

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA

MESTRADO ACADÊMICO

DISSERTAÇÃO DE MESTRADO

**Classificação e Recuperação de Imagens
por Cor
Utilizando Técnicas de
Inteligência Artificial**

por

Túlio Cléber Bender

Dissertação submetida à avaliação, como requisito parcial
para a obtenção do grau de
Mestre em Computação Aplicada

Professor Dr. Fernando Santos Osório
Orientador

São Leopoldo, junho de 2003.




UNIVERSIDADE DO VALE DO RIO DOS SINOS – UNISINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

Aluno: Túlio Cléber Bender
Título da Dissertação: Classificação e Recuperação de Imagens por Cor utilizando Técnicas de Inteligência Artificial

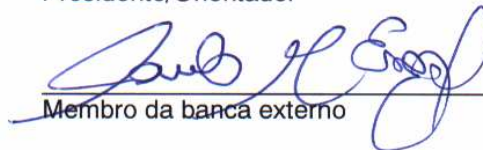
Banca:

Dr. Fernando Santos Osório



Presidente/Orientador

Dr. Paulo Martins Engel



Membro da banca externo

Dr. Adelmo Luis Cechin



Membro da banca interno

A banca examinadora da Dissertação, sob registro de Ata nº 14/2003 - PIPCA, em cumprimento ao Regimento do Programa Interdisciplinar de Pós-graduação em Computação Aplicada, julga esta Dissertação aprovada para o processo de obtenção de título de Mestre a Túlio Cléber Bender.

São Leopoldo, 24 de julho de 2003.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Bender, Túlio Cléber

Classificação e Recuperação de Imagens por Cor Utilizando Técnicas de Inteligência Artificial. – São Leopoldo: PIPCA da UNISINOS, 2003.

p.: 125 il.

Dissertação (mestrado) - Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar em Computação Aplicada, São Leopoldo, BR – RS, 2003. Orientador: Osório, Fernando Santos.

1. Classificação e Recuperação de Imagens. 2. Redes Neurais Artificiais. 3. Cor. I. Osório, Fernando Santos. II. Título.

AGRADECIMENTOS

Agradeço em especial à minha família por todo o incentivo e pelas oportunidades de crescimento que recebi ao longo de minha vida.

Aos meus colegas de trabalho no Laboratório de Automação Industrial, Prof. Giancarlo M. Pereira, Prof. Felipe C. Schneider, Carlos A. Mendel, Carlos H. Bauer e Farlei J. Heinen meu obrigado pela oportunidade de convívio e troca de conhecimento ao longo de quase uma década.

Agradeço ao Prof. Marcelo G. Perin pela oportunidade para participar como bolsista de iniciação científica em seu projeto de pesquisa no início de minha vida acadêmica.

Ao meu orientador durante o mestrado, Prof. Fernando S. Osório, meu muito obrigado por sua orientação, várias horas de conversa, correções e exigência durante a realização dos créditos e da preparação da dissertação.

Por fim, gostaria de expressar minha gratidão para com a Universidade do Vale do Rio dos Sinos / Laboratório de Automação Industrial do Centro de Ciências Exatas e Tecnológicas pelo aporte financeiro na forma de bolsa de estudos que me permitiu cursar este mestrado.

SUMÁRIO

AGRADECIMENTOS	3
SUMÁRIO	4
LISTA DE FIGURAS	6
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS	10
RESUMO	12
ABSTRACT	13
1 INTRODUÇÃO	14
1.1 Definições de Termos Utilizados	15
1.2 Organização do Trabalho	16
1.3 OBJETIVOS	17
1.4 Objetivos Específicos	17
2 SISTEMAS DE INDEXAÇÃO E RECUPERAÇÃO DE IMAGENS	19
2.1 Estrutura Genérica de Sistemas CBIR	20
2.1.1 Bases de Dados de Imagens Homogêneas	21
2.1.2 Bases de Dados de Imagens Heterogêneas	21
2.2 Extração de Características	22
2.2.1 Utilizando Cor	22
2.2.1.1 Fundamentos da cor.....	22
2.2.1.2 Sistemas de cor	24
2.2.1.3 Redução de Cores.....	27
2.2.2 Utilizando Forma	28
2.2.3 Utilizando Textura	28
2.3 Comparação de Características	29
2.3.1 Utilizando Cor	29
2.3.2 Utilizando Textura	30
2.3.3 Medidas de Similaridade	31
2.4 Sistemas Atuais para Classificação e Recuperação de Imagens	33
2.4.1 IKONA (Surfimage).....	33
2.4.2 Photobook	34
2.4.3 QBIC.....	36
2.5 Resumo do Capítulo	36

3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL	38
3.1 Redes Neurais Artificiais	39
3.1.1 Modelo Computacional de uma Rede Neural Artificial	40
3.1.2 RNA do Tipo <i>Multilayer Perceptron</i>	43
3.1.3 RNA do Tipo MLP com Algoritmo de Aprendizagem <i>Cascade-Correlation</i>	45
3.1.4 Utilização de RNAs na Classificação de Imagens	46
3.2 Clusterização	48
3.2.1 Clusterização Utilizando o Algoritmo k-Means	50
3.3 Resumo do Capítulo.....	51
4 SUPERFÍCIES DE SEPARAÇÃO EM RNAS MLP	53
4.1 Compreensão Geométrica das Superfícies de Separação	53
4.1.1 Superfícies Abertas de Separação (SAS)	54
4.1.2 Superfícies Fechadas de Separação (SFS)	62
4.2 Abordagem para Fechamento de Superfícies de Separação	69
4.3 Outras Considerações Sobre o Trabalho de Gori e Scarselli.....	70
4.4 Resumo do Capítulo.....	73
5 CLASSIFICAÇÃO E RECUPERAÇÃO DE IMAGENS UTILIZANDO RNAS MLP	
 CASCADE-CORRELATION.....	75
5.1 Metodologia	77
5.2 Experimentos.....	80
5.2.1 Seleção do Método de Extração de Características (BDI B0)	81
5.2.1.1 Sistema de cor RGB, 5 cores, método T1	81
5.2.1.2 Sistema de cor RGB, 5 clusters, método T2	83
5.2.1.3 Sistema de cor RGB, 5 cores, método T3	86
5.2.2 Seleção do Sistema de Cor (BDI B0, HSV, T3, 5 cores)	92
5.2.3 Análise dos Resultados	94
5.2.4 Desempenho em Base de <i>Benchmark</i> (BDI B1)	95
5.2.4.1 Sistema de cor RGB, 5 cores, método T3	96
5.2.5 Desempenho com Imagens Não Controladas (BDI B2)	99
5.2.5.1 Sistema de cor RGB, 5 cores, método T3	99
5.3 Resumo do Capítulo.....	101
6 CONCLUSÃO	103
6.1 Perspectivas	104
ANEXO A	105
BIBLIOGRAFIA	122

LISTA DE FIGURAS

Figura 1.1 - Exemplo de duas imagens do tipo <i>imagem-exemplo</i> .	15
Figura 1.2 - Exemplo de <i>imagens-recuperadas</i> .	16
Figura 1.3 - Exemplo de <i>imagens-de-treinamento</i> composto de regiões de latas de Coca-Cola (primeira linha) e <i>imagens-de-fundo</i> diversas (segunda linha).	16
Figura 1.4 - Exemplo de generalização.	17
Figura 2.1 - Exemplo de pesquisa utilizando MPEG-7. Adaptação [DAY 01].	20
Figura 2.2 - Esquema típico de um sistema atual CBIR [ANT 02].	21
Figura 2.3 - Gradiente de cores com alteração na luminância da imagem.	23
Figura 2.4 - Exemplo de diferentes tonalidades (crominância) para a cor vermelha.	23
Figura 2.5 - Cubo RGB.	25
Figura 2.6 - Hexacone HSV.	26
Figura 2.7 - Exemplo de pesquisa no sistema IKONA. Escolha de uma imagem para ser usado como exemplo (segunda linha, terceira coluna).	35
Figura 2.8 - Exemplo do resultado de uma busca por <i>imagem-exemplo</i> no IKONA.	35
Figura 3.1 - Modelo não-linear de um neurônio [HAY 01].	41
Figura 3.2 - Função de ativação sigmóide (Eq. (4.4)). Temperatura da sigmóide variando em 0; 0,2; 0,5 e 1.	43
Figura 3.3 - Três arquiteturas de RNA mais comuns: redes alimentadas adiante com camada única; redes alimentadas diretamente com múltiplas camadas e redes recorrentes.	44
Figura 3.4 - Exemplo de duas possibilidades de clusterização para o mesmo conjunto de exemplos segundo o critério de: finalidade (A) e tipo de mídia (B).	48
Figura 3.5 - Exemplos de classificação de tipo de clusterização quanto ao grau de pertinência de seus elemento ao cluster. Em A temos <i>cluster</i> do tipo <i>hard</i> e em B definidos por conjuntos de lógica nebulosa.	50
Figura 4.1 - Superfície de separação para aceitar / rejeitar padrões (círculos representam os padrões que devem ser aceitos e quadrados representam os padrões a serem rejeitados).	53
Figura 4.2 - Plano de separação determinado pelos pesos de um neurônio e a influência do <i>Bias</i> .	55
Figura 4.3 - Planos de separação de Classe e Não Classe.	56
Figura 4.4 - Disposição dos quadrantes adotada nos exemplos deste capítulo, origem x,y no centro.	56
Figura 4.5 - Caso 1 SAS, 6 exemplos <i>Nuvem0</i> e 16 exemplos <i>Nuvem1</i> .	58
Figura 4.6 - Caso 1 SAS, 6 exemplos <i>Nuvem0</i> e 16 exemplos <i>Nuvem1</i> e com visualização de padrões classificados como <i>Nuvem1</i> .	59
Figura 4.7 - Caso 2 SAS, 7 exemplos de <i>Nuvem0</i> e 11 exemplos de <i>Nuvem1</i> .	60
Figura 4.8 - Caso 2 SAS, 7 exemplos de <i>Nuvem0</i> e 11 exemplos de <i>Nuvem1</i> e critério de limiar de 0,2.	61
Figura 4.9 - Caso 2 SAS, 7 exemplos de <i>Nuvem0</i> , 11 exemplos de <i>Nuvem1</i> e critério de limiar de 0,8.	62
Figura 4.10 - Caso 1 SFS, 20 exemplos da classe <i>Nuvem0</i> e 16 exemplos da classe <i>Nuvem1</i> .	64
Figura 4.11 - Caso 1 SFS, 20 exemplos da classe <i>Nuvem0</i> e 16 exemplos da classe <i>Nuvem1</i> e limiar de 0,5.	65
Figura 4.12 - Caso 2 SFS, 31 exemplos da classe <i>Nuvem0</i> e 11 exemplos da classe <i>Nuvem1</i> .	66
Figura 4.13 - Caso 2 SFS, 31 exemplos da classe <i>Nuvem0</i> , 11 exemplos da classe <i>Nuvem1</i> e limiar de 0,2.	67
Figura 4.14 - Caso 2 SFS, 31 exemplos da classe <i>Nuvem0</i> , 11 exemplos da classe <i>Nuvem1</i> e limiar de 0,8.	68
Figura 4.15 - Exemplo de conjunto de Pareto para o problema do valor de limiar.	69
Figura 4.16 - Exemplo de aperfeiçoamento futuro para abordagem NNC.	70
Figura 4.17 - SFS no cubo RGB determinada por um hiperplano e pelos limites dos atributos.	72
Figura 4.18 - Exemplo de situações em que uma RNA RBF não classifica adequadamente.	73

Figura 4.19 – Exemplo de duas possibilidades de classificação em uma RNA RBF que dependem do algoritmo de aprendizagem. Em A temos maior ocorrência de FP e em B maior ocorrência de FN.	73
Figura 5.1 – Interface principal do IMatch.....	76
Figura 5.2 – Exemplo da preparação das imagens utilizadas no treinamento da RNA.	78
Figura 5.3 – Exemplo de contagem múltipla para regiões de C e FP e FN devido à sobreposição da janela de procura. Tem-se a contagem total: 05 Classe (C), 06 falso-positivo (FP) e 01 falso-negativos (FN).	79
Figura 5.4 – Gráfico comparativo da taxa de generalização da RNA x Abordagem para RGB e HSV, método T3.	94
Figura 5.5 - Gráfico comparativo das médias para FP, FN, C x abordagem para HSV e RGB, método T3...	95

FIGURAS EM CORES, ANEXO A

Fig. Anexo A1 - Nuvem de pontos em VRML, RGB, do padrão <i>Classe</i> (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.	105
Fig. Anexo A2 - Nuvem de pontos em VRML, RGB, do padrão <i>Classe</i> (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.	106
Fig. Anexo A3 - Nuvem de pontos em VRML, RGB, do padrão <i>Não-Classe</i> (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.	107
Fig. Anexo A4 - Nuvem de pontos em VRML, RGB, do padrão <i>Não-Classe</i> (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.	108
Fig. Anexo A5 – Padrões de aprendizagem da BDI B0 do tipo <i>Classe</i>	109
Fig. Anexo A6 - Padrões de aprendizagem da BDI B0 do tipo <i>Não-Classe</i>	109
Fig. Anexo A7 - Padrões de teste da BDI B0 do tipo <i>Classe</i>	110
Fig. Anexo A8 - Padrões de teste da BDI B0 do tipo <i>Não-Classe</i>	110
Fig. Anexo A9 – Imagens de teste para BDI B0. Todas possuem o padrão a ser procurado.	111
Fig. Anexo A10 – Resultado do reconhecimento executado na imagem B0_CP_09.PNG com treino inicial da RNA. BDI B0, 5 cores, T3.	112
Fig. Anexo A11 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com duas iterações de RIFP. BDI B0, 5 cores, T3.	112
Fig. Anexo A12 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com duas iterações de NNC e aprendizado incremental. BDI B0, 5 cores, T3.	113
Fig. Anexo A13 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com uma iteração de NNC e sem utilizar aprendizado incremental. BDI B0, 5 cores, T3.	113
Fig. Anexo A14 – Objetos que compõem a base COIL-100.	114
Fig. Anexo A15 – Algumas regiões do <i>Objeto 25</i> da base COIL-100 utilizado como padrão de treinamento para a RNA.	114
Fig. Anexo A16 – Exemplos dos objetos que foram utilizados na base de pré-validação para BDI B1.	115
Fig. Anexo A17 - Regiões de latas de Coca-Cola utilizadas para compor os exemplos de aprendizado de <i>Classe</i> para BDI B2.	115
Fig. Anexo A18 – Regiões de <i>imagem-de-fundo</i> utilizadas para compor os exemplos de aprendizado de <i>Não-Classe</i> para BDI B2.	116
Fig. Anexo A19 - Regiões de latas de Coca-Cola utilizadas para compor os exemplos de teste de <i>Classe</i> para BDI B2.	116
Fig. Anexo A20 - Regiões de <i>imagem-de-fundo</i> utilizadas para compor os exemplos de teste de <i>Não-Classe</i> para BDI B2.	117
Fig. Anexo A21 – Imagens utilizadas para teste de recuperação BDI B2.	118
Fig. Anexo A22 – Resultado da classificação para BDI B2 com treino inicial da RNA.	119
Fig. Anexo A23 - Resultado da classificação para BDI B2 após segunda iteração de RIFP.	120

Fig. Anexo A24 - Resultado da classificação para BDI B2 após segunda iteração de NNC..... 121

LISTA DE TABELAS

Tabela 4.1 – Tabela-resumo de decisão do rótulo com a condição de saída vs. limiar.....	55
Tabela 5.1 – Resumo dos experimentos a serem apresentados.....	80
Tabela 5.2 – Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T1 (Histograma de Cores).....	82
Tabela 5.3 - Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T1.....	82
Tabela 5.4 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T1.....	82
Tabela 5.5 - Totalizações de regiões FP, FN e C para treino inicial da RNA. - BDI B0, RGB 5 cores, T1.....	83
Tabela 5.6 - Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T2 (clusterização k-Means).....	84
Tabela 5.7 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 clusters, T2... ..	85
Tabela 5.8 – Melhor treinamento da RNA para BDI B0, RGB 5 clusters, T2.....	85
Tabela 5.9 - Totalizações de regiões FP, FN e C para treino inicial da RNA, BDI B0, RGB 5 clusters, T2....	85
Tabela 5.10 - Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T2 (clusterização k-Means).....	87
Tabela 5.11 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T3.....	87
Tabela 5.12 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T3.....	87
Tabela 5.13 – Totalizações de regiões de FP, FN e C para treino inicial da RNA, BDI B0, 5 cores, T3.....	87
Tabela 5.14 – Resumo das médias e desvios padrões do treino inicial das RNAs para os métodos T1, T2 e T3.....	88
Tabela 5.15 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T3.....	88
Tabela 5.16 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T3.....	89
Tabela 5.17 – Totalizações de regiões FP, FN e C para treino inicial da RNA, RIFP e NNC após duas iterações. - BDI B0, RGB 5 cores, T3.....	89
Tabela 5.18 - Médias e desvios padrões do treinamento de vinte RNAs para BDI B0, RGB 5 cores, NNC, T3, sem aprendizado incremental.....	90
Tabela 5.19 - Melhor treinamento da RNA para BDI B0, RGB 5 cores, NNC, T3, sem aprendizado incremental.....	90
Tabela 5.20 - Totalizações de regiões FP, FN e C para treino inicial da RNA, NNC após uma iteração. - BDI B0, RGB 5 cores, T3, sem aprendizado incremental.....	90
Tabela 5.21 – Resumo das médias e desvios padrões do melhores resultados obtidos para o método T3 utilizando as abordagens RIFP e NNC para BDI B0, 5 cores.....	91
Tabela 5.22 - Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, HSV 5 cores, T3.....	92
Tabela 5.23 – Melhor treinamento da RNA para BDI B0, HSV 5 cores, T3.....	92
Tabela 5.24 - Totalizações de regiões FP, FN e C para treino inicial da RNA, RIFP e NNC após duas iterações, - BDI B0, HSV 5 cores, T3.....	93
Tabela 5.25 - Médias e desvios padrões do treinamento de 25 RNAs para BDI B1, RGB 5 cores, T3.....	96
Tabela 5.26 - Melhor treinamento da RNA para base de pré-validação BDI B1, RGB 5 cores, T3.....	96
Tabela 5.27 – Total de imagens recuperadas para base de pré-validação BDI B1, RGB 5 cores, RIFP, T3.....	97
Tabela 5.28 - Total de imagens recuperadas para base de pré-validação BDI B1, RGB 5 cores, NNC, T3.....	97
Tabela 5.29 - Total de imagens recuperadas para BDI B1, RGB 5 cores, RIFP, T3.....	98
Tabela 5.30 - Total de imagens recuperadas para BDI B1, RGB 5 cores, NNC, T3.....	98
Tabela 5.31 - Resultados obtidos com recuperação por histograma na base COIL-100 [MOS 01].....	98
Tabela 5.32 - Médias e desvios padrões do treinamento de 25 RNAs para BDI B2, RGB 5 cores, T3.....	99
Tabela 5.33 - Melhor treinamento da RNA para base BDI B2, RGB 5 cores, T3.....	99
Tabela 5.34 - Total de imagens recuperadas para BDI B2, RGB 5 cores, RIFP, T3.....	100
Tabela 5.35 - Total de imagens recuperadas para BDI B2, RGB 5 cores, NNC, T3.....	100

LISTA DE ABREVIATURAS

BDI – Base de Dados de Imagens
bpp - Bits Per Pixel
C - Classe
CBIR - Content Based Image and Video Retrieval
CCV - Color Coherence Vector
COIL – Columbia Object Image Library
DBMS - Data Base Management System
DDL - Description Definition Language
DLL – Dynamic Link Library
FN – Falso-negativo
FP – Falso-positivo
HC - Histograma de Cores
HSV - Hue Saturation Value
IA – Inteligência Artificial
IMatch – Image Matching
ML – Machine Learning
MLP – Multilayer Perceptron
MPEG – Moving Picture Expert Group
NC – Não-Classe
NNC – Nuvem de Não-Classe
PCA – Principal Component Analysis
PNG – Portable Network Graphics
QBE - Query by Example
QMF - Quadrature Mirror Filter
QPE - Query by Pictorial Example
RBF – Radial Basis Function
RNA - Redes Neurais Artificiais
RGB - Red Green Blue
RIFP – Regiões de Imagens Falso-Positivo
SAD – Sum of Absolute Diference
SAS – Superfície Aberta de Separação
SFS – Superfície Fechada de Separação
SOM – Self Organizing Maps
SSD – Sum of Square Diference

VRML – Virtual Reality Modeling Language

RESUMO

A recuperação e classificação de imagens é um tema bastante pesquisado atualmente. Além dos desafios encontrados no campo teórico e prático para permitir que máquinas possuam a capacidade de visão, sua pesquisa resulta em várias aplicações práticas para o dia-a-dia. A visão computacional, grande área na qual está inserida a recuperação e classificação de imagens, possui aplicações práticas dentre as quais podemos citar softwares capazes de recuperar imagens em bases de dados de imagens, reconhecimento de pessoas por características de biometria (impressões digitais, reconhecimento por íris ou face), localização e quantificação de logomarcas na mídia, localização de objetos numa cena e mecanismos de visão para a robótica.

A pesquisa desenvolvida nesta dissertação foca-se em obter uma generalização através do aprendizado das características de uma coleção de imagens pertencentes a uma mesma classe as quais servirão como exemplo de aprendizagem, com isto obtendo um modelo que identifique esta classe. Para tanto, foi feita uma revisão do estado-da-arte das técnicas utilizadas em sistemas para recuperação e classificação de imagens e Aprendizado de Máquina.

Outro importante tópico abordado neste trabalho são as superfícies de separação que uma Rede Neural Artificial determina durante o aprendizado e seu impacto na tarefa de recuperação e classificação de imagens. É estudada a compreensão geométrica destas superfícies de separação em um espaço multidimensional de padrões de entrada. Como resultado deste estudo são propostas duas abordagens para determinar superfícies fechadas de separação no espaço multidimensional de padrões de entrada.

Neste trabalho foi desenvolvido o software IMatch para recuperação e classificação de imagens baseado em Aprendizado de Máquina utilizando como principal característica discriminante à cor. Para isto se utilizou Redes Neurais Artificiais do tipo *Multilayer Perceptron* com algoritmo de aprendizagem do tipo *Cascade Correlation*. Ao contrário dos métodos tradicionalmente empregados para a recuperação de imagens baseados em técnicas complexas de pré-processamento de imagens e algoritmos de *matching* o IMatch utiliza aprendizado supervisionado para aprender o “modelo” do padrão que se pretende reconhecer a partir de uma coleção de exemplos do padrão.

Através do IMatch foi possível demonstrar a validade, vantagens e pontos a serem aprofundados, referentes ao uso prático das abordagens propostas nesta dissertação. A validação foi realizada utilizando-se diferentes bases de dados de imagens, que permitiram caracterizar o comportamento do sistema em casos artificiais, casos reais controlados e situações reais sem intervenção em relação à aquisição dos dados.

Palavras-chave: redes neurais artificiais, superfícies de separação fechadas, recuperação de imagens, classificação de imagens, cor.

ABSTRACT

Image retrieval and classification are today the subject of extensive research. This topic poses both theoretical and practical challenges as researchers attempt to give machines such as computers and robots the ability to “see”. Image retrieval and classification are part of a wider field known as Computer Vision, which encompasses several practical applications such as image retrieval from databases storing only raw images, biometric recognition (from images of fingerprints, face or iris), retrieval of visual trademarks and logos from advertisements, location of objects in a scene and vision techniques for applications in robotics.

The research developed in this work is focused on obtaining a generalization of characteristics extracted from a collection of images belonging to a single class using supervised learning techniques. The result is a model that “identifies” a given class of images. To achieve this, a review of the state-of-the-art in content-based image retrieval systems and Machine Learning techniques was needed.

There is also a discussion on the separation surfaces that an artificial neural network draws during the training stage and how they affect retrieval performance. A thorough study on the geometric understanding of these separation surfaces in the multidimensional input space is made and two iterative techniques to help achieve closed separation surfaces are proposed.

This work resulted in the development of a software package (IMatch) that performs image retrieval and classification using Machine Learning techniques based on color. A choice was made for Artificial Neural Networks using a Multilayer Perceptron architecture with the Cascade Correlation learning algorithm to accomplish the task. Unlike traditional methods, which rely on complex techniques of image pre-processing and matching algorithms, IMatch takes advantage of supervised learning to “learn” the model of a collection of patterns representing the pattern which should be retrieved from a collection of images.

The use of IMatch demonstrated the validity, the advantages and the issues that require further study of the approaches proposed in this research. Three image databases were used to validate the two approaches proposed to achieve closed separation surfaces, which made it possible to characterize the behavior of the system using artificial pictures, pictures taken under controlled conditions and pictures taken without any kind of conditioning.

Keywords: artificial neural networks, closed separation surfaces, image retrieval, image classification, color.

1 INTRODUÇÃO

Atualmente com o aumento considerável de bases de dados de imagens e vídeos devido às inovações tecnológicas tornou-se necessário manter algum tipo de mecanismo de indexação para posterior recuperação de dados. Executar esta indexação manualmente é uma tarefa sensível a interpretações subjetivas, é laboriosa e sujeita a erros [ANT 02, SEB 01, BRU 00, RUI 97, SMI 96]. Métodos automáticos de indexação e recuperação baseados em algum tipo de característica como cor, forma, textura ou uma combinação destas, são interessantes neste contexto porque reduzem a intervenção humana. A idéia de recuperação de imagens por exemplos é uma extensão da idéia de recuperação de páginas web e documentos que contenham no texto as palavras-chave especificadas na pesquisa (por exemplo, Google, Altavista e outros). No caso de imagens, busca-se recuperar em uma base de dados de imagens aquelas que contenham o elemento gráfico procurado como, por exemplo, uma determinada combinação de cores. Sistemas que adotam estes métodos são denominados como: “Content Based Image (and Video) Retrieval” (CBIR), “Query By Example” (QBE) ou “Query by Pictorial Example” (QPE) [ANT 02] [RUI 97].

As técnicas de recuperação de imagens vem sendo desenvolvidas desde a década de 1970 principalmente pelas áreas de gerenciamento de banco de dados (métodos baseados em indexação textual) e visão computacional [RUI 97]. Nos métodos baseados em indexação textual existe um DBMS que gerencia a informação semântica anexada a imagem. Esta indexação usualmente é feita manualmente, imagem a imagem, e é sensível a interpretação da pessoa que a está executando. Já nos CBIR procura-se extrair automaticamente das imagens características que as identifiquem. A partir das características da imagem de exemplo procura-se por características similares nas imagens disponíveis na base de pesquisa. As melhores similaridades são apresentadas como resultado da pesquisa.

Os métodos de recuperação e classificação de imagens atualmente utilizados são baseados em técnicas cada vez mais complexas de pré-processamento das imagens e algoritmos de *matching*. As pesquisas desenvolvidas utilizando técnicas de aprendizado de máquina para classificação de imagens são de domínio específico tais como reconhecimento de impressões digitais ou faces de pessoas. Neste contexto é interessante verificar qual seria o desempenho de um método baseado em aprendizado de máquina utilizando técnicas simples de pré-processamento para a tarefa de recuperação e classificação de imagens.

Optou-se por se utilizar cor e frequência de ocorrência da cor porque são características facilmente percebidas numa cena, são de fácil compreensão por pessoas e em muitas situações permitem uma caracterização adequada do objeto procurado. A escolha por técnicas de *Machine Learning* (ML) da IA com aprendizado supervisionado foi feita porque estas permitem que um conhecimento seja adquirido através de exemplos e que este conhecimento seja “refinado” pela experiência (novos exemplos).

Para simular a aplicação destas técnicas, desenvolveu-se o programa *Image Matching* (IMatch) que implementa a classificação e recuperação de imagens bitmap utilizando a com-

posição de cores (nos sistemas de cores RGB e HSV) e frequência de ocorrência das cores (histograma de cores) para representar as características extraídas da imagem. E o emprego de Redes Neurais Artificiais (RNA) do tipo *Multilayer Perceptron* (MLP) com algoritmo de aprendizado construtivo e incremental *Cascade-Correlation* [FAH 90] para executar a classificação das imagens.

1.1 Definições de Termos Utilizados

A nomenclatura utilizada na área de classificação e recuperação de imagens ainda não está totalmente padronizada. Diversos autores utilizam-se de termos diferentes para se referirem a um mesmo significado. Por isso iremos a seguir definir os termos que utilizaremos neste trabalho descrevendo o significado atribuído a estes.

Definimos *imagem-exemplo* como sendo um exemplo que é fornecido ao sistema e que contém o elemento gráfico procurado durante uma recuperação. Pode ser uma imagem completa, região de uma imagem contendo um objeto representado por cores, formas, texturas ou uma combinação destas. Na Figura 1.1 tem-se dois exemplos de imagens deste tipo.

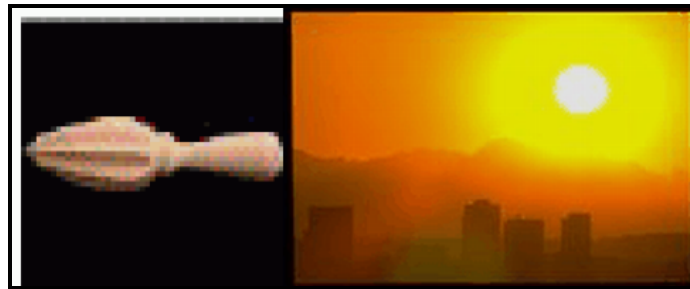


Figura 1.1 - Exemplo de duas imagens do tipo *imagem-exemplo*.

Imagens-de-busca é um conjunto de imagens disponíveis na base de dados de imagens no qual o CBIR irá fazer a busca.

A Figura 1.2 apresenta um exemplo de *imagens-recuperadas* que vem a ser imagens que possuem um determinado grau de semelhança com a *imagem-exemplo* e são apresentadas ao usuário do CBIR como possíveis resultados da pesquisa.

As *características* (ou *vetores de*) extraídas de uma imagem são atributos derivados da cor, textura e ou forma e que denotam alguma propriedade da imagem que pode ser utilizada para comparação entre imagens.

A *comparação de características* é uma função que retorna o grau de diferença entre as características extraídas da *imagem-exemplo* e das *imagens-de-busca* (por exemplo, a distância euclidiana entre dois histogramas de cor).

Definimos *imagem-de-treinamento* (Figura 1.3) como sendo as imagens das quais serão extraídas as características que irão compor os vetores que serão fornecidos à RNA durante o treinamento da rede.

As *imagens-de-fundo* são regiões de imagens que não são o alvo da pesquisa (para um exemplo, ver Figura 1.3, segunda linha), também podemos denominar as *imagens-de-fundo* como *contra-exemplos*.



Figura 1.2 - Exemplo de *imagens-recuperadas*¹.



Figura 1.3 - Exemplo de *imagens-de-treinamento* composto de regiões de latas de Coca-Cola (primeira linha) e *imagens-de-fundo* diversas (segunda linha).

1.2 Organização do Trabalho

Este trabalho se consiste de seis capítulos. No **Capítulo 1** fazemos uma introdução ao assunto e apresentamos os objetivos e a metodologia empregada. No **Capítulo 2** apresentamos os conceitos envolvidos na tarefa de classificação e recuperação de imagens e apresentamos uma breve descrição de alguns sistemas de classificação e recuperação de imagens existentes. Uma in-

¹ Exemplo obtido a partir de uma pesquisa no CBIR *PicToSeek* utilizando como *imagem-exemplo* (Figura 1.1, imagem à direita). Este CBIR está disponível na web em http://zomax.wins.uva.nl:5345/ret_user/ (data do último acesso, 30/03/2003).

trodução à Inteligência Artificial e Aprendizado de Máquina, bem como a descrição de técnicas da IA utilizadas nesta dissertação é feita no **Capítulo 3**. No **Capítulo 4** fazemos um aprofundamento do problema das Superfícies de Classificação em RNAs do tipo MLP. O **Capítulo 5** apresenta a implementação do programa IMatch que utiliza técnicas de IA para classificar e recuperar imagens e os resultados dos experimentos realizados. E, por fim, o **Capítulo 6** apresenta as conclusões deste trabalho.

1.3 OBJETIVOS

Os sistemas atuais de recuperação de imagens utilizam-se de técnicas baseadas em uma medida de distância entre as características extraídas das imagens. Esta abordagem permite a recuperação de imagens que possuem características similares através da comparação entre uma *imagem-exemplo* e imagens de uma base de dados, portanto executando uma **comparação um-a-um**. O objetivo é obter uma **generalização** através do aprendizado das características de uma **coleção de imagens-exemplo** pertencentes a uma mesma classe obtendo, assim, um modelo que identifique esta classe de imagem. Neste trabalho iremos aplicar algumas das técnicas de ML da IA com aprendizado supervisionado na tarefa de classificação e recuperação de imagens *bitmap* com o objetivo de obter esta generalização.

Para ilustrar, podemos citar como exemplo a recuperação de imagens que contenham diferentes tipos e ou vistas de latas de Coca-Cola (Figura 1.4) em sua composição sob diferentes tipos de iluminação, escala e ou rotação. Cada ângulo, versão ou tipo de iluminação irá produzir um conjunto de características com algum grau de semelhança entre si, estas características serão aprendidas, generalizadas e mapeadas para a classe *Lata de Coca-Cola* na fase de aprendizado.



Figura 1.4 - Exemplo de generalização

1.4 Objetivos Específicos

Como objetivos específicos este trabalho irá abordar os seguintes tópicos:

- Revisão do estado-da-arte de técnicas de extração de características utilizadas nos sistemas CBIR;
- Revisão dos conceitos de ML utilizados neste trabalho;

- Análise do problema de Superfícies de Separação em RNAs do tipo MLP e seu impacto na tarefa de classificação e recuperação de imagens utilizando RNAs do tipo MLP;
- Propor duas abordagens para atacar o problema de Superfícies de Separação;
- Aplicação das técnicas utilizadas em um sistema desenvolvido para a tarefa de classificação e recuperação de imagens;
- Análise dos resultados obtidos com a simulação.

2 SISTEMAS DE INDEXAÇÃO E RECUPERAÇÃO DE IMAGENS

Para atingir a automação na indexação e recuperação de imagens e, deste modo, evitar os problemas decorrentes da anotação manual e interpretação subjetiva feita por pessoas, os CBIR utilizam técnicas de reconhecimento de padrões aplicadas às tarefas de extração de características, geração de índices da base de dados de imagens e para a comparação de características. Antani [ANT 02] e Rui [RUI 97] apresentam uma pesquisa dos principais sistemas e técnicas utilizadas atualmente nas tarefas de indexação e recuperação de imagens e vídeos. Já Schettini [SCH 01] apresenta uma pesquisa mais detalhada de métodos para indexação e busca em base de imagens utilizando cor, que vem a ser o principal atributo da imagem a ser utilizado neste trabalho.

Nos sistemas CBIR, uma imagem (ou região) de exemplo, esboço, descrição textual da imagem ou do assunto ou uma categoria de imagem (dentre várias pré-definidas) é apresentada ao sistema como exemplo da procura desejada (*imagem-exemplo*). A partir deste exemplo, o sistema CBIR aplica métodos de extração de características e executa uma varredura na base de dados de imagens procurando as maiores similaridades entre as características da *imagem-exemplo* e as características extraídas das *imagens-de-busca*, obtendo um conjunto de *imagens-recuperadas* que é apresentado para o usuário como resultado da pesquisa.

A recuperação de imagens pode ser assistida ou não, ou mesmo empregar as duas abordagens. A recuperação utilizando uma fase assistida consiste numa interação com o usuário buscando melhorar a precisão da recuperação através de exemplos positivos e ou exemplos negativos das imagens recuperadas. Os sistemas que empregam as duas abordagens normalmente utilizam a fase assistida para executar um refinamento nas características extraídas, em fase inicial não assistida, visando obter melhor performance de classificação (o sistema IKONA [IKO 02] é um exemplo de sistema que utiliza esta abordagem mista).

Uma iniciativa atual que busca atacar o problema de recuperação / anotação de imagem, vídeo e áudio é o MPEG-7 [DAY 01] que é um padrão para descrever as características de um conteúdo multimídia. Seu objetivo é facilitar, pela padronização, a procura e recuperação de conteúdos multimídia, mas não determina como a extração automática das características e descrições irá ocorrer, nem qual o tipo de programa ou máquina de pesquisa que irá utilizar estas descrições. As descrições das características do conteúdo multimídia são baseadas em: (i) catálogo que contém informações como título, autor e direitos autorais; (ii) informação semântica do tipo quem e o quê compõem o conteúdo; (iii) características estruturais do tipo histograma de cores ou timbre de um instrumento. Este padrão precisa ser bastante flexível e expansível para descrever os conteúdos multimídia em diferentes tipos de aplicações e ambientes. O MPEG-7 define uma biblioteca de métodos e ferramentas composta de:

- Um conjunto de descritores que definem a sintaxe e a semântica da representação das características;
- Um conjunto de esquemas de descrição que especificam a estrutura e a semântica dos relacionamentos entre seus componentes;

- Uma linguagem que define a descrição dos esquemas, *Description Definition Language* (DDL);
- E um ou mais métodos para codificar as descrições com o objetivo de permitir eficiência na compressão, robustez a falhas e acesso aleatório.

A Figura 2.1 exemplifica como o padrão MPEG-7 pretende facilitar a recuperação de imagens baseadas nas suas características. Pode-se, por exemplo, executar pesquisas a partir do descritor HC (1) quando a distribuição espacial das cores não é importante. Caso a distribuição espacial das cores seja importante utiliza-se o descritor (2). Também se poderia procurar uma imagem utilizando o seu descritor para distribuição espacial de bordas (3) ou considerando a forma de um objeto na cena (4).

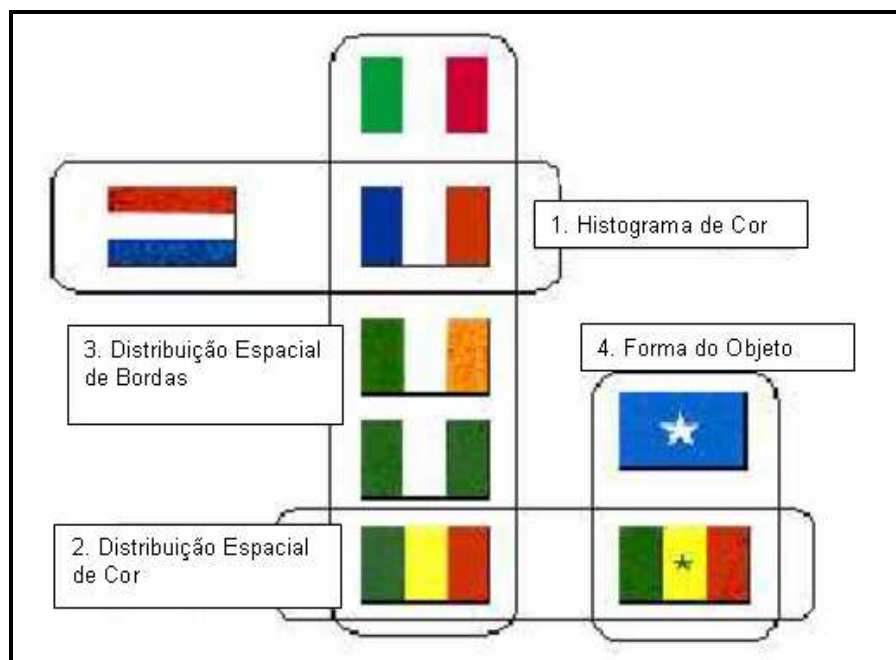


Figura 2.1 - Exemplo de pesquisa utilizando MPEG-7. Adaptação [DAY 01].

2.1 Estrutura Genérica de Sistemas CBIR

Na Figura 2.2 apresentamos um esquema genérico de um sistema atual de recuperação de imagens baseado em pesquisa por exemplos. Existem dois sub-sistemas, o *"Database Generation Subsystem"* que contém as imagens em seu formato original e executa um pré-processamento nelas para extrair as características e armazená-las de forma a permitir uma recuperação eficiente pelo subsistema de recuperação. No *"Database Query Subsystem"* ocorre o processamento da pesquisa feita através da interface do usuário, após a extração de características do exemplo são aplicadas às técnicas de comparação (*matching*) de características entre as características do exemplo e das armazenadas pelo *"Database Query Subsystem"*. As melhores similaridades são então retornadas ao usuário.

2.1.1 Bases de Dados de Imagens Homogêneas

Bases de dados de imagens (BDI) são classificadas como homogêneas (ou de domínio específico) quando as imagens que compõem esta base pertencem a uma mesma categoria, elas possuem uma similaridade implícita entre as imagens como, por exemplo, imagens de impressões digitais, retina ou faces. Neste caso pode-se assumir a certeza de que existem exemplos na base de imagens da imagem procurada. Bases deste tipo permitem se obter uma boa assinatura das imagens que a compõem e permitem uma boa performance porquê permitem que se utilizem métodos específicos para a extração e comparação de características para uma BDI específica [NAS 98].

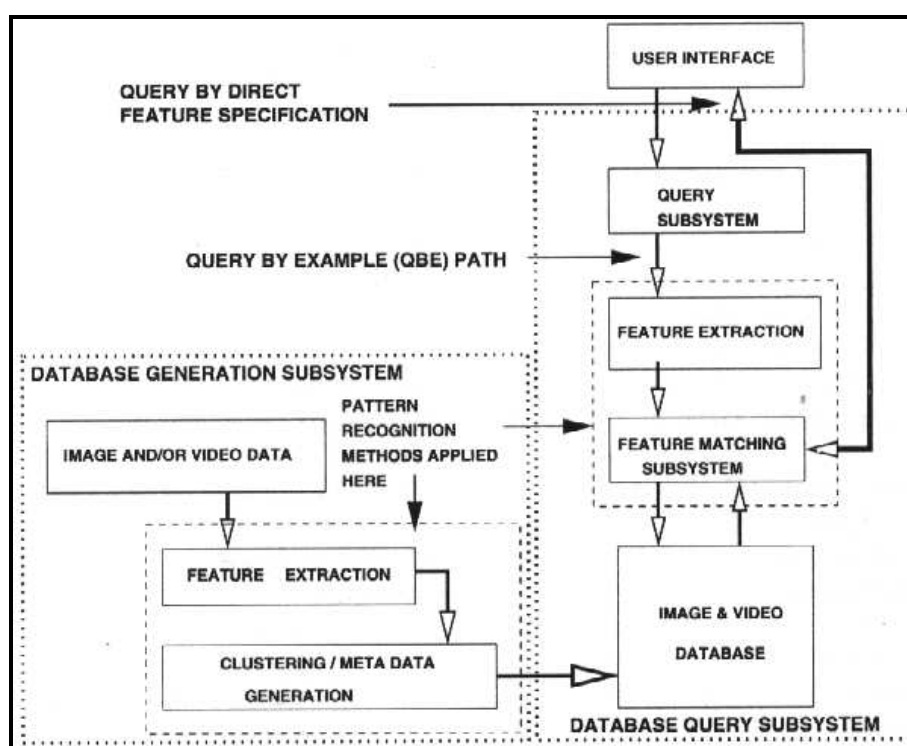


Figura 2.2 - Esquema típico de um sistema atual CBIR [ANT 02].

