0 + 0.2/1 + 0.2/2$$

$$\neg A = 0.5/-2 + 0.6/-1 + 0.0/0 + 0.7/1 + 0.3/2$$

A figura 3.3 mostra as operações de união e intersecção dos conjuntos “fuzzy” graficamente.

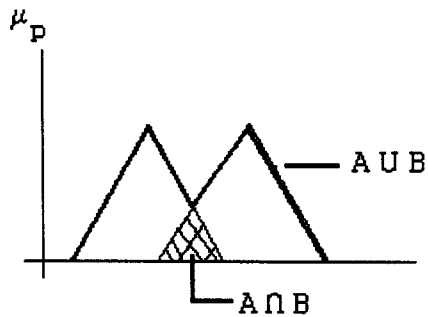


Figura 3.3 - Operação de união e intersecção nos conjuntos "fuzzy". A área hachurada representa a intersecção e a linha mais forte nos contornos representa a união.

Produto Cartesiano: O produto cartesiano entre dois conjuntos “fuzzy” A e B é definido como:

$$\mu_A \times \mu_B = \min\{\mu_A(x), \mu_B(x)\}$$

Relação “Fuzzy”: Sejam A e B dois conjuntos “fuzzy” não vazios. A relação “fuzzy” R é um subconjunto “fuzzy” de $A \times B$. Se $A = B$, então R é uma relação binária “fuzzy” em A.

Um exemplo simples de uma relação binária “fuzzy”. Seja $U = \{1, 2, 3\}$. Se for definido o quanto um elemento do conjunto é igual ao outro, poderia-se ter:

$$R(1,1) = R(2,2) = R(3,3) = 1$$

$$R(1,2) = R(2,1) = R(2,3) = R(3,2) = 0.7$$

$$R(1,3) = R(3,1) = 0.4$$

Em notação matricial, tem-se:

	1	2	3
1	1	0.7	0.4
2	0.7	1	0.7
3	0.4	0.7	1

Composição Sup-Star: Se R e S são relações “fuzzy” em $U \times V$ e $V \times W$, respectivamente, a composição R e S é uma relação “fuzzy” definida por:

$$RoS = \{(u,w), \sup[\mu_R(u,v) * \mu_S(v,w)]\} \quad u \in U, v \in V e w \in W.$$

onde * pode ser qualquer operador da classe de norma triangular (geralmente min).

Exemplo: Sejam duas relações “fuzzy”:

		y1	y2	y3	y4
R = x é maior que y =	x1	0.8	0.1	0.1	0.7
	x2	0	0.8	0	0
	x3	0.9	1	0.8	0.8

		z1	z2	z3
S = y é fechado em z =	y1	0.4	0.9	0.3
	y2	0	0.4	0
	y3	0.9	0.5	0.8
	y4	0.6	0.7	0.5

então a composição RoS é:

		z1	z2	z3
então a composição RoS é:	x1	0.6	0.8	0.5
	x2	0	0.4	0
	x3	0.7	0.9	0.7

Portanto, a composição RoS, nada mais é do que o produto cartesiano clássico de matrizes, sendo que no lugar da adição é utilizado o operador de união “fuzzy” e no lugar da multiplicação é utilizado o operador de intersecção “fuzzy”.

3.3 - Normas triangulares (t-norms)

As normas triangulares (t-norms) foram introduzidas para modelar distância no espaço métrico probabilístico. Essas normas são exhaustivamente usadas para modelar a conectiva AND. A tabela 3.1 mostra as t-norms básicas. A maior norma triangular é a intersecção e a menor é o produto drástico [Lee, 1990b]. As operações associadas à norma triangular são definidas para todo $x, y \in [0,1]$. Essas normas são definidas como:

$$T: [0,1] \times [0,1] \rightarrow [0,1]$$

Tabela 3.1 - Normas triangulares básicas (t-norms)

Normas triangulares	
Intersecção ou Mínimo	$\text{MIN}(x,y) = x \wedge y = \min \{x,y\}$
Lukasiewicz ou prod.limitado	$\text{LANG}(x,y) = x \odot y = \max\{0, x + y - 1\}$
prod.algébriico ou prod.probabilístico	$\text{PAND}(x,y) = x.y = x.y$
Fraca	$\text{WEAK}(x,y) = x \wedge y, \text{ se } x \vee y = 1$ 0 caso contrário
Hamacher	$H_\gamma(x,y) = x.y / (\gamma + (1-\gamma)(x+y-x.y))$
Dubois and Prade	$\sigma_\alpha = x.y / (\max\{x,y,\alpha\}), \quad \alpha \in [0,1]$
Prod.drástico	$x \cap y = x, \text{ se } y = 1$ $y, \text{ se } x = 1$ 0, se $x,y < 1$

3.4 - Co-normas triangulares (t-conorms)

As co-normas triangulares são usadas para modelar a conectiva OR. A tabela 3.2 mostra as t-conormas básicas. As t-conorms são definidas pelo seguinte mapeamento:

$$S: [0,1] \times [0,1] \rightarrow [0,1]$$

Tabela 3.2 - Co-normas triangulares básicas

Co-Normas triangulares	
união ou máximo	$\text{MAX}(x,y) = x \vee y = \max \{x,y\}$
Lukasiewicz ou soma limitada	$\text{LOR}(x,y) = x \oplus y = \min\{1, x + y\}$
soma algébriico ou soma probabilístico	$\text{POR}(x,y) = x + y - xy$
Forte	$\text{STRONG}(x,y) = x \vee y, \text{ se } x \wedge y = 1$ 1 caso contrário
soma disjunta	$x \Delta y = \max\{\min(x, 1-y), \min(1-x, y)\}$
soma drástica	$x \dot{\cup} y = x, \text{ se } y = 0$ $y, \text{ se } x = 0$ 0, se $x,y > 0$

3.5 - Implicação “fuzzy”

A implicação “fuzzy”(ou regras “fuzzy” IF-THEN, ou condição de estado “fuzzy” ou simplesmente regra “fuzzy”) assume a seguinte forma:

SE x é A ENTÃO y é B ou $A \rightarrow B$

onde A e B são valores lingüísticos definidos por conjuntos “fuzzy” no universo de discurso X e Y, respectivamente. A implicação “fuzzy” se divide em duas partes, uma chamada de antecedente ou premissa (“x é A”) e outra denominada conseqüente ou conclusão (“y é B”). Exemplo de regra “fuzzy”:

SE pressão é alta ENTÃO volume é pequeno

A regra “fuzzy” pode ser definida como uma relação binária “fuzzy” R no produto do espaço X e Y, uma vez que descreve a relação entre duas variáveis x e y. Portanto, $A \rightarrow B$ deve ser definida ponto a ponto, e isso é feito através dos operadores de implicação “fuzzy” (tabela 3.3).

Tabela 3.3 - Operadores de implicação "fuzzy"

Operadores de implicação "fuzzy"	
Larsen	$x \rightarrow y = xy$
Lukasiewicz	$x \rightarrow y = \min\{1, 1 - x + y\}$
Mamdani	$x \rightarrow y = \min\{x, y\}$
Standart Strict	$x \rightarrow y = 1$ se $x \leq y$ 0 c.c.
Gödel	$x \rightarrow y = 1$ se $x \leq y$ y c.c.
Gaines	$x \rightarrow y = 1$ se $x \leq y$ y/x c.c.
Kleini-Dienes	$x \rightarrow y = \max\{1-x, y\}$
Kleini-Dienes-Luk	$x \rightarrow y = 1 - x + xy$

A figura 3.4 mostra um exemplo:

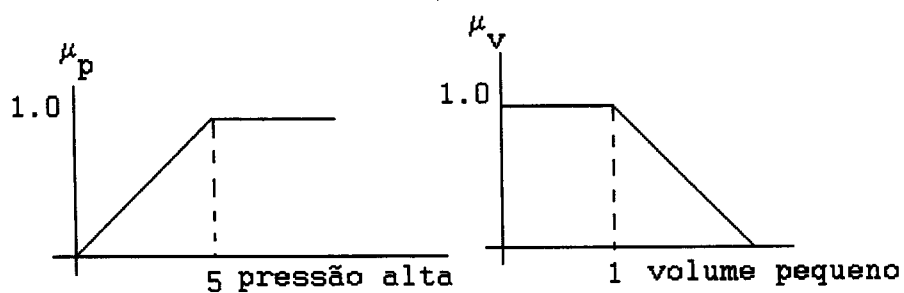


Figura 3. 4 - Exemplos de implicação "fuzzy".

Se $x = 4 \Rightarrow$ pressão alta $\mu_A = 0.75$ e

se $y = 1 \Rightarrow$ volume pequeno $\mu_B = 1$

$A(4) \rightarrow B(1) = 0.75 \rightarrow 1$ (=1 através do operador Gödel)

(=0.75 através do operador Larsen)

3.6 - Raciocínio “fuzzy”

Raciocínio “fuzzy” é um procedimento de inferência usado para obter conclusão de um conjunto de regras IF-THEN de uma ou mais condições. Há duas importantes regras de inferência “fuzzy” denominada *modus ponens* generalizada (GMP) e *modus tollens* generalizada (GMT) [Lee, 1990], dada por:

fato: $x \text{ é } A'$
regra: $SE x \text{ é } A \text{ ENTÃO } y \text{ é } B$ (GMP)
 conseqüência: $y \text{ é } B'$

fato: $y \text{ é } B'$
regra: $SE x \text{ é } A \text{ ENTÃO } y \text{ é } B$ (GMT)
 conseqüência: $x \text{ é } A'$

A implicação “fuzzy” é baseada na regra composicional. O GMP se reduz ao *modus ponens* clássico quando $A'=A$ e $B'=B$, e está intimamente relacionado à inferência dirigida a dados, a qual é útil em lógica “fuzzy” de controle [Lee, 1990]. O GMT se reduz ao *modus tollens* clássico quando $B'= \text{não } B$ e $A'= \text{não } A$, e está intimamente ligado à inferência dirigida a objetivos, o que é comumente usado em sistemas especialistas, especialmente na esfera de diagnósticos médicos [Lee, 1990].

As regras “fuzzy” podem ser classificadas em três tipos dependendo de sua forma consequente [Lee, Kwak & Kwang, 1996]:

Tipo 1: regras “fuzzy” com a parte consequente constante

R: SE x_1 é A_1 e e x_m é A_m

ENTÃO y é C_1

Tipo 2: regras “fuzzy” com a parte consequente sendo uma combinação linear

R: SE x_1 é A_1 e e x_m é A_m

ENTÃO y é $g(x_1, \dots, x_m) = b_0 + b_1x_1 + \dots + b_mx_m$

Tipo 3: regras “fuzzy” com a parte consequente sendo um conjunto “fuzzy”

R: SE x_1 é A_1 e ... e x_m é A_m

ENTÃO y é B

Nas regras acima, x_i e y denotam variáveis de entrada e saída respectivamente.

Os termos lingüísticos são conjuntos “fuzzy” parametrizados, no qual a forma é determinada por poucos parâmetros (conjunto “fuzzy” triangular, trapezoidal, sigmoidal, entre outros). O termo c_1 denota um valor constante, e $g(x_1 + \dots + x_m) = b_0 + b_1x_1 + \dots + b_mx_m$ é uma combinação linear das variáveis de entrada onde b_i 's são coeficientes constantes. Por fim, B indica um conjunto “fuzzy” parametrizado ou não parametrizado. Conjuntos “fuzzy” não parametrizados são definidos pela representação de seus valores e seu grau de pertinência, como exemplo temos:

velocidade baixa = $\{(0, 1.0), (80, 0.5), (120, 0.1)\}$

Podemos ter regras simples com antecedentes simples:

fato: x é A'

regra: SE x é A ENTÃO z é C

conseqüência: z é C'

Ou podemos ter múltiplas regras com múltiplos antecedentes:

fato: $x \text{ é } A' \text{ e } y \text{ é } B'$

regra 1: $SE \ x \text{ é } A1 \text{ e } y \text{ é } B1 \text{ ENTÃO } z \text{ é } C1$

regra 2: $SE \ x \text{ é } A2 \text{ e } y \text{ é } B2 \text{ ENTÃO } z \text{ é } C2$

conseqüência: $z \text{ é } C'$

Nas conexões AND (e) são utilizadas as normas triangulares e para realizar as ligações entre as diversas regras são utilizados as co-normas triangulares.

Ainda pode-se classificar as regras “fuzzy” quanto ao número de suas entradas e saídas. Nesse caso o sistema será referenciado como MIMO (múltiplas entradas e múltiplas saídas) e MISO (múltiplas entradas e saída simples) [Lee, 1990a].

3.7 - Modelos “fuzzy” (sistema de inferência “fuzzy”)

O sistema de inferência “fuzzy” (FIS) é um sistema que utiliza a teoria dos conjuntos “fuzzy” para mapear entradas (características, no caso de classificador “fuzzy”) em saídas (classes, no caso do classificador) [Knapp, 1996].

3.7.1 - Saída do modelo “fuzzy”

A saída de um modelo “fuzzy” se realiza através de seis passos [Knapp, 1996]:

- Determinação de um conjunto de regras “fuzzy” (criação das regras)
- “Fuzzificação” das entradas usando as funções de pertinência
- Combinação das entradas “fuzzificadas” em cada regra “fuzzy” para estabelecer a ativação da regra
- Cálculo da parte consequente da regra

- Combinação das partes consequente de todas as regras para obter a distribuição da saída
- “Desfuzzificação” da saída.

3.7.1.1 - Criação da regra “fuzzy”

As regras “fuzzy” descrevem como o FIS deve realizar a decisão de classificação ou controle. São escritas na seguinte forma:

SE (entrada 1)... E/OU (entrada 2)...ENTÃO (saída)

3.7.1.2 - “Fuzzificação”

A etapa de “fuzzificação” mapeia a entrada (ou característica) entre valores de 0 a 1, através das funções de pertinência. As funções mais utilizadas são:

Triangular: é especificada por três parâmetros {a,b,c}, a qual determina a coordenada x dos três cantos do triângulo (fig. 3.5(a)).

$$\text{Triangulo}(x;a,b,c) = \max(0, \min[(x-a)/(b-a), (c-x)/(c-b)])$$

Trapezoidal: é especificada por quatro parâmetros {a,b,c,d} (fig. 3.5(b)).

$$\text{Trap}(x;a,b,c,d) = \max(0, \min[(x-a)/(b-a), 1, (d-x)/(d-c)])$$

Gaussiana: é especificada por dois parâmetros {σ,c} (fig. 3.5(c)).

$$\text{Gaussiana}(x; \sigma, c) = \exp\{-(x-c)/\sigma\}^2$$

Sino: é especificada por três parâmetros {a,b,c} (fig.3.5(d)).

$$\text{Sino}(x; a, b, c) = 1/(1+[x-c]/a)^{2b}$$

Sigmoidal: é especificada por dois parâmetros {a,c} (fig. 3.5(e)).

$$\text{Sig}(x; a, c) = 1/(1 + \exp[-a(x-c)])$$

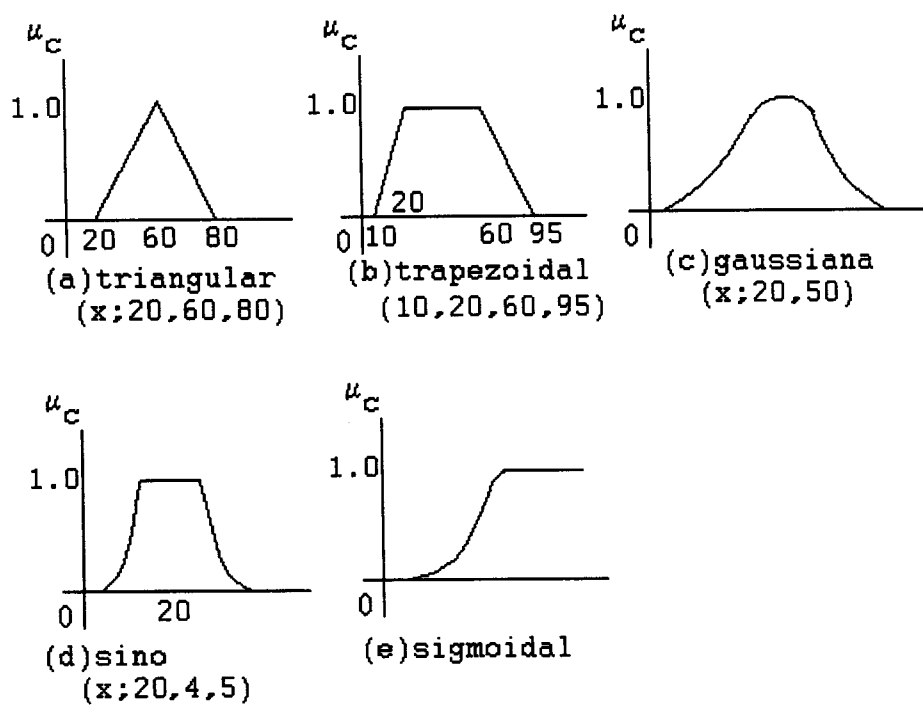


Figura 3. 5 - Funções utilizadas para "fuzzificação". (a) função triangular; (b)função trapezoidal; (c) função gaussiana; (d) função sino e (e) função sigmoidal.

3.7.1.3 - Combinação das entradas nas regras “fuzzy”

Nas regras “fuzzy” são utilizados os conceitos AND e OR e para implementá-los são utilizados as t-norms (item 3.3 deste capítulo) e as t-conorms (item 3.4 deste capítulo), respectivamente.

3.7.1.4 - Parte consequente de cada regra

A parte consequente de cada regra é conseguida através do uso de algum operador de implicação (ver item 3.5 deste capítulo). Os mais utilizados são a de Mamdani e Larsen.

3.7.1.5 - Combinação de todas as partes consequentes

As saídas de todas as regras devem ser combinadas para obter a distribuição de saída “fuzzy”. Geralmente é utilizado o operador de união.

3.7.1.6 - “Desfuzzificação”

Em muitos casos é desejável que a saída do sistema não seja “fuzzy” e sim uma saída definida. A etapa de “desfuzzificação” mapeia a saída “fuzzy” em um valor definido. Algumas estratégias de “desfuzzificação” são:

Método do máximo critério (MAX): O máximo critério acha o ponto no qual a distribuição de saída possui o seu primeiro valor de máximo (fig. 3.6).

Método da média dos máximos (MOM): O MOM acha o valor médio onde a saída possui seus máximos valores. Isto pode ser computado como (fig. 3.6):

$$z_o = \Sigma (w_j / l)$$

onde w_j são os valores no qual a função de pertinência possui seu máximos valores e l é o número de valores máximos encontrados.

Método de centro de massa (COA): É largamente usado. Essa estratégia procura pelo centro de gravidade da distribuição de saída "fuzzy" (fig. 3.6)[Lee, 1990b].

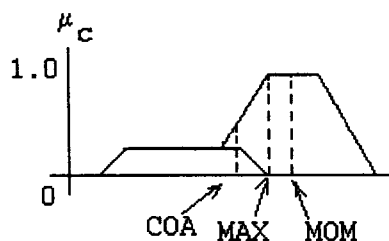


Figura 3. 6 - Estratégias de "defuzzificação". Método do máximo critério (MAX). Método da média dos máximos(MOM). Método do centro de máximo (COA).

3.7.2 - Alguns modelos "fuzzy"

Alguns modelos "fuzzy" são largamente utilizadas em várias aplicações. Geralmente as diferenças entre os sistemas está na sua parte consequente, no seu processo de agregação das várias regras e na "defuzzificação" do sistema.

3.7.2.1 - Modelo “fuzzy” de Mamdani

Neste modelo as implicações “fuzzy” são modeladas pelo operador de Mamdani (mínimo) e as regras são agregadas através do operador de união (máximo). Assumindo duas regras SE-ENTÃO da seguinte forma:

Regra 1: SE x é $A1$ e y é $B1$ ENTÃO z é $C1$

Regra 2: SE x é $A2$ e y é $B2$ ENTÃO z é $C2$

fato: x é x_0 e y é y_0

conseqüência: z é C

A ativação de cada regra, denotado por α_i , $i=1,2$; são calculadas por:

$$\alpha_1 = A1(x_0) \wedge B1(y_0) \quad (\text{conectiva AND})$$

$$\alpha_2 = A2(x_0) \wedge B2(y_0) \quad (\text{conectiva AND})$$

onde x_0 e y_0 são as entradas x_0 e y_0 “fuzzificadas”. A saída de cada regra é dada por:

$$C1'(w) = \alpha_1 \wedge C1(w) \quad (\text{operador de Mamdani})$$

$$C2'(w) = \alpha_2 \wedge C2(w) \quad (\text{operador de Mamdani})$$

E a saída geral do sistema é computado pelo operador de união:

$$C(w) = C1'(w) \vee C2'(w) = ((\alpha_1 \wedge C1(w)) \vee (\alpha_2 \wedge C2(w)))$$

A figura 3.7 mostra graficamente este modelo de raciocínio “fuzzy”.

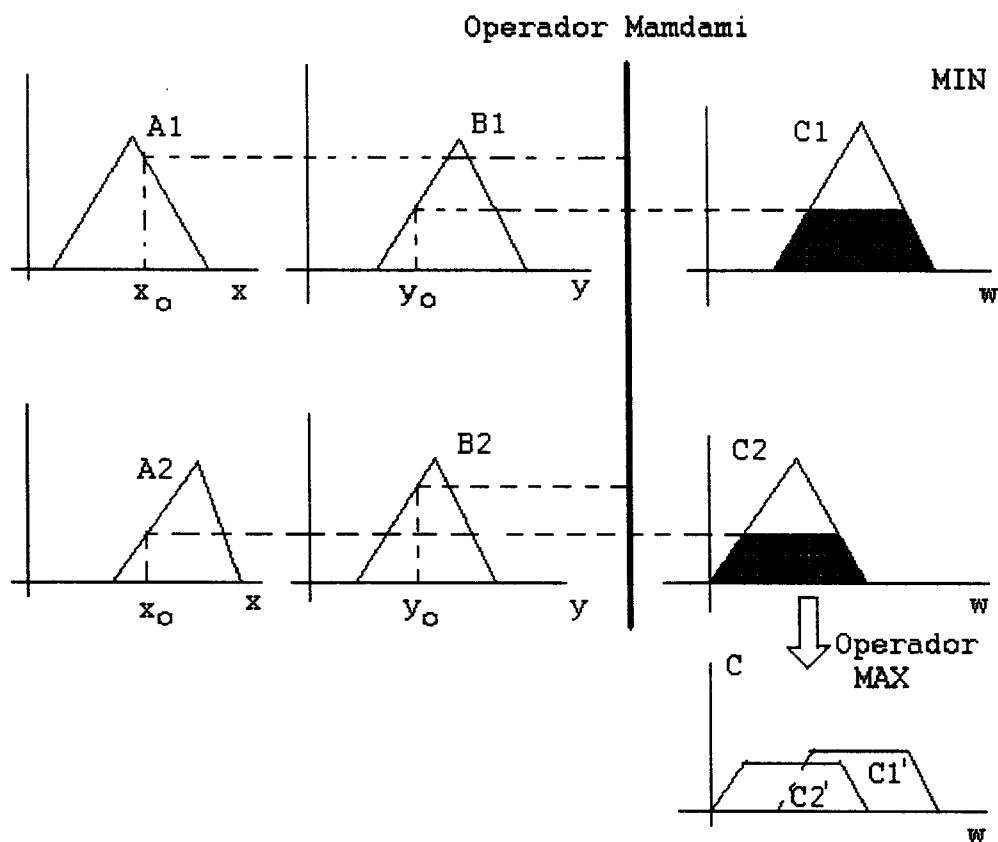


Figura 3. 7 - Raciocínio "fuzzy" de Mamdani.

3.7.2.2 - Modelo "fuzzy" de Larsen

É uma variação do modelo de Mamdani frequentemente encontrado na literatura. Ao invés de utilizar-se do operador de Mamdani nas implicações "fuzzy" das regras, é utilizado o operador de Larsen (produto). Portanto o modelo "fuzzy" de Mamdani pode ter variações se forem feitas diferentes escolhas para as conectivas AND (t-norms) e OR (t-conorms). A figura 3.8 mostra o raciocínio "fuzzy" do modelo de Larsen.

