

Aprendizagem e Recuperação de Imagens utilizando Mapas Auto-Organizáveis e Representação Log-Polar

Luana Bezerra Batista

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Informática da Universidade Federal de Campina Grande como parte dos
requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Modelos Computacionais e Cognitivos

Herman Martins Gomes

Orientador

Campina Grande, Paraíba, Brasil

©Luana Bezerra Batista, Fevereiro de 2004

UFCG - BIBLIOTECA - CAMPUS I

847	28-10-04
-----	----------

BATISTA, Luana Bezerra
B333A

Aprendizagem e Recuperação de Imagens utilizando Mapas Auto-Organizáveis e Representação Log-Polar.

Dissertação de Mestrado, Universidade Federal de Campina Grande, Centro de Ciências e Tecnologia, Coordenação de Pós-Graduação em Informática, Campina Grande, Paraíba, Março de 2004.

101 p. Il.

Orientador: Herman Martins Gomes

Palavras Chave:

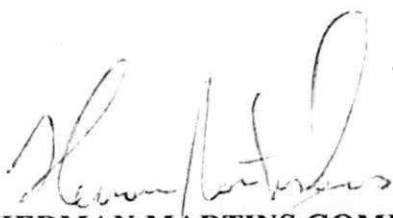
1. Redes Neurais
2. Aprendizagem Não-Supervisionada
3. Recuperação de Imagens Baseada em Conteúdo
4. Invariância a Orientação e Escala

CDU - 007.52

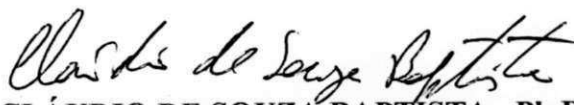
**“APRENDIZAGEM E RECUPERAÇÃO DE IMAGENS UTILIZANDO
MAPAS AUTO-ORGANIZÁVEIS E REPRESENTAÇÃO LOG-POLAR”**

LUANA BEZERRA BATISTA

DISSERTAÇÃO APROVADA EM 19.02.2004



PROF. HERMAN MARTINS GOMES, Ph.D
Orientador



PROF. CLÁUDIO DE SOUZA BAPTISTA, Ph.D
Examinador



PROF. JOÃO MARQUES DE CARVALHO, Ph.D
Examinador

CAMPINA GRANDE – PB

Agradecimentos

A Deus, por ter me dado forças para realizar este trabalho.

À minha família e a Jorge, por terem me apoiado em todo o período em que estive em Campina Grande.

Ao professor Herman, que orientou este trabalho com paciência e dedicação.

A Alisson e Fabian, por terem colaborado na implementação do sistema.

Ao prof. João Marques, por ceder o laboratório LAPS/DEE para realização de alguns experimentos.

A Bruno (CIN/UFPE), por ter colaborado com algumas bibliografias.

Aos colegas do Mestrado e do LIA, pelos incentivos e momentos de descontração.

A Aninha e Zeneide, que sempre se mostraram dispostas a me auxiliar.

Aos membros da banca examinadora, pelas críticas e sugestões que contribuíram para o enriquecimento deste trabalho.

A todos que fazem a COPIN.

A CAPES, que apoiou financeiramente este trabalho.

A todos que de alguma forma contribuíram para a conclusão deste trabalho.

Resumo

A cada dia, um número crescente de organizações vem coletando e armazenando uma grande quantidade de imagens digitais. Além disso, imagens também vêm sendo massivamente adicionadas à World Wide Web. Portanto, a estruturação dessas informações, de forma a permitir uma recuperação eficiente, é de fundamental importância. Nos primeiros sistemas de recuperação de imagens, a indexação era feita a partir de palavras-chave. No entanto, com o rápido crescimento das coleções de imagens digitais, dois problemas com esse tipo de abordagem foram evidenciados: (i) a vasta quantidade de trabalho requerido na anotação manual das imagens e (ii) a subjetividade humana de percepção. Dessa forma, a partir da década de 1990, surgiu o conceito de Recuperação de Imagens Baseada em Conteúdo (RIBC), que se caracteriza pela indexação automática de imagens a partir de suas próprias características visuais, como cor, textura e forma. Muitos métodos de indexação baseados em B-Trees vêm sendo utilizados em RIBC, com o objetivo de reduzir o espaço de busca. No entanto, tais métodos são geralmente ineficientes ao lidar com altas dimensões. Além disso, as técnicas utilizadas para extrair características visuais podem causar a perda de informações valiosas da imagem. Nesta dissertação, investigamos o uso de Redes Neurais (mais especificamente os Mapas Auto-Organizáveis) para classificar, indexar e recuperar imagens nesse tipo de problema. A representação de imagem utilizada (log-polar) facilita o reconhecimento de imagens de forma independente de orientação e escala, além de permitir uma compactação da imagem original. Os resultados experimentais obtidos (no reconhecimento de objetos e imagens genéricas) mostraram que a combinação de Mapas Auto-Organizáveis com a representação log-polar é uma estratégia promissora para classificação de imagens. Assim, um protótipo de um sistema de RIBC foi implementado com a estratégia proposta e aplicado a dois estudos de caso em recuperação de imagens da Web.

Abstract

Everyday, a growing number of organizations is collecting and storing a large amount of digital images. Furthermore, images have also been massively added to the World Wide Web. Therefore, structuring this information, in order to allow efficient retrieval, is a very important task. In the initial image retrieval systems, an image indexing scheme based on keywords was used. However, due to the fast growth of the digital image collections, two problems became evident: (i) the vast amount of labor required in manual image annotation and (ii) the subjectivity of human perception. Thus, in the 1990's, the idea of Content Based Image Retrieval (CBIR) emerged, which is characterized by automatically indexing images using their own visual features, such as color, texture and shape. Many indexing methods based on B-Trees have been used in CBIR, in order to reduce the search time. However, these methods are generally inefficient when working with high dimensions. Moreover, the feature extraction techniques can cause the loss of valuable image information. In this work, we investigate the use of Neural Networks (more specifically, the Self-Organizing Maps) to classify, index and retrieve image in this class of problems. The image representation (log-polar) adopted in this work helps recognizing objects in a way that is independent of orientation and scale, besides being more compact than the original input image. The experimental results (related to individual objects and arbitrary image recognition) showed that the combination of Self-Organizing Maps with the log-polar representation is a promising strategy for image classification. Thus, a CBIR system prototype was built using the proposed strategy and applied to two case studies of image retrieval from the Web.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Descrição do Problema	2
1.3	Objetivos e Relevância	4
1.4	Estrutura da Dissertação	5
2	Mapas Auto-Organizáveis	6
2.1	Introdução às Redes Neurais	6
2.2	Breve Histórico das Redes Neurais	9
2.3	Face Biológica	10
2.4	Treinamento	13
2.5	Mapas de Kohonen	13
2.5.1	Algoritmo de Treinamento	14
2.5.2	Vizinhança Topológica	16
2.6	Mapas Hierárquicos	17
2.7	Conceito de Mapa Contextual	19
2.8	Incorporando Invariâncias	21
2.9	Considerações Finais	22
3	Recuperação de Imagens Baseada em Conteúdo	23
3.1	Extração de Características	23
3.1.1	Cor	24
3.1.2	Forma	24
3.1.3	Textura	28

3.2	Indexação em SRIBCs	29
3.3	Considerações finais	32
4	Trabalhos Relacionados	33
4.1	Indexação com SOM	33
4.1.1	Extração de Características	34
4.1.2	Construção de Histogramas	35
4.1.3	Indexação e Recuperação	35
4.1.4	Experimentos	36
4.1.5	Discussão	37
4.2	Indexação com H-SOM	37
4.2.1	Agrupamento	38
4.2.2	Projeção	38
4.2.3	Recuperação	38
4.2.4	Experimentos	41
4.2.5	Discussão	42
4.3	Indexação com TS-SOM	42
4.3.1	Funcionamento dos sistemas	43
4.3.2	Experimentos	43
4.3.3	Discussão	45
4.4	Considerações Finais	45
5	Sistema Proposto	47
5.1	Representação Log-Polar	47
5.2	Funcionamento do Sistema	50
5.3	Considerações Finais	54
6	Experimentos e Resultados	56
6.1	Incorporando Invariâncias por Treinamento (Experimento 1)	56
6.2	Incorporando Invariâncias por Espaço de Características	60
6.2.1	Experimento 2	60
6.2.2	Experimento 3	62

6.3	Comparando a Performance de SOMs com GH-SOMs (Experimento 4) . . .	66
6.4	Comparando a Representação Log-Polar com Características de Forma e Intensidade de Cor (Experimento 5)	70
6.5	Aplicando o Sistema Proposto na Web	72
6.5.1	Estudo de Caso 1	73
6.5.2	Estudo de Caso 2	76
6.6	Considerações Finais	81
7	Conclusão	83
7.1	Sumário da Dissertação	83
7.2	Contribuições	85
7.3	Trabalhos Futuros	86
7.3.1	SOMs <i>com consciência</i> e LVQ	87
7.3.2	Aprendizagem Incremental	88
7.3.3	Representação Log-Polar <i>versus</i> Texturas	88
7.3.4	Bloqueio de Conteúdos Proibidos	88
7.3.5	Sistemas de Apoio à Decisão	89
A	Amostras de imagens indexadas	97

Lista de Tabelas

6.1	Rótulos dos padrões de treinamento.	58
6.2	Agrupamentos por orientação.	60
6.3	Padrões de teste com suas variações vencedoras.	63
6.4	Taxas de acerto.	63
6.5	BMUs e classificações dos padrões de teste.	67
6.6	Taxas de classificação.	67
6.7	Taxas de classificação do GH-SOM e SOM.	69
6.8	Taxas de classificação do GH-SOM1 e GH-SOM2.	72
6.9	Taxas de acerto do Estudo de Caso 1.	75
6.10	Matriz de Confusão do Estudo de Caso 1.	76

Lista de Figuras

2.1	Rede <i>Feedforward</i>	7
2.2	Rede <i>Feedback</i>	8
2.3	Mapa citoarquitetural do córtex cerebral. Os números do mapa indicam as diferentes áreas sensoriais do córtex cerebral.	11
2.4	Função Chapéu Mexicano.	12
2.5	Mapa de Kohonen Bidimensional.	14
2.6	Arquitetura de um GH-SOM	18
2.7	Inserção de neurônios.	19
2.8	Geração de um mapa contextual. DE = Distância Euclidiana.	20
2.9	Arquitetura de um sistema de espaço de características invariantes.	21
3.1	Histograma com 256 níveis de cinza.	24
3.2	Exemplo de binarização. (a) Imagem em tons de cinza. (b) Imagem binária.	25
3.3	Exemplo de extração de bordas. (a) Imagem em tons de cinza. (b) Imagem de bordas.	25
3.4	Contorno em um plano complexo.	26
3.5	Representação de um vetor de características no espaço 3D.	29
3.6	Particionamento do espaço sem MBRs (a) e com MBRs (b).	30
3.7	Árvore com sobreposição de MBRs.	31
4.1	Intersecção entre segmentos.	34
4.2	Recuperação de logotipos.	37
4.3	H-SOM proposto por Zhang e Zhong.	38
4.4	Diagrama de blocos do PicSOM.	44
4.5	Procedimento utilizado nos testes.	45

5.1	Estrutura da máscara retinal utilizada.	48
5.2	Transformação log-polar.	49
5.3	Exemplo de transformação log-polar. (a)Imagem original. (b)Imagem retinal. (c)Imagem log-polar.	50
5.4	Mapa contextual representando quatro classes.	51
5.5	Estratégia de classificação/indexação.	51
6.1	Classes de objetos do Experimento 1.	57
6.2	Imagens de treinamento da Classe 4.	57
6.3	Mapa Contextual. Cor branca = Classe 1 (ANACIN); cor cinza-claro = Classe 2 (CARRO); cor cinza-escuro = Classe 3 (FORMA); cor preta = Classe 4 (PATO).	58
6.4	Mapa Contextual enfatizando os agrupamentos por orientação. Cor branca = 0°; cor cinza-claro = 90°; cor cinza-médio = 180°; cor cinza-escuro = 30°; cor preta = -30°.	59
6.5	Imagens de treinamento do Experimento 2.	61
6.6	Mapa Contextual ilustrando 4 classes. Cor branca = Classe 1 (ANACIN); cinza-claro = Classe 2 (CARRO); cor cinza-escuro Classe 3 (FORMA); cor preta Classe 4 (PATO).	62
6.7	Imagens de treinamento do Experimento 3.	64
6.8	Imagens de teste da Classe 1.	64
6.9	Imagens de teste da Classe 2.	65
6.10	Imagens de teste da Classe 3.	65
6.11	Mapa Contextual ilustrando 3 classes. Cinza-claro = Classe 1 (CARRO); cor cinza-escuro Classe 2 (CASTELO); cor preta Classe 3 (FACE).	66
6.12	Alguns exemplos da base MNIST.	68
6.13	Classes de treinamento do Experimento 5.	71
6.14	Arquitetura do SRBCW.	73
6.15	Exemplo de recuperação de imagens.	75
6.16	Interface do SRBCW.	77
6.17	Recuperação com similaridade 0.003 e sem variações.	78

6.18	Amostra das imagens indexadas pela BMU 30.	79
6.19	Recuperação com similaridade 0.03 e variações.	80
6.20	Imagens indexadas pela BMU 9.	81
6.21	Classe desconhecida.	82
A.1	Amostra de imagens indexadas pela BMU 5.	98
A.2	Amostra de imagens indexadas pela BMU 13.	99
A.3	Amostra de imagens indexadas pela BMU 16.	100
A.4	Amostra de imagens indexadas pela BMU 40.	101

Lista de Siglas e Abreviaturas

- BMU: *Best Map Unit*
- B-Tree: *Balanced Tree*
- EDM: *Estrutura de Dados Multidimensional*
- GH-SOM: *Growing Hierarchical Self-Organizing Map*
- HFM: *Hierarchical Feature Map*
- H-SOM: *Hierarchical SOM*
- KDB-Tree: *K-Dimensional Balanced Tree*
- LVQ: *Learning Vector Quantization*
- MBR: *Minimum Bounding Region*
- MR-SAR: *Multi-Resolução Simultânea Auto-Regressiva*
- RIBC: *Recuperação de Imagens Baseada em Conteúdo*
- R-Tree: *Rectangle Tree*
- SOM: *Self-Organizing Maps*
- SRIBC: *Sistemas de Recuperação de Imagens Baseada em Conteúdo*
- TDF: *Transformada Discreta de Fourier*
- TS-SOM: *Tree-Structured Self-Organizing Map*
- TV-Tree: *Telescopic Vector Tree*

Capítulo 1

Introdução

Nesta dissertação, é investigado o uso dos Mapas Auto-Organizáveis (SOM: *Self-Organizing Maps*), juntamente com a representação de imagem log-polar, para classificar, indexar e recuperar imagens genéricas. A seguir, apresentamos a motivação para o desenvolvimento de Sistemas de Recuperação de Imagens Baseada em Conteúdo (SRIBCs), as principais limitações existentes nas soluções atuais e os principais objetivos desta pesquisa. O capítulo é concluído com uma pequena descrição da estrutura da dissertação.

1.1 Motivação

A cada dia, um número crescente de organizações vem coletando e armazenando uma grande quantidade de imagens digitais. Além disso, imagens também vêm sendo massivamente adicionadas à World Wide Web. Portanto, a estruturação dessas informações, de forma a permitir uma recuperação eficiente, é de fundamental importância para muitos grupos de usuários, como jornalistas, policiais, cientistas, médicos, designers, entre outros.

Na década de 1970, a comunidade de Banco de Dados iniciou as pesquisas na área de recuperação de imagens. Os primeiros sistemas desenvolvidos eram baseados em texto, nos quais as imagens eram indexadas a partir de palavras-chave. Existem duas grandes dificuldades nessa abordagem. A primeira é a vasta quantidade de trabalho requerido na anotação manual das imagens. A segunda, resulta do rico conteúdo das imagens e da subjetividade humana de percepção [Rui et al., 1999].

A partir de 1990, com o rápido crescimento das coleções de imagens digitais - devido à

queda de preços dos equipamentos de digitalização e armazenamento [Subrahnian, 1998] - os problemas da recuperação baseada em texto tornaram-se ainda mais evidentes [Rui et al., 1999]. Desde então, a comunidade de Banco de Dados e, especialmente, a comunidade de Visão Computacional vêm desenvolvendo SRIBCs, os quais se caracterizam pela indexação de imagens utilizando suas próprias características visuais, como cor, textura, forma, etc. Alguns desses sistemas, citados por Rui *et alli* [Rui et al., 1999], são: QBIC, Virage, Retrievalware, Photobook, VisualSEEk, WebSEEk, Netra e MARS.

Para exemplificar a utilidade de um SRIBC, considere o caso de um policial investigador [Subrahnian, 1998]: a partir de uma fotografia de alguém que não se sabe a identidade, ele pode pedir a um sistema de recuperação de imagens fotografias que são similares à que ele possui, com suas respectivas identidades. O mesmo pode ser feito para identificar se um suspeito é ou não um criminoso procurado pela polícia.

Um outro exemplo prático de RIBC é o suporte a diagnósticos médicos [Rosa et al., 2002]. Ou seja, a partir de uma imagem médica de um novo paciente, são recuperadas imagens similares, às quais já foram atribuídos diagnósticos que poderão auxiliar na interpretação da imagem corrente.

Esse tipo de consulta é diferente das consultas tradicionais por duas razões: (i) a fotografia é incluída como parte da consulta e (ii) a consulta por imagens similares utiliza a noção de “casamento impreciso¹”.

1.2 Descrição do Problema

Em geral, o funcionamento de um sistema de Visão Computacional pode ser dividido nas seguintes etapas [Gonzalez and Woods, 1992]:

1. *Aquisição de Imagens*: nesta etapa, as informações visuais do ambiente são convertidas em sinais digitais através de dispositivos ou sensores ópticos;
2. *Pré-processamento*: nesta etapa, a imagem é melhorada (através de aumento de contraste, remoção de ruído, realce, etc.), com o objetivo de aumentar as chances de sucesso das etapas seguintes.

¹As imagens recuperadas não são necessariamente iguais à imagem de consulta.

3. *Segmentação*: nesta etapa, a imagem é dividida em regiões que podem constituir os diversos objetos nela presentes. A identificação de um objeto baseia-se na detecção de descontinuidades na imagem, gerando uma representação de seu contorno ou da região que ocupa.
4. *Descrição*: para cada objeto identificado, características compactas, que podem ser úteis na discriminação entre classes de objetos, são extraídas e armazenadas em um vetor de características.
5. *Reconhecimento*: nesta etapa, é realizado um processo no qual se decide a que classe pertence cada objeto (representado por seu vetor de características).
6. *Interpretação*: a interpretação semântica de um conjunto de objetos classificados no domínio de uma aplicação finaliza o processo de visão, dando lugar a uma ação. Por exemplo, um sistema pode reconhecer dígitos em uma imagem de código de barras e interpretá-los como sendo o número de acesso a um dado produto em um supermercado. A ação neste caso poderia ser a contabilização do valor do produto na compra [Falcão, 2003]. Em SRIBCs, esta etapa pode ser vista como a recuperação de imagens similares à imagem reconhecida.

Em particular, a maioria dos SRIBCs possuem as seguintes características em comum [Subrahmanian, 1998]:

1. as características visuais extraídas são geralmente cor, forma e textura;
2. durante a construção do sistema, os vetores de características são organizados em algum tipo de Estrutura de Dados Multidimensional (EDM), com o objetivo de reduzir o espaço de busca. Esta etapa é conhecida como *Indexação* e é equivalente à etapa de *Aprendizagem* em outros sistemas de Visão Computacional.
3. na etapa de *Reconhecimento*, geralmente se utiliza a distância Euclidiana como medida de dissimilaridade.

Na abordagem tradicional de Recuperação de Imagens Baseada em Conteúdo (RIBC), geralmente são utilizadas EDMs derivadas da B-Tree (Balanced Tree) [Bayer and McCreight,

1972] como estruturas de indexação. Entretanto, esse tipo de estrutura possui diversos problemas (abordados no Capítulo 3), como por exemplo, ineficiência ao lidar com altas dimensões. Além disso, as técnicas utilizadas para extrair características visuais podem causar a perda de informações valiosas da imagem. Tais limitações são, portanto, o ponto de partida desta pesquisa.