2.1.2 Bases de Dados de Imagens Heterogêneas

Nas BDIs heterogêneas não existe necessariamente uma similaridade implícita entre as imagens que compõem a base. A similaridade neste caso é subjetiva, dependendo do usuário, do contexto e do tipo de aplicação. Diferentes usuários podem executar pesquisas nesta base com objetivos muito diversificados. Neste caso a obtenção de uma assinatura que caracterize a base não é trivial.

2.2 Extração de Características

O objetivo da extração de características (*feature extraction*) é obter um tipo de “assinatura”, uma transformação do sinal gráfico no domínio espacial para outro domínio mais apropriado (domínio da frequência, por exemplo), da *imagem-exemplo* e das imagens que estão na base de exemplos. Então se aplica uma função de comparação obtendo-se desta maneira, uma medida de similaridade que será utilizada na escolha das melhores imagens. Boa parte dos sistemas atuais utiliza uma extração de características em duas fases [SEB 01]:

- a) obter um vetor de características que identifique algumas das propriedades da imagem e armazenar este vetor em um banco de dados de características;
- b) a partir da imagem de exemplo, calcular o seu vetor de características e comparar com os vetores armazenados no banco de dados recuperando os que possuem os melhores índices de similaridades.

Os métodos **CBIR** dependem inteiramente desta extração de características. As características mais comumente utilizadas segundo Antani [ANT 02] e Rui [RUI 97] são: cor, textura e forma, bem como uma combinação dessas características pode ser utilizada. No domínio visual de características, estas podem ser classificadas em: **características gerais** e de **domínio específico**. Nas características gerais enquadram-se cor, textura e forma e nas de domínio específico podemos ter, por exemplo, características referentes a impressões digitais e faces humanas.

2.2.1 Utilizando Cor

Quando se utiliza a cor como característica para a recuperação de imagens o objetivo é utilizá-la para recuperar todas as imagens que possuem uma composição de cor similar, mesmo que elas sejam de contexto diferente. A cor é uma das características mais utilizadas [ANT 02, CIO 01, GEV 99, RUI 97] porque é relativamente independente quanto ao tamanho, orientação e resolução da imagem e é computacionalmente menos cara. Quando utilizamos a cor podemos querer obter a sua distribuição global (histograma de cores) ou considerar apenas uma parte correspondente a uma região da imagem [SMI 96]. Este trabalho utiliza a cor como característica principal a ser extraída das imagens e por isto iremos apresentar a seguir os fundamentos da cor e os dois sistemas de cores, RGB e HSV, utilizados.

2.2.1.1 Fundamentos da cor

A cor é uma manifestação perceptual da luz quando esta incide (e é refletida) numa superfície ou passa através de um meio não totalmente opaco à luz como, por exemplo, um líquido ou um vidro colorido. Desta interação entre a energia luminosa e o meio, a cor pode se formar através de (i) processo aditivo, (ii) processo subtrativo e (iii) formação por pigmentação. No processo aditivo, ocorre uma combinação de dois ou mais raios luminosos de frequências diferentes. Nes-

ta combinação a formação da cor ocorre pela soma da energia dos fótons. No processo de formação por subtração a luz é transmitida através de um filtro que absorve a radiação luminosa de um determinado comprimento de onda. Também a luz pode ser transmitida através de um corante, que é constituído por partículas que agem como filtros absorvendo radiação luminosa de um determinado comprimento de onda. Na formação por pigmentação os pigmentos podem absorver, refletir ou transmitir a radiação luminosa.

Dois conceitos importantes para a compreensão do conceito de percepção de cor são a **luminância** e a **crominância**. A luminância contém a informação da quantidade de preto e branco que uma cor contém. O cérebro humano compreende esta informação como a quantidade de cinza presente na cor (brilho da imagem). A Figura 2.3 apresenta um exemplo da influência da luminância em uma imagem. A Figura 2.3-C é a imagem original sem alteração no brilho, em A diminui-se o brilho e em B o brilho foi aumentado.

Já a crominância contém a informação da tonalidade da cor (Figura 2.4), é frequência dominante do raio de luz. Estas duas informações combinadas, em diferentes proporções, permitem ao cérebro humano perceber todo o espectro de cores visível numa cena.

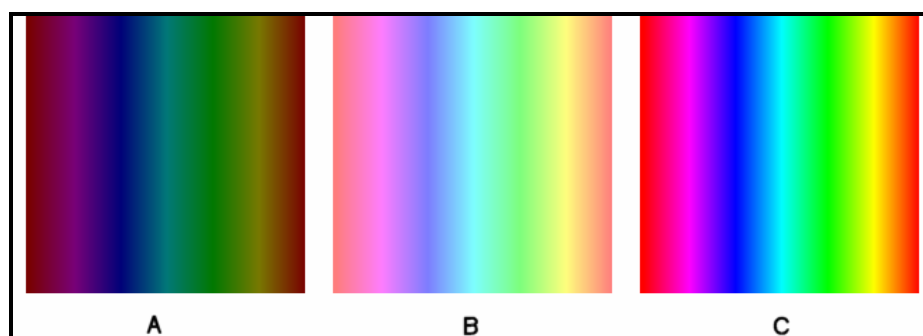


Figura 2.3 - Gradiente de cores com alteração na luminância da imagem.

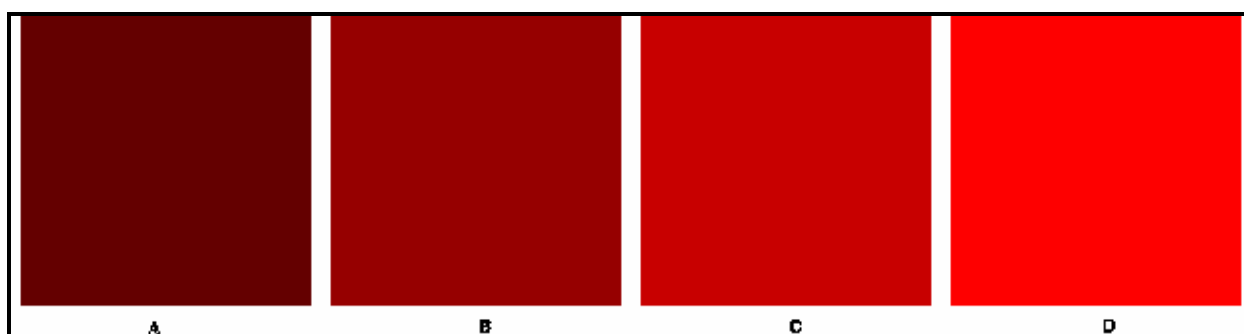


Figura 2.4 – Exemplo de diferentes tonalidades (crominância) para a cor vermelha.

2.2.1.2 Sistemas de cor

Um sistema de cor é um sólido de cor no qual se define um sistema de coordenadas com o objetivo de encontrar uma representação das cores (gamute) que melhor atenda às necessidades de uma aplicação específica. É interessante que o sistema de cor seja definido a partir de um sistema de cor padrão e permita a conversão para outros sistemas de cores.

Podemos classificar os sistemas de cor de acordo com o seu propósito em quatro categorias que são:

- **Sistemas Padrão** estabelecem sistemas de cor universais de cor (CIE-RGB, CIE-XYZ) que são independentes de dispositivo (e.g. monitores de vídeo, impressoras, etc.) e de aplicações;
- **Sistemas dos Dispositivos** são os sistemas definidos para os dispositivos gráficos de entrada (câmeras de vídeo, fotográfica ou scanners), processamento e saída (impressoras e monitores de vídeo, por exemplo);
- **Sistemas de Interface** são sistemas definidos de forma apropriada para os usuários especificarem a informação de cor nos sistemas de tratamento gráfico como, por exemplo, o sistema de cor HSV;
- **Sistemas Computacionais** são sistemas apropriados para aplicações de domínio específico.

Quando se utiliza uma comparação baseada na cor precisamos determinar quanto uma cor difere perceptualmente de outra. Denominamos esta diferença de *distância perceptual no espaço de cor*, e a métrica de *métrica jnd (just noticeable color difference metric)*. Quando esta distância no sistema de cor for igual para um conjunto de cores que possuem a mesma diferença perceptual se diz que este é um sistema **perceptualmente uniforme**. Neste caso a distância entre as cores no sistema de cores pode ser obtida utilizando-se uma métrica euclidiana. Formalmente, um sistema de cor é perceptualmente uniforme se a métrica jnd for igual à métrica euclidiana do sólido de cor.

Para a aplicação neste trabalho iremos nos concentrar nos Sistemas de Interface, que são os sistemas projetados para interagir com o usuário e, portanto, são apropriados para utilização em CBIR. Para mais detalhes nos outros sistemas consultar Gomes e Velho [GOM 94], capítulo 3, Fundamentos da Cor e capítulo 4, Sistemas de Cor. Os sistemas de cor RGB e HSV foram adotados neste trabalho para representar as cores das imagens. O sistema RGB foi adotado por ser um sistema amplamente suportado em CBIR, ser um sistema nativo em diversas bibliotecas de manipulação gráfica e por ser um exemplo “de pior caso” por não ser um sistema perceptualmente uniforme. A escolha do sistema HSV deve-se a este sistema decompor a cor em sua tonalidade predominante e pureza (o que permite a uma função de comparação de características baseada em cor uma discriminação adequada de tonalidades semelhante) e na componente de brilho, que pode variar em diferentes cenas ou condições / tipos de iluminação.

O sistema de cor RGB (*Red, Green, Blue*) é definido em um cubo unitário (Figura 2.5). Pelo ajuste dos valores de R, G, B obtém-se um modelo computacionalmente prático porque permite especificar diretamente as três cores primárias. Estas cores possuem uma relação direta com o monitor de vídeo. Do ponto de vista do usuário este não é um sistema prático porque não permite uma forma fácil de especificar os três componentes de cores que corresponda a um valor de tonalidade, saturação e iluminação. Outra característica do sistema RGB é que ele não é perceptualmente uniforme [CIO 01], portanto as cores especificadas neste sistema não podem ser adequadamente comparadas utilizando-se uma métrica euclidiana.

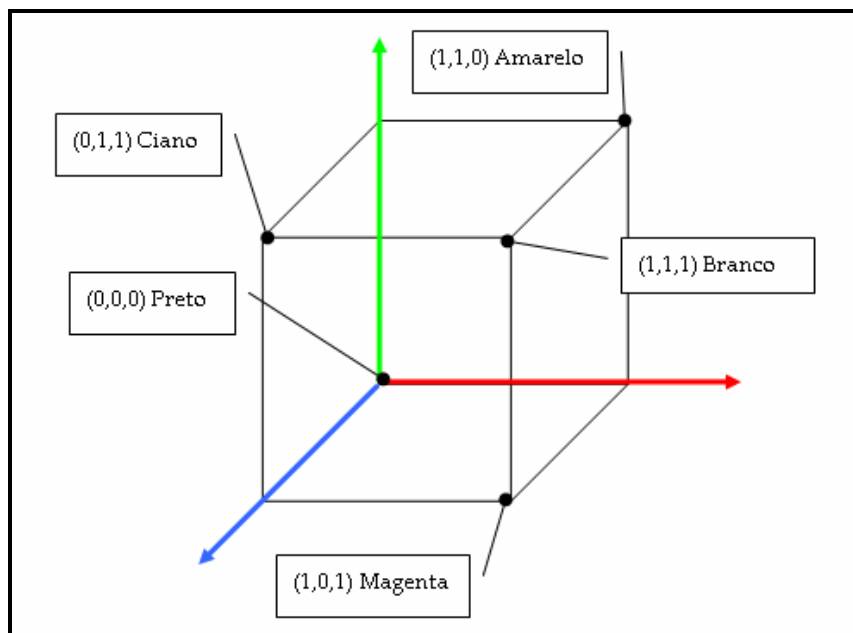


Figura 2.5 – Cubo RGB.

O sistema de cor HSV (Figura 2.6), *Hue, Saturation e Value*, é um modelo definido de forma a ser mais prático de se utilizar porque o usuário pode informar a tonalidade (*Hue*), a quantidade de branco presente na tonalidade (*Saturation*) e o brilho (*Value*), sendo este modo similar ao que artistas utilizam para compor as cores. O sólido de cor deste sistema é um hexacôno, em cada vértice do hexágono encontram-se as cores primárias do RGB ou suas complementares (ciano, magenta e amarelo). O sistema HSV também não é um sistema perceptualmente uniforme (no entanto é “mais” perceptualmente uniforme do que o RGB). Mas optamos por utilizá-lo porque a cor está decomposta na sua informação de tonalidade, qualidade que utilizamos para diferenciar uma “família de cores” de outra (e.g. o quanto vermelho é diferente do amarelo), pureza da cor (*Saturation*) e no brilho (*Value*), que são informações perceptíveis e discriminantes para uma pessoa que observa uma determinada cena.

Como exemplo de sistemas perceptualmente uniformes podemos citar o CIE-Luv e CIE-Lab¹, mas estes sistemas não são muito utilizados em processamento de imagens devido ao seu custo computacional elevado para conversão entre sistemas [GOM 94]. Para uma referência mais aprofundada em outros sistemas de cor bem como em algoritmos para conversão verificar em Gomes e Velho [GOM 94] e Foley et alii. [FOL 90].

Gevers e Smeudlers [GEV 99] demonstram que os modelos normalizados rgb , saturação (*saturation*) S e tonalidade (*hue*) H e os novos modelos de cores por eles propostos $c_1 c_2 c_3$, $l_1 l_2 l_3$ são invariantes a mudanças significativas no ponto de vista, geometria do objeto e iluminação (assumindo iluminação branca e reflexão dicromática nas imagens). Também demonstram que a tonalidade H e o modelo $l_1 l_2 l_3$ são invariantes a pontos de brilho intenso e que o modelo $m_1 m_2 m_3$ não sofre influência da cor da fonte de luz ou de inter-reflexão entre objetos.

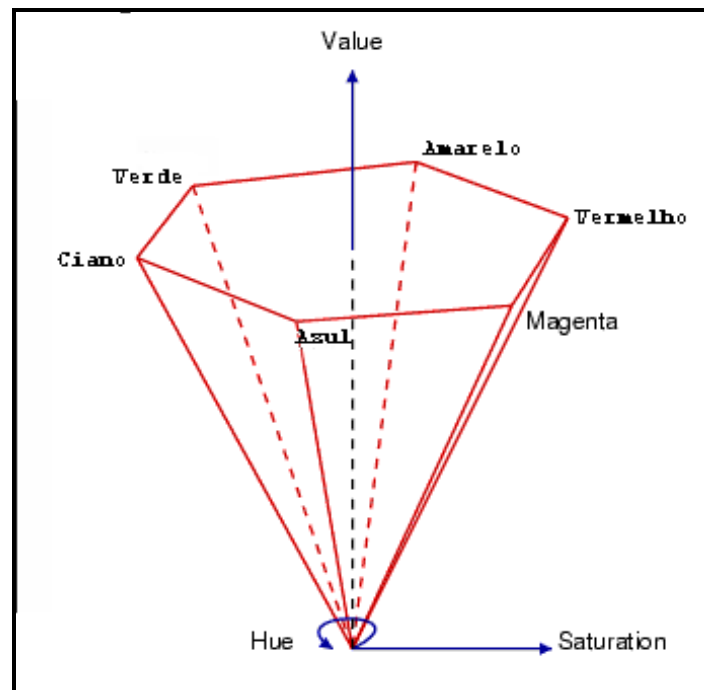


Figura 2.6 – Hexacôno HSV.

A aplicação do método *Retinex* como pré-processamento na imagem antes da fase de extração de características é proposto por Ciocca [CIO 01]. Este método basicamente ajusta a clareza de uma imagem analisando a distribuição espacial das cores. Os resultados apresentados mostram que a eficácia dos métodos de reconhecimento utilizados, momentos da cor em HSV (média, variância e inclinação), histograma de cores, histograma de *color coherence vector* (CCV) [PAS 96] e histograma de cromaticidade espacial, foram superiores quando se aplica o pré-processamento com o método *Retinex*. A utilização dos momentos da cor para a indexação de imagens foi proposta por Stricker e Orengo [STR 95] e fundamenta-se no fato que uma imagem po-

¹ Homepage da COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE): <http://www.cie.co.at/cie/>, visitado em 20/04/2003.

de ser compreendida como uma distribuição de probabilidades das cores dos *pixels* que compõem a imagem.

2.2.1.3 Redução de Cores

A redução de cores de uma imagem consiste em transformar uma imagem que utiliza M bits de cor para N bits, sendo que $M > N$. Ou seja, se temos uma imagem que possui 256 tons de cinza podemos transformar o espaço de cor desta imagem para 64 tons de cinza. Formalmente, a redução de cores é um processo de “discretização” de cor que denominamos de *quantização*. A quantização é uma transformação sobrejetiva $q: C \rightarrow C'$, de um sólido de cor C utilizando M bits de cor para um sólido de cor C' com N bits de cor, sendo que $M > N$. Note-se que o processo de quantização altera a resolução de cor de uma imagem. Utiliza-se a quantização de cores principalmente para reduzir a quantidade necessária de memória para armazenar / representar uma imagem e para transferência entre dispositivos nos quais serão exibidas estas imagens (p. ex. imprimir uma imagem de 16 milhões de cores numa impressora que somente pode exibir 32 mil cores). Quando aplicamos uma quantização nós estamos dividindo um espaço de cor numa série de conjuntos de cores. Cada um destes conjuntos da partição é denominado de célula de quantização e a cada célula está associado um valor constante denominado de nível de quantização. Por exemplo, para quantizar uma imagem de 256 tons de cinza (8 bits) para 16 tons de cinza (4 bits) iremos necessitar de quatro células de quantização. O nível de quantização de cada célula poderia ser 32, 96, 128 e 192. Todos os *pixels* com valor entre 0 e 64 da imagem original receberiam o valor 32 na imagem quantizada, todos os *pixels* com valor entre 65 e 128 na imagem original receberiam o valor 96 na imagem quantizada e assim por diante. A este tipo de quantização denomina-se *quantização uniforme* e o nível de quantização é dado pela equação:

$$q_i = \frac{c_i + c_{i-1}}{2}, 1 \leq i \leq L \quad (3.1)$$

Onde q_i é o nível de quantização da célula de quantização i , c_i é o valor da célula de quantização e L é quantidade de cores que se deseja reduzir. Apesar de fácil de ser obtida a quantização uniforme pode resultar numa imagem que possui cores que não são “fiéis” a imagem original porque o método não considera as cores presentes na imagem original. Para melhorar o resultado (visual) da quantização existem métodos de *quantização adaptativos*. Entre os métodos de quantização adaptativa podemos citar a *quantização por seleção direta*, *quantização por subdivisão recursiva* e *algoritmo do corte mediano*. Iremos detalhar somente o algoritmo do corte mediano que foi o método utilizado neste trabalho. Para maiores detalhes sobre os outros dois métodos ver Gomes e Velho [GOM 94].

O algoritmo do corte mediano (*Median Cut*) foi desenvolvido por Heckbert [HEC 82] e sua idéia básica é subdivisão repetitiva do cubo de cor (no sistema de cor RGB) em retângulos menores. Seja K o nível de quantização desejado. O algoritmo do corte mediano para reduzir para K cores uma imagem é o seguinte:

1. Encontrar o menor e maior valor de vermelho, verde e azul presente na imagem;
2. Estes limites determinam um paralelepípedo de volume mínimo que contém todas as cores presentes na imagem;
3. Encontrar a componente de cor que possui o maior intervalo entre seus limites superior e inferior (que vem a ser a maior aresta deste paralelepípedo);
4. Ordenar as triplas RGB que compõem a imagem pela componente de cor que foi encontrada na etapa 3;
5. Calcular a mediana das triplas de cores ordenadas pela etapa 4;
6. Obtém-se assim duas sub-regiões do paralelepípedo que são cada uma delas uma célula de quantização. Aplicar recursivamente para cada sub-região o algoritmo a partir da etapa 3 até que (a) as duas sub-regiões não contenham mais cores presentes na imagem ou (b) a quantidade K de células foi alcançada;
7. Calcular o nível de quantização de cada célula a partir da média das cores que compõem cada célula de quantização.

O algoritmo do corte mediano é um algoritmo bastante utilizado para a redução de cores devido a sua facilidade de implementação, baixo custo computacional e bons resultados de nível visual [GOM 94].

2.2.2 Utilizando Forma

A extração de características a partir da forma em sistemas CBIR normalmente é abordada utilizando uma ferramenta do tipo QBE ou através de um esboço (*sketch*) da forma que se deseja procurar. A vantagem de se utilizar uma abordagem por esboço é de não ser necessário implementar, no sistema, medidas de similaridade compatíveis com o nível de compreensão humano [ANT 02]. Quando se pretende utilizar a forma para a fase de extração de características, é desejável que as técnicas aplicadas sejam relativamente independentes de translação, rotação, escala e oclusão (parcial) do objeto na imagem porque a percepção de forma de um objeto pode ser alterada devido à projeção de 3D para 2D, o que causa uma “perda de informações”. A representação por formas pode ser dividida em duas categorias: baseadas em fronteira (*boundary-based*) e baseadas em regiões (*region-based*) sendo que o método mais utilizado para representar cada uma das características são Descritores de Fourier e Invariantes de Momento [RUI 97].

2.2.3 Utilizando Textura

Texturas constituem outro método de extração de características utilizado em CBIR e são encontradas em muitos tipos de superfícies como, por exemplo, superfícies de madeira, tecidos, vegetação e nuvens. As texturas são caracterizadas pela distribuição espacial de um padrão de

cores (ou tons de cinza) e homogeneidade deste padrão na imagem e não podem ser caracterizadas através de um histograma unidimensional [RUI 97] [SMI 95] [JAI 95]. Em [DRI 01] texturas são definidas como um padrão que descreve o relacionamento entre sua distribuição cromática e estrutural.

Drimbarean e Whelan [DRI 01] classificam as técnicas de extração de características baseadas em textura em quatro categorias: estatística, baseada em modelo, processamento de sinais e estrutural. Texturas que possuem primitivas pequenas são classificadas como microtexturas e são normalmente tratadas com processos estatísticos tais como: matrizes de co-ocorrência de tons de cinza, contraste, entropia e homogeneidade. Já as texturas cujas primitivas são grandes são classificadas como macrotexturas e requerem um pré-processamento para determinar a forma e as propriedades da primitiva básica. As texturas podem possuir propriedades estatísticas, estruturais ou ambas. Smith e Chang [SMI 95] salientam que é necessário haver na imagem, ou pelo menos na região que está sendo analisada, um grau mínimo de homogeneidade de um padrão que se caracterize como textura.

Neste trabalho vamos considerar as texturas como sendo padrões de combinação de cores e da intensidade (saturação e brilho) destas cores. Portanto a detecção de relações entre os atributos de cor de uma imagem nos permite identificar uma determinada textura.

2.3 Comparação de Características

Na etapa de comparação de características (*matching*) procura-se através de uma **medida de distância** identificar similaridades entre a imagem de exemplo e as imagens da base de dados.

2.3.1 Utilizando Cor

O método mais básico de comparação de características de cor é executar uma comparação pixel-a-pixel da imagem, calculando-se a soma absoluta da diferença (Eq. (3.2)) entre os pixels da *imagem-exemplo* e das *imagens-de-busca*. Na Eq. (3.2), Soma Absoluta da Diferença (SAD) para comparação pixel-a-pixel, soma é medida de similaridade entre as duas imagens, quanto menor for o valor de soma, maior é a similaridade entre as imagens. $nLin$ e $nCol$ são, respectivamente, a quantidade de linhas e colunas das imagens. R_{imgB} , G_{imgB} , B_{imgB} e R_{imgE} , G_{imgE} , B_{imgE} são os valores da componente *Red*, *Green* e *Blue* na *imagem-de-busca* e na *imagem-exemplo*, respectivamente. Este método é sensível a pequenas variações no enquadramento, rotação, escala, oclusão e iluminação da cena.

$$SAD = \sum_{l=1}^{nLin} \sum_{c=1}^{nCol} \left| R_{lc \text{ Im } gB} - R_{lc \text{ Im } gE} \right| + \left| G_{lc \text{ Im } gB} - G_{lc \text{ Im } gE} \right| + \left| B_{lc \text{ Im } gB} - B_{lc \text{ Im } gE} \right| \quad (3.2)$$

Um histograma é uma representação de uma distribuição de ocorrências de alguma grandeza. Um HC (ou histograma de frequência de cores) é uma aproximação da distribuição de

probabilidades das cores presentes numa imagem [GOM 94], e é um dos métodos mais utilizados para se comparar características [ANT 02]. Uma das limitações do HC é que se perde a informação de como as cores estão distribuídas espacialmente na imagem [SEB 01]. O valor adotado para a célula de quantização do histograma influencia na “resolução” das cores para as quais está se calculando a frequência de ocorrência. Valores pequenos para as células de quantização irão gerar um histograma com um grau maior de discriminação na frequência de ocorrência das cores. Valores maiores irão agrupar mais cores similares numa mesma célula acumulando, deste modo, as frequências de ocorrência.

Seja H_1 e H_2 histogramas quantizados em n intervalos e i o seu índice. A definição da diferença entre os dois histogramas é dada pela equação:

$$D = \sum_{i=0}^n |H_1(i) - H_2(i)| \quad (3.3)$$

E sua intersecção é descrita pela equação (valores próximos a 1.0 quando normalizados representam uma similaridade):

$$D_{\text{int}} = \frac{\sum_{i=1}^n \min(H_1(i), H_2(i))}{\sum_{i=1}^n H_1(i)} \quad (3.4)$$

A intersecção de histogramas possui robustez quanto aos problemas de mudança no ponto-de-vista, oclusão e resolução da imagem.

Brunelli e Mich [BRU 99] apresentam uma análise da eficiência do uso de histogramas para a recuperação de imagens utilizando uma nova abordagem de *curva de capacidade de histograma*. Os resultados sugerem que a norma L_1 (ver seção 2.3.3, Medidas de Similaridade) é a mais eficiente em relação à outras medidas de similaridade para a indexação de imagens por histograma e que podem-se utilizar histogramas com baixa resolução (histogramas com um valor de célula de quantização alto) sem perda significativa de sua capacidade de indexação.

Huang [HUA 97] propõe o uso de *correlogramas de cor* (*Color Correlograms*) como uma nova característica baseada em cor para indexação e recuperação de imagens. Este método inclui a informação de como as cores estão espacialmente distribuídas, pode ser utilizado para descrever a representação, global ou de uma região, das correlações de cor na imagem e requer pequena quantidade de memória para armazenar esta informação.

2.3.2 Utilizando Textura

Smith e Chang [SMI 95] propõem um método para comparação de características baseados em texturas a partir de um conjunto de características obtidas dos padrões de energia

espacial e frequência das sub-bandas da imagem. As informações de textura são extraídas de regiões da imagem utilizando-se *wavelets* com filtros *QMF* (o filtro de Haar foi o escolhido) e então são armazenadas numa forma binária que permite uma indexação eficiente das regiões de textura.

Os filtros de Gabor constituem outro método para extração de características de texturas porque permitem que elas sejam decompostas no domínio de espaço e frequência, resultando em boa discriminação da textura e segmentação. A utilidade do método também é reforçada, pois foi demonstrado que a tarefa de segmentação de texturas no sistema visual humano utiliza-se de informações do tipo espacial e frequência [DRI 01] [SMI 95]. Outra abordagem baseada em estudos psicológicos de como o sistema visual humano percebe as texturas e citada em [RUI 97] foi feita por Tamura que identificou seis propriedades visuais (significativas para humanos) que são: granularidade, direcionalidade, semelhança, regularidade e aspereza.

Paschos [PAS 00] propõe uma nova técnica para o reconhecimento de texturas baseadas no conceito de cromaticidade definida pelo modelo de cor CIE XYZ. A idéia básica é a partir do diagrama de cromaticidade e do histograma da imagem obter um vetor de característica composto pelos momentos destas duas medidas.

A descrição de outros métodos utilizados na comparação de características de texturas tais como matrizes de co-ocorrência podem ser encontrados em [ANT 02] e [RUI 97].

2.3.3 Medidas de Similaridade

Para decidir se duas imagens são ou não similares é necessário comparar as características extraídas. Esta comparação envolve algum tipo de mensuração utilizando alguma métrica. A seguir iremos abordar alguns tipos de medidas de similaridade (ou de “distância”) e métricas utilizadas.

As medidas de similaridades podem estar no espaço **Euclidiano** (espaço métrico) ou serem **não-métricas**. Muitos métodos assumem incorretamente que o vetor de características a ser comparado está no espaço Euclidiano [ANT 02], onde a percepção visual humana de similaridade nem sempre se encontra neste espaço [RUI 97] [JAC 00]. Nos trabalhos estudados, as medidas não-métricas por vezes referem-se a uma medida qualitativa, de ordenação por posição ou de distância. É interessante que uma função de distância $d(i,j)$ entre duas imagens possua as seguintes propriedades:

1. $d(i,j) \geq 0$;
2. $d(i,j) = 0$ se e somente se $i = j$ (positividade);
3. $d(i,j) = d(j,i)$ (simetria);
4. $d(i,k) \leq d(i,j) + d(i,k)$ (desigualdade triangular).

No contexto de classificação e recuperação podemos fazer os seguintes comentários sobre o porque das funções de distância observarem estas quatro propriedades. Sejam duas imagens i e j que se deseja verificar a diferença entre elas. Pela propriedade (1), a distância entre i e j deve ser igual a zero (i e j são idênticas) ou maior do que zero (i e j possuem um grau de diferença determinado por $d(i,j)$). Caso $d(i,j)$ for menor do que zero, a imagem j seria mais semelhante a i do que própria imagem i em relação a ela mesma! Se a propriedade (2) não for satisfeita, existem imagens na BDI que são diferentes, mas não é possível distingui-las com a função de distância utilizada. Se a simetria (3) entre as imagens i e j não for observada, poderíamos ter como resposta da função de diferença um resultado do tipo: a imagem i é similar a j , mas a imagem j não é similar a i . Já a propriedade da desigualdade triangular (4) nos diz que entre três imagens, i , j e k , se i é similar a j e i é similar a k , j e k devem ser similares também.

Nas tarefas de recuperação de imagens é normal que o vetor de características possua uma dimensão grande (normalmente de 10^2 [RUI 97]), calcular um tamanho vetor de tamanho como este possui um custo computacional elevado. Visando reduzir este custo de processamento é interessante aplicar algum método para reduzir esta dimensão, entre os quais podemos citar: análise de componentes principais (PCA) [HAY 01] ou clusterização tipo *k-Means* [KAY 99] [THE 98].

Existem vários tipos de métricas para se mensurar as distâncias, as mais comuns são a Euclidiana, *Minkowski*, *Manhattan* e *max metric*. Uma breve descrição de cada métrica é dada a seguir. Dado dois pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$, a distância Euclidiana é definida pela equação:

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.5)$$

Definimos a distância de Minkowski entre dois pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$ no plano X-Y com a equação:

$$d_p(P_1, P_2) = \left[|x_1 - x_2|^p + |y_1 - y_2|^p \right]^{\frac{1}{p}} \quad (3.6)$$

Variando p obtêm-se diversas métricas de distâncias. Se $p = 1$ se obtém a métrica de Minkowski de primeira ordem, também conhecida como *city block distance*. Se $p = 2$ temos a métrica de Minkowski de segunda ordem que vem a ser a distância Euclidiana.

A métrica Manhattan (ou Norma L_1) para determinar distâncias, também conhecida como métrica do taxista (*taxicab metric*), tem sua origem no problema de determinar as distâncias entre ruas de Manhattan. Por exemplo, qual é a distância entre a esquina 33° e 1° com a 69° e 5°? São 36 blocos e 4 avenidas, ou 40 unidades. O objetivo da métrica de Manhattan é abstrair o conceito de mensurar distâncias em termos de metros para alguma medida que possa representar distância. A equação para calcular a distância dois pontos $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$, poderia ser expresso pela Eq. (3.7), que vem a ser a métrica de Minkowski de primeira ordem.

$$d(P_1, P_2) = |x_1 - x_2| + |y_1 - y_2| \quad (3.7)$$

A *max metric*, também é conhecida como métrica L_∞ , é definida na Eq.(3.8), onde d é distância e $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$ são dois pontos. A *max metric*, Eq. (3.8), reduz n dimensões para apenas uma, que será a maior dentre todas as dimensões através da função *max*.

$$d(P_1, P_2) = \max \left[(x_1 - x_2), (y_1 - y_2) \right] \quad (3.8)$$

$$SSD = \sum_{l=1}^{nLin} \sum_{c=1}^{nCol} (x_{1lc} - x_{2lc})^2 + (y_{1lc} - y_{2lc})^2 \quad (3.9)$$

As métricas mais utilizadas para determinar distâncias em visão computacional são as métricas soma do quadrado das diferenças (Eq. (3.9)) e a soma absoluta das diferenças (Eq. (3.2)). A escolha da métrica é avaliada verificando-se a distribuição de ruído aditivo. Quando esta distribuição for Guasiana utiliza-se a métrica SAD, se a distribuição for do tipo exponencial a SSD é a mais indicada [SEB 00].

Deve-se considerar executar a normalização das características extraídas da imagem porque podemos estar trabalhando com as grandezas em unidades diferentes. Quando estamos utilizando um vetor de características composto por características diferentes (por exemplo, cor, saturação ou textura) pode ser necessário atribuir um peso para cada uma destas características para que elas sejam consideradas na comparação de características com a real importância. Ajustar estes valores de normalização e ponderação na função que irá executar o *matching* pode ser uma tarefa complicada.

2.4 Sistemas Atuais para Classificação e Recuperação de Imagens

Entre os vários CBIR existentes, comerciais e de pesquisa acadêmica, podemos citar aqueles mais conhecidos [ANT 02, RUI 97, PEN 94, MIN 96, IKO 02, NAS 98, DAS 97]: QBIC, Virage, RetrievalWare, VisualSEEK (e WebSEEK) [SMI 95, SMI 96, SMI 97], Netra, MARS, VideoQ, Blobworld, AMORE, Leiden, Photobook [PEN 94, MIN 96], PicToSeek, IKONA (Surfimage) [IKO 02, NAS 98], FOCUS [DAS 97]. A seguir faremos uma breve descrição de três destes sistemas que consideramos mais relevantes em relação ao nosso estudo porque permitem indexação baseada em atributos de cor (em diferentes sistemas de cor) e possuem algum tipo de refinamento da pesquisa. O sistema QBIC da IBM possui relevância especial por ser um dos primeiros sistemas a serem utilizados fora do ambiente acadêmico, sendo dos três sistemas aqui descritos o único a ser comercializado.

2.4.1 IKONA (Surfimage)

O IKONA (Surfimage) [IKO 02] [BOU 01] [NAS 98] é um sistema CBIR desenvolvido pelo *INRIA-IMEDIA Project Research Team* e implementado utilizando uma arquitetura cliente-

servidor. O Ikona vem a ser uma evolução do sistema Surfimage [NAS 98]. O método de recuperação padrão é uma pesquisa por similaridade visual a partir de uma imagem de exemplo (QBE). É possível também executar uma pesquisa especificando-se uma região da imagem ou por palavras-chave. O sistema também oferece uma medida de relevância para cada *imagem-recuperada* para auxiliar numa recuperação mais precisa.

É possível utilizar o IKONA com bases de dados de imagens homogêneas ou heterogêneas. Em bases homogêneas é possível executar um ajuste fino nos parâmetros do sistema obtendo-se assim uma melhor eficiência. Quando se utiliza uma base heterogênea o sistema oferece um índice de relevância para cada imagem recuperada, o que permite ao usuário refinar a pesquisa interativamente indicando um conjunto de imagens como sendo exemplos **positivos** e **negativos**. A combinação de características extraídas das imagens é possível, o que em bases heterogêneas é uma estratégia que permite melhores resultados de imagens recuperadas.

O sistema oferece uma série métodos extração de características para serem aplicadas na preparação da base de *imagens-de-busca*. A escolha de quais métodos a se utilizar é tarefa de quem prepara a base. Dentre as características de baixo-nível que o sistema dispõe podemos citar: histograma de cor e orientação, co-ocorrência e transformadas de Fourier e Wavelets. Nas de alto-nível tem-se, por exemplo: autoimagens (*eigenimages*) e formato do espectro da imagem. A comparação (*matching*) é feita utilizando-se uma das medidas de distância implementadas (distâncias L_p de Minkowski, métrica do Co-seno, métrica de Hellinger entre outras).

A Figura 2.7 apresenta um exemplo de recuperação de imagens a partir de *imagem-exemplo* no sistema IKONA. Das seis *imagens-exemplo* disponíveis, escolhemos a imagem que está na segunda linha, terceira coluna como sendo o exemplo apresentado ao sistema. A partir desta escolha o sistema recuperou outras seis imagens como pode ser visto na Figura 2.8. Para cada imagem está atribuído um valor de distância, quanto menor for este valor, maior será a similaridade entre a *imagem-exemplo* e a *imagem-recuperada*.

2.4.2 Photobook

Photobook¹ [PEN 94] é um conjunto de ferramentas para navegação (*browsing*) e procura de imagens a partir de uma base de dados desenvolvido no *Media Laboratory Perceptual Learning – M.I.T.* Nesse sistema são extraídas um conjunto de características que podem ser do tipo cor, forma, textura ou faces. A tarefa de comparação destas características é feita com a utilização de bibliotecas de algoritmos de *matching* externos ao Photobook. Com o objetivo de melhorar a performance (já que não é possível encontrar um modelo genérico ótimo para todas as imagens) o sistema incorpora um **agente de aprendizado iterativo** *FourEyes*² [MIN 96] sendo este um diferencial em relação aos outros sistemas. Com esta ferramenta de aprendizado o usuário seleciona regiões da imagem, que serão exemplos positivos ou negativos, atribui um rótulo a esta região e o sistema

¹ <http://web.media.mit.edu/~tpminka/photobook/>, data da última visita em 21/04/2003.

² <http://web.media.mit.edu/~tpminka/photobook/foureyes/>, data da última visita em 21/04/2003.

extrapola este rótulo para outras regiões da imagem. A partir das características extraídas das regiões o *FourEyes* forma um modelo daquele rótulo para aquele espaço de características. Quando é feita a extrapolação o *FourEyes* seleciona e combina os modelos disponíveis na sua coleção de modelos (*society of models*) para formar os rótulos.

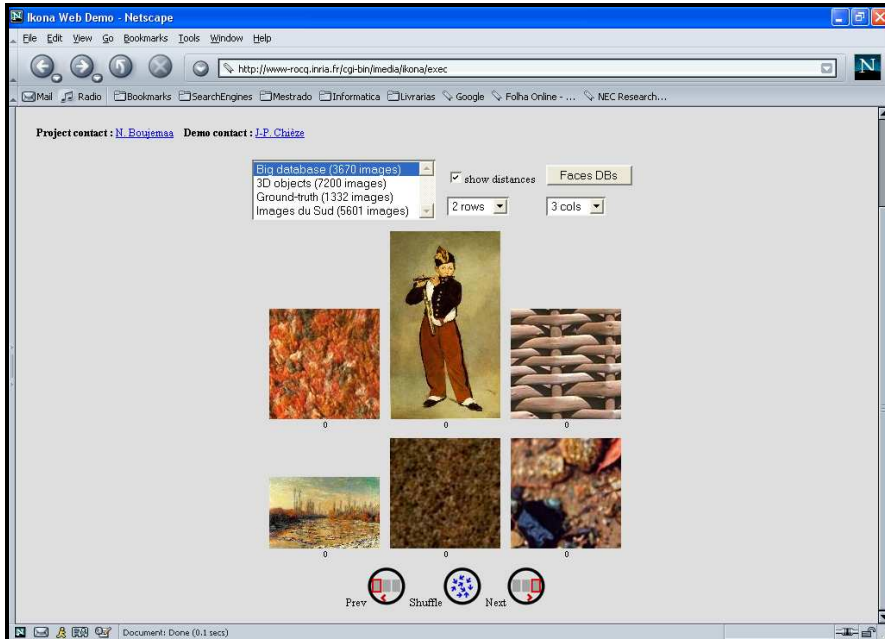


Figura 2.7 - Exemplo de pesquisa no sistema IKONA. Escolha de uma imagem para ser usado como exemplo (segunda linha, terceira coluna).

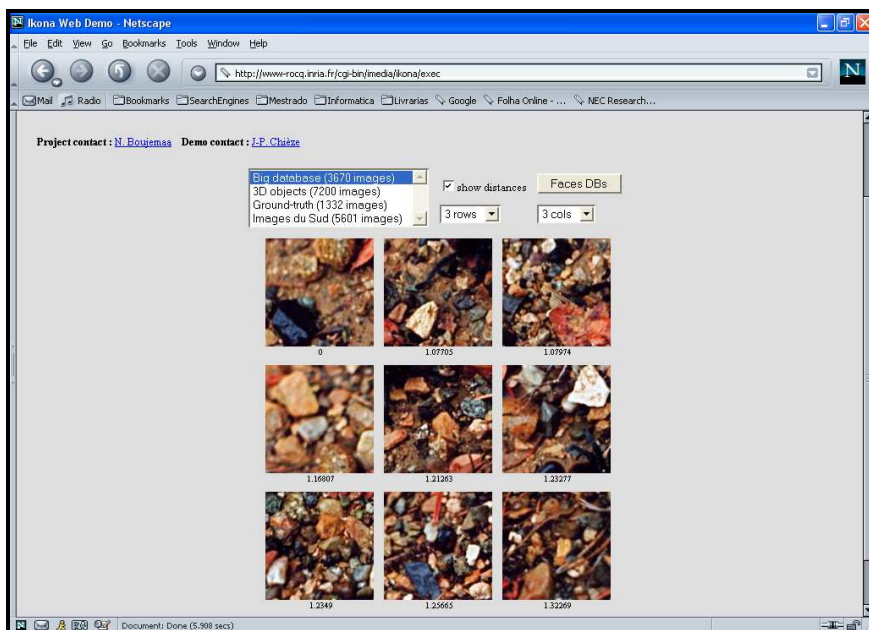


Figura 2.8 - Exemplo do resultado de uma busca por *imagem-exemplo* no IKONA.

2.4.3 QBIC

O sistema QBIC¹ (Query by Image Contents) [FAL 94] foi desenvolvido pela IBM para executar pesquisa e recuperação de imagens em base de dados de imagens baseado no conteúdo visual da imagem. As características extraídas das imagens utilizadas pelo sistema são derivadas da cor (histograma de cor), textura (granularidade, contraste e orientação) e forma (área, excentricidade, orientação do maior eixo, entre outros).