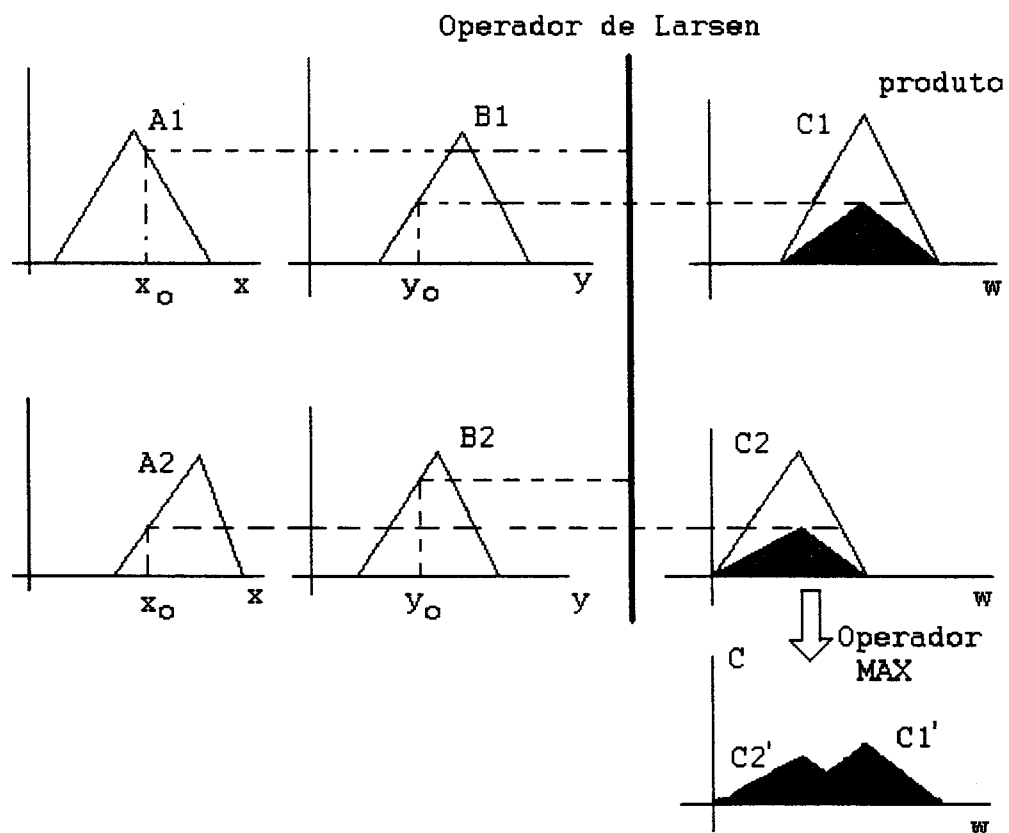


Figura 3. 8 - Modelo "fuzzy" de Larsen.

Nesses dois modelos se for necessário uma saída definida é utilizado uma etapa de "desfuzzificação" através de um dos modelos mostrado no ítem 3.7.1.6, ou seja, método do máximo critério, método da média máxima ou método do centro de massa.

3.7.2.3 - Modelo de Sugeno

Sugeno e Takagi [Sugeno & Takagi, 1985] usaram a seguinte arquitetura:

Regra 1: SE x é $A1$ e y é $B1$ ENTÃO $z1 = a_1x + b_1y + r_1$

Regra 2: SE x é $A2$ e y é $B2$ ENTÃO $z2 = a_2x + b_2y + r_2$

fato: x é x_0 e y é y_0

conseq.: z é z_0

A ativação da regra é dada por:

$$\alpha_1 = A1(x_0) \wedge B1(y_0) \quad \alpha_2 = A2(x_0) \wedge B2(y_0)$$

A saída de cada regra é conseguida através de uma relação polinomial:

$$z1 = a_1x_0 + b_1y_0 + r_1 \quad z2 = a_2x_0 + b_2y_0 + r_2$$

Na agregação das regras, feita através do centro de massa, é conseguido uma saída definida (fig. 3.9).

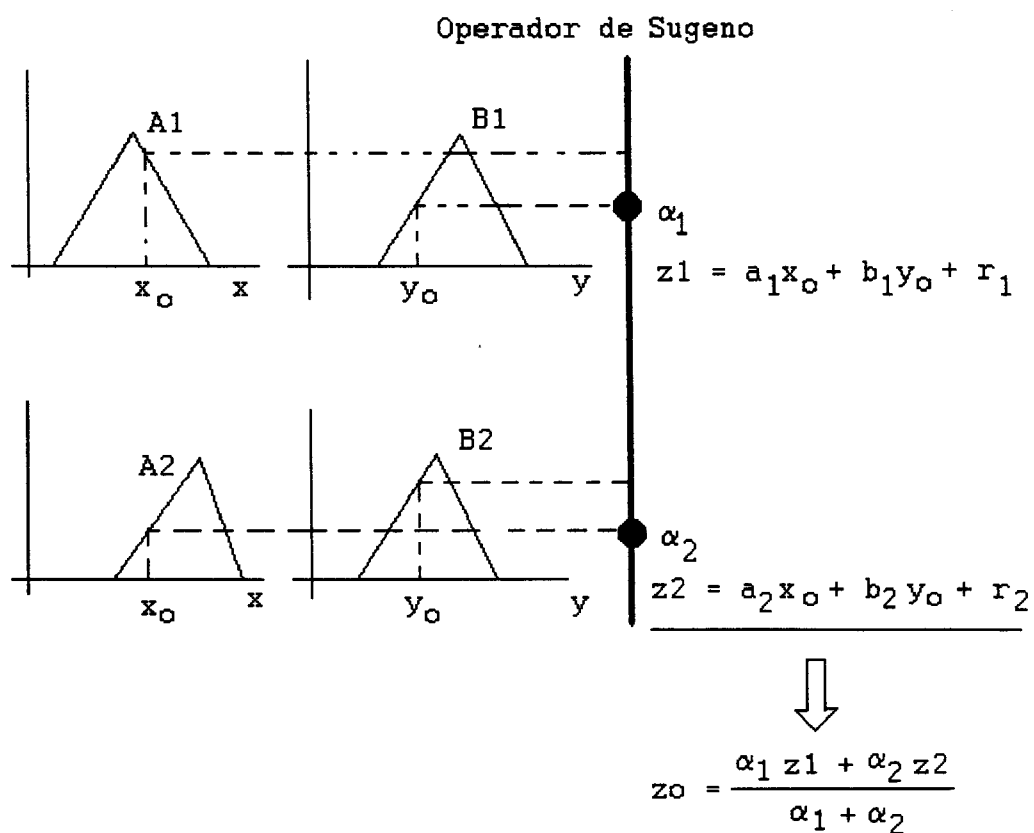


Figura 3.9 - Modelo "fuzzy" de Sugeno.

3.7.2.4 - Modelo "fuzzy" de Tsukamoto

Neste modelo a todas as funções de pertinência são monotônicas. O centro de massa é utilizado para obter a saída, que no caso já é definida. A figura 3.10 mostra o raciocínio deste modelo.

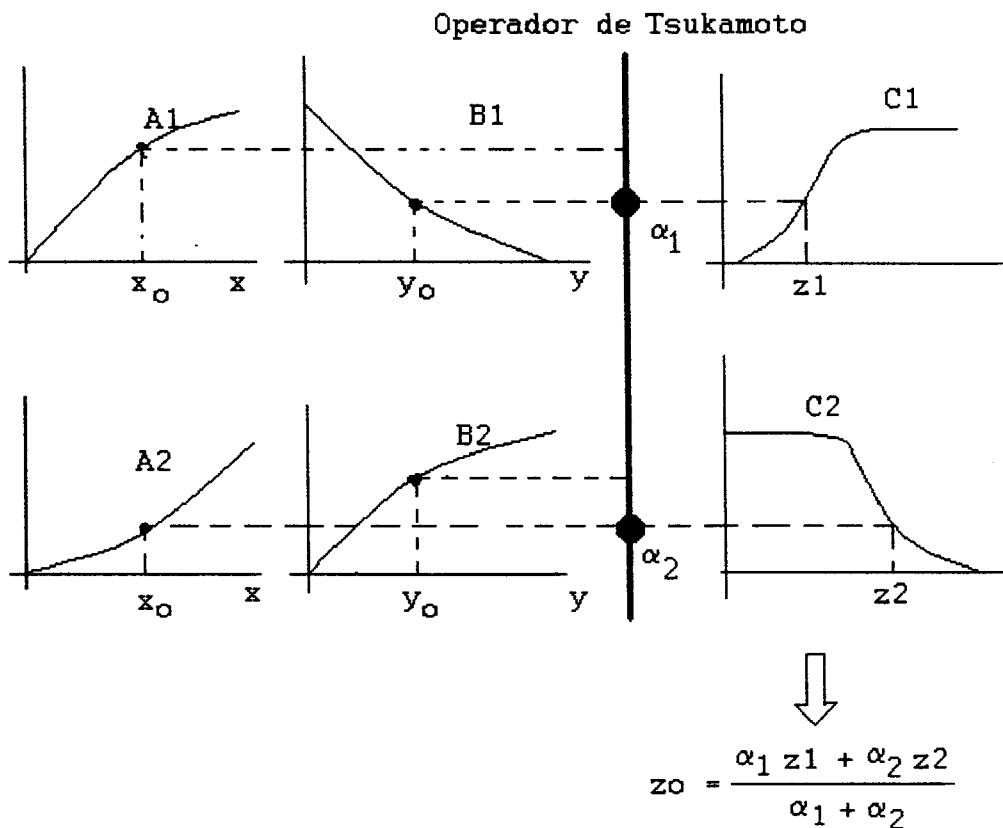


Figura 3. 10 - Modelo "fuzzy" de Tsukamoto.

3.8 - Conclusão

O sistema lógico clássico é poderoso para resolver problemas especificados em termos de dois valores (binário). O sistema lógico "fuzzy", como extensão dos sistemas clássicos consegue manipular problemas em que há vários estados de

decisão, incluindo o binário. Além disso, o sistema “fuzzy” trata com conceitos lingüísticos (mais, menos, maior,...) de uma forma parecida com a do ser humano.

O procedimento de raciocínio “fuzzy” é paralelo, ou seja, a combinação da parte antecedente da regra com a entrada pode ser computada de forma paralela. Isso faz com que o procedimento “fuzzy” seja apropriado para ser implementado em processadores paralelos.

Os modelos “fuzzy” de Mamdani e Larsen exigem técnicas de “desfuzzificação” caso uma saída definida seja desejada, enquanto que os modelos de Tsukamoto e Sugeno já fornecem a saída definida. Segundo Jang [Jang & Gulley, 1995] as vantagens dos modelos de Sugeno e Tsukamoto são: eficiência computacional, trabalha bem com técnicas lineares (controle PID), garante a continuidade da superfície de saída e é adequado a análise matemática.

As vantagens dos modelos de Mamdani e Larsen são, segundo Jang [Jang & Gulley, 1995]: mais intuitivo, larga divulgação e é mais adequado às entradas humanas.

O modelo de Mamdani trunca as funções de pertinência nas partes conseqüentes das regras através do operador MIN. O modelo de Larsen diminui a escala das funções de pertinência das saídas das regras usando o operador do produto. O valor de saída das regras entre os dois modelos podem ser iguais ou diferentes dependendo do método de “desfuzzificação” utilizado.

O sistema proposto nesta tese tem interesse em conhecer a classe vencedora e não o seu valor definido, por isso foi empregado o método MIN-MÁX utilizado por Law [Law, Itoh & Seki, 1996] que é baseado no método de Mamdani.

4 – REDES NEURAIIS

4.1 - Introdução

O interesse em redes neurais é justificado quando se faz a comparação entre o cérebro com os computadores dos dias de hoje. O cérebro humano é superior ao computador em várias tarefas, como visão e reconhecimento de voz.

Os computadores são melhores em tarefas sequenciais como aritmética (soma, multiplicação,...) e busca em banco de dados. Isso se deve ao fato dos computadores serem máquinas seriais de alta velocidade, enquanto o cérebro possui uma velocidade de processamento baixa, mas sendo altamente paralelo.

O cérebro humano possui as seguintes características que são desejáveis a um sistema artificial [Hertz et al, 1993]:

- É robusto e tolerante à falha. Células nervosas morrem diariamente sem afetar seu desempenho significativamente.
- É flexível. Pode ajustar facilmente a um novo ambiente pela "aprendizagem".
- Trabalha com informações nebulosas, probabilísticas, ruidosas ou inconsistentes.

- É altamente paralelo.
- É pequeno, compacto e dissipa pouca energia.

Essas características motivaram as pesquisas em sistemas baseados no cérebro humano, conhecidas como sistemas neurais.

4.2 - Neurônio

O cérebro contém dez bilhões (10^{10}) de células nervosas denominadas de neurônios. Cada neurônio se conecta com dez mil (10^4) outros neurônios [Beale & Jackson, 1990]. A figura 4.1 mostra um neurônio.

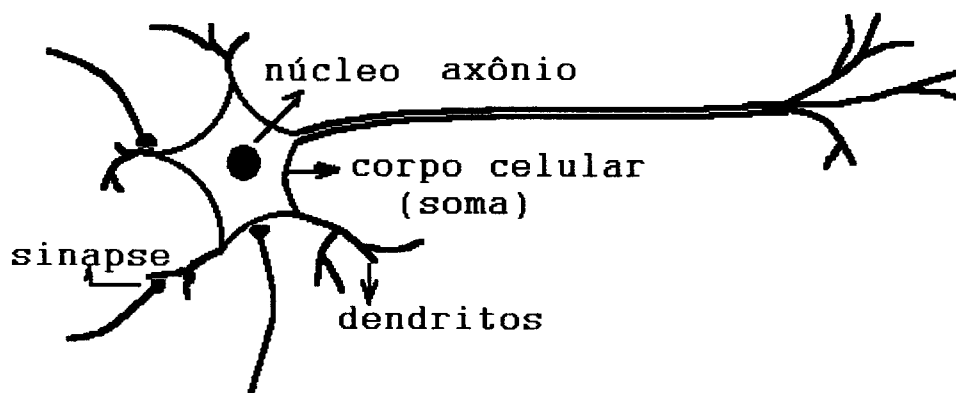


Figura 4. 1 - Neurônio.

Não há ligação direta entre o axônio de um neurônio e o dendrito de outro. A transmissão do sinal de uma célula a outra é um processo químico com uma específica substância transmissora sendo enviada de uma lado para o outro da junção (sinapse).

Os sinais entram através dos dendritos e/ou corpo celular. O efeito desses sinais é aumentar (caso o sinal for excitatório) ou diminuir (caso o sinal for de inibição) o

potencial elétrico no corpo celular. Se esse potencial atingir um limiar, um pulso (potencial de ação) é enviado pelo axônio às outras células através da arborização do axônio. Depois de ativada a célula possui um tempo de espera (período refratário) antes de poder ser ativada novamente.

Como o cérebro possui bilhões de neurônios conectados, cada célula contribui com um pedaço muito pequeno da informação. Portanto, a morte de algumas células não afetam o desempenho do cérebro significativamente. Além disso, o cérebro possui um auto-aprendizado dinâmico, isso faz com que a morte de algumas células sejam compensadas por outras.

Já os computadores precisam de programas sequenciais para aprenderem. Dependendo da aplicação esses programas são grandes e complexos. Se um "bug" ocorrer no programa, o computador não tem meios de se auto-programar. Como o computador possui apenas uma unidade de processamento, se a mesma estragar, o sistema todo fica inválido.

4.3 - Perceptrons

A função básica de um neurônio é somar as entradas, que chegam através dos dendritos e/ou corpo celular, e ativar uma saída caso essas entradas ultrapassem um certo limiar ("threshold").

Para modelar, da forma mais simples um neurônio, são necessárias as seguintes características:

- Algumas entradas, com as respectivas constantes de multiplicação (pesos), para simular as sinapses que colaboram com a quantidade de sinais transmitidos.

Alguns pesos podem ser negativos em referência às sinapses inibitória dos neurônios reais.

- A saída da célula, que pode ser ativada ou não dependendo da soma das entradas atingir ou não um limiar. A figura 4.2 mostra um modelo básico de neurônio.

As entradas do sinal são multiplicados pelos pesos. A saída é uma função “heaviside” da soma das entradas.

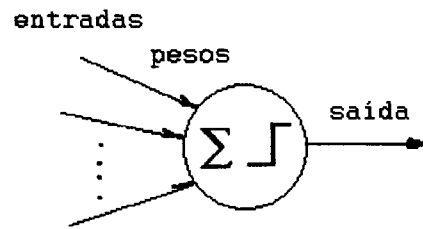


Figura 4. 2 - Modelo básico do neurônio.

$$y = f_h \left[\sum_{i=1}^n w_i x_i - \theta \right] \quad (4.1)$$

onde: y - saída

w_i - peso da i-ésima entrada

x_i - i-ésima entrada

θ - threshold. É o limiar de comparação

f_h - função “heaviside”: = 1 se x > 0

= 0 caso contrário.

A figura 4.3 mostra a função “heaviside”.

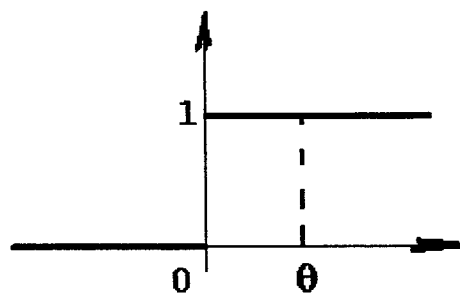


Figura 4. 3 - Função Heaviside.

Fazendo $\theta = -w_0x_0$, a eq.4.1 pode ser reescrita da seguinte forma:

$$y = f_h\left[\sum_{i=0}^n w_i x_i\right] \quad (4.2)$$

Esse modelo de neurônio foi proposto por McCulloch e Pitts em 1943. Em 1962, Frank Rosenblatt deu o nome de "perceptron" ao modelo de neurônio.

4.3.1 - Aprendizagem dos perceptrons

Como o modelo mais simples de perceptron possui dois níveis de saída, se o mesmo produz uma saída incorreta, deve-se reduzir as chances dessa saída ocorrer novamente. Se a saída for correta, nada se deve fazer. Assim temos o seguinte algoritmo para aprendizagem do perceptron:

1. Inicializar pesos e threshold aleatoriamente, fazendo $w_0 = -\theta$ e $x_0 = 1$.
2. Apresentar as entradas e a saída desejada.
3. Calcular a saída por:

$$y = f_h\left[\sum_{i=0}^n w_i x_i\right]$$

4. Adaptar pesos:

se correto

$$w_i(t+1) = w_i(t), \text{ vá para o passo 6}$$

se saída 0 e deveria ser 1

$$w_i(t+1) = w_i(t) + \eta x_i(t)$$

se saída 1 e deveria ser 0

$$w_i(t+1) = w_i(t) - \eta x_i(t)$$

5. Voltar ao passo 3.

6. Fim.

O η do passo 4 é uma taxa de adaptação que diminui a mudança dos pesos para que a rede não ultrapasse a solução desejada. O η é um valor inteiro positivo entre 0 e 1.

O perceptron é capaz de resolver vários problemas, mas é limitado quando o problema é linearmente inseparável, como o caso do OU-exclusivo. No caso do OU-exclusivo não há maneira de separar os dois resultados "1" dos resultados "0" através de uma linha no espaço bi-dimensional. A figura 4.4 mostra o espaço do problema OU-exclusivo.

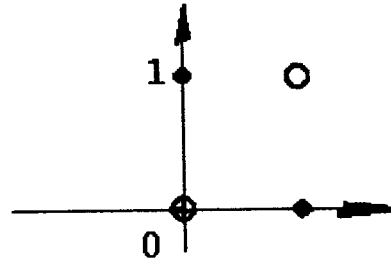


Figura 4. 4 - Problema do OU exclusivo.

4.4 - Classificação das redes neurais

Existem vários modelos de estruturas de redes em uso. Entretanto os modelos podem ser classificados por alguns elementos básicos próprios das redes. Esses elementos são: operação das unidades de processamento (“perceptrons”), topologia das redes, relação entre a entrada e saída da rede e o algoritmo de aprendizagem.

4.4.1 - Operação das unidades de processamento

O neurônio pode ser classificado quanto a natureza de seus sinais e sua função de transferência. A natureza dos sinais de entrada nos “perceptrons” podem ser discretas

$\{0,1\}$ ou $\{-1,0,1\}$ ou podem ser contínuas, assumindo valores nos intervalos $[0,1]$ ou $[-1,+1]$ [Gallant, 1994].

O tipo do modelo é dependente do tipo de não linearidade (função de ativação) incorporada na camada escondida [Brown & Harris, 1994]. Diferentes funções de ativação podem ser implementadas: função sigmoideal (em redes de “backpropagation”), funções trigonométricas (em redes de “links” funcionais-FLN), funções gaussianas (em redes de bases radiais-RBF) e funções polinomiais (em rede de memória associativa-AMN).

4.4.2 - Topologia das redes

A topologia das redes é classificada pela forma na qual as unidades básicas estão arrançadas. As estruturas podem ser classificadas nos tipos direto e recorrente [Jang & Sun, 1995].

A *estrutura direta* pode ser descrita como um conjunto de unidades de processamento dispostos em camadas que se interconectam da saída de uma unidade à entrada de outra na camada subsequente. Não há ligações cíclicas e nem retroalimentação (fig. 4.5(a)).

A *estrutura recorrente* possui retroalimentação da saída de um neurônio para outra unidade em camadas predecessoras ou possuem nós cíclicos (fig. 4.5(b)). Exemplo desta estrutura é a rede de Hopfield.

Alguns autores consideram outras duas estruturas, que são casos particulares da estrutura direta. Uma é a *estrutura em treliça* que é composta de uma camada de elementos de entrada que tem o papel de passar informações para a camada de

processamento da rede. As camadas são arranjadas na forma de treliça (fig. 4.5(c)). A outra *estrutura* é a *pirâmide* na qual as unidades vão diminuindo da primeira à última camada, que possui apenas um neurônio (fig. 4.5(d)).

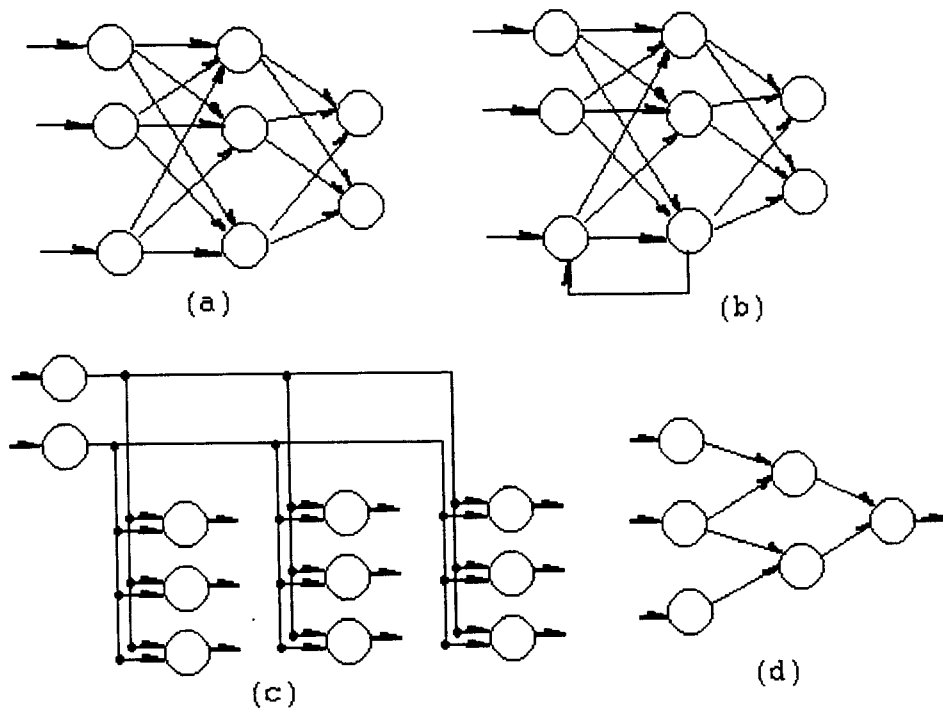


Figura 4. 5 - Estruturas de redes neurais. (a) direta, (b) recorrente, (c) treliça e (d) pirâmide.

4.4.3 - Relação entre entrada e saída de dados

A relação entre a entrada e saída dos dados em uma rede neural pode ser classificada em dois tipos: as redes autoassociativas e as redes hetero-associativas. As *redes autoassociativas* reconstróem a versão original de sinais de entrada corrompidos ou ruidosos [Brown & Harris, 1994]. A rede de Hopfield é um exemplo deste tipo.

As *redes hetero-associativas* mapeiam as entradas em uma representação alternativa, e isto é a maioria das classes de redes que tratam do problema de classificação e tarefas de aproximação funcional [Brown & Harris, 1994].

4.4.4 - Algoritmo de aprendizagem

A capacidade de aprender informações de seu ambiente é uma das características mais importantes das redes neurais. Esta capacidade consiste num processo iterativo onde a rede ajusta seus pesos até que a informação do ambiente seja aprendida.