1.3 Objetivos e Relevância

O objetivo geral desta dissertação é propor e implementar uma nova estratégia de aprendizagem e recuperação de imagens, visando superar as principais limitações existentes nas técnicas tradicionais para implementação de SRIBCs.

Apesar de terem sido pouco aplicados em SRIBCs, os Mapas Auto-Organizáveis mostraram ser uma ferramenta promissora de indexação de imagens. No entanto, importantes características desse tipo de rede neural não foram exploradas de forma a melhorar a performance desses sistemas. Nesse sentido, pretendemos atingir dois objetivos específicos:

1. apresentar um estudo detalhado sobre a estrutura e funcionamento dos SOMs, possibilitando um melhor entendimento da técnica e conseqüente formulação de uma estratégia de indexação eficiente.
2. investigar o uso dos GH-SOMs (*Growing Hierarchical SOMs*) nesse tipo de problema. O GH-SOM além de ter todas as vantagens dos Mapas de Kohonen, possui uma arquitetura em forma de árvore, a qual é definida automaticamente durante a fase de treinamento.

Um outro objetivo desta pesquisa é investigar e utilizar uma representação de imagem inspirada na retina humana, alternativamente à utilização de características visuais como cor, forma e textura. Finalmente, objetivamos desenvolver um SRIBC e aplicá-lo em um problema de recuperação de imagens da World Wide Web.

A realização de experimentos e análises comparativas constituem a estratégia utilizada para a execução dos objetivos desta pesquisa. Adicionalmente, visamos estimular o desenvolvimento de novos trabalhos envolvendo RIBC e Mapas Auto-Organizáveis.

1.4 Estrutura da Dissertação

Esta dissertação está dividida em sete capítulos. No Capítulo 2, é apresentado um estudo teórico sobre os Mapas Auto-Organizáveis, um tipo de rede neural não-supervisionada inspirada nos mapas topologicamente ordenados do cérebro. Além do mapa de Kohonen - modelo mais conhecido de SOM - também é detalhado o funcionamento do GH-SOM, cuja arquitetura é composta por SOMs independentes organizados hierarquicamente.

No Capítulo 3, são apresentadas as principais características dos Sistemas de Recuperação de Imagens Baseada em Conteúdo. São discutidas algumas das técnicas mais utilizadas para extração de características visuais, como Histogramas de Cor, Descritores de Fourier, Momentos Invariantes e Wavelets. Além disso, é discutido também o funcionamento de algumas estruturas de dados tradicionalmente utilizadas na indexação de imagens (i.e., KDB-Tree, R-Tree e TV-Tree), suas principais limitações, e como os Mapas Auto-Organizáveis podem ser utilizados nesse tipo de problema.

O Capítulo 4 apresenta uma revisão bibliográfica dos principais trabalhos que propõem a utilização de Mapas Auto-Organizáveis como ferramenta de indexação em SRIBCs. Tais trabalhos são divididos de acordo com o tipo de Mapa Auto-Organizável utilizado.

No Capítulo 5, é descrita a representação de imagem log-polar, cujos funcionamento e estrutura são inspirados na retina humana. Em seguida, é proposto um SRIBC biologicamente inspirado que utiliza um GH-SOM, como estrutura de indexação, e a representação de imagem log-polar. O principal objetivo desse SRIBC é preencher lacunas existentes nos sistemas descritos no Capítulo 4.

O Capítulo 6 relata os principais experimentos realizados ao longo desta pesquisa, os quais foram fundamentais na formulação da melhor estratégia de classificação/indexação e nos testes de performance do sistema proposto. O capítulo é concluído com dois estudos de caso nos quais o sistema proposto é adaptado e utilizado na busca de imagens da Web.

O Capítulo 7 conclui a dissertação apresentando um resumo dos principais pontos estudados, as contribuições da pesquisa desenvolvida e algumas sugestões de trabalhos futuros.

Capítulo 2

Mapas Auto-Organizáveis

Este capítulo aborda as principais características e funcionalidades dos Mapas Auto-Organizáveis, um tipo de rede neural não-supervisionada inspirada nos mapas topologicamente ordenados do cérebro. Uma ênfase maior é dada aos mapas de Kohonen, um modelo simples de SOM que possibilita a compressão dos dados de entrada.

Como veremos no decorrer desta dissertação, as estruturas hierárquicas são amplamente utilizadas na indexação de imagens, e o uso de SOMs Hierárquicos nesse tipo de problema vem produzindo resultados promissores. Portanto, além do modelo de Kohonen, também é detalhado o funcionamento do GH-SOM, cuja arquitetura é composta por mapas de Kohonen independentes e organizados em forma de árvore.

2.1 Introdução às Redes Neurais

Redes Neurais são modelos computacionais inspirados no cérebro humano e amplamente utilizados em tarefas de reconhecimento de padrões. Segundo Haykin [Haykin, 2001], uma rede neural é um sistema de processamento distribuído em paralelo, de forma maciça, constituído de unidades de processamento simples (neurônios), que tem uma propensão natural para armazenar conhecimento derivado da experiência e torná-lo disponível para o uso.

As unidades de uma rede neural são organizadas em camadas e interligadas através de conexões ponderadas (sinapses). A Figura 2.1 apresenta a arquitetura básica de uma rede neural de múltiplas camadas. A primeira camada, chamada *camada de entrada*, simplesmente propaga a informação para uma *camada intermediária* (ou camada escondida) que

efetivamente realiza algum tipo de processamento. É possível que haja uma ou mais camadas intermediárias seguidas por uma *camada de saída*, de onde o resultado do processamento é obtido.

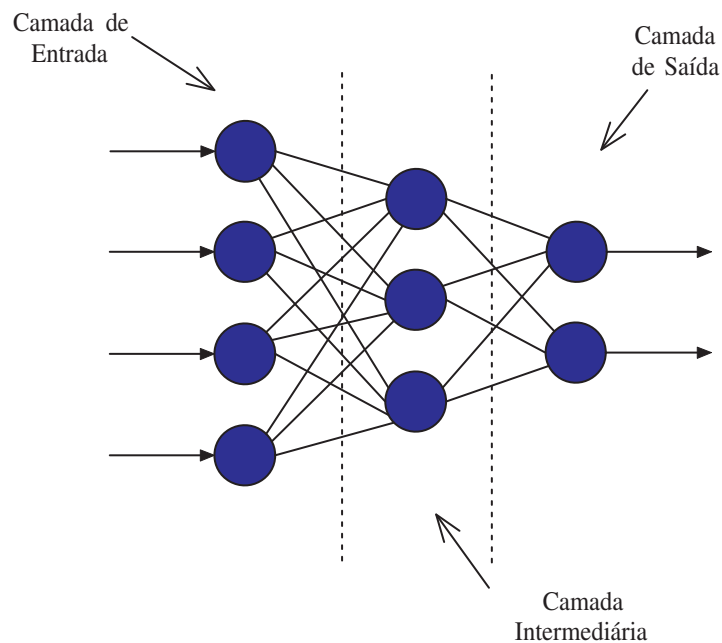


Figura 2.1: Rede *Feedforward*.

Do ponto de vista das conexões, a arquitetura de uma rede neural pode ser classificada em dois tipos:

- *Feedforward*: quando a saída de um neurônio é utilizada como entrada para os neurônios da camada seguinte. A Figura 2.1 constitui um exemplo de rede *Feedforward*.
- *Feedback*: quando a saída de um neurônio serve de entrada para neurônios da mesma camada, ou de camadas anteriores. A Figura 2.2 ilustra um exemplo de rede *Feedback*.

Uma importante característica das redes neurais é a habilidade que elas têm de aprender a partir de exemplos e melhorar seu desempenho ao longo do processo de aprendizagem. Na *aprendizagem supervisionada*, um conjunto de alvos de interesse é fornecido por um

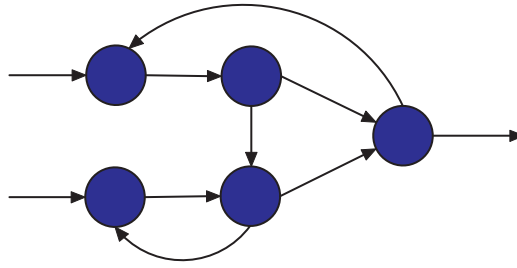


Figura 2.2: Rede *Feedback*.

professor externo. Ou seja, existe um mapeamento entrada-saída desejada, o qual a rede deve aproximar. Já na *aprendizagem não-supervisionada*, não existem saídas desejadas. O objetivo aqui é descobrir padrões significativos ou características nos dados de entrada e fazer essa descoberta sem professor. Esse processo de aprendizagem consiste em modificar repetidamente os pesos sinápticos de todas as conexões do sistema em resposta a padrões de entrada e de acordo com regras pré-determinadas, até se desenvolver uma configuração final [Haykin, 2001].

A arquitetura e o processo de aprendizagem utilizados são os principais aspectos que diferem entre os diversos tipos de redes neurais. No decorrer do capítulo, é apresentado um estudo sobre os Mapas Auto-Organizáveis, tipo de rede neural não-supervisionada utilizado neste trabalho.

Um SOM é caracterizado pela formação de um mapa topográfico dos padrões de entrada no qual as localizações espaciais (i.e., coordenadas) dos neurônios no mapa são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada, por isso o termo “Mapa Auto-Organizável” [Haykin, 2001]. Na forma mais simples de um SOM, os neurônios são dispostos em forma de grade, constituindo uma rede *feedforward* com uma única camada computacional (ver Figura 2.5).

Os Mapas Auto-Organizáveis foram originalmente propostos por von der Malsburg na década de 1970 [von der Malsburg, 1973; Willshaw and von der Malsburg, 1976], com o objetivo de reproduzir computacionalmente experimentos com estímulos visuais realizados por Hubel e Wiesel em 1962 [Hubel and Wiesel, 1962]. Posteriormente, no início da década de 1980, Kohonen [Kohonen, 1982] propôs um modelo mais simples que, além de obter resultados semelhantes aos obtidos pelo modelo de Malsburg, possibilitava a compressão (ou

quantização) dos dados de entrada (ver Seção 2.5.1). Dessa forma, os mapas de Kohonen receberam muito mais atenção na literatura que os mapas de Malsburg.

2.2 Breve Histórico das Redes Neurais

O marco inicial na modelagem de sistemas inspirados no cérebro foi o trabalho de McCulloch e Pitts, em 1943 [McCulloch and Pitts, 1943]. Nesse trabalho, um modelo de neurônio artificial foi apresentado como sendo uma unidade de processamento binária capaz de computar várias funções. Apesar de muito simples, esse modelo trouxe uma grande contribuição nas discussões sobre a construção dos primeiros computadores digitais.

Em 1949, Donald Hebb [Hebb, 1949] publicou o livro *The Organization of Behavior*, onde é apresentada a primeira regra de aprendizagem para os neurônios. Baseada na hipótese de que a estrutura de conexões do cérebro não é estática, podendo sofrer alterações com a experiência, tal regra diz que “a efetividade de uma sinapse entre dois neurônios é reforçada ou incrementada pela repetida ativação de um neurônio pelo outro através desta sinapse”.

Em 1957, Rosenblatt [Rosenblatt, 1957] introduziu uma nova abordagem para o problema de reconhecimento de padrões com o desenvolvimento de uma rede neural supervisionada conhecida como Perceptron. Mais tarde, em 1969, Minsky e Papert [Minsky and Papert, 1969] publicaram o livro “Perceptrons”, no qual prova-se matematicamente que esse tipo de rede neural - com uma única camada de neurônios - é incapaz de solucionar problemas não-linearmente separáveis (como o OU exclusivo, por exemplo). Além disso, os autores não acreditavam que um algoritmo de ajuste de pesos pudesse ser desenvolvido para o treinamento de Perceptrons de múltiplas camadas de forma a superar essa limitação. Após esse trabalho, as redes neurais sofreram uma retração significativa, de aproximadamente dez anos, devido a falta de investimentos e motivação para realizar pesquisas na área.

Apesar do aparente desinteresse pela área nesse período, alguns trabalhos continuaram a ser desenvolvidos por pesquisadores como Grossberg, Kohonen, Hopfield, von der Malsburg e outros. Um acontecimento importante na década de 1970 foi o trabalho de simulação computacional feito por von der Malsburg, em 1973 [von der Malsburg, 1973], que demonstrou a auto-organização. Em 1976, Willshaw e von der Malsburg [Willshaw and von der Malsburg, 1976] publicaram o primeiro artigo sobre a formação de Mapas Auto-Organizáveis,

motivado pelos mapas topologicamente ordenados do cérebro.

Em 1980, Grossberg [Grossberg, 1980] propôs um novo princípio de auto-organização conhecido como “teoria da ressonância adaptativa”. Em 1982, Hopfield [Hopfield, 1982] desenvolveu uma arquitetura de rede neural recorrente para memórias associativas. Nesse mesmo ano, Kohonen desenvolveu um modelo de mapa auto-organizável capaz de realizar quantização vetorial.

No entanto, o fato que efetivamente fez com que as redes neurais ganhassem novamente credibilidade foi o surgimento do algoritmo *Backpropagation* - desenvolvido por Rumelhart, Hilton e Williams, em 1986 [Rumelhart et al., 1986] - para o treinamento de Perceptrons de múltiplas camadas.

O algoritmo *Backpropagation* derrubou as barreiras (impostas aos Perceptrons de múltiplas camadas) que influenciaram na estagnação das redes neurais na década de 1970. A partir de então, passou-se a visualizar muitas aplicações interessantes, nas mais diferentes áreas, utilizando redes neurais.

2.3 Face Biológica

O desenvolvimento dos Mapas Auto-Organizáveis como modelo neural artificial é motivado por duas características distintivas do cérebro humano:

1. diferentes entradas sensoriais - como a tátil [Kaas, 1983], a visual [Hubel and Wiesel, 1962; Hubel and Wiesel, 1977] e a acústica [Suga and Tsuzuki, 1985] - são mapeadas para áreas diferentes do córtex cerebral de uma maneira topologicamente ordenada [Haykin, 2001]. Ou seja, neurônios que lidam com partes relacionadas de informação estão próximos entre si de modo a poderem interagir através de conexões sinápticas curtas [Knudsen et al., 1987]. A Figura 2.3 (adaptada de Haykin [Haykin, 2001]) mostra os mapas cito-arquiteturais do córtex cerebral. O córtex motor é representado pelas áreas 4, 6 e 8; o córtex somatosensório, pelas áreas 1, 2 e 3; o córtex visual, pelas áreas 17, 18 e 19; e o córtex auditivo, pelas áreas 41 e 42.
2. Apesar das principais estruturas neurais do cérebro serem determinadas geneticamente, há evidências de que projeções sensoriais são afetadas por freqüentes experiências

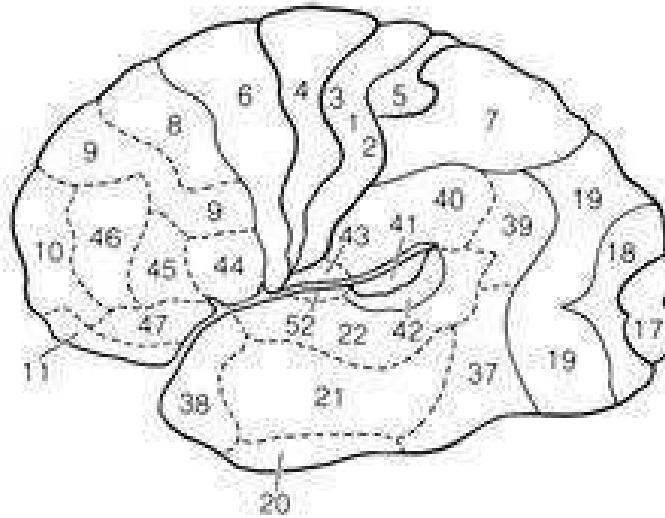


Figura 2.3: Mapa citoarquitetural do córtex cerebral. Os números do mapa indicam as diferentes áreas sensoriais do córtex cerebral.

de aprendizado, como por exemplo:

- reconstituição de órgãos sensoriais ou tecidos cerebrais em pessoas jovens, uma vez que algumas projeções ainda não estão desenvolvidas por completo;
- recrutamento de células para diferentes tarefas dependendo da experiência que se tem sobre elas.

Hubel e Wiesel [Hubel and Wiesel, 1962; Hubel and Wiesel, 1977] demonstraram uma forte relação entre as regiões de excitação na retina - formadas por células sensíveis à luz, conhecidas como *fotoreceptores* - e a ativação de neurônios localizados no córtex visual. Foram observados, em uma espécie de macacos, dois tipos de regiões na retina associadas a determinados neurônios do córtex visual [Hubel and Wiesel, 1962]:

- *região excitatória* - que, quando estimulada, provoca aumento da taxa de ativação do neurônio;
- *região inibitória* - que, quando estimulada, provoca diminuição da taxa de ativação do neurônio.

Há, ainda, uma *região neutra*, que não provoca qualquer efeito sobre a taxa de ativação do neurônio. Portanto, o *campo receptivo* de determinados neurônios do córtex visual é

formado pela união das suas regiões excitatória e inibitória na retina, e pode ser modelado pela função Chapéu Mexicano¹ [Kohonen, 1989], como ilustrado na Figura 2.4. A região excitatória estende-se circularmente até uma determinada distância radial do ponto de maior ativação do neurônio. A partir daí, os estímulos passam a ser inibitórios².

Uma vez que é possível haver intersecção entre campos receptivos, conclui-se que para determinados neurônios corticais existe um conjunto de neurônios em sua vizinhança que contribuem positivamente no seu valor de ativação; enquanto outros, situados a uma distância maior, contribuem negativamente neste valor. Esse comportamento é conhecido como *relações laterais da célula*.

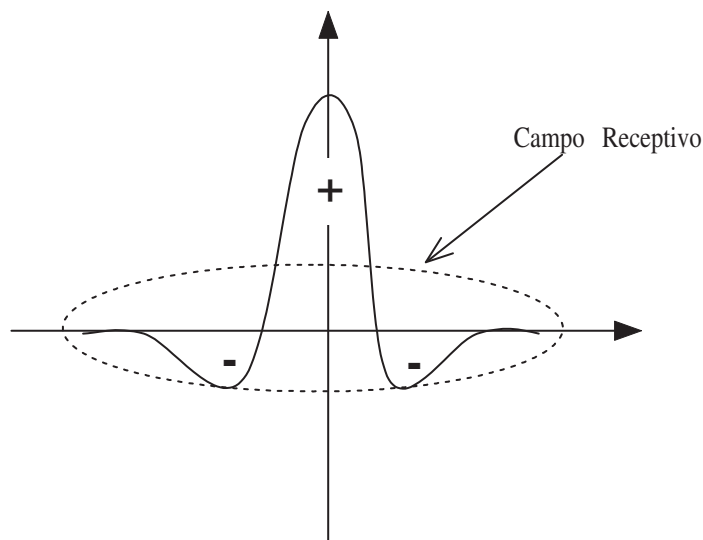


Figura 2.4: Função Chapéu Mexicano.

Hubel e Wiesel [Hubel and Wiesel, 1962] também demonstraram, em experimentos com gatos, que certas características visuais (como barras e bordas) com orientações similares são mapeadas em regiões topologicamente próximas no córtex visual primário, formando *grupos de orientações*. Conceitualmente, partia-se para níveis mais elevados de abstração: era possível identificar regiões do córtex que respondiam não ao conceito mais genérico de estímulo visual, mas a uma característica específica deste [Vasconcelos, 2000]. Este comportamento

¹A função Chapéu Mexicano pode ser aproximada por uma diferença de Gaussianas, bem como pela Laplaciana da Gaussiana [Marr, 1982].

²Nos primatas, a região excitatória estende-se por um raio de 50 a 100 μm . Enquanto que a região inibitória estende-se por um raio de 200 a 500 μm [Kohonen, 1989].

é conhecido como *ordenação topológica*.

Os conceitos de *campo receptivo* (ou *relações laterais da célula*) e *ordenação topológica*, inicialmente desenvolvidos nos trabalhos de Hubel e Wiesel, foram incorporados pelos principais modelos de Mapas Auto-Organizáveis.