O sistema permite especificar as características que se deseja procurar tais como composição e proporção de cores, disposição espacial das cores na imagem e textura. Utilizando-se uma paleta de cores o usuário informa ao sistema quais composições de cores se deseja encontrar nas imagens da BDI. Por exemplo, na pesquisa por cores o usuário pode especificar até cinco cores que serão procuradas nas imagens e na pesquisa por disposição de cores (*layout search*) podem-se especificar formas geométricas (círculos e retângulos coloridos) que representam a disposição e proporção daquela cor na imagem que se deseja recuperar. Também é possível utilizar uma *imagem-exemplo* no estilo QBE. O sistema foi implantado para o *The State Hermitage Museum: Digital Collection*² e permite a procura na sua base de imagens de pinturas. Este foi o primeiro sistema comercial do tipo e influenciou muitos outros sistemas posteriores que procuram melhorar as técnicas que foram empregadas no QBIC.

2.5 Resumo do Capítulo

Neste capítulo apresentamos o conceito de Sistemas de Classificação e Recuperação de Imagens. CBIR atuais normalmente estão organizados em dois sub-sistemas bem definidos, o “*Database Generation Subsystem*” e o “*Database Query Subsystem*”. Até o presente momento não se possui um método ótimo para comparação e ou extração de características das imagens para qualquer tipo de BDI. Por isto os CBIR procuram subdividir as suas BDI para que cada base contenha a maior quantidade de imagens com conteúdo semelhante. Também se verifica que uma das abordagens possíveis para melhorar a recuperação de imagens em bases não homogêneas é incluir uma fase assistida por uma pessoa para auxiliar no refinamento da recuperação de imagens para determinadas BDI.

A quantidade de métodos utilizados para extração e comparação de características em CBIR é bastante grande atualmente. Esta grande variedade demonstra o esforço que vem sendo feito para encontrar uma transformação adequada do sinal gráfico que permita um alto grau de discriminação do conteúdo gráfico e que este possa ser utilizado em comparações que produzam resultados equiparáveis, ou aceitáveis, ao julgamento de semelhança gráfica entre imagens por pessoas. Os métodos de extração estão baseados em características gráficas percebidas por pessoas,

¹ <http://www.qbic.almaden.ibm.com/>, data da última visita em 21/04/2003.

² <http://www.hermitagemuseum.org/fcgi-bin/db2www/qbicSearch.mac/qbic?selLang=English>, data da última visita em 21/04/2003.

como por exemplo, cor, textura, forma e sombra. Mas os métodos de comparação são fortemente calcados em uma função de diferença. As métricas utilizadas por estas funções de diferença nem sempre representam adequadamente a forma humana de perceber a diferença entre imagens. A percepção humana de semelhança nem sempre ocorre em espaço métrico, com distâncias euclidianas, mas em um espaço que inclui contexto, conceitos e medidas qualitativas. Modelar uma função de distância compatível com a percepção humana de semelhança é uma tarefa difícil, como pode ser visto no grande número de funções de distâncias utilizadas pelos CBIR. Neste contexto de buscar uma alternativa para modelar a percepção humana de semelhança entre imagens é que estamos propondo a utilização de Aprendizado de Máquina que iremos abordar no próximo capítulo.

3 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) é uma área da Ciência da Computação que vem se desenvolvendo aceleradamente nas últimas décadas. Entre seus objetivos estão o estudo e desenvolvimento de técnicas de aprendizado, representações do conhecimento e capacidade de se adaptar a novas situações. No momento que tornamos possível o aprendizado por uma máquina temos uma ferramenta que pode “descobrir” de modo automático o caminho que leva a este conhecimento sem termos a necessidade de explicitar ele de modo algorítmico através de uma linguagem de programação. Entre as áreas da IA que vem sendo estudadas atualmente está a **Visão Computacional** que busca uma maneira de desenvolver a habilidade de máquinas poderem ver e “entender” o que estão vendo, portando, reproduzir em máquinas a capacidade de reconhecimento de objetos, faces, árvores entre outros. Neste contexto é que este trabalho foi desenvolvido, realizando um estudo sobre a aplicação de técnicas de IA à classificação e recuperação de imagens *bitmap*. A seguir introduziremos o conceito de aprendizado de máquina e as técnicas de IA que foram utilizadas neste trabalho.

Mitchell [MIT 97] define o aprendizado de máquinas (*Machine Learning*, ML) como: “Diz-se que um programa de computador aprende com uma experiência E a partir de alguma classe de tarefas T e uma medida de performance P , se sua performance nas tarefas em T , medidas por P , melhoram com a experiência”. Portanto, para termos um problema de aprendizado de máquina bem definido devemos identificar estes três pontos: a **classe da tarefa**, uma **medida de performance** a ser melhorada e a **fonte de experiência**. Além destes pontos precisamos definir ainda uma **função alvo** e uma **função de aproximação**.

A *classe da tarefa* identifica a tarefa a ser aprendida. Neste trabalho pretende-se aprender a identificar composições de cores que discriminem o objeto procurado, por exemplo, latas de Coca-Cola ou logotipos.

A *medida de performance* é a quantidade de identificações corretas dos padrões aprendidos nas *imagens-exemplo* e a quantidade de rejeições corretas de padrões. Ou seja, dada uma imagem, não deixar de reconhecer um padrão aprendido (falso-negativo) e não reconhecer erroneamente um padrão na imagem (falso-positivo).

A *fonte de experiência* determina como será obtido o conhecimento que pode ser supervisionado ou não. No caso deste trabalho adotaremos o aprendizado supervisionado, onde a experiência será fornecida por um “professor” que irá apresentar uma série de imagens **previamente classificadas** que contém os padrões a serem aprendidos. Por exemplo, a fonte de experiência seria composta por uma série de *imagens-exemplo* que contém partes de latas de Coca-Cola vista de diferentes ângulos, sob diferentes tipos de iluminação e de diferentes estampas identificadas como pertencentes à classe *Coca-Cola* e possivelmente também serão apresentados exemplos de imagens não pertencentes à classe *Coca-Cola* (*imagens-de-fundo, contra-exemplos*).

A *função alvo* é uma função que permite melhorar a medida de performance a partir da fonte de experiência a cada iteração do aprendizado. Como pretendemos reconhecer padrões em imagens a função alvo ideal será aquela que irá maximizar a detecção de uma “assinatura” para uma classe de imagens a partir de suas características (cor, textura, forma ou algum processamento envolvendo estas características primitivas) e minimizar a ocorrência de falsos-positivos.

Obter uma função alvo ideal tende a ser computacionalmente uma tarefa cara de ser atingida, por isto busca-se uma aproximação desta função. A *função de aproximação* é uma função que visa diminuir a complexidade computacional da função alvo ideal. No nosso caso, se montarmos um vetor de características para uma imagem de 64×64 pixels, com três componentes de cores (RGB), teríamos um vetor com uma dimensão de 12288 elementos. A utilização de uma técnica que permita reduzir este vetor, em quantidade de elementos ou dimensões, traria uma redução no custo computacional da função alvo.

Podemos classificar os métodos utilizados no aprendizado de máquina quanto à origem da fonte de experiência como sendo: (i) não supervisionados e, (ii) supervisionados ou aprendizado com professor. Nos métodos de aprendizagem supervisionados existe um “professor” que é responsável por informar ao algoritmo de aprendizado a “resposta correta” (exemplos rotulados) para determinada instância da fonte de experiência. A diferença (erro) entre a resposta correta e a obtida pelo método é a *medida de performance* que será utilizada para corrigir o aprendizado. Portanto, nos métodos supervisionados existe uma convergência para a resposta correta. Como exemplo de abordagens que utilizam este método podemos citar RNAs MLP [HAY 01] e Árvores de Decisão [RUS 95, MIT 97]. Nos métodos não supervisionados é o próprio algoritmo de aprendizado que irá organizar os exemplos utilizando uma medida de similaridade. Sempre que necessário o método irá criar uma nova classe para agrupar os exemplos que não são similares aos outros. Como exemplos deste tipo de aprendizagem temos Mapas de Kohonen (SOM) [KOH 87] e clusterização [THE 98].

Nas próximas seções iremos abordar em mais detalhes RNAs do tipo *Multilayer Perceptron* (MLP, com múltiplas camadas) e clusterização, em especial o método *k-Means* que foram as técnicas utilizadas neste trabalho.

3.1 Redes Neurais Artificiais

As RNAs, ou redes conexionistas são modelos computacionais que possuem a habilidade de se adaptar, aprender, generalizar, organizar ou “clusterizar” dados [KRO 95]. Sua construção é motivada nos neurônios biológicos, mas ainda não alcançaram o mesmo grau de sofisticação. As redes conexionistas constituem-se de várias unidades básicas, os neurônios, interligados através de conexões ponderadas (conexões sinápticas, que procuram modelar como o cérebro realiza uma tarefa em particular). Haykin [HAY 01] define uma RNA como:

“Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhe-

cimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.” (Haykin, 2001. p. 28).

Uma das características mais importantes de uma RNA é a sua capacidade de generalização do conhecimento adquirido, ou seja, a rede é capaz de produzir saídas adequadas para dados de entrada que não estavam presentes no treinamento da rede. Outras características que consideramos importantes nas RNA para este trabalho são:

1. *Não-linearidade* é a capacidade que uma RNA possui, com topologia adequada, de aproximar uma função não-linear.
2. *Mapeamento de Entrada-Saída* é a capacidade que uma RNA possui de estabelecer uma relação entre um exemplo de entrada e uma saída apropriada.
3. *Adaptabilidade* é a capacidade das RNAs em ajustar seus pesos sinápticos em respostas a modificações no meio ambiente, o que permite que a rede possa operar em um ambiente diferente daquele para o qual ela foi inicialmente treinada.
4. *Resposta a Evidências* é o grau de confiança que uma RNA tem ao classificar um padrão visto quando executa a sua classificação. Este grau de confiança pode ser utilizado para aceitar ou rejeitar a classificação feita pela RNA quando o comparamos com um valor pré-determinado de limiar.

Os métodos baseados em aprendizagem neural oferecem uma abordagem robusta para a tarefa de aproximação de uma função alvo a partir de sua capacidade de generalização, processamento distribuído e adaptabilidade. O domínio desta função inclui valores reais, discretos ou vetoriais, o que torna as RNAs apropriadas para tarefas nas quais os dados são de baixo nível de abstração como, por exemplo, dados numéricos provenientes de sensores ou imagens.

3.1.1 Modelo Computacional de uma Rede Neural Artificial

O neurônio (artificial) é a menor unidade de processamento de informação em uma RNA e é composto por três unidades básicas, a saber:

1. Um conjunto de sinapses que possuem um peso (peso sináptico) associado a cada sinapse. Um sinal de entrada x_j aplicado à sinapse j e conectado ao neurônio k é multiplicado pelo peso w_{kj} . É nos pesos sinápticos que o “conhecimento” adquirido por uma RNA está representado;

2. Um somador usado para integrar os sinais de entrada. Estes sinais são multiplicados pelos respectivos pesos sinápticos associados a cada sinapse;
3. Uma função de ativação para restringir a amplitude do sinal de saída.

Um modelo de um neurônio não-linear é apresentado na Figura 3.1. Os sinais de entrada x_1 a x_m são aplicados nas respectivas sinapses sendo multiplicados pelos seus respectivos pesos sinápticos. Estes sinais ponderados pelos pesos sinápticos são somados na junção de ativação resultando no sinal v_k . Este sinal é passado para a função de ativação que irá restringir a amplitude do sinal para um intervalo finito. Neste modelo há uma entrada especial no neurônio para representar o *bias*. O *bias* (ou *offset*) é um sinal que tem o efeito de aumentar ou diminuir a entrada da função de ativação causando um deslocamento (*offset*) no campo local induzido (potencial de ativação do neurônio).

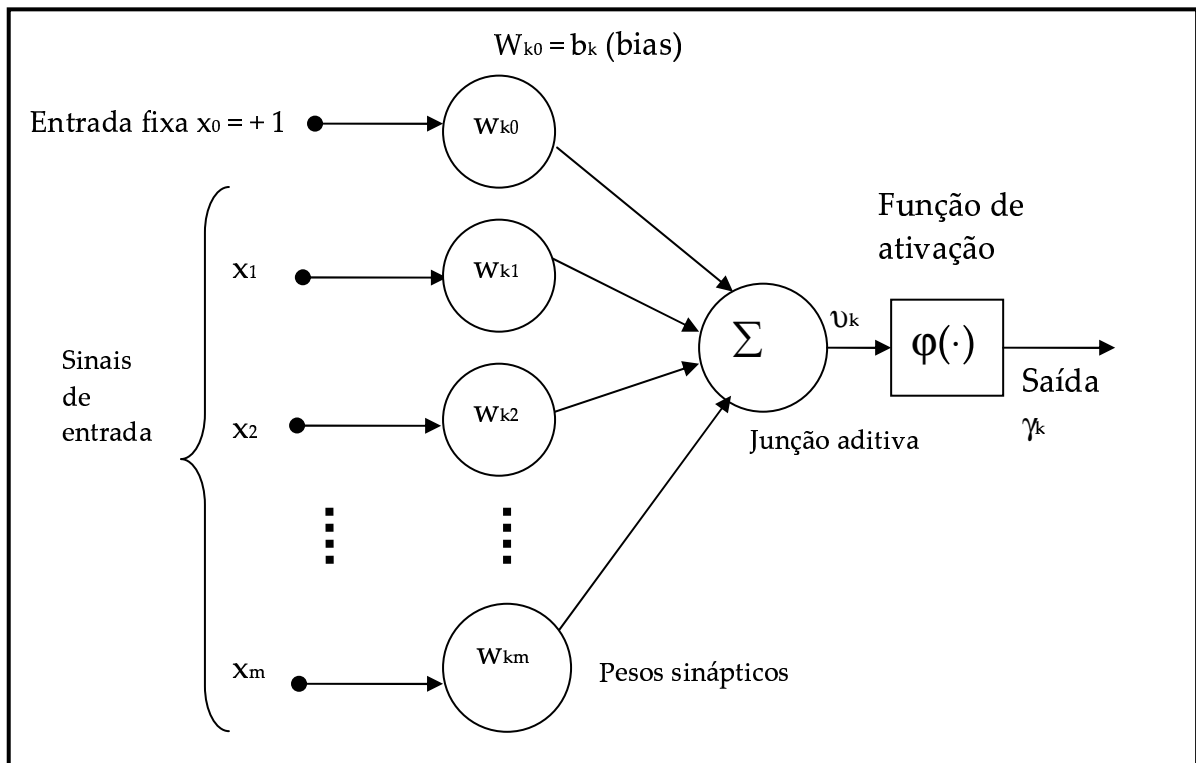


Figura 3.1 – Modelo não-linear de um neurônio [HAY 01].

Matematicamente um neurônio pode ser descrito por duas equações, uma para o campo local induzido e a segunda para a sua saída. A Eq. (4.1) descreve o cálculo do campo local induzido como o somatório das entradas nas sinapses multiplicado por seus pesos sinápticos. O campo local induzido calculado é aplicado na função de ativação Eq. (4.2) para se obter a saída do neurônio.

$$v_k = \sum_{j=0}^m \omega_{kj} x_j \quad (4.1)$$

$$y_k = \varphi(v_k) \quad (4.2)$$

Existem três tipos de funções de ativação normalmente utilizadas no neurônio:

1. *Função de limiar* Eq. (4.3);
2. *Função linear por partes* Eq. (4.4);
3. *Função sigmóide* Eq. (4.5).

A função de limiar é a função utilizada pelo neurônio de *McCulloch-Pitts*. Este neurônio, por utilizar esta função, sempre é ativado quando o campo local induzido for maior ou igual a zero, demonstrando uma propriedade do tipo *tudo-ou-nada*.

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (4.3)$$

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \quad (4.4)$$

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (4.5)$$

A função sigmóide é a função de ativação mais utilizadas em RNAs por possuir um balanceamento adequado entre comportamento linear e não-linear [HAY 01]. O parâmetro a da equação é o parâmetro de inclinação da sigmóide (ou “temperatura” da sigmóide). O efeito de sua variação pode ser verificado na Figura 3.2.

Existem basicamente três tipos de neurônios numa RNA. Os *neurônios de entrada* (nós de fonte), onde o sinal proveniente do meio externo (padrões de ativação) é aplicado, não executam nenhum tipo de processamento de informação, apenas distribuem o sinal para os neurônios seguintes (da camada oculta ou para os de saída). Os neurônios do tipo *saída* (nós computacionais) executam o processamento do sinal conforme ilustrado na Figura 3.1. Existe ainda uma classe de neurônios que fica entre os de entrada e de saída que são denominados de *ocultos*. A função destes neurônios é adicionar poder computacional, o que permite que uma RNA com uma ou mais camadas de neurônios ocultos possam classificar padrões e aproximar funções não lineares como será visto mais adiante.

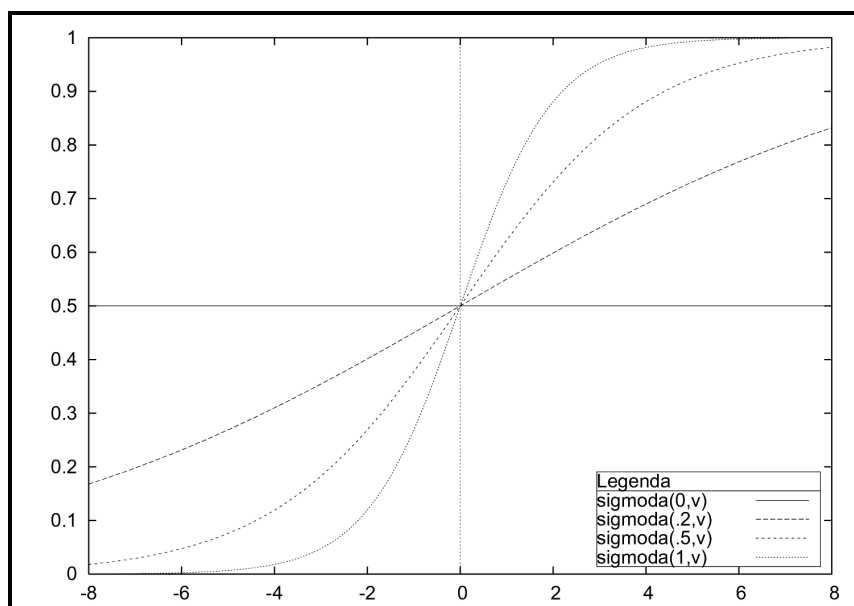


Figura 3.2 – Função de ativação sigmóide (Eq. (4.4)). Temperatura da sigmóide variando em 0; 0,2; 0,5 e 1.

A forma como os neurônios estão interconectados numa RNA denomina-se *arquitetura da rede* ou *topologia da rede*. A arquitetura de uma RNA está relacionada com o algoritmo de aprendizagem que será utilizado para treinar a rede. Existem basicamente três classes de arquitetura de RNA: (i) *redes alimentadas adiante com camada única*; (ii) *redes alimentadas diretamente com múltiplas camadas* e (iii) *redes recorrentes*. As redes alimentadas adiante com camada única (Figura 3.3 – A) possuem uma camada de neurônios de entrada que se conectam com uma camada de neurônios de saída, mas não existem ligações no sentido dos neurônios de saída para os de entrada. As redes nas quais as ligações sinápticas sempre são projetadas em direção as próximas camadas são denominadas de *redes feed-forward*. Já nas redes alimentadas diretamente com múltiplas camadas (Figura 3.3 – B) existe uma ou mais camadas de neurônios ocultos, sempre com as ligações sinápticas em direção as próximas camadas. As conexões sinápticas nas *redes feed-forward* podem ser do tipo totalmente conectada (quando um neurônio de uma camada se conecta a todos os outros neurônios da camada subsequente) ou parcialmente conectada. As *redes recorrentes* (Figura 3.3 - C) são redes que possuem pelo menos um laço de realimentação, normalmente através de um operador de atraso unitário.

3.1.2 RNA do Tipo *Multilayer Perceptron*

As RNAs do tipo *feed-forward perceptron* começaram a ser estudadas a partir do final de 1950 principalmente por Rosenblatt [HAY 01, KRO 95]. A origem do *perceptron* é baseada no neurônio de McCulloch-Pitts (Figura 3.1, Eq. (4.1) e Eq. (4.2)). A topologia inicial destas RNAs era de uma camada porque na época somente existiam algoritmos de aprendizagem eficientes para RNAs de uma camada [RUS 95]. As RNAs do tipo *perceptron* (de uma camada) possuem uma severa limitação, elas só podem representar funções linearmente separáveis. As RNAs MLP, são capazes de aproximar função linearmente e não linearmente separáveis. As redes MLP começaram a

ser utilizada em maior escala com o invento do algoritmo de aprendizagem com retropropagação (*backpropagation* ou *backprop*) [HAY 01, MIT 97, RUS 95]. Um problema que existe numa RNA MLP está em determinar a quantidade adequada de neurônios na camada oculta [RUS 95]. Um estudo sobre as superfícies de separação abertas e fechadas em RNAs do tipo MLP apresentado por Gori [GOR 98] aborda, indiretamente, o problema de determinar a quantidade de neurônios da camada oculta (as superfícies de separação serão abordadas em detalhes no capítulo 4).

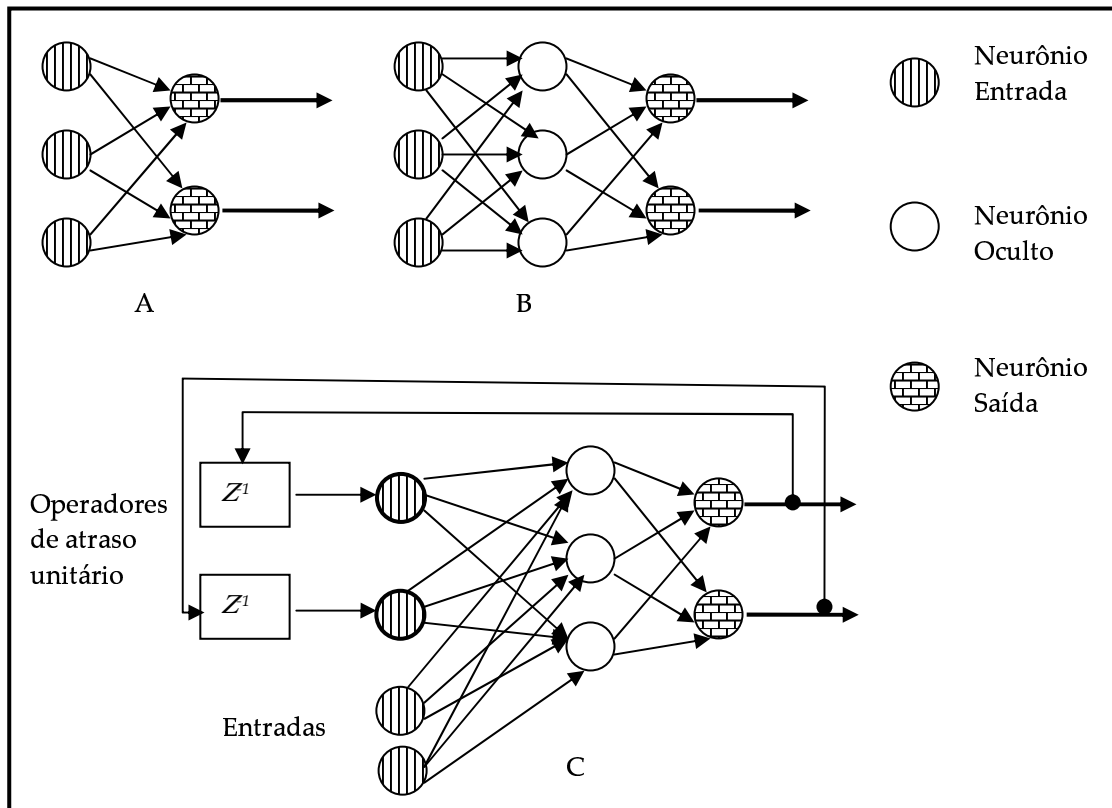


Figura 3.3 – Três arquiteturas de RNA mais comuns: redes alimentadas adiante com camada única; redes alimentadas diretamente com múltiplas camadas e redes recorrentes.

O algoritmo de aprendizagem *backprop* somente se aplica a RNAs MLP que possuam um conjunto fixo de neurônios e conexões sinápticas. A estratégia adotada pelo algoritmo para o ajuste dos pesos sinápticos consiste na utilização de uma pesquisa por gradiente descendente. Esta pesquisa ocorre no espaço dos pesos sinápticos e o gradiente está na superfície de erro, onde se procura minimizar o somatório do erro quadrático entre os valores de saída da RNA e o valor esperado para um determinado exemplo de treino. A retropropagação do erro ocorre a partir dos neurônios de saída em direção à camada oculta precedente, e desta para a camada oculta precedente sucessivamente, caso exista mais de uma camada oculta. Assume-se que cada neurônio da camada oculta é responsável por uma parcela do erro percebido nos neurônios de saída aos quais ele está conectado. O ajuste dos pesos ocorre por épocas, cada época consiste no ajuste de todos os pesos da RNA para todos os exemplos de treinamento.

Como a superfície de erro pode conter vários mínimos locais somente tem-se a garantia deste método convergir para um dos mínimos locais. Os resultados dos experimentos reali-

zados com o *backprop* sugerem que nos casos observados este mínimo local tende a ser também o mínimo global ou uma boa aproximação da solução ideal [MIT 97, FAH 90].

3.1.3 RNA do Tipo MLP com Algoritmo de Aprendizagem *Cascade-Correlation*

A RNA com algoritmo de aprendizagem por correlação em cascata (*Cascade-Correlation*) foi desenvolvida por Fahlman e Lebiere [FAH 90] a partir da análise de performance do algoritmo *backprop*. Investigando os motivos deste algoritmo necessitar de uma quantidade elevada de épocas para convergir os autores identificaram dois problemas. O problema do **tamanho do passo** na descida pelo vetor de gradiente de pesos (*Step-Size Problem*). Se o tamanho do passo for bastante pequeno o aprendizado converge para um mínimo local, mas o treinamento será demorado. Caso o tamanho do passo seja demasiado grande (quanto maior o passo mais rápido será o treinamento), o aprendizado pode não convergir para uma boa solução, onde o algoritmo *backprop* padrão não possui um mecanismo para determinar um tamanho adequado de passo. O segundo problema denominado de **alvo móvel** (*Moving Target Problem*) ocorre porque cada neurônio está ajustando os seus pesos de modo a executar uma computação útil dentro da RNA. O ajuste dos pesos se torna difícil porque todos os neurônios da RNA estão ajustando seus pesos ao mesmo tempo e, como existe a retropropagação do erro, este se comporta como um alvo que está sempre modificando sua posição tornando mais difícil o ajuste dos pesos sinápticos de cada neurônio.

A arquitetura por correlação em cascata é uma arquitetura que permite um aprendizado construtivo e incremental. Esta arquitetura tem duas idéias principais: (i) A “arquitetura em cascata” se deve a abordagem de adicionar um neurônio por vez na camada oculta (quando for necessário) e seus pesos neste momento são **congelados** (este procedimento visa atacar o problema do alvo móvel); (ii) O algoritmo de aprendizagem que é responsável pela criação e instalação do novo neurônio na camada. Este algoritmo busca maximizar a magnitude de correlação da saída do novo neurônio com o erro residual da RNA. O problema do tamanho do passo é resolvido utilizando-se o algoritmo de aprendizagem *quickprop* [FAH 88].

Inicialmente a RNA é constituída apenas por neurônios de entrada e saída. Quando um novo neurônio é instalado pelo algoritmo de aprendizagem ele recebe conexões de todos os neurônios de entrada e de todas as saídas dos neurônios da camada oculta. Este novo neurônio é instalado em uma **nova camada de um neurônio somente** após o ajuste de seus pesos. Neste momento os pesos do novo neurônio são congelados. As únicas conexões que são treinadas novamente são as dos neurônios de saída. Adotar esta estratégia de treinamento faz desnecessária a utilização da retropropagação do erro. O novo neurônio adicionado é “retirado” de um *conjunto de neurônios candidatos* que foram treinados. Todos estes neurônios recebem as conexões de entrada como se fossem uns dos neurônios já participantes da rede, mas a saída dos neurônios candidatos não está conectada à rede. Todos estes neurônios são treinados, ao final do treinamento destes neurônios o que atingiu o menor erro possível, ou seja, o que possuir a maior correlação entre a saída com o erro residual da rede, é inserido e seus pesos sinápticos são congelados. Este processo in-

cremental continua até que a condição de parada estabelecida seja atingida. Em pseudocódigo o processo poderia ser assim descrito:

1. Treinar a saída dos neurônios de saída;
2. Treinar os neurônios do conjunto de neurônios candidatos;
3. Inserir o neurônio candidato que possuir a maior correlação entre sua saída com o erro residual da rede numa nova camada oculta de um neurônio na rede;
4. Congelar os pesos do neurônio inserido;
5. Repetir a partir do passo 1 até a condição de parada ser alcançada.

Comparando-se as redes MLP com algoritmo de aprendizagem *backprop*, *quickprop* e correlação em cascata, as redes que utilizam a correlação em cascata necessitam de menos épocas de treinamento e geram redes mais compactas (com menos neurônios na camada oculta). Elas também apresentam uma menor dependência do ajuste de parâmetros de aprendizado e não requerem que se defina a quantidade de neurônios da camada oculta antes de começar o treinamento.

3.1.4 Utilização de RNAs na Classificação de Imagens

A aplicabilidade das RNA nas tarefas de reconhecimento de padrões tem sido amplamente estudada em vários trabalhos atuais [MCG 01, DRI 01, GOR 98, MIT 97]. Neste trabalho será destacado o uso de RNAs MLP *Cascade-Correlation*.

Drimbaream e Whelan [DRI 01] utilizam um tipo especial de RNA, *Learning Vector Quantization*, para a tarefa de classificação de texturas coloridas utilizando o algoritmo *LVQ1*. As técnicas utilizadas na fase de extração de características para compor o vetor de características utilizado na classificação foram: transformada discreta do co-seno (DTC), filtros de Gabor e matrizes de co-ocorrência. O trabalho também verificou o impacto de diferentes modelos de cor (HSI, CIE-XYZ, YIQ e CIE-Lab) na tarefa de classificação. As conclusões que os autores chegaram foram que o método DTC oferece o melhor resultado e que o modelo de cor empregado não traz contribuições significativas.

Bittencourt e Osório [BIT 02] desenvolveram um trabalho utilizando RNAs MLP para treinar RNAs a agirem como filtro (filtros neurais) para detecção de pele em imagens capturadas de uma *WebCam*. O foco específico deste trabalho é utilizar a detecção de pele para o reconhecimento de gestos. As imagens são fornecidas à RNA no sistema de cor RGB e consideram a vizinhança dos pixels (matriz de contexto). Os autores constatam que existe dificuldade para “construir um detector universal de pele”. Por isto sugerem a utilização de filtros neurais treinados especificamente para cada usuário (base de dados homogênea, ver seção 2.1.1).

Outro trabalho que utiliza RNAs, mas do tipo SOM, foi desenvolvido por Siqueira [SIQ 02] para a obtenção de medidas de estruturas cardíacas fetais a partir de imagens ecocardiográficas (em tons de cinza). A rede SOM é utilizada na etapa de segmentação da imagem (executando um pré-processamento da imagem) e o autor reporta bons resultados obtidos com a técnica aplicada a imagens de ultra-som.

Para reconhecimento de caracteres em placas de veículos automotores brasileiros obtidos a partir de imagens Gesualdi et alii [GES 02] apresentam uma abordagem que utiliza os métodos de análise de componentes principais (PCA) e discriminação de componentes principais (PCD) como métodos de pré-processamento da imagem (em tons de cinza e previamente segmentada) para compor o vetor de características. Este vetor é apresentado como padrões de entrada para uma RNA MLP com algoritmo de aprendizagem *backprop*. Os autores utilizaram duas RNAs, uma treinada para reconhecimento de letras e outra para o reconhecimento de algarismos.

Para a tarefa de sensoriamento remoto, Todt [TOD 98] apresenta uma avaliação do desempenho de classificadores neurais para aplicações em sensoriamento remoto. O trabalho compara a performance dos métodos de Máxima Verossimilhança Gaussiana e RNAs MLP com algoritmo de aprendizagem *backprop* para classificar tipos de cobertura de solo (vegetação nativa, áreas de reflorestamento, pastagens e solos a descoberto) a partir de imagens obtidas de satélites (*LANDSAT 5TM*). As conclusões apresentadas no trabalho sugerem que a utilização de RNAs MLP com *backprop* apresentaram um aumento da acurácia em relação ao método de Máxima Verossimilhança Gaussiana, especialmente na classificação de pixels desconhecidos. Em um trabalho posterior Todt et alii [TOD 02] executaram uma comparação destes resultados com uma nova abordagem utilizando uma matriz 3x3 pixels das imagens (entrada contextualizada) como padrões de entrada de uma RNA MLP com algoritmo de aprendizagem *Cascade-Correlation*. Os resultados obtidos apresentaram uma melhoria na acurácia de até 84,8% (contra 58,9% da abordagem original utilizando RNA MLP com *backprop*).

A escolha pela utilização de RNA para a tarefa de aprendizado nesta dissertação justifica-se por:

- Boa capacidade em lidar com dados quantitativos (valores numéricos no domínio \mathcal{R}) para a tarefa de classificação [HAY 01] [MIT 97]. A fase de pré-processamento (extração de características) das imagens produz basicamente dados neste formato, e pela sua capacidade de obter uma generalização, o que permite que uma RNA apresenta uma resposta satisfatória para padrões de entrada (imagens) que não compuseram os padrões de treinamento;
- Possibilidade de se utilizar uma **coleção** de *imagens-exemplo* o que permite modelo que identifique uma classe de imagem;
- Natureza inerentemente paralela das RNAs, o que permite que a tarefa de classificação e ou recuperação seja distribuídas em várias unidades de processamento. Uma outra abordagem que explora o paralelismo em problemas de reconhecimento de imagens é descrita por Moschetta [MOS 02] que apre-

senta uma aplicação distribuída para o reconhecimento de imagens (utilizando distância euclidiana entre pixels das imagens e diferença entre histogramas) cuja arquitetura poderia ser facilmente adaptada para a utilização com RNAs.

3.2 Clusterização

A clusterização é uma técnica de classificação não supervisionada, portanto não existem rótulos pré-determinados para os padrões de treinamento. Técnicas de clusterização buscam agrupar (ou descobrir a organização que permeia o conjunto de exemplos), em ou mais grupos, um conjunto de exemplos de acordo com uma métrica de similaridade ou dissimilaridade entre os exemplos. Ao final do processo, cada cluster irá conter um representante (protótipo) que identifica aquele grupo. O resultado desta classificação é dependente do critério (medida de similaridade) adotado. Por exemplo, temos um conjunto de programas de computador de diferentes aplicações gravados em diferentes tipos de mídia e queremos “organizar” estes programas de acordo com uma medida de similaridade. Seja o conjunto C de programas compostos por: Windows/CD, Linux/DVD, Editor de Textos/Disquete, Compilador C/CD, Compilador Pascal/Disquete, Enciclopédia/DVD, Planilha de Dados/Fita. Caso o critério de organização fosse agrupar por finalidade teremos a clusterização da Figura 3.4 – A. Se o critério fosse por tipo de mídia no qual o programa se encontra armazenado teremos a clusterização da Figura 3.4 – B.

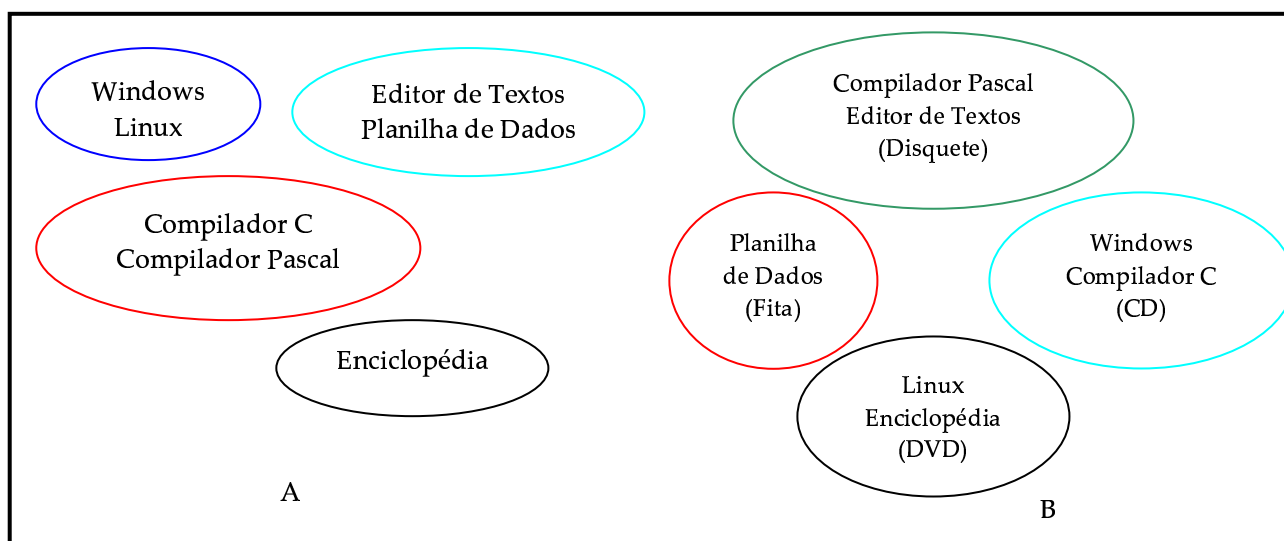


Figura 3.4 - Exemplo de duas possibilidades de clusterização para o mesmo conjunto de exemplos segundo o critério de: finalidade (A) e tipo de mídia (B).

A clusterização é um dos processos mentais mais primitivos do ser humano, desenvolvido para auxiliar a processar grandes quantidades de informações e agrupá-las por atributos comuns [THE 98]. Este agrupamento nos permite inferir que um exemplo particular irá possuir as

mesmas características do cluster do qual ele faz parte mesmo que este exemplo nunca tenha sido “visto” antes (**generalização**).

Usualmente as características dos exemplos são representadas na forma de um vetor de característica de n -dimensões. Para se executar uma tarefa de clusterização normalmente os seguintes passos devem ser seguidos:

1. Selecionar as características;
2. Determinar uma medida de similaridade (ou dissimilaridade);
3. Determinar um critério de clusterização;
4. Escolher um algoritmo de clusterização;
5. Validar os resultados da clusterização;
6. Interpretar os resultados.

Theodoridis [THE 98] define mais formalmente a clusterização como: seja X um conjunto de exemplos tal que,

$$X = \{x_1, x_2, \dots, x_N\} \quad (4.6)$$

x_i são atributos de X .

Define-se como uma m -clusterização de X em \mathfrak{R} , o fracionamento de X em m conjuntos, C_1, \dots, C_m desde que as seguintes condições sejam atendidas:

$$\begin{aligned} C_i &\neq \emptyset, i = 1, \dots, m \\ \bigcup_{i=1}^m C_i &= X \\ C_i \cap C_j &= \emptyset, i \neq j, i, j = 1, \dots, m \end{aligned} \quad (4.7)$$

Podemos classificar os clusters quanto ao grau de pertinência de seus elementos ao cluster em *hard* (ou *crisp*) ou em termos de conjuntos de lógica nebulosa. Na Figura 3.5 –A temos um exemplo de dois clusters do tipo *hard*. Neste tipo de clusterização os elementos são parte de um cluster ou de outro, sua pertinência ao cluster é total. Já a Figura 3.5 – B apresenta três clusters definidos por conjuntos de lógica nebulosa. Cada elemento pode pertencer a um ou mais clusters com diferentes graus de pertinência em cada um deles.

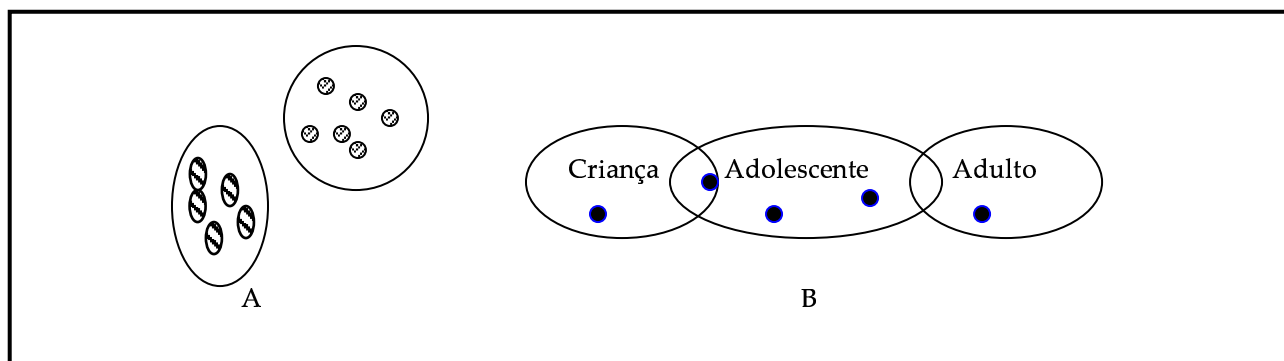


Figura 3.5 – Exemplos de classificação de tipo de clusterização quanto ao grau de pertinência de seus elemento ao cluster. Em A temos *cluster* do tipo *hard* e em B definidos por conjuntos de lógica nebulosa.

Os métodos de clusterização podem ser classificados conforme a estratégia adotada em *métodos baseados em centróides* e *métodos hierárquicos*. Os métodos hierárquicos subdividem-se em métodos de aglomeração e métodos de subdivisão. No método hierárquico por aglomeração inicialmente cada elemento é considerado um cluster. Estes clusters são então combinados, formando um cluster “maior” de acordo com a medida de similaridade adotada. Já no método hierárquico por divisão todos os elementos fazem parte de um só cluster e são subdivididos sucessivamente até que cada elemento seja um cluster. Nos métodos baseados em centróide, cada cluster possui um elemento central que é o protótipo do cluster (o centróide). Todos os elementos que estiverem a uma distância máxima d_i do cluster C_i são atribuídos a este cluster.

3.2.1 Clusterização Utilizando o Algoritmo k-Means

O método de clusterização *k-Means*¹ [THE 98] é baseado em centróide para fragmentar um conjunto de N elementos em K subconjuntos disjuntos S_j que contém N_j elementos, observando o critério de minimização da soma-dos-quadrados da função de distância euclidiana entre os elementos. Cada centróide é a posição média dos elementos daquela classe (cluster). Este método produz clusters do tipo *hard* e seu algoritmo é do tipo voraz. Apresentamos a seguir em pseudocódigo o algoritmo:

1. Determinar a quantidade K desejada de clusters;
2. Inicializar as médias m_1, m_2, \dots, m_k ;
3. Repetir até não haver mais variação nas k -médias (m_1, m_2, \dots, m_k);
 - a. Utilizar as médias estimadas para classificar os exemplos (atribuir um elemento N a um cluster C);
 - b. Faça $i = 1$ até K

¹ <http://lib.stat.cmu.edu/apstat/136>, implementação em Fortran do algoritmo k-Means. Visitado pela última vez em 28/04/2003.

i. Atribuir m_i à média dos exemplos de C_i

Inicialmente deve-se determinar a quantidade de clusters desejada para fragmentar os elementos. As médias são inicializadas aleatoriamente a partir do vetor de elementos a ser clusterizado. A partir destas médias, verificar quais os exemplos que estão mais próximos de cada média e atribuir este ponto a esta classe. Como os elementos atribuídos a cada cluster foram modificados é necessário recalculá-las. Este processo se repete até que não haja mais alteração nos valores das médias.