Os ajustes dos pesos podem ser feitos através de três tipos: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. O *aprendizado supervisionado* é o mais comum; utiliza-se de um erro, a diferença entre a resposta desejada e a resposta obtida, como orientador ou supervisor do ajuste dos pesos. As redes de MLP (perceptrons múltiplas camadas) com algoritmo de “backpropagation” utilizam este aprendizado.

No *aprendizado não supervisionado* não existe a presença do supervisor; não há neste caso exemplos de entrada-saída associados para treinamento. O processo de aprendizagem da rede é que deve desenvolver suas próprias representações de acordo com algum critério de similaridade definido. Este treinamento só é aplicado nos casos em que não se tem informações das classes. Lin [Lin & Lu, 1995] comenta que a desvantagem deste aprendizado não supervisionado é sua ineficiência. A rede da teoria de ressonância adaptativa (ART) e Hopfield são exemplos que utilizam-se deste aprendizado.

O *aprendizado por reforço* é considerado por alguns autores como um tipo próprio de aprendizado, [Lin & Lu, 1995] e outros o consideram um caso especial de aprendizado supervisionado [Hertz et al, 1993]. Este aprendizado é utilizado onde as informações de saída não estão disponíveis, isto é, a resposta é avaliada em termos de certo ou errado apenas. A rede neural recebe um sinal de “feedback” de seu ambiente, sendo este sinal apenas avaliativo e não instrutivo. É um método de tentativa e erro. A rede não sabe, por exemplo, porque uma saída é errada e então tenta diferentes resposta até atingir a resposta certa [Hertz et al, 1993].

4.4.5 - Taxonomia das redes neurais

Considerando os itens anteriores, uma classificação das redes neurais mais utilizadas na prática pode ser vista na tabela 4.1 abaixo.

Tabela 4.1 - Taxonomia de redes neurais

Estrutura	Aprendizagem	Redes
Direta	supervisio- nado	Adaline/Madaline (fç. limiar) MLP com “Backpropagation” (fç. sigmoidal) FLN - (fç. Trigonométrica) RBF - (fç. Gaussiana) AMN - (fç. Polinomial) LVQ “counterpropagation”
	não supervisio- nado	matriz de aprendizagem Kohonen memória associativa
Recorren- Te	supervisio- nado	Cauchy “machine” Boltzamn “machine” “Brain-state-in-a-box”
	não supervisio- nado	Hopfield ART BAM “Additive Grossberg” (AG)

onde:

MLP – perceptrons múltiplas camadas.

FLN - redes de “links” funcionais.

RBF - redes de função de base radial.

AMN - rede de memória associativa.

LQV - Aprendizagem do vetor de quantização.

As redes de “counterpropagation” são consideradas como híbridas uma vez que combina o treinamento supervisionado com o não supervisionado. A idéia geral é haver uma camada que aprende de uma forma não supervisionada, seguida de uma (ou mais) camadas que são treinadas pelo algoritmo de “backpropagation” [Hertz et al, 1993].

4.5 - Escolha do modelo de rede neural

Para escolher um modelo de rede é necessário procurar, dentre as várias opções apresentadas na literatura, aquela que possui as características apropriadas para a aplicação.

O tempo de classificação é uma das características importantes na escolha do modelo. Geralmente as redes diretas são muito mais rápidas que as redes recorrentes, uma vez que estas dependem de uma realimentação dos nós posteriores gastando mais tempo para atingir o seu estado estável.

Quanto ao treinamento, se existirem modelos de pares de entrada-saída para treinamento o mesmo deve ser supervisionado, uma vez que o treinamento não supervisionado pode ser ineficiente [Lin & Lu, 1995].

Através dessas duas características e levando-se em conta que o presente trabalho tem como finalidade a classificação de placas de madeira para aplicação em ambiente industrial, no qual o tempo de processamento é um item essencial, o modelo de rede utilizado será direto. O aprendizado será supervisionado, pois há modelos para o treinamento.

Dentre as redes diretas e supervisionadas, a AMN não serve ao propósito deste trabalho por ser uma rede associativa. As outras redes possuem a mesma estrutura diferenciando-se através da função de transferência de seus neurônios, assim há: FLN com funções de ativação trigonométricas, MLP com “backpropagation” com funções de ativação sigmoidais, RBF com funções de ativação gaussianas e Madaline com funções de ativação limiares.

Outras características, não menos importantes, podem ser levantadas antes da escolha do modelo: o tempo de treinamento da rede, a capacidade de armazenamento, requerimento de memória, facilidade de implementação em “hardware”, facilidade de programação em computadores seriais e paralelos e facilidade de se adaptar a mudanças do ambiente.

4.6 – Perceptrons Múltiplas Camadas

Os perceptrons múltiplas camadas são unidades de perceptrons arranjadas em camadas, que podem resolver problemas mais complexos que um único perceptron.

Neste modelo há, geralmente, três camadas: a camada de entrada, a camada de saída e a camada escondida. Pode-se ter apenas a camada de entrada e de saída, ou

também, ter várias camadas escondidas entre as camadas de entrada e saída. A figura 4.6 mostra o modelo de um perceptron múltiplas camadas.

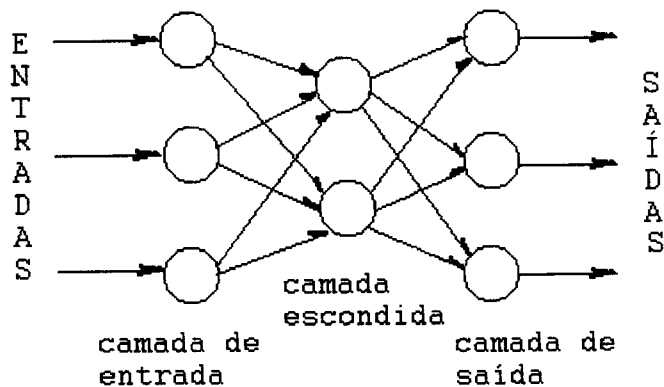


Figura 4. 6 - Perceptron múltiplas camadas.

A função heaviside, usada no perceptron simples, é substituída por uma função sigmoideal nas unidades de perceptrons da rede. Isso se faz necessário porque a função heaviside remove informações da saída de uma unidade à entrada da outra, tornando mais difícil o ajuste dos pesos da rede. A figura 4.7 mostra a função sigmoideal. Essa função possui infinitos valores entre 0 e 1, portanto qualquer variação nos pesos modifica a saída da função, o que não ocorre com uma função heaviside.

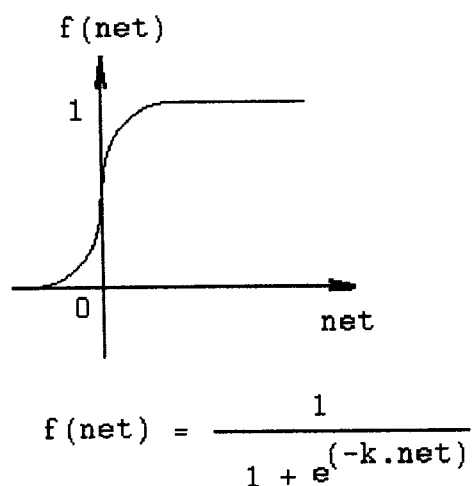


Figura 4. 7 - Função sigmoideal.

A função sigmoideal é uma das mais eficiente[Masters, 1994]. É uma função real, limitada, diferenciável e possui sempre derivada positiva. Como na fase de aprendizagem, as redes necessitam das derivadas de suas funções de ativação, torna-se

claro o porque da escolha das redes de “backpropagation” pela maioria dos pesquisadores[Masters, 1994]. A forma da função de ativação provoca profundo efeito na sua velocidade de aprendizagem, mas não na capacidade da rede [Masters, 1994].

4.6.1 – Classificador

Considerando uma rede com dois nós de entrada e um de saída (figura 4.8), onde cada unidade na camada de entrada define uma reta no espaço, assim a unidade na segunda camada realiza a classificação baseada na combinação das duas retas.

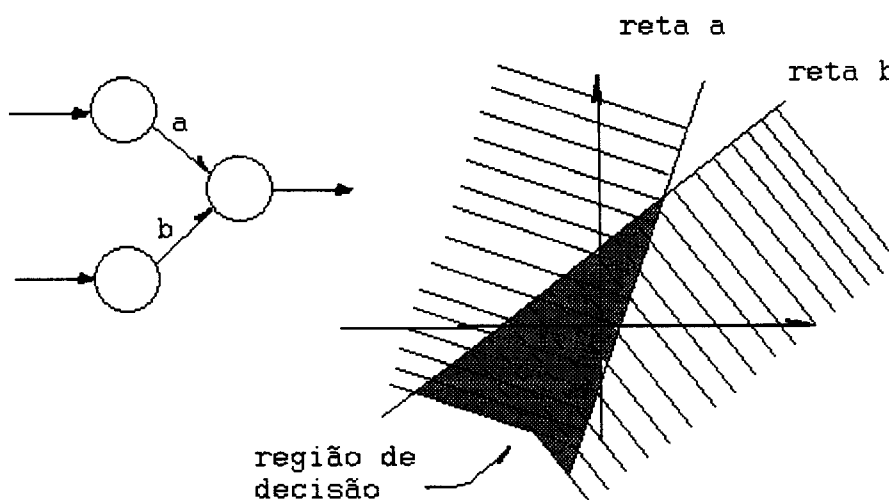


Figura 4. 8 - Rede de duas camadas e classificação do espaço.

Adicionando mais unidades de perceptrons na primeira camada, conseguiremos mais linhas de decisão definindo mais bordas como mostra a figura 4.9. Deve-se notar que essas regiões são convexas, ou seja, dois pontos de uma dessas regiões podem ser ligados por uma reta, e essa reta não cruza nenhuma borda.

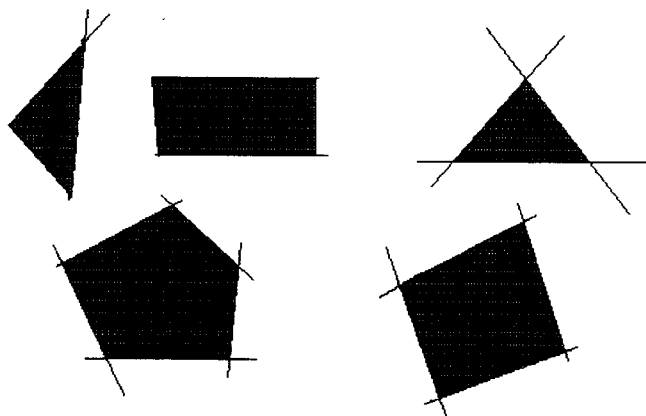


Figura 4. 9 - Regiões de decisão.

Se for adicionada outra camada de perceptrons, as unidades nessa camada recebem como entradas as regiões convexas da primeira camada. A combinação dessas regiões podem se interceptarem, sobrepremem ou se separarem uma das outras criando formas arbitrárias como da figura 4.10.

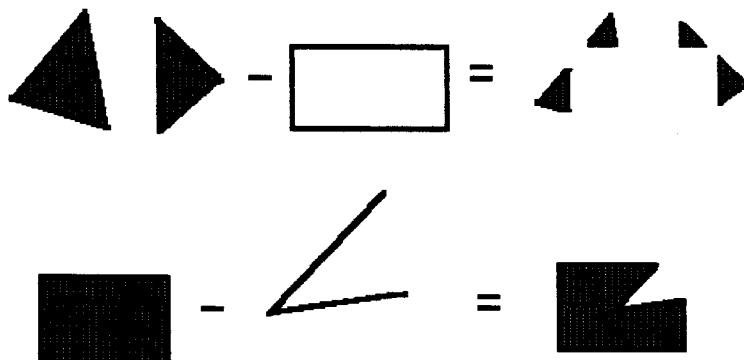
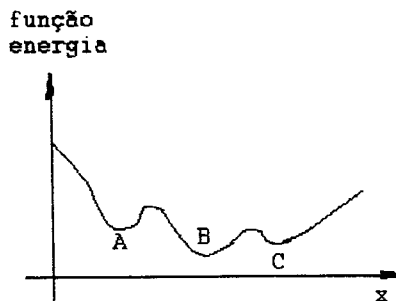


Figura 4. 10 - Combinações das regiões na segunda camada.

4.6.2 - Dificuldades no aprendizado

O objetivo do aprendizado da rede é achar o ponto no qual a função erro ou função energia seja mínima como mostrado na figura 4.11.

Às vezes um aprendizado pode achar o ponto C da figura 4.11 como a solução estável do problema, ao invés do ponto B. Isso pode acontecer caso o termo de ganho η seja grande,



levando a rede a dar largos passos, passando da solução ótima.

Figura 4. 11 - Função de energia de uma rede neural.

Para não ultrapassar o ponto de minimização, pode-se reduzir o termo de ganho η e aumentar a quantidade de unidades na camada escondida, levando à função de energia ter menos pontos de mínimos [Beale and Jackson, 1990]. Pode-se, ainda, adicionar o termo de momento:

$$w_{pi}(t+1) = w_{ji}(t) + \eta \delta_{pj} O_{pi} + \alpha (w_{ji}(t) - w_{ji}(t-1))$$

onde α - fator de momento (de 0 a 1).

Esse termo de momento permite uma convergência mais rápida da rede no início. Uma vez achada a melhor inclinação, a convergência se torna mais lenta.

4.7 - Conclusão

Uma das maiores características das redes neurais é sua habilidade de generalizar, isto é, classificar com sucesso modelos que não foram anteriormente apresentados [Beale & Jackson, 1990].

Isso quer dizer que tendo apresentado à rede uma gama larga de modelos no treinamento, qualquer novo modelo com entradas similares a um dos modelos treinados será bem classificado. Isso permite à rede trabalhar adequadamente com modelos que possam vir a ter ruídos, desde que esses ruídos não venham a denegrir inteiramente o modelo.

Como a rede de perceptrons de múltiplas camadas possuem seus elementos de processamento distribuídos paralelamente, qualquer peso ou nó danificado ou perdido não faz o desempenho cair drasticamente. Ao invés disso, a rede tem uma degradação suave em seu desempenho.

Outra vantagem de redes neurais é que elas podem ser implementadas em arquiteturas paralelas. As redes, também, podem modificar seu comportamento dependendo de seu ambiente, pois elas podem ajustar-se mediante algoritmos de aprendizagem.

O modelo de rede MLP com “backpropagation” foi utilizado nesta tese por possuir as seguintes características:

- ter topologia direta que fornece um processamento mais rápido do que a estrutura recorrente.
- ser hetero-associativa, isto é, que mapeia as entradas em uma representação alternativa.
- ter aprendizado supervisionado, pois existem modelos de pares de entrada-saída.

- utilizar a função sigmoïdal para a ativação dos “perceptrons”, pois é uma das funções mais eficiente segundo Masters [Masters, 1994]. Também é uma função real, limitada, diferenciável e com derivada positiva.

5 – REDE NEURO-FUZZY

5.1 - Introdução

Cada técnica de inteligência artificial possui características que a torna adequada a determinadas aplicações. Por causa do alto paralelismo na sua estrutura, as redes neurais tem resposta mais rápida e com melhor desempenho do que os computadores seqüenciais na emulação do cérebro humano [Kwan & Cai, 1994]. Assim processamentos de grandes quantidades de dados tornam-se fáceis mediante “hardware” adequado.

De outro lado, a lógica “fuzzy” é uma poderosa ferramenta para modelar o pensamento e a percepção humana [Kwain & Cai, 1994]. O raciocínio humano é de alguma forma nebuloso, pois é capaz de trabalhar com dados incertos, incompletos e ambíguos. As imagens na vida real, frequentemente contém dados que são incompletos ou ambíguos [Law, Itoh & Seki, 1996]. A utilidade da lógica “fuzzy” está na sua habilidade de tratar esses dados que são encontrados frequentemente na prática [Pal & Mitra, 1992].

Os sistemas híbridos combinam duas ou mais técnicas de maneira a aproveitar suas vantagens, e também superar algumas limitações que as técnicas individuais possuem. Assim, para habilitar um sistema a manipular situações da vida real de uma forma mais parecida com a do ser humano, uma forma é incorporar os conceitos dos conjuntos “fuzzy” dentro de redes neurais [Pal & Mitra, 1992].

As redes neuro-fuzzy podem ser divididas em duas categorias [Lin & Cunningham III, 1995]. Um grupo de redes neurais para o raciocínio “fuzzy” usam pesos “fuzzy” nas conexões entre nós. Ishibuchi [Ishibuchi et al, 1993] mostra um exemplo deste tipo de rede. No segundo grupo, os dados são “fuzzificados” na primeira ou segunda camada, mas os pesos da rede não são “fuzzy”, sendo este o grupo mais utilizado na prática.

5.2 - Neurônio “fuzzy”

Um neurônio não “fuzzy” convencional possui N entradas, N pesos e uma saída. O neurônio soma o produto das N entradas com seus pesos correspondentes e através de uma função de ativação é calculada a saída.

Um neurônio “fuzzy” possui N entradas com seus N pesos, e M saídas. Todas as entradas e pesos são valores reais e as saídas são valores reais no intervalo $[0, 1]$. Cada saída expressa o grau com que um dado modelo pertence ao conjunto “fuzzy” [Kwan & Cai, 1994]. A figura 5.1 mostra um neurônio “fuzzy” com N entradas e M saídas, onde:

$$z = h[w_1x_1, w_2x_2, \dots, w_Nx_N]$$

$$s = f[z - T]$$

$$y_i = g_j[s] \text{ para } j = 1 \text{ até } M$$

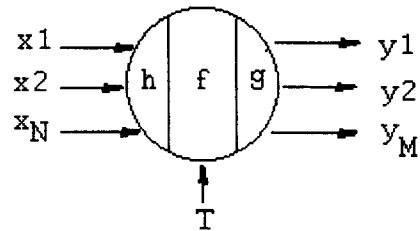


Figura 5. 1 - Neurônio "fuzzy".

Aqui z representa a entrada do neurônio "fuzzy"; $h[]$ é uma função de agregação; s é o estado do neurônio; T é "threshold" de ativação e $g_j[]$ são as funções de saídas do neurônio, a qual representa a função de pertinência do modelo de entrada. Nem sempre o neurônio "fuzzy" precisa da função de saída($g_j[]$) podendo utilizar a função de agregação ou a função de ativação para computar as saídas.

5.2.1 - Neurônio AND

O neurônio AND agrega sinais de entrada combinando-os com seus respectivos pesos e após isso realiza a operação AND desses resultados [Hirota & Pedrycz, 1994].

$$z = \text{AND}(x; w)$$

ou de uma forma mais geral (fig. 5.2):

$$z = T[x_i \text{ s } w_i]$$

onde T representa as t-normas

e s representa as t-conormas.

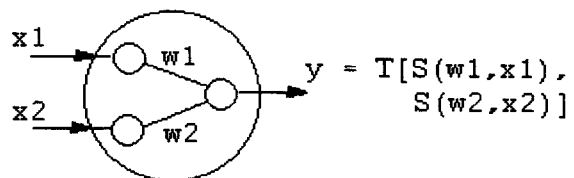


Figura 5. 2 - Neurônio AND.

5.2.2 - Neurônio “fuzzy” mínimo (MIN FN)

É um caso particular do neurônio AND onde a função de mínimo nas t-normas é utilizada como a função de agregação do neurônio “fuzzy”.

$$z = \min(w; x_i)$$

5.2.3 - Neurônio OR

O neurônio OR agrega os sinais de entrada com seus pesos e realiza a operação OR desses resultados [Hirota & Pedrycz, 1994].

$$z = OR(x; w)$$

ou de uma forma mais geral (fig. 5.3):

$$z = S[x_i \text{ t } w_i]$$

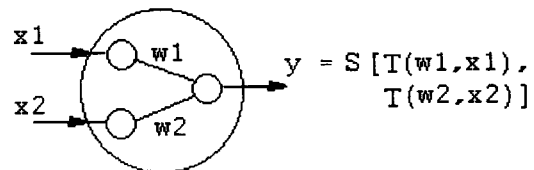


Figura 5.3 - Neurônio OR.

5.2.4 - Neurônio “fuzzy” máximo (MAX FN)

É um caso particular do neurônio OR onde é utilizado a função de máximo nas t-conormas.

$$z = \max(w; x_i)$$

5.2.5 - Neurônio “fuzzy” competitivo

Se o neurônio “fuzzy” tem uma variável de “threshold” T e somente uma saída, tal que:

$$y = g[s - T] = 0 \text{ se } s < T$$

$$1 \text{ se } s \geq T$$

$$T = t[c_1, c_2, \dots, c_k]$$

onde s é o estado do neurônio “fuzzy”; $T[\]$ é uma função “Threshold”; e c_k são variáveis competitivas do neurônio “fuzzy”. Este neurônio é denominado de neurônio “fuzzy” competitivo [Kwan & Cai, 1994].

Portanto, os neurônios “fuzzy” podem combinar suas entradas e seus pesos usando as t-normas, as t-conormas ou qualquer outra operação contínua. A função de agregação é realizada através das t-normas e t-conormas.

5.3 - Camadas das redes neuro-fuzzy

Um sistema neuro-fuzzy pode ser representado através das seguintes camadas [Higgins & Goodman, 1994]:

- Camada de entrada (“fuzzificação”): esta camada implementa a função de pertinência gerando valores entre 0 e 1 para um dado na entrada.
- Camada de regras (AND/OR ou MIN/MAX): esta camada contém um nó para cada regra, aplicando as operações “fuzzy” de AND e OR. As regras AND/OR podem ser

implementadas em camadas diferentes [Lee, Kwak & Kwang, 1996] [Lin & Lu, 1995] ou na mesma camada [Higgins & Goodman, 1994]. As operações AND e OR podem, também, ser implementadas na mesma regra [Higgins & Goodman, 1994].

- Camada de saída (“desfuzzificação”): esta camada transforma o sinal nebuloso em sua entrada em um sinal definido utilizando algumas das estratégias de “desfuzzificação” mostrada no item 3.7.1.6 do capítulo 3.

5.4 - Construção das regras SE-ENTÃO de uma rede neuro-fuzzy

As redes neuro-fuzzy são construídas através das regras SE-ENTÃO baseando-se em algum dos modelos “fuzzy” apresentado no item 3.7.2, ou seja Mamdani, Sugeno, Larsen ou Tsukamoto. Supondo que o objetivo seja separar os padrões em duas classes (C1-rápido e C2-lento), sendo x_1 e x_2 as características de entrada. A figura 5.4 mostra as funções de pertinência para x_1 e x_2 , sendo x_1 dividida em pequeno (A1) e grande (A2) e x_2 em baixo (B1) e alto (B2).

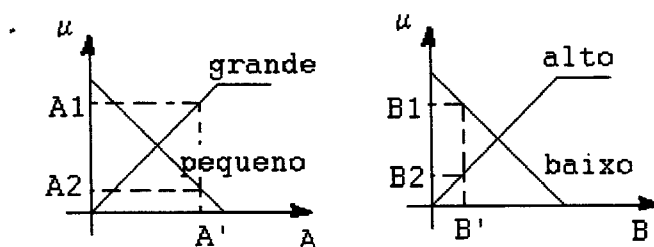


Figura 5. 4 - Funções de pertinências para duas entradas da rede neuro-fuzzy.

O primeiro passo é determinar a função de pertinência para x_1 e x_2 através do gráfico da figura 5.4. O segundo passo é montar as regras “fuzzy” baseado em algum estudo preliminar ou ensaios. Se as regras forem assim determinadas:

SE x_1 é A1 e x_2 é B1 ENTÃO o objeto é da classe C1

SE x_1 é A1 e x_2 é B2 ENTÃO o objeto é da classe C1

SE x_1 é A2 e x_2 é B1 ENTÃO o objeto é da classe C2

SE x_1 é A2 e x_2 é B2 ENTÃO o objeto é da classe C2

Note que na parte consequente das regras poderia-se associar um grau de incerteza. A rede deste exemplo fica estruturada como na figura 5.5. Uma simplificação deste exemplo é utilizar os operador MIN no AND e MAX no OR.

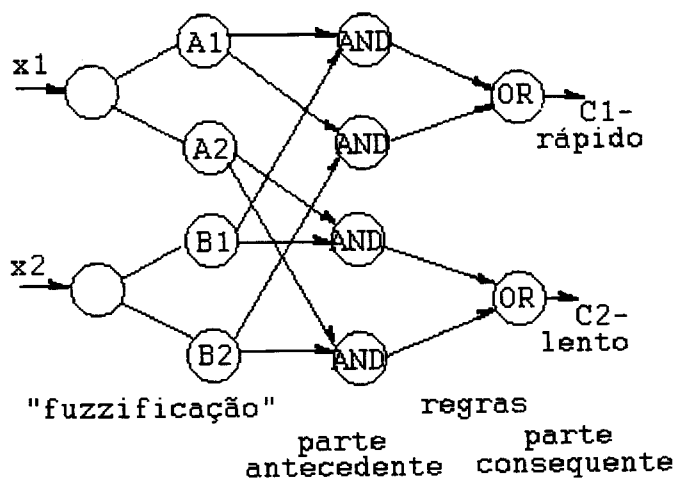


Figura 5. 5 - Estrutura da rede neuro-fuzzy do exemplo.