2.4 Treinamento

Como vimos na Seção 2.1, na forma mais simples de um SOM, os neurônios são dispostos em forma de grade, constituindo uma rede *feedforward* com uma única camada computacional. O principal objetivo do SOM é mapear (auto-organizar) os padrões de entrada em seus neurônios de uma forma topologicamente ordenada. Isso é feito durante a fase de treinamento que, de acordo com Haykin [Haykin, 2001], possui três etapas básicas:

1. *competição*. Para cada padrão de entrada, os neurônios do mapa calculam seus respectivos valores de uma função discriminante. Esta função fornece a base para a competição entre os neurônios. O neurônio com o maior valor da função discriminante é declarado vencedor da competição;
2. *cooperação*. O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados que cooperarão entre si;
3. *adaptação sináptica*. Os neurônios excitados aumentam seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes adequados aplicados a seus pesos sinápticos. Os ajustes feitos são tais que a resposta do neurônio vencedor à aplicação subsequente de um padrão de entrada similar é melhorada.

Na próxima seção - a qual descreve, em particular, o funcionamento dos mapas de Kohonen - tais etapas serão melhor compreendidas.

2.5 Mapas de Kohonen

Representar dados de forma econômica, preservando seus relacionamentos, é um dos principais problemas estudados nas Ciências da Informação. Estudos neurobiológicos postulam

que em nosso cérebro, são formadas *representações reduzidas* sobre fatos relevantes, sem a perda dos seus relacionamentos. Kohonen utiliza este conceito, além daqueles citados na Seção 2.3 (i.e., *relações laterais da célula e ordenação topológica*), para armazenar padrões em seu Mapa Auto-Organizável.

A Figura 2.5 ilustra a arquitetura de um mapa de Kohonen bidimensional³. Os nós fonte da camada de entrada são totalmente conectados com os neurônios do mapa, representando uma estrutura alimentada adiante (*feedforward*) com uma única camada computacional. Um mapa unidimensional é um caso especial da arquitetura apresentada na Figura 2.5, onde a camada computacional consiste de uma única coluna ou linha de neurônios [Haykin, 2001].

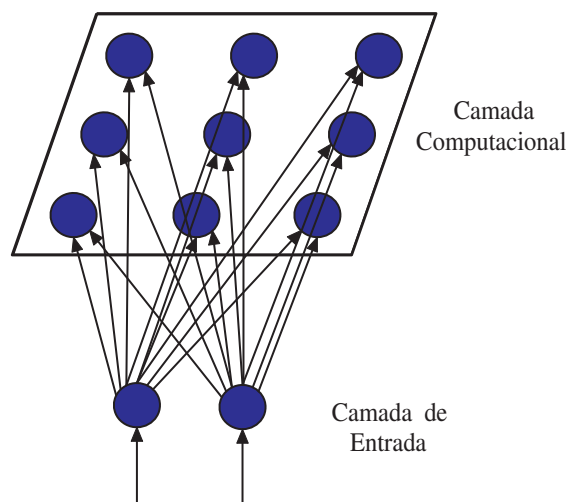


Figura 2.5: Mapa de Kohonen Bidimensional.

2.5.1 Algoritmo de Treinamento

O mapa de Kohonen é auto-organizado através de um processo cíclico de comparação dos padrões de entrada com os vetores de pesos armazenados em cada neurônio [Beale and Jackson, 1990]. O treinamento desse modelo (ver Algoritmo 2.1) baseia-se simplesmente na procura do neurônio cujos pesos são mais próximos de um determinado padrão de entrada (com a menor distância Euclidiana) e no aumento da similaridade entre eles (padrão de entrada e pesos do neurônio vencedor).

³Grande parte das redes neurais do cérebro, especialmente do córtex, são formadas basicamente por camadas bidimensionais de neurônios. [Kohonen, 1989]

Algoritmo 2.1 Treinamento dos mapas de Kohonen.

1. Inicialize o mapa M

Sejam $w_{ji}(t)$ ($0 \leq j \leq n - 1$) o peso da entrada j ao neurônio i no tempo t e k o número de épocas de treinamento. Inicialize os vetores de pesos das n entradas com valores pequenos e randômicos. Inicialize o raio da vizinhança em torno de cada neurônio i com um valor alto.

2. Apresente o padrão de entrada X ao mapa M

$$X = x_0(t), x_1(t), \dots, x_{n-1}(t)$$

3. Calcule a distância d_i entre a entrada X e cada neurônio i do mapa M

$$d_i = \sum_{j=0}^{n-1} (x_j(t) - w_{ji}(t))^2$$

4. Selecione a menor distância

O neurônio vencedor i^* é aquele com menor d_i .

5. Atualize os pesos para o neurônio i^* e para sua vizinhança $N_{i^*}(t)$

$$w_{ji}(t+1) = w_{ji}(t) + \eta(t)(x_j(t) - w_{ji}(t))$$

em que $\eta(t)$ é o parâmetro de taxa de aprendizagem. Tanto a vizinhança $N_{i^*}(t)$ quanto a taxa de aprendizagem $\eta(t)$ decrescem com o tempo.

6. Continue a partir do passo 2

até atingir o número de épocas de treinamento k .

O treinamento dos mapas de Kohonen pode ser visto como uma *quantização vetorial*, uma vez que é possível armazenar um conjunto grande de vetores de entrada em um conjunto menor de protótipos (vetores de pesos), caso a dimensão da entrada seja maior que a dimensão do mapa⁴. Essa característica é útil em tarefas de agrupamento, já que padrões semelhantes podem ser mapeados em um único neurônio.

2.5.2 Vizinhança Topológica

Vimos na Seção 2.3 que um neurônio ativado tende a excitar mais fortemente os neurônios localizados em sua vizinhança imediata do que aqueles mais distantes dele.

No modelo de Kohonen, uma função N_i representa a *vizinhança topológica* em torno do neurônio vencedor i . Considerando c como sendo um dos neurônios cooperativos dessa vizinhança e que $d_{i,c}$ represente a distância lateral entre i e c , podemos assumir que a vizinhança topológica N_i é uma função unimodal da distância $d_{i,c}$, desde que ela satisfaça as seguintes exigências [Haykin, 2001]:

- a vizinhança topológica N_i alcança o seu valor máximo no neurônio vencedor i para o qual a distância $d_{i,c}$ é zero;
- a área da vizinhança topológica N_i decresce monotonicamente com o aumento da distância $d_{i,c}$, decaindo a zero para $d_{i,c} \rightarrow \infty$; esta é uma condição necessária para a convergência.

Uma função N_i que satisfaz estas exigências é a Gaussiana:

$$N_i = \exp\left(\frac{-d_{i,c}^2}{2\varphi^2}\right) \quad (2.1)$$

em que φ , que decresce com o tempo, mede o grau com o qual neurônios excitados na vizinhança do neurônio vencedor participam do processo de aprendizagem. Geralmente, a dependência de φ com o tempo t é dada por:

$$\varphi(t) = \varphi_0 \cdot \exp\left(\frac{-t}{\tau}\right) \quad (2.2)$$

⁴No modelo de Malsburg, como a dimensão de entrada é igual à dimensão de saída, não é possível realizar quantização vetorial.

em que φ_0 é o valor inicial de φ e τ é uma constante de tempo.

Como vimos no Algoritmo 2.1, os neurônios na vizinhança topológica N_i também são modificados durante o processo de aprendizado. O mapa tenta criar regiões, denominadas de *bolhas de atividade*, que irão responder a um conjunto de valores em torno do padrão treinado; possibilitando que padrões de teste similares aos padrões de treinamento sejam classificados corretamente, ainda que a rede não os tenha visto anteriormente. Isto demonstra a propriedade de generalização dos mapas de Kohonen.

2.6 Mapas Hierárquicos

Em um mapa de Kohonen, a busca pelo neurônio vencedor durante a fase de uso tem complexidade temporal igual a $O(n)$, em que n é o número de neurônios do mapa [Koikkalainen and Oja, 1990]. Dessa forma, quando o número de neurônios é muito grande, o uso deste tipo de rede neural em certas aplicações torna-se inviável.

Esta complexidade pode, entretanto, ser reduzida se os neurônios do mapa forem organizados em uma estrutura de árvore [Friedman, 1977], ou seja, em hierarquias. O HFM (*Hierarchical Feature Map*) [Miikkulainen, 1990] e o TS-SOM (*Tree-Structured SOM*) [Koikkalainen and Oja, 1990; Koikkalainen, 1994] são exemplos de Mapas Auto-Organizáveis com tal estrutura.

Um Mapa Auto-Organizável Hierárquico (H-SOM: *Hierarchical SOM*) é, portanto, uma rede neural composta por SOMs independentes organizados em forma de árvore, capaz de representar relações hierárquicas entre os dados de entrada.

Assim como no mapa de Kohonen, os Mapas Auto-Organizáveis Hierárquicos necessitam que as dimensões dos vários SOMs que compõem a hierarquia sejam definidas antes da fase de treinamento.

Em um modelo mais recente de H-SOM, o GH-SOM [Dittenbach et al., 2000], as dimensões dos SOMs, bem como a profundidade da hierarquia são determinadas automaticamente durante a fase de treinamento. A Figura 2.6 ilustra a arquitetura de um GH-SOM com uma unidade no nível 0, um SOM 2X2 no nível 1, quatro SOMs no nível 2 (um SOM 2X3, um SOM 3X2 e dois SOMs 2X2) e um SOM 3X3 no nível 3. Note que cada um desses mapas podem ter diferentes números e disposições dos neurônios.

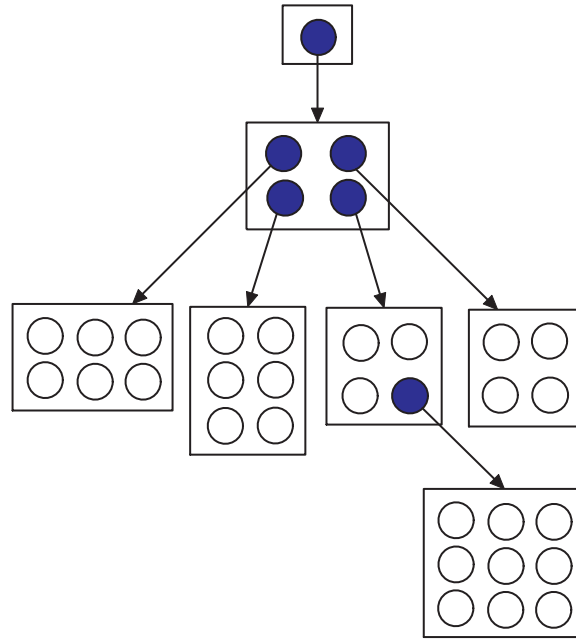


Figura 2.6: Arquitetura de um GH-SOM

O processo de treinamento do GH-SOM é realizado de acordo com o algoritmo SOM de Kohonen, porém, a cada λ iterações, o neurônio i com maior erro de quantização q_i é identificado.

O erro de quantização de um neurônio é calculado como a soma das distâncias Euclidianas entre seu vetor de pesos W_i e os vetores de entrada X_k mapeados neste neurônio, ou seja:

$$q_i = \sum_{k=0}^n d(X_k, W_i) \quad (2.3)$$

Em seguida, entre esse neurônio - denominado *error unit* - e seu vizinho imediato mais dissimilar, uma nova linha ou coluna de neurônios é inserida.

A Figura 2.7 ilustra a inserção de neurônios no GH-SOM. No lado esquerdo da figura - situação anterior à inserção - o neurônio A é a *error unit* e o neurônio B é seu vizinho mais dissimilar. O lado direito ilustra a nova linha de neurônios inserida entre A e B.

Um mapa cresce até que a média dos erros de quantização de seus neurônios seja reduzida a um certo valor α_1 , dado por:

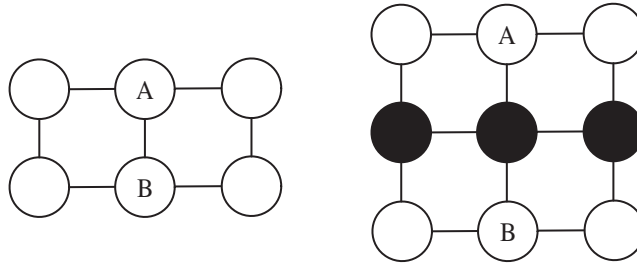


Figura 2.7: Inserção de neurônios.

$$\alpha_1 = \tau_1 \cdot q_i \quad (2.4)$$

em que τ_1 representa uma fração do erro de quantização q_i do neurônio i que deu origem a este mapa.

Em seguida, os neurônios com erro de quantização maiores que um limiar α_2 são expandidos, dando origem a outros mapas. α_2 é dado por:

$$\alpha_2 = \tau_2 \cdot q_0 \quad (2.5)$$

em que τ_2 representa uma fração do erro de quantização inicial q_0 (na camada 0).

Assim, padrões mapeados em neurônios que não conseguiram reduzir seus erros de quantização têm uma outra oportunidade de se auto-organizar em um espaço maior (nos filhos desse neurônio). α_1 e α_2 indicam, portanto, o nível de granularidade desejado para representação dos dados de entrada.

A hierarquia cresce até que não haja mais neurônios a serem expandidos, i.e., quando todos os neurônios do GH-SOM possuírem erros de quantização abaixo de α_2 .

O Capítulo 6 apresenta um experimento que compara os desempenhos do SOM e do GH-SOM no problema de reconhecimento de dígitos manuscritos.

2.7 Conceito de Mapa Contextual

Uma forma simples de se visualizar um SOM é através de um Mapa Contextual. Para gerá-lo, atribuem-se rótulos aos neurônios do SOM dependendo de como cada padrão excita um

neurônio em particular [Haykin, 2001]. Ou seja, cada neurônio da rede é rotulado pelo padrão para o qual produz a melhor resposta (menor distância Euclidiana).

A Figura 2.8 ilustra a geração de um Mapa Contextual. Inicialmente, temos apenas as respostas mais fortes (BMUs: Best Map Units) para os padrões P1, P2, P3 e P4. Em seguida, todos os neurônios são rotulados com os padrões mais representativos, eliminando as ambigüidades existentes.

Com essa estrutura, é possível visualizar as “bolhas de atividade” formadas durante o treinamento. Ou seja, os neurônios que não foram BMUs (vencedores) nessa fase, provavelmente serão BMUs de padrões de teste desconhecidos, porém próximos aos padrões de treinamento.

Neurônio	Padrão	DE
1	P1	0.011
1	P3	0.002
2	P1	0.5
2	P2	0.8
3	P3	0.3
3	P4	0.1
4	P2	0.003
4	P4	0.018

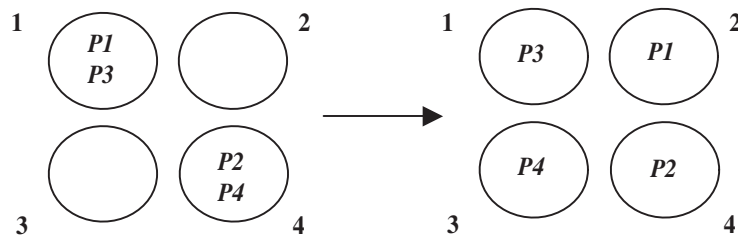


Figura 2.8: Geração de um mapa contextual. DE = Distância Euclidiana.

Além da visualização, os Mapas Contextuais podem ser utilizados para auxiliar a classificação de padrões. Isto é, a partir deles, é possível identificar a que classe pertence um padrão de teste propagado no SOM.

2.8 Incorporando Invariâncias

Um ponto fundamental em tarefas de reconhecimento de padrões é a construção de classificadores invariantes, ou seja, classificadores cujas saídas não são afetadas por transformações da entrada (como variações de escala e orientação, por exemplo).

De acordo com Barnard e Casasent [Barnard and Casasent, 1991], existem no mínimo três técnicas diferentes para a construção de classificadores neurais invariantes:

1. *Invariância por Estrutura*. Nesta técnica, a invariância é incorporada ao projeto da rede neural. Especificamente, conexões sinápticas entre os neurônios da rede são criadas de tal forma que versões transformadas do mesmo objeto são forçadas a produzir a mesma saída.
2. *Invariância por Treinamento*. Nesta técnica, a rede é treinada apresentando-se um número de diferentes exemplos - correspondendo a diferentes variações - do mesmo objeto. Com um número de exemplos suficientemente grande, espera-se que a rede generalize corretamente, reconhecendo variações não vistas anteriormente.
3. *Espaço de características Invariantes*. Nesta técnica, apresenta-se ao classificador apenas um conjunto de características invariantes que representa a informação essencial do objeto. Quando tais características são extraídas, o classificador é poupado da tarefa de aprender as variações de um objeto com complicadas superfícies de decisão. A Figura 2.9 ilustra a arquitetura de um sistema de espaço de características invariantes.

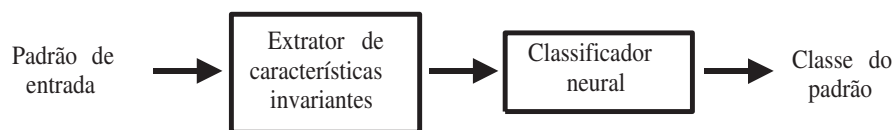


Figura 2.9: Arquitetura de um sistema de espaço de características invariantes.

Veremos, no Capítulo 4, que os sistemas cujos classificadores são Mapas Auto-Organizáveis geralmente utilizam *espaços de características invariantes*. Uma das razões

para isso é o fato de não ter como impedir o agrupamento de padrões com base em características como orientação, escala, cor, etc., já que os Mapas Auto-Organizáveis utilizam aprendizagem não-supervisionada. Ou seja, é possível que objetos de classes diferentes pareçam similares em uma mesma orientação, por exemplo, e sejam agrupados em uma mesma região do mapa. O Capítulo 6 apresenta um experimento que ilustra esse tipo de problema.

2.9 Considerações Finais

Neste capítulo foram apresentadas as principais características e funcionalidades dos Mapas Auto-Organizáveis, dando ênfase ao modelo de Kohonen.

O Mapa Auto-Organizável foi originalmente proposto por von der Malsburg, inspirado em características do cérebro - tais como *campo receptivo* e *ordenação topológica* - estudadas inicialmente por Hubel e Wiesel [Hubel and Wiesel, 1962], conforme discutido na *Seção 2.3*. Entretanto, por realizar *quantização vetorial*, o modelo de Kohonen [Kohonen, 1982] foi bem mais difundido na literatura. Já o GH-SOM, além de conservar as principais propriedades do modelo de Kohonen (i.e., *quantização vetorial* e formação de *bolhas de atividade*), define automaticamente as dimensões dos SOMs que compõem sua hierarquia (durante a fase de treinamento) e, por se tratar de um modelo cujos neurônios são organizados em forma de árvore [Friedman, 1977], possui tempo de resposta menor que um mapa de Kohonen (durante a fase de uso).

Considerando que o problema investigado requer uma solução que agregue uma representação compacta para imagens e um método eficiente de classificação e indexação, utilizamos o GH-SOM em um *sistema de espaço de características invariantes*, descrito no Capítulo 5.

Capítulo 3

Recuperação de Imagens Baseada em Conteúdo

Nos Sistemas de recuperação de Imagens Baseada em Conteúdo, as imagens são indexadas a partir de suas próprias características visuais, como cor, forma e textura. Tais sistemas surgiram na década de 1990 devido ao rápido crescimento das coleções de imagens digitais e conseqüente dificuldade de anotação manual das imagens nos sistemas baseados em texto. Neste capítulo, são apresentadas as duas etapas fundamentais para o funcionamento dos SRIBCs: *extração de características e indexação*.

Inicialmente, são descritas algumas técnicas de extração e representação de características visuais utilizadas tanto por SRIBCs quanto por sistemas de Visão Computacional, em geral. Em seguida, são apresentadas as principais estruturas de indexação utilizadas por SRIBCs. Finalmente, os Mapas Auto-Organizáveis são apresentados como uma promissora ferramenta de indexação.

3.1 Extração de Características

Esta seção apresenta as principais características e técnicas de extração e representação destas, utilizadas pelos SRIBCs e pelos sistemas de Visão Computacional, em geral.

3.1.1 Cor

Em SRIBCs, a característica “cor” é geralmente representada a partir de histogramas [Rui et al., 1999]. Um histograma de cor é uma função de distribuição de densidade que indica o percentual (ou o número) de pixels, em uma imagem, que apresenta uma determinada intensidade de cor. Em imagens coloridas, calcula-se o histograma correspondente a cada um de seus componentes. Para uma imagem do tipo RGB, por exemplo, são calculados três histogramas, um para cada componente (R, G e B).

A Figura 3.1 ilustra uma imagem em níveis de cinza e seu respectivo histograma em forma de gráfico de barras.