Nos problemas de análise de imagem, é normal que o vetor de características resultante da fase de extração de características possua uma grande dimensão [RUI 97]. Em sistemas CBIR é comum existir algum tipo de processamento visando reduzir esta dimensionalidade, mas preservando a tendência que aquelas características representam. Uma prática usual é empregar métodos estatísticos para obter momentos da imagem tais como média, desvio padrão e variância. Além destes métodos, técnicas de aprendizado não supervisionado também são empregadas para obter uma redução da dimensão para posterior classificação ou para a própria classificação das características [MIN 96, RUI 97]. Neste trabalho executamos experimentos com clusterização do tipo *k-Means* para reduzir (quantizar) o número de cores numa imagem com o objetivo de reduzir o tamanho do vetor de características, mas mantendo um conjunto de cores que ainda representasse a imagem original (no sentido de suas cores predominantes). No capítulo 5, seção 5.2.1.2 iremos apresentar alguns resultados da utilização desta técnica.

3.3 Resumo do Capítulo

Neste capítulo estabelecemos que o problema de classificação e recuperação de imagens pode ser tratado utilizando-se uma abordagem de ML da IA. Para isto propomos a utilização de dois métodos de aprendizagem, um com aprendizado supervisionado (RNAs) e outro não supervisionado (clusterização).

RNAs MLP são adequadas para tratar o tipo de dado que pode ser extraído de uma imagem (cor e histograma, por exemplo) pela sua natureza numérica (ponderação das entradas) estarem em um domínio contínuo e serem de “baixo nível”. A capacidade de aproximar funções não-lineares das RNAs MLP oferece boas perspectivas para a rede aprender a classificar imagens e, sua capacidade de generalização, de classificar adequadamente imagens não vistas no treinamento da rede (por exemplo, cenas com alteração de iluminação ou mudança de perspectiva).

A arquitetura de RNAs MLP *Cascade Correlation* foi escolhida principalmente porque permite determinar automaticamente a quantidade de neurônios da camada oculta. Esta automatização permite reduzir a complexidade dos experimentos que seriam necessários para determinar uma quantidade adequada de neurônios na camada oculta numa RNA MLP com algoritmos de aprendizagem do tipo *Backpropagation*. Esta redução de complexidade vem a viabilizar a parte experimental desta dissertação.

Apesar de uma RNA com arquitetura MLP *Cascade Correlation* ser um pouco diferente de uma RNA com arquitetura MLP *Backpropagation*, e isto poder ter uma influência nos estudos que serão apresentados no próximo capítulo, optou-se por usar um modelo com as características de performance mais adequadas em relação ao número de experimentos previstos neste trabalho.

A clusterização nos oferece a perspectiva de utilizar um método que praticamente não necessita de intervenção, o que é interessante especialmente na fase de extração de características (quantização de cores).

No próximo capítulo iremos nos aprofundar em como RNAs MLP determinam as superfícies de separação. Iremos, partir da concepção geométrica de como estas superfícies são estabelecidas, verificar a importância de uma fase de extração de características adequada para gerar exemplos e contra-exemplos que permita a RNA executar uma rejeição confiável de um padrão.

4 SUPERFÍCIES DE SEPARAÇÃO EM RNAS MLP

Na tarefa de reconhecimento e recuperação de imagens é interessante que o método empregado recupere imagens com a menor diferença possível entre as características escolhidas da *imagem-exemplo* com as características das *imagens-de-busca* procurando evitar ao máximo falsos reconhecimentos. No contexto da classificação de imagens utilizando RNAs isto significa evitar os falso-positivos (FP). Para que uma RNA atenda a esta necessidade não é suficiente apenas que ela classifique corretamente os padrões apresentados, mas também ela deve ser capaz de **rejeitar** um padrão. Para exemplificar, na Figura 4.1 temos dois padrões, círculos representam o espaço de padrões que se deseja aceitar (*imagens-exemplo* de latas de Coca-Cola, por exemplo) e quadrados (*imagens-de-fundo*) que se quer rejeitar. As linhas tracejadas representam a região (superfície de separação que será explicada em mais detalhes na seção 4.1) ideal e/ou buscada de classificação. No exemplo da Figura 4.1 – A, a linha sólida representa uma superfície de separação determinada por uma RNA fictícia que inclui os dois padrões, portanto, estes dois padrões são classificados por esta RNA como pertencentes a mesma classe (indesejável porque não estamos rejeitando o padrão de quadrados). Já na Figura 4.1 – B, a superfície de separação determinada pela RNA fictícia é adequado porque permite a rejeição dos padrões dos quadrados. A partir de um estudo feito por Gori e Scarselli [GORI 98] sobre superfícies de separação (ou a classificação que a RNA determina com estas superfícies) em RNAs MLP iremos neste capítulo detalhar como isto se aplica a tarefa de reconhecimento e classificação de imagens utilizando RNAs MLP.

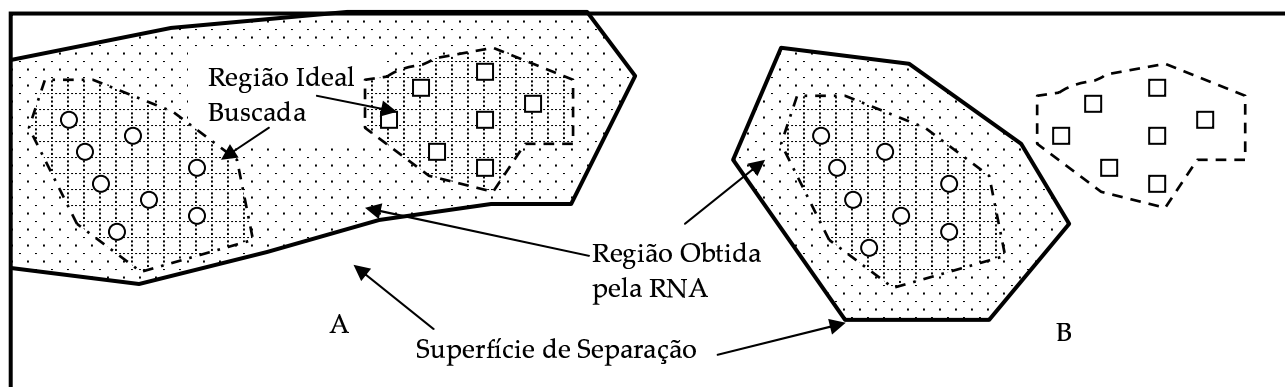


Figura 4.1 – Superfície de separação para aceitar / rejeitar padrões (círculos representam os padrões que devem ser aceitos e quadrados representam os padrões a serem rejeitados).

4.1 Compreensão Geométrica das Superfícies de Separação

Gori e Scarselli [GORI 98] apresentam um estudo sobre a utilização de RNAs do tipo MLP para a tarefa de classificação de padrões baseado na compreensão geométrica das superfícies de separação obtida pelo ajuste dos pesos dos neurônios da rede. Os padrões apresentados a uma RNA possuem um grau variado de pertinência (traduzido sob forma de um valor de ativação obtida na saída da RNA) com uma das classes para as quais a RNA foi treinada para classificar. Quanto maior for o grau de pertinência em relação ao rótulo atribuído a classe, maior será a “certeza” deste padrão pertencer àquela classe. Portanto, quando este grau de pertinência estiver abai-

xo de um determinado valor de limiar (no contexto de RNAs é conhecido como critério de limiar, *thresholding criteria*) o padrão deve ser rejeitado. Por exemplo, seja RN uma RNA e sua saída limitada por uma função de ativação ao intervalo $[0, 1]$. Seja “1” o rótulo atribuído à classe C e um valor de limiar L estabelecido em 0,8. Um padrão será aceito como pertencente à classe C quando a saída da rede RN for maior ou igual do que L .

Os autores apresentam uma análise de que, ao contrário dos bons resultados que vários trabalhos na área de classificação e validação de padrões utilizando RNAs MLP relatam, RNAs do tipo MLP não são adequadas para serem utilizadas como classificadores quando os padrões apresentados à RNA não se enquadram perfeitamente nas classes pré-definidas, portanto realizando uma rejeição confiável do padrão. Para ocorrer uma rejeição confiável de um padrão é necessário que as superfícies de separação sejam **fechadas** em torno do padrão (ver Figura 4.1 – B). A utilização do critério de limiar também é criticada pelos autores do artigo que, segundo eles, não é adequada para a verificação de padrões. Ainda segundo Gori, em redes MLP com uma estrutura piramidal, nas quais a quantidade de neurônios da camada oculta é menor do que a quantidade de neurônios de entrada, **nunca** irão ocorrer superfícies fechadas de classificação. Caso a quantidade de neurônios da camada oculta seja maior do que a quantidade de neurônios de entrada, não há garantia de ocorrer superfícies fechadas de classificação, e determinar esta condição é um problema da classe *NP-Hard*. Os autores do artigo sugerem que os casos de sucesso no uso de RNAs do tipo MLP para a tarefa de classificação se devem ao pré-processamento aplicado aos dados e a natureza destes.

Para exemplificar a proposta de Gori e Scarselli, treinou-se uma RNA do tipo MLP com a topologia 2-5-1 (duas entradas, cinco neurônios na camada oculta e um neurônio de saída) utilizando o algoritmo de aprendizagem *backprop*. Foram utilizadas duas classes para os padrões de aprendizado que foram denominadas de *Nuvem0* e *Nuvem1*. Para *Nuvem0* o rótulo atribuído foi “0” e para *Nuvem1* “1”. Os padrões foram gerados aleatoriamente com uma distribuição normal em relação a um ponto central. O objetivo é verificar se aumentado à quantidade de padrões *Nuvem0* **em torno** dos padrões *Nuvem1* consegue-se que a RNA determine superfícies de separação fechadas em torno dos padrões *Nuvem1*. Nas próximas seções iremos detalhar o que são e como são visualizadas e interpretadas as superfícies de separação.

4.1.1 Superfícies Abertas de Separação (SAS)

A partir da Eq. (4.1) podemos obter as equações dos hiperplanos n -dimensionais que determinam as superfícies de classificação. No caso do exemplo a seguir a superfície é uma reta porque a rede em questão possui somente duas entradas, o que permite que seu espaço de padrões de entrada possa ser representado em 2D. Da Eq. (4.1) obtém-se:

$$\text{Saída} = \sum_{i=1}^n \omega X_i + \text{Bias} \quad (4.8)$$

Onde n é quantidade de atributos de entrada (que vem a ser a quantidade de neurônios de entrada), ω_i é o peso sináptico do neurônio i e X_i é o atributo (valor) de entrada no neurônio i . Expandindo o somatório teremos uma equação, similar a equação da reta ($Ax + By + C = 0$), como segue:

$$X_1\omega_1 + X_2\omega_2 + Bias = 0 \quad (4.9)$$

Que pode ser reescrito como:

$$X_2 = \frac{-\omega_1 X_1 - Bias}{\omega_2} \quad (4.10)$$

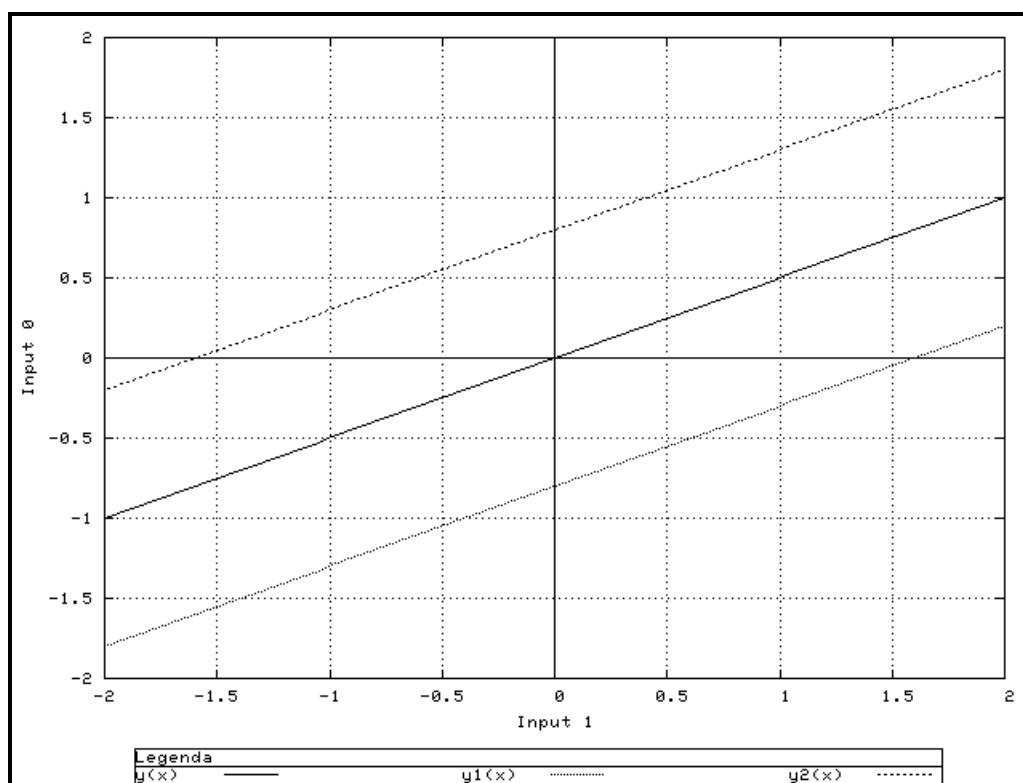


Figura 4.2 – Plano de separação determinado pelos pesos de um neurônio e a influência do *Bias*.

Tabela 4.1 – Tabela-resumo de decisão do rótulo com a condição de saída vs. limiar.

Rótulo	Estado do Neurônio de Saída	Condição
Classe	Neurônio Ativado	Saída > Limiar
Não Classe	Neurônio Não Ativado	Saída < Limiar
-----	Hiperplano ou Reta de Separação	Saída = 0

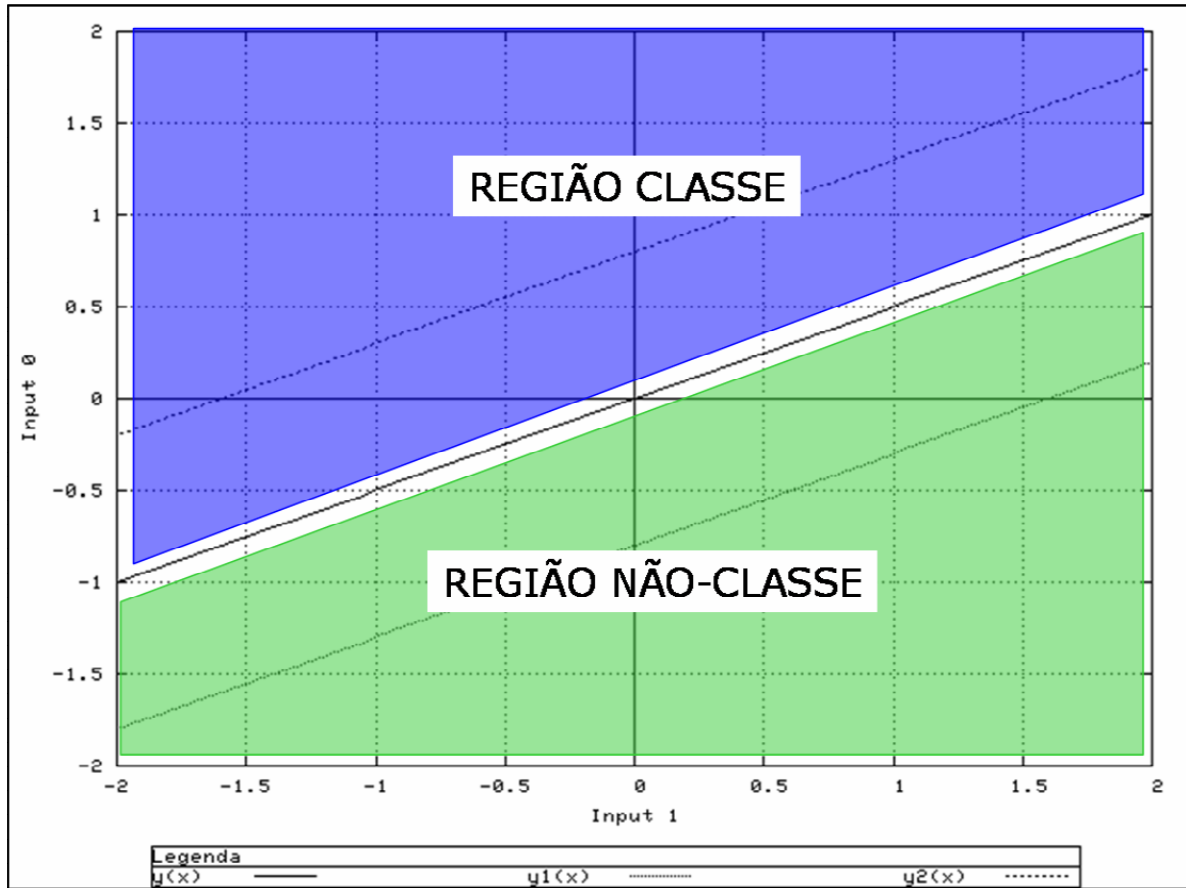


Figura 4.3 – Planos de separação de Classe e Não Classe.

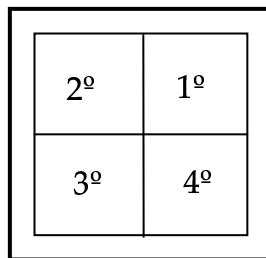


Figura 4.4 – Disposição dos quadrantes adotada nos exemplos deste capítulo, origem x, y no centro.

Portanto a Eq. (4.10) nos permite para um dado valor de w_1 , w_2 , $Bias$ obtidos no aprendizado visualizar como a rede responde dados valores de X_1 e X_2 (entradas) as retas de separação para a RNA deste exemplo. Na Figura 4.2 podemos verificar que a equação $y(x) = 0,5 * x + 0$ (e.g. $w_1 = -0,5$, $w_2 = 1$ e $Bias = 0$) divide a área dos atributos de entrada ($Input\ 0$ e $Input\ 1$) em dois semiplanos que denominaremos de *classe* e *não-classe*. Cada semiplano estará associado a uma classe que a RNA irá identificar (ver Figura 4.3). As retas $y1(x) = 0,5 * x - 0,8$ e $y2(x) = 0,5 * x + 0,8$ demonstram o deslocamento que o *bias* determina nos hiperplanos. A saída do neurônio será ativada quando $y(x) > 0$, portanto o neurônio identificou uma *classe* (o padrão de entrada se encontra no

semiplano da *classe*). Se $y(x) < 0$ então a saída do neurônio não foi ativada e o padrão de entrada está situado no semiplano *não-classe*. A Tabela 4.1 apresenta um resumo destas condições.

O programa *NeuSim* [OSO 99], que simula o aprendizado de RNAs com *backprop*, foi utilizado para treinar as RNAs utilizadas nestes exemplos e disponibiliza na forma de um arquivo os pesos e os bias dos neurônios. Nos exemplos a seguir as retas plotadas, $y_0(x)$ a $y_4(x)$, representam as superfícies de separação determinadas pelos neurônios da camada oculta, sendo obtidas utilizando a Eq. (4.10) e o arquivo de pesos do *NeuSim* após treino com as bases de aprendizado criadas a partir da *Nuvem0* e *Nuvem1*.

Na Figura 4.5 (Caso 1) temos os padrões de treinamento compostos por 6 exemplos de *Nuvem0* e 16 exemplos de *Nuvem1*. Nota-se que todos os padrões *Nuvem0* estão situados no 3º e no 4º quadrante (conforme convenção adotada na Figura 4.4). Esta distribuição específica dos exemplos (tanto de *Nuvem0* como de *Nuvem1*) permitiu que a RNA ajustasse as retas de forma a dividir a superfície de separação em duas regiões. Esta divisão fica evidente na Figura 4.6. Foram gerados padrões de entrada no intervalo $[-10, 10]$ com passo de uma unidade para *Input0* e *Input1*. Somente os padrões que produziram uma ativação na saída da RNA com valor maior ou igual a 0,5 (limiar de 0,5) foram plotados. Ou seja, nesta figura estão representados os padrões que foram classificados pela RNA como pertencentes à classe *Nuvem1*. Pode-se verificar claramente que este ajuste obtido pelas superfícies de separação não determina uma superfície fechada de separação (SFS) em torno dos padrões *Nuvem1*. A consequência prática desta situação é que padrões que estão afastados do aglomerado *Nuvem1* (e que podem pertencer à classe *Nuvem0*) seriam sempre classificados como pertencentes à classe *Nuvem1*, caso se situassem nesta região acima da superfície de separação definida.

No Caso 2, Figura 4.7, tem-se outro exemplo de uma superfície aberta de separação (SAS). É interessante notar a ausência de padrões *Nuvem0* no 1º quadrante do gráfico o que faz com que a RNA não possa determinar um plano de separação neste quadrante por falta de (**contra**) exemplos. A Figura 4.8 (Caso 2) apresenta sobreposto aos padrões de treinamento e às retas, os padrões de entrada para os quais a RNA treinada teve a sua saída ativada (padrões da classe *Nuvem1*), utilizando valor de limiar de 0,2. Com este de valor de limiar somente os padrões de entrada que produziram na saída da RNA um valor igual ou superior a 0,2 foram plotados e estes encontram-se na região (aberta) delimitada pelas retas $y(0)$ e $y(1)$. Na Figura 4.9 foi utilizado um valor de limiar de 0,8 e comparando-se com o valor de limiar de 0,2 podemos notar que os exemplos plotados estão todos “acima” da região delimitada pelas retas $y(0)$ e $y(1)$. Podemos verificar nestes exemplos que a RNA obteve FP.

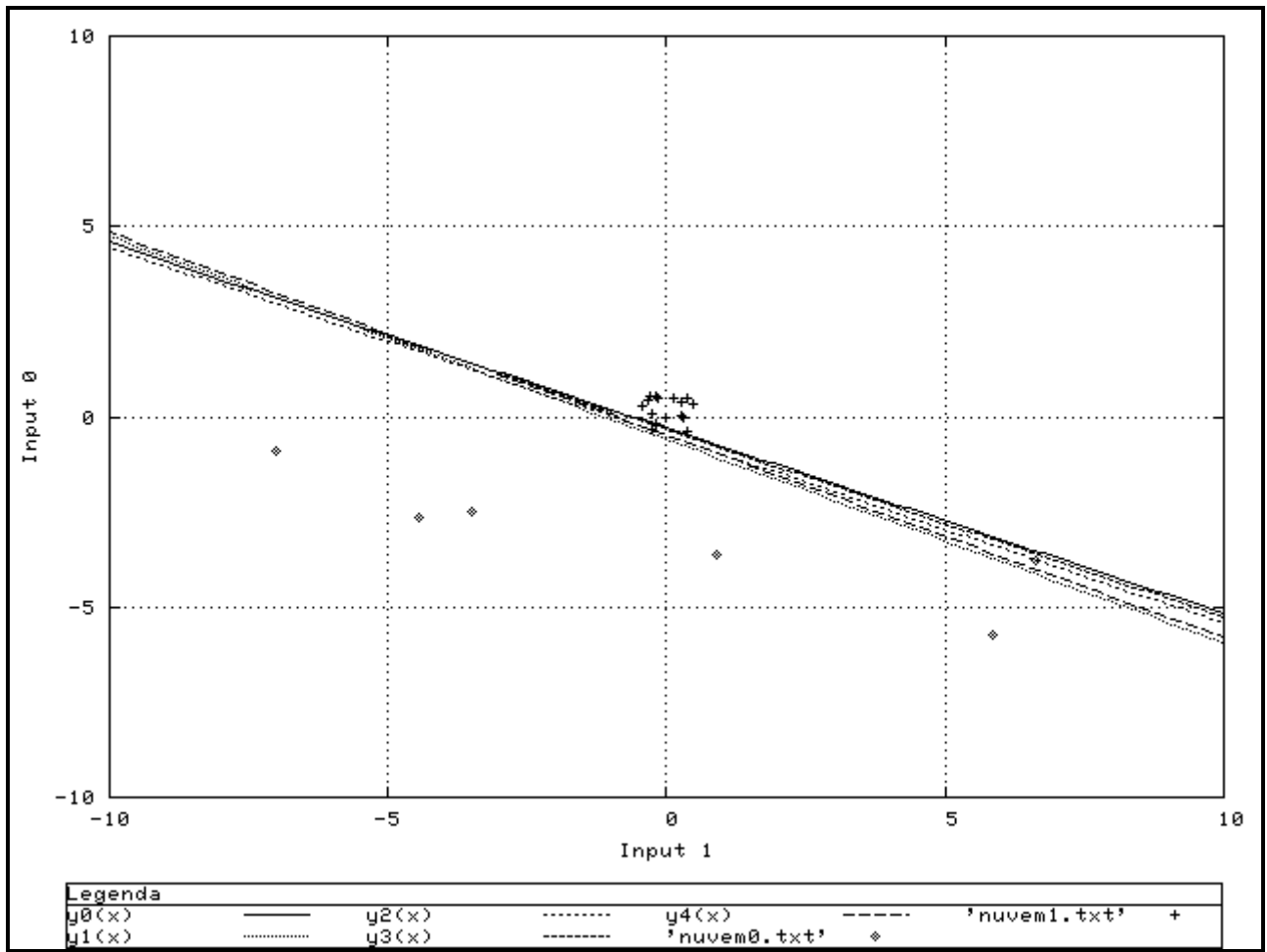


Figura 4.5 – Caso 1 SAS, 6 exemplos *Nuvem0* e 16 exemplos *Nuvem1*.

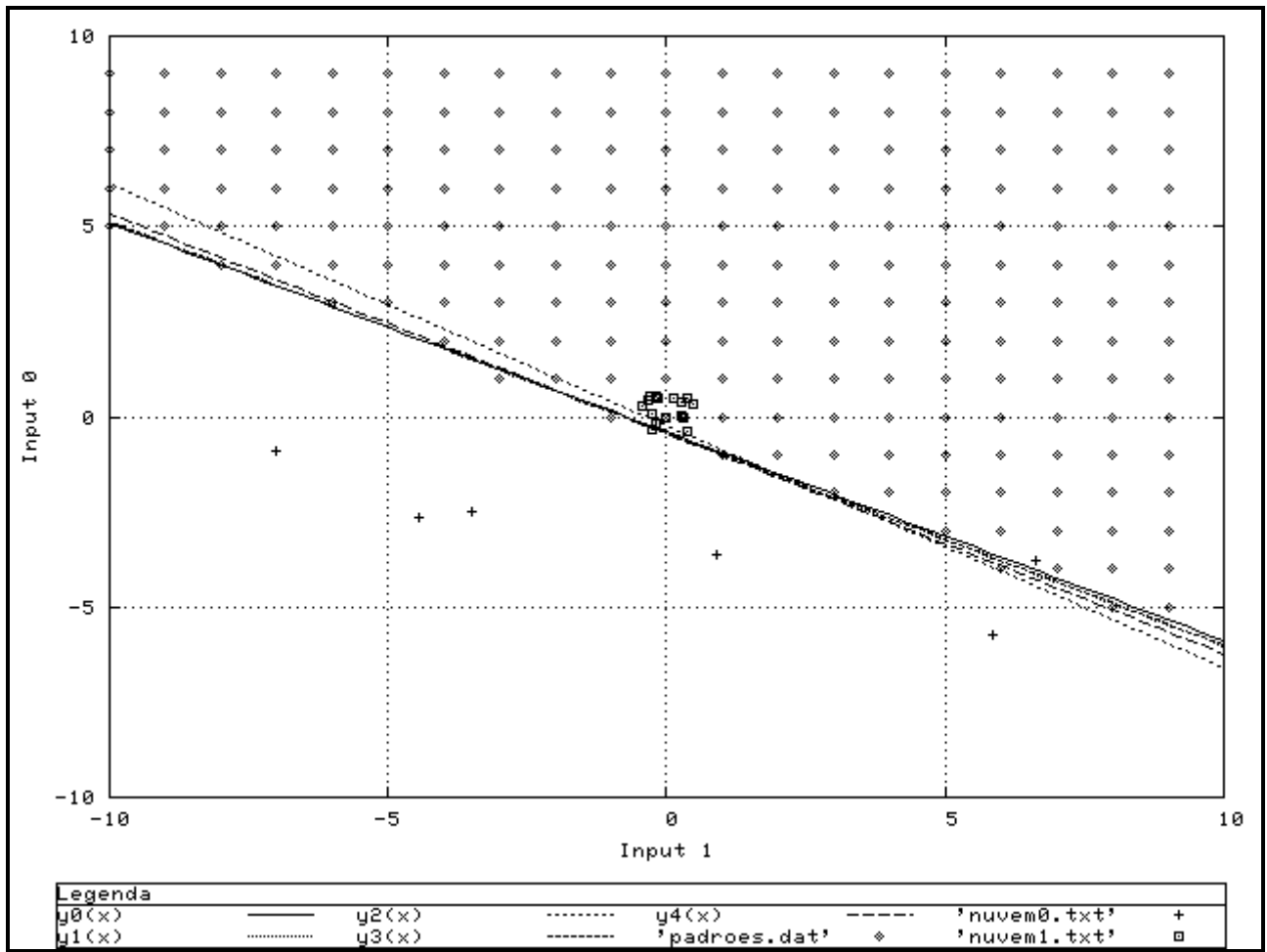


Figura 4.6 – Caso 1 SAS, 6 exemplos *Nuvem0* e 16 exemplos *Nuvem1* e com visualização de padrões classificados como *Nuvem1*.

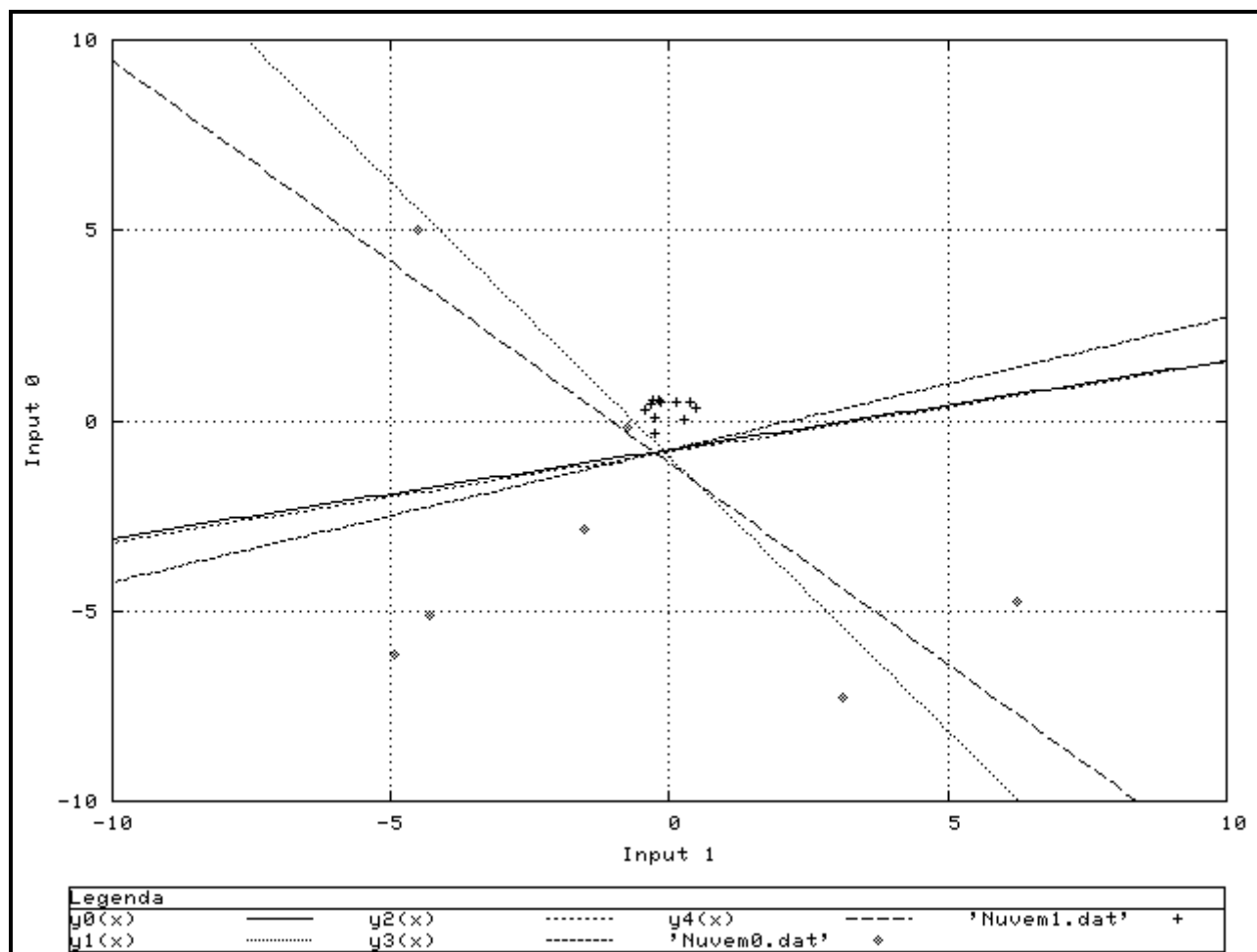


Figura 4.7 – Caso 2 SAS, 7 exemplos de *Nuvem0* e 11 exemplos de *Nuvem1*.

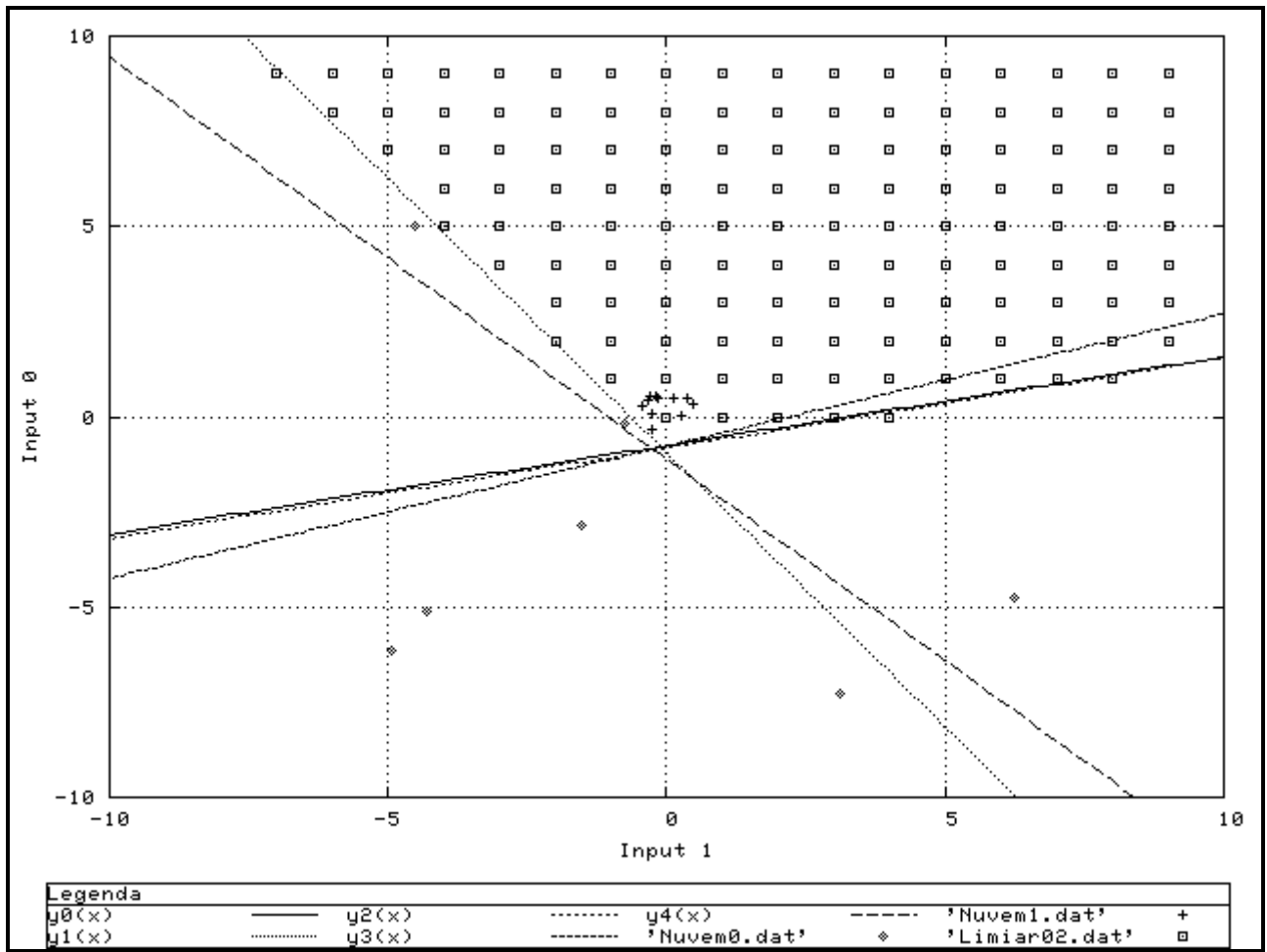


Figura 4.8 – Caso 2 SAS, 7 exemplos de *Nuvem0* e 11 exemplos de *Nuvem1* e critério de limiar de 0,2.

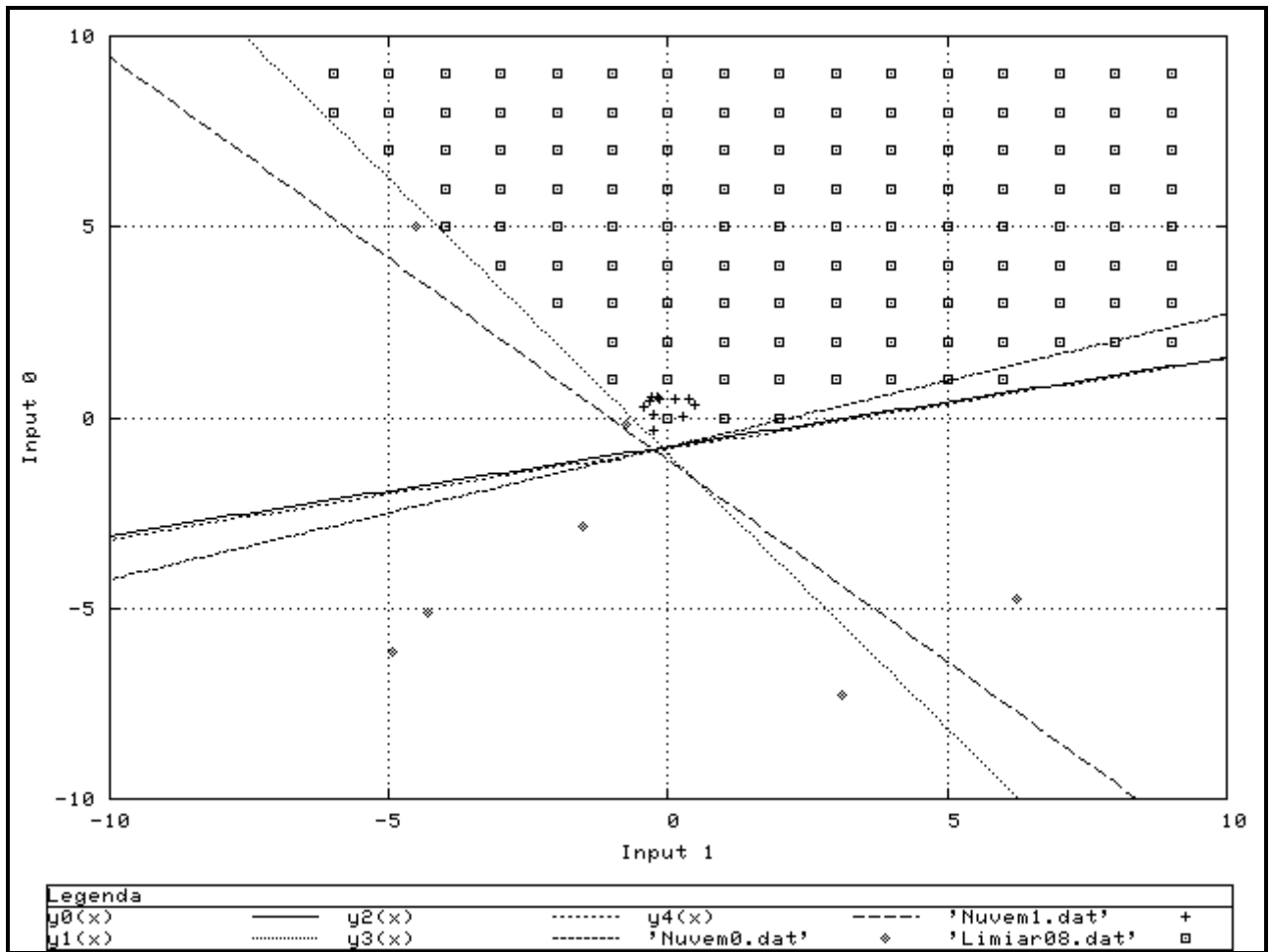


Figura 4.9 – Caso 2 SAS, 7 exemplos de *Nuvem0*, 11 exemplos de *Nuvem1* e critério de limiar de 0,8.

Com estes exemplos encerramos a seção de SAS onde mostramos como o ajuste de pesos obtidos por uma RNA MLP a partir de seus exemplos de treinamento determina superfícies de separação que podem resultar em FP durante uma tarefa de classificação. Na próxima seção iremos abordar as superfícies fechadas de separação utilizando estas mesmas RNAs, mas acrescentando contra-exemplos para tentar fechar as superfícies de classificação.

4.1.2 Superfícies Fechadas de Separação (SFS)

De acordo com Gori e Scarselli, quando a quantidade de neurônios na camada oculta é igual ou maior do que a quantidade de neurônios de entrada, a RNA **pode vir** a determinar SFS, mas determinar esta condição é um problema de complexidade computacional *NP-Hard*. Mas os autores também demonstram que em determinadas condições, quando são apresentados padrões de treinamento mais apropriados, a RNA consegue determinar pesos que resultam numa superfície fechada de separação. Continuando com os exemplos da seção anterior, vamos demons-

trar que aquela mesma RNA utilizada anteriormente pode conseguir determinar SFS em torno da classe *Nuvem1*.