5.5 - Arquitetura ANFIS

A arquitetura ANFIS foi proposta por Jang [Jang, 1993] com o objetivo de modelar o raciocínio “fuzzy”. Para exemplificar a arquitetura ANFIS vamos considerar três regras “fuzzy” SE-ENTÃO:

SE x_1 é A_1 e x_2 é A_2 e x_3 é A_3 ENTÃO c é C_1

SE x_1 é B_1 e x_2 é B_2 e x_3 é A_3 ENTÃO c é C_2

SE x_1 é B_1 e x_2 é B_2 e x_3 é B_3 ENTÃO c é C_3

sendo A_1 e B_1 função de pertinência para x_1 ; A_2 e B_2 para x_2 e A_3 e B_3 para x_3 .

A figura 5.6 mostra o raciocínio "fuzzy" de Tsukamoto (qualquer outro poderia ser utilizado).

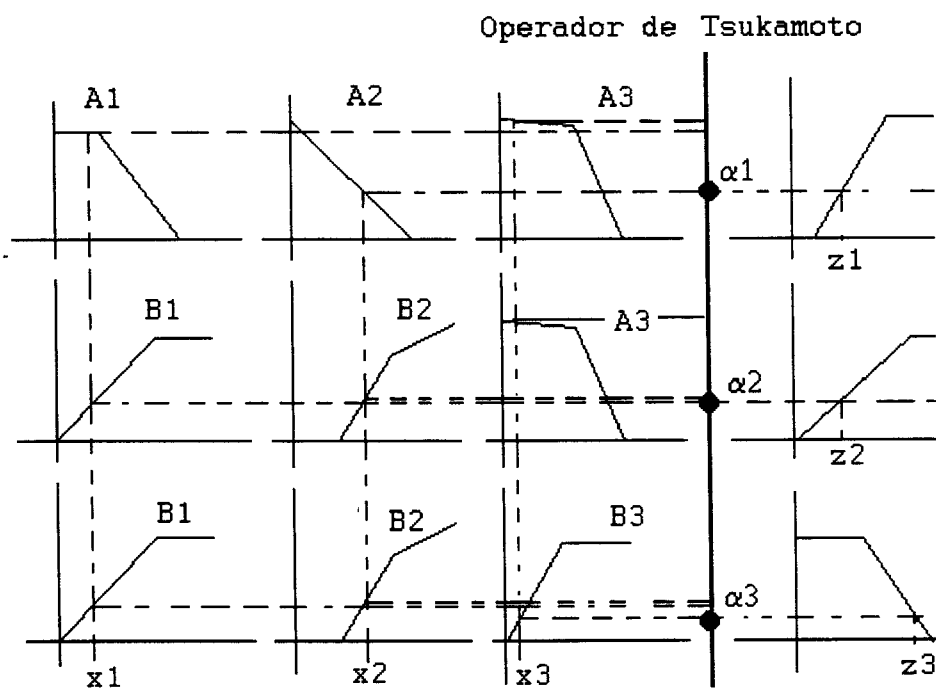


Figura 5. 6 - Raciocínio "fuzzy" de Tsukamoto para três entradas.

A ativação das regras é dada por:

$$\alpha_1 = \min(A_1(x_1), A_2(x_2), A_3(x_3))$$

$$\alpha_2 = \min(B_1(x_1), B_2(x_2), A_3(x_3))$$

$$\alpha_3 = \min(B_1(x_1), B_2(x_2), B_3(x_3))$$

e a saída do sistema é dado por:

$$z_o = (\alpha_1.z_1 + \alpha_2.z_2 + \alpha_3.z_3)/(\alpha_1 + \alpha_2 + \alpha_3)$$

a estrutura ANFIS deste raciocínio é mostrada na figura 5.7.

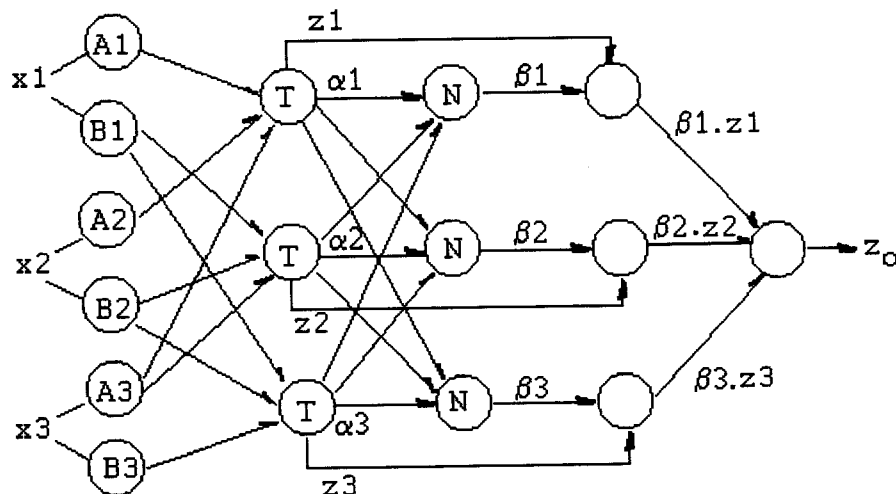


Figura 5. 7 - Estrutura ANFIS do raciocínio de Tsukamoto.

Na camada 1 é realizada a “fuzzificação” da entrada através de uma função de “fuzzificação”. Na camada 2 representada pela letra T dentro dos nós é computado a ativação das regras, ou seja é realizada a parte antecedentes das regras, e é realizada através das t-normas para modelar o operador AND.

A camada 3 rotulada pela letra N dentro dos nós, realiza a normalização dos valores de ativação das regras e é calculada por:

$$\beta_i = (\alpha_i)/(\alpha_1 + \alpha_2 + \alpha_3); \quad i = 1, 2, 3$$

A camada 4 é referente à parte consequente da regra, calculando o produto do nível de ativação normalizado pela saída das respectiva regra:

$$O^i = \beta_i \cdot z_i; \quad i = 1, 2, 3$$

Na camada 5 é realizada a soma de todas as saídas, fornecendo a resposta da rede.

Deve-se notar que esta arquitetura despreza os pesos entre os nós e no aprendizado são atualizados os parâmetros das funções de ativação e das funções de saída. O aprendizado pode ser feito através do gradiente descendente substituindo no lugar dos pesos originais os parâmetros. Como exemplo, se houvesse na função de saída de z_1 um parâmetro d_1 , o mesmo seria atualizado por:

$$d_1(t+1) = d_1(t) - \eta(\partial E_k / \partial d_1)$$

onde E_k é o erro na saída

5.6 - Classificação neuro-fuzzy

Os problemas referentes à classificação de padrões convencional está ligada à falta de um modo efetivo de definir uma borda limite entre as classes. A classificação “fuzzy” assume a borda limite entre classes como uma área contínua e sobreposta na qual um modelo tem um valor de pertinência para cada classe.

A tarefa de classificação é gerada pela partição “fuzzy” do espaço de características. Cada partição está associada a uma regra e estas devem ser

otimizadas, com o objetivo de diminuir o número de regras. Considerando a figura 5.8:

Pela figura 5.8 existe nove partições, o que implicaria em nove regras. Nota-se pela figura que se x_1 é A2 então o padrão pertence à classe C2, não

importando qual seja x_2 . Portanto as regras poderiam ser reduzidas para sete:

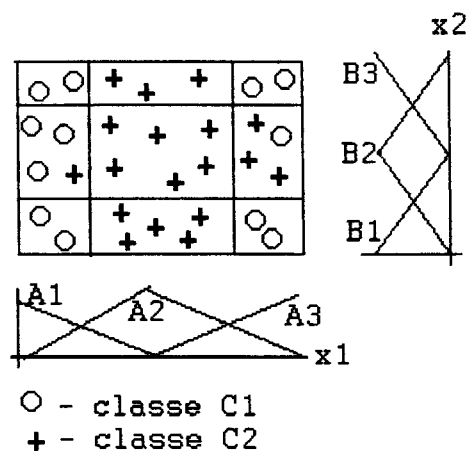


Figura 5.8 - Partição "fuzzy" do espaço.

- SE x_1 é A1 e x_2 é B1 ENTÃO x pertence à classe C1
- SE x_1 é A1 e x_2 é B2 ENTÃO x pertence à classe C1
- SE x_1 é A1 e x_2 é B3 ENTÃO x pertence à classe C1
- SE x_1 é A2 ENTÃO x pertence à classe C2
- SE x_1 é A3 e x_2 é B1 ENTÃO x pertence à classe C1
- SE x_1 é A3 e x_2 é B2 ENTÃO x pertence à classe C2
- SE x_1 é A3 e x_2 é B3 ENTÃO x pertence à classe C1

5.7 - Conclusão

Com este capítulo encerra-se a parte I deste trabalho. Os capítulos 3 e 4 foram as bases para o entendimento do sistema neuro-fuzzy discutido neste capítulo. Os

sistemas “fuzzy” e de redes neurais determinam a função de transferência entre o espaço de características a uma dada classe.

Em um classificador “fuzzy” a função de pertinência é inicializada perto da sua solução correta, enquanto a rede neural é inicializada em um estado aleatório [Knapp, 1996]. Isto faz com que o classificador “fuzzy” seja mais rápido no treinamento do que uma rede neural.

Um sistema “fuzzy” tem dificuldade em tratar com muitas características, funções de pertinências e regras and/or. As redes neurais trabalham bem com grandes quantidades de características e classes [Knapp, 1996].

Um sistema híbrido neuro-fuzzy utilizam das vantagens das duas técnicas, fazendo que o sistema seja apto a tratar com dados incertos e nebulosos e que o sistema possua a habilidade do aprendizado juntamente com uma estrutura altamente paralela.

6 – INSPEÇÃO VISUAL EM PLACAS DE MADEIRAS

6.1 – Introdução

A inspeção da qualidade das placas de madeiras no ambiente industrial geralmente é realizada por pessoas devidamente treinadas que levam em conta a homogeneidade visual de cada placa. A homogeneidade pode ser definida como a distribuição das fibras ou nós da madeira sobre a superfície da placa e que reflete diretamente sobre a qualidade das mesmas. Como se trata de um processo repetitivo, as falhas na inspeção manual acontecem devido ao cansaço, humor e às condições físicas e mentais das pessoas envolvidas no processo. Além disso, por causa das variações intrínsecas das placas torna difícil ao operador humano desempenhar sua tarefa de uma maneira ótima e consistente.

Portanto, é desejável e atrativo construir um sistema para a classificação da qualidade das placas de madeiras, a fim de obter uma maior confiabilidade na classificação e também de obter uma padronização no processo produtivo.

6.2 – Método de classificação de madeiras

Os trabalhos propostos em inspeção visual de placas de madeiras focalizam a atenção no reconhecimento de defeitos que as mesmas possam apresentar, exceção feita ao trabalho de Rodrigues [Rodrigues, 1998].

Defeitos em madeira são quaisquer características que a torna imprópria para o uso, tais como irregularidades na estrutura da fibra [Szymani & McDonald, 1981]. Segundo esses autores, os defeitos podem ser: biológicos ou de fabricação. Os defeitos biológicos podem ser naturais (grã¹ cruzada², nós, medula³) ou causados por fungos ou insetos (podridão⁴ clara ou parda, manchas, furos de insetos). Os defeitos de fabricação são gerados a partir da imprecisão da serragem e das práticas de manipulação do material (empilhamento inadequado).

Esses autores comparam os métodos ótico a laser, de ultra-som, microondas, raio-X e radiografia com neutron para detectar defeitos em madeiras. Eles mostram o tipo de defeito detectado, as vantagens e as limitações de cada método.

Conners [Conners et al, 1983] descreve um sistema de processamento automático de madeiras (ALPS) que utiliza tomografia computadorizada, tecnologia de varredura ótica, cálculo da estratégia de corte ótimo e equipamento de corte a laser dirigido por computador.

A tomografia é empregada para localizar nós internos e estabelecer a geometria de corte. Depois de cortada, as madeiras ainda contém defeitos visíveis na

¹ Grã: é a direção, tamanho, arranjo, aparência ou qualidade das fibras na madeira [ASTM D9-1981]

² Grão cruzada: desvio das fibras de uma linha paralela para os lados, o desvio pode ser diagonal ou espiral [ASTM D9-1981].

³ Medula: termo aplicado para a resina que ocorre na madeira de certas coníferas [ASTM D9-1981].

⁴ Podridão: decomposição gradual e avançada do tecido lenhoso produzida pela ação de fungos [ABPM,1990].

superfície (nós, furos de insetos, manchas) que não são detectados usando a tomografia [Conners et al, 1983]. Para detectar esses defeitos na superfície da madeira, os autores utilizaram equipamentos óticos a laser, é nessa fase que o artigo focaliza sua atenção. A estratégia básica empregada é dividir a imagem da placa de madeira em regiões retangulares disjuntas e classificar cada região independentemente, verificando se cada região contém ou não os defeitos. As características analisadas são feitas a partir de propriedades tonais (média, variância, assimetria e curtose) e também por medidas de segunda ordem onde é utilizado a matriz de co-ocorrência (inércia, tonalidade de agrupamento, homogeneidade local, energia e entropia). O propósito do trabalho é determinar se a combinação das propriedades tonais e das medidas de segunda ordem diferem com precisão uma região sem defeito de outra que o possua, além de identificar o tipo de defeito. Pelos resultados apresentados, individualmente nenhuma das duas medidas é superior à outra. A combinação das duas medidas forneceu uma faixa de reconhecimento de 57,35% para a podridão até 98% para a madeira limpa (sem defeito).

Bustler [Bustler et al, 1989] utiliza classificadores estatísticos para detectar defeitos em madeiras. Eles sub-dividem a imagem em regiões retangulares. Em cada região é processada uma transformação da intensidade do "pixel" (RGB) em um canal de intensidade definido como $(R+G+B)/3$ e um canal de cor definido como $(R-B)/2$. De cada janela são extraídas quatro características que são: a média e a variância do canal de intensidade e do canal de cor. São construídos os histogramas das quatro características e de cada um é computado um índice ("threshold"). As regiões que possuem valores de características menores que esses índices são marcadas como defeituosas. Em um segundo método, esses autores verificam as quatro regiões

adjacentes à região marcada e as comparam com um segundo “threshold”. Eles chegam a detectar 100% de furos e 70,3% das listras.

Sobey [Sobey & Semple, 1989] dividem a placa de madeira em regiões não sobrepostas, que denominam de área local, e de cada área calcula as seguintes medidas tonais: média, variância, assimetria e curtose. Utilizando um classificador linear, os autores conseguem detectar 95% das áreas com defeitos.

Koivo [Koivo & Kim, 1989] utilizam um classificador linear em árvore para detectar oito tipos de defeitos. As características de entrada são obtidas dos coeficientes de um modelo autoregressivo causal (CAR) e também da média dos “pixels”. Eles conseguem 97,2% de classificação dos oito defeitos.

Steele [Steele et al, 1991] detectam defeitos, analisando a inclinação da grã da placa de madeira. Zhu [Zhu et al, 1996] utiliza um tomógrafo computadorizado (CT) para localizar e identificar certas classes de defeitos em madeiras. Um filtro adaptativo suaviza cada uma das imagens bi-dimensionais do CT. Então, um sistema de segmentação bi-dimensional com múltiplos “thresholds” é aplicado para separar áreas potencialmente defeituosas das áreas limpas da madeira. Operações morfológicas (erosão e dilatação) são aplicadas na fatia da imagem segmentada para eliminar bordas irregulares e áreas espúrias. Um conjunto de testes é aplicado na imagem. O problema do sistema reside nos seus múltiplos “thresholds” e do tempo de processamento.

Rodrigues [Rodrigues, 1998] aplicou um método de limiar automático que utiliza a entropia de Shannon. A placa de madeira foi dividida em 36 regiões e de cada região é extraído o valor de limiar, a entropia total, a razão de assimetria, a diferença absoluta entre os valores de picos das parcelas da entropia $p \cdot \log(p)$ e $(1-p) \cdot \log(1-p)$ e

a razão entre o limiar da região como o limiar médio da placa. O autor aplica dois métodos para a classificação da qualidade das placas: o método de co-ocorrência e o método de redução. Seus resultados comparados aos dos operadores humanos variam de 56,28% a 62,81%.

6.3 – Madeiras

A madeira é uma importante matéria-prima por causa da diversidade de suas aplicações, tais como: estruturas e assoalhos, móveis e utensílios, tamancos e cachimbos, lenha para aquecimento, dormentes de estradas de ferro entre outras.

Devido a sua estrutura celular, a madeira apresenta um alto índice resistência/peso, sendo duas a quatro vezes mais resistente à tração que uma barra de aço comum de construção do mesmo peso [Larousse Cultural, 1998]. Além disso, a madeira apresenta boas características de isolamento térmico, acústico e elétrico, aliadas a uma baixa reatividade química [Larousse Cultural, 1998].

As madeiras de interesse comercial pertencem ao grupo das *gimnospermas* (coníferas) ou ao grupo das *angiospermas* (folhosas). No Brasil há três espécies de coníferas nativas, e dessas três apenas a *Araucaria angustifolia* tem importância econômica, mas por ser exigente em clima e solo, não pode ser usada em qualquer área de reflorestamento [Suassuna, 1997]

Várias espécies foram introduzidas no Brasil para a exploração da madeira. Nas regiões Norte e centro do estado de São Paulo, onde se situam os cerrados, caracterizados por inverno e primavera secos e solos pobres, as espécies que melhor

se adaptaram foram as de origem tropical (*Pinus caribaea hondurensis*, *Pinus caribaea caribaea*, *Pinus caribaea bahamensis*, *Pinus oocarpa* e *Pinus kasiya*) [Suassuna, 1997].

A espécie *Pinus caribaea hondurensis* é, segundo Suassuna [Suassuna, 1997], a mais promissora para reflorestamento por apresentar ótimo comportamento no viveiro e no campo. As placas de madeira estudadas nesta tese provém dessa espécie, por causa do clima e do solo do local onde as mesmas são cultivadas (centro do estado de São Paulo).

6.3.1 – Estrutura geral da madeira

Pela figura 6.1 distinguem-se as seguintes partes que compõe a seção transversal do tronco: casca, floema, câmbio, cerne, alburno e medula [Koga, 1988].

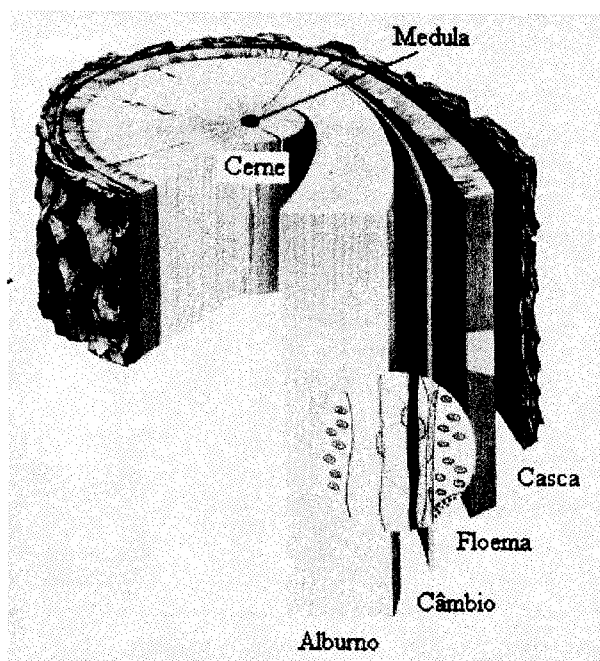


Figura 6. 1 - Seção transversal do tronco de uma árvore.

Casca – Composta por tecidos mortos, também denominada casca externa, tem a função de proteger os tecidos vivos da árvore contra o ressecamento e ataques de agentes externos.

Floema – Principal tecido condutor da seiva elaborada.

Câmbio - Situada entre o floema e o alburno é responsável pela produção das novas células nestas camadas. A espessura do alburno é maior do que a do floema porque o câmbio produz um número muito maior de células no alburno do que no floema. Além disso, após um certo período, o floema perde sua atividade deslocando-se para o exterior substituindo a casca que descama periodicamente.

Cerne e Alburno (xilema) – O alburno, ou xilema, é o principal tecido condutor da seiva bruta. Na planta jovem, o xilema consiste de células vivas ou funcionais que com o passar do tempo, tornam-se inativas para a condução e reserva. Essas células inativas constituem o cerne que possuem uma cor mais escura que o alburno. O cerne é mecanicamente mais resistente e menos susceptível ao ataque de microorganismo do que o alburno. O cerne e o alburno são designados de lenho.

Medula - Tecido primário localizado na região central do caule. Sua função é armazenar substâncias nutritivas para a planta. É muito susceptível ao ataque de microorganismo.

6.3.2 – Madeira de coníferas

A aparência e a textura da madeira são conseqüências da forma, da dimensão e da disposição das células que a compõe. Os tipos de células das coníferas são os traqueídes e os raios fusiformes.

Os raios fusiformes (ou raios medulares, ou traqueídes de raio) são conjuntos de células alongadas e achatadas que se dispõem radialmente em forma de fitas da casca até o centro da árvore, com a função de transportar e armazenar a seiva.

Os traqueídes são células alongadas com 3 a 5 mm de comprimento e de 0,04 a 0,06 mm de diâmetro, constituem até mais de 90% da madeira das coníferas [Hellmeister, 1972].

A espessura da parede dos traqueídes varia conforme a época do ano em que são formadas; as mais espessas são formadas durante os períodos de menos atividade vegetal, ou seja, nas estações em que os dias são mais curtos, a temperatura é mais baixa e as precipitações pluviométricas são mais escassas [Larousse Cultural, 1998]. O corte transversal de uma árvore apresenta anéis concêntricos e alternados de madeira clara e escura referentes à estação da formação dos traqueídes, que são designados anéis de crescimento (fig. 6.2).

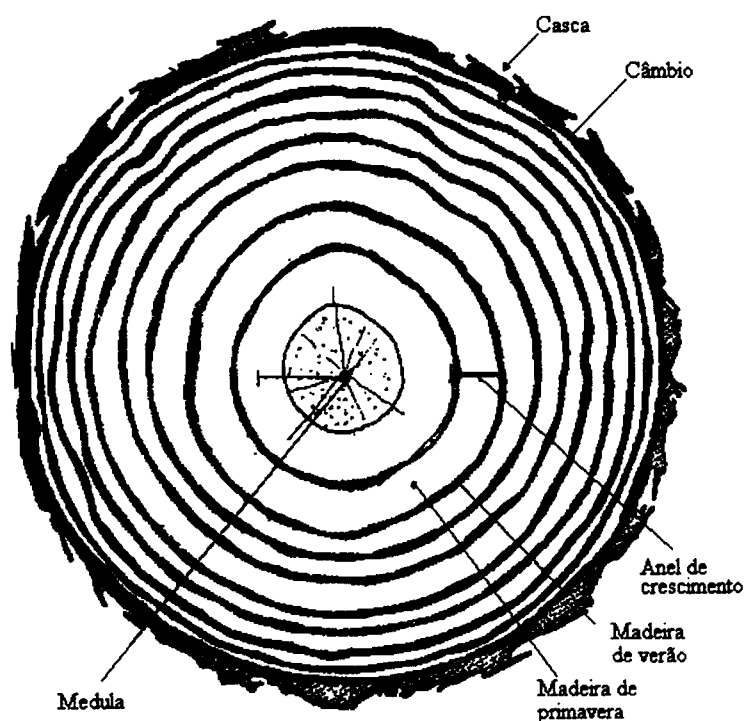


Figura 6. 2 – Desenho do corte transversal do tronco de uma árvore.

A zona mais clara do anel de crescimento é formada na primavera, possui células maiores e paredes celulares mais finas constituindo uma madeira mais mole e de menor densidade e resistência, é mais acessível à água e mais clara sendo designada de madeira de primavera (ou lenho inicial) [Hellmeister, 1972]. A zona mais escura do anel é denominada madeira de verão (ou lenho tardio), possui células menores com paredes celulares mais espessas constituindo uma madeira mais dura e de maior densidade e resistência mecânica [Hellmeister, 1972].

Devido ao tipo de crescimento da árvore e a disposição das células no caule, distinguem-se três planos principais nos quais a madeira normalmente é estudada (fig. 6.3) [Koga, 1988].