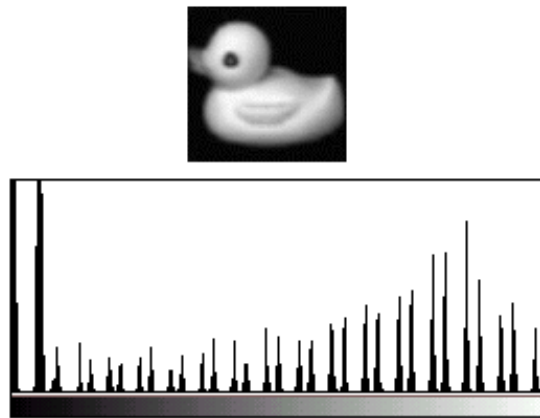


Figura 3.1: Histograma com 256 níveis de cinza.

3.1.2 Forma

Essa característica diz respeito às formas (*shapes*) dos objetos que compõem uma imagem. A maneira mais simples de identificar tais objetos é através da binarização, em que os pixels com valores abaixo de um determinado limiar são transformados em 0 (preto) e os pixels acima desse limiar são transformados em 1 (branco)¹. A Figura 3.2 ilustra um exemplo de binarização com limiar igual a 128.

Os detectores de bordas - como *Canny*, *Prewitt*, *Sobel*, entre outros - também provêm

¹É importante observar que a binarização não funciona bem em imagens com baixo contraste entre suas regiões componentes.

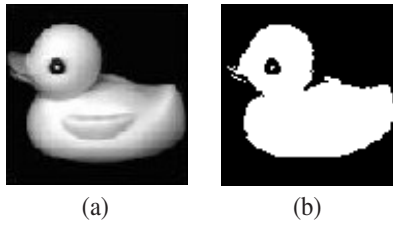


Figura 3.2: Exemplo de binarização. (a) Imagem em tons de cinza. (b) Imagem binária.

uma indicação da existência física de um objeto dentro de uma imagem [Pratt, 1991]. Para esses detectores, uma borda é caracterizada por uma mudança brusca de um intervalo de níveis de cinza para outro, representando a fronteira entre duas regiões. A Figura 3.3 ilustra um exemplo de detecção de bordas, no qual é obtida apenas a informação do contorno do objeto.

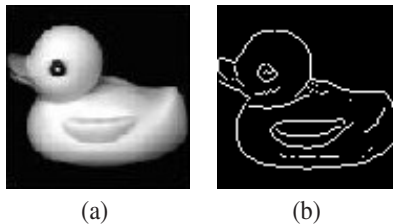


Figura 3.3: Exemplo de extração de bordas. (a) Imagem em tons de cinza. (b) Imagem de bordas.

Uma outra forma de segmentar imagens em objetos componentes é a partir de técnicas baseadas em regiões. A técnica conhecida como *Crescimento de Regiões* [Brice and Fenema, 1970], por exemplo, agrupa iterativamente pixels com propriedades similares (i.e., nível de cinza, textura, cor, etc.) em regiões maiores, até alcançar um determinado critério de parada.

Existem SRIBCs - como o SAFE [Smith and Chang, 1997], por exemplo - em que é possível realizar consultas baseadas em regiões (ou objetos) independentes da imagem. Esse tipo de consulta, a qual independe da localização física do objeto na imagem, é chamada de *espacial*.

Após as formas dos objetos que compõem uma imagem terem sido identificadas, é geralmente necessário descrevê-las com base em propriedades invariantes a translação, escala e

rotação. Segundo Rui *et alli* [Rui et al., 1999] os descritores de formas podem ser divididos em duas classes: (i) os que consideram apenas o contorno dos objetos e (ii) os que consideram a região inteira dos objetos. Os exemplos mais representativos dessas duas classes são os Descritores de Fourier [Pratt, 1991] e os Momentos Invariantes de Hu [Hu, 1962], respectivamente.

Descritores de Fourier

A Transformada de Fourier [Pratt, 1991] é uma técnica utilizada para representar, descrever e processar sinais através da combinação linear de funções elementares.

Um Descritor de Fourier para formas é obtido a partir da Transformada Discreta de Fourier (TDF) aplicada aos N pontos do contorno do objeto em um plano de domínio complexo. A TDF é dada pela seguinte equação:

$$F(\mu) = \left(\frac{1}{N}\right) \sum_{k=0}^{N-1} f(k) \cdot (\cos(A) - i \cdot \text{sen}(A)) \quad (3.1)$$

em que $\mu = 0, 1, 2, \dots, f(k) = x_k + i \cdot y_k, i = \sqrt{-1}$ e $A = 2\pi\mu k/N$.

A Figura 3.4 ilustra o contorno de uma forma representado em um plano de domínio complexo. A aplicação de uma TDF do ponto P_0 a P_7 , nos dá os valores do Descritor de Fourier correspondente para esta forma.

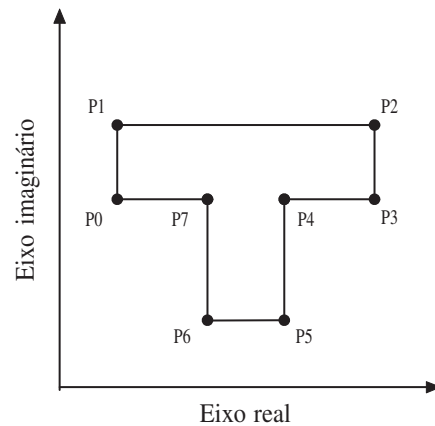


Figura 3.4: Contorno em um plano complexo.

A invariância à orientação, escala e translação é obtida a partir de simples modificações nos descritores obtidos. A invariância à translação, por exemplo, é obtida atribuindo-se o

valor zero ao descritor $F(0)$ [Keyes and Winstanley, 1999].

Momentos Invariantes de Hu

Em Visão Computacional, os momentos são utilizados para medir a distribuição de pixels dos objeto de interesse na imagem. Hu [Hu, 1962] propôs sete momentos invariantes a orientação, escala e translação que podem ser utilizados como descritores de forma. Tais descritores são calculados a partir de expressões simples, tendo como base a seguinte equação [Keyes and Winstanley, 2001]:

$$M_{pq} = \sum_x^x \sum_y^y x^p y^q f(x, y) \quad (3.2)$$

M_{pq} é o momento bi-dimensional da imagem $f(x, y)$, de ordem $p + q$, em que p e q são números naturais.

A invariância à translação é obtida a partir do momento central, dado por:

$$\mu_{pq} = \sum_x \sum_y (x - X)^p \cdot (y - Y)^q \cdot f(x, y) \quad (3.3)$$

$$X = \frac{M_{10}}{M_{00}} \quad (3.4)$$

$$Y = \frac{M_{01}}{M_{00}} \quad (3.5)$$

em que X e Y são os centróides (centro de gravidade) da imagem.

A invariância à escala é obtida a partir da seguinte equação:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (3.6)$$

em que $\gamma = \left(\frac{p+q}{2}\right) + 1$

Finalmente, os momentos de Hu são definidos por sete expressões que são adicionalmente invariantes à orientação:

$$\phi_1 = \mu_{20} + \mu_{02} \quad (3.7)$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (3.8)$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (\mu_{03} - 3\mu_{21})^2 \quad (3.9)$$

$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{03} + \mu_{21})^2 \quad (3.10)$$

$$\begin{aligned} \phi_5 = & (3\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ & (3\mu_{21} - \mu_{03}) \cdot (\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (3.11)$$

$$\phi_6 = (\mu_{20} - \mu_{02}) \cdot [(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12}) \cdot (\mu_{21} + \mu_{03}) \quad (3.12)$$

$$\begin{aligned} \phi_7 = & (3\mu_{21} - \mu_{03}) \cdot (\mu_{30} + \mu_{12}) \cdot [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ & (3\mu_{12} - \mu_{30}) \cdot (\mu_{21} + \mu_{03}) \cdot [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (3.13)$$

Tradicionalmente, os momentos invariantes são aplicados em toda região do objeto. Entretanto, os sete momentos de Hu podem ser computados utilizando apenas o contorno C dos objetos [Chien, 1993], baseados na seguinte equação:

$$M_{pq} = \sum_{(x,y) \in C} x^p y^q f(x,y) \quad (3.14)$$

Nesse caso, o momento central é definido por:

$$\mu_{pq} = \sum_{(x,y) \in C} (x - X)^p \cdot (y - Y)^q \cdot f(x,y) \quad (3.15)$$

Como antes, X e Y são os centróides da imagem.

3.1.3 Textura

Uma textura é caracterizada por um ou vários elementos visuais que se repetem ao longo de uma superfície. Geralmente, estas repetições envolvem variações de cor, escala, orientação, etc. Portanto, os SRIBCs ou qualquer outro sistema de visão, devem ser capazes de classificar texturas de maneira independente de tais variações.

Baseados em experimentos psicológicos, Tamura *et alli* [Tamura et al., 1976] consideraram seis características visuais para a representação de texturas: *coarseness* (fineza), *contrast* (contraste), *directionality* (direcionalidade), *line-likeness* (tipo-linha), *regularity* (regularidade) e *roughness* (aspereza). A característica *coarseness*, por exemplo, diz respeito à distância entre os elementos primitivos que formam a textura.

Também é possível representar texturas através de múltiplas resoluções ou escalas. A Multi-Resolução Simultânea Auto-Regressiva (MR-SAR) [Xu et al., 2000] e as Wavelets -

especialmente os filtros de Gabor [Manjunath and Ma, 1998] - são exemplos de técnicas de representação de texturas, nas quais diferentes características podem ser observadas em cada resolução.

As Wavelets são funções, também chamadas de filtros, obtidas através de dilatações e translações de uma função modelo conhecida como Wavelet mãe. A função de Gabor é um tipo de Wavelet mãe que descreve uma senóide de frequência W modulada por uma Gaussiana de duração σ , como descreve a equação abaixo:

$$g(x,y) = \left(\frac{1}{2\Pi\sigma_x\sigma_y} \right) \cdot \exp \left[\frac{-1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) + 2\Pi(-1)^{1/2}Wx \right] \quad (3.16)$$

Os filtros de Gabor podem ser utilizados na detecção de linhas e bordas em diferentes orientações e escalas. Além disso, medidas estatísticas obtidas a partir de imagens filtradas podem ser utilizadas para representar texturas. Martins *et alli* [Martins et al., 2002] e Manjunath e Ma [Manjunath and Ma, 1998], por exemplo, utilizaram a média e o desvio padrão dos coeficientes de Gabor para representar texturas em imagens de sensoriamento remoto.

3.2 Indexação em SRIBCs

Quando uma imagem é representada a partir de um vetor de características, ela pode ser vista como um ponto no espaço k -dimensional, em que k é o número de características do vetor [Brown and Gruenwald, 1998]. A Figura 3.5 ilustra um vetor de características de uma imagem representado em um espaço tridimensional.

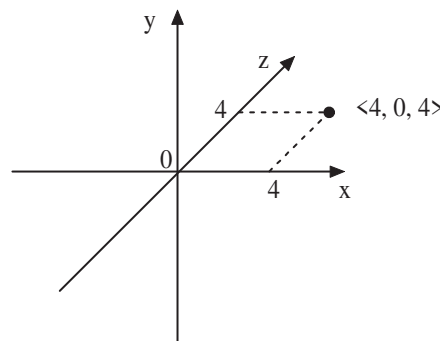


Figura 3.5: Representação de um vetor de características no espaço 3D.

Na abordagem tradicional de RIBC, geralmente são utilizadas Estruturas de Dados Multidimensionais derivadas da B-Tree [Bayer and McCreight, 1972] como estruturas de indexação. Essas estruturas são classificadas de acordo com a presença ou ausência de MBRs (Minimum Bounding Region), em que MBR é a menor região capaz de delimitar um certo conjunto de elementos contidos no espaço multidimensional. Em RIBC, esses elementos correspondem aos vetores de características extraídos das imagens.

Tanto nas *EDMs sem MBRs* quanto nas *EDMs com MBRs*, cada nó contém um conjunto de valores que identifica uma região R do espaço multidimensional e um conjunto de ponteiros para seus filhos, que correspondem a subregiões de R . O nó-pai representa todo o espaço multidimensional e os nós-folha contém ponteiros para os elementos que estão contidos em suas respectivas regiões. A Figura 3.6 exemplifica o particionamento do espaço realizado por esses dois tipos de EDMs. Nessa figura, as regiões são rotuladas por números e os elementos são rotulados por letras.

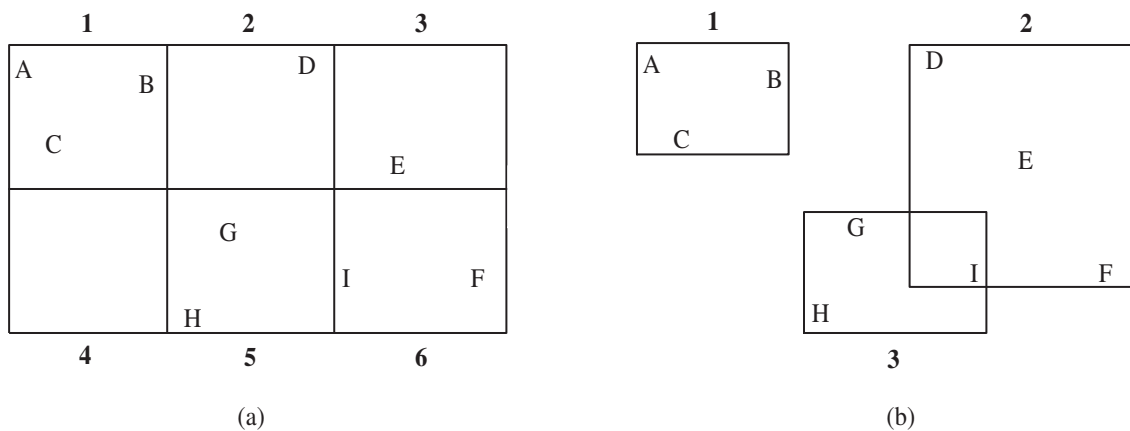


Figura 3.6: Particionamento do espaço sem MBRs (a) e com MBRs (b).

Uma vantagem das *EDMs com MBRs* é o fato das MBRs cobrirem apenas os elementos contidos no espaço multidimensional, reduzindo, assim, o espaço de busca. A maior parte das *EDMs sem MBRs* - como a KDB-Tree (K-Dimensional B-Tree) [Robinson, 1981], por exemplo - armazenaria a região 4 da Figura 3.6 (a) em um nó da árvore.

Por outro lado, as *EDMs sem MBRs* particionam o espaço em regiões que não se sobrepõem. Dessa forma, apenas um caminho é percorrido na árvore durante a busca. A maior parte das *EDMs com MBRs* - como R-Tree (Rectangle Tree) [Guttman, 1984] e TV-Tree

(Telescopic Vector Tree) [Lin et al., 1994], por exemplo - permite sobreposição de MBRs, como ocorre com as regiões 2 e 3 da Figura 3.6 (b). Para serem retornados os 3-vizinhos mais próximos de I (i.e., E, F e G), dois caminhos precisam ser percorridos na árvore, como ilustrado na Figura 3.7.

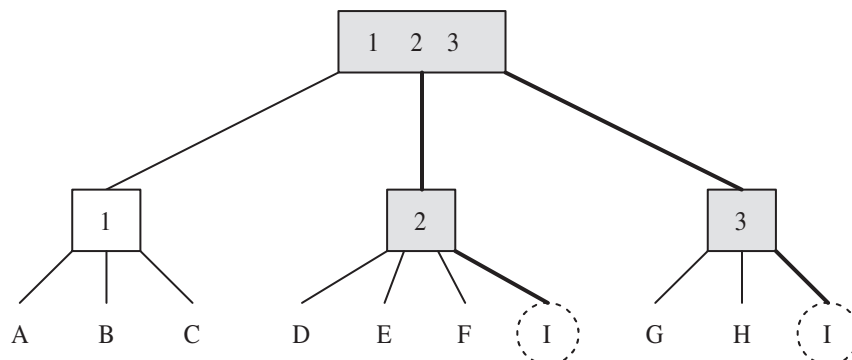


Figura 3.7: Árvore com sobreposição de MBRs.

Em SRIBCs, normalmente são utilizados vetores de características com muitos elementos. Entretanto, grande parte das EDMs derivadas da B-Tree são ineficientes ao lidar com altas dimensões [Alexandrov et al., 1995; Brown and Gruenwald, 1998; Zhang and Zhong, 1995]. Considere, por exemplo, uma EDM (derivada da B-Tree) cujas folhas possuem espaço para armazenar apenas um vetor de características. Em uma busca por k-vizinhos mais próximos seria necessário percorrer mais de um caminho na árvore, uma vez que a informação de vizinhança geralmente não é mantida nesse tipo de estrutura [Zhang and Zhong, 1995].

As TV-Trees tentam solucionar esse problema ao escolher apenas as dimensões necessárias para distinguir entre os diversos vetores de características. Considere, por exemplo, uma MBR de uma TV-Tree contendo os vetores $\langle 2, 4, 1, 3 \rangle$, $\langle 2, 6, 7, 3 \rangle$ e $\langle 2, 1, 0, 3 \rangle$. Nesse caso, as dimensões 2 e 3 seriam escolhidas como ativas, já que as dimensões 1 e 4 são iguais para todos os vetores. Uma desvantagem da TV-Tree é que suas MBRs podem se sobrepor.

Além das EDMs, os SOMs hierárquicos podem ser utilizados como estruturas de indexação de dados multidimensionais. Esse tipo de rede neural, além de possuir uma topologia em forma de árvore onde apenas um caminho é percorrido durante a busca, é capaz de realizar compressão de dados (quantização vetorial), reduzindo ainda mais o espaço de busca. O

Capítulo 4 descreve o funcionamento de quatro sistemas que utilizam SOMs nesse tipo de problema. Em tais sistemas, os vetores de características são mapeados em BMUs (de acordo com o algoritmo de Kohonen) que indexam a base de imagens.

3.3 Considerações finais

Neste capítulo, foram apresentadas as principais características dos sistemas de recuperação de imagens baseada em conteúdo. Tais sistemas caracterizam-se por indexar imagens a partir de suas próprias características visuais, como cor, forma e textura. Portanto, foram descritas algumas das técnicas mais utilizadas por SRIBCs tanto para extração de características - como Histogramas de Cor, Descritores de Fourier, Momentos Invariantes e Wavelets - quanto para indexação de dados multidimensionais - como KDB-Tree, R-Tree, TV-Tree e, brevemente, SOMs Hierárquicos.

O próximo capítulo discute alguns SRIBCs que utilizam Mapas Auto-Organizáveis como ferramenta de indexação. Esse tipo de rede neural elimina alguns dos principais problemas existentes nas estruturas de indexação derivadas da B-Tree.

Capítulo 4

Trabalhos Relacionados

O objetivo deste capítulo é realizar um levantamento dos principais trabalhos que propõem a utilização de Mapas Auto-Organizáveis como ferramenta de indexação em Sistemas de Recuperação Baseada em Conteúdo. Os Mapas Auto-Organizáveis vêm sendo utilizado nesse tipo de sistema devido, principalmente, às limitações existentes nas estruturas de indexação tradicionais, tais como, sobreposição de regiões, armazenamento de regiões vazias e ineficiência ao lidar com altas dimensões. Os quatro trabalhos relatados nesse capítulo são divididos de acordo com o tipo de SOM utilizado para indexar as imagens.

Inicialmente, é apresentado um método de indexação de formas baseado em SOMs *com consciência* [DeSieno, 1988]. O segundo trabalho [Zhang and Zhong, 1995] propõe um novo modelo de SOM Hierárquico utilizado para indexar características visuais, como texturas, por exemplo. Finalmente, são apresentados dois sistemas que utilizam TS-SOMs [Koikkalainen and Oja, 1990; Koikkalainen, 1994] para indexar imagens genéricas e faces, respectivamente. Tais sistemas permitem a interação do usuário na busca por imagens similares.

4.1 Indexação com SOM

Suganthan [Suganthan, 2002] propôs um método para indexação de formas, baseado em Mapas Auto-Organizáveis *com consciência* [DeSieno, 1988]. Esse tipo de SOM tenta utilizar uniformemente todos os neurônios do mapa com uma probabilidade igual a $1/n$, em que n é o número de neurônios do mapa. Para cada neurônio i é associado um limiar b_i , inicializado

com o valor $1/n$. Cada vez que um neurônio i vence a competição, sua chance de vencer novamente é reduzida aumentando-se o valor do seu limiar b_i .

4.1.1 Extração de Características

No método proposto, cada forma é representada por segmentos de reta. O ponto de intersecção i entre cada par de segmentos é identificado como mostra a Figura 4.1. Em seguida, são calculados atributos que representam propriedades invariantes à orientação, escala e translação.