No Caso 1, Figura 4.10, adicionou-se aleatoriamente novos exemplos de *Nuvem0* que acabaram completando principalmente o 1º e 2º quadrantes. Estes exemplos permitiram que a RNA ajustasse seus pesos de modo que se determinassem superfícies de separação (quase) fechadas em torno do padrão *Nuvem1*. Como se percebe na Figura 4.11 estes padrões adicionados ainda não foram suficientes para determinar superfícies de separação suficientemente fechadas em torno do padrão *Nuvem1*. No contexto deste trabalho de classificação de imagens, vamos assumir que os padrões *Nuvem1* representam a classe de imagem que se deseja reconhecer. Toda a região interna de um perímetro imaginário em torno destes padrões representa os exemplos possíveis (mas que não fizeram parte do treinamento da rede) que se deseja que a RNA classifique como pertencentes a esta classe. Vamos assumir também que toda a região externa a este perímetro imaginário em torno dos padrões *Nuvem1* são padrões de *imagens-de-fundo* e que queremos rejeitar. Se esta rede fosse utilizada para classificar imagens, as *imagens-de-fundo* que estivessem nas regiões A, B, C (ver Figura 4.11) seriam classificadas como pertencentes a classe de imagens que desejamos reconhecer, portanto teríamos FP nestas regiões e rejeições de *imagens-de-fundo* apenas nas demais regiões. Ressalta-se que a dimensão do vetor de características para a tarefa de classificação de imagens é elevado (ver seção 2.3.3) na ordem de dezenas ou até mesmo centenas. Isto resulta numa explosão dimensional de entradas da RNA que torna a tarefa de determinar SFS mais complexa para a mesma.

A Figura 4.12, do Caso 2, é um exemplo de uma SFS obtida pela adição de *contra-exemplos* a partir de uma RNA que anteriormente não determinava superfícies fechadas de separação (ver Figura 4.7, Figura 4.8 e Figura 4.9). Comparando-se a Figura 4.13 com a Figura 4.14 percebe-se claramente a influência do valor de limiar. O valor de limiar de 0,8 (Figura 4.14) exclui aqueles padrões que possuem um grau de pertinência pequeno com a classe *Nuvem1*. Somente os padrões que estão dentro do polígono determinados por **todas** as retas são classificados como pertencentes à classe *Nuvem1*. Este exemplo representa um caso ótimo para a tarefa de classificação de imagens utilizando RNAs MLP, pois as superfícies de separação permitem uma classificação confiável e uma rejeição confiável de padrões.

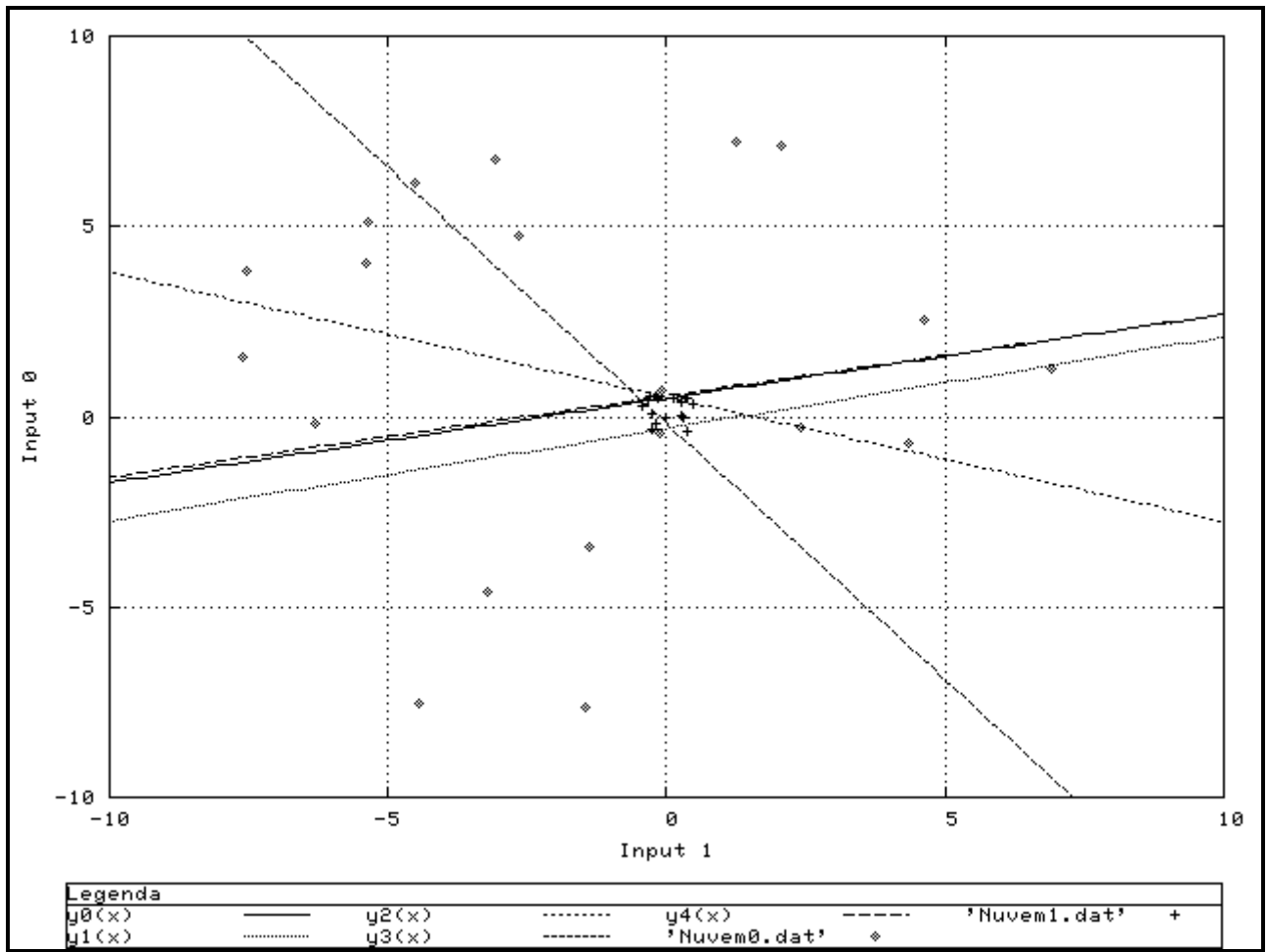


Figura 4.10 – Caso 1 SFS, 20 exemplos da classe *Nuven0* e 16 exemplos da classe *Nuven1*.

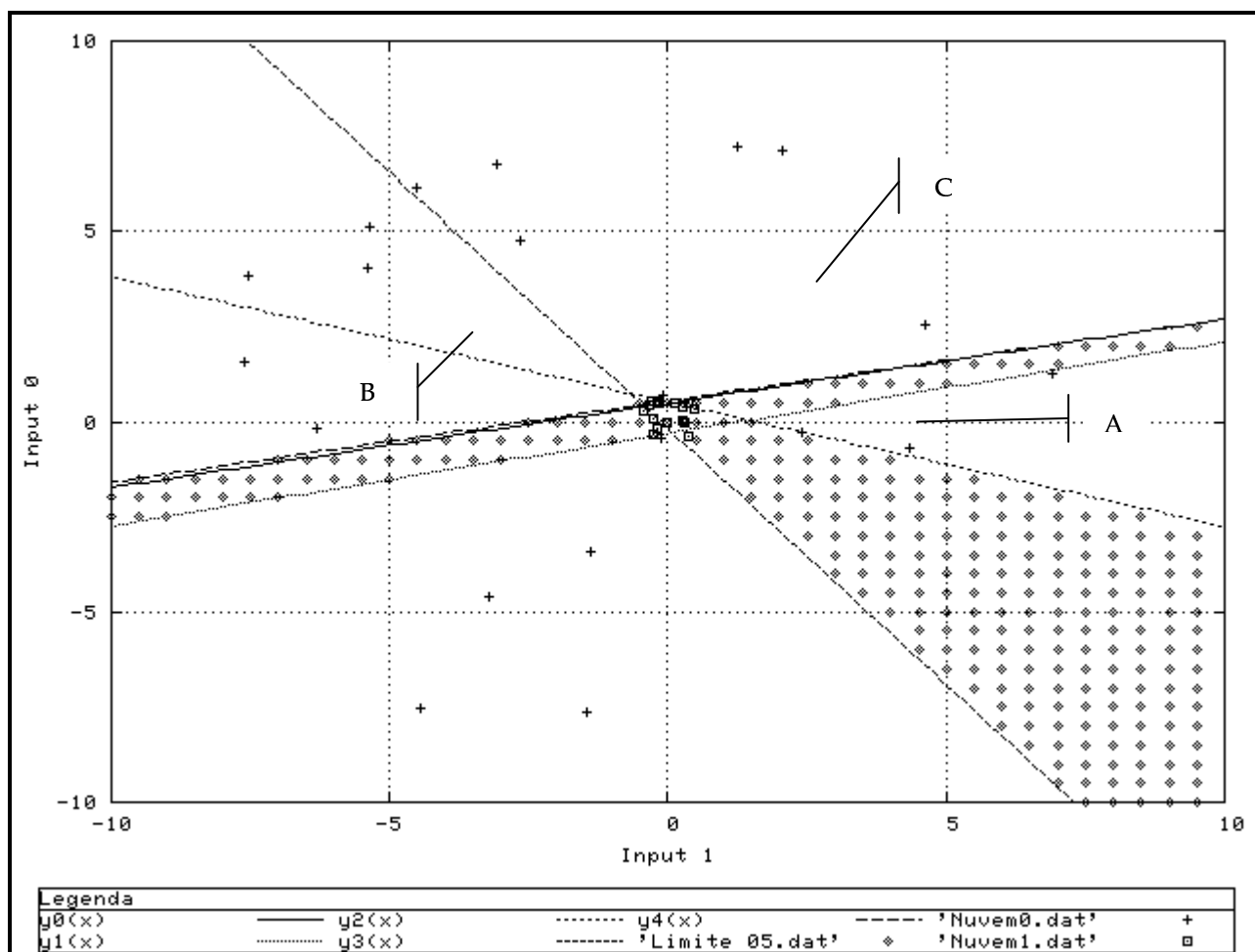


Figura 4.11 – Caso 1 SFS, 20 exemplos da classe *Nuvem0* e 16 exemplos da classe *Nuvem1* e limiar de 0,5.

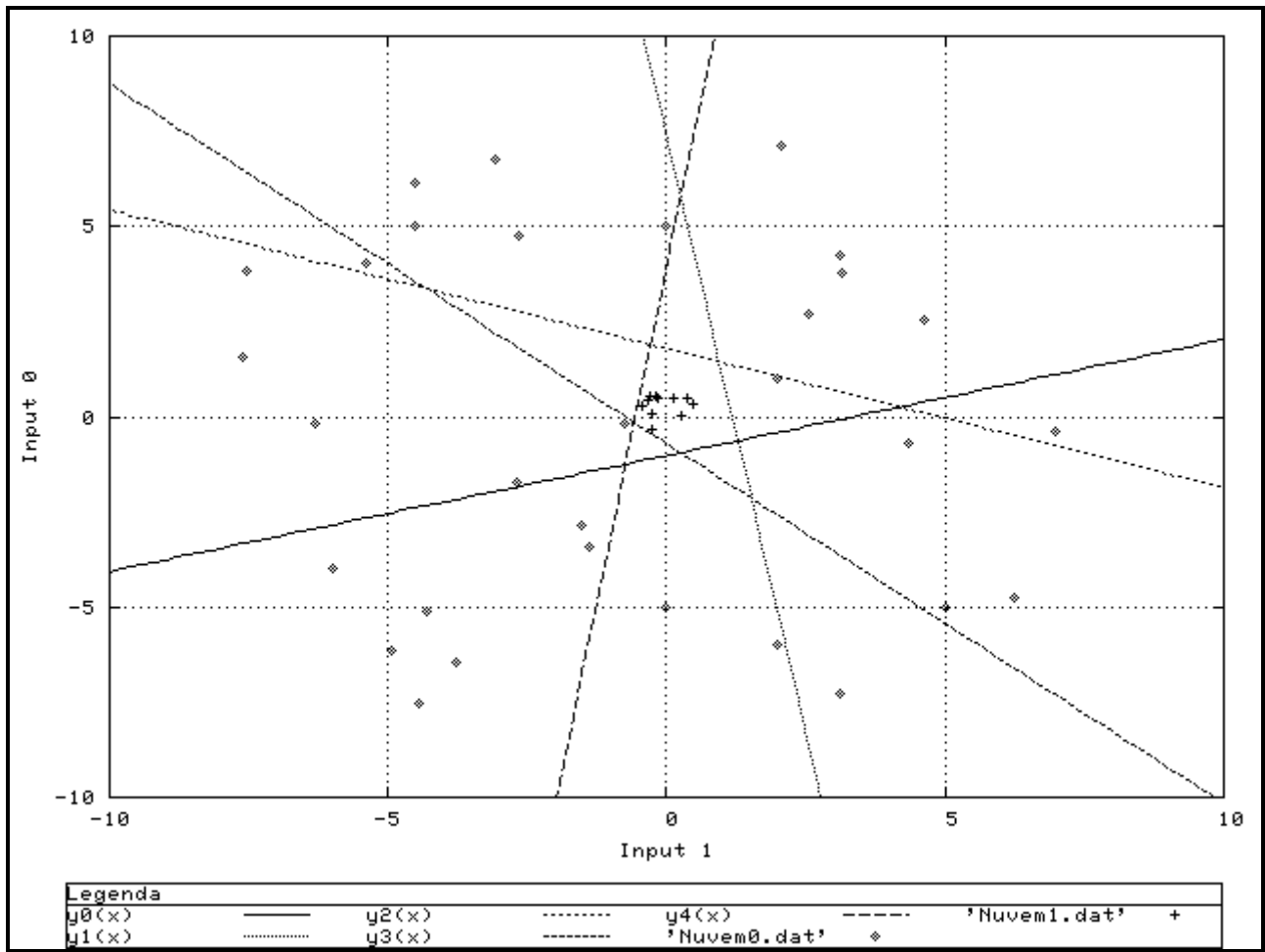


Figura 4.12 – Caso 2 SFS, 31 exemplos da classe *Nuven0* e 11 exemplos da classe *Nuven1*.

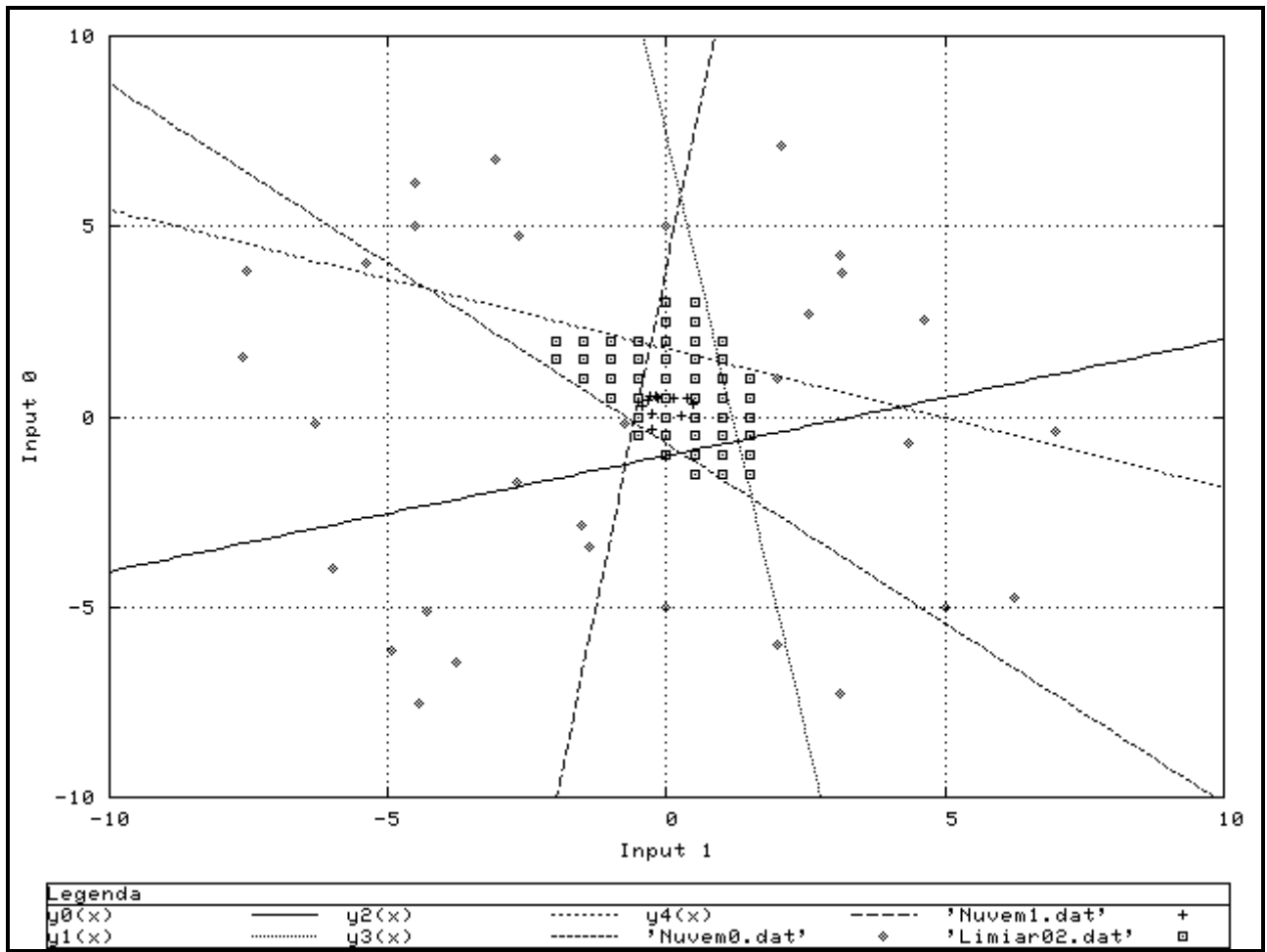


Figura 4.13 – Caso 2 SFS, 31 exemplos da classe *Nuvem0*, 11 exemplos da classe *Nuvem1* e limiar de 0,2.

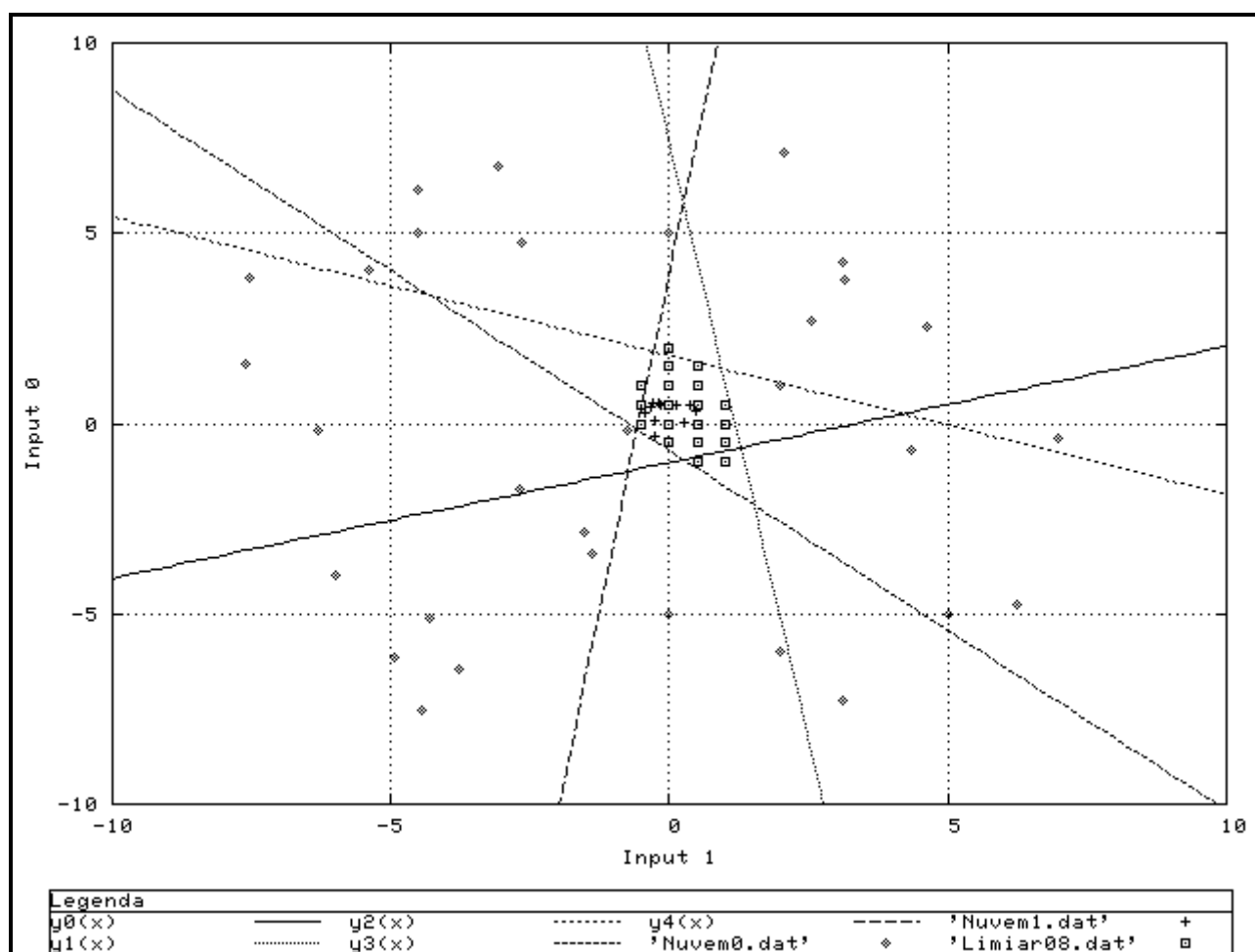


Figura 4.14 – Caso 2 SFS, 31 exemplos da classe *Nuvem0*, 11 exemplos da classe *Nuvem1* e limiar de 0,8.

Observando a influência do limiar na Figura 4.13 e na Figura 4.14 podemos notar que um valor pequeno de limiar tem como resultado uma classificação “mais relaxada”, abrangendo uma região maior de padrões. Um valor maior de limiar causa uma classificação mais restrita. No caso de um valor de limiar pequeno é razoável esperar-se que isto ocasione uma maior ocorrência de FP enquanto que um valor muito alto de limiar, um aumento na ocorrência de falso-negativos (FN) durante a classificação. A minimização de FP através do aumento do valor de limiar ocasiona um aumento de ocorrência de FN. Portanto tem-se aqui um problema de otimização multi-objetivo, onde a melhoria de um objetivo implica na degradação de outro. Os valores de limiar que satisfazem este problema de otimização estão em um conjunto de Pareto [COS 02] que vem a ser a fronteira (Figura 4.15) que separa o espaço de soluções possíveis das não possíveis.

Procuramos demonstrar nesta seção o significado geométrico de SFS de classificação e de que maneira elas influenciam na tarefa de classificação de imagens. Também mostramos que apesar das afirmações feitas por Gori e Scarselli, obter uma SFS é factível e que a utilização de um bom critério bem ajustado de limiar pode produzir resultados adequados e que este ajuste pode

ser relacionado com um problema de otimização multi-objetivo. Na próxima seção iremos apresentar duas alternativas para tentar fechar adequadamente as superfícies de classificação.

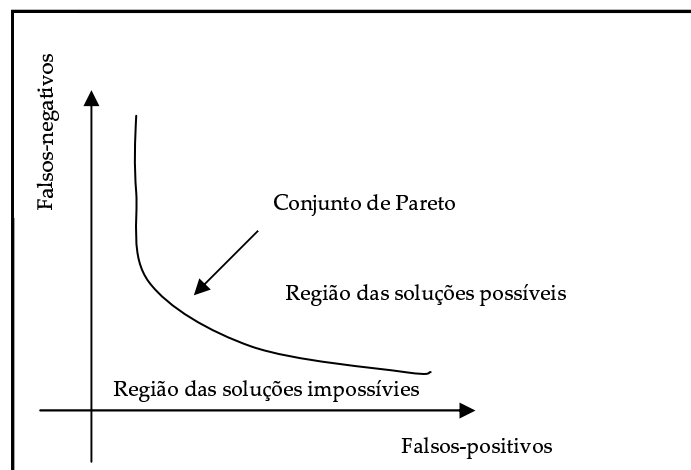


Figura 4.15 – Exemplo de conjunto de Pareto para o problema do valor de limiar.

4.2 Abordagem para Fechamento de Superfícies de Separação

Nas seções anteriores verificamos que com exemplos adequados de treinamento e um bem ajustado critério de limiar pode-se obter SFS ou, pelo menos, superfícies quase fechadas que podem ser adequadas para determinada aplicação. Com base nestas observações propomos duas abordagens para gerar SFS adequadas para a tarefa de classificação de imagens utilizando RNAs MLP.

A primeira abordagem é denominada de “Regiões de Imagens Falso-Positivo” (RIFP). Após o primeiro treinamento da RNA executa-se uma rodada de classificação na BDI. Selecionam-se algumas regiões de imagens da BDI que resultaram em FP. Estes FP são rotulados como exemplos de *não-classe* (que queremos rejeitar) e são adicionados aos padrões de treinamento. Executa-se então um novo treinamento da RNA e uma nova rodada de classificação. O processo é repetido até que a acurácia seja considerada satisfatória para esta BDI. Este vem a ser um método iterativo de refinamentos sucessivos para eliminar FP. O sistema Ikona emprega uma abordagem similar de refinamento. Após a determinação dos métodos de extração e comparação de características, executam-se simulações de recuperação na base preparada. O usuário pode então executar um refinamento especificando nas imagens recuperadas, imagens como sendo exemplos **positivos** ou **negativos** que o sistema irá utilizar para melhor refinar as pesquisas futuras. O ponto negativo desta abordagem é a possibilidade de obter uma especialização demasiada e o excesso de intervenção humana. Modificações no conteúdo da base de imagens tornariam necessária uma nova simulação de recuperação e talvez um novo refinamento. Um outro ponto negativo que poderia ser apontado é o tempo necessário para executar novos treinamentos da RNA. Este argumento é válido para RNAs MLP que utilizam *backprop* ou *quickprop*. No nosso caso utilizamos uma RNA MLP com algoritmo de aprendizagem *Cascade-Correlation* que permite um aprendizado **incremental** [OSO 99], ou seja, todo o “conhecimento” que a RNA possuía antes será reaproveitado (pois os pesos da

rede estão congelados) e será necessário treinar a RNA apenas para os novos exemplos o que resulta em um tempo de treinamento significativamente inferior.

A segunda abordagem, denominada de “**Nuvem de Não-Classe**” (NNC), visa aumentar a automatização do processo de geração de contra-exemplos. Para isso geramos uma “nuvem de pontos” aleatoriamente utilizando uma distribuição uniforme no espaço de atributos (cores) a exemplo da abordagem utilizada na seção 4.1.2 Esta nuvem irá compor junto com padrões de *não-classe* obtidos de regiões de imagens, os padrões de contra-exemplos (*imagens-de-fundo*) apresentados a RNA. Executa-se o treinamento da RNA seguido de uma rodada de testes numa BDI. Repete-se o processo de adição de NNC, treinamento e recuperação até que a recuperação seja considerada satisfatória. Este processo pode ser classificado como um método de força bruta, mas é factível para a tarefa de recuperação de imagens e como estamos utilizando RNA MLP *Cascade-Correlation* continuamos aproveitando sua vantagem de aprendizado incremental.

Determinar a quantidade de novos exemplos a serem adicionados é uma questão importante, pois isto reduziria a quantidade de iterações. Como este problema ainda não está completamente resolvido, trabalhos futuros estão sendo delineados na direção de como determinar automaticamente a quantidade ideal de padrões na NNC. Uma possibilidade consiste em estimar a fronteira (no espaço de atributos) dos exemplos que se deseja classificar e gerar contra-exemplos um pouco afastados desta fronteira evitando introduzir ruído no espaço dos exemplos. Estes contra-exemplos somente precisariam ser gerados em torno desta fronteira porque estes é que irão determinar os hiperplanos de separação. Na Figura 4.16 – A tem-se o exemplo de uma classe e sua fronteira e na Figura 4.16 – B os padrões NNC gerados em torno desta fronteira sem invadir o interior da classe.

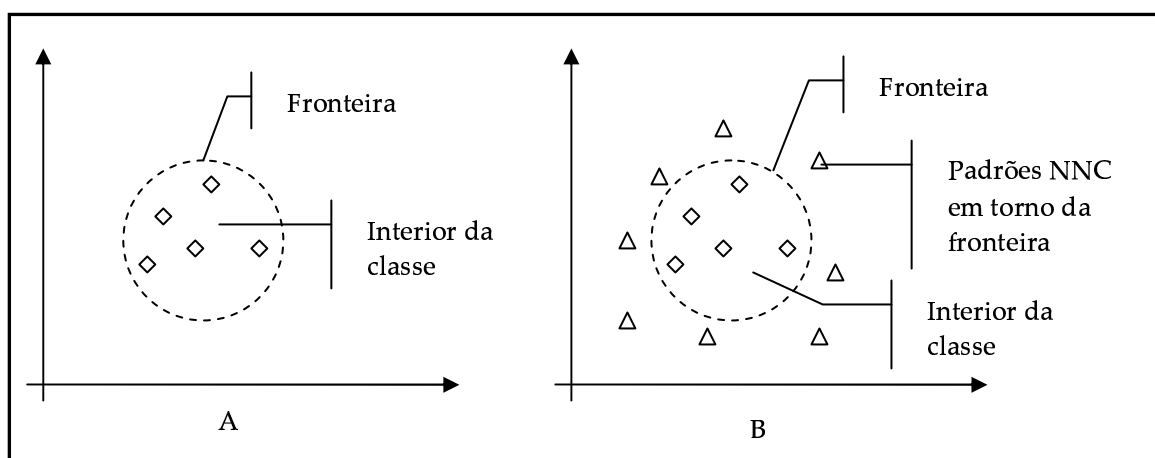


Figura 4.16 – Exemplo de aperfeiçoamento futuro para abordagem NNC.

4.3 Outras Considerações Sobre o Trabalho de Gori e Scarselli

O estudo que os autores executaram foi basicamente feito utilizando RNAs MLP com uma camada oculta, função de ativação do tipo sigmóide monotônica e algoritmo de apren-

dizagem *backprop*. No trabalho de Gori e Scarselli são apresentados três teoremas os quais iremos comentar brevemente.

Teorema 1: Este teorema trata da influência que a quantidade de neurônios na camada oculta possui sobre a possibilidade de a RNA conseguir um ajuste de pesos de forma a determinar uma SFS. Caso esta quantidade seja inferior a quantidade de neurônios de entrada a RNA não irá conseguir determinar uma SFS para as classes.

Se a quantidade de neurônios na camada oculta for igual à quantidade de neurônios de entrada, então o tipo de função de ativação adotado possui influência na capacidade de se obter uma SFS. Se a função de ativação for monotônica então não será possível à RNA determinar uma SFS, independente do ajuste de pesos.

No caso da quantidade ser maior, existe a possibilidade de se obter uma SFS que depende dos ajustes dos pesos sinápticos que a RNA irá conseguir com os padrões de treinamento.

Teorema 2: Nenhum critério de rejeição baseado somente no valor de limiar pode “transformar” uma SAS em uma SFS, mesmo que a RNA possua mais de uma camada oculta ou mais de um neurônio de saída (classificação de várias classes numa mesma RNA).

Teorema 3: A capacidade de uma RNA em determinar SFS não depende do tipo de função de ativação sigmóide adotada para os neurônios da camada oculta. Esta capacidade é principalmente determinada pela arquitetura adotada e pelos valores dos pesos sinápticos.

Um ponto que não é tratado no trabalho de Gori e Scarselli é o significado dos limites inferior e superior dos atributos que compõem os padrões de entrada tem nas superfícies de separação. Por exemplo, a representação numérica de uma cor no sistema RGB está compreendida no intervalo $[0, 255]$. Como não existem valores fora deste intervalo no sistema RGB podemos assumir que estes limites determinam um hiperplano de separação no limite inferior e superior para o atributo. Portanto, mesmo com apenas um hiperplano é possível determinar uma SFS (Figura 4.17) independentemente da quantidade de neurônios da camada oculta ser inferior aos de entrada. Esta SFS provavelmente não será uma superfície que irá minimizar a ocorrência de FP, mas é uma superfície fechada.

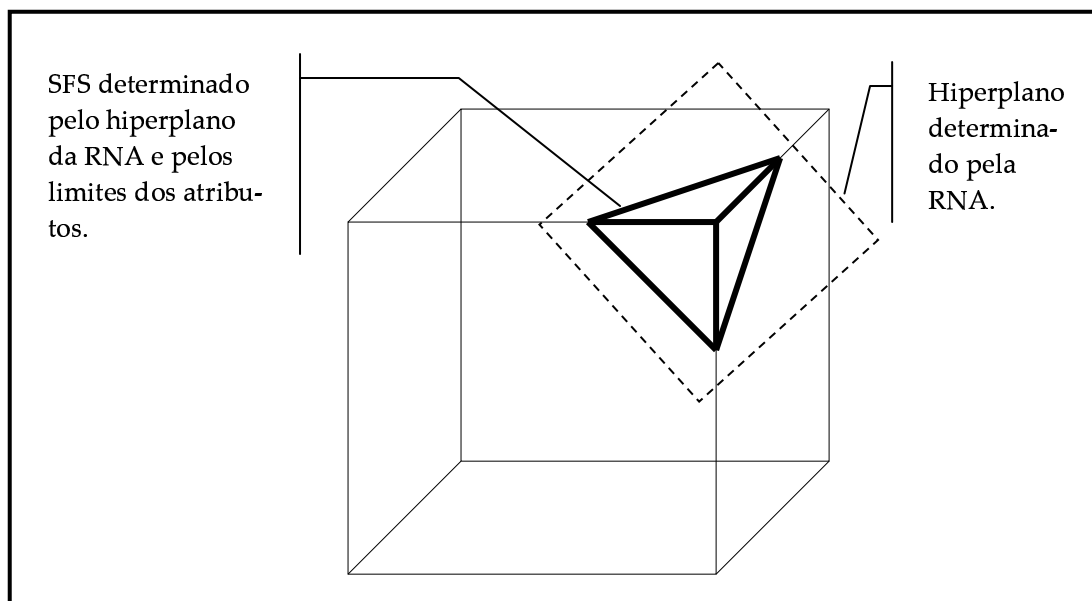


Figura 4.17 – SFS no cubo RGB determinada por um hiperplano e pelos limites dos atributos.

Outro ponto que se deve considerar é que uma RNA MLP *Cascade-Correlation* não é uma MLP “tradicional”. Cada um de seus neurônios da(s) camada(s) oculta(s) possui ligação com os neurônios de entrada (atalhos) e com os outros neurônios das outras camadas ocultas. Portanto faz-se necessário um estudo mais aprofundado se o trabalho de Gori e Scarselli aplica-se totalmente neste tipo de RNA MLP.

Os autores sugerem que entre as alternativas para o problema das superfícies de separação na verificação de padrões utilizando RNAs pode-se utilizar RNAs de Função de Base Radial (*Radial Basis Function*, RBF) [HAY 01] que determinariam SFS. Porém, RNAs do tipo RBF possuem algumas características que não as tornam uma solução superior à RNA MLP no contexto deste trabalho. As RNA RBF executam um particionamento (clusterização) do espaço dos padrões de entrada através de hiperelipsóides, construindo aproximadores locais. A quantidade de partições em uma RNA RBF é determinada pela quantidade de neurônios da camada oculta. O tamanho do raio dos hiperelipsóides está diretamente relacionado à quantidade de partições. Se for determinado no projeto da rede várias partições (Figura 4.19 – B), é possível que o tamanho do raio seja pequeno e o particionamento não enquadre todos os exemplos ou a ocorrência de uma especialização nos padrões apresentados à RNA durante o treinamento (*overfitting*, maior ocorrência de FN). Caso sejam determinadas poucas partições a tendência é que o raio seja maior e enquadre regiões que não são adequadas, portanto existindo ainda a possibilidade de ocorrência de FP (Figura 4.19 – A). Estes comportamentos são determinados pelo tipo de algoritmo de aprendizagem utilizado no projeto da RNA RBF e através de ajustes de parâmetros de aprendizado.

Existem situações de classificações de padrões que são problemáticas para uma RNA RBF executar (Figura 4.18). Nestes exemplos o problema está na região central que não possui padrões e na dificuldade de se criar um hiperelipsóide com um “furo” no meio.

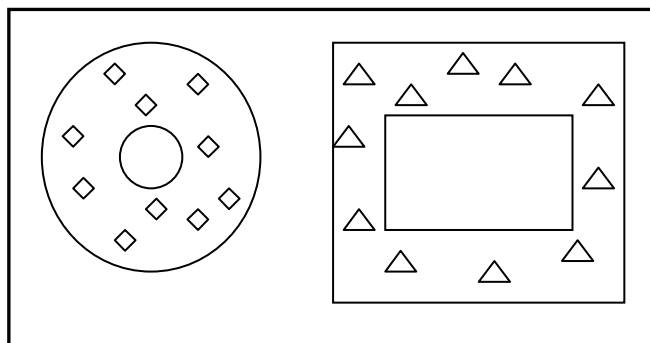


Figura 4.18 – Exemplo de situações em que uma RNA RBF não classifica adequadamente.

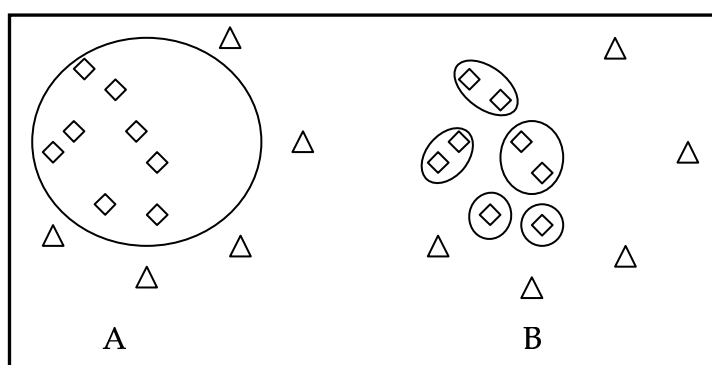


Figura 4.19 – Exemplo de duas possibilidades de classificação em uma RNA RBF que dependem do algoritmo de aprendizagem. Em A temos maior ocorrência de FP e em B maior ocorrência de FN.

4.4 Resumo do Capítulo

Neste capítulo apresentamos uma explicação do porquê em alguns casos os resultados obtidos utilizando RNAs MLP na tarefa de classificação e/ou verificação de padrões podem não ser satisfatórios mesmo que no treinamento da RNA a taxa de generalização tenha sido considerada boa. Esta explicação está baseada na compreensão geométrica das superfícies de separação que os pesos sinápticos ajustados pela RNA durante o treinamento determinam no espaço n -dimensional dos padrões. Quando estes pesos determinam uma SFS, a RNA é capaz de rejeitar padrões adequadamente, ou seja, minimizando a ocorrência de FP ou até mesmo eliminando-os. A possibilidade de ocorrer uma SFS para RNAs MLP é basicamente determinada pela arquitetura adotada. No caso de RNAs MLP com algoritmo *backprop* este estudo mostra alternativas de como estimar a quantidade de neurônios necessária na camada oculta, que, conforme comentamos na seção 3.1.2, ainda é um problema com poucos estudos feitos até o momento.

Com a utilização de um pré-processamento adequado nos sinais de entrada que irão gerar os padrões de treinamento é possível que uma RNA MLP venha a determinar SFS. Isto reforça a importância de um estudo e experimentação de técnicas de pré-processamento a serem desenvolvidos pelo responsável pela preparação dos padrões de treinamento quando se utiliza RNAs MLP. Deve-se considerar uma preparação cuidadosa nos padrões, tanto de exemplos quanto de

contra-exemplos, visto que os contra-exemplos podem efetivamente contribuir para a obtenção de SFS. Para isto apresentamos duas alternativas para geração de contra-exemplos que julgamos adequadas para a tarefa de classificação de imagens com RNAs MLP.

No próximo capítulo iremos apresentar o programa que implementa as técnicas de IA para classificação e recuperação de imagens que foram apresentados até agora e os resultados que obtivemos.

5 CLASSIFICAÇÃO E RECUPERAÇÃO DE IMAGENS UTILIZANDO RNAS MLP CASCADE-CORRELATION

Para a aplicação das técnicas de classificação e recuperação de imagens apresentadas neste trabalho utilizando técnicas de IA foi desenvolvido o programa *Image Matching* (IMatch). O IMatch foi desenvolvido em Visual C++ versão 6.0/MFC para a plataforma Windows. Para algumas tarefas específicas foram utilizados códigos-fonte e bibliotecas disponíveis em domínio público. Para a importação e manipulação (acesso à memória que contém os *pixels*) das imagens no IMatch foi utilizada a biblioteca *FreeImage*¹ desenvolvida por *Floris van den Berg* e que está disponível para utilização e modificação através da licença *GNU General Public License*. Para clusterização com método *k-Means* utilizou-se a implementação (em formato binário) disponível na *dynamic link library* (DLL) *mva.dll* que compõem o software de computação estatística R^2 em sua distribuição para a plataforma Windows. O gerador de números de aleatórios utilizado na inicialização das médias no *k-Means* e geração da NNC foi à biblioteca de classes em linguagem C++ *NewRan v.2.0* desenvolvida por *Robert Davis*³. O código-fonte para o algoritmo de redução de cores *Median Cut* foi desenvolvido por *Anton Kruger*⁴ em linguagem C e foi modificado para permitir que seja possível utilizar outro nível de quantização diferente de 256. O programa utilizado para simular o aprendizado de uma RNA MLP *Cascade-Correlation* foi o *NeuSim* desenvolvido por Osório [OSO 99]. Por fim, o *SimNeu* [OSO 99] é um simulador de RNAs que usa uma rede previamente treinada pelo *NeuSim* e realiza a ativação da rede a fim de classificar os padrões que lhe são apresentados e que pode ser integrado a outros *softwares*.

O IMatch permite a escolha de um entre três métodos de pré-processamento a ser aplicado nas imagens para gerar o vetor de características. Na descrição destes métodos, n é o nível de quantização de cor adotado. O primeiro tipo, denominado T1, é um vetor formado apenas pelo HC após a imagem ter sido quantizada pelo método *Median Cut* e o tamanho deste vetor é n . No segundo método, T2, o vetor de características é formado apenas pelas n triplas de componentes das cores após a imagem ter sido quantizada pelo método *k-Means*. O tamanho do vetor do método T2 é $n*3$. O método T3 gera um vetor de características composto pelo HC (histograma de cor) e com as triplas de componentes das cores da imagem quantizada com *Median Cut*. O tamanho do vetor deste método é $n*4$. Para todos os três métodos é possível utilizar o sistema de cor RGB ou HSV.