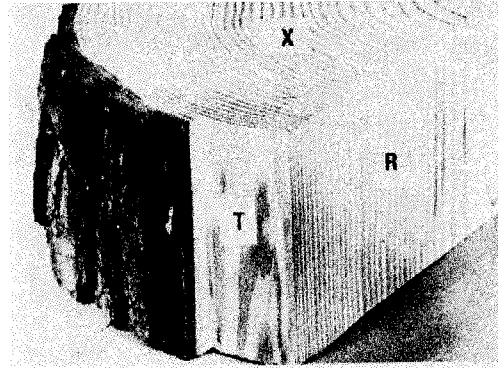


Figura 6.3 - Plano de orientação da madeira mostrando superfície transversal (X), superfície tangencial (T) e superfície radial (R).

Superfície transversal – É exposta quando a seção é perpendicular ao eixo longitudinal do caule; neste corte pode-se distinguir facilmente os anéis de crescimento, o cerne, o alburno e os raios medulares.

Superfície radial – É exposta quando a seção é longitudinal, passando pelo centro do tronco, paralela aos raios e perpendicular aos anéis de crescimento.

Superfície tangencial – É exposta quando a seção é longitudinal, perpendicular aos raios lenhosos e, também, tangencial aos anéis de crescimento.

Segundo Findlay [Findlay, 1975] o método empregado para o corte das toras de madeira é determinante para a classificação das pranchas obtidas, pois o mesmo mostra como os anéis de crescimento estão dispostos na madeira. Na figura 6.4 estão

representadas pranchas de madeira obtidas pelo emprego dos métodos de serragem tangencial e radial.

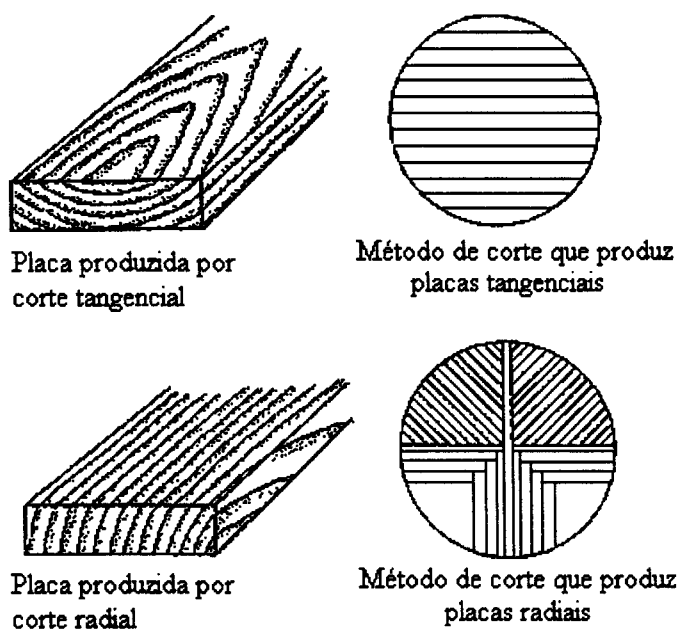


Figura 6. 4 - Métodos de corte da madeira.

6.3 – Definição das classes

As placas utilizadas neste trabalho são classificadas em cinco categorias, levando em consideração a homogeneidade visual, ou seja, a forma pela qual os anéis de crescimento estão dispostos nas superfícies das placas:

Na primeira classe (classe A), as placas são desprovidas da madeira dura dos anéis de crescimento e não há manchas em suas superfícies, por isso possuem melhor homogeneidade visual (fig. 6.5). São as melhores placas para a fabricação de lápis,

pois a presença da madeira dura dos anéis faz com que o lápis possa empenar ou rachar.

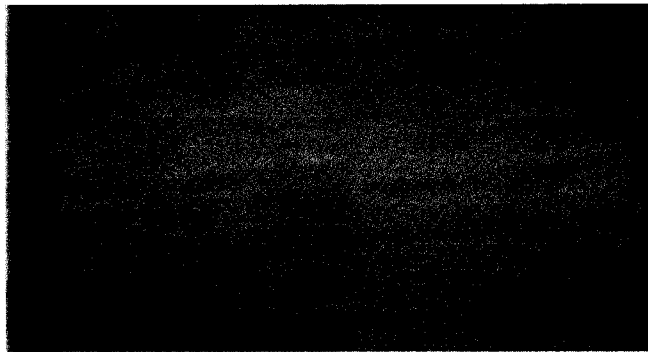


Figura 6. 5 - Placa A.

As placas da classe B podem apresentar poucos anéis de crescimento nas

superfícies ou pequenas áreas

mais escuras que o resto da

placa (fig. 6.6). Essa classe

contém as placas de transição

entre as placas lisas (classe

A) e as obtidas pelo corte

radial (classe C).



Figura 6. 6 - Placa B.

As placas da classe C são provenientes do corte radial, onde pode-se notar a presença de listras longitudinais que correspondem à madeira dura dos anéis de crescimento (fig. 6.7). Os

técnicos especializados

classificam nesta categoria as

placas que contêm de cinco a

nove listras.

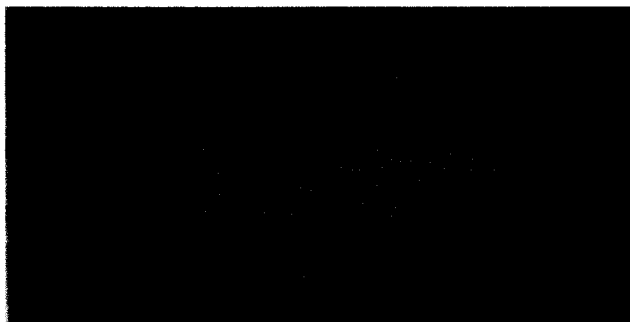


Figura 6. 7 - Placa C.

A classe D contém as placas de transição entre as placas do corte radial (classe C) e aquelas obtidas do corte tangencial (classe S). Podem possuir várias listras longitudinais de madeira dura, ou seja, o número de listras ser superior a nove, ou podem apresentar áreas pequenas de concentração de madeira dura (fig.6.8).

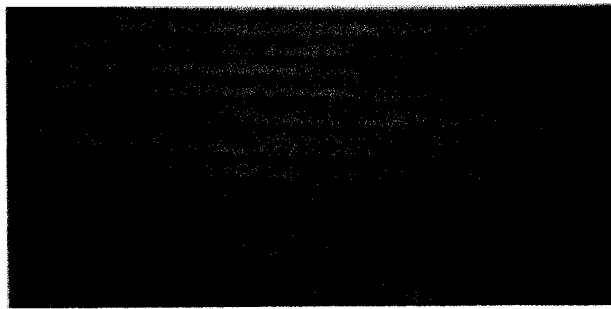


Figura 6. 8 - Placa D.

As placas da classe S são obtidas pelo corte tangencial da tora de madeira (fig. 6.9). Possuem grande quantidade de madeira dura e por causa disso, os lápis

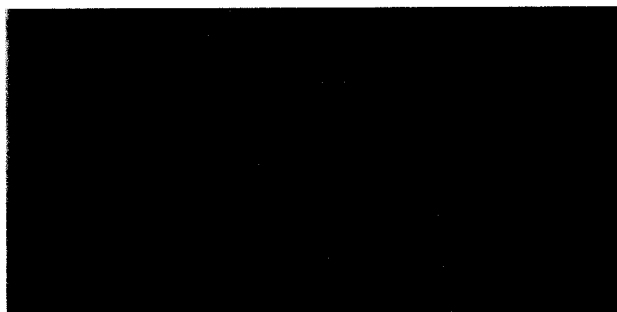


Figura 6. 9 - Placa S.

fabricados com esta placa tem maiores probabilidades de racharem. É, portanto, a pior classe para manufatura de lápis.

6.4 – Conclusão

Este capítulo mostrou alguns trabalhos de inspeção visual realizado em madeiras. Os trabalhos, em geral, focalizam a atenção no problema de detectar defeitos. Para isso aplicam estatísticas de primeira e de segunda ordem, sendo que não há uma garantia de qual seja a melhor para detectar os defeitos, pois isso depende

muito do tipo de madeira analisada, do método de aquisição de imagem e do tipo de defeito em análise. Portanto o sistema deve ser dedicado ao problema, já que é inviável tecnologicamente um sistema responder adequadamente a todos os casos.

O anexo 1 mostra as classificações realizadas pelos três inspetores humanos e pelos vários métodos utilizados, enquanto as imagens das 199 placas de madeira para teste estão gravadas no cd-rom anexo.

Através do anexo 1 pode-se notar que a classificação realizadas pelos especialistas humanos é muito subjetiva, isto porque cada inspetor detém um conhecimento próprio baseado em suas experiências anteriores. Além disso, os inspetores podem se confundir por causa das grandes variações de tonalidades entre as placas, isto quer dizer que uma placa mais escura pode ser classificada em uma classe diferente de outra placa mais clara apesar das duas apresentarem a mesma área de lenho tardio e com, praticamente, a mesma disposição espacial.

Muitas vezes, também, a diferença entre o lenho tardio e o lenho inicial é pequena, dificultando a classificação.

Por causa dessas dificuldades, a confiabilidade da inspeção humana é pequena. Caso um conjunto de placas seja apresentado mais de uma vez a um classificador humano, dificilmente ele manterá as mesmas classificações das placas feita anteriormente.

O sistema automático é desejado para sanar essas dificuldades, ou seja, padronizar o processo e manter a confiabilidade. Por padronizar o processo entende-se tornar o processo dependente de uma base de conhecimento fixa, o que é difícil em uma linha de produção composta de operadores humanos. A confiabilidade do sistema é pequena em uma linha de produção, pois as classificações das placas depende do

estado de espírito dos inspetores (humor, problemas pessoais) e do cansaço físico de fazer uma tarefa repetitiva durante o dia todo, por isso um conjunto de placas terá classificação diferente se a mesma for inspecionada de manhã ou de tarde, mesmo sendo feita pela mesma pessoa.

Como será mostrado no próximo capítulo, as curvas de características das placas não são discriminantes tornando os métodos tradicionais pobres para a classificação. Por isso a proposta desta tese é usar um sistema neuro-fuzzy, pois os métodos tradicionais tiveram um desempenho fraco e segundo Masters [Masters, 1993] qualquer tarefa de classificação feita por análise discriminante tradicional pode ser feita tão bem (e geralmente muito melhor) pelas redes neurais. Em outro livro, Masters [Masters, 1994] destaca a superioridade, quanto à velocidade, das redes neurais sobre os métodos tradicionais indicando sua aplicação em processamento de sinais e imagens que exigem operação em tempo real.

Nesta tese será utilizada duas redes neurais operando independentemente uma da outra, cada uma realizando sua própria classificação. A lógica “fuzzy” será utilizada para combinar as saídas das redes neurais com o objetivo de refinar a resposta final. O que será comprovado nesta tese é a superioridade do sistema neuro-fuzzy proposto sobre os classificadores tradicionais em aplicações nas quais os dados de entrada não são discriminantes.

7 - MÉTODOS PARA CLASSIFICAÇÃO

7.1 – Introdução

Como suporte para a aquisição das imagens e para classificação das placas de madeira, foi desenvolvido um equipamento específico. Este equipamento é na essência uma esteira rolante similar às utilizadas na indústria, acrescido de um sistema de aquisição de imagens e acionamento mecânico para separação das placas classificadas. Um microcomputador com uma placa “frame grabber” controla todo o sistema (fig. 7.1).

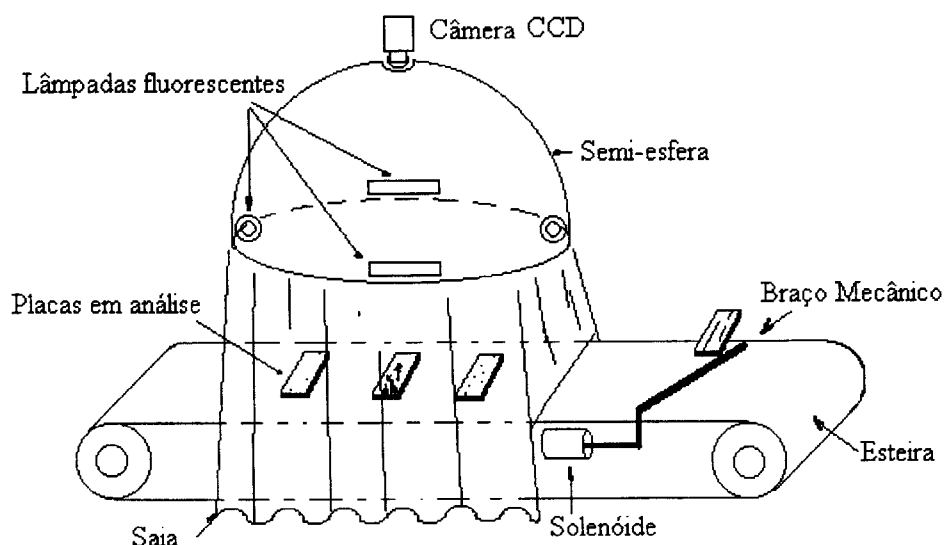


Figura 7. 1 - Sistema mecânico para classificação das placas.

A parte elétrica é composta por uma câmera CCD Hitachi, uma “frame grabber” da Data Translations IRIS (DT-2851 e DT-2858) para a aquisição da imagem e de um microcomputador 486 DX-2 com frequência de 66 MHz. O software para analisar e controlar o sistema foi feito em Microsoft C- versão 6.0, pois é o único software em C que a placa Iris suporta.

Foram utilizadas lâmpadas fluorescentes para a iluminação artificial alimentadas por reatores eletrônicos com uma frequência de chaveamento de 33 kHz para diminuir as variações da iluminação. As figuras de 7.2 a 7.5 mostram o sistema montado.



Figura 7. 2 - Sistema montado I.



Figura 7. 3 - Sistema montado II.

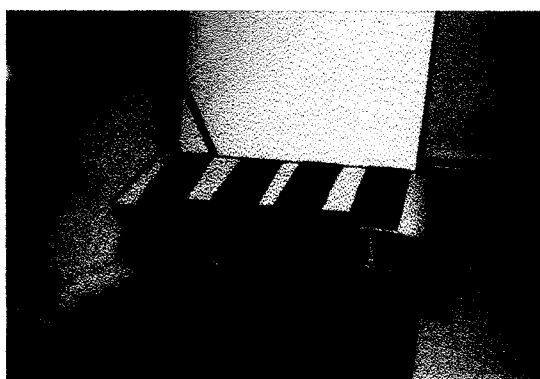


Figura 7. 4 - Sistema montado III.

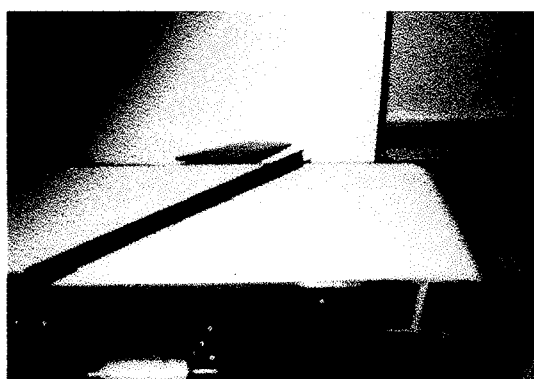


Figura 7. 5 - Sistema montado IV.

7.2 – Metodologia

O sistema para avaliação da qualidade das placas de madeira é dividido em cinco partes: aquisição de imagem, pré-processamento, extração de características, redes neurais e lógica “fuzzy” (fig. 7.6).

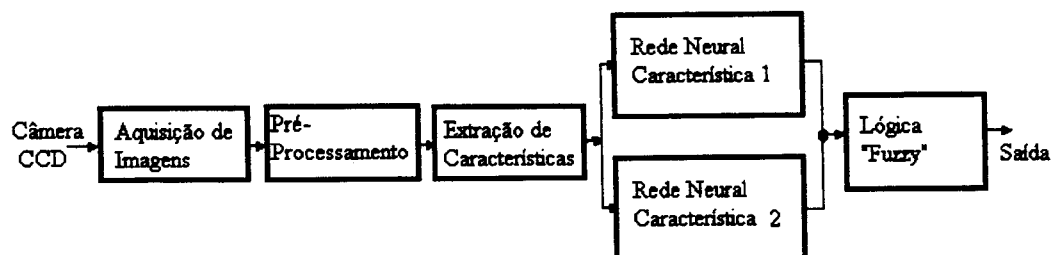


Figura 7.6 - Sistema em blocos para a classificação das placas.

7.2.1 – Aquisição de Imagem

A imagem é adquirida através da câmera CCD pela “frame-grabber” no tamanho de 128 x 256 “pixels” e com uma resolução espacial de 256 níveis de cinza. Os dados usados para treinamento devem ser divididos, pelo menos, em duas partes: uma na qual o treinamento é desenvolvido, denominada dados de treinamento e outra parte na qual o desempenho da rede é avaliado, conhecida como conjunto de teste [Prechelt, 1994].

Os conjuntos de treinamento e de teste devem ter aproximadamente o mesmo tamanho se houver números suficientes de amostras, caso contrário deve ser dada preferência ao conjunto de treinamento [Patterson, 1996]. Para Tafner [Tafner et al,

1996] quanto maior o número de dados de treinamento submetidos à rede melhor o seu desempenho. Os fatos correspondentes a cada classe devem ser representativos do maior número possível de situações correlatas (utopicamente, todas as situações possíveis) [Tafner et al, 1996]. O conjunto de teste constou de 120 placas pré-classificadas por especialistas humanos, a fim de representar o maior número de situações possíveis. O conjunto de teste constou de 199 placas de madeira.

7.2.2 – Pré-processamento

As imagens adquiridas pela câmera possuem baixos contrastes, portanto torna-se necessário uma etapa de pré-processamento. A transformação linear nos níveis de cinza foi utilizado para enriquecer o contraste por possuir uma fórmula simples e de processamento rápido. Além disso, para tornar o sistema mais independente da iluminação foi realizada uma binarização nas imagens enriquecidas através do método de seleção interativa que escolhe o limiar (“threshold”) automaticamente. O método da transformação linear nos níveis de cinza é realizado da seguinte maneira:

$$F(i, j) = \frac{(R_{MAX} - R_{MIN})}{(G_{MAX} - G_{MIN})} (G(i, j) - G_{MIN}) + R_{MAX}$$

Onde: $F(i,j)$: imagem nova

$G(i,j)$: imagem velha

R_{MAX} : valor máximo do nível de cinza para a imagem nova

R_{MIN} : valor mínimo do nível de cinza para a imagem nova

G_{MAX} : valor máximo do nível de cinza para a imagem velha

G_{MIN} : valor mínimo do nível de cinza para a imagem velha

Utilizando a equação acima o nível de cinza das imagens foi reduzido de 256 para 64. Isso reduz, relativamente, a sensibilidade à condição de iluminação [Unser, 1986]. O algoritmo 7.1 ilustra essa etapa:

Algoritmo 7.1 – Método da transformação linear nos níveis de cinza para 64 níveis.

1. Início
2. Encontrar os valores dos níveis de cinza máximo (G_{MAX}) e mínimo (G_{MIN}) da imagem velha
3. Calcular a nova imagem

$$F(i, j) = \frac{63 \cdot (G(i, j) - G_{MIN})}{(G_{MAX} - G_{MIN})}$$
4. Fim.

As figuras 7.7 e 7.8 mostram a diferença entre uma placa adquirida pela câmera sem enriquecimento e a mesma placa após o enriquecimento.

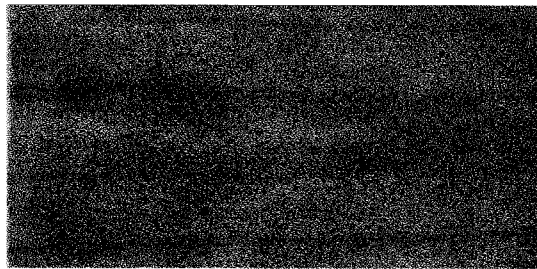


Figura 7. 7 - Placa sem enriquecimento.

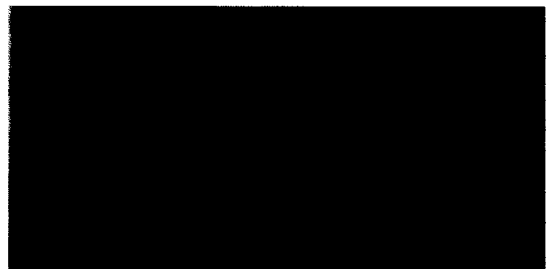


Figura 7. 8 - Placa após enriquecimento.

Vale ressaltar que foram adquiridas características antes e depois do enriquecimento e as mesmas se mostraram muito confusas mesmo depois do enriquecimento. Isso levou a procurar uma binarização da imagem através da escolha de um limiar (“threshold”). O método denominado seleção iterativa foi o melhor no que se refere à velocidade e resultados. O método inicia supondo que a média de uma região é o “threshold” (t_1) procurado, depois é feita a média dos valores menores que t_1 (que seria o fundo (t_b)) e dos valores maiores que t_1 (que seria o objeto (t_o)). Depois disso é calculada a média (t_2) entre a média do objeto (t_o) e do fundo (t_b). Se

o valor t_2 for igual ao “threshold” inicial (t_t), então esse é o “threshold” procurado, caso contrário ao “threshold” inicial (t_t) será igual ao valor de t_2 e o processo reinicia a partir dos cálculos das médias do objeto e do fundo. O processo é ilustrado no algoritmo 7.2.

Algoritmo 7.2 – Procura do “threshold” usando a seleção iterativa

1. Início
2. Zerar o “threshold” do objeto (t_o), o “threshold” do fundo (t_b), e o “threshold” entre fundo e objeto (t_2).
3. Faça Pare = FALSO
4. Enquanto Pare = FALSO faça
 - Calcular média do objeto (t_o) para os “pixels” $\geq t_t$
 - Calcular média do fundo (t_b) para os “pixels” $< t_t$
 - Faça $t_2 = (t_o + t_b)/2$
 - Se $t_2 = t_t$, faça Pare = VERDADEIRO
 - Senão continue
5. Fim.

O programa em linguagem C desse algoritmo pode ser visto em Parker [Parker, 1994]. As figuras 7.9 e 7.10 mostram o resultado da aplicação desse “threshold”, ressaltando, porém, que nem todas as imagens responderam da mesma forma ao algoritmo. Uma vantagem da seleção iterativa do “threshold” é apresentar os mesmos resultados independentemente da iluminação, caso esta não sature a peça em análise.

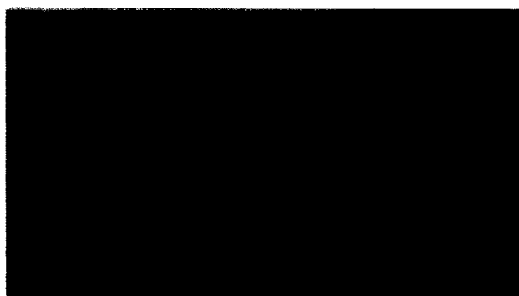


Figura 7. 9 - Placa sem binarização.



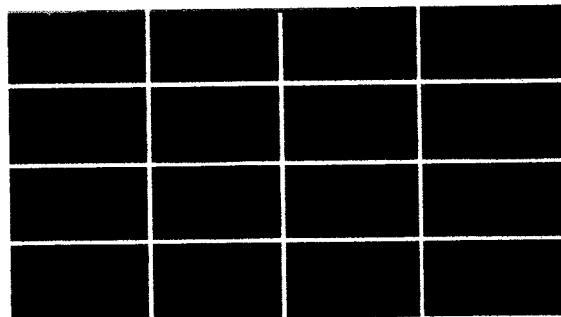
Figura 7. 10 - Placa depois da binarização

7.2.3 – Extração de características

Através dos artigos de Connors [Connors et al, 1983], Bustler [Bustler et al, 1989] e Sobey [Sobey & Semple, 1989], verifica-se que é difícil, senão impossível, classificar as placas de madeira realizando apenas um processamento global. Por isso, e também pelos resultados das características obtidas da placa inteira, a solução adotada foi a de sub-dividir a placa em regiões menores. A redução de domínio de inspeção garantiria que as características nas regiões possam ser mais facilmente extraídas por terem maior similaridade.

Foram analisadas as curvas de cada característica em janelas de 64 x 64 (8 regiões), 64 x 32 (16 regiões), 32 x 32 (32 regiões) e 64 x 14 (36 regiões). Os melhores resultados ocorreram na janela de 64 x 32 (16 regiões) (fig.7.11). Nas janelas menores, os resultados para a classe A (placa sem defeito) são bons, mas as características de outras classes são

confusas por causa do tamanho reduzido das janelas e porque os resultados dependem muito de como a característica (áreas mais escuras)



se distribui na janela.