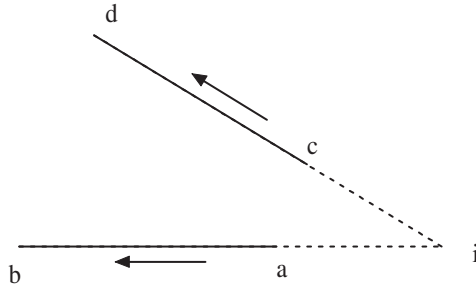


Figura 4.1: Intersecção entre segmentos.

No primeiro experimento, foram consideradas apenas propriedades invariantes à orientação e translação, constituindo vetores com 7 atributos:

1. $\theta_{\vec{ab}, \vec{cd}} = \arccos\left(\frac{\vec{ab} \cdot \vec{cd}}{|\vec{ab}| \cdot |\vec{cd}|}\right)$
2. Comprimento do segmento \vec{ab}
3. Comprimento do segmento \vec{cd}
4. Distância \vec{ac}
5. Distância \vec{bd}
6. Distância \vec{ad}
7. Distância \vec{bc}

Já no segundo experimento, foram consideradas propriedades invariantes à orientação, translação e escala, constituindo vetores com 5 atributos:

$$1. \theta_{\bar{a}\bar{b},\bar{c}\bar{d}} = \arccos\left(\frac{\bar{a}\bar{b}\cdot\bar{c}\bar{d}}{|\bar{a}\bar{b}|\cdot|\bar{c}\bar{d}|}\right)$$

$$2. \frac{1}{\frac{1}{2} + \left(\frac{\bar{b}}{\bar{a}}\right)}$$

$$3. \frac{\min(\bar{a}\bar{b},\bar{c}\bar{d})}{\max(\bar{a}\bar{b},\bar{c}\bar{d})}$$

$$4. \frac{\min(\bar{a}\bar{c},\bar{b}\bar{d})}{\max(\bar{a}\bar{c},\bar{b}\bar{d})}$$

$$5. \frac{\min(\bar{a}\bar{d},\bar{b}\bar{c})}{\max(\bar{a}\bar{d},\bar{b}\bar{c})}$$

Como para cada forma existem inúmeras combinações de pares de segmentos, os pares foram gerados com apenas 6 vizinhos mais próximos de cada segmento.

4.1.2 Construção de Histogramas

Após a extração de características de cada forma, os vetores obtidos são utilizados no treinamento de um SOM *com consciência*, denominado SOM1. Em seguida, histogramas são construídos de acordo com o Algoritmo 4.1.

Algoritmo 4.1 Construção de Histogramas.

Para cada forma,

1. Inicialize com zero um array de n elementos, em que n é igual ao número de neurônios do SOM1;
 2. Para cada vetor de característica extraído,
 - Apresente-o ao SOM1 e identifique o neurônio vencedor;
 - Incremente o elemento do array correspondente ao neurônio vencedor;
-

4.1.3 Indexação e Recuperação

Os histogramas gerados, cada um com dimensão igual ao número de neurônios do SOM1, são utilizados no treinamento de um outro SOM *com consciência*, denominado SOM2. Ao final do treinamento, cada forma é associada a três neurônios vencedores.

Na fase de recuperação, descrita pelo Algoritmo 4.2, a medida de dissimilaridade utilizada para identificação dos neurônios vencedores é a “intersecção de histogramas”, dada pela seguinte equação:

$$IH = \sum_n^{j=1} \min(Q_j, V_j) / \sum_n^{j=1} V_j$$

em que n é o número de elementos do histograma, Q é o histograma da forma de entrada, V é um vetor de pesos arbitrário do SOM2 e $\min(Q_j, V_j)$ é o menor elemento entre Q_j e V_j .

Algoritmo 4.2 Recuperação de Formas.

1. Extraia os vetores de características da forma de entrada;
 2. Apresente-os ao SOM1 e obtenha o histograma;
 3. Apresente o histograma ao SOM2 e identifique os três neurônios vencedores utilizando a “intersecção de histogramas” como medida de similaridade;
 4. Recupere todas as formas associadas a esses três neurônios eliminando as duplicações.
-

4.1.4 Experimentos

Foram realizados dois experimentos com imagens de logotipos, as quais são formadas por até 2000 segmentos de linha. No primeiro experimento, utilizando características invariantes à translação e rotação, 330 imagens originais foram variadas em escala, formando um conjunto de 990 imagens. No segundo experimento, utilizando características invariantes à translação, rotação e escala, 990 imagens originais foram utilizadas.

Os resultados obtidos nos dois experimentos, utilizando 1600 neurônios no SOM1 e 225 neurônios no SOM2, mostraram que os logotipos são recuperados de forma satisfatória mesmo quando há ruídos na imagem de entrada. A Figura 4.2 (adaptada de Suganthan [Suganthan, 2002]) ilustra o resultado da recuperação quando um logotipo com apenas 70% dos segmentos é apresentado ao SOM2.

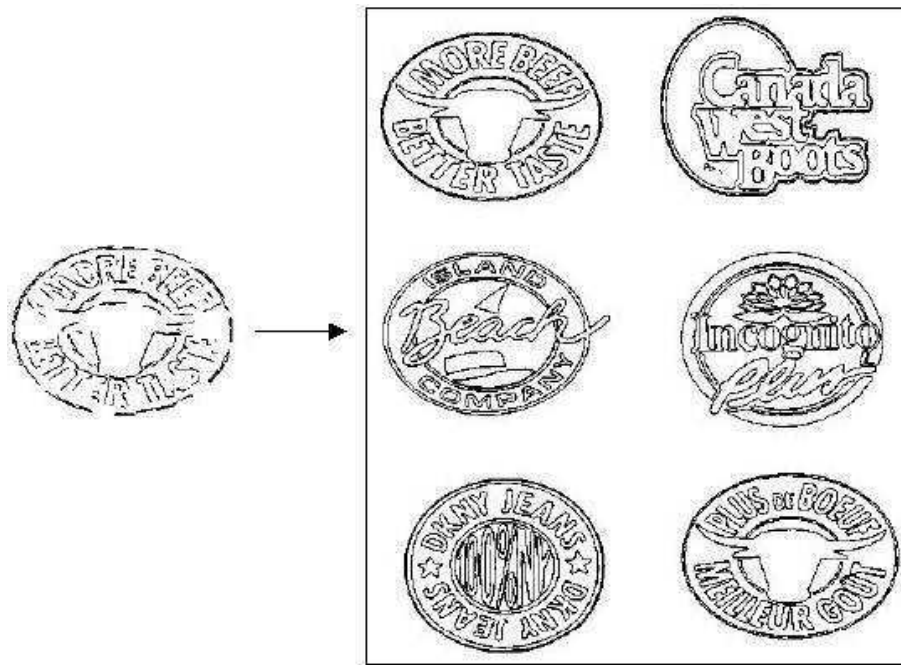


Figura 4.2: Recuperação de logotipos.

4.1.5 Discussão

O método proposto por Suganthan [Suganthan, 2002], apesar de ser robusto à ruídos da imagem de entrada e realizar quantização vetorial, possui as seguintes limitações:

1. dependendo da quantidade de segmentos de uma forma, a computação das características invariantes podem demorar muito tempo;
2. como o SOM2 utilizado é não-hierárquico, a busca por BMUs é seqüencial.

A seguir, é apresentado um trabalho que propõe um tipo de Mapa Auto-Organizável Hierárquico para indexar características visuais.

4.2 Indexação com H-SOM

Zhang e Zhong [Zhang and Zhong, 1995] propuseram um modelo de H-SOM para indexação de características visuais, o qual é construído através três etapas: auto-organização, agrupamento e projeção. A arquitetura desse modelo é ilustrada na Figura 4.3.

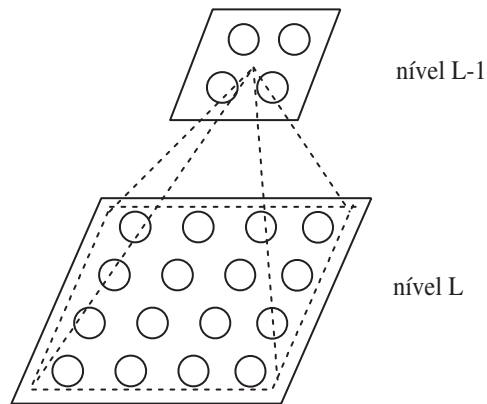


Figura 4.3: H-SOM proposto por Zhang e Zhong.

4.2.1 Agrupamento

Após o treinamento (auto-organização) da camada L^1 com o algoritmo de Kohonen, cada neurônio - que representa um conjunto de imagens similares entre si - é associado a uma classe. Em seguida, as classes são agrupadas em classes maiores, de acordo com o Algoritmo 4.3.

4.2.2 Projeção

Após o conjunto de classes K ser formado na fase de agrupamento, o mapa menor (de nível L-1) é construído de acordo com o Algoritmo 4.4.

Os processos de agrupamento e projeção podem ser aplicados iterativamente até o número de neurônios no topo da hierarquia ser menor ou igual ao número de classes existentes. Cada neurônio de uma hierarquia possui um vetor de pesos e uma lista de busca que aponta para seus filhos.

4.2.3 Recuperação

Na fase de recuperação, a busca pode ser iniciada em qualquer nível do H-SOM. Dado um vetor de entrada q , a busca por vetores similares (a partir do topo da hierarquia) é dada pelo Algoritmo 4.5. Este algoritmo é utilizado por todos os Mapas Auto-Organizáveis Hi-

¹Nesse sistema, apenas a camada L é, de fato, um Mapa Auto-Organizável.

Algoritmo 4.3 Algoritmo de agrupamento.

1. Sejam M , número máximo de neurônios em uma classe, e D , distância entre classes contíguas. Duas classes A e B são contíguas se existe no mínimo um neurônio $p \in A$ e um neurônio $q \in B$, tal que p é um dos oito vizinhos de q .
 2. Calcule a distância entre todos os pares de classes contíguas existentes.
 3. Se todas as distâncias calculadas forem maiores ou iguais a D , pare.
 4. Para cada par de classes contíguas
 - (a) Se o número de nós nas duas classes for maior que M , ajuste a distância entre elas para D ;
 - (b) Senão, agrupe as duas classes. Ajuste o vetor de pesos da nova classe com o centro da classe (i.e., o vetor média). Recalcule as distâncias entre a nova classe e seus vizinhos.
 5. Vá para o passo 3.
-

Algoritmo 4.4 Algoritmo de projeção.

1. Ajuste o tamanho do mapa de nível L-1 como igual ou um pouco maior que o número de classes K.
2. Encontre a maior classe A do conjunto de classes K.
3. Para cada neurônio da classe A

- Aloque um neurônio c na coordenada

$$i_{L-1} = \left(\frac{S_{L-1}}{S_L} \right) i_L,$$

$$j_{L-1} = \left(\frac{S_{L-1}}{S_L} \right) j_L$$

em que S_{L-1} e S_L são os tamanhos dos mapas dos níveis L-1 e L, respectivamente; e (i_L, j_L) são as coordenadas do neurônio no nível L;

- Se c não tiver sido ocupado por outra classe, ajuste o vetor de pesos de c com o vetor de pesos da classe A; Senão, ajuste o vetor de pesos de c com a média dos os vetores de A e de c .
 - Adicione os neurônios da classe A à lista de busca do neurônio c .
4. Remova a classe A do conjunto K.
 5. Se não há mais classes em K, pare; caso contrário, vá para o passo 2.
-

erárquicos descritos nesta dissertação (inclusive o GH-SOM), em que apenas o último passo pode ser alterado.

Algoritmo 4.5 Algoritmo de busca.

1. Inicialize os parâmetros l , nível inicial, igual ao topo da hierarquia e f , faixa de busca, igual ao mapa inteiro. Encontre o neurônio vencedor c_l para q .
 2. Se l é o nível mais baixo, vá para o passo 4.
 3. Desça para o próximo nível ($l+1$). Ajuste f com a lista de busca do neurônio c . Encontre o neurônio vencedor c_{l+1} para q . Vá para o passo 2.
 4. Encontre os k -vizinhos mais próximos de q associados ao neurônio c_{l+1} (e aos seus vizinhos, se o número de vetores em c_{l+1} for menor que k).
-

4.2.4 Experimentos

Zhang e Zhong [Zhang and Zhong, 1995] utilizaram um H-SOM de três camadas para indexar texturas representadas por três tipos de vetores de características: Multi-Resolução Simultânea Auto-Regressiva, representação de Tamura e histogramas de cores. Nos três experimentos realizados, foram utilizadas texturas da base Brodatz [Brodatz, 1966], a qual contém 112 imagens de 8-bits 512X512. Nove imagens 128X128 foram extraídas a partir do centro de cada imagem da base Brodatz, resultando em 112 classes com 9 imagens (sub-imagens) em cada classe.

O H-SOM utilizado nos experimentos foi construído com as seguintes características:

1. Três camadas de dimensões 30X24, 20X16 e 10X8, do nível mais baixo para o mais alto, respectivamente;
2. Número máximo de neurônios em uma classe (parâmetro M) igual a 4;
3. Distância entre classes contíguas (parâmetro D) igual a ∞ ;
4. Medidas de dissimilaridade: distância Euclidiana e distância de Mahalanobis.

No primeiro experimento, utilizou-se um vetor de características MRSAR de dimensão 20 para cada imagem, obtendo-se uma taxa de acerto de 74%. No segundo experimento, utilizou-se características MRSAR em conjunto com a representação de Tamura, resultando em um vetor de características de dimensão 24. Obteve-se, nesse experimento, uma taxa de acerto de 78%, ou seja, 4% a mais que utilizando apenas características MRSAR. No último experimento, adicionou-se ao vetor do experimento anterior um histograma com 6 níveis de cinza, resultando em um vetor de características de dimensão 30. Com essa configuração, obteve-se 80% acerto.

4.2.5 Discussão

O modelo de H-SOM proposto por Zhang e Zhong [Zhang and Zhong, 1995], ao contrário de outras estruturas hierárquicas, é construído do topo para a base, no qual apenas a base (camada L) constitui um Mapa Auto-Organizável. Na camada L, o algoritmo de Kohonen realiza uma quantização vetorial do espaço de entrada, permitindo que um único neurônio represente várias imagens. Entretanto, nas demais camadas, essa quantização é feita a partir da média dos vetores de pesos de um conjunto de neurônios, podendo gerar uma representação inadequada dos dados.

Na próxima seção, são apresentados dois SRIBCs baseados em TS-SOM, um tipo de SOM Hierárquico onde todas as camadas constituem Mapas Auto-Organizáveis de Kohonen.

4.3 Indexação com TS-SOM

Os Sistemas de Recuperação de Imagens Baseada em Conteúdo propostos por Laaksonen *et alli* [Laaksonen et al., 1999b; Laaksonen et al., 1999a] e del-Solar e Navarrete [del Solar and Navarrete, 2002] utilizam TS-SOMs para indexar imagens genéricas e faces, respectivamente.

Assim como no GH-SOM, discutido no Capítulo 2, no TS-SOM, cada nível da hierarquia é treinado de acordo com o algoritmo de Kohonen (do topo para à base). Após um número máximo de atualizações de um neurônio, os padrões mapeados nele passam a ser auto-organizados em seus filhos, com o objetivo de minimizar o erro de quantização. A principal diferença entre TS-SOM e GH-SOM é que a topologia do primeiro (i.e., número de

níveis, dimensões dos SOMs e número de filhos dos neurônios) é estática, definida antes da fase de treinamento.

4.3.1 Funcionamento dos sistemas

Tanto no sistema de Laaksonen *et alli* [Laaksonen et al., 1999b; Laaksonen et al., 1999a] quanto no sistema de del-Solar e Navarrete [del Solar and Navarrete, 2002], as buscas são iterativamente refinadas de acordo com a preferência do usuário. Os SRIBCs com essa característica geralmente seguem os seguintes passos:

1. inicialmente, um conjunto de imagens de referência (extraídas da própria base de imagens) são apresentadas ao usuário;
2. o usuário seleciona um subconjunto de imagens que correspondem às suas expectativas;
3. o sistema retorna ao usuário um conjunto de imagens similares às imagens selecionadas;
4. a iteração continua até o usuário obter as imagens desejadas.

No sistema de Laaksonen *et alli* [Laaksonen et al., 1999b; Laaksonen et al., 1999a], o *PicSOM*, imagens genéricas são representadas por suas características de cor, textura e forma, as quais são indexadas por três TS-SOMs diferentes, ou seja, é utilizado um TS-SOM para cada tipo de vetor de características.

Na fase de recuperação, as respostas dos três TS-SOMs (as imagens associadas a k -BMUs) são combinadas e apresentadas ao usuário. A Figura 4.4 ilustra o funcionamento desse sistema.

O funcionamento do sistema de del-Solar e Navarrete [del Solar and Navarrete, 2002] é similar ao do *PicSOM*, exceto o fato de apenas um TS-SOM ser utilizado para indexar imagens de faces, representadas por projeções-PCA.

4.3.2 Experimentos

No *PicSOM*, experimentos foram realizados com 4350 imagens genéricas obtidas em <ftp://ftp.sunet.se/pub/pictures>. Três tipos de vetores de características foram extraídos de

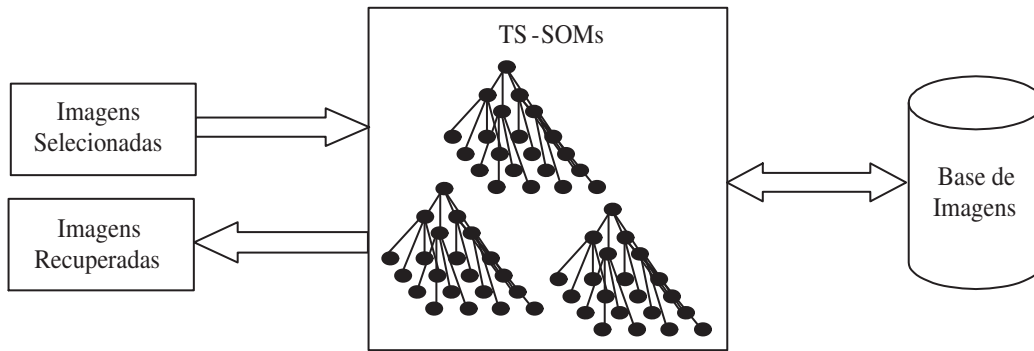


Figura 4.4: Diagrama de blocos do PicSOM.

cinzo zonas das imagens (i.e., direita, esquerda, centro, topo, base) da seguinte forma:

- *vetor cor* - a média dos valores RGB foram calculadas, resultando em um vetor de dimensão 15.
- *vetor textura* - a luminância de cada píxel foi comparada com as dos seus 8 vizinhos. A probabilidade estimada de que o píxel central tem um valor maior que os seus vizinhos foi usada como informação de textura, resultando em um vetor de dimensão 40.
- *vetor forma* - primeiramente, as bordas das imagens (binarizadas) foram extraídas através de máscaras de Sobel 3x3. As direções das bordas foram então discretizadas para 8 valores e, em seguida, um histograma de 8 níveis foi gerado a partir desses valores, resultando em um vetor de dimensão 40.

Cada tipo de característica foi mapeada em um TS-SOM diferente, composto por três camadas de dimensões 4X4, 16X16 e 64X64. Com essa configuração, cada neurônio em uma hierarquia possui 4 filhos.

Das 4350 imagens utilizadas para treinamento, 1200 foram também utilizadas para teste. A performance do sistema foi calculada dividindo-se o número de imagens recuperadas pelo sistema até se atingir a resposta desejada, pelo número total de imagens do BD. Os resultados mostraram que o sistema geralmente obtém a melhor performance ao combinar as respostas dos três TS-SOMs. A Figura 4.5 ilustra o procedimento utilizado nos testes.