A interface do IMatch (Figura 5.1) é composta por quatro vistas: em (a) configuram-se os parâmetros de sistema de cor, método de redução de cor e valor de quantização, tamanho da *imagem-de-treinamento*, passo da janela de varredura e quantidade de entradas e saídas da RNA; na vista (b) visualiza-se a *imagem-de-busca* e as regiões da *imagem-de-busca* identificadas que contém a

¹ <http://freeimage.sourceforge.net/>, data da última visita em 14/05/2003.

² <http://cran.r-project.org/>, data da última visita em 14/05/2003.

³ <http://www.robertnz.net/download.html>, data da última visita em 14/05/2003.

⁴ <http://www.ddj.com/ftp/1994/1994.09/color.zip>, data da última visita em 14/05/2003.

imagem-exemplo; a vista em (c) apresenta uma pré-visualização da *imagem-de-busca* e também permite a inspeção de uma região da *imagem-de-busca* com as cores reduzidas com o algoritmo *Median Cut*; a vista (d) apresenta mensagens diversas do sistema.

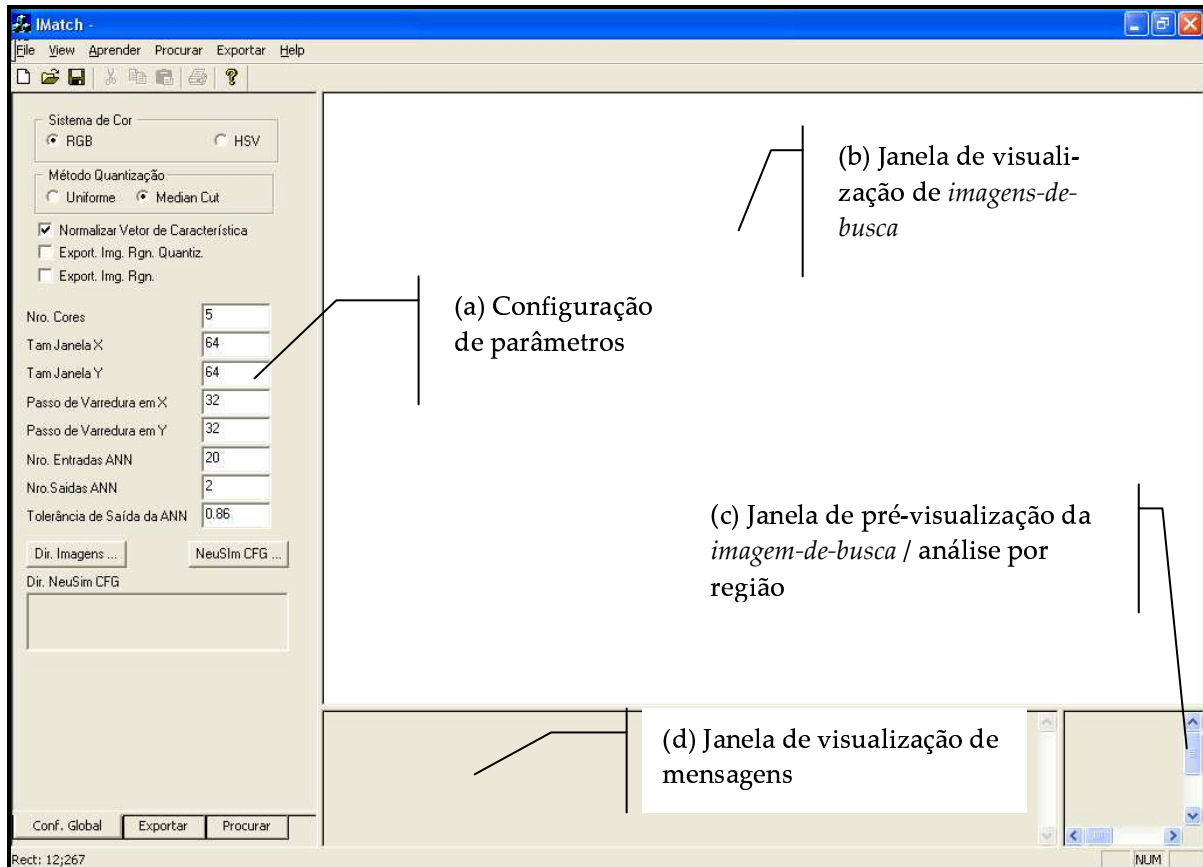


Figura 5.1 – Interface principal do IMatch.

A seguir listaremos as funcionalidades implementadas no IMatch:

- Composição do vetor de características:
 - Somente histograma de cor em RGB ou HSV;
 - Somente clusters de cores (k-Means) em RGB ou HSV;
 - Vetor de características composto por histograma de cores e cores quantizadas através de *Median Cut* em RGB ou HSV;
- Método de ordenação do vetor de características:
 - Por distância euclidiana da cor em RGB a partir da origem do cubo RGB;
 - Permutação das n-cores quantizadas;
 - Ordenação por valor do histograma de cores;

- Visualização do vetor de características (nuvem de pontos de cores) em VRML:
 - Nuvem de pontos de cores de várias imagens em um mesmo VRML;
 - Nuvem de pontos de cores no cubo no RGB ou no hexacone HSV;
 - Nuvem de pontos de cores com os clusters (k-Means) no cubo RGB ou no hexacone HSV;
- Conversão do sistema de cor RGB \Leftrightarrow HSV;
- Quantização de cores por método *Median Cut* ou uniforme;
- Nuvem de Não-Classe:
 - Gerar Nuvem de Não-Classe por distribuição uniforme de cores no espaço de cores;
 - Gerar Nuvem de Não-Classe por distribuição uniforme de cores com saturação de histograma;
- Regiões de Imagens Falso-Positivo:
 - Captura e salvamento automático de região FP durante reconhecimento;
 - Captura por coordena de imagem;
- Geração de exemplos (janelamento) de *imagens-de-treinamento* a partir de uma imagem;
- Visualização das regiões FP, FN e C após reconhecimento;
- Fase de reconhecimento e classificação de imagens pode ser interativa ou em lotes;
- Contagem automática de regiões e imagens recuperadas do tipo FP, FN e C.

5.1 Metodologia

Foram preparadas três BDI com *imagens-de-treinamento* e *imagens-de-fundo* para executar o treinamento da RNA. Sendo que 2/3 das imagens são para aprendizado e 1/3 para teste de generalização. Todas as *imagens-de-treinamento* e *imagens-de-fundo* possuem dimensão de 64 x 64 pixels, resolução de cores de 24 bpp (16 milhões de cores) e formato de arquivo PNG (Portable Network Graphics).

A preparação das *imagens-de-treinamento* e *imagens-de-fundo* (Figura 5.2) utilizadas para o treinamento da RNA é feita a partir de várias regiões de imagens que contém o objeto a ser aprendido. Destas imagens se extraem regiões de dimensões iguais (no caso 64 x 64 pixels). O processo é automatizado para extrair regiões deslocadas definidas pelo passo de varredura em X e Y a partir da origem da imagem como se fosse uma janela que varre toda a imagem extraindo estas regiões.

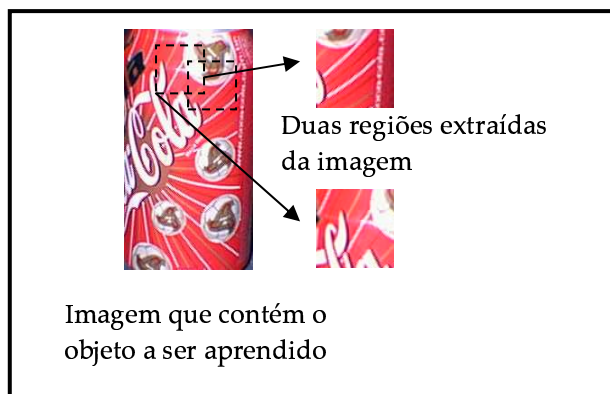


Figura 5.2 – Exemplo da preparação das imagens utilizadas no treinamento da RNA.

Para o treinamento das RNAs foram executadas vinte e cinco simulações de aprendizagem da RNA utilizando como critério de parada 100% de acerto nos exemplos de aprendizado. Destas vinte e cinco simulações aquela que obteve o melhor índice de generalização é treinada novamente (mantendo a semente de inicialização desta simulação) com critério de parada na melhor época. Esta RNA é que será utilizada para a classificação de imagens.

Definimos genericamente de *Classe* a um padrão que causou a ativação da RNA (tipicamente imagens similares a *imagens-de-treinamento*) e o rótulo associado é “1 0”. *Não-Classe* são imagens classificadas pela RNA como *imagens-de-fundo* e seu rótulo é “0 1”.

O IMatch “procura” por *imagens-de-treinamento* aprendidas pela RNA utilizando uma “janela” que percorre a *imagem-de-busca* a partir de sua origem até o fim no sentido da esquerda para direita e de cima para baixo. Esta janela possui a mesma dimensão dos padrões utilizados para treinar a RNA e o seu passo de deslocamento em X e Y é ajustável de forma que seja possível obter sobreposição com a janela anterior. O conteúdo pré-processado pela aplicação de um dos três métodos descritos (T1, T2 ou T3) de cada uma destas janelas é que irá compor os padrões de entradas apresentados à RNA.

A verificação de *performance* para a redução de FP **por imagem** nos experimentos é feita através da contagem de regiões classificadas como *Classe* (C), regiões FP e de FN. Para a automatização da avaliação dos experimentos baseada em um critério consistente (e não baseado na interpretação do usuário), para determinar se a região de uma janela de procura classificada pela RNA é FP, FN ou C, se implementou no IMatch uma ferramenta que permite anotar nas imagens uma ou mais regiões retangulares que contém o padrão C e salvar estas coordenadas em um arquivo associado à imagem. Estas regiões são utilizadas durante o reconhecimento para automatizar a contagem da ocorrência de FP, FN e C. Quando o conteúdo de uma região da imagem causa a ativação da RNA para do tipo C, se verifica a intersecção deste retângulo com os retângulos armazenados. Se a área do retângulo resultante da intersecção for maior ou igual a um valor configurado pelo usuário esta região será contada como sendo tipo C. Caso contrário será contada como um FP. Conta-se como FN quando uma região é classificada pela RNA como sendo *Não-Classe* (NC) ou a saída é desconhecida (i.e. os valores da saída da RNA não configuram nenhum rótulo após a aplicação do critério de limiar), e esta região possui intersecção com alguns dos retângulos

anotados na imagem. Devido à sobreposição da janela de procura, regiões C, FP e FN podem possuir mais de uma contagem por padrão (Figura 5.3).

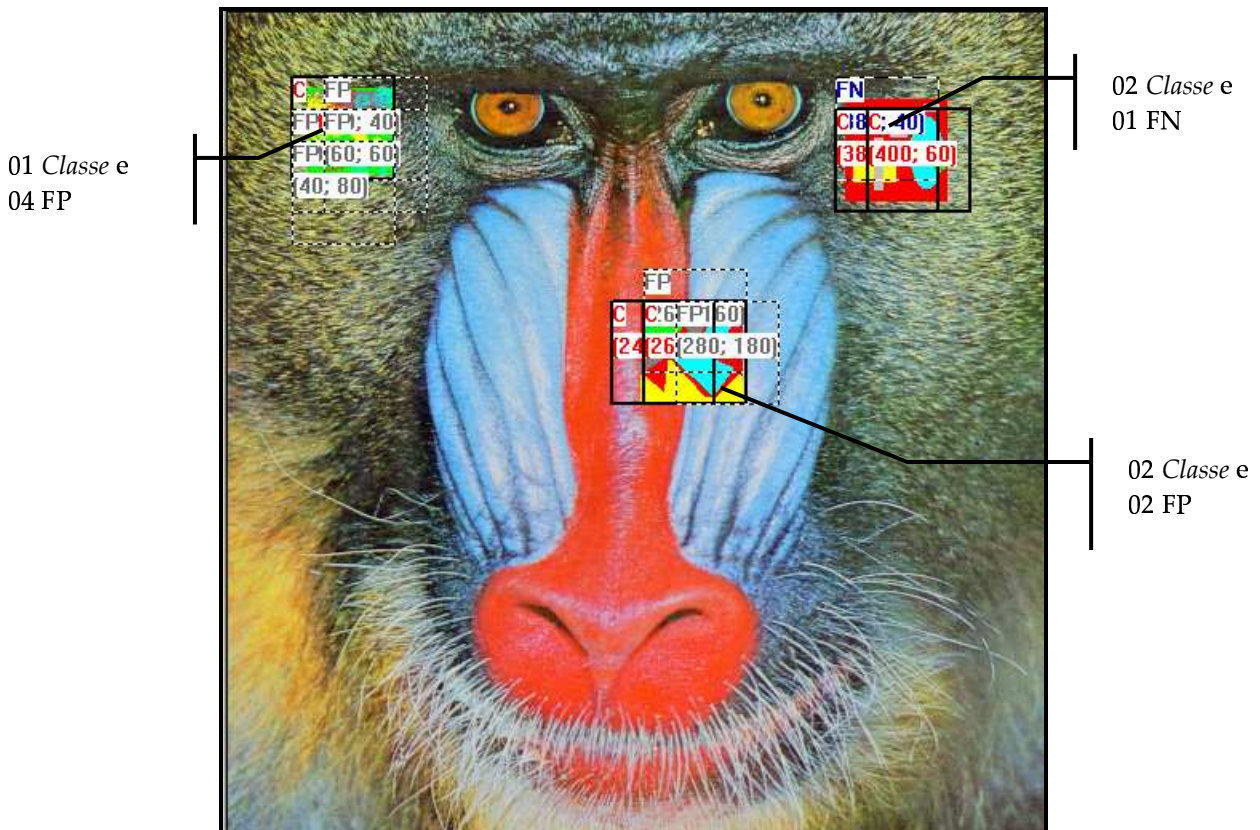


Figura 5.3 – Exemplo de contagem múltipla para regiões de C e FP e FN devido à sobreposição da janela de procura. Tem-se a contagem total: 05 Classe (C), 06 falso-positivo (FP) e 01 falso-negativos (FN).

A verificação de *performance* de **recuperação** de imagens é feita através dos seguintes critérios:

1. Ocorrendo **pelo menos uma** região da imagem classificada como **C** a imagem é contabilizada como recuperada do tipo C (portanto que possuiu o padrão procurado). Neste caso não se contabilizam regiões de FP e/ou FN que possam ter ocorrido na imagem;
2. Ocorrendo **pelo menos uma** região FP e **nenhuma** C a imagem é contabilizada como imagem recuperada do tipo FP. A ocorrência de regiões FN na imagem não é contabilizada neste caso;
3. Ocorrendo **pelo menos uma** região FN e **nenhuma** C e/ou FP a imagem é contabilizada como sendo não recuperada do tipo NC. Esta imagem possuía o padrão procurado, mas a RNA não o reconheceu.

5.2 Experimentos

Os experimentos com os métodos T1 e T2 somente foram executados para a BDI B0 em RGB, ambos com redução de cor para cinco cores sem a aplicação das abordagens RIFP e NNC. Apresentamos estes dois métodos com o intuito de demonstrar a evolução das técnicas utilizadas neste trabalho, que começou utilizando-se apenas o HC, em seguida, somente os índices das cores após a redução de cores com clusterização *k-Means* e, por fim, a combinação do HC com o índice das cores após a redução de cores com *Median Cut*. Na Tabela 5.1 apresentamos um resumo dos experimentos que serão apresentados neste capítulo. A utilização dos métodos T1 e T2 foi descartada com base nos resultados apresentados a seguir e em outros experimentos anteriores que não são apresentados neste trabalho. Verificou-se que estes métodos apresentaram uma performance inferior ao método T3.

Tabela 5.1 – Resumo dos experimentos a serem apresentados.

Método	BDI	Descrição do Método	Quantização de Cores	Abordagem	Sistema de Cor
T1	B0	Histograma de Cores	5	Treino Inicial	RGB
T2	B0	k-Means	5	Treino Inicial	RGB
T3	B0	Median Cut e Histograma de Cores	5	Treino Inicial / RIFP / NNC	RGB / HSV
T3	B1	Median Cut e Histograma de Cores	5	Treino Inicial / RIFP / NNC	RGB
T3	B2	Median Cut e Histograma de Cores	5	Treino Inicial / RIFP / NNC	RGB

As tabelas de médias e desvios padrões, como por exemplo a Tabela 5.3, apresentam a média e o desvio padrão do número de épocas de aprendizagem, da melhor época de generalização e da taxa de generalização da RNA para vinte e cinco simulações da RNA, exceto para BDI B0 na qual foram executadas cinco simulações. Estes valores são apresentados para o treino inicial da RNA e, para o método T3, para as duas iterações de RIFP e duas iterações de NNC.

Nas tabelas de treinamento da RNA (por exemplo,

A coluna *generalização* da tabela acima (e tabelas similares) refere-se a taxa de generalização, ou seja, o percentual de acertos na classificação dos exemplos da base de teste.

Tabela 5.4) apresentamos a melhor simulação da RNA (considerando a taxa de generalização como critério) sendo os pesos desta a que foi utilizada no IMatch para o reconhecimento.

Nas tabelas de totalização, FP(0), FN(0) e C(0) é a contagem de FP, FN e C para o treinamento inicial da RNA, i.e., onde não foram empregadas as abordagens RIFP e NNC para fe-

chamento de superfícies. RIFP $FP(it)$, RIFP $FN(it)$ e RIFP $C(it)$ são as contagens de FP, FN e C para a abordagem RIFP após it iterações do método. NNC $FP(it)$, NNC $FN(it)$ e NNC $C(it)$ são as contagens de FP, FN e C para a abordagem NNC após it iterações do método. Para cada imagem é indicada a quantidade mínima de padrões que existem na imagem, como indicado entre parênteses ao lado de cada imagem.

5.2.1 Seleção do Método de Extração de Características (BDI B0)

A BDI *B0* é uma base onde foram geradas de forma artificial as *imagens-de-treinamento* em um software de edição de gráficos com padrões geométricos simples tais como: diagonais, elipses e quadrados com **cinco cores** somente. Esta é uma BDI “controlada” para verificar a adequação da abordagem proposta utilizando RNAs. Seu objetivo é determinar qual o método de extração de características que permite uma maior redução na ocorrência de regiões de imagens FP. Também se pretende verificar a capacidade de reconhecimento de padrões que possuem diferentes níveis de rotação, translação e, em condição mais restrita, a escala e sobreposição parcial dos padrões.

Para os experimentos utilizaram-se 42 exemplos de C (Fig. Anexo A5) e 42 exemplos de NC (Fig. Anexo A6) para compor a base de aprendizagem e 21 exemplos de C (Fig. Anexo A7) e 21 exemplos de NC (Fig. Anexo A8) para compor a base de teste de aprendizagem. Foi executada uma redução de cor para 5 cores com o algoritmo de *Median Cut* [GOM 94] em RGB e HSV. O passo de varredura da janela de procura em X, Y utilizado foi (20, 20) pixels, valor de limiar de 0,90 e área de intersecção do retângulo da janela de procura com os retângulos de *Classe* anotados nas *imagens-de-busca* maior ou igual a 75%.

Para os testes de classificação foram preparadas quinze imagens nas quais foram inseridas *imagens-exemplo* que não fizeram parte do treinamento da RNA (Fig. Anexo A9). Seis destas imagens são fotografias da BDI *Miscellaneous* do *Signal and Image Processing Institute* da *University of Southern California*¹. Cinco imagens são do tipo *clipart* e obtidas do CD *Media Content Office XP-Clipart 1* e as restantes do CD *Media Content Office XP-PHOTO1*. As *imagens-exemplo* foram inseridas nas imagens aplicando-se rotação, alteração de escala e sobreposição.

5.2.1.1 Sistema de cor RGB, 5 cores, método T1

Este experimento utiliza somente o HC para compor o vetor de características. Após a aplicação da redução de cores com *Median Cut* calcula-se o HC com base na paleta de cores gerada. O vetor de características contendo o HC é então ordenado por ordem crescente de valor de frequência de ocorrência e normalizado para o intervalo [0, 1]. Como o *Median Cut* executa a redução de cores no sistema RGB e o HC é calculado pelo **índice da paleta** gerada e não pelas compo-

¹ <http://sipi.usc.edu/services/database/database.cgi?volume=misc>, data da última visita em 01/06/2003.

mentos de cor, não existe diferença entre o HC para o sistema RGB e HSV. Sendo assim, este experimento não foi executado para sistema de cor HSV.

Para uma redução de cores a n cores, o tamanho do vetor de características deste método é n . A Tabela 5.2 apresenta um exemplo do vetor de características obtido com a utilização método T1, histograma de cores onde:

1. HC_1 é a contagem de ocorrência da cor Cor_1 (cor obtida pelo *Median Cut* com a menor frequência de ocorrência);
2. HC_2 é a contagem de ocorrência da cor Cor_2 (cor obtida pelo *Median Cut* com a segunda menor frequência de ocorrência);
3. ...;
4. HC_n é a contagem de ocorrência da cor Cor_n (cor obtida pelo *Median Cut* com a maior frequência de ocorrência).

Tabela 5.2 – Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T1 (Histograma de Cores).

Método	Entradas					Saída / Rótulo	
	HC_1	HC_2	HC_3	...	HC_n	C_1	C_2
T1	0,03	0,10	0,23		0,45	1	0

A utilização do HC como primeira tentativa foi fundamentada na revisão bibliográfica feita, que apontava o HC como um dos métodos mais utilizados em CBIR e reportado como um método satisfatório para a recuperação de imagens [ANT 02] [BRU 99].

Tabela 5.3 - Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T1.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Média	2614,800	1380,400	84,285
Desvio Padrão	843,300	1081,185	1,304

A coluna *generalização* da tabela acima (e tabelas similares) refere-se a taxa de generalização, ou seja, o percentual de acertos na classificação dos exemplos da base de teste.

Tabela 5.4 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T1.

	Treino Inicial
Épocas de Aprendizagem	1650
Melhor Época	1104

Melhor Generalização (%)	85,714
Neurônios de Entrada + Ocultos	5
Neurônios Adicionados	4
Semente	1054908379

Na coluna *Imagem* da Tabela 5.5 tem-se as quinze imagens utilizadas para o reconhecimento, a quantidade de padrões inseridos na imagem está indicada entre parênteses. Para um exemplo desta ver a figura Fig. Anexo A9.

Tabela 5.5 - Totalizações de regiões FP, FN e C para treino inicial da RNA. - BDI B0, RGB 5 cores, T1.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)
B0_CP_01 (1)	3	3	0
B0_CP_02 (3)	61	1	1
B0_CP_03 (3)	46	14	2
B0_CP_04 (3)	2	2	4
B0_CP_05 (3)	196	11	3
B0_CP_06 (4)	7	15	6
B0_CP_07 (3)	14	8	7
B0_CP_08 (2)	60	7	2
B0_CP_09 (3)	228	5	7
B0_CP_10 (3)	97	5	7
B0_CP_11 (6)	53	13	4
B0_CP_12 (4)	63	10	5
B0_CP_13 (3)	79	3	5
B0_CP_14 (4)	65	5	4
B0_CP_15 (3)	127	7	6
Média	73,400	7,267	4,200
Desvio Padrão	66,582	4,464	2,242

Verifica-se que com uma taxa de generalização de aproximadamente 85% a ocorrência de FP e FN é elevada. O HC possui somente a frequência de ocorrência da cor, mas não a informação da cor em si. Portanto regiões da imagem que tenham frequência similar às do padrão aprendido, mas com cores diferentes, são classificadas erroneamente. A falta da informação da cor associada ao HC nos levou a realizar experiências com o método T2 que será apresentado na próxima seção.

5.2.1.2 Sistema de cor RGB, 5 clusters, método T2

Visando uma melhor compreensão do significado dos padrões que estavam sendo apresentados à RNA procurou-se uma maneira de visualizá-los espacialmente. Assumindo que

padrões do tipo C ocupam uma região “próxima” no espaço de cor adotado porque possuem cores e proporção de cores semelhantes, e que padrões NC não possuiriam esta semelhança espacial, exportou-se em VRML as componentes da cor para cada pixel do padrão (o índice de cada pixel possui uma correspondência espacial no sistema de cor, conforme foi visto na seção 2.2.1.2) gerando uma nuvem de pontos em 3D.

A análise visual de algumas das nuvens geradas indicou que padrões semelhantes ocupavam uma *região aparentemente semelhante* no espaço de cor. Também se verificou que algumas nuvens apresentavam uma translação no espaço de cor, a qual foi atribuída à diferença no grau de iluminação entre as cenas. Como resultado desta análise se procurou um método que permitisse encontrar uma quantidade reduzida de pontos (i.e., protótipos, centróides dos aglomerados) que caracterizassem a nuvem. Métodos de clusterização possuem esta capacidade (ver seção 3.2) onde optou-se pelo método *k-Means* [THE 98]. Nas figuras Fig. Anexo A1 e Fig. Anexo A2 pode-se visualizar duas nuvens de pontos em RGB com os seus respectivos centróides (cinco centróides). Os padrões a partir dos quais as nuvens foram geradas são oriundos do mesmo objeto (lata de Coca-Cola), mas de imagens com rotação diferente da lata. Nota-se que a “forma” da nuvem permanece similar entre as duas cenas, mas alguns centróides sofreram uma translação dentro da nuvem. As figuras Fig. Anexo A3 e Fig. Anexo A4 apresentam duas nuvens de pontos de *Não-Classe* onde pode-se verificar que suas formas são bem distintas.

O vetor de características do método T2 contendo somente os centróides é ordenado por ordem crescente de valor da distância euclidiana de cada centróide e normalizado para o intervalo [0, 1]. Para uma clusterização a n centróides, o tamanho do vetor de características deste método é $n * 3$. Na Tabela 5.6 apresentamos um exemplo do vetor de características obtido com a utilização método T2, redução de cores com método k-Means onde:

1. R_{C1} , G_{C1} , B_{C1} são os centróides de *Red*, *Green* e *Blue* da cor Cor_1 ;
2. R_{C2} , G_{C2} , B_{C2} são os centróides de *Red*, *Green* e *Blue* da cor Cor_2 ;
3. ...;
4. R_{Cn} , G_{Cn} , B_{Cn} são os centróides de *Red*, *Green* e *Blue* da cor Cor_n .

Tabela 5.6 - Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T2 (clusterização k-Means).

Método	Entradas										Saída / Rótulo	
T2	R_{C1}	G_{C2}	B_{C1}	R_{C2}	G_{C2}	B_{C2}	...	R_{Cn}	G_{Cn}	B_{Cn}	C1	C2
	0,0	0,90	0,0	0,30	0,45	0,9		0,1	0,4	0,3	1	0

Durante a geração das bases de aprendizado e de teste para BDI B0 verificou-se que a clusterização com *k-Means* não é adequada para imagens que possuem pouca variação de cores. Nestes casos, nem sempre o algoritmo *k-Means* consegue inicializar cada centróide com valores diferentes dos outros centróides, o que causa que o algoritmo seja encerrado prematuramente com

erro. Como a inicialização dos centróides é aleatória e existem muitos valores iguais no vetor de entrada, alguns centróides são inicializados com valores iguais, sendo esta uma característica do algoritmo utilizado.

Dos 42 exemplos de aprendizado, o método conseguiu clusterizar apenas 23. E dos 21 exemplos de teste, apenas 15 foram clusterizados. Para cada imagem tenta-se até 4.000 vezes clusterizar com sucesso a imagem antes de finalizar o processo com a indicação de insucesso da clusterização.

Tabela 5.7 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 clusters, T2.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Média	170,000	88,400	87,222
Desvio Padrão	10,886	67,652	1,521

Tabela 5.8 – Melhor treinamento da RNA para BDI B0, RGB 5 clusters, T2.

	Treino Inicial
Épocas de Aprendizagem	169
Melhor Época	160
Melhor Generalização (%)	88,888
Neurônios de Entrada + Ocultos	15
Neurônios Adicionados	0
Semente	1055127292

Tabela 5.9 - Totalizações de regiões FP, FN e C para treino inicial da RNA, BDI B0, RGB 5 clusters, T2.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)
B0_CP_01 (1)	12	0	3
B0_CP_02 (3)	52	0	2
B0_CP_03 (3)	282	3	13
B0_CP_04 (3)	133	0	6
B0_CP_05 (3)	54	2	12
B0_CP_06 (4)	273	0	21
B0_CP_07 (3)	136	3	12
B0_CP_08 (2)	6	4	5
B0_CP_09 (3)	64	0	12
B0_CP_10 (3)	45	4	8
B0_CP_11 (6)	55	4	13
B0_CP_12 (4)	29	10	5
B0_CP_13 (3)	9	6	2
B0_CP_14 (4)	12	5	4
B0_CP_15 (3)	172	5	8
Média	88,933	3,067	8,400

Desvio Padrão	91,494	2,865	5,343
----------------------	--------	-------	-------

Como podemos verificar na Tabela 5.9, o método T2 possui uma ocorrência média de regiões FP maior que o método T1 (ver Tabela 5.5) apesar da taxa de generalização obtida no treino da RNA ser maior para o método T2. Contudo, a ocorrência de regiões FN é menor e a ocorrência de regiões C é maior que no método T1. Numa análise estatística (não apresentada neste trabalho) que foi executada a partir dos cinco centróides obtidos de regiões de latas de Coca-Cola, verificou-se que o desvio padrão para cada componente de cor de cada centróide era elevado (como também pode ser verificado na Tabela 5.9). Isto demonstra uma grande variação dos centróides que, em uma análise visual inicial, indicavam ocupar *regiões aparentemente semelhantes*.

A partir destes resultados, verifica-se que a frequência de ocorrência de cores é uma informação útil, além das cores. Sendo assim, o método T3, que é composto pelo HC e índices de cores quantizadas, é proposto como uma consequência natural de aprimoramento das técnicas de pré-processamento utilizadas.

5.2.1.3 Sistema de cor RGB, 5 cores, método T3

O objetivo deste experimento é verificar a *performance* do método T3 e compará-lo com os métodos T1 e T2.

O vetor de características do método T3 contém o HC com as triplas de componentes das cores da imagem quantizada com *Median Cut*. Portanto, para uma redução de cores a n cores, o tamanho do vetor de características resultante é $n * 4$. Este vetor é ordenado por ordem crescente do valor de frequência de ocorrência e normalizado para o intervalo $[0, 1]$. Na Tabela 5.6 apresentamos um exemplo do vetor de características obtido com a utilização método T3 onde:

1. HC_1 é a contagem de ocorrência da cor Cor_1 (cor obtida pelo *Median Cut* com a menor frequência de ocorrência);
2. HC_2 é a contagem de ocorrência da cor Cor_2 (cor obtida pelo *Median Cut* com a segunda menor frequência de ocorrência);
3. ...;
4. HC_n é a contagem de ocorrência da cor Cor_n (cor obtida pelo *Median Cut* com a maior frequência de ocorrência).
5. R_{C1}, G_{C1}, B_{C1} são os centróides de *Red, Green* e *Blue* da cor Cor_1 ;
6. R_{C2}, G_{C2}, B_{C2} são os centróides de *Red, Green* e *Blue* da cor Cor_2 ;
7. ...;
8. R_{Cn}, G_{Cn}, B_{Cn} são os centróides de *Red, Green* e *Blue* da cor Cor_n .

Tabela 5.10 - Exemplo da composição do vetor de características apresentado como entrada à RNA para o método T2 (clusterização k-Means).

Método	Entradas										Saída / Rótulo	
	HC_1	...	HC_n	RC_1	GC_1	BC_1	...	RC_n	GC_n	BC_n	C_1	C_2
T2	0,0		1,0	0,30	0,45	0,9		0,1	0,4	0,3	1	0

As tabelas Tabela 5.11 e Tabela 5.12 apresentam os resultados das simulações de aprendizagem das RNAs, podemos verificar que a taxa de generalização média e de melhor generalização são inferiores as das obtidas para os métodos T1 e T2. Na Tabela 5.13 apresentamos os resultados obtidos para o reconhecimento na BDI B0.

Tabela 5.11 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T3.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	136,000	100,400	82,381
Treino Inicial σ	18,534	7,829	1,304

Tabela 5.12 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T3.

	Treino Inicial
Épocas de Aprendizagem	112
Melhor Época	95
Melhor Generalização (%)	83,333
Neurônios de Entrada + Ocultos	20
Neurônios Adicionados	0
Semente	1055160204

Tabela 5.13 – Totalizações de regiões de FP, FN e C para treino inicial da RNA, BDI B0, 5 cores, T3.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)
B0_CP_01 (1)	58	1	1
B0_CP_02 (3)	0	3	0
B0_CP_03 (3)	46	12	4
B0_CP_04 (3)	9	3	3
B0_CP_05 (3)	153	12	2
B0_CP_06 (4)	4	17	4
B0_CP_07 (3)	6	10	5
B0_CP_08 (2)	36	7	2
B0_CP_09 (3)	174	6	6
B0_CP_10 (3)	79	7	5
B0_CP_11 (6)	32	12	5
B0_CP_12 (4)	26	11	4

B0_CP_13 (3)	49	3	5
B0_CP_14 (4)	48	4	5
B0_CP_15 (3)	16	6	7
Média	49,067	7,600	3,867
Desvio Padrão	51,674	4,548	1,922

A Tabela 5.14 apresenta um resumo comparativo dos três métodos propostos. Verifica-se que o método T3 proporcionou uma redução significativa na ocorrência de regiões de FP, demonstrando que o método tem uma *performance* melhor do que os outros dois métodos propostos somente utilizando-se o treino inicial da RNA.

Tabela 5.14 – Resumo das médias e desvios padrões do treino inicial das RNAs para os métodos T1, T2 e T3.

	T1	T2	T3
FP(0) μ	73,400	88,933	49,067
FP(0) σ	66,582	91,494	51,674
FN(0) μ	7,267	3,067	7,600
FN(0) σ	4,464	2,865	4,548
C(0) μ	4,200	8,400	3,867
C(0) σ	2,242	5,343	1,922

Uma vez verificado que o método T3 apresenta *performance* melhor do que os outros dois métodos, os experimentos a seguir irão se concentrar em comparar a *performance* das abordagens RIFP e NNC no sistema de cor RGB e demonstrar um exemplo prático do uso da aprendizagem incremental da RNA *Cascade Correlation*. A partir do treino inicial da RNA, cada iteração das abordagens utiliza uma RNA previamente treinada e continua-se a aprendizagem a partir deste ponto. Ou seja:

- Treino Inicial \rightarrow Pesos \Rightarrow RIFP 1 \rightarrow Pesos \Rightarrow RIFP 2
- Treino Inicial \rightarrow Pesos \Rightarrow NNC 1 \rightarrow Pesos \Rightarrow NNC 2

Tabela 5.15 – Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, RGB 5 cores, T3.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	136,000	100,400	82,381
Treino Inicial σ	18,534	7,829	1,304
RIFP 1 μ	585,400	227,400	84,762
RIFP 1 σ	53,543	311,430	2,130
RIFP 2 μ	123,000	1,000	88,095
RIFP 2 σ	0,000	0,000	0,000
NNC 1 μ	409,000	1,000	83,333

NNC 1 σ	10,223	0,000	0,000
NNC 2 μ	826,200	1,000	83,333
NNC 2 σ	126,529	0,000	0,000

Tabela 5.16 – Melhor treinamento da RNA para BDI B0, RGB 5 cores, T3.

	Treino Inicial	RIFP 1	RIFP 2	NNC 1	NNC 2
Épocas de Aprendizagem	112	624	123	398	731
Melhor Época	95	609	1	1	1
Melhor Generalização (%)	83,333	88,095	88,095	83,333	83,333
Neurônios de Entrada + Ocultos	20	20	22	20	20
Neurônios Adicionados	0	2	0	0	0
Semente	1055160204	1055164153	1055164863	1055166611	1055167014
Quantidade de RIFP Adicionadas	N/A	153 de 736	16 de 154	N/A	N/A
Quantidade de NNC Adicionadas	N/A	N/A	N/A	150	150

Para a segunda iteração da abordagem RIFP a quantidade de regiões de imagens adicionadas ao treinamento não foi suficiente para causar uma modificação na taxa de generalização em relação à primeira iteração de RIFP.

Verifica-se que para BDI B0 com redução para cinco cores nesta configuração a abordagem NNC não apresentou melhora de generalização, pois a melhor generalização ocorreu já na primeira época de treino. Isto indica que a nuvem de não-classe adicionada causou uma diminuição da taxa de generalização em relação ao treino inicial. Executando o treinamento da RNA para esta abordagem sem a utilização do aprendizado incremental (i.e., executando o treinamento completo da RNA), conseguiu-se melhorar a taxa de generalização conforme pode ser verificado na Tabela 5.18. Na Tabela 5.20 podemos verificar que neste novo experimento (não incremental) as médias de FP, FN e C melhoraram em relação ao treino inicial. A verificação e melhor compreensão deste comportamento requer a realização de mais experimentos, onde infelizmente estes não puderam ser realizados em função das limitações de tempo relacionadas à realização deste trabalho.

Tabela 5.17 – Totalizações de regiões FP, FN e C para treino inicial da RNA, RIFP e NNC após duas iterações. - BDI B0, RGB 5 cores, T3.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)	RIFP FP(2)	RIFP FN(2)	RIFP C(2)	NNC FP(2)	NNC FN(2)	NNC C(2)
B0_CP_01 (1)	58	1	1	0	2	1	0	3	0
B0_CP_02 (3)	0	3	0	7	1	1	58	1	1
B0_CP_03 (3)	46	12	4	11	12	4	43	13	3
B0_CP_04 (3)	9	3	3	7	3	3	8	3	3
B0_CP_05 (3)	153	12	2	2	11	3	150	12	2
B0_CP_06 (4)	4	17	4	3	19	2	4	17	4
B0_CP_07 (3)	6	10	5	14	1	14	6	10	5
B0_CP_08 (2)	36	7	2	14	4	5	35	7	2

B0_CP_09 (3)	174	6	6	19	1	11	174	6	6
B0_CP_10 (3)	79	7	5	2	10	2	78	7	5
B0_CP_11 (6)	32	12	5	14	8	9	32	13	4
B0_CP_12 (4)	26	11	4	13	11	4	26	11	4
B0_CP_13 (3)	49	3	5	20	0	8	49	3	5
B0_CP_14 (4)	48	4	5	23	1	8	48	4	5
B0_CP_15 (3)	16	6	7	5	2	11	16	6	7
Média	49,067	7,600	3,867	10,267	5,733	5,733	48,467	7,733	3,733
Desvio Padrão	51,674	4,548	1,922	7,206	5,700	4,131	51,297	4,698	1,907

Na Tabela 5.17 verifica-se que a abordagem RIFP com aprendizado incremental contribuiu para a queda de ocorrência de FP e FN e aumento da classificação correta dos padrões. Verifica-se também que a abordagem NNC com aprendizado incremental não apresentou melhorias significativas em relação ao treino inicial, o que está de acordo com a taxa de generalização da RNA obtida para a abordagem NNC (Tabela 5.16).

Tabela 5.18 - Médias e desvios padrões do treinamento de vinte RNAs para BDI B0, RGB 5 cores, NNC, T3, sem aprendizado incremental.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	136,000	100,400	82,381
Treino Inicial σ	18,534	7,829	1,304
NNC 1 μ	781,053	543,947	75,939
NNC 1 σ	181,516	186,266	5,662

Tabela 5.19 - Melhor treinamento da RNA para BDI B0, RGB 5 cores, NNC, T3, sem aprendizado incremental.

	Treino Inicial	NNC 1
Épocas de Aprendizagem	112	743
Melhor Época	95	730
Melhor Generalização (%)	83,333	92,857
Neurônios de Entrada + Ocultos	20	20
Neurônios Adicionados	0	3
Semente	1055160204	1055333557
Quantidade de NNC Adicionadas	N/A	150

Tabela 5.20 - Totalizações de regiões FP, FN e C para treino inicial da RNA, NNC após uma iteração. - BDI B0, RGB 5 cores, T3, sem aprendizado incremental.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)	NNC FP(1)	NNC FN(1)	NNC C(1)
B0_CP_01 (1)	58	1	1	0	1	2
B0_CP_02 (3)	0	3	0	20	1	1
B0_CP_03 (3)	46	12	4	25	9	7
B0_CP_04 (3)	9	3	3	13	3	3

B0_CP_05 (3)	153	12	2	12	11	3
B0_CP_06 (4)	4	17	4	11	15	6
B0_CP_07 (3)	6	10	5	19	2	13
B0_CP_08 (2)	36	7	2	41	5	4
B0_CP_09 (3)	174	6	6	32	1	11
B0_CP_10 (3)	79	7	5	23	7	5
B0_CP_11 (6)	32	12	5	26	3	14
B0_CP_12 (4)	26	11	4	46	7	8
B0_CP_13 (3)	49	3	5	47	3	5
B0_CP_14 (4)	48	4	5	59	0	9
B0_CP_15 (3)	16	6	7	19	2	11
Média	49,067	7,600	3,867	26,200	4,667	6,800
Desvio Padrão	51,674	4,548	1,922	16,037	4,320	4,074

Na Fig. Anexo A10 tem-se um exemplo do resultado do reconhecimento utilizando o treino inicial da RNA e na Fig. Anexo A11 o resultado após duas iterações de RIFP. Nota-se que além de uma grande redução na ocorrência de regiões FP (que ficaram localizadas em torno dos padrões), a RNA foi capaz de classificar corretamente dois dos três padrões existentes na imagem. A Fig. Anexo A12 demonstra o efeito da abordagem NNC com aprendizado incremental que praticamente não apresenta redução nas regiões de FP em relação ao reconhecimento obtido com o treino inicial da RNA. Já na Fig. Anexo A13, obtida com um refinamento com a abordagem NNC sem utilizar o aprendizado incremental, percebe-se uma redução significativa nas regiões FP com resultados similares aos obtidos com duas iterações de RIFP.

Tabela 5.21 – Resumo das médias e desvios padrões dos melhores resultados obtidos para o método T3 utilizando as abordagens RIFP e NNC para BDI B0, 5 cores.

Regiões	Treino Inicial	RIFP (2)	NNC (2) Incremental	NNC (1) Não incremental
FP μ	49,067	10,276	48,467	26,200
FP σ	51,674	7,206	51,297	16,037
FN μ	7,600	5,733	7,733	4,667
FN σ	4,548	5,700	4,698	4,320
C μ	3,867	5,733	3,733	6,800
C σ	1,922	4,131	1,907	4,074

Na Tabela 5.21 apresentamos um resumo comparativo dos melhores resultados obtidos para o treino inicial da RNA, abordagens RIFP e NNC. Pode-se verificar que a abordagem RIFP apresenta a melhor *performance* para redução de regiões FP e que a abordagem NNC com uma iteração sem aprendizagem incremental apresentou melhorias significativas. Ambas as abordagens, RIFP e NNC, apresentaram redução nas regiões de FP nas imagens da BDI B0.