Figura 7. 11 - Placa dividida em 16 regiões.

A proposta desta tese é o desenvolvimento de uma nova metodologia aplicando um sistema neuro-fuzzy que da mesma forma que os classificadores humanos, possa realizar a separação das placas de madeira, mas de uma forma automática.

Em cada região foram extraídas características baseadas em estatística de primeira e segunda ordem. Os descritores de textura de segunda ordem foram extraídos do histograma da soma e da diferença proposto por Unser [Unser, 1986]. Esses descritores de segunda ordem antes da binarização foram superiores aos descritores de primeira ordem, mas mesmo assim não conseguiram uma boa classificação. Após a binarização, os descritores de primeira ordem conseguiram uma classificação desejável, e os descritores de segunda ordem tiveram um desempenho mais pobre. Portanto como características de entrada foram utilizadas a média, a variância, a entropia, o momento da diferença inversa e a contagem do número de “pixels” escuros até o valor de “threshold” da placa. Esta última para conhecer a proporção de características presentes na janela. Os algoritmos de 7.3 a 7.8 descrevem o método para se obter essas características:

Algoritmo 7.3 – Cálculo do Histograma

1. Início
2. Para $i=0$ até o (número_de_tons_de_cinza - 1) Faça $p(i) = 0$
3. Ler da “frame-grabber” e colocar no buffer[tamanho_janela_X * tamanho_janela_Y]
4. Para $i=0$ até (tamanho_janela_X * tamanho_janela_Y)
 - Faça incremente $p[\text{buffer}(i)]$
 - Fim i
5. Para $i=0$ até (número_de_tons_de_cinza - 1)
 - Faça $p(i) = p(i)/(\text{tamanho_janela_X} * \text{tamanho_janela_Y})$
 - Fim i
6. Fim

Algoritmo 7.4 – Cálculo da média

1. Início
2. Fazer Soma=0
3. Fazer o histograma ($p(i)$) da janela
4. Para $i=0$ até (número_de_tons_de_cinza - 1)
 - Fazer Soma = Soma + $i * p(i)$
 - Fim i
5. Fazer Média_janela = Soma/(tamanho_janela_X * tamanho_janela_Y)
6. Fim.

Algoritmo 7.5 – Cálculo da variância (indica o quanto a imagem está dispersa)

1. Início
2. Fazer Variância_janela = 0
3. Fazer histograma (p(i)) da janela
4. Calcular a média da janela
5. Para i=0 até (número_de_tons_de_cinza - 1)
 - Fazer Variância_janela = Variância_janela + (i - Média_janela)² * p(i)
 - Fim i
6. Fim

Algoritmo 7.6 – Cálculo da entropia (mede a não uniformidade da imagem)

1. Início
2. Fazer Entropia_janela = 0
3. Fazer histograma (p(i)) da janela
4. Para i=0 até (número_de_tons_de_cinza - 1)
 - Se p(i)≠0 Fazer Entropia_janela = Entropia_janela + p(i).log(p(i))
 - Fim i
5. Fim

Algoritmo 7.7 – Cálculo do momento da diferença inversa (IDM) (mede a uniformidade local)

1. Início
2. Fazer IDM_janela = 0
3. Fazer histograma (p(i)) da janela
4. Para i=0 até (número_de_tons_de_cinza - 1)
 - Fazer IDM_janela = IDM_janela + p(i)/(1+i²)
 - Fim i
5. Fim

Algoritmo 7.8 – Cálculo do número de “pixels” escuros (fornece a proporção de característica presente na imagem)

1. Início
2. Fazer Contagem_pixel = 0
3. Escolher valor_limite = “threshold” calculado pela seleção iterativa (no qual os níveis de cinza abaixo desse valor será considerado característica)
4. Fazer o histograma (p(i)) da janela
5. Para i=0 até valor_limite
 - Fazer Contagem_pixel = Contagem_pixel + p(i)
 - Fim i
6. Fim

7.2.4 – Definição da rede neural a ser empregada

Usam-se redes híbridas para modelar sistemas de redes neurais, ou seja, criar uma macro-estrutura completa constituindo de múltiplas redes neurais interagindo, cada uma realizando alguma tarefa vital e única na resolução de um problema complexo [Loesch & Sari, 1996]. As redes híbridas podem ser de diferentes tipos como mostra a figura 7.12 [Loesch & Sari, 1996].

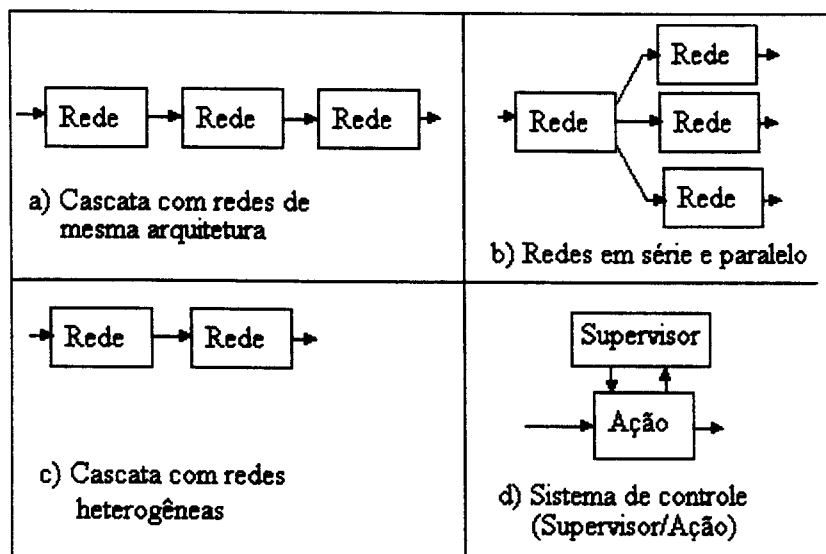


Figura 7.1 2 - Tipos de arquitetura híbrida.

A figura 7.12 (d) é uma arquitetura de controle na qual uma rede influencia nos pesos da outra rede [Loesch & Sari, 1996]. Nota-se nessas quatro arquiteturas que o vetor de características alimenta apenas uma rede. Propomos nesta tese com que cada rede neural trabalhe independentemente da outra, e também que cada uma trabalhe com uma única característica. Isso é inspirado no funcionamento biológico no qual os neurônios agrupam-se para formar o sistema nervoso central composto de

uma variedade de estruturas identificáveis anatomicamente cada uma desempenhando possivelmente uma função específica [Kovács, 1997].

Como a placa de madeira é dividida em 16 regiões, cada rede pode trabalhar um vetor de característica específico, assim uma rede pode manipular o vetor das médias, enquanto outro trata com o vetor de entropia. Uma vantagem dessa metodologia é que o vetor de entrada terá o mesmo domínio e também a possibilidade de trabalhar com arquiteturas diferentes de redes escolhendo aquela que melhor se adaptar a certas características.

Cada rede fornece sua classificação de acordo com a característica escolhida. A arquitetura de rede utilizada é o “perceptrons” múltiplas camadas (MLP) com algoritmo de “backpropagation” possuindo dezesseis nós de entradas referentes a cada janela, cinco nós de saídas referentes às classes. Os nós da camada escondida foram escolhidos de forma interativa partindo de 8, 10, 15, 20 e 18. Loesch [Loesch & Sari, 1996] recomenda variar o número de nós partindo de uma média geométrica entre o número de nós de entradas e os de saídas. Se a camada escondida possuir poucos nós, a rede fica incapaz de criar limites de decisão complexos, enquanto um número excessivo na camada escondida acarreta perda da capacidade de generalização da rede [Loesch & Sari, 1996]. Pelos resultados obtidos as redes responderam melhor com 18 nós na camada escondida.

O algoritmo 7.9 mostra como treinar e testar a rede.

Algoritmo 7.9 – Treinamento e teste de redes neurais MLP com “backpropagation”**TREINAMENTO**

1. Início
2. Escolher o número de nós das camadas de entrada, escondida e de saída
3. Escolher o erro e o número_de_interações_total para a parada do treinamento
4. Inicializar pesos w_{ij} com valores pequenos $[-1,1]$, fazer interação = 0 e erro da rede $E_p = 0$
5. Selecionar um par de treinamento $\{x^p, t^p\}$ (entrada, saída) e computar para cada camada q

$$O_j^q = f[\sum_i O_i^{q-1} \cdot w_{ij}^q] \quad \text{se } f \text{ for sigmoidal: } O_j^q = 1/(1+\exp\{-k[\sum_i O_i^{q-1} \cdot w_{ij}^q]\})$$

Notando que $O_i^0 = x_i$

6. Usar o valor O_j^q computado na camada final e seu correspondente valor alvo t_j^p para computar a quantia delta e o erro do modelo

$$\delta_j^q = (O_j^q - t_j^p) f'(H_j^q) \quad \text{se } f \text{ for sigmoidal: } \delta_j^q = k \cdot O_j^q (1 - O_j^q) (t_j^p - O_j^q)$$

para todos j usando o modelo P .
 Erro do modelo $E_p = E_p + \frac{1}{2} \cdot (t_j^p - O_j^q)^2$

7. Computar a regra delta para cada uma das camadas precedentes usando

$$\delta_i^{q-1} = f'(H_i^{q-1}) \sum_j \delta_j^q w_{ji}^q \quad \text{se } f \text{ for sigmoidal: } \delta_i^{q-1} = K \cdot O_i^{q-1} (1 - O_i^{q-1}) \sum_j \delta_j^q w_{ji}^q$$

para todos os nós j nas camadas $q = Q, Q-1, \dots, 2$

8. Atualizar os pesos

$$w_{ij} = w_{ij} + \Delta w_{ij}^q \quad \text{onde } \Delta w_{ij}^q = \eta \delta_i^q O_i^{q-1}$$

9. Retornar ao passo 5 e repetir para cada amostra do conjunto de treinamento
10. Incrementar interações em uma unidade e fazer $E_p = E_p/\text{número_de_amostras}$
11. Se o erro total do conjunto (E_0) for menor que o erro escolhido ou se o número de interações for maior que o número_de_interações_total parar, caso contrário rerepresentar novamente o modelo partindo do passo 5.
12. Salvar os pesos w_{ij}

Fim

TESTE

1. Início
2. Carregar os pesos w_{ij}
3. Apresentar o vetor de características e computar as saídas nas camadas

$$O_j^q = f[\sum_i O_i^{q-1} \cdot w_{ij}^q] \quad \text{se } f \text{ for sigmoidal: } O_j^q = 1/(1+\exp\{-k[\sum_i O_i^{q-1} \cdot w_{ij}^q]\})$$

Sendo que $O_i^0 = x_i$

4. Fim

7.2.5 – Lógica “fuzzy”

Cada rede neural fornece um valor de saída para cada classe variando de 0 a 1. O objetivo nessa etapa é analisar os dados das redes neurais e combiná-los para obter um refinamento final da classificação. Como cada rede possui 5 saídas, é fácil combinar as saídas de duas redes utilizando a lógica “fuzzy”, o que implica em 25 regras. Caso houvessem mais redes, a utilização da lógica “fuzzy” poderia se tornar impraticável, como exemplo se fossem utilizadas 4 redes com 5 saídas cada uma, haveria a necessidade de se gerar 5^4 (625) regras.

As cinco saídas de cada rede servem como valores de pertinência de entrada das classes para o sistema de regra “fuzzy” que utiliza o raciocínio de Mandami descrito no cap.3 – item 3.7.2.1. A figura 7.13 mostra o diagrama desse sistema.

As regras foram construídas dando preferências às classes que obtiveram

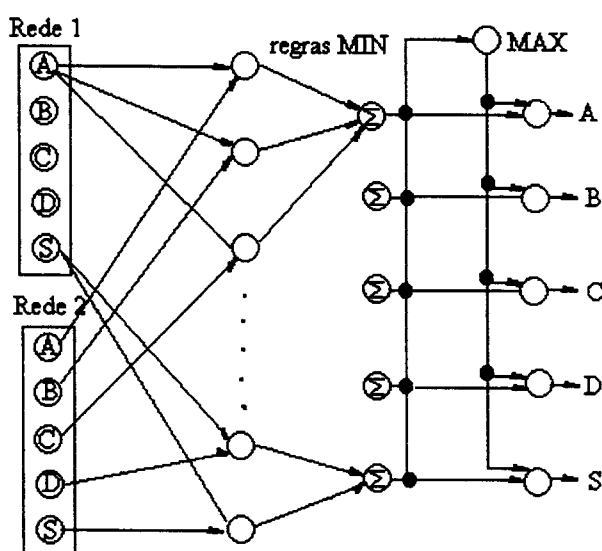


Figura 7.13 - Regras "fuzzy" MIN-MAX.

melhor desempenho nas saídas das redes neurais. Como exemplo, a classe A possui o menor erro entre as classes, portanto todos os valores das regras em que a classe A está envolvida são computados na classe A. O algoritmo 7.10 mostra o raciocínio de Mandami (ou MIN-MAX).

Algoritmo 7.10 – Raciocínio de decisão “fuzzy”

1. Início
2. Ler os valores de pertinências das redes 1 e 2 ($\mu_{A1}, \dots, \mu_{S1}, \mu_{A2}, \dots, \mu_{S2}$)
3. Aplicar o operador MIN-MAX
 - 3.1 operador MIN

$$R[A] = \min(\mu_{A1}, \mu_{A2})$$

$$R[B] = \min(\mu_{B1}, \mu_{B2})$$

$$R[C] = \min(\mu_{C1}, \mu_{C2})$$

$$R[D] = \min(\mu_{D1}, \mu_{D2})$$

$$R[S] = \min(\mu_{S1}, \mu_{S2})$$

$$R[A] = R[A] + \min(\mu_{A1}, \mu_{X2}) \text{ onde } x = B, C, D \text{ e } S *$$

$$R[A] = R[A] + \min(\mu_{X1}, \mu_{A2}) \text{ onde } x = B, C, D \text{ e } S *$$
Outras regras para ser acumulada nas outras classes dependendo do menor erro entre as saídas das redes.....
 - 3.2 operador MAX
 escolher o maior valor acumulado entre as classes
 Saída desejada = maior $R[X]$ onde $X = A, B, C, D \text{ e } S$
4. Fim.

Com referência ao algoritmo anterior vale ressaltar que algumas regras poderiam ser desprezadas, uma vez que seus valores são praticamente nulos, como por exemplo a regra entre as classes A e S, pois é praticamente impossível uma característica indicar que uma placa não apresenta defeito nenhum, enquanto a outra indicar que a mesma possui um grande número de defeitos. A classificação da placa é fornecida pela saída da lógica “fuzzy”.

Outra variação na metodologia proposta foi a fuzzificação de cada nó de saída das redes neurais utilizando 2, 3 e 4 variáveis lingüísticas para cada classe, sendo que os resultados pouco diferiram entre si e o que obteve melhor desempenho foi aquele que utilizou três variáveis lingüísticas (fig. 7.14).

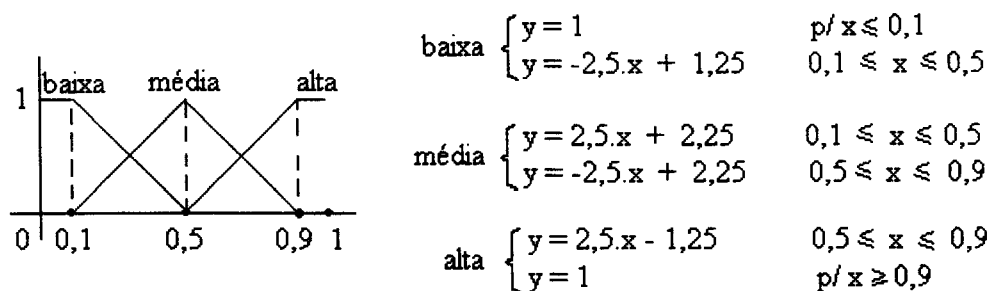


Figura 7.14 - Variáveis linguísticas para cada classe.

7.2.5.1 – Resultados obtidos com os sistemas propostos

Para cada rede neural das metodologias propostas foi exigido um erro de 0,01% ou cem mil interações para o critério de parada das redes. Os erros estacionaram num patamar de 0,035%. A tabela 7.1 mostra os ensaios realizados no conjunto das 199 placas de teste utilizando as saídas das redes neurais diretamente nas regras “fuzzy”, enquanto que a tabela 7.2 mostra a variação aplicada na metodologia com o uso de três variáveis linguísticas nas saídas das redes neurais.

Tabela 7.1 – Número de placas coincidentes entre os inspetores humanos e as características das redes neurais combinadas com lógica “fuzzy” de um total de 199 placas, utilizando diretamente as saídas das redes neurais nas regras “fuzzy”.

Inspetores	I-P	I-E	I-V	V-P	I-M	E-P	M-P	M-V	E-M	E-V
1	132	131	129	127	127	126	127	124	123	121
2	127	123	121	122	117	119	117	116	118	121
3	128	128	124	126	126	128	126	120	122	126

Onde: V-variância; P-número de “pixels” escuros; I-IDM; E-entropia e M-média.

Tabela 7.2 – Número de placas coincidentes entre os inspetores humanos e as características das redes neurais combinadas com lógica “fuzzy” de um total de 199 placas, com o uso de três variáveis linguísticas nas saídas das redes neurais.

Inspetores	I-P	I-M	M-P	I-E	E-P	I-V	V-P	E-V	M-V	E-M
1	134	130	129	128	128	127	125	123	122	121
2	127	123	123	122	122	120	120	121	117	115
3	133	128	127	128	128	125	125	123	125	121

Onde: V-variância; P-número de “pixels” escuros; I-IDM; E-entropia e M-média.

Os resultados das tabelas mostram que os sistemas que contém as características de IDM tiveram um melhor desempenho. Em geral, tanto o sistema que utiliza as saídas das redes neurais diretamente nas regras “fuzzy” quanto o sistema que aplica a fuzzificação nas saídas das redes neurais obtiveram resultados semelhantes, com uma pequena vantagem para esta última. As tabelas 7.3 e 7.4 mostram os resultados dos dois métodos propostos utilizando como características de entrada o momento da diferença inversa (IDM) e número de “pixels” escuros comparado com os três inspetores.

Tabela 7.3 – Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador de redes neurais com lógica “fuzzy” (IDM e número de “pixels” escuros), utilizando diretamente as saídas das redes neurais nas regras “fuzzy”.

Inspetor	A	B	C	D	S	TOTAL
1	9	27	27	25	44	132
2	9	19	26	32	41	127
3	9	24	23	30	42	128

Tabela 7.4 – Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador de redes neurais com lógica “fuzzy” (IDM e número de “pixels” escuros), utilizando a etapa de fuzzificação nas saídas das redes neurais.

Inspetor	A	B	C	D	S	TOTAL
1	9	26	28	26	45	134
2	9	20	27	30	41	127
3	9	25	25	31	43	133

7.3 – Outros Classificadores

Para que se pudesse ter parâmetros de comparação entre os resultados obtidos através da proposta desta tese e outros métodos de classificação, testes com classificadores tradicionais foram realizados.

7.3.1 – Classificador neural

Foi implementado um classificador utilizando apenas uma rede neural. A figura 7.15 mostra o esquema testado.



Figura 7. 15 - Diagrama de blocos do classificador neural.

Os passos de aquisição da imagem, pré-processamento e extração de características são idênticas aos descritos nos itens 7.2.1, 7.2.2 e 7.2.3 deste capítulo.

A rede neural utilizada aqui possui 32 nós de entradas que equivalem aos dois vetores de características escolhidos e 5 nós de saídas referentes a cada classe. Os nós escondidos perfazem um total de 50, escolhidos depois de testes com 30, 40, 50 e 60. Neste método a rede neural já fornece a classificação final. Nota-se aqui uma quantidade muito grande de ligações (1850 ligações entre nós) enquanto que no método anterior cada rede necessitou de 378 ligações (756 ligações no total). O algoritmo para treinamento e teste segue o mesmo modelo do algoritmo 7.9.

Também foi definido para esta rede neural, como critérios de parada, um erro menor que 0,01% ou o número de 100000 interações. O erro não foi atingido, e após as cem mil interações o erro dos ensaios ficaram em torno de 0,038%. Esta rede possui 32 nós de entrada correspondentes às janelas de duas características. A tabela 8.6 mostra os resultados dos ensaios com as características de entrada.

Tabela 7.5 – Número de placas coincidentes entre os inspetores humanos as características da rede neural de um total de 199 placas.

Inspetores	I-P	I-M	I-V	I-E	E-P	M-P	E-M	V-P	M-V	E-V
1	130	128	125	125	130	125	116	115	113	115
2	120	119	117	119	118	118	106	107	106	107
3	125	123	120	118	125	118	113	116	110	116

Onde: V-variância; P-número de “pixels” escuros; I-IDM; E-entropia e M-média.

Os resultados da tabela acima, mostram que a rede neural respondeu melhor às entradas que tinham as características de IDM em seu vetor. A tabela 7.6 mostra o resultados da rede neural, com vetor de entrada de IDM e número de “pixels” escuros, comparado com os três inspetores.

Tabela 7.6 – Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e a rede neural (IDM e número de “pixels” escuros).

Inspetor	A	B	C	D	S	TOTAL
1	7	27	26	25	45	130
2	7	19	24	29	41	120
3	8	25	23	29	40	125

7.3.2 – Classificador Estatístico

As figuras 7.16 à 7.25 mostram as curvas de distribuição normalizadas e a distribuição espacial dos valores das características extraídas do conjunto de treinamento para a região 6. As curvas das outras regiões se encontram no programa “PLOTTER” gravado no cd-rom anexo.

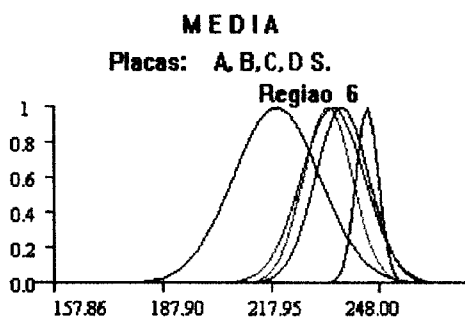


Figura 7.16 - Distribuição dos valores de média.

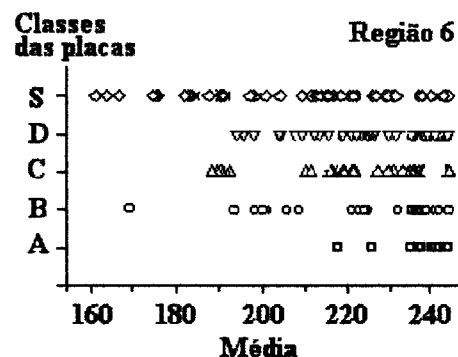


Figura 7.17 - Distribuição espacial dos valores de média.

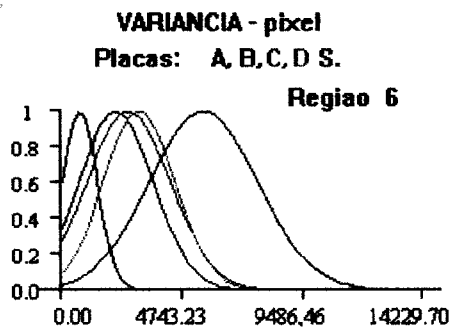


Figura 7. 18 - Distribuição dos valores de variância

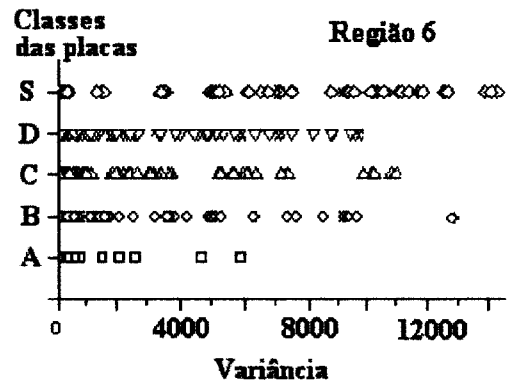


Figura 7. 19 - Distribuição espacial dos valores de variância.

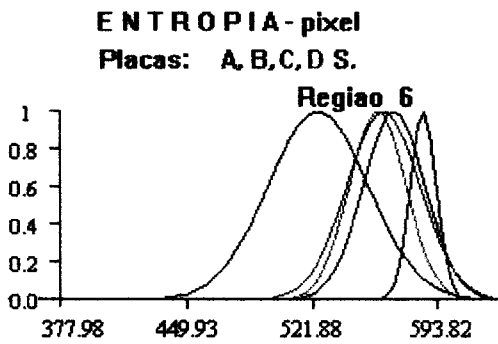


Figura 7. 20 - Distribuição dos valores de entropia.