No sistema de del-Solar e Navarrete [del Solar and Navarrete, 2002], 684 imagens extraídas da base FERET [Phillips et al., 1998] foram mapeadas em um TS-SOM de 6 níveis

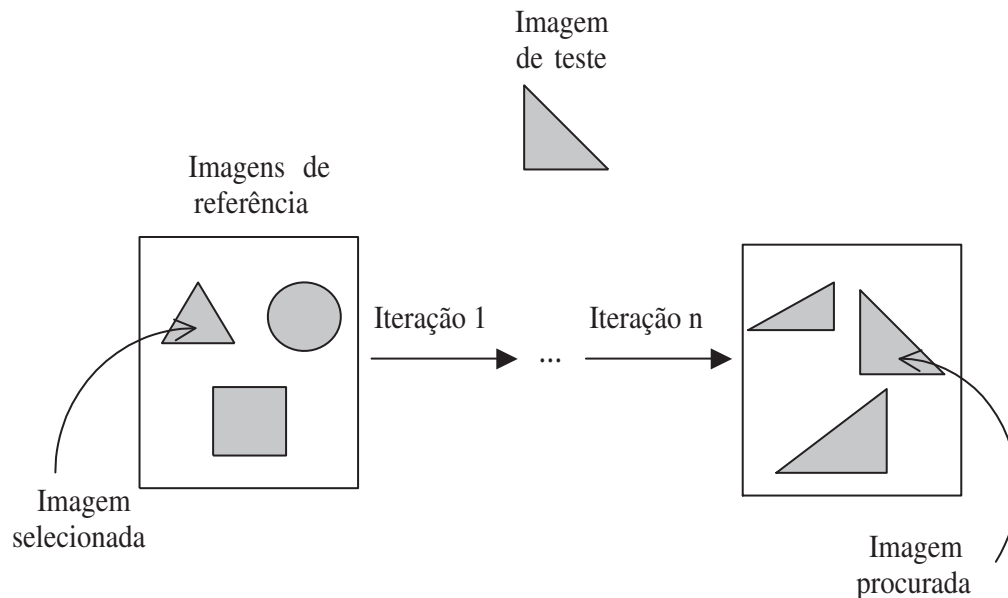


Figura 4.5: Procedimento utilizado nos testes.

com 32X32 neurônios na base. Os testes realizados, utilizando o mesmo procedimento do PicSOM, mostraram que a imagem procurada é retornada pelo sistema em poucas iterações.

4.3.3 Discussão

Os sistemas apresentados nesta seção, apesar de utilizarem TS-SOMs - nos quais todas as camadas da hierarquia constituem Mapas Auto-Organizáveis de Kohonen - não fazem uso da propriedade de quantização vetorial para a redução do espaço de entrada. O uso dessa propriedade é vantajoso em SRIBCs com um grande número de imagens, uma vez que é possível mapear diversas imagens similares em um único neurônio. Assim, a árvore de indexação é reduzida e, conseqüentemente, a busca por imagens similares é mais rápida.

4.4 Considerações Finais

Neste capítulo, foram analisados quatro trabalhos que tratam da indexação de imagens em SRIBCs através de Mapas Auto-Organizáveis. A descrição de cada um deles foi apresentada de acordo com o tipo de SOM utilizado. Tais trabalhos, apesar de apresentarem estratégias que resolvem muitos dos problemas existentes nas estruturas de indexação tradicionais (dis-

cutidas no Capítulo 3), não fazem uso de importantes características dos SOMs, as quais poderiam melhorar a performance do SRIBC.

Um outro ponto a ser observado, é que, em todos esses trabalhos, foram realizados experimentos com imagens extraídas do próprio conjunto de treinamento (ou com pequenas variações delas). No caso dos sistemas propostos por Laaksonen *et alli* [Laaksonen et al., 1999b; Laaksonen et al., 1999a] e del-Solar e Navarrete [del Solar and Navarrete, 2002], por serem SRIBCs de um tipo específico, não é permitido que o usuário forneça ao sistema uma imagem externa.

No próximo capítulo, é proposto um SRIBC que combina representação de imagem log-polar com GH-SOM na tentativa de solucionar os problemas existentes nesse tipo de sistema.

Capítulo 5

Sistema Proposto

Neste capítulo, é proposto um Sistema de Recuperação de Imagens Baseada em Conteúdo que utiliza um GH-SOM, como estrutura de indexação, e uma representação de imagem log-polar, inspirada na retina humana. Inicialmente, é descrito o funcionamento da representação log-polar que, por possuir a propriedade de conversão de mudanças de escala e orientação no espaço de entrada em translações no espaço log-polar, facilita o reconhecimento de objetos em diferentes escalas e orientações. A partir dessa representação, é apresentada uma nova estratégia de classificação, indexação e recuperação, na qual apenas as imagens mais representativas de cada classe são utilizadas na fase de treinamento.

5.1 Representação Log-Polar

Na retina humana, os sinais luminosos são capturados por fotorreceptores (células sensíveis à luz) e, em seguida, transmitidos aos neurônios localizados na própria retina. Cada um desses neurônios recebe as saídas de vários fotorreceptores sobre uma área aproximadamente circular da retina, chamada de campo receptivo. O objetivo desses neurônios é realizar um processamento inicial nos sinais luminosos antes de enviá-los ao córtex visual.

No modelo de retina artificial proposto por Gomes e Fisher [Gomes et al., 1998], a fóvea (região central) é formada por campos receptivos de tamanho aproximadamente constante e organizados hexagonalmente; enquanto que na região mais externa, os campos receptivos são distribuídos circularmente com uma área decrescendo exponencialmente a medida em que se aproxima do centro da retina. Existe uma intersecção entre os campos receptivos e

seus vizinhos mais próximos que previne algumas áreas da imagem de não serem cobertas pela retina (devido à geometria circular dos campos receptivos).

A Figura 5.1, extraída com permissão do trabalho de Gomes e Fisher [Gomes and Fisher, 2001], ilustra a estrutura da retina utilizada neste trabalho, denominada “máscara retinal”.

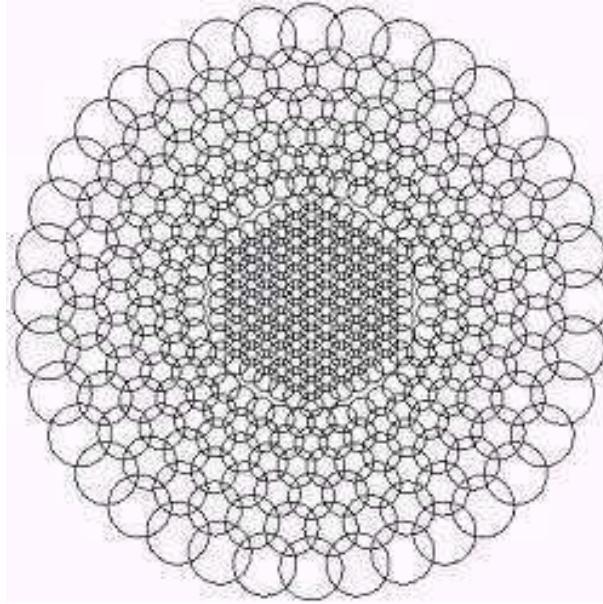


Figura 5.1: Estrutura da máscara retinal utilizada.

Quando a máscara retinal é aplicada sobre uma imagem, cada campo receptivo corresponde a um conjunto menor (na fóvea) ou maior (na periferia) de pixels, ocasionando uma perda de resolução à medida em que se afasta do centro da imagem retinal.

Uma imagem log-polar é basicamente uma imagem Cartesiana derivada da máscara retinal, onde o *eixo x* corresponde aos logaritmos das distâncias dos anéis ao centro da retina e o *eixo y* corresponde aos setores da máscara. A Figura 5.2, adaptada de [Gomes and Fisher, 2003], ilustra o mapeamento log-polar de uma retina hipotética com 5 anéis, dos quais 3 estão localizados na fóvea.

O valor do campo receptivo $V(n, s)$ indexado pelo anel n e pelo setor s é computado da seguinte forma:

$$V(n, s) = O((n, s), r), \quad 0 \leq n < N, \quad 0 \leq s < f(n) \quad (5.1)$$

em que (n, s) é o centro do campo receptivo em coordenadas Cartesianas, $r(n)$ é o raio dos

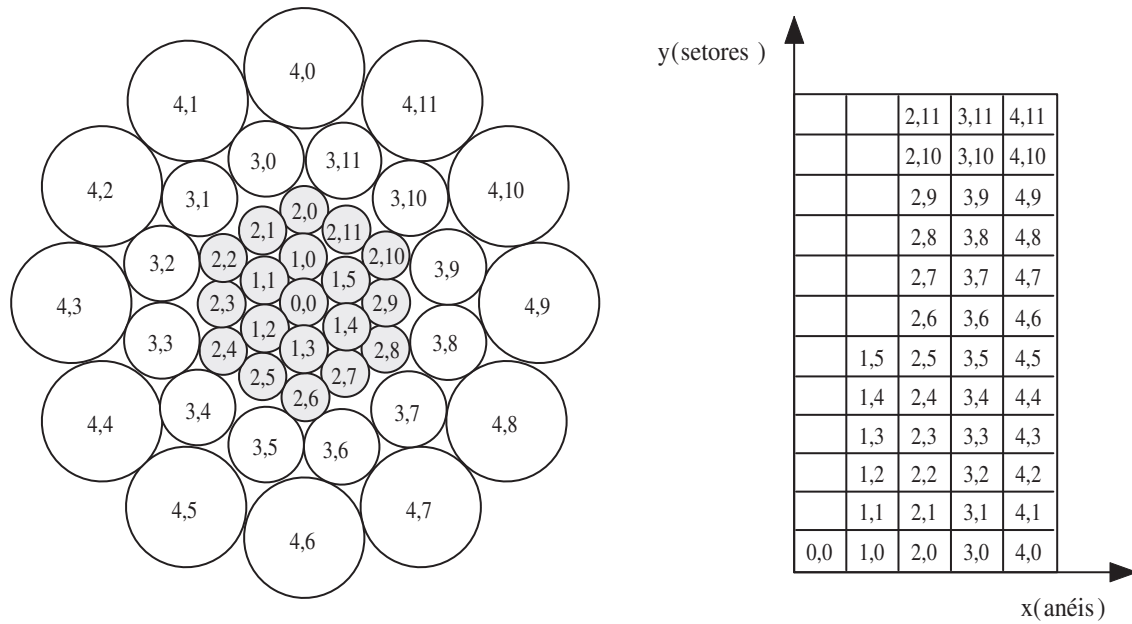


Figura 5.2: Transformação log-polar.

campos receptivos no anel n , N é o número total de anéis na retina (incluindo a fóvea) e $f(n)$ é o número de campos receptivos no anel n . A função $O((n, s), r)$, que representa a saída de um dado campo receptivo, é definida pela seguinte equação:

$$O((n, s), r) = \log(E) + \sum_{x=-r}^r \sum_{y=-r}^r \log(R(n+x, s+y)) \cdot F(x, y), \quad x^2 + y^2 \leq r^2 \quad (5.2)$$

em que E é a luminância incidindo sobre o objeto, $R(n+x, s+y)$ é a reflectância local da superfície e $F(x, y)$ é a função do campo receptivo, definida como uma Gaussiana normalizada aplicada aos pontos (x, y) no domínio circular do campo receptivo de raio r .

Esse tipo de representação possui a propriedade de conversão de mudanças de escala e orientação no espaço de entrada em translações no espaço log-polar. É possível tirar vantagem dessa propriedade no reconhecimento de objetos em diferentes escalas e orientações, uma vez que as variações no espaço log-polar são bem menores que as variações no espaço de entrada.

A Figura 5.3 apresenta uma imagem em três formatos distintos: original, retinal e log-polar. Por questão de simplicidade de manipulação e por representar uma porção muito pequena da imagem, a região correspondente à fóvea (área triangular no topo da imagem (c)

da Figura 5.3) foi removida.

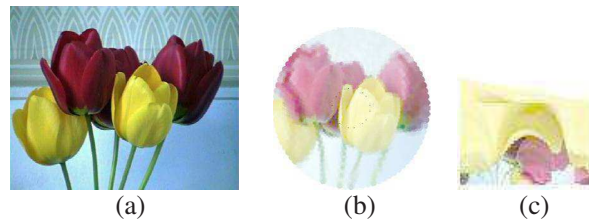


Figura 5.3: Exemplo de transformação log-polar. (a)Imagem original. (b)Imagem retinal. (c)Imagem log-polar.

A máscara retinal utilizada neste trabalho possui 68 anéis de campos receptivos em toda a retina, dos quais 15 deles estão localizados na fóvea. Isso corresponde a uma máscara que cobre um espaço Cartesiano de 88x88 pixels (da imagem de entrada) e produz uma imagem log-polar com 84x53 pixels, representando uma redução do tamanho da imagem de entrada em aproximadamente 57,5%.

5.2 Funcionamento do Sistema

Nesta seção, é proposto um sistema RIBC que pode ser visto como um *sistema de espaço de características invariantes* (ver Seção 2.8), em que apenas as imagens mais representativas de cada classe são utilizadas na fase de treinamento¹.

A utilização de imagens do tipo log-polar em níveis de cinza, permite que variações de orientação, escala e intensidade de cor de um mesmo objeto sejam excluídas da fase de treinamento. Além disso, dependendo dos parâmetros utilizados na máscara retinal, objetos de uma mesma classe podem possuir imagens log-polares muito próximas, não havendo, portanto, a necessidade de incluir todas elas nessa fase.

A principal vantagem dessa estratégia é a redução significativa do tempo de treinamento, da árvore de indexação - que nesse sistema é representada por um GH-SOM - e, conseqüentemente, do tempo de recuperação.

¹Uma forma de se definir o conjunto de treinamento é, para cada classe, escolher imagens bastante distintas umas das outras, com o objetivo de se obter a representação mais abrangente possível.

Após a fase de treinamento, um mapa contextual é construído de forma que todos os neurônios com o mesmo rótulo pertençam à mesma classe. A Figura 5.4 ilustra um mapa contextual hipotético de dimensão 8X8, no qual quatro imagens - cada uma representando uma classe - foram mapeadas. Os círculos menores representam as BMUs de cada imagem de treinamento e os círculos maiores representam as “bolhas de atividade” de cada classe.

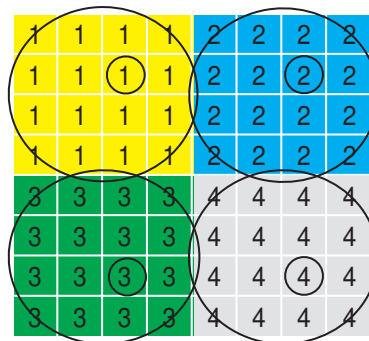


Figura 5.4: Mapa contextual representando quatro classes.

Em seguida, cada imagem que não foi utilizada na fase de treinamento, é transformada em uma imagem log-polar e propagada no GH-SOM. Se ela não for parecida com nenhuma das imagens mapeadas durante o treinamento, não haverá neurônio vencedor. Então, são geradas variações de escala e orientação dessa imagem com o intuito de determinar se ela já foi vista anteriormente em uma escala e/ou orientação diferentes. Em seguida, essas variações são propagadas no GH-SOM. Se uma dessas variações for parecida com alguma imagem de treinamento, a referência para a imagem de entrada é armazenada na base de dados e indexada pelo neurônio vencedor. Este procedimento é ilustrado na Figura 5.5 e detalhado no Algoritmo 5.1.

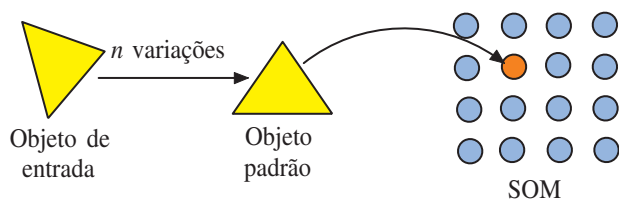


Figura 5.5: Estratégia de classificação/indexação.

As variações de orientação e escala das imagens são obtidas através de translações hori-

zontais e verticais, respectivamente, no espaço log-polar. O deslocamento de uma coluna de pixels, à direita ou à esquerda, em uma imagem log-polar de 84X53 pixels, equivale a girar a imagem original aproximadamente $\pm 4,3^\circ$, respectivamente. Enquanto que o deslocamento de uma linha de pixels, para cima ou para baixo, equivale a diminuir ou aumentar a imagem original em aproximadamente 3,8%, respectivamente.

Algoritmo 5.1 Algoritmo de Classificação/Indexação.

Para cada imagem de entrada X

1. Transforme X em uma imagem log-polar X_{LP} com 256 níveis de cinza
 2. Propague X_{LP} no mapa e obtenha um neurônio vencedor i
 3. Se a distância Euclidiana d entre i e X_{LP} for menor ou igual a um limiar L_1
 - $BMU = i$
 - $dist = d$
 - $classe = rótulo(BMU)$ // O rótulo da BMU é obtido a partir do mapa contextual
 - Armazene X^* , BMU , $Dist$, $Classe$ // X^* é a referência para X .
 4. Senão
 - Gere variações de escala/orientação de X_{LP} , propague-as no mapa e obtenha os neurônios vencedores
 - Determine a variação (de escala e/ou orientação) X'_{LP} que obteve a menor distância Euclidiana d' do seu neurônio vencedor correspondente i'
 - Se d' for menor ou igual a um limiar L_2
 - $BMU = i'$
 - $dist = d'$
 - $classe = rótulo(BMU)$
 - Armazene X^* , BMU , $Dist$, $Classe$
-

Antes de serem transformadas em log-polar, as imagens cujas áreas não são totalmente cobertas pela máscara retinal utilizada são redimensionadas, conservando suas proporções

(altura/largura). Ou seja, como a máscara retinal utilizada cobre um espaço equivalente a uma imagem de 88X88 pixels, esse redimensionamento é feito de forma que a maior dimensão da imagem possua 88 pixels. As novas dimensões da imagem são calculadas da seguinte forma:

1. se (altura = largura),
 - altura' = 88;
 - largura' = 88;
2. senão
 - se (altura > largura),
 - altura' = 88;
 - largura' = $\text{div}(\text{largura}, \text{altura}) * 88$;
 - senão, //se (altura < largura)
 - largura' = 88;
 - altura' = $\text{div}(\text{altura}, \text{largura}) * 88$;

Em que $\text{div}(\text{altura}, \text{largura})$ representa a divisão inteira entre o número de linhas (altura) e colunas (largura) da imagem original e altura' e $\text{largura}'$ são as novas dimensões.

As referências para as imagens são armazenadas em uma tabela de um Banco de Dados Relacional, chamada IMAGEM, a qual possui os seguintes atributos:

- X^* → referência para a imagem de entrada X ;
- BMU → BMU da imagem X ;
- $Dist$ → distância euclidiana (ao quadrado) entre X e sua BMU;
- $Classe$ → classe a qual pertence a BMU de X .

Portanto, o passo “Armazene X^* , BMU , $Dist$, $Classe$ ”, do Algoritmo 5.1, é executado através de um simples comando SQL do tipo INSERT (ex., **insert into** IMAGEM (X^* , BMU , $Dist$, $Classe$) **values** ('img1.jpg', 10, 0.001, 5)).

Na fase de utilização (recuperação), o mesmo processo é repetido, porém, ao invés de se armazenar a referência para a imagem de entrada, retorna-se todas as imagens similares a ela através de comandos SQL do tipo SELECT (ex., **select X* from IMAGEM where BMU = 10**). O grau de similaridade utilizado nessa fase é especificado pelo usuário através da interface do sistema. Ou seja, pode-se informar um intervalo de distância desejado entre a imagem de entrada e as imagens a serem recuperadas. Além disso, podem ser recuperadas todas as imagens pertencentes a uma mesma BMU ou a uma mesma classe.

5.3 Considerações Finais

Neste capítulo, foi proposto um SRIBC que combina um GH-SOM - utilizado como ferramenta de classificação/indexação - com a representação de imagem log-polar. Esse sistema possui três novidades com relação aos sistemas descritos no Capítulo 4:

1. Um pequeno conjunto de imagens de cada classe é escolhido para representá-las, permitindo uma redução significativa do tempo de treinamento (uma vez que esse tempo é proporcional à dimensão de entrada) e da árvore de indexação.
2. A utilização de uma representação de imagem inspirada na retina humana facilita o reconhecimento de imagens independente de orientação e escala, além de permitir uma compactação da imagem original.
3. Como, nos GH-SOMs, as dimensões dos SOMs, bem como a profundidade da hierarquia são determinadas automaticamente durante a fase de treinamento, pouco tempo é gasto na obtenção da arquitetura mais adequada (como constatado nos experimentos das seções 6.3, 6.4 e 6.5).

Como pudemos observar, o sistema proposto não utiliza características visuais do tipo cor, forma e textura, utilizadas pela abordagem tradicional - incluindo os sistemas descritos no Capítulo 4. Nesses sistemas, em geral, as transformações sofridas pelas imagens podem acarretar na perda de informações valiosas para sua correta identificação no futuro.