5.2.2 Seleção do Sistema de Cor (BDI B0, HSV, T3, 5 cores)

Os objetivos deste experimento são comparar a performance das abordagens RIFP e NNC no sistema de cor HSV e RGB (considerando os resultados obtidos para RGB em 5.2.1.3) para determinar a influência destes sistemas de cor na redução de regiões de imagens FP e demonstrar a utilização da aprendizagem incremental da RNA *Cascade Correlation*. A partir do treino inicial da RNA, cada iteração das abordagens utiliza uma RNA previamente treinada e continua-se a aprendizagem a partir deste ponto. Ou seja:

- Treino Inicial -> Pesos => RIFP 1 -> Pesos => RIFP 2
- Treino Inicial -> Pesos => NNC 1 -> Pesos => NNC 2

Tabela 5.22 - Médias e desvios padrões do treinamento de cinco RNAs para BDI B0, HSV 5 cores, T3.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	97,400	83,000	87,924
Treino Inicial σ	10,738	14,916	4,496
RIFP 1 μ	1597,400	258,600	90,952
RIFP 1 σ	345,311	576,011	1,065
RIFP 2 μ	729,000	1,000	92,857
RIFP 2 σ	45,233	0,000	0,000
NNC 1 μ	781,000	1,000	90,476
NNC 1 σ	84,507	0,000	0,000
NNC 2 μ	1031,600	1,000	90,476
NNC 2 σ	43,050	0,000	0,000

Tabela 5.23 – Melhor treinamento da RNA para BDI B0, HSV 5 cores, T3.

	Treino Inicial	RIFP 1	RIFP 2	NNC 1	NNC 2
Épocas de Aprendizagem	82	1758	664	727	987
Melhor Época	82	1289	1	1	1
Melhor Generalização (%)	90,476	92,857	92,857	90,476	90,476
Neurônios de Entrada + Ocultos	20	20	23	20	20
Neurônios Adicionados	0	3	0	0	0
Semente	1055169549	1055176694	1055178369	1055185100	1055185438
Quantidade de RIFP Adicionadas	N/A	163 de 907	10 de 69	N/A	N/A
Quantidade de NNC Adicionadas	N/A	N/A	N/A	150	150

Na coluna “RIFP 1” e “RIFP 2”, linha “quantidade de RIFP adicionadas” da Tabela 5.23 estão indicados a quantidade de regiões FP adicionadas nesta iteração e a quantidade total de regiões FP obtidas na iteração anterior.

Tabela 5.24 - Totalizações de regiões FP, FN e C para treino inicial da RNA, RIFP e NNC após duas iterações, - BDI B0, HSV 5 cores, T3.

Imagem (total de padrões)	FP(0)	FN(0)	C(0)	RIFP FP(2)	RIFP FN(2)	RIFP C(2)	NNC FP(2)	NNC FN(2)	NNC C(2)
B0_CP_01 (1)	2	1	2	1	2	1	2	1	2
B0_CP_02 (3)	50	1	1	3	2	0	50	1	1
B0_CP_03 (3)	23	13	3	4	11	5	22	13	3
B0_CP_04 (3)	54	1	5	4	2	4	53	1	5
B0_CP_05 (3)	174	11	3	4	10	4	173	11	3
B0_CP_06 (4)	23	12	9	4	18	3	23	12	9
B0_CP_07 (3)	16	8	7	2	14	1	15	8	7
B0_CP_08 (2)	48	7	2	4	5	4	48	7	2
B0_CP_09 (3)	143	4	8	3	7	5	143	4	8
B0_CP_10 (3)	81	5	7	9	6	6	81	6	6
B0_CP_11 (6)	72	7	10	9	12	5	72	7	10
B0_CP_12 (4)	49	5	10	6	13	2	49	5	10
B0_CP_13 (3)	73	2	6	3	3	5	73	2	6
B0_CP_14 (4)	57	5	4	8	6	3	56	5	4
B0_CP_15 (3)	42	8	5	5	12	1	42	8	5
Média	60,467	6,000	5,467	4,600	8,200	3,267	60,133	6,067	5,400
Desvio Padrão	45,832	3,946	2,973	2,414	5,046	1,870	45,801	3,936	2,947

Na segunda linha da Tabela 5.24 (imagem B0_CP_02) verifica-se que houve a ocorrência de zero C reconhecidas para a abordagem RIFP. Este comportamento pode estar relacionado com a adição de exemplos FP que causam um ajuste das superfícies de separação muito restrito (e.g. adição de regiões FP que foram reconhecidas como FP por não atenderem ao critério de área mínima de sobreposição da janela de pesquisa com a área do padrão). Alterando-se o valor do critério de limiar conforme discutido na seção 4.1.2 e Figura 4.15, de 0,90 para 0,80, obteve-se a seguinte contagem: FP = 4, FN = 1 e C = 1. Uma outra possibilidade seria a de modificar o passo de varredura da janela de pesquisa, permitindo assim um enquadramento diferente. Por exemplo, alterando-se o valor de passo X, Y da janela de varredura para (3, 15) obteve-se a seguinte contagem: FP = 23, FN = 11 e C = 5.

Estas duas alternativas implicam que se deve encontrar também um balanço adequado para uma determinada BDI entre aumentar a quantidade de C reconhecidas e diminuir a ocorrência de FP e/ou FN. Este “balanço adequado” pode, portanto, ser obtido através de alterações do valor de limiar e do passo da janela de procura.

O sistema de cor HSV apresenta comportamento similar aos experimentos realizados com o sistema de cor RGB apresentados na seção anterior. A segunda iteração da abordagem RIFP também não apresenta melhorias em relação com a primeira e a abordagem NNC com aprendizado incremental não apresenta melhorias na redução de ocorrência de FP e FN em relação ao treino inicial da RNA em HSV.

5.2.3 Análise dos Resultados

Na classificação e recuperação de imagens é interessante minimizar a ocorrência de FP e FN e maximizar a de C. A Figura 5.4 apresenta um gráfico comparativo da taxa de generalização obtida nos treinamentos da RNA para cada abordagem por sistema de cor. A maior generalização ocorre para o sistema de cor HSV utilizando a abordagem RIFP na segunda iteração. Comparando-se a contagem de FP, FN e C para as abordagens RIFP e NNC utilizando RGB e HSV (Figura 5.5) verifica-se que o sistema de cor RGB oferece desempenho médio entre FP, FN e C melhor do que o sistema de cor HSV para as abordagens RIFP e NNC. Note-se que se for avaliada somente a taxa de generalização obtida no treinamento da RNA o sistema de cor mais apropriado para ser utilizado seria o HSV, e não o RGB. Note-se também que os valores de regiões de FP, FN e C para RGB são, em média, melhores. Temos um caso de decisão multi-critério onde devemos escolher entre a melhor taxa de generalização ou a redução de FP. A partir desta análise, iremos adotar somente o sistema de cor RGB para os experimentos com as BDIs B1 e B2.

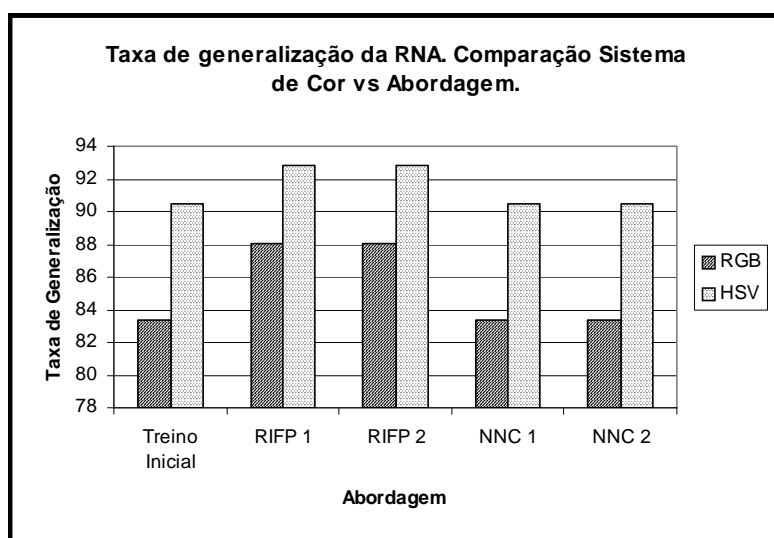


Figura 5.4 – Gráfico comparativo da taxa de generalização da RNA x Abordagem para RGB e HSV, método T3.

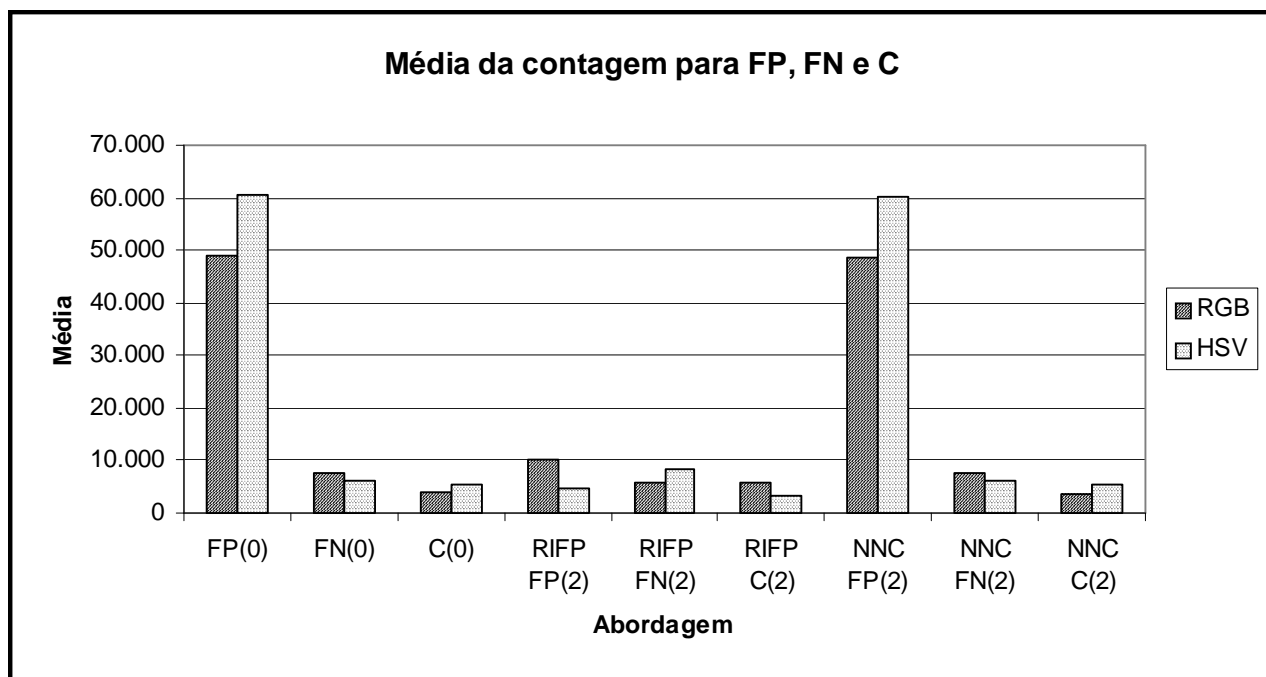


Figura 5.5 - Gráfico comparativo das médias para FP, FN, C x abordagem para HSV e RGB, método T3.

5.2.4 Desempenho em Base de *Benchmark* (BDI B1)

A base de imagens *Columbia Object Image Library* (COIL) [NEN 96] é uma base de *benchmark* disponível na Internet para *download*. Esta base contém fotografias de 100 objetos (identificados por *Obj0* a *Obj100*). Para cada objeto da base foram tiradas 72 fotografias, cada uma delas com uma rotação de 5 graus do objeto entre fotos, totalizando 7200 imagens. Todos os objetos foram fotografados com fundo preto. Esta base de 7200 imagens é conhecida como COIL-100 (Fig. Anexo A14).

Destes 100 objetos selecionou-se o *Obj25* (Fig. Anexo A15) para ser utilizado como *imagem-de-treinamento*. Das 72 imagens do *Obj25*, 36 foram escolhidas e submetidas à fase de preparação (janelamento das imagens para regiões de 64 x 64 pixels) resultando em 192 *imagens-de-treinamento* para aprendizado do tipo *Classe* e 96 *imagens-de-treinamento* para teste do tipo *Classe*. Os exemplos para aprendizado (192 exemplos) e teste (96 exemplos) de *Não-Classe* foram extraídos de imagens fotográficas diversas.

Devido a grande quantidade de imagens da base COIL, selecionou-se 333 imagens das 7200 para executar uma pré-verificação da *performance* obtida para cada abordagem. No caso da abordagem RIFP, as regiões de FP utilizadas para o refinamento vieram exclusivamente destas 333 imagens. Esta pequena base foi denominada de *base de pré-validação*.

Os objetivos para os experimentos com esta base são de verificar a redução na ocorrência de regiões FP nas imagens utilizando uma BDI não “artificial” com grande quantidade de

imagens e determinar a *performance* de **recuperação** de imagens. Também visou-se comparar os resultados obtidos com outro método de recuperação de imagens baseado em HC que não se utiliza de técnicas de IA cujos resultados obtidos com esta mesma base (COIL-100) são apresentados por Moschetta [MOS 01].

5.2.4.1 Sistema de cor RGB, 5 cores, método T3

Para estes experimentos não foi utilizado o aprendizado incremental na RNA *Cascade Correlation*. A Tabela 5.25 apresenta as médias e desvios padrões para vinte e cinco RNAs. É característico de uma RNA *Cascade Correlation* apresentar pouca variação na taxa de generalização obtida e isto pode ser verificado pelos desvios padrões apresentados nesta tabela.

Tabela 5.25 - Médias e desvios padrões do treinamento de 25 RNAs para BDI B1, RGB 5 cores, T3.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	808,400	767,200	86,625
Treino Inicial σ	208,060	214,110	5,937
RIFP 1 μ	3863,200	3337,160	92,500
RIFP 1 σ	722,210	586,203	2,781
RIFP 2 μ	4162,08	3768,48	93,729
RIFP 2 σ	895,364	914,132	3,429
NNC 1 μ	1351,440	1254,920	87,480
NNC 1 σ	407,173	385,369	4,491
NNC 2 μ	1345,880	1260,600	89,022
NNC 2 σ	386,184	362,309	3,642

Tabela 5.26 - Melhor treinamento da RNA para base de pré-validação BDI B1, RGB 5 cores, T3.

	Treino Inicial	RIFP 1	RIFP 2	NNC 1	NNC 2
Épocas de Aprendizagem	1044	4190	3638	937	884
Melhor Época	1037	3976	3562	832	882
Melhor Generalização (%)	96,88	97,40	98,96	96,88	94,27
Neurônios de Entrada	20	20	20	20	20
Neurônios Adicionados	3	12	7	2	3
Semente	1056125940	1056127595	1056128831	1056131000	1056131915
Quantidade de RIFP Adicionadas	N/A	202 de 1076	84 de 171	N/A	N/A
Quantidade de NNC Adicionadas	N/A	N/A	N/A	150	150

Na coluna RIFP 1 e RIFP 2, linha “quantidade de RIFP adicionadas” (Tabela 5.26), os valores descritos referem-se a quantidade de regiões FP adicionadas ao treino retiradas do total de regiões FP que ocorreram na última iteração (por exemplo, na RIFP 1, o treino inicial gerou 1076 regiões FP das quais apenas 202 foram utilizadas para o refinamento).

Tabela 5.27 – Total de imagens recuperadas para base de pré-validação BDI B1, RGB 5 cores, RIFP, T3.

Imagens (333)	FP(0)	FN(0)	C(0)	RIFP FP(1)	RIFP FN(1)	RIFP C(1)	RIFP FP(2)	RIFP FN(2)	RIFP C(2)
Média de Regiões	3,231	0,000	0,000	0,513	0,000	0,000	0,480	0,000	0,000
Dev. Padrão de Regiões	4,077	0,000	0,000	1,248	0,000	0,000	1,188	0,000	0,000
Total Recuperadas	195	0	0	91	0	0,000	77	0	0

Tabela 5.28 - Total de imagens recuperadas para base de pré-validação BDI B1, RGB 5 cores, NNC, T3.

Imagens (333)	FP(0)	FN(0)	C(0)	NNC FP(1)	NNC FN(1)	NNC C(1)	NNC FP(2)	NNC FN(2)	NNC C(2)
Média de Regiões	3,231	0,000	0,000	3,555	0,000	0,000	3,606	0,000	0,000
Dev. Padrão de Regiões	4,077	0,000	0,000	4,440	0,000	0,000	6,081	0,000	0,000
Total Recuperadas	195	0	0	220	0	0	178	0	0

A Fig. Anexo A16 apresenta os objetos que foram utilizados para compor as 333 imagens utilizadas na base de pré-validação. Para a abordagem RIFP as regiões de FP obtidas nesta base de pré-validação foram utilizadas para refinar o treinamento da RNA. Conforme verifica-se na Tabela 5.26 a abordagem RIFP nas duas iterações apresentou aumento na taxa de generalização da RNA e diminuição significativa nas regiões de FP (de 1076 para 171 regiões na segunda iteração). Na Tabela 5.27 temos os totais de imagens recuperadas para cada iteração de RIFP e a média de regiões de FP. Conforme as regiões de FP diminuem a cada iteração da RIFP em cada imagem, existe também uma diminuição na quantidade de imagens FP recuperadas, demonstrando que o método contribuiu para melhorar a *performance* de recuperação de imagens.

A Tabela 5.28 apresenta os resultados obtidos para a abordagem NNC. Na primeira iteração de NNC o método apresentou um aumento na quantidade de regiões FP por imagem e imagens FP recuperadas. Já na segunda iteração ocorreu uma redução na quantidade de imagens FP recuperadas em relação ao treino inicial da rede, mas não nas regiões de FP por imagem. Este comportamento anômalo não pode ser estudado com mais detalhes durante a realização deste trabalho. Acredita-se que este comportamento pode estar associado à particularidade de todos os objetos da base COIL possuírem um fundo preto, muitas vezes numa proporção maior do que as demais cores do objeto. Neste caso, a geração de nuvem de não-classe por distribuição aleatória uniforme pode não ser a melhor abordagem devido a BDI apresentar uma forte tendência relacionada à presença de matizes de preto como fundo.

Tabela 5.29 - Total de imagens recuperadas para BDI B1, RGB 5 cores, RIFP, T3.

Imagens (7200)	<i>FP(0)</i>	<i>FN(0)</i>	<i>C(0)</i>	RIFP FP(1)	RIFP FN(1)	RIFP C(1)	RIFP FP(2)	RIFP FN(2)	RIFP C(2)
Média de Regiões	2,626	0,003	0,246	0,650	0,020	0,229	0,347	0,036	0,213
Dev.Padrão de Regiões	4,443	0,06	2,451	1,637	0,231	2,286	1,041	0,392	2,131
Total Recuperadas	2712	0	72	1785	0	72	1183	0	72

Tabela 5.30 - Total de imagens recuperadas para BDI B1, RGB 5 cores, NNC, T3.

Imagens (7200)	<i>FP(0)</i>	<i>FN(0)</i>	<i>C(0)</i>	NNC FP(1)	NNC FN(1)	NNC C(1)	NNC FP(2)	NNC FN(2)	NNC C(2)
Média de Regiões	2,626	0,003	0,246	3,207	0,001	0,248	3,951	0,008	0,242
Dev.Padrão de Regiões	4,443	0,06	2,451	5,020	0,035	2,472	6,632	0,106	2,411
Total Recuperadas	2712	0	72	3063	0	72	3326	0	72

Tabela 5.31 - Resultados obtidos com recuperação por histograma na base COIL-100 [MOS 01].

Imagem	Total (Recuperados)	Positivos	Negativos
Xícara (Obj25)	3090	72	3018
Pêra (Obj83)	3205	70	3135
Refrigerante (Obj62)	4505	63	4442
Carro Verde (Obj27)	3014	31	2983
Caixa de Remédio (Obj54)	4268	55	4213

Na tarefa de recuperação de imagens com a abordagem RIFP (Tabela 5.29) para o objeto 25 da base COIL-100 (xícara rosa) verifica-se uma redução na quantidade de imagens FP recuperadas, mantendo-se a quantidade de recuperações corretas (72 imagens) ao longo das iterações. Para a abordagem NNC (Tabela 5.30) verifica-se que o método causou um aumento na quantidade de imagens FP ao longo das duas iterações, mas sem deixar de recuperar as setenta e duas imagens com o padrão aprendido. A falha deste método nesta BDI acredita-se estar relacionado ao comportamento anômalo verificado na base de pré-validação. A Tabela 5.31 apresenta os resultados obtidos por Moschetta [MOS 01] para a tarefa de recuperação na base COIL-100 para cinco objetos desta BDI. Na coluna *imagem* está descrito o objeto (*imagem-exemplo*) e o seu índice na BDI COIL-100, na coluna *total recuperados* tem-se o total de imagens que o método por HC recuperou a partir de *imagem-exemplo* apresentada ao programa. Estes valores são discriminados em *positivos* (recuperações corretas do objeto) e *negativos* (imagens FP). Comparando-se os valores para o objeto 25 (xícara rosa) da recuperação baseada em *matching* por HC com os resultados obtidos com emprego de RNA após duas iterações RIFP obteve-se uma redução de 1835 imagens FP. Mesmo so-

mente com o treino inicial da RNA a abordagem proposta resultou numa quantidade menor de imagens FP recuperadas, em relação aos resultados obtidos por Moschetta [MOS 01].

5.2.5 Desempenho com Imagens Não Controladas (BDI B2)

A última BDI B2 é composta por 100 regiões de latas de Coca-Cola (*imagem-exemplo*, Fig. Anexo A17) e 100 regiões de imagens de ambiente diversas (*imagem-de-fundo*, Fig. Anexo A18) utilizadas para executar o aprendizado da RNA. Para o testes de generalização foram utilizadas 50 regiões de latas de Coca-Cola (*imagem-exemplo*, Fig. Anexo A19) e 50 regiões de imagens de ambiente diversas (*imagem-de-fundo*, Fig. Anexo A20). Todas as imagens foram capturadas com câmara fotográfica digital sem controle do tipo de iluminação do ambiente. Para os teste de recuperação preparou-se uma base de imagens com setenta e sete imagens (Fig. Anexo A21), das quais trinta e nove possuem o padrão Coca-Cola. Estas imagens também foram feitas com câmara fotográfica digital sem controle de iluminação em ambientes diversos e não foram submetidas a nenhum tipo de tratamento gráfico.

5.2.5.1 Sistema de cor RGB, 5 cores, método T3

Para a simulação de aprendizagem das RNAs não foi utilizado o aprendizado incremental nestes experimentos. O objetivo deste experimento é verificar a *performance* da abordagem de recuperação e classificação de imagens proposta numa BDI com imagens não “controladas”.

Tabela 5.32 - Médias e desvios padrões do treinamento de 25 RNAs para BDI B2, RGB 5 cores, T3.

	Épocas de Aprendizagem	Melhor Época	Generalização (%)
Treino Inicial μ	297,167	260,042	86,708
Treino Inicial σ	54,959	69,260	2,476
RIFP 1 μ	1308,208	930,167	86,375
RIFP 1 σ	308,326	299,928	1,469
RIFP 2 μ	2391,600	1052,600	77,240
RIFP 2 σ	504,640	909,378	2,634
NNC 1 μ	462,320	435,520	86,720
NNC 1 σ	97,535	103,588	2,923
NNC 2 μ	490,880	445,320	85,200
NNC 2 σ	102,594	109,741	2,646

Tabela 5.33 - Melhor treinamento da RNA para base BDI B2, RGB 5 cores, T3.

	Treino Inicial	RIFP 1	RIFP 2	NNC 1	NNC 2
Épocas de Aprendizagem	364	1669	2913	704	427
Melhor Época	363	1652	2274	698	412
Melhor Generalização (%)	93,00	88,00	86,00	94,00	90,00

Neurônios de Entrada	20	20	20	20	20
Neurônios Adicionados	1	5	6	2	1
Semente	1055910441	1055964898	1055966910	1056045894	1056050380
Quantidade de RIFP Adicionadas	N/A	115 de 2872	107 de 1986	N/A	N/A
Quantidade de NNC Adicionadas	N/A	N/A	N/A	150	150

Tabela 5.34 - Total de imagens recuperadas para BDI B2, RGB 5 cores, RIFP, T3.

77 Imagens (39 possuem o padrão procurado)	<i>FP(0)</i>	<i>FN(0)</i>	<i>C(0)</i>	RIFP FP(1)	RIFP FN(1)	RIFP C(1)	RIFP FP(2)	RIFP FN(2)	RIFP C(2)
Média de Regiões	39,888	2,180	5,513	27,583	3,500	4,194	21,486	4,361	3,333
Dev. Padrão de Regiões	111,277	3,341	9,391	85,799	4,847	7,823	81,203	6,047	7,108
Total de Imagens Recuperadas	17	5	34	22	0	32	9	11	27

A abordagem RIFP (Tabela 5.34) contribuiu para uma redução na recuperação de imagens FP, mas após duas iterações também ocorreu uma redução na recuperação de imagens que possuíam o padrão Coca-Cola que coincide com a redução na taxa de generalização da RNA (Tabela 5.33, RIFP 1 e RIFP 2). Isto é um indicativo de que o tipo de região FP que está sendo adicionado está “destruindo” o conhecimento do padrão Coca-Cola. Uma maneira de abordar isto seria uma extensão natural da abordagem RIFP, ou seja, começar a adicionar regiões de imagens classificadas como FN como sendo C a cada iteração do método.

Tabela 5.35 - Total de imagens recuperadas para BDI B2, RGB 5 cores, NNC, T3.

77 Imagens (39 possuem o padrão procurado)	<i>FP(0)</i>	<i>FN(0)</i>	<i>C(0)</i>	NNC FP(1)	NNC FN(1)	NNC C(1)	NNC FP(2)	NNC FN(2)	NNC C(2)
Média de Regiões	39,888	2,180	5,513	39,955	2,367	5,161	33,208	2,583	5,111
Dev. Padrão de Regiões	111,277	3,341	9,391	103,728	3,480	9,402	102,143	3,855	8,959
Total de Imagens Recuperadas	17	5	34	20	2	32	11	7	32

Na Tabela 5.35 temos os resultados para a abordagem NNC. Neste caso a abordagem NNC obteve uma redução na recuperação de imagens FP com pouca degradação no reconhecimento de imagens que possuam o padrão procura, saindo-se melhor do que a abordagem RIFP. Na Fig. Anexo A22 apresentamos um exemplo do resultado da classificação utilizando o treino inicial da RNA e na Fig. Anexo A23 e Fig. Anexo A24 o resultado para a segunda iteração de RIFP e NNC respectivamente.

5.3 Resumo do Capítulo

Neste capítulo apresentamos os experimentos realizados para validar a classificação e recuperação de imagens utilizando RNAs e a metodologia de aprendizado usada para treinar as RNAs nesta dissertação. As duas abordagens apresentadas, Regiões de Imagens de Falso-Positivo e Nuvem de Não-Classe, utilizadas para fechar as superfícies de separação da RNA visando, desta maneira, reduzir a ocorrência de regiões de FP e recuperação de imagens FP foram utilizadas em três bases de imagens diferentes. Cada uma destas bases possui uma particularidade. A BDI B0 é uma base artificial, onde os padrões foram gerados em um software de edição gráfico. Estes padrões foram então inseridos em imagens existentes para se executar os testes de classificação. Seus objetivos eram: (i) de optar entre os métodos T1, T2 e T3; (ii) de determinar a influência que um sistema de cor (RGB ou HSV) possui na classificação de imagens utilizando RNA (ocorrência de FP, FN e C) e a robustez do método quanto à influência de rotação, translação, escala e oclusão dos padrões nas imagens. Já a BDI B1 é uma base de imagens de objetos reais, mas produzida em condições controladas. Por estar disponível na Internet é uma base útil para ser utilizada como referência para comparação de técnicas de recuperação e classificação de imagens, além de possuir uma quantidade elevada de imagens. Com a BDI B2 apresentamos uma análise da *performance* do método proposto para imagens obtidas em condições de utilização em aplicações normais, i.e., nas quais não houve preocupação em preparar previamente a cena ou pós-processar as imagens.

Verificamos com os experimentos realizados com a BDI B0 que o sistema de cor adotado não possui influência significativa na ocorrência de regiões de FP, FN e C. As abordagens RIFP e NNC obtiveram uma redução da contagem de regiões FP e no reconhecimento dos padrões de C, sendo que a abordagem NNC apresentou comportamento fora do esperado quando se utilizou a aprendizagem incremental.

Para os testes de recuperação de imagens utilizando a BDI B1 com as 7200 imagens, verificou-se que a abordagem RIFP obteve melhores resultados do que a abordagem NNC. Também verificou que na comparação entre o método “tradicional” de recuperação de imagens por *matching* de HC e as abordagens propostas, os métodos baseados em IA, com exceção da abordagem NNC, possuem *performance* melhor, recuperando menos imagens FP e recuperando todas as imagens C que estavam na base.

Para a tarefa de recuperação e classificação de imagens com BDIs não controladas como a BDI B2 verificou-se que ambas as abordagens, RIFP e NNC, obtiveram uma redução na recuperação de imagens FP. No contexto de cada imagem, o reconhecimento por janelamento atua como um mecanismo de atenção no sentido de que as coordenadas da imagem que possuem o padrão são identificadas e podem ser utilizadas para algum processamento posterior.

Também está claro que a abordagem NNC deve ser melhor estudada para procurar identificar as causas dos casos anômalos verificados nos experimentos. Devido à quantidade de combinações de variáveis que podem ser modificadas no IMatch não foi possível, no tempo disponível para a realização deste trabalho, explorar todos os seus efeitos. Podemos citar as variáveis de

valor do limiar, passo de varredura X,Y da janela, quantidade de cores a serem utilizadas na quantização, área mínima de intersecção da janela de procura com a área do padrão na imagem, dimensão da janela de varredura e a geração e distribuição dos pontos NNC como fatores que podem modificar os resultados obtidos.

6 CONCLUSÃO

A tarefa de recuperação e classificação de imagens executada de maneira automática possui grande importância atualmente devido a grande quantidade de imagens armazenadas, muitas vezes de forma desordenada, em meios eletrônicos. Portanto sistemas que são capazes de recuperar uma imagem a partir de um exemplo possuem grande importância para permitir um rápido acesso ao conteúdo visual procurado. Dentro da grande área da visão computacional estes métodos formam a base para tarefas mais desenvolvidas no campo de visão de máquina. Seu estudo, portanto consiste em tema atual de pesquisa e possuem aplicações práticas imediatas em áreas tão diversas quanto localização de objetos numa cena, levantamento de ocorrência de logomarcas na mídia, recuperação de imagens com conteúdo visual semelhante para posterior organização até aplicações em robótica.

Neste trabalho executamos um estudo do estado-da-arte das técnicas utilizadas em CBIR para extração de características baseadas em cor, textura e forma. Abordamos também alguns dos algoritmos de comparação e métricas mais utilizadas de acordo com o tipo de característica utilizado. É de consenso entre os trabalhos revisados que ainda não existe uma única técnica que apresente boa *performance* de recuperação e classificação para qualquer tipo de base de imagens. Dentre os CBIRS apresentados demos destaque a três deles por sua relevância ao tema deste trabalho e por sua importância dentro da área. Todos utilizam técnicas complexas e mistas para extrair as características das imagens, compor as métricas e executar a comparação de características buscando uma maior precisão das imagens recuperadas com o exemplo oferecido ao sistema.

Buscando uma abordagem diferenciada para a tarefa de recuperação e classificação de imagens propomos o uso de aprendizado de máquina para não mais executar uma comparação entre uma imagem de exemplo contra uma imagem da base de imagens, mas sim obter um “modelo” que identifique o objeto procurado a partir de uma coleção destes objetos utilizando a cor como principal característica. Para tanto se optou por utilizar RNAs do tipo MLP com arquitetura *Cascade Correlation* para a tarefa de classificação

Verificamos através dos experimentos realizados que a maior dificuldade para a RNA não era a de aprender o padrão (objeto) procurado na cena, mas sim realizar uma rejeição dos padrões para os quais a RNA não foi treinada (tipicamente *imagens-de-fundo*). Através da compreensão geométrica que as superfícies de classificação obtidas pela RNA durante o seu treino determinam no espaço (n-dimensional) dos padrões (características extraídas das imagens) e, fundamentado em trabalhos teóricos realizados por outros autores, propomos duas abordagens para fechar adequadamente as superfícies de separação de uma RNA. A primeira abordagem é focada fortemente numa interação com o usuário. Nesta abordagem (RIFP) o usuário apresenta regiões de imagens FP à rede como exemplos de *Não-Classe* para refinar o treinamento da rede. Na segunda abordagem (NNC), é adicionado aos padrões de treinamento da RNA exemplos de *Não-Classe* gerados aleatoriamente, buscando-se desta maneira melhorar a capacidade de rejeição de padrões da RNA através da adição de “ruído”.

A realização dos experimentos com as duas abordagens propostas demonstraram que a utilização da abordagem RIFP permite à RNA determinar superfícies de separação fechadas adequadas para a tarefa de recuperação e classificação de imagens, inclusive com resultados superiores ao *matching* de imagens utilizando HC em uma base de imagens de referência. Para a abordagem NNC verificamos que esta obteve uma redução na ocorrência de regiões de FP e recuperação de imagens FP para a maioria dos casos. Mas ainda são necessários mais estudos afim de averiguar as causas e propor soluções para os casos nos quais a abordagem não apresentou resultado satisfatório.

6.1 Perspectivas

Acredita-se ter contribuído com este trabalho nos campos teórico e prático para a utilização de técnicas de IA na tarefa de recuperação e classificação de imagens. Devido ao tempo disponível e grande quantidade de variáveis que foram identificadas, nem todas as possibilidades foram suficientemente exploradas. Por isto são sugeridos alguns pontos que poderão ser mais aprofundados em trabalhos futuros:

- Executar um estudo mais detalhado do comportamento da abordagem NNC para determinar as causas dos comportamentos observados;
- Modificar a abordagem NNC para gerar padrões de *Não-Classe* que não invadam a região ocupada pelos padrões de *Classe*;
- Utilizar outro tipo de RNA, notadamente redes do tipo RBF, com o mesmo pré-processamento e verificar o seu resultado na ocorrência de regiões de FP e imagens FP recuperadas;
- Utilizar o IMatch como um mecanismo para focar a atenção, executando assim uma pré-seleção de regiões de imagens que seriam utilizadas por um segundo sistema que utilizasse cor e forma para identificar objetos;
- Aplicar o IMatch em situações práticas tais como reconhecimento de placas automotivas (em condições "reais"); localização de objetos em uma cena; auxílio para navegação de robôs;
- Verificar se a aplicação de técnicas de pré-processamento mais elaborada contribui para melhorar o fechamento de superfícies de separação.

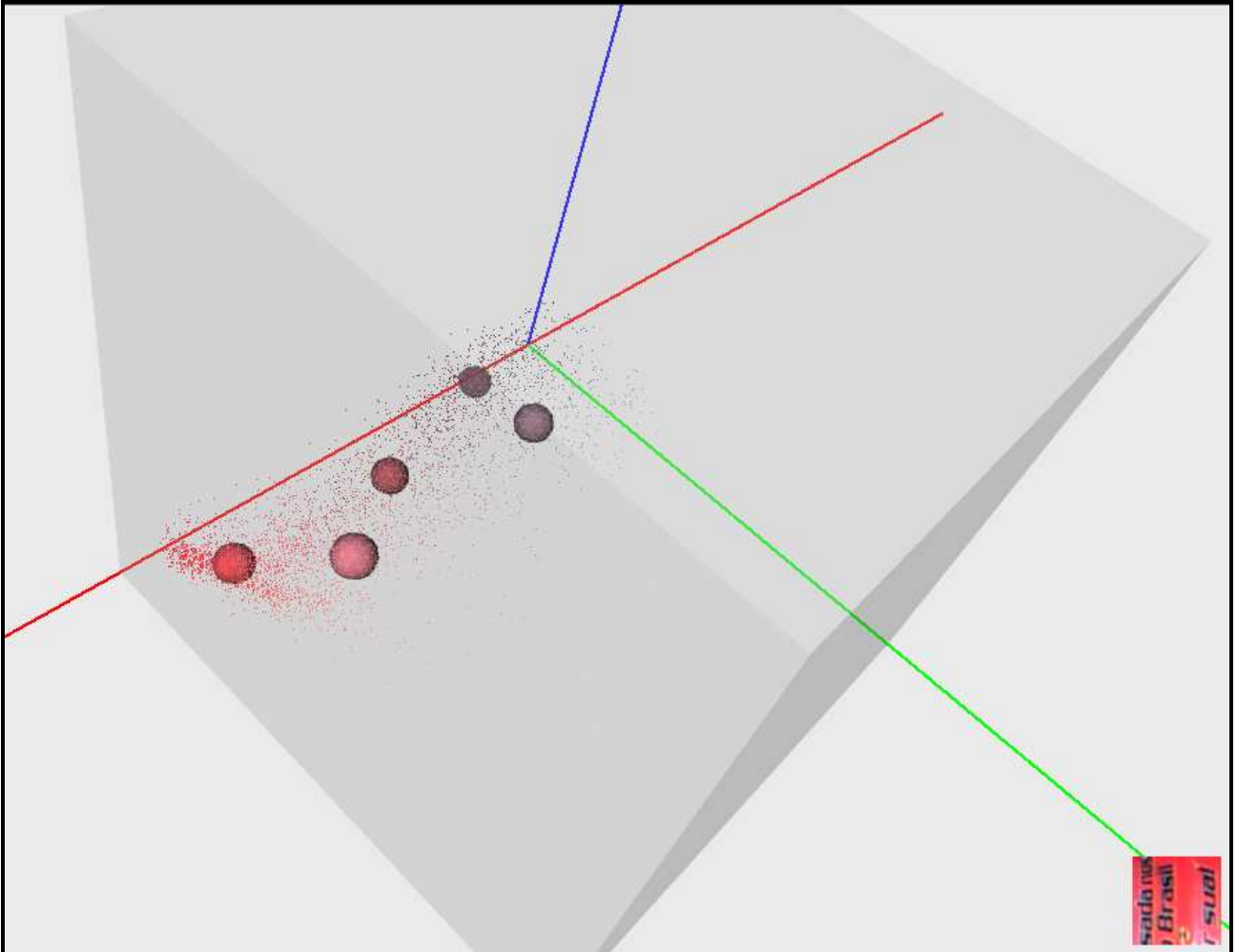
ANEXO A

Fig. Anexo A1 - Nuvem de pontos em VRML, RGB, do padrão *Classe* (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.

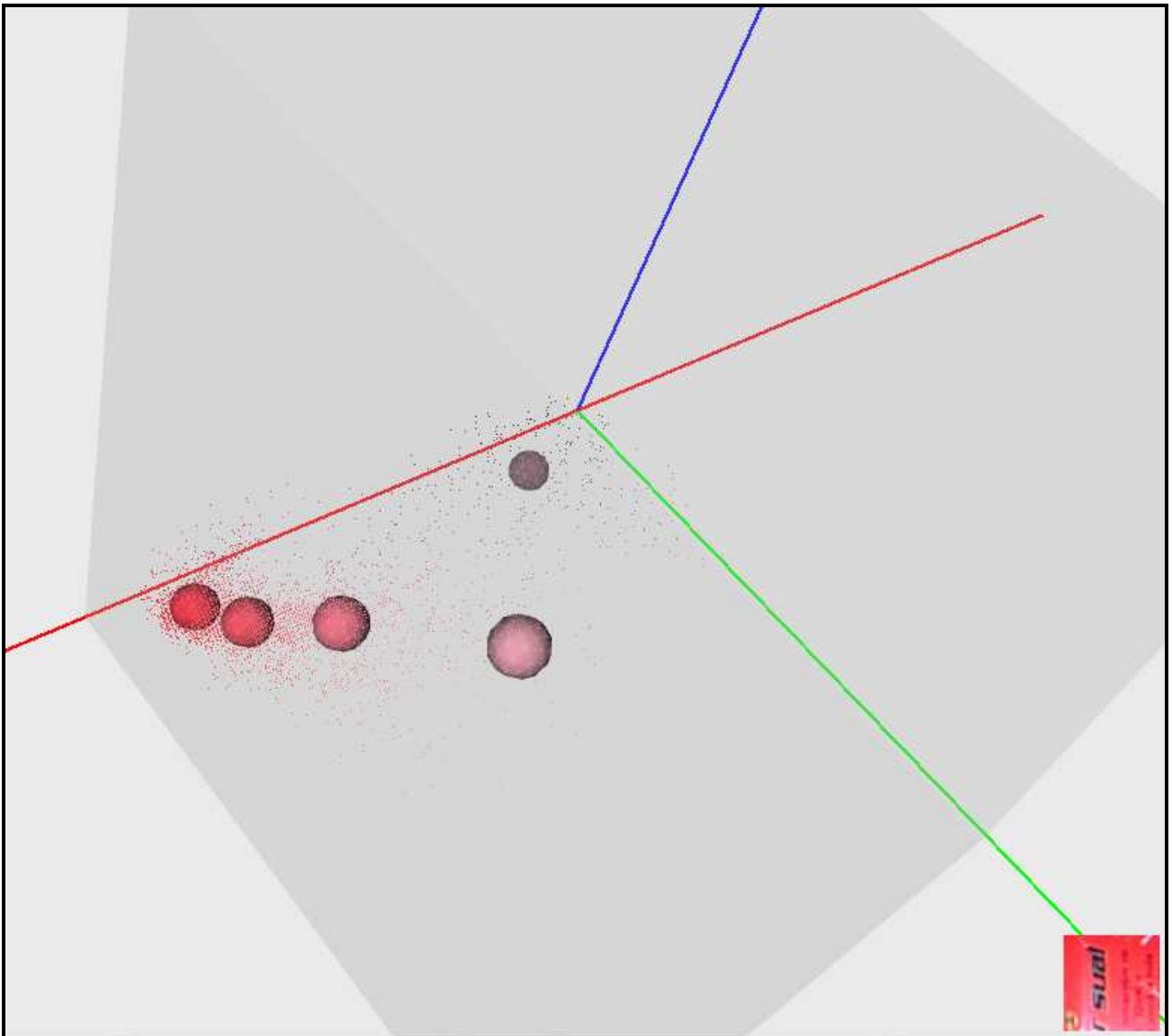


Fig. Anexo A2 - Nuvem de pontos em VRML, RGB, do padrão *Classe* (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.