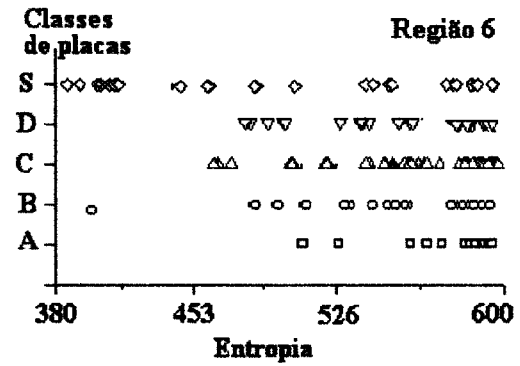


Figura 7. 21 - Distribuição espacial dos valores de entropia.

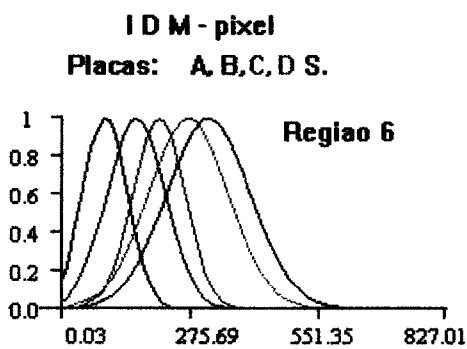


Figura 7. 22 - Distribuição dos valores de IDM

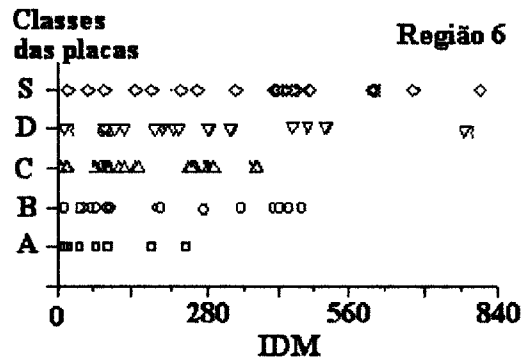


Figura 7. 23 - Distribuição espacial dos valores de IDM.

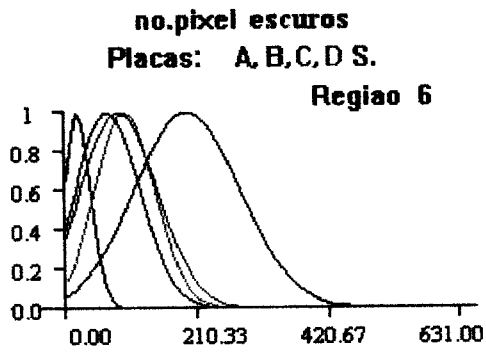


Figura 7.24 - Distribuição dos valores de "pixels" escuros.

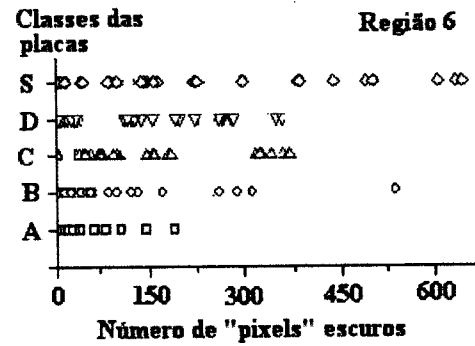


Figura 7.25 - Distribuição espacial dos valores de "pixels" escuros.

Pelas curvas das características percebe-se que fica muito difícil, senão impossível, classificar as placas usando a função densidade normal no classificador estatístico. As funções de características possuem uma grande área sobreposta entre as classes.

Mesmo assim, foi testado um classificador estatístico com um vetor de duas características, usando a densidade normal multivariável:

$$p(x/w_i) = \frac{1}{2 \cdot \pi^{d/2} \cdot |\Sigma|^{1/2}} \exp \left[\frac{-1}{2} (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) \right]$$

Onde: x é o vetor de características

D - dimensão do vetor

μ - média

Σ - matriz de covariância

O vetor média e a matriz de covariância são dados por:

$$\bar{u} = \frac{1}{n_i} \cdot \sum_k^{n_i} x_k \quad \Sigma = \frac{1}{n_i - 1} \cdot \sum_{k=1}^{n_i} (x_k - \bar{\mu}) \cdot (x_k - \bar{\mu})^t$$

O algoritmo 7.11 mostra a implementação do classificador estatístico

Algoritmo 7.11 – Classificador estatístico

TREINAMENTO

1. Início
2. Ler vetor de características
3. Calcular a média e a matriz de covariância dos vetores para cada classe i

$$\bar{u}_i = \frac{1}{n_i} \sum_k^m x_k \quad \Sigma_i = \frac{1}{n_i - 1} \sum_{k=1}^m (x_k - \bar{\mu})(x_k - \bar{\mu})^t$$

4. Calcular o determinante e a inversa da matriz de covariância
5. Salvar dados
6. Fim.

TESTE

1. Início
2. Ler dados de média, matriz de covariância inversa e o determinante da matriz de covariância das classes i
3. Ler vetor desconhecido
4. Calcular para cada classe i

$$p(x/w_i) = \frac{1}{2 \cdot \pi^{d/2} \cdot |\Sigma|^{1/2}} \exp \left[\frac{-1}{2} (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) \right]$$

5. Escolher k entre as classes que satisfaça
 $p(x/w_k) > p(x/w_i)$ para $i \neq k$ e $i=1 \dots$ número total de classes
6. Fim

A tabela 7.7 mostra os resultados das características de IDM e do número de “pixels” escuros que forneceram os

Tabela 7.7 – Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador estatístico.

Inspetor	A	B	C	D	S	Total
1	0	20	0	0	54	74
2	0	18	0	0	48	66
3	0	26	0	0	50	76

melhores resultados para esse classificador. Os resultados são comparados com os inspetores humanos. O classificador conseguiu discriminar apenas duas classes (B e S). Para algumas características (entropia e média, entropia e variância) o determinante da matriz de covariância foi nulo. Isso acontece, quando uma componente do vetor de característica x tem variância zero, ou quando duas componentes são idênticas [Duda & Hart, 1973].

7.3.3 – K-vizinhos mais próximos

O método do K-vizinhos mais próximos (KNN) utilizado neste trabalho foi implementado utilizando a distância Euclidiana. O número de vizinhos foram escolhidos interativamente. O algoritmo 7.12 implementa este método.

Algoritmo 7.12 – Classificador do k-vizinhos mais próximos

1. Início
2. Ler vetor de treinamento (y)
3. Ler um vetor desconhecido (x)
4. Zerar contagem_classe_W
5. Para i=0 até (número de amostra de treinamento) Fazer
 - Calcular distância Euclidiana entre vetor desconhecido e os vetores de treinamento e armazenar na estrutura: $d_i(x,y)$ = que contém o valor da distância
 - rótulo = que contém o nome da classe
$$d_i(x,y) = \left[\sum_j (x_j - y_j)^2 \right]^{1/2}$$
 - Fim i
6. Para i=0 até (número de amostra de treinamento) Fazer
 - Ordenar $d_i(x,y)$ em ordem crescente
 - Fim i
7. Para i=0 até (k vizinho mais próximo)
 - verificar o rótulo da classe e incrementar a contagem_classe correspondente
 - Fim i
8. Classificar a classe vencedora como aquela que possui o maior número de contagem
9. Fim

O classificador dos k-vizinhos mais próximos foi realizado sobre a placa inteira e também sobre as janelas da placa. Para classificar a placa usando os k-vizinhos mais próximos nas janelas, foi feita a contagem dos rótulos das janelas e o rótulo vencedor classificou a placa. Nota-se que os valores obtidos tanto pelo classificador KNN sobre a placa inteira quanto ao aplicado nas janelas são praticamente iguais e sua diferença se deve ao fato da disposição das características nas janelas, uma vez que a placa recebe o rótulo da classe que obteve a maior frequência. Na classificação foram permutadas as características duas a duas. As

tabelas 7.8 e 7.9 mostram os resultados entre as características empregadas para o classificador KNN sobre a placa inteira e sobre a placa com janelas, respectivamente.

Tabela 7.8 – Número de placas coincidentes entre os inspetores humanos e as características do classificador KNN da placa inteira de um total de 199 placas.

Inspetores	V-P	M-V	I-V	E-V	E-M	M-P	E-P	E-I	I-M	I-P
1	94	90	92	91	83	81	83	83	81	82
2	95	100	94	90	82	80	80	79	80	78
3	91	90	89	90	77	74	76	75	75	74

Onde: V-variância; P-número de “pixels” escuros; I-IDM; E-entropia e M-média.

Tabela 7.9 – Número de placas coincidentes entre os inspetores humanos e as características do classificador KNN da placa com janelas de um total de 199 placas.

Inspetores	V-P	M-V	I-V	E-V	E-M	M-P	E-P	E-I	I-M	I-P
1	82	82	82	83	80	82	84	83	83	83
2	96	92	95	93	92	93	90	95	95	90
3	93	90	91	91	89	92	90	91	91	90

Onde: V-variância; P-número de “pixels” escuros; I-IDM; E-entropia e M-média.

Pelas tabelas nota-se que os k-vizinhos mais próximos da placa inteira tem uma maior coincidência com o inspetor 1, enquanto os k-vizinhos mais próximo da placa com janela tem coincidência maior com o inspetor 2 e 3. Nos testes onde a característica de variância esteve envolvida, o desempenho foi melhor. Desses resultados, o ensaio envolvendo a variância e número de “pixels” escuros dos k-vizinhos mais próximos da placa toda tem uma melhor coincidência com os três inspetores humanos, portanto esse ensaio será usado para comparação entre as técnicas.

A tabela 7.10 mostra os resultados do ensaio com as características de variância e número de “pixels” escuros comparados com a classificação dos inspetores.

Tabela 7.10 – Distribuição por classes do número de placas coincidentes entre os três inspetores humanos e o classificador KNN da placa toda (variância e número de “pixels” escuros).

Inspetor	A	B	C	D	S	TOTAL
1	8	13	17	20	36	94
2	6	10	22	22	35	95
3	7	14	17	20	33	91

7.4 – Conclusão

Neste capítulo foram descritos os métodos utilizados para classificações da qualidade das placas de madeira. Dos métodos propostos, já era esperado que o classificador estatístico fosse pobre, uma vez que as curvas de características entre as classes estão sobrepostas.

O método do k-vizinhos mais próximos conseguiu em média uma coincidência com os inspetores de aproximadamente 94 placas em um total de 199 ($\cong 47\%$), o que demonstra a necessidade de se aplicar outras metodologias diferentes das tradicionais para se obter melhores resultados.

A rede neural obteve bons resultados, mesmo assim esses resultados ficaram abaixo daqueles obtidos pelas metodologias propostas. Além disso, a rede neural possui uma grande quantidade de ligações, o que pode dificultar uma possível manutenção (ou atualização).

O classificador de rede neural e o k-vizinhos mais próximos servirão de comparação com o método proposto que utiliza cada rede neural trabalhando com apenas uma característica e com a variação desta proposta, utilizando uma etapa de fuzzificação nas saídas das rede neurais. As saídas das redes neurais são combinadas através da lógica “fuzzy”. Outra proposta do método é a extração de características após um processo de binarização da imagem, o que faz com que o sistema fique mais imune às variações de iluminação.

8 – RESULTADOS E CONCLUSÕES

8.1 – Introdução

No ambiente industrial é exigida que uma tarefa, dependendo da aplicação, seja executada da maneira mais rápida possível. Às vezes, a velocidade do sistema deve ser um pouco sacrificada a fim de se conseguir uma melhor precisão.

Com esse intuito, o objetivo desta tese foi o de propor uma nova metodologia que permitisse classificar de forma rápida e precisa, placas de madeira para a fabricação de lápis.

Foram utilizados dois conjuntos de placas de madeiras para análise dos métodos. Um conjunto para treinamento composto por 120 placas previamente classificadas e outro conjunto para teste composto de 199 placas. Esses conjuntos foram os mesmo utilizados por Rodrigues [Rodrigues, 1998], portanto os resultados aqui, além dos métodos descritos, serão confrontados com os obtidos pelo trabalho de Rodrigues [Rodrigues, 1998].

8.2 – Resultados

O anexo 1 apresenta os resultados da classificação do conjunto de placas para teste. Esses resultados foram realizados por três técnicos especialistas, pelos dois algoritmos propostos por Rodrigues [Rodrigues, 1998] (método da co-ocorrência e método da redução), e pelos classificadores de Bayes, k-vizinhos mais próximos, rede neural e pelas redes neurais combinadas com lógica “fuzzy”.

Dos algoritmos mostrados nesta tese, o anexo 1 mostra os resultados das melhores características. Foi evitado utilizar a classificação alternativa dos inspetores, uma vez que ela somente serve para aumentar as porcentagens de coincidências entre as classes. O ideal seria não ter essa segunda classificação, mas sim haver vários inspetores para classificar o mesmo lote de placas a fim de se obter uma melhor comparação.

8.3 – Comparações entre os métodos

A tabela 8.1 mostra as frequências de ocorrências das classes para cada método implementado. Pelos resultados da tabela 8.1, nota-se que o classificador de Bayes não consegue discriminar as classes, isto por causa da sobreposição entre classes. Os outros classificadores conseguem uma melhor distribuição entre as classes e suas diferenças se devem ao grau de dificuldade do problema em análise. As placas em análise possuem vários níveis de cinza distribuídos da forma mais aleatória

possível, isto faz com que os valores de características sejam bem próximos uns dos outros.

Tabela 8.1 – Frequências de ocorrência de classes nos métodos analisados.

	Classe A	Classe B	Classe C	Classe D	Classe S	Total
Inspetor 1	12	33	49	49	56	199
Inspetor 2	16	30	52	52	49	199
Inspetor 3	14	37	54	54	51	199
Co-ocorrência *	15	37	51	51	47	199
Redução *	14	42	45	45	45	199
Estatístico	0	69	0	0	130	199
KNN	20	30	44	52	53	199
Rede Neural	9	42	39	47	62	199
Sistema Neuro-fuzzy (1)	11	38	43	47	60	199
Sistema Neuro-fuzzy (2)	11	38	43	46	61	199

(*) obtidos da tese de Rodrigues [Rodrigues, 1998].

(1) - Sistema neuro-fuzzy, utilizando diretamente as saídas das redes neurais nas regras "fuzzy"

(2) - Sistema neuro-fuzzy, aplicando uma etapa de fuzzificação nas saídas das redes neurais

A tabela 8.2 mostra a coincidências entre os inspetores humanos, e entre estes e os métodos implementados. As coincidências entre os inspetores humanos variam de 69,85% a 73,37%. Os métodos proposto por Rodrigues [Rodrigues, 1998] variam de 56,28% a 63,83%. Os classificadores KNN variam de 45,73% a 47,73%, o classificador estatístico nem foi mostrado por causa de seu fraco desempenho para o problema em questão. O classificador realizado através de rede neural variaram de 60,3% a 65,33%. O classificador proposto com redes neurais trabalhando com apenas uma característica específica para cada rede e utilizando diretamente as saídas das redes nas regras "fuzzy" variaram de 63,81% a 66,33%, enquanto que a variação deste método com a aplicação da etapa de fuzzificação nas saídas das rede neurais variaram de 63,81% a 67,33%.

Tabela 8.2 – Comparação entre os métodos estudados.

Coincidências	Classe A	Classe B	Classe C	Classe D	Classe S	Total de coincidências	Total %
Inspetor 1 Inspetor 2	7	16	34	36	46	139	69,85
Inspetor 1 Inspetor 3	8	22	31	38	47	146	73,37
Inspetor 2 Inspetor 3	12	23	31	37	43	146	73,37
Inspetor 1 * Co-ocorrência	7	23	30	25	36	121	60,80
Inspetor 1 * Redução	7	24	33	26	37	127	63,83
Inspetor 2 * Co-ocorrência	10	15	27	27	33	112	56,28
Inspetor 2 * Redução	8	18	32	26	34	118	59,30
Inspetor 3 * Co-ocorrência	10	25	25	28	36	124	62,31
Inspetor 3 * Redução	9	26	28	27	35	125	62,81
Inspetor 1 KNN	8	13	17	20	36	94	47,23
Inspetor 2 KNN	6	10	22	22	35	95	47,73
Inspetor 3 KNN	7	14	17	20	33	91	45,73
Inspetor 1 Rede neural	7	27	26	25	45	130	65,33
Inspetor 2 Rede neural	7	19	24	29	41	120	60,30
Inspetor 3 Rede neural	7	25	23	29	40	125	62,81
Inspetor 1 Sist.neuro-fuzzy (1)	9	27	27	25	44	132	66,33
Inspetor 2 Sist.neuro-fuzzy (1)	9	19	26	32	41	127	63,81
Inspetor 3 Sist.neuro-fuzzy (1)	9	24	23	30	42	128	64,32
Inspetor 2 Sist.neuro-fuzzy (2)	9	26	28	26	45	134	67,33
Inspetor 2 Sist.neuro-fuzzy (2)	9	20	27	30	41	127	63,81
Inspetor 2 Sist.neuro-fuzzy (2)	9	25	25	31	43	133	66,83

* - Obtidos da tese de Rodrigues [Rodrigues, 1998]

(1) Sistema neuro-fuzzy utilizando diretamente as saídas das redes neurais nas regras "fuzzy"

(2) Sistema neuro-fuzzy aplicando uma etapa de fuzzificação nas saídas das redes neurais

Portanto tanto o sistema proposto quanto a sua variação com o uso da camada de fuzzificação foram os que mais se aproximaram da classificação dos inspetores humanos, seguidos pela rede neural e pelo método de redução.

Vale ressaltar, também, que entre os três inspetores verifica-se que apenas 116 placas coincidem, sendo 6 placas A, 13 placas B, 25 placas C, 31 placas D e 41 placas S. Isso mostra o quanto o problema é subjetivo, pois cada inspetor realiza a classificação baseado em seu conhecimento e em seu estado de espírito. Uma vantagem dos métodos propostos sobre os operadores humanos é a repetibilidade dos resultados, pois apresentando as mesmas placas aos sistemas obtém-se as mesmas respostas, já os operadores humanos dificilmente apresentariam os mesmos resultados.

Outra vantagem dos sistemas automáticos é a padronização do processo, pois numa linha de produção cada operador possui seu próprio conhecimento, o que pode significar que uma placa pode ter várias classificações dependendo do operador. Já o sistema automático trabalha com uma base fixa de conhecimento.

Quanto à velocidade de classificação, as redes neurais combinadas com lógica “fuzzy” possibilitam a análise de 2 placas por segundo, conseguindo classificar em um minuto 120 placas. Isto pode ser pouco no ambiente industrial, mas o sistema montado está limitado à velocidade da esteira e principalmente do micro-computador utilizado (486 DX-2, 66MHz), o que abre uma grande possibilidade para o melhoramento do sistema.

8.4 – Contribuições

O presente trabalho contribui na aplicação na área de classificação de placas de madeira, analisando as características visuais, através de redes neurais e lógica “fuzzy”. Isso faz com que o processo de classificação aproxime-se dos classificadores humanos, mas com repetibilidade e com um padrão definido.

Outra contribuição refere-se à divisão do vetor característica da rede neural, fazendo com que cada característica específica seja tratada na rede própria, com a vantagem do vetor de característica ter o mesmo domínio. Depois as redes são combinadas numa lógica “fuzzy” fornecendo o resultado da classificação.

As combinações dessas contribuições forneceu melhores resultados do que os outros métodos. Além disso, o sistema possui uma arquitetura simples, o que faz que a classificação seja realizada rapidamente, ficando a velocidade do mesmo limitada aos equipamentos utilizados.

8.5 – Conclusão final

Um sistema para classificar qualidade de placas de madeiras é de difícil implementação, em parte por causa das grandes variações de níveis de cinza presentes nas superfícies das placas e também por ser um trabalho subjetivo.

Os problemas envolvem iluminação (como fazer uma iluminação mais estável), seleção e extração das características (quais as melhores características) e a definição de um classificador para tratar essas características.

Um sistema automático é desejável por causa de sua repetibilidade. Outra característica desejável é a padronização, uma vez que serão as mesmas características tratadas com os mesmos critérios.

A metodologia proposta aqui obteve uma melhor classificação quando comparada com outros métodos. Vale ressaltar, também, que além do método proposto por Rodrigues [Rodrigues, 1998], não há na literatura artigos voltados para a classificação de placas de madeira, quanto à qualidade, para a fabricação de lápis. A maioria dos artigos tratam da detecção de defeitos, geralmente para a fabricação de móveis.

8.6 – Sugestões para futuros trabalhos

Algumas sugestões são decorrentes das dificuldades encontradas neste trabalho.

- O conjunto de treinamento deveria ser analisado por um número maior de inspetores, a fim de se conseguir um conjunto mais confiável das classes.
- Realizar o trabalho utilizando imagens coloridas. Os padrões de cores das placas podem trazer informações significantes na classificação.
- Utilizar equipamentos mais modernos a fim de incrementar a velocidade do sistema, trocando o micro-computador 486 DX-2 66MHz por um pentium 2-300MHz, aumentar a velocidade da esteira (que também está limitada pelo micro-computador).

- A troca da placa de aquisição também pode melhorar a velocidade do sistema, já que a utilizada (IRIS) trabalha em DOS com memória RAM de 8Mbytes (o aumento da memória RAM provoca conflito com a placa de aquisição de dados).
- Estudos alterando as arquiteturas das redes podem ser realizados, usando a máquina de Boltzmann ou de Cauchy ou outras redes que são indicadas para classificação de padrões.
- Implementações em “hardware” dedicado (computadores paralelos).

ANEXO 1

TABELA DE CLASSIFICAÇÃO DO CONJUNTO DE 199 PLACAS

Este apêndice mostra as classificações realizadas pelos três inspetores, pelos dois métodos proposto por Rodrigues [Rodrigues, 1998] (método da matriz de co-ocorrência e método de redução), por dois métodos convencionais (Bayes e KNN), e pelos dois métodos propostos nesta tese utilizando sistemas neuro-fuzzy. O primeiro denominado sistema neuro-fuzzy 1 aplica diretamente as saídas das redes neurais nas regras “fuzzy”, enquanto que o segundo (sistema neuro-fuzzy 2) utiliza três variáveis lingüísticas para a fuzzificação das saídas das redes neurais.