No próximo capítulo, são apresentados os principais experimentos realizados no decorrer desta pesquisa, os quais foram fundamentais na formulação da melhor estratégia de classifi-

cação. Além disso, é apresentado um estudo de caso no qual o sistema proposto é adaptado e aplicado na busca de imagens da Web.

Capítulo 6

Experimentos e Resultados

Este capítulo apresenta os principais experimentos realizados no decorrer desta pesquisa. Os três primeiros experimentos tratam da classificação de imagens incorporando invariâncias à orientação e escala. No quarto experimento, comparamos a performance do SOM com o GH-SOM no problema de reconhecimento de dígitos manuscritos. Em seguida, comparamos a representação de imagem utilizada, a log-polar, com duas características muito utilizadas em Sistemas de Recuperação de Imagens Baseada em Conteúdo: histogramas e formas. O capítulo é concluído com dois estudos de caso nos quais o sistema proposto é adaptado e utilizado na busca de imagens da Web.

6.1 Incorporando Invariâncias por Treinamento (Experimento 1)

O objetivo deste experimento foi responder à seguinte questão: seria possível obter um classificador (um Mapa Auto-Organizável) invariante à orientação a partir do agrupamento de objetos em diferentes orientações de acordo com suas classes?

A fim de se obter condições experimentais controladas, foram utilizadas imagens originalmente adquiridas contra um *background* escuro. Tais imagens foram obtidas da base COIL20 da Universidade de Columbia, disponível para download em www1.cs.columbia.edu/CAVE/research/softlib/coil-20.html. A base COIL20 é composta por 20 classes de objetos, cada uma com 72 imagens de 128X128 pixels em 256 níveis

de cinza.

Das vinte classes disponíveis, foram utilizadas apenas quatro delas (redimensionadas para 88X88 pixels) como mostra a Figura 6.1.

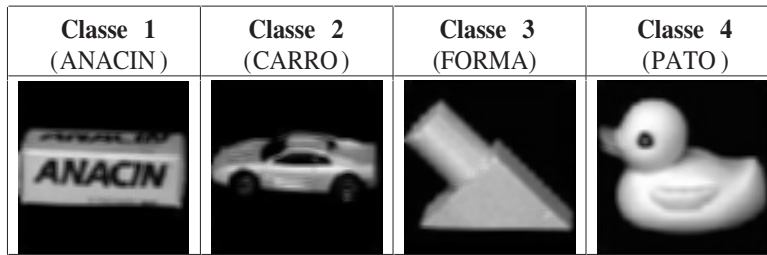


Figura 6.1: Classes de objetos do Experimento 1.

Um SOM 14X14 foi treinado com 6 variações de orientação (0° , 90° , -90° , 180° , 30° , -30°) de cada uma das 4 classes, totalizando 24 imagens¹. Considerando o número de imagens de treinamento (24), bem como a geração de “bolhas de atividade”, um número grande de neurônios foi necessário neste experimento, ou seja, aproximadamente 8 neurônios por imagem. A Figura 6.2 apresenta as diferentes orientações utilizadas no treinamento da Classe 4 (PATO).

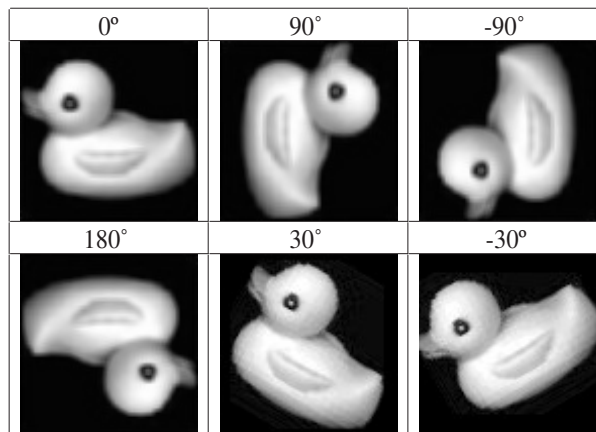


Figura 6.2: Imagens de treinamento da Classe 4.

¹É importante observar que as variações 90° , -90° , 180° , 30° e -30° não foram extraídas da base COIL20, uma vez que essa base é composta por “aparências” de objetos. Ou seja, utilizamos apenas as imagens na orientação 0° , da base COIL20, para gerar as variações utilizadas neste experimento.

	0°	90°	-90°	180°	30°	-30°
ANACIN	P1	P5	P9	P13	P17	P21
CARRO	P2	P6	P10	P14	P18	P22
FORMA	P3	P7	P11	P15	P19	P23
PATO	P4	P8	P12	P16	P20	P24

Tabela 6.1: Rótulos dos padrões de treinamento.

O resultado do agrupamento é visualizado a partir do Mapa Contextual ilustrado na Figura 6.3. A Classe 1 (ANACIN) é representada pelas regiões de cor branca; a Classe 2 (CARRO), pelas regiões de cor cinza-claro; a Classe 3 (FORMA), pelas regiões de cor cinza-escuro; e a Classe 4 (PATO), pelas regiões de cor preta. A Tabela 6.1 - a qual apresenta os rótulos dos padrões de treinamento² e suas respectivas classes e orientações - facilita a leitura da Figura 6.3.

P20	P20	P22	P22	P22	P22	P2	P2	P2	P2	P1	P1	P1	P1
P20	P20	P22	P22	P22	P14	P2	P2	P2	P18	P18	P1	P1	P1
P16	P16	P22	P22	P14	P14	P14	P2	P18	P18	P18	P18	P21	P21
P16	P16	P16	P14	P14	P14	P14	P18	P18	P18	P18	P18	P21	P21
P7	P7	P7	P14	P14	P14	P14	P18	P18	P18	P18	P21	P21	P21
P7	P7	P7	P13	P11	P11	P11	P11	P18	P18	P19	P19	P9	P9
P7	P7	P13	P13	P13	P11	P11	P11	P11	P19	P19	P19	P9	P9
P6	P6	P13	P13	P13	P11	P11	P11	P23	P19	P19	P19	P9	P9
P6	P6	P13	P13	P13	P3	P3	P23	P23	P23	P19	P19	P5	P5
P6	P6	P6	P3	P3	P3	P3	P23	P23	P23	P23	P5	P5	P5
P12	P12	P4	P4	P3	P3	P3	P23	P23	P23	P8	P8	P5	P5
P12	P12	P4	P4	P4	P3	P3	P24	P23	P8	P8	P8	P8	P5
P17	P17	P15	P15	P15	P4	P24	P24	P24	P8	P8	P8	P10	P10
P17	P17	P15	P15	P15	P15	P24	P24	P24	P24	P8	P10	P10	P10

Figura 6.3: Mapa Contextual. Cor branca = Classe 1 (ANACIN); cor cinza-claro = Classe 2 (CARRO); cor cinza-escuro = Classe 3 (FORMA); cor preta = Classe 4 (PATO).

Analizando-se o Mapa Contextual, observa-se que vários padrões de uma mesma classe

²Os rótulos dos padrões de treinamento são representados pela letra *P* (de padrão) e por um valor que vai de 1 a *n*, onde *n* é o número de padrões utilizados no experimento.

foram mapeados em regiões distantes no mapa. Além disso, nota-se uma certa confusão no mapa em agrupar por orientação ou por classe de objetos. A Figura 6.4 e a Tabela 6.2 identificam os agrupamentos por orientação. As regiões de cor branca correspondem aos agrupamentos na orientação 0° ; as regiões de cor cinza-claro, aos agrupamentos na orientação 90° ; a região de cor cinza-médio, ao agrupamento na orientação 180° ; a região de cor cinza-escuro, ao agrupamento na orientação 30° ; e a região de cor preta, ao agrupamento na orientação -30° . Observe que as regiões de cor branca que não possuem rótulos indicam que não foram formados agrupamentos por orientação.

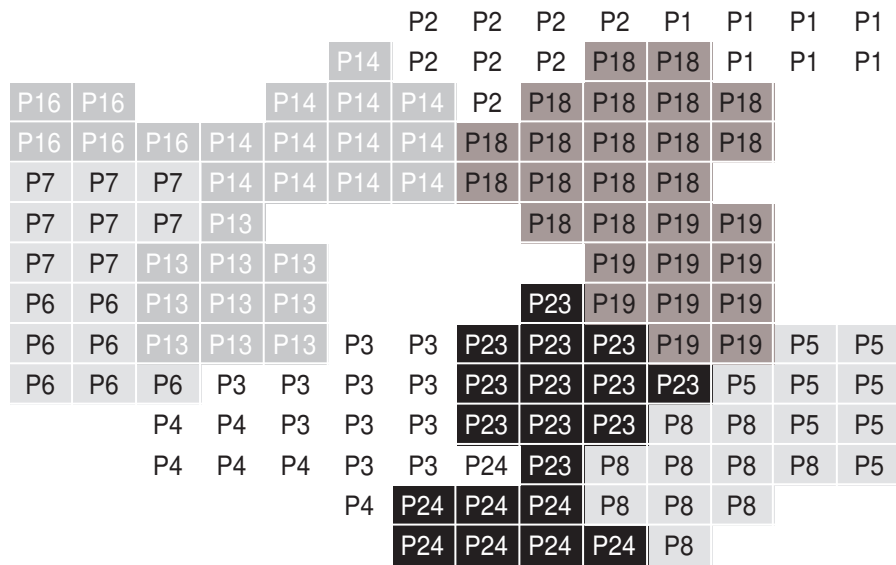


Figura 6.4: Mapa Contextual enfatizando os agrupamentos por orientação. Cor branca = 0° ; cor cinza-claro = 90° ; cor cinza-médio = 180° ; cor cinza-escuro = 30° ; cor preta = -30° .

A partir deste experimento, pudemos concluir que a invariância *por treinamento* não é adequada para sistemas RIBC que utilizam SOMs como ferramenta de indexação, já que a busca por “k-vizinhos mais próximos” pode ser comprometida por agrupamentos mal formados. A próxima seção apresenta experimentos nos quais este problema é resolvido a partir da utilização de um *espaço de características invariantes*.

Orientação	Padrões
0°	P1, P2
0°	P3, P4
90°	P5, P8
90°	P6, P7
180°	P13, P14, P16
30°	P18, P19
-30°	P23, P24

Tabela 6.2: Agrupamentos por orientação.

6.2 Incorporando Invariâncias por Espaço de Características

Como discutido na Seção 2.8, a invariância *por espaço de características* é obtida apresentando-se ao classificador apenas um conjunto de características invariantes que representa a informação essencial do conjunto de treinamento.

Os dois experimentos apresentados nesta seção utilizam o algoritmo de classificação proposto no Capítulo 5 para reconhecer - de forma invariante à orientação e escala - imagens da base COIL20 e imagens genéricas, respectivamente. Os limiares L_1 e L_2 do algoritmo foram escolhidos como “zero” e “infinito”, respectivamente. O limiar L_1 igual a zero significa que a imagem de teste só não sofrerá transformações (de orientação e escala) quando a distância Euclidiana entre ela e sua BMU for igual a zero. Já o limiar L_2 igual a infinito, significa que toda imagem de teste terá uma BMU no SOM, independentemente do valor da distância Euclidiana.

6.2.1 Experimento 2

O principal objetivo deste experimento foi investigar a viabilidade da utilização de imagens log-polares no reconhecimento de objetos em múltiplas orientações e escalas.

Um SOM 10x10 foi treinado durante 1500 épocas com apenas 4 imagens log-polares, cada uma representando uma classe (visando obter cerca de 25 neurônios por classe). A

Figura 6.5 apresenta as imagens originais (na primeira linha) - extraídas da base COIL20 - e suas respectivas imagens log-polares (na segunda linha) utilizadas neste experimento. Para reduzir o tempo de treinamento, as imagens log-polares foram redimensionadas de 84X53 pixels para 29X18 pixels. O redimensionamento foi feito de forma que a maior dimensão da imagem D_1 possuísse 29 pixels. Já a menor dimensão D_2 foi calculada como $div(largura, altura) * 29$, em que *altura* e *largura* são o número de linhas e colunas da imagem log-polar original, respectivamente.

O Mapa Contextual é apresentado na Figura 6.6. A Classe 1 é representada pela região branca (rotulada pelo valor 1); a Classe 2, pela região de cor cinza-claro (rotulada pelo valor 2); a Classe 3, pela região de cor cinza-escuro (rotulada pelo valor 3); e a Classe 4, pelas regiões de cor preta (rotulada pelo valor 4).

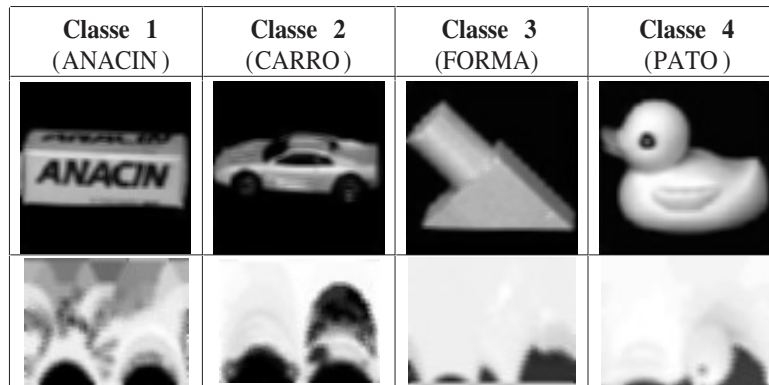


Figura 6.5: Imagens de treinamento do Experimento 2.

Para cada padrão de treinamento, foram escolhidas aleatoriamente 3 variações de escala e orientação para formarem o conjunto de teste (i.e., 12 padrões de teste). Em seguida, 195 variações de cada padrão de teste - resultantes da combinação de 15 variações de orientação com 13 variações de escala - foram apresentadas ao SOM. As variações de orientação foram geradas a partir do deslocamento de 2 em 2 pixels à direita, correspondendo a $+24,8^\circ$ em cada deslocamento. Enquanto que as variações de escala foram geradas a partir do deslocamento de 1 em 1 pixel para baixo e para cima, correspondendo a $\pm 10,5\%$ em cada deslocamento.

A Tabela 6.3 apresenta a variação vencedora para cada padrão de teste. A coluna “Padrão” indica o nome do padrão de entrada e, entre colchetes, os valores de variação de orientação (em graus) e escala (fator de magnificação/redução), respectivamente. A colu-

1	1	1	1	1	4	4	4	4	4
1	1	1	1	1	4	4	4	4	4
1	1	1	1	3	3	3	3	4	4
1	1	1	3	3	3	3	3	4	4
4	3	3	3	3	3	3	3	3	3
4	4	4	3	3	3	3	3	2	2
4	4	4	3	3	3	3	3	2	2
4	4	4	3	3	3	3	2	2	2
4	4	4	3	3	3	3	2	2	2
4	4	4	3	3	3	3	2	2	2

Figura 6.6: Mapa Contextual ilustrando 4 classes. Cor branca = Classe 1 (ANACIN); cinza-claro = Classe 2 (CARRO); cor cinza-escuro Classe 3 (FORMA); cor preta Classe 4 (PATO).

na “Variação” indica qual variação do padrão de entrada foi reconhecida pelo SOM (o sinal +/- indica o acréscimo/descréscimo de escala em relação ao tamanho original para o objeto ser reconhecido). Por exemplo, ANACIN[0°;1,17], que representa o objeto ANACIN em sua orientação original mais um aumento de 17% em tamanho, teve que ser diminuído em 17% para ser reconhecido. Já CARRO[137°;1,00] teve que ser rotacionado em 223° para ser reconhecido (de fato, $137^\circ + 223^\circ = 360^\circ$).

As taxas de acerto de *classificação*, *orientação* (considerando um erro máximo igual a $\pm 24,8^\circ$, que corresponde a 1 deslocamento de orientação) e *escala* (considerando um erro máximo igual a $\pm 10,5\%$, que corresponde a 1 deslocamento de escala) são apresentadas na Tabela 6.4.

As taxas de acerto relativamente altas obtidas neste experimento mostraram que, mesmo com dimensões reduzidas³, as imagens log-polares são úteis no reconhecimento invariante a orientação e escala.

6.2.2 Experimento 3

O objetivo deste experimento foi utilizar o algoritmo de classificação proposto para reconhecer imagens arbitrárias, extraídas da base SUNET (Swedish University Network’s)

³A dimensão das imagens log-polares utilizadas neste experimento, 29X18 pixels, representa apenas 6,74% do tamanho das imagens originais, as quais possuem 88X88 pixels.

Padrão	Variação	BMU	Classe
ANACIN[0°;1,17]	0°; 0,17-	32	1
ANACIN[62°;0,50]	298°; 0,50+	4	1
ANACIN[310°;1,50]	322°; 1,00	5	1
CARRO[137°;1,00]	223°; 1,00	59	2
CARRO[211°;1,25]	149°; 0,25-	78	2
CARRO[335°;0,34]	25°; 0,66+	78	2
FORMA[0°;0,35]	0°; 0,65+	38	3
FORMA[124°;0,25]	248°; 0,66+	49	3
FORMA[248°;1,17]	124°; 1,00	49	3
PATO[347°;1,00]	25°; 1,00	53	4
PATO[37°;1,34]	347°; 1,00	6	4
PATO[186°;0,42]	174°; 0,58+	62	4

Tabela 6.3: Padrões de teste com suas variações vencedoras.

Classificação	100,00%
Orientação	91,66 %
Escala	75,00 %
Classificação + Orientação + Escala	75,00 %

Tabela 6.4: Taxas de acerto.

disponível em <ftp://ftp.sunet.se/pub/pictures>. A base de imagens SUNET é composta por 36 classes de imagens, nas quais não há distinção entre objetos e *background*.

A Figura 6.7 apresenta as 3 imagens utilizadas no treinamento e as Figuras 6.8, 6.9 e 6.10, as imagens de teste. Nessas figuras, linhas contendo imagens originais e sua representação log-polar (84X53 pixels)⁴ são intercaladas.



Figura 6.7: Imagens de treinamento do Experimento 3.

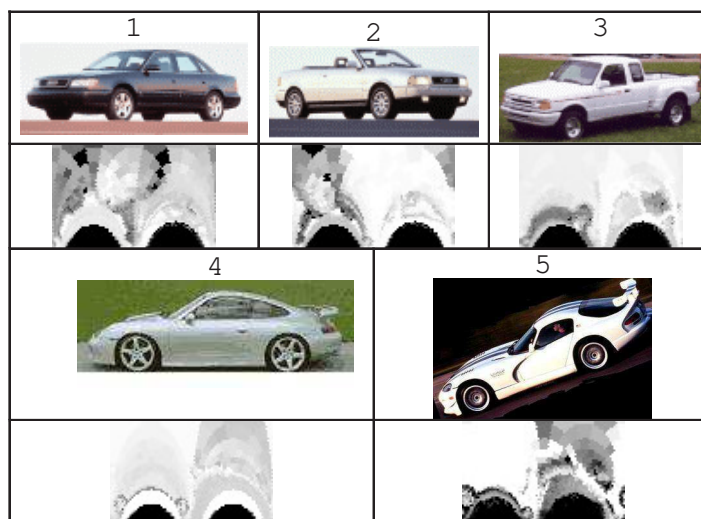


Figura 6.8: Imagens de teste da Classe 1.

O treinamento foi realizado com um SOM 6x6 (visando obter cerca de 12 neurônios por classe) durante 2000 épocas. O Mapa Contextual obtido é ilustrado na Figura 6.11. A Classe

⁴Antes de serem transformadas para log-polar, as imagens cujas áreas não eram totalmente cobertas pela máscara retinal utilizada foram redimensionadas, conservando suas proporções (ver Capítulo 5).

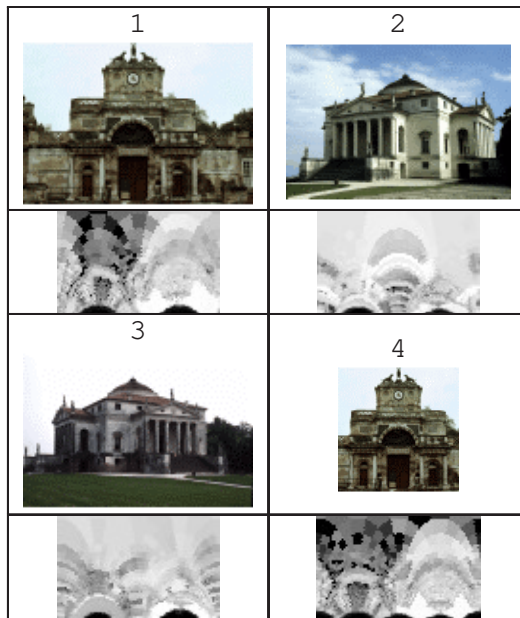


Figura 6.9: Imagens de teste da Classe 2.