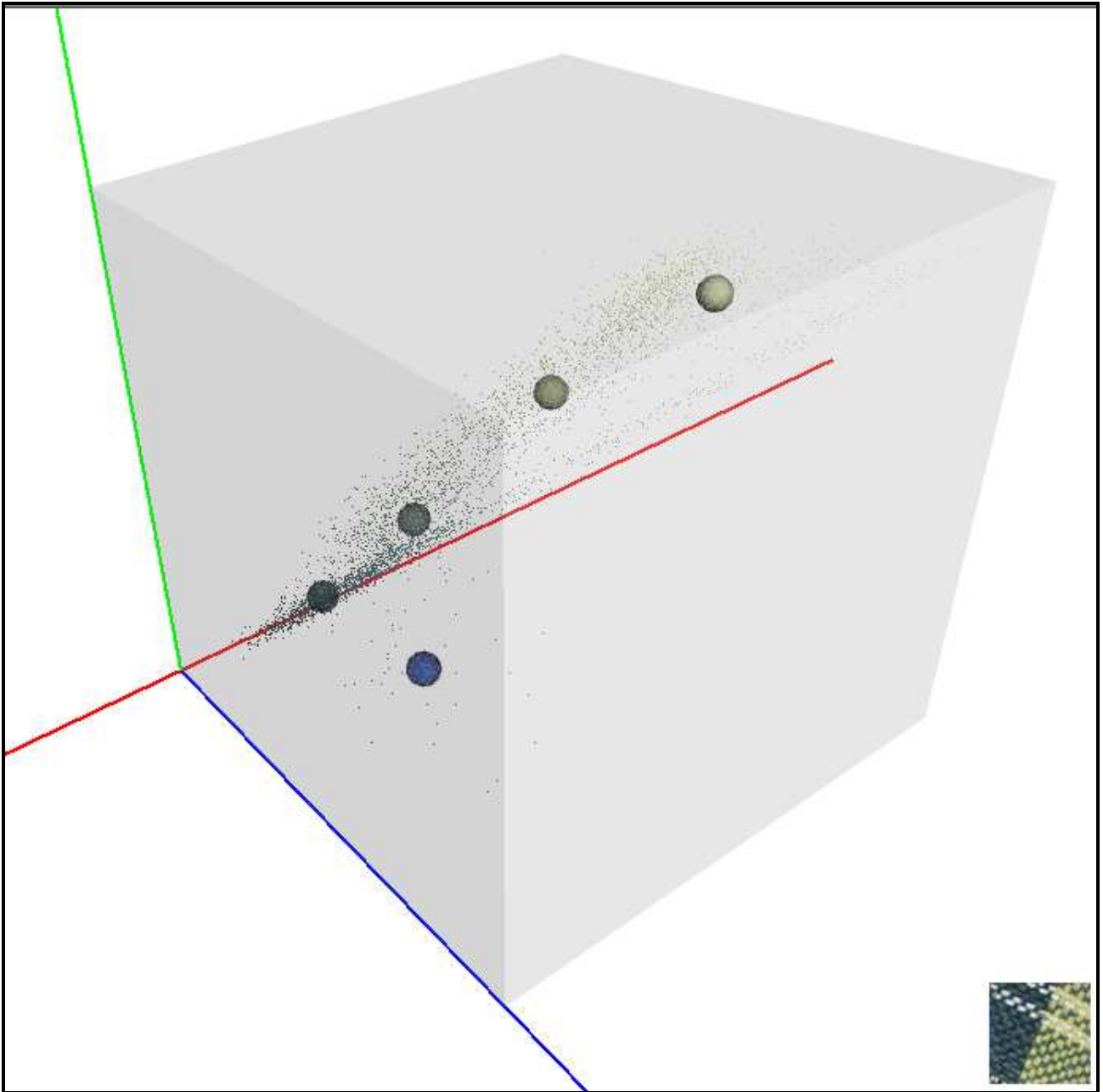


Fig. Anexo A3 - Nuvem de pontos em VRML, RGB, do padrão *Não-Classe* (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.

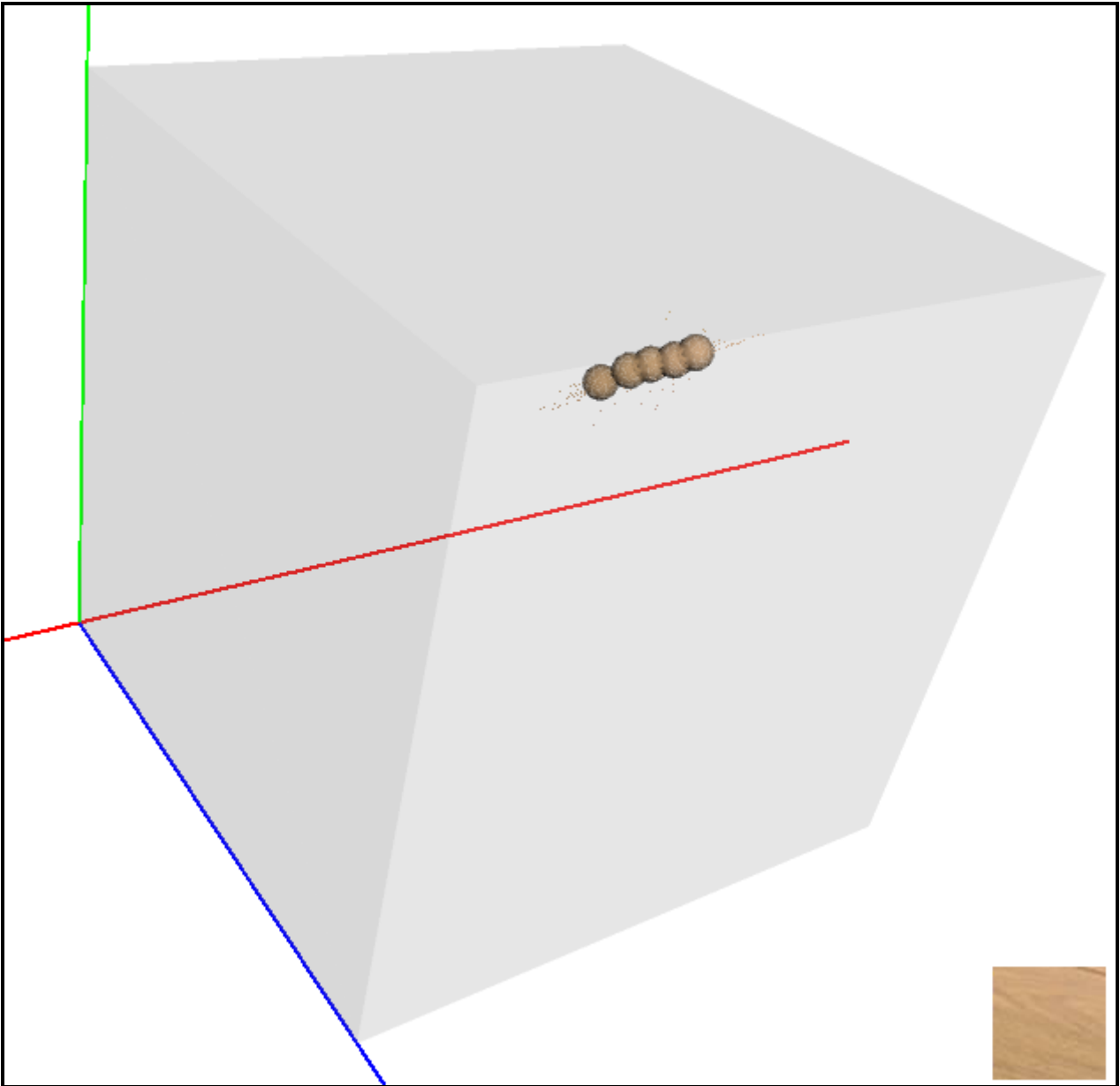


Fig. Anexo A4 - Nuvem de pontos em VRML, RGB, do padrão *Não-Classe* (canto inferior direito). Pontos da nuvem estão na cor do pixel correspondente no padrão. Esferas representam os centróides obtidos por clusterização k-Means. Cor da esfera é a cor do centróide.

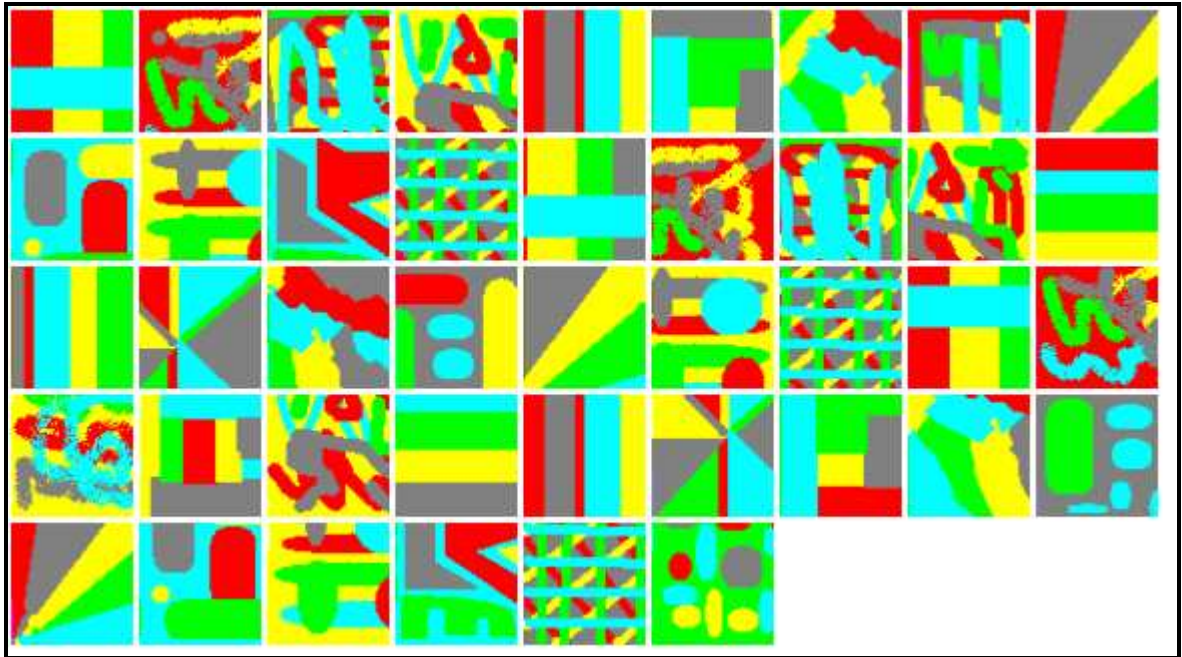


Fig. Anexo A5 – Padrões de aprendizagem da BDI B0 do tipo *Classe*.

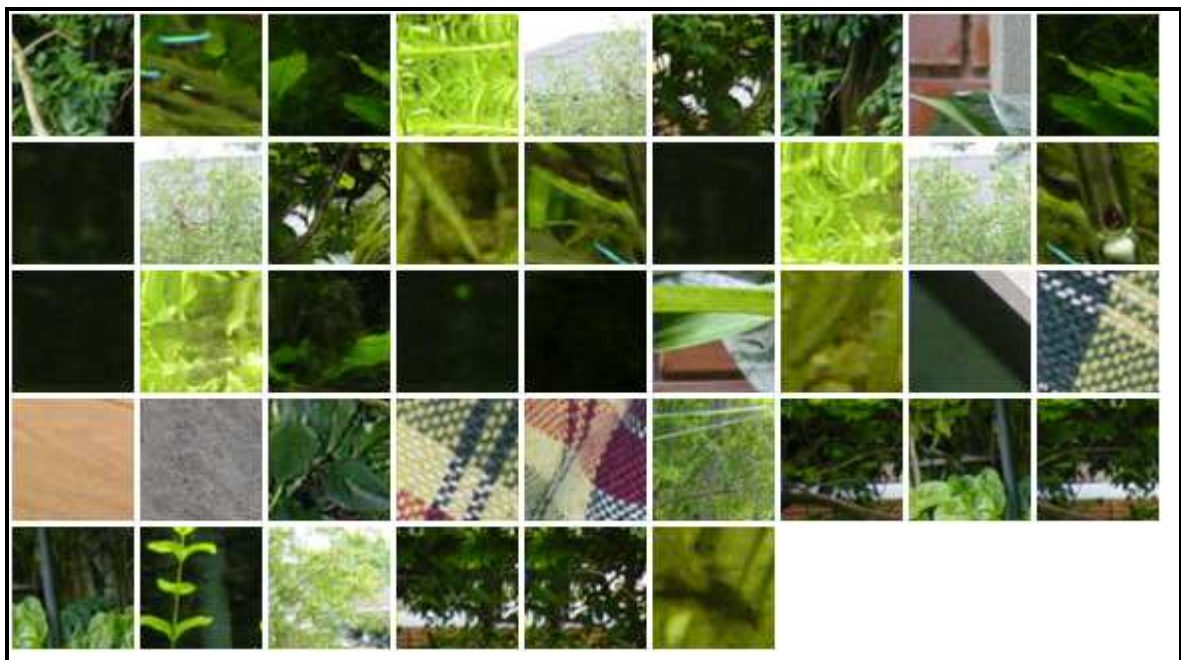


Fig. Anexo A6 - Padrões de aprendizagem da BDI B0 do tipo *Não-Classe*.

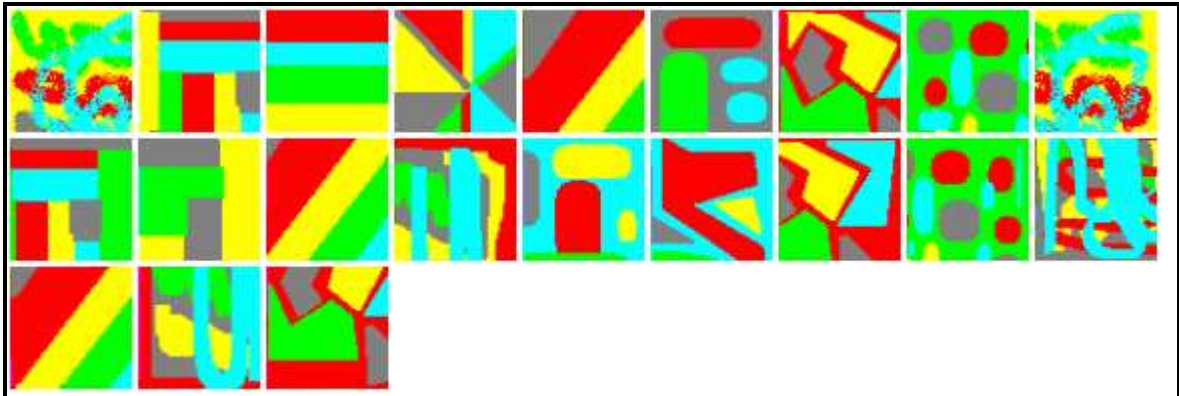


Fig. Anexo A7 - Padrões de teste da BDI B0 do tipo *Classe*.

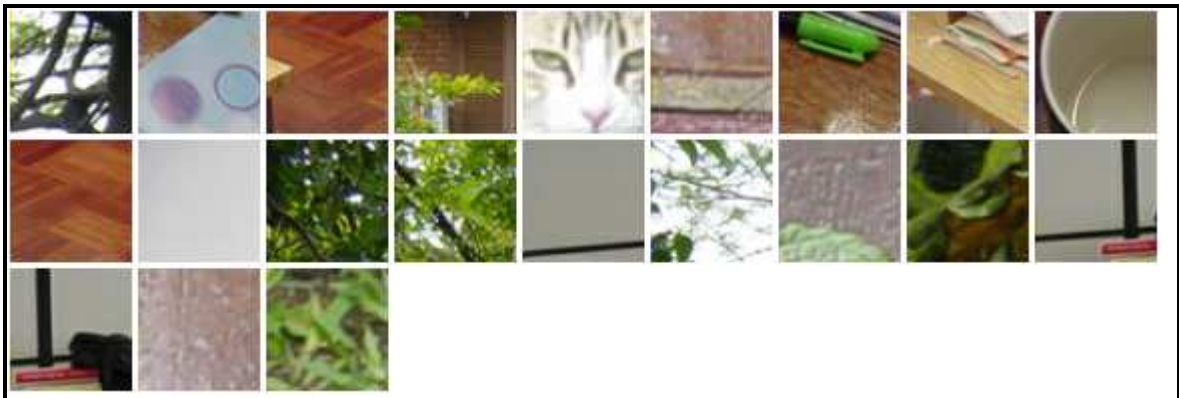


Fig. Anexo A8 - Padrões de teste da BDI B0 do tipo *Não-Classe*.

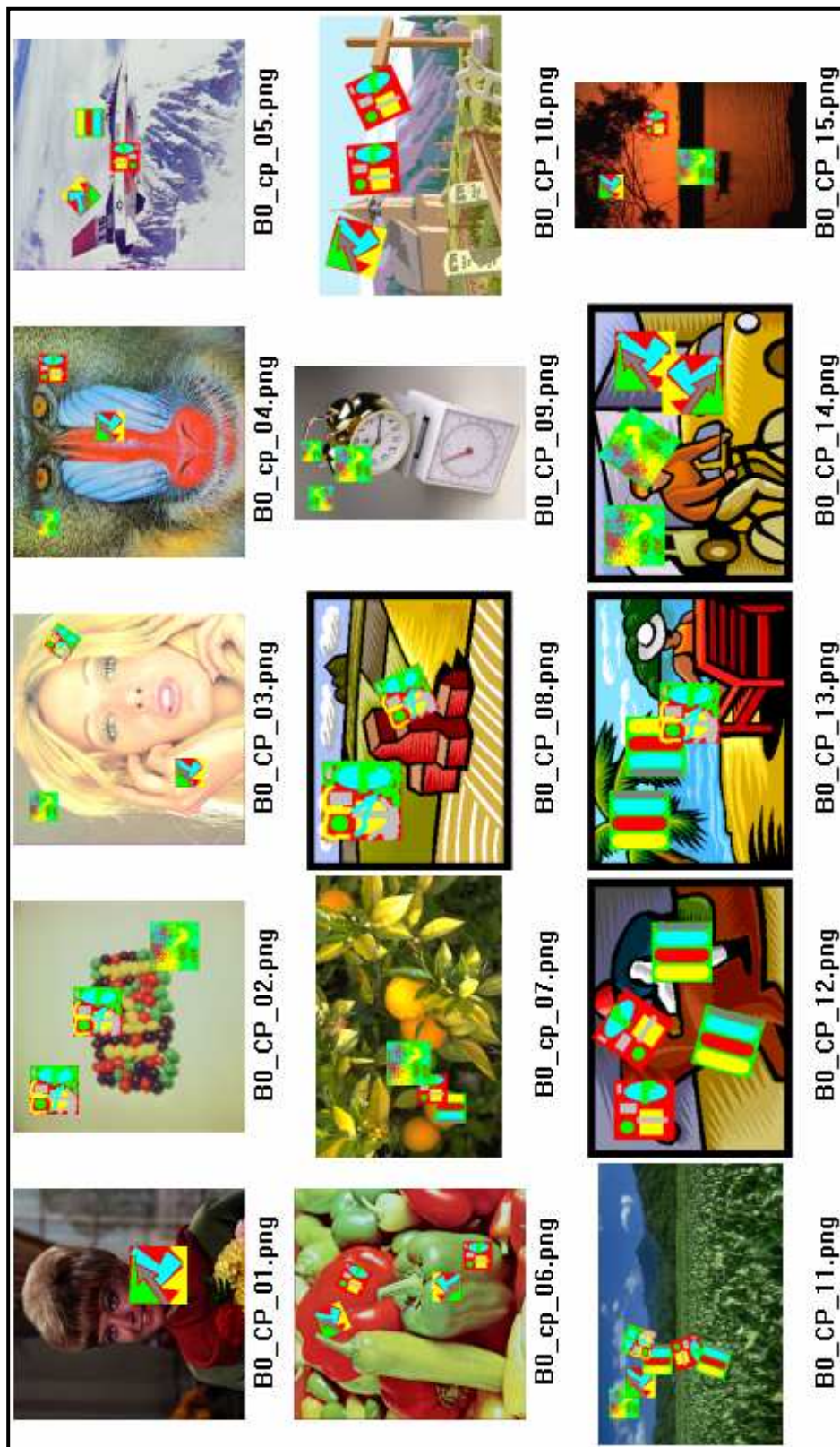


Fig. Anexo A9 – Imagens de teste para BDI B0. Todas possuem o padrão a ser procurado.



Fig. Anexo A10 – Resultado do reconhecimento executado na imagem B0_CP_09.PNG com treino inicial da RNA. BDI B0, 5 cores, T3.



Fig. Anexo A11 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com duas iterações de RIFP. BDI B0, 5 cores, T3.

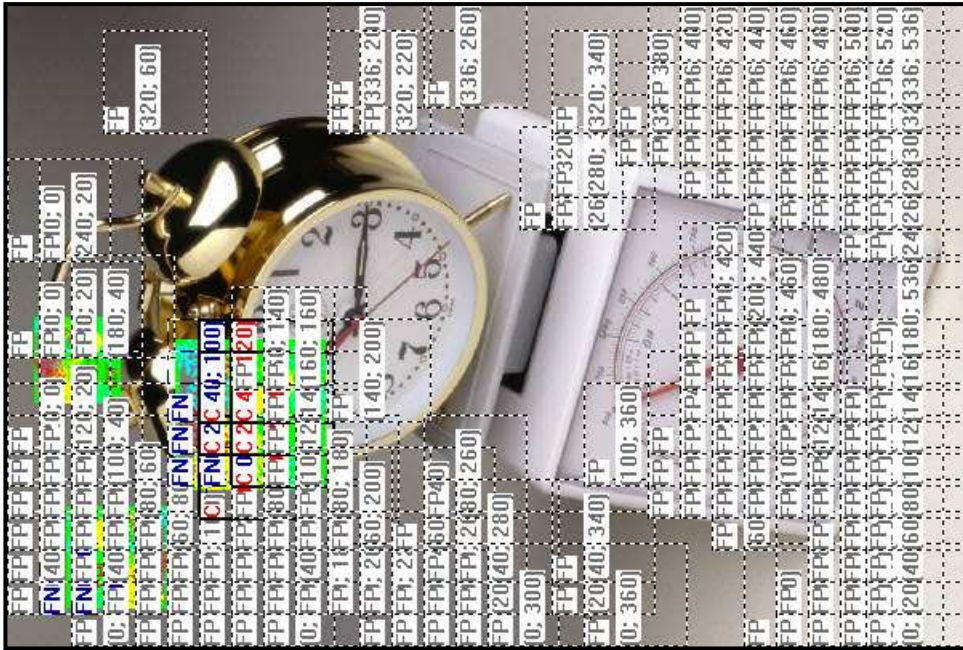


Fig. Anexo A12 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com duas iterações de NNC e aprendizado incremental. BDI B0, 5 cores, T3.



Fig. Anexo A13 - Resultado do reconhecimento executado na imagem B0_CP_09.PNG com uma iteração de NNC e sem utilizar aprendizado incremental. BDI B0, 5 cores, T3.



Fig. Anexo A14 – Objetos que compõem a base COIL-100.



Fig. Anexo A15 – Algumas regiões do *Objeto 25* da base COIL-100 utilizado como padrão de treinamento para a RNA.



Fig. Anexo A16 – Exemplos dos objetos que foram utilizados na base de pré-validação para BDI B1.

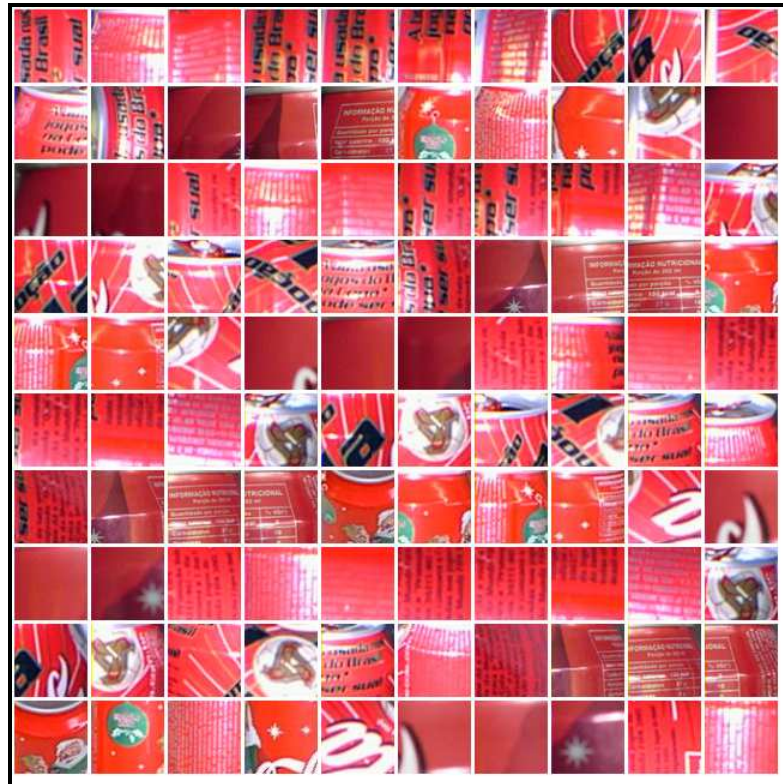


Fig. Anexo A17 - Regiões de latas de Coca-Cola utilizadas para compor os exemplos de aprendizado de *Classe* para BDI B2.

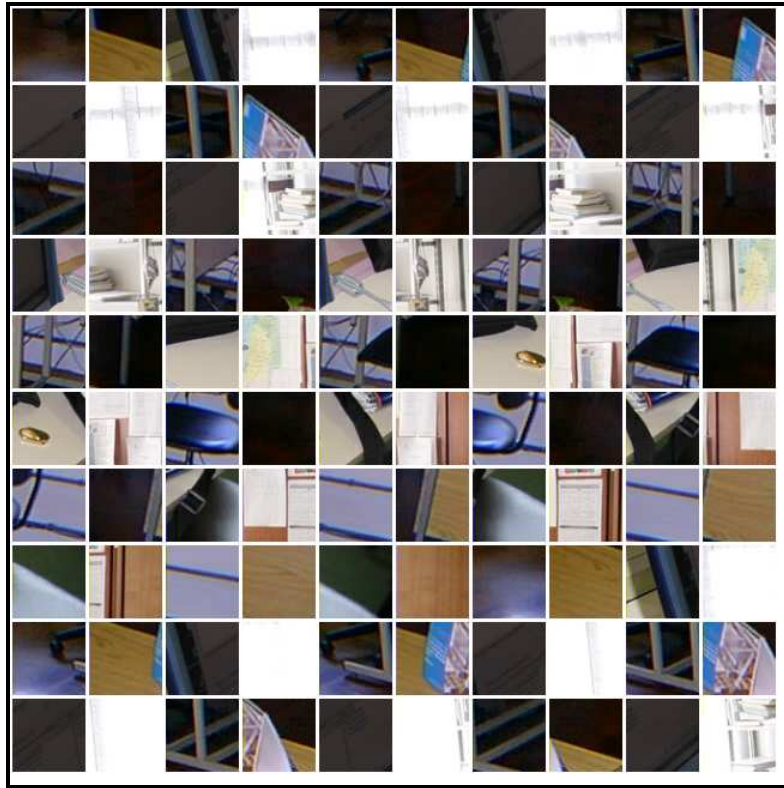


Fig. Anexo A18 – Regiões de *imagem-de-fundo* utilizadas para compor os exemplos de aprendizado de *Não-Classe* para BDI B2.



Fig. Anexo A19 - Regiões de latas de Coca-Cola utilizadas para compor os exemplos de teste de *Classe* para BDI B2.

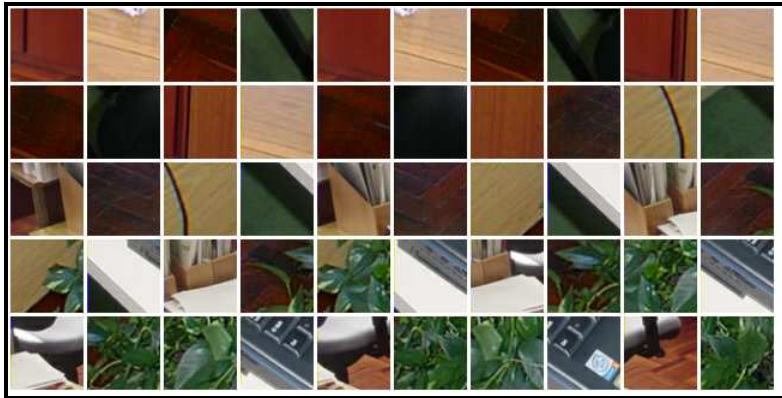


Fig. Anexo A20 - Regiões de *imagem-de-fundo* utilizadas utilizadas para compor os exemplos de teste de *Não-Classe* para BDI B2

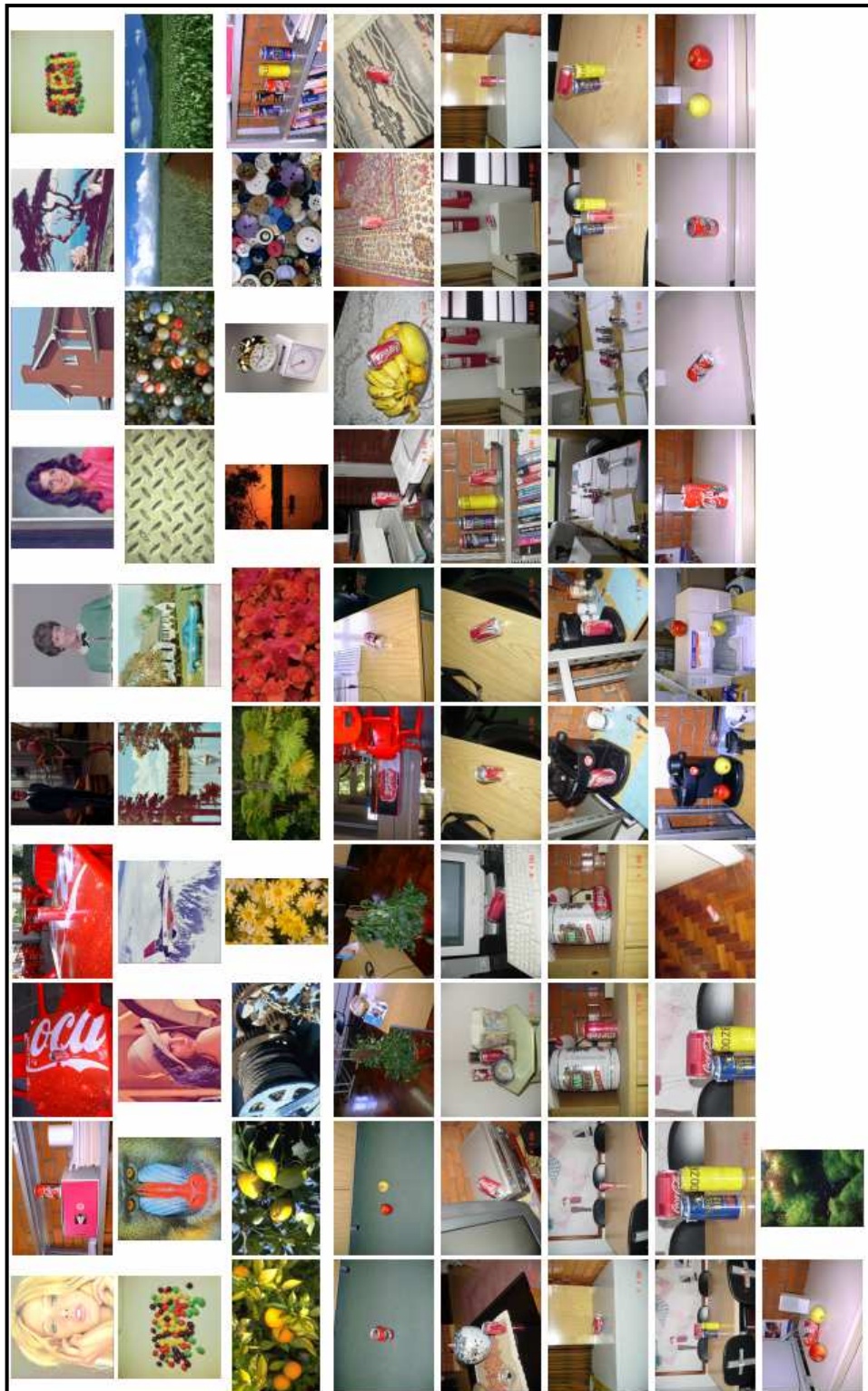


Fig. Anexo A21 – Imagens utilizadas para teste de recuperação BDI B2.



Fig. Anexo A22 – Resultado da classificação para BDI B2 com treino inicial da RNA.



Fig. Anexo A23 - Resultado da classificação para BDI B2 após segunda iteração de RIFP.



Fig. Anexo A24 - Resultado da classificação para BDI B2 após segunda iteração de NNC.

BIBLIOGRAFIA

- [ANT 02] ANTANI, S. et alii. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. **Pattern Recognition**, Amsterdã, Países Baixos, Elsevier Science B.V., n. 35, p. 945-965, 2002.
- [BIT 02] BITTENCOURT, J. R.; OSÓRIO, F. S. O Uso de Redes Neurais Artificiais na Detecção de Pele em Imagens Digitais visando o Reconhecimento de Gestos. **Anais do XI Seminário de Computação (SEMINCO)**, Universidade Regional de Blumenau, Blumenau, p. 189-202, 2002.
- [BOU 01] BOUJEMAA, N. et alii. IKONA: Interactive Generic and Specific Image Retrieval. **International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'2001)**, Rocquencourt, France, 2001.
- [BRU 99] BRUNELLI, R.; MICH, O. Histogram Analysis For Image Retrieval. **ITC-irst Technical Report 9812-03**. Instituto per la Ricerca Scientifica e Tecnologica I-38050 Povo, Trento, Itália, 1999.
- [BRU 00] BRUNELLI, R.; MICH, O. Image Retrieval by Examples. **IEEE TRANSACTIONS ON MULTIMEIA**, IEEE, v.20, n.3, p. 164-171, Set 2000.
- [CIO 01] CIOCCA, G. et alii. On Pre-Filtering with Retinex in Color Image Retrieval. **Internet Imaging II, Proceedings of SPIE**. v. 4311, p. 140-147, 2001.
- [COS 02] COSTA, M. A. et alii. Control of Generalization with a Bi-Objective Sliding Mode Control Algorithm. **In Proceedings VII Brazilian Symposium on Neural Networks SBRN 2002, IEEE**. PERNANBUCO, BRASIL, IEEE Computer Science, p. 38-43, 2002.
- [DAS 97] DAS, M; RISEMAN, E.; DAPER, B. FOCUS: Searching for Multi-coloured Objects in a Diverse Image Database. **IEEE Proceedings in Computer Vision and Pattern Recognition**. p. 756-761, 1997.
- [DAY 01] DAY, N.; MARTÍNEZ, J. M. eds. Introduction to MPEG-7 (v3.0). **International Organization for Standardisation – ISO/IEC/JTC1/SC29/WG11 M4325 Coding of Moving Pictures and Audio**. Sidnei, Jul 2001.
- [DRI 01] DRIMBAREAN, A.; WHELAN, P. F. Experiments in Colour Texture Analysis. **Pattern Recognition Letters**. Amsterdã, Países Baixos, Elsevier Science B.V., n. 22, p. 1161-1167, 2001.
- [FAH 88] FAHLMAN, S. Faster-Learning Variations on Back-Propagation: An Empirical Study. **Proceedings, 1988 Connectionist Models Summer School**. Morgan-Kaufmann, Los Altos CA, 1988.
- [FAH 90] FAHLMAN, S E.; LEBIERE, C. The Cascade-Correlation Learning Algorithm. **Advances in Neural Information Processing Systems 3**. D. S. Touretzky (ed.), Morgan Kaufmann, Los Altos CA, p. 190-196, 1990.

- [FAL 94] FALOUTSOS, C. et alii. Efficient and Effective Querying by Image Content. **Journal of Intelligent Information Systems**. v.3, n. 3/4 , p. 231-262, 1994.
- [FOL 90] FOLEY, J. D et alii. Computer Graphics: Principles and Practice. **Addison-Wesley Publishing Company, Inc.** Reading, Massachusetts, 2.ed., p. 1174, 1992.
- [GES 02] GESUALDI, A. da R. et alii. Character Recognition in Car License Plates Based on Principal Components and Neural Processing. **In Proceedings VII Brazilian Symposium on Neural Networks SBRN 2002, IEEE**. PERNANBUCO, BRASIL, IEEE Computer Science, p. 206-211, 2002.
- [GEV 99] GEVERS, T.; SMEUDERS, A. W. M. Color-based Object Recognition. **Pattern Recognition**. Amsterdã, Países Baixos, Elsevier Science B.V., n. 32, p. 453-464, 1999.
- [GOM 94] GOMES, J.; VELHO, L. Computação Gráfica: Imagem. **IMPACTA/SBM**, Rio de Janeiro, Brasil p. 424, 1994.
- [GOR 98] GORI, M.; SCARSELLI, F. Are Multilayer Perceptrons Adequate for Pattern Recognition and Verification ? **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 20, n. 11, Nov. 1998.
- [HAY 01] HAYKIN, S. Redes Neurais: Princípios e Prática. **BOOKMAN**, Porto Alegre, 2 ed., 2001.
- [HEC 82] HECKBERT, P. S. Color Image Quantization for Frame Buffer Display. **ACM Computer Graphics (ACM SIGGRAPH '82 Proceedings)**. v. 16, n. 3, p. 297-307, 1982.
- [HUA 97] HUANG, J. et alii. Image Indexing Using Color Correlgrams. **Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)**, Porto Rico, IEEE, p.762-768, Jun. 1997.
- [IKO 02] INRIA-IMEDIA Project Research Team; IKONA. <http://www-rocq.inria.fr/imedia/ikona/index.html>. (visitado em 15 de maio de 2002).
- [JAI 95] JAIN, R.; et alii. Machine Vision. **McGraw-Hill**, Singapore, 1995.
- [JAC 00] JACOBS, D. W. et alii. Classification with Nonmetric Distances: Image Retrieval and Class Representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. v. 22, n. 6, Jun. 2000.
- [KAY 99] KARYPIS, G. et alii. Chameleon: Hierarchical Clustering Using Dynamic Modeling. **IEEE Computer**. v.32, n. 8, p. 68-75, 1999.
- [KOH 87] KOHONEN, T. Self-organization and Associative Memory. **Springer Series in Information Sciences**, Springer-Verlag Berlin Heidelberg, 2 ed., 1987.
- [KRO 95] KRÖSE, B. et alii. An Introduction to Neural Networks. **Universidade de Amsterdã**, Amsterdã, Países Baixos, 8 ed., Nov.1996.

- [MCG 01] MCGUIRE, P.; D'ELEUTERIO, G. M.T. Eigenpixels and a Neural-Network Approach to Image Classification. **IEEE Transactions on Neural Networks**, v. 12, n. 3, Mai. 2001.
- [MIT 97] MITCHELL, T. M. Machine Learning. **WCB/McGraw-Hill**, Boston-Massachusetts, 1997.
- [MIN 96] MINKA, T. An Image Database that Learns from User Interaction. **TR. #365, MIT Media Laboratory – Vision and Modeling Group**. 1996.
- [MOS 01] MOSCHETTA, E. et alii. Reconhecedor de Imagens Concorrente. **SCIENTIA**. São Leopoldo, v. 12, n.1, p.11-25, 2001.
- [MOS 02] MOSCHETTA, E. et alii. Reconhecedor de Imagens Usando Técnicas de Alto Desempenho. **II ERAD – Escola Regional de Alto Desempenho**. São Leopoldo, Anais, 2002.
- [NAS 98] NASTAR, C. et alii. Surfimage: A Flexible Content-Based Image Retrieval System. **International Multimedia Conference Proceedings of the sixth ACM International Conference on Multimedia**. Bristol, Reino Unido, p. 339-344, 1998.
- [NEN 96] SAMEER, A. N. et alii. Columbia Object Image Library (COIL-100). **Department of Computer Science – Columbia University, TR. #CUCS-006-96**. New York, 1996.
- [OSO 99] OSÓRIO, F. S.; AMY, B. INSS: A hybrid system for constructive machine learning. **Neurocomputing**. Amsterdã, Países Baixos, Elsevier Science B.V., v. 28, p. 191-205, 1999.
- [PAS 96] PASS, G. et alii. Comparing Images Using Color Coherence Vectors. **ACM Multimedia**. p. 65-73, 1996.
- [PAS 00] PASCHOS, G. Fast Color Texture Recognition Using Chromaticity Moments. **Pattern Recognition Letters**. Amsterdã, Países Baixos, Elsevier Science B.V., n. 21, p. 837 – 841, 2000.
- [PEN 94] PENTLAND, A. et alii. Photobook: Content-Based Manipulation of Images Databases. **SPIE Storage and Retrieval Image and Video Database II**. San Jose, n.2185, Fev. 1994.
- [RUI 97] RUI, Y. et alii. Image Retrieval: Past, Present, and Future. **International Symposium on Multimedia Information Processing**, Taipei, Dez. 1997.
- [RUS 95] RUSSEL, S. J.; NORVIG, P. Artificial Intelligence: a modern approach. **Prentice-Hall, Inc.** 1995.
- [SCH 01] SCHETTINI, R. et alii. A Survey for Methods for Colour Image Indexing and Retrieval in Image Databases. **Color Imaging Science: Exploiting Digital Media**, R. Luo, L. MacDonald eds., J. Wiley, 2001.
- [SEB 00] SEBE, N. et alii. Toward Improved Ranking Metrics. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. v.22, n. 10, Out. 2000.

- [SEB 01] SEBE, N.; LEW, M. S. Color-Based Retrieval. **Pattern Recognition Letters**, Amsterdã, Países Baixos, Elsevier Science B.V., n. 22, p. 223-230, 2001.
- [SIQ 02] SIQUEIRA, M. L. Medidas de Estruturas Cardíacas Fetais Através de Imagens Ecocardiográficas Segmentadas. **Porto Alegre: PPGC DA UFRGS**, Porto Alegre, p. 131, 2002. (Dissertação de Mestrado)
- [SMI 95] SMITH, J. R.; CHANG, S. Automated Image Retrieval Using Color and Texture. **Columbia University Technical Report TR# 414-95-20**, Jul. 1995.
- [SMI 96] SMITH, J. R.; CHANG, S. Tools and Techniques for Color Image Retrieval. **IS&T/SPIE In Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV**, San Jose, CA. v. 2670, Fev. 1996.
- [SMI 97] SMITH, J. R.; CHANG, S. Querying by Color Regions Using VisualSEEK Content-Based Visual Query System. **Intelligent Multimedia Information Retrieval IJCAI**, editor M.T. Maybury, 1997.
- [STR 95] STRICKER, M.; ORENGO, M. Similarity of Color Images. **Storage and Retrieval for Image and Video Database – SPIE**. San Jose, 1995.
- [THE 98] THEODORIDAS, S.; KOUTROUMBAS, K. Pattern Recognition. **Academic Press**, p. 351-495, 1998.
- [TOD 98] TODT, V. Avaliação do Desempenho de Classificadores Neurais para Aplicações em Sensoriamento Remoto. **Porto Alegre: UFRGS**, Porto Alegre, p. 233, 1988. (Dissertação de Mestrado)
- [TOD 02] (DIVÉRIO) TODT, V. et alii. A Neural Approach to Classification of Satellite Images. **Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP'02)**. IEEE, Fortaleza, Brasil, p.413, Out. 2002.