Placa	Inspetor 1	Inspetor 2	Inspetor 3	Co-ocorrência	Redução	Bayes	KNN	Rede neural	Sistema neuro-fuzzy 1	Sistema neuro-fuzzy 2
1	B	B	B	B	B	B	B	B	B	B
2	C	C	C	B	B	S	D	C	C	C
3	A	A	B	B	B	B	A	B	A	A
4	C	B	B	B	B	S	B	B	B	B
5	A	A	A	A	A	B	C	A	A	A
6	C	B	B	C	C	S	D	B	B	B
7	A	A	A	A	B	S	B	B	A	A
8	D	D	C	D	B	S	A	B	B	B
9	B	B	B	B	B	S	A	B	B	B
10	B	A	B	B	B	S	B	B	B	B
11	C	B	B	A	A	S	D	B	B	B
12	A	A	A	A	A	B	A	A	A	A

Placa	Inspetor 1	Inspetor 2	Inspetor 3	Co-ocorrência	Redução	Bayes	KNN	rede neural	Sistema neuro-fuzzy 1	Sistema neuro-fuzzy 2
13	B	B	B	B	B	B	A	B	B	B
14	A	A	A	B	A	B	A	A	A	A
15	C	B	B	C	C	B	A	C	C	C
16	C	C	C	C	C	B	C	C	C	C
17	A	A	A	A	A	S	A	A	A	A
18	B	A	A	B	B	S	D	B	B	B
19	D	C	C	C	C	S	S	C	C	C
20	B	B	B	B	B	S	D	S	C	C
21	B	B	B	B	C	S	B	B	B	B
22	C	C	C	C	C	B	D	C	C	C
23	C	C	B	B	B	B	B	B	B	B
24	C	C	C	C	B	B	C	S	D	C
25	B	A	B	A	B	B	C	B	B	B
26	B	B	C	B	B	B	A	B	B	B
27	B	B	C	C	C	S	B	B	B	B
28	B	A	A	A	B	S	B	B	B	B
29	B	A	A	B	B	S	D	B	B	B
30	B	A	A	A	A	B	A	A	A	A
31	A	A	A	A	A	B	A	A	A	A
32	A	B	B	B	B	B	B	B	B	B
33	S	D	D	D	D	S	D	D	D	D
34	C	C	C	D	C	B	C	C	C	C
35	D	D	D	D	D	S	C	C	C	C
36	S	D	D	S	D	S	D	D	D	D
37	S	C	D	C	C	S	B	S	S	S
38	S	S	S	C	C	S	S	S	S	S
39	D	D	D	D	D	S	D	D	D	D
40	S	S	S	S	S	S	S	S	S	S
41	B	B	B	B	B	B	C	B	B	B
42	B	B	C	B	B	S	D	B	B	B
43	B	B	B	B	B	B	B	B	B	B
44	C	C	C	B	B	B	C	C	C	C
45	C	C	C	D	C	B	A	C	C	C
46	C	C	C	C	C	B	B	C	S	C
47	C	C	C	D	C	S	S	C	C	C
48	C	C	C	D	C	B	D	C	C	C
49	B	C	B	B	A	B	B	B	B	B
50	S	D	D	D	D	B	C	S	S	S
51	B	A	A	A	A	S	B	B	B	B
52	B	B	B	B	B	B	B	B	B	B
53	C	C	C	C	D	S	S	C	C	C
54	C	C	C	C	C	B	C	C	C	C
55	D	D	C	C	C	S	C	C	C	C
56	D	C	C	D	C	S	C	C	C	C
57	C	D	D	D	C	S	S	C	C	C
58	C	C	C	C	C	S	C	C	C	C
59	C	B	B	C	B	B	C	C	C	C
60	C	C	C	C	C	S	S	C	C	C
61	C	C	C	C	C	B	C	C	C	C
62	B	B	B	A	A	B	B	B	B	B

Placa	Inspetor 1	Inspetor 2	Inspetor 3	Co-ocorrência	Redução	Bayes	KNN	rede neural	Sistema neuro-fuzzy 1	Sistema neuro-fuzzy 2
63	A	B	B	B	B	B	C	B	C	B
64	D	D	D	D	D	B	C	S	S	S
65	C	C	C	C	C	S	D	C	C	C
66	B	D	C	B	B	B	C	B	D	C
67	A	B	B	B	B	B	A	C	C	B
68	C	C	B	B	C	B	B	D	D	C
69	B	C	B	B	B	B	C	C	C	C
70	C	B	B	C	B	S	S	C	C	C
71	C	C	B	B	B	B	C	C	C	C
72	B	A	A	B	B	B	C	B	B	B
73	C	B	B	C	B	B	C	C	C	C
74	C	B	C	C	C	S	D	D	D	D
75	C	D	D	C	C	S	B	D	D	D
76	C	C	C	C	C	B	C	C	C	C
77	B	A	B	A	A	B	B	A	A	A
78	D	D	C	C	C	S	D	S	S	S
79	D	D	D	D	D	S	D	D	D	D
80	B	B	B	C	B	B	B	D	B	D
81	A	B	A	A	A	B	A	A	A	A
82	A	B	A	A	A	S	A	A	A	A
83	B	C	B	B	B	B	C	B	B	B
84	B	C	B	B	B	B	C	C	C	C
85	B	C	B	B	B	S	C	B	B	B
86	B	C	B	B	B	B	C	B	B	B
87	C	C	B	B	C	S	D	B	B	B
88	B	B	B	B	B	B	B	B	B	B
89	B	B	B	A	A	B	A	B	B	B
90	D	D	D	D	D	B	C	C	C	C
91	D	S	D	S	S	S	S	C	C	C
92	S	D	S	S	D	S	S	S	S	S
93	D	C	D	S	D	B	C	S	S	S
94	C	C	C	C	D	S	C	C	C	C
95	C	C	D	C	C	S	D	D	D	D
96	D	C	C	C	C	B	C	B	B	D
97	C	C	C	C	B	S	D	D	D	D
98	D	D	D	C	D	S	D	S	S	S
99	D	D	D	D	S	S	D	D	D	D
100	S	S	S	D	D	S	S	S	S	S
101	C	C	C	C	C	B	C	D	D	C
102	D	C	D	C	C	S	D	D	D	D
103	S	S	S	C	C	B	C	S	S	S
104	S	S	S	S	S	S	S	S	S	S
105	S	S	S	D	D	S	D	S	S	S
106	S	S	D	S	S	S	D	S	S	S
107	D	D	D	D	D	S	D	D	D	D
108	S	S	D	D	D	S	S	S	S	S
109	C	C	C	C	C	S	B	D	C	S
110	D	D	D	C	C	S	D	D	D	D
111	C	C	C	C	C	S	B	B	B	B
112	D	D	D	D	D	B	D	D	D	D

Placa	Inspetor 1	Inspetor 2	Inspetor 3	Co-ocorrência	Redução	Bayes	KNN	rede neural	Sistema neuro-fuzzy 1	Sistema neuro-fuzzy 2
113	S	S	S	D	D	S	D	S	S	S
114	D	D	D	D	D	S	D	D	D	D
115	C	C	C	D	C	S	C	S	S	S
116	D	D	D	D	D	S	S	S	S	S
117	C	D	D	C	C	S	B	D	B	B
118	C	D	C	D	C	S	B	S	D	D
119	C	C	C	D	C	B	D	C	C	C
120	C	C	D	C	C	B	B	D	D	D
121	D	D	D	D	D	S	S	S	S	S
122	D	C	C	C	D	S	S	S	S	S
123	D	D	D	D	D	S	D	C	C	C
124	C	C	D	C	C	B	C	S	C	D
125	D	D	D	D	C	S	C	D	D	D
126	C	C	C	B	B	B	A	B	C	D
127	D	D	D	C	C	S	D	S	S	S
128	C	D	C	B	B	B	B	C	D	D
129	B	B	B	D	C	S	S	B	B	B
130	D	C	D	C	C	B	D	C	C	C
131	D	D	D	S	S	S	S	D	S	S
132	S	S	S	S	S	S	S	S	S	S
133	S	S	S	S	S	S	S	S	S	S
134	S	D	S	S	D	S	D	S	S	S
135	D	D	D	C	C	B	A	B	D	D
136	D	D	D	D	D	B	C	D	D	D
137	D	D	D	C	C	S	C	D	D	D
138	C	C	D	C	C	S	B	B	B	B
139	B	C	C	B	B	S	A	B	B	B
140	S	D	S	S	S	S	S	D	D	S
141	S	S	S	S	S	S	D	S	S	S
142	S	S	S	S	S	S	S	S	S	S
143	S	S	S	S	S	S	D	S	C	C
144	S	S	S	S	S	S	S	C	C	C
145	S	S	S	S	S	S	S	D	S	S
146	S	S	S	S	S	S	S	S	S	S
147	S	S	S	S	S	S	D	S	S	S
148	S	S	D	D	D	S	S	D	D	D
149	S	S	S	S	D	S	D	S	S	S
150	C	B	C	C	C	B	C	C	C	C
151	D	D	D	D	D	B	C	D	D	D
152	S	S	S	S	D	S	D	S	S	S
153	S	S	S	C	D	S	S	S	S	S
154	D	D	D	D	D	S	D	D	D	D
155	S	S	S	D	D	S	S	S	S	S
156	D	S	S	S	D	S	S	D	S	S
157	D	D	D	S	D	S	S	C	D	D
158	S	D	S	S	S	S	D	D	D	D
159	D	D	D	S	S	B	C	D	D	D
160	S	S	D	D	S	S	S	S	C	C
161	D	C	D	D	C	B	A	D	D	D
162	C	C	C	D	D	S	D	D	D	D

Placa	Inspetor 1	Inspetor 2	Inspetor 3	Co-ocorrência	Redução	Bayes	KNN	rede neural	Sistema neuro-fuzzy 1	Sistema neuro-fuzzy 2
163	D	D	D	S	D	S	S	D	D	D
164	S	S	S	D	S	S	S	D	D	D
165	S	S	S	D	S	S	S	S	S	S
166	D	C	D	D	D	S	B	D	D	D
167	S	S	S	S	S	S	S	S	S	S
168	S	D	S	D	S	S	S	S	S	S
169	D	D	S	S	S	S	D	D	D	D
170	S	S	S	S	S	S	S	S	S	S
171	D	D	D	D	D	S	D	D	D	D
172	S	S	S	D	S	S	S	S	S	S
173	S	S	S	D	S	S	S	D	D	D
174	S	S	S	D	S	S	S	S	D	D
175	D	C	D	C	C	S	C	S	S	S
176	D	D	D	D	S	S	D	S	S	S
177	D	D	D	D	D	B	C	D	D	D
178	D	D	D	D	D	S	D	D	D	D
179	C	D	C	C	C	B	C	D	D	D
180	S	S	S	S	S	S	S	S	S	S
181	S	S	S	S	S	S	S	S	S	S
182	S	S	S	S	S	S	D	S	S	S
183	S	S	S	S	S	S	D	S	S	S
184	D	D	S	D	D	S	D	D	S	S
185	S	S	S	S	S	S	S	S	S	S
186	S	S	S	S	S	S	S	S	S	S
187	S	S	S	S	S	S	S	S	S	S
188	S	S	S	S	S	S	S	S	S	S
189	S	S	S	S	D	S	S	S	S	S
190	S	S	D	S	S	S	S	S	S	S
191	D	D	D	S	S	B	S	S	D	D
192	S	S	S	S	S	S	S	S	S	S
193	S	S	S	D	D	S	D	S	S	S
194	D	D	D	S	S	S	S	D	D	D
195	S	D	S	S	S	S	D	D	D	D
196	S	S	S	S	S	S	B	S	S	S
197	D	S	S	S	D	S	D	S	S	S
198	S	S	S	S	S	S	S	S	S	S
199	S	S	S	S	S	S	S	D	S	S

ANEXO 2

CONTEÚDO DO CD-ROM

DIRETÓRIO	CONTEÚDO
CARACT	programa "PLOTTER", que mostra as curvas de características das variáveis (média, variância, IDM, entropia e número de "pixels" escuros).
IMAGEM	contém as imagens das placas de madeira utilizadas no conjunto de teste.
PROGRAMAS	contém os códigos fontes dos "softwares" desenvolvidos: testat.c – programa para treinamento do classificador estatístico estat.c – programa do classificador estatístico knn.c - programa para treinamento e teste do classificador KNN trede.c - programa para treinamento da rede neural rede.c - programa da rede neural MLP para classificação tnefz.c - programa para treinamento da rede neural dos sistemas neuro-fuzzy1 e neuro-fuzzy2 nefz1.c – programa do classificador do sistema neuro-fuzzy1 nefz2.c – programa do classificador do sistema neuro-fuzzy2
TESE	contém cópia dos arquivos da tese.
VIDEO	filme PLACA.AVI mostrando o sistema em funcionamento.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABPM, 1990] Associação Brasileira de Produtores de Madeiras. - "Catálogo de Normas de Madeira Serrada de Pinus" - Caxias do Sul, R.S, pp. 9-34.
- [ASTM D-9, 1981] American Society for Testing and Materials - "Standart Definitions of Terms Relating to Wood" - Philadelphia, pp. 8-22.
- [Ballard & Brown, 1982] Ballard, Dana H. & Brown, Christopher M. - "Computer Vision" - Prentice-Hall INC, New Jersey, 1982.
- [Beale and Jackson, 1990] Beale, R. and Jackson, T. - "Neural Computing: An Introduction" - Adam Hilger, Bristol, 1990.
- [Bishop, 1995] Bishop, Cristopher M. - "Neural Networks for Pattern Recognition" - Clarendon Press-Oxford, 1995.
- [Brown & Harris, 1994] Brown, Martin and Harris, Chris - "Neurofuzzy Adaptative Modelling and Control" - Prentice Hall, New York, 1994.
- [Bustler et al, 1989] Bustler, D.A; Brunner, C.C. and Funck, J.W. - "A Dual-Threshold Image Sweep-and-Mark Algorithm for Defect Detection in Veneer"- Forest Products Journal, vol.39, n° 5, May, pp.25-28, 1989.
- [Chen et al, 1995] Chen, Y.Q.; Nixon, Mark S. and Thomas, David W. - "Statical Geometrical Features for Texture Classification" - Pattern Recognition, vol.28, n° 4, pp.537-552, 1995.
- [Connors et al, 1983] Connors, R.W.; McMillin, C.W.; Lin, K. and Vasques-Espinosa, R.E. - "Identifying and Location Surface Defects in Wood: Part of na Automated Lumber Processing System"- IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5, n° 6, November, pp. 573-583, 1983.
- [Duda el al, 1979] Duda, R.O.; Nitzan, D. and Barret, P. - "Use of Range and Reflectance Data to Find Planar Surface Regions" - IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-1, n° 3, July, pp. 259-271, 1979.

- [Duda & Hart, 1973] Duda, R.O and Hart, P.E.H. – “Pattern Classification and Scene Analysis” – John Wiley & Sons, New York, 1973.
- [Findly, 1975] Findlay, W.P.K. – “Timber: Properties and Uses”, William Clowes & Sons, London, 1975.
- [Fu, 1974] Fu, K.S. - “Syntactic Methods in Pattern Recognition” - Academic Press, New York, 1974.
- [Gallant, 1994] Gallant, Stephen I. - “Neural Network Learning and Expert Systems” - The MIT Press, Massachusetts, 1994.
- [Gonzalez & Wintz, 1987] Gonzalez, Rafael C. & Wintz, Paul - “Digital Image Processing” - Addison-Wesley Publishing Company, 2nd edition, 1987.
- [Han et al, 1994] Han, Joon H.; Kóczy, László T. and Poston, Timothy - “Fuzzy Hough Transform” - Pattern Recognition Letters, n^o 15, July, pp.649-658, 1994.
- [Haralick et al, 1973] Haralick, Robert M.; Shanmugam, K. and Dinstein, It'shak - “Textural Features for Image Classification” - IEEE Transactions on Systems, Man and Cybernetics, vol.SMC-3, n^o 6, November, pp.610-621, 1973.
- [Haralick, 1979] Haralick, Robert M. - “Statistical and Approaches to Texture”- Proceedings of the IEEE, vol.67, n^o 5, May, pp.786-804, 1979.
- [Hellmeister, 1972] Hellmeister, J.C. – “Sobre a Determinação das Características Físicas da Madeira” – Tese, São Carlos, Dezembro, 1972.
- [Hertz et al, 1993] Hertz, John; Krogh, Anders and Palmer, Richard G. - “Introduction to the Theory of Neural Computation” - Addison_Wesley Publishing Company, June, 1993.
- [Higgins & Goodman, 1994] Higgins, Charles M. and Goodman, Rodney M. - “Fuzzy Rule-Based Networks for Control” - IEEE Trans. On Fuzzy Systems, vol.2, n^o 1, February, pp. 82-88, 1994.
- [Hirota & Pedrycz, 1994] Hirota, Kaoru and Pedrycz, Witold - “OR/AND Neuron in Modelling Fuzzy Set Connectives” - IEEE Trans. On Fuzzy Systems, vol.2, n^o 2, May, pp. 151-161, 1994.
- [Ishibuchi et al, 1993] Ishibuchi, Hisao; Fujioka, Ryosuke and Tanaka, Hideo - “Neural Networks that Learn from Fuzzy IF-THEN Rules” - IEEE Trans. On Fuzzy Systems, vol.1, n^o 2, May, pp. 85-97, 1993.
- [Jang, 1993] Jang, Jyh-Shing Roger - “ANFIS: Adaptive-Network-Based-Fuzzy-Inference-System” - IEEE Trans. On Systems, Man and Cybernetics, vol.29, n^o 3, pp. 665-685, May, 1993.

- [Jang & Sun, 1995] Jang, Jyh-Shing Roger and Sun, Chuen-Tsai - "Neuro-Fuzzy Modelling and Control" - The Proceedings of the IEEE, vol. 83, n° 3, pp. 378-406, March, 1995.
- [Jang & Gulley, 1995] Jang, J.S.R. and Gulley, N. - "Fuzzy Logic Toolbox for Use With Matlab" - The Math Works Inc., 1995.
- [Klir & Yuan, 1995] Klir, George J. and Yuan, Bo - "Fuzzy Sets and Fuzzy Logic" - Theory and Applications, Prentice Hall PTR, New Jersey, 1995.
- [Knapp, 1996] Knapp, R.Benjamin - "Fuzzy Sets and Pattern Recognition" - <http://www-engr.sjsu.edu/~Knapp/HCIFUZZY/fuzzy.htm>.
- [Koga, 1988] Koga, M.E.T. - "Matérias-Primas Fibrosas" - Celulose e Papel-Tecnologia de fabricação de pasta celulósica, vol.1, cap.II, IPT, 1988.
- [Kwan & Cai, 1994] Kwan, Hon K. and Cai, Yaling - "A Fuzzy Neural Network and its Application to Pattern Recognition" - IEEE Trans. On Fuzzy Systems, vol.2, n° 3, August, pp. 185-193, 1994.
- [Koivo & Kim, 1989] Koivo, A J. and Kim, C.W. - "Automatic Classification of Surface Defects on Red Oak Boards" - Forest Products Journal, vol.39, n° 9, September, pp.20-30, 1989.
- [Kovács, 1997] Kovács, Z.L. - "O Cérebro e Sua Mente - Uma Introdução à Neurociência Computacional" - Edição Acadêmica, 1997.
- [Larousse Cultural, 1998] - "Grande Enciclopédia Larousse Cultural" - Nova Cultural, vol. 15, 1998.
- [Law, Itoh & Seki, 1996] Law, Todd; Itoh, Hidenori and Seki, Hirohisa - "Image Filtering, Edge Detection and Edge Tracing Using Fuzzy Reasoning" - IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 18, n° 5, May, pp. 481-491, 1996.
- [Lee, 1990a] Lee, Chuen Chien - "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I" - IEEE trans.on Systems, Man and Cybernetics, vol.20, n° 2, March/April, pp.404-418, 1990.
- [Lee, 1990b] Lee, Chuen Chien - "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II" - IEEE trans.on Systems, Man and Cybernetics, vol.20, n° 2, March/April, pp.419-435, 1990.
- [Lee, Kwak & Kwang, 1996] Lee, Keon-Myung; Kwak, Dong-Hoon and Kwang, Hyung Lee - "Fuzzy Inference Neural Network for Fuzzy Model Tuning" - IEEE Trans.on Systems, Man and Cybernetics-Part B: Cybernetics, vol.26, n° 4, August, pp.637-645, 1996.

[Lee & Wong, 1996] Lee, C.K. and Wong, S.P. - "A Mathematical Morphological Approach for Segmenting Heavily Noise-Corrupted Images"- Pattern Recognition, vol.29, n° 8, pp. 1347-1358, 1996.

[Lin & Cunningham III, 1995] Lin, Yinghua and Cunningham III, George A. - "A New Approach to Fuzzy-Neural System Modeling" - IEEE Trans. On Fuzzy Systems, vol.3, n° 2, May, pp. 190-197, 1995.

[Lin & Lu, 1995] Lin, Chin-Teng and Lu, Ya-Ching - "A Neural Fuzzy Systems with Linguistic Teaching Signals" - IEEE Trans. On Fuzzy Systems, vol.3, n° 2, May, pp. 169-189, 1995.

[Loesch & Sari, 1996] Loesch, C. and Sari, S.T. - "Redes Neurais Artificiais - Fundamentos e Modelos" - Editora da FURB, 1996.

[Masters, 1993] Masters, Timothy - "Practical Neural Network Recipes in C++" Academic Press, New York, 1993.

[Masters, 1994] Masters, Timothy - "Signal and Image Processing with Neural Networks a C++ Sourcebook" - John Wiley & Sons, Inc., New York, 1994.

[Ojala et al, 1996] Ojala, Timo; Pietikäinen, Matti and Harwood, David - "A Comparative Study of Texture Measures with Classification Based on Feature Distributions" - Patter Recognition, vol.29, n° 1, pp.51-59, 1996.

[Pal & Mitra, 1992] Pal, Sankar K. and Mitra, Sushmita - "Multilayer Perceptron, Fuzzy Sets and Classification" - IEEE Trans. On Neural Networks, vol.3, n° 5, September, pp.683-697, 1992.

[Parker, 1994] Parker, J.R. - "Practical Computer Vision Using C" - John Wiley & Sons, Inc., New York, 1994.

[Patterson, 1996] Patterson, D.W. - "Artificial Neural Networks - Theory and Applications", Prentice Hall, Singapore, 1996.

[Pedrycz, 1990] Pedrycz, W. - "Fuzzy Sets in Pattern Recognition: Methodology and Methods" - Pattern Recognition, vol.23, n° 1/2, pp. 121-146, 1990.

[Pichler et al, 1996] Pichler, Olaf; Teuner, Andreas and Hosticka, Bedrich J. - "A Comparison of Texture Feature Extraction using Adaptive Gabor Filtering, Pyramidal and Tree Structured Wavelet transforms" - Patter Recognition, vol.29, n° 5, pp. 733-742, 1996.

[Pratt, 1991] Pratt, William K. - "Digital Image Processing" - A Wiley-Interscience Publication, 1991.

[Prechelt, 1994] Prechelt, Lutz - "Proben 1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules", Technical Report 21/94, Universität

Karlsruhe , anonymous <ftp://pub/papers/techreports/1994/1994-21.ps.Z> em <ftp.ira.uka.de>

[Rauber, 1997] Rauber, T. W. – “Pattern Recognition” – XVII Congress of The Brazilian Computer Science Society, Brasília, DF. 1997.
<http://www.inf.ufes.br/~thomas>

[Ray & Ghoshal, 1997] Ray, Kumar S. and Ghoshal, Jayati - “Neuro Fuzzy Approach to Pattern Recognition” - Neural Networks, vol.10, n° 1, pp.161-182, 1997.

[Rodrigues, 1998] Rodrigues, E.L.L. – “Metodologia de Inspeção Visual Utilizando Limiar (“Threshold”) Entrópico com Aplicação na Classificação de Placas de Madeira” – tese de Doutorado defendida no Instituto de Física de São Carlos, Maio, 1998.

[Schalkoff, 1989] Schalkoff, Robert J. - “Digital Image Processing and Computer Vision” - John Wiley & Sons, INC - New York, 1989.

[Simpson, 1992] Simpson, Patrick K. - “Fuzzy Min-Max Neural Networks-Part 1. Classification” - IEEE Trans. On Neural Networks, vol.3, n° 5, September, pp. 776-786, 1992.

[Sobey & Semple, 1989] Sobey, P.J. and Semple, E.C. – “Detection and Sizing Visual Features in Wood Using Tonal Measures and a Classification Algorithm” – Pattern Recognition, vol.22, n° 4, pp. 367-380, 1989.

[Steele et al, 1991] Steele, P.H.; Neal, S.C.; McDonald, K.A and Cramer,S.M. – “The Slope-of-Grain Indicator for Defect Detection in Unplaned Hardwood Lumber” – vol.41, n° 1, January, pp.15-20, 1991.

[Sugeno & Takagi, 1985] Sugeno, Tomohiro and Takagi, Michio - “Fuzzy Identification of Systems and Its Applications to Modeling and Control” - IEEE Trans. On Systems, Man and Cybernetics, vol.SMC-15, n° 1, January/February, pp.116-132, 1985.

[Suassuna, 1997] Suassuna, J. – “A Cultura do Pinus: uma Perspectiva e uma Preocupação” – Brasil Florestal, n° 29, ano VIII, Janeiro/Março, 1977 ou <http://www.fundaj.gov.br/docs/tropico/desat/pinus.html>

[Szymani & McDonald, 1981] Szymani, R. and McDonald, K.A. – “Defect Detection in Lumber: State of Art”- Forest Products Journal, vol.31, n° 11, November, pp.34-44, 1981.

[Tafner et al, 1996] Tafner, M.A.A; Xerz, M. and Rodrigues Filho, I.W. – “Redes Neurais Artificiais – Introdução e Princípios de Neurocomputação”, ed. EKO e FURB, Blumenau, 1996.

[Tamura et al, 1978] Tamura, Hideyuki; Mori, Shunji and Yamawaki, Takashi - "Textural Features Corresponding to Visual Perception" - IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-8, n° 6, June, pp.460-473, 1978.

[Unser, 1986] Unser, M. - "Sum and Difference Histograms for Texture Classification" - IEEE Transaction on Pattern Analysis and Machine Intelligence"- vol. PAMI-8, n° 1, January, pp.118-125, 1986.

[Wang & Mendel, 1992] Wang and Mendel, J.M. - "Generating Fuzzy Rules by Learning from Examples" - IEEE Trans. On Systems, Man and Cybernetics, vol.22, n° 6, pp. 1414-1427, 1992.

[Zadeh, 1965] Zadeh, L.A. - "Fuzzy Sets" - Information and Control, vol.8, 1965, pp.338-353.

[Zhu et al, 1996] Zhu, D; Conners, R.W.; Schmoldt, D.L. and Araman, P.A - "A Prototype Vision System for Analyzing CT Imagery of Hardwood Logs" - IEEE transactions on Systems, Man and Cybernetics - Part B: Cybernetics, vol.26, n° 4, August, pp.521-532, 1996.