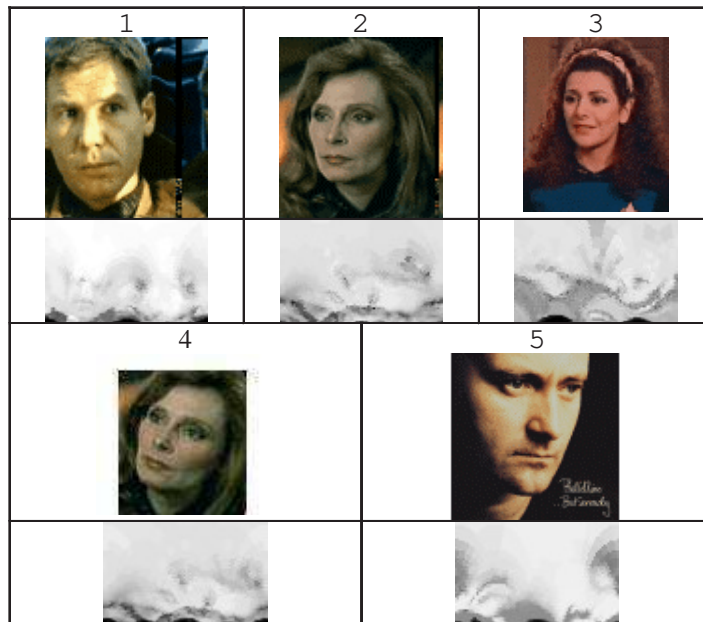


Figura 6.10: Imagens de teste da Classe 3.

1 é representada pela região de cor cinza-claro (rotulada pelo valor 1); a Classe 2, pela região de cor cinza-escuro (rotulada pelo valor 2); e a Classe 3, pelas regiões de cor preta (rotulada pelo valor 3).

Em seguida, 182 variações de cada padrão de teste - resultantes da combinação de 14 variações de orientação com 13 variações de escala - foram apresentadas ao SOM. As variações de orientação foram geradas a partir do deslocamento de 6 em 6 pixels à direita, correspondendo a $+25,8^\circ$ em cada deslocamento. Enquanto que as variações de escala foram geradas a partir do deslocamento de 3 em 3 pixels para baixo e para cima, correspondendo a $\pm 11,4\%$ em cada deslocamento.

As BMUs e classificações das variações vencedoras são apresentadas na Tabela 6.5. A Tabela 6.6 apresenta as taxas de classificação.

2	2	3	3	3	3
2	2	3	3	3	3
2	2	1	1	1	1
2	3	1	1	1	1
3	3	1	1	1	1
3	3	1	1	1	1

Figura 6.11: Mapa Contextual ilustrando 3 classes. Cinza-claro = Classe 1 (CARRO); cor cinza-escuro Classe 2 (CASTELO); cor preta Classe 3 (FACE).

A alta taxa de reconhecimento da classe 3 (100%), representada por faces, pode ser explicada pelo fato de suas imagens possuírem uma maior homogeneidade/regularidade nos elementos que as compõem (olhos, nariz, boca), comparados aos carros e castelos.

A próxima seção apresenta um experimento de reconhecimento de dígitos manuscritos, no qual as performances do SOM e do GH-SOM são comparadas.

6.3 Comparando a Performance de SOMs com GH-SOMs (Experimento 4)

Neste experimento, que teve como objetivo comparar as performances do SOM e do GH-SOM, foram utilizadas imagens de dígitos manuscritos extraídas da base MNIST, disponível

Padrão	BMU	Classe
CARRO 1	13	2
CARRO 2	17	1
CARRO 3	3	3
CARRO 4	3	3
CARRO 5	15	1
CASTELO 1	13	2
CASTELO 2	19	2
CASTELO 3	3	3
CASTELO 4	13	2
FACE 1	5	3
FACE 2	20	3
FACE 3	26	3
FACE 4	3	3
FACE 5	3	3

Tabela 6.5: BMUs e classificações dos padrões de teste.

Classe 1	40,00%
Classe 2	75,00 %
Classe 3	100,00 %
Classe 1 + Classe 2 + Classe 3	75,00 %

Tabela 6.6: Taxas de classificação.

para download em <http://yann.lecun.com/exdb/mnist/>. A base MNIST é composta por 70.000 imagens de dígitos manuscritos (de 0 a 9) de 28X28 pixels em 256 níveis de cinza, representando um subconjunto pré-processado (i.e., cujos dígitos são centralizados e normalizados quanto ao tamanho) da base NIST (SD3 e SD1⁵). A Figura 6.12 apresenta alguns exemplos da base MNIST.

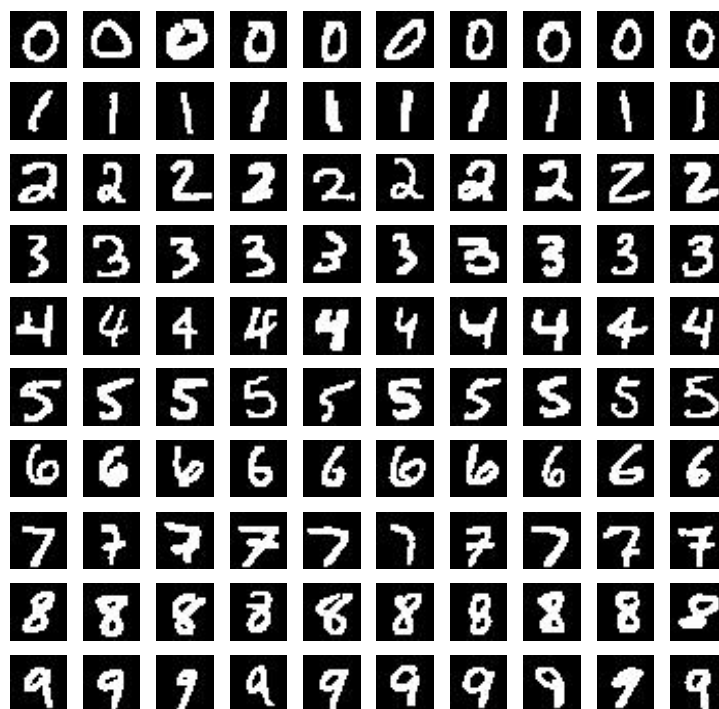


Figura 6.12: Alguns exemplos da base MNIST.

Dos 70.000 dígitos da base MNIST, foram escolhidos aleatoriamente 2000 dígitos para treinamento e 3000 para teste os quais foram redimensionados para 16X16 pixels, com o objetivo de reduzir o tempo de treinamento.

Na primeira etapa deste experimento, um GH-SOM consistindo de 206 SOMs distribuídos em 4 níveis hierárquicos foi gerado durante a fase de treinamento. Na segunda etapa, um SOM 35x33 - equivalente ao número de folhas do GH-SOM - foi treinado até atingir um erro de quantização aproximadamente igual ao obtido na primeira etapa. Observe que tanto no

⁵As bases SD3 (Special Database 3) e SD1 (Special Database 1) do National Institute for Standards and Technology (<http://www.nist.gov/>) são utilizadas como conjuntos de treinamento e teste, respectivamente, em diversas pesquisas de reconhecimento de manuscritos.

Classe	GH-SOM	SOM
0	96,7%	95,1%
1	96,3%	98,6%
2	87,4%	88,7%
3	77,6%	78,2%
4	67,7%	65,1%
5	79,6%	75,5%
6	90,5%	94,5%
7	78,7%	82,2%
8	80,5%	73,2%
9	83,8%	80,5%
Média	83,9	83,2

Tabela 6.7: Taxas de classificação do GH-SOM e SOM.

GH-SOM como no SOM as imagens de treinamento foram quantizadas, uma vez que foram utilizados 1155 neurônios para representar 2000 dígitos.

As taxas de classificação do GH-SOM (83,9%) e do SOM (83,23%) foram aproximadamente iguais, como apresentado na Tabela 6.7. Entretanto, enquanto o GH-SOM levou aproximadamente 0,3 segundos para reconhecer um padrão, o SOM levou aproximadamente 30 segundos. Ou seja, o GH-SOM foi 100 vezes mais rápido que o SOM.

A partir deste experimento, foi possível observar que o GH-SOM é equivalente ao SOM convencional, com respeito às taxas de reconhecimento, além de possuir um tempo de busca reduzido. Dessa forma, o GH-SOM foi escolhido para ser utilizado no método de RIBC desenvolvido nesta pesquisa.

6.4 Comparando a Representação Log-Polar com Características de Forma e Intensidade de Cor (Experimento 5)

Este experimento teve como objetivo comparar a representação log-polar com formas e histogramas de intensidades de cor no reconhecimento de imagens genéricas.

Um conjunto de 425 imagens, dividido em 10 classes distintas, foi obtido aleatoriamente na World Wide Web. Uma parte desse conjunto (25%, ou seja, 100 imagens) foi escolhido para treinamento, com 10 imagens representativas de cada classe. Os outros 75% do conjunto (325 imagens) foi utilizado para testes. A Figura 6.13 apresenta algumas imagens de treinamento de cada classe.

O primeiro GH-SOM (GH-SOM1) foi treinado com imagens log-polares de 84X53 pixels (obtidas de acordo com a máscara retinal descrita na Seção 5.1) com 256 níveis de cinza com os parâmetros $\tau_1 = 0.01$ e $\tau_2 = 0.001^6$ (ver Seção 2.6) resultando em três níveis hierárquicos com 128 folhas. O segundo GH-SOM (GH-SOM2) foi treinado com características de intensidade de cor, a partir de histogramas com 256 níveis de cinza; e forma, a partir dos sete momentos invariantes de Hu aplicados às bordas das imagens previamente extraídas com detectores de Sobel. Utilizando-se os mesmos parâmetros do GH-SOM1, $\tau_1 = 0.01$ e $\tau_2 = 0.001$, foi obtido uma estrutura de três níveis hierárquicos com 143 folhas.

Foram apresentadas ao GH-SOM1, 110 variações log-polar de cada imagem de teste - resultantes da combinação de 11 variações de orientação com 10 variações de escala⁷. As variações de orientação foram geradas a partir de deslocamentos de 4 em 4 pixels à direita e à esquerda, correspondendo a $\pm 17,2^\circ$ em cada deslocamento. Ou seja, foram utilizadas apenas as orientações entre $+90^\circ$ e -90° . Já as variações de escala foram geradas a partir do deslocamento de 3 em 3 pixels para baixo, correspondendo a um aumento de 11,4% em cada deslocamento. Como, durante o treinamento, foram utilizadas imagens cujos objetos ocupam o máximo de suas áreas e supondo que a maioria das imagens possui objetos que não

⁶Através desses parâmetros, foi obtido um erro de quantização de aproximadamente 0.1% do erro de quantização inicial.

⁷Observe que foram apresentadas variações da imagem de entrada apenas para o GH-SOM1, uma vez que foi utilizada a representação log-polar.





















Avião		
Barco		
Cão		
Carro		
Castelo		
Cavalo		
Gato		
Face		
Flor		
Urso		

Figura 6.13: Classes de treinamento do Experimento 5.

Classe	GH-SOM1	GH-SOM2
Avião	71,5%	64,3%
Barco	83,4%	72,5%
Cão	86,6%	81,4%
Carro	66,0%	59,4%
Castelo	60,1%	75,1%
Cavalo	86,5%	91,3%
Gato	95,2%	70,8%
Face	80,6%	66,1%
Flor	70,1%	64,1%
Urso	42,9%	62,7%
Média	74,3	70,8

Tabela 6.8: Taxas de classificação do GH-SOM1 e GH-SOM2.

ultrapassam suas áreas, não foi necessário gerar variações para diminuir as imagens de teste. Portanto, foram geradas variações que aumentam a imagem em até 100% do seu tamanho original. Assim como nos experimentos da seção 6.2, os limiares L_1 e L_2 escolhidos foram “zero” e “infinito”. O GH-SOM1 obteve uma taxa média de acerto de 74,2% (241 imagens), enquanto que o GH-SOM2 obteve uma taxa 70,8% (230 imagens). As taxas de acerto de cada classe são apresentadas na Tabela 6.8.

Os resultados obtidos mostram que a taxa de classificação utilizando a representação log-polar é ligeiramente maior (3,5%) que utilizando formas em conjunto com histogramas.

6.5 Aplicando o Sistema Proposto na Web

Nesta seção, são apresentados dois estudos de caso nos quais o sistema proposto no Capítulo 5 é adaptado e aplicado na busca de imagens da Web. A arquitetura desse SRIBCW (SRIBC para Web), ilustrada na Figura 6.14, é composta por três módulos principais:

1. **Interface**, que recebe a imagem de consulta do usuário, bem como as opções de busca desejadas.

2. **Classificador**, que funciona de acordo com o Algoritmo 5.1, exceto o fato de apenas URLs de imagens serem armazenadas no BD. Ou seja, o campo *URL* é utilizado no lugar do campo *X*.
3. **Robô**, que, constantemente, captura novas imagens da Web e remove “links quebrados” (ou inacessíveis) do BD.

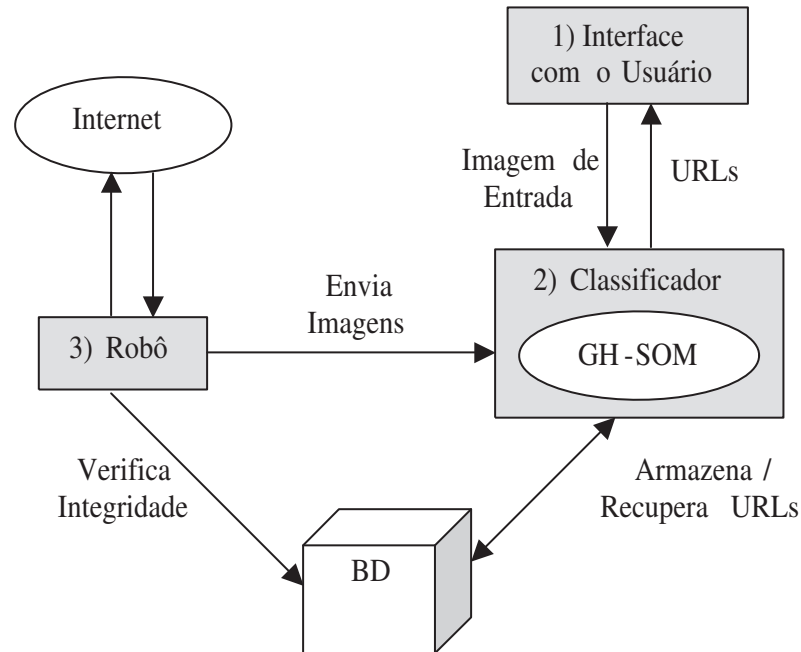


Figura 6.14: Arquitetura do SRIBCW.

Um protótipo deste SRIBCW foi implementado em cooperação com dois alunos de Iniciação Científica e está disponível para testes em <http://www.gia.dsc.ufcg.edu.br/alisson>. As tecnologias utilizadas na implementação do sistema foram: JSP - Java Server Pages, banco de dados **MySQL** (<http://www.mysql.com>) e as linguagens **Java** (<http://java.sun.com>) e **C**. O MySQL foi escolhido por ser um banco de dados gratuito e exigir poucos recursos computacionais, além de possuir um desempenho satisfatório.

6.5.1 Estudo de Caso 1

O objetivo desse estudo de caso foi utilizar a arquitetura proposta para classificar e recuperar imagens de um único local da Web, no caso <http://www.dsc.ufcg.edu.br/~luana/img/>. Nesse

local, existem 200 imagens (similares às utilizadas no experimento da Seção 6.4) divididas em quatro classes: 50 da classe CARRO, 50 da classe CASTELO, 50 da classe FACE e 50 da classe ROSA.

Como o GH-SOM foi treinado com apenas 10 imagens log-polares (84X53 em níveis de cinza) de cada classe e com os parâmetros $\tau_1 = 0.01$ e $\tau_2 = 0.001$, foi obtido um mapa não-hierárquico com 110 neurônios. Após o treinamento, cada imagem foi capturada de sua URL pelo Robô, enviada ao Classificador e armazenada no BD de acordo com o neurônio vencedor. As variações de orientação e escala das imagens foram geradas de forma similar à do experimento da Seção 6.4.

Na fase de recuperação, cada uma das 200 URLs da base foi fornecida novamente ao sistema e, em seguida, as imagens recuperadas foram analisadas quanto às classes pertencentes. A Figura 6.15 ilustra as imagens recuperadas pelo sistema ao se introduzir uma imagem da classe CARRO. De acordo com a figura, o sistema retornou cinco imagens da classe CARRO e uma imagem da classe CASTELO. Ou seja, foi obtida uma taxa de acerto de 83,33%, que pode ser vista como a *confiança* (ou precisão) da consulta. Já a quantidade de imagens recuperadas da classe CARRO - que no caso representa 10% das imagens dessa classe (5/50) - pode ser vista como o *suporte* (ou cobertura) da consulta ⁸.

A Tabela 6.9 apresenta as taxas de acerto total de cada classe e a Tabela 6.10 ilustra a matriz de confusão. Por questão de simplicidade, apenas as 5 imagens mais próximas de cada imagem de entrada foram consideradas no cálculo dos valores dessas tabelas. A coluna “Qtd. Img. Recup.” na Tabela 6.9 representa o somatório das imagens recuperadas, em todo o experimento, para cada classe. Já a coluna “Acertos” representa o somatório das imagens recuperadas que pertencem à mesma classe da imagem de entrada.

A partir dos resultados obtidos neste estudo de caso, concluímos que a arquitetura proposta possui um desempenho satisfatório para um local específico da Web e, provavelmente, pode ser expandida para um domínio maior. O próximo estudo de caso visa atingir esse objetivo em particular.

⁸Os termos *confiança* e *suporte* são amplamente utilizados pela comunidade de Banco de Dados.

Imagem de entrada:



Resultado da pesquisa:

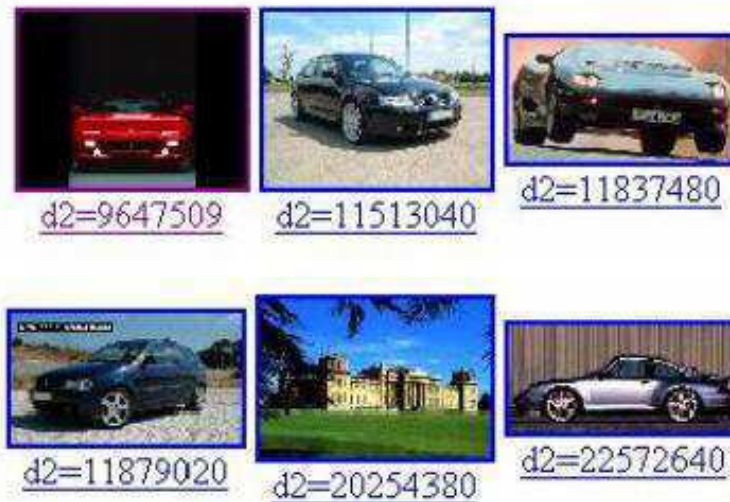


Figura 6.15: Exemplo de recuperação de imagens.

	Qtd. Img. Recup.	Acertos	Tx. Acerto
CARRO	228	125	54,82%
CASTELO	243	154	63,37%
FACE	250	193	77,2%
ROSA	246	113	45,93%
Total	967	585	60,5%

Tabela 6.9: Taxas de acerto do Estudo de Caso 1.

	CARRO	CASTELO	FACE	ROSA
CARRO	125	58	8	37
CASTELO	47	154	5	37
FACE	8	13	193	36
ROSA	43	51	39	113

Tabela 6.10: Matriz de Confusão do Estudo de Caso 1.

6.5.2 Estudo de Caso 2

Neste estudo de caso, o SRIBCW utiliza o GH-SOM1 obtido no experimento da Seção 6.4, o qual possui mais neurônios que a quantidade de imagens de treinamento. O objetivo desse treinamento foi obter um mapa com uma maior generalização (i.e., “bolhas de atividade” maiores), uma vez que estamos lidando com um ambiente onde existem infinitas classes de imagens.

A base de dados foi preenchida com URLs de várias imagens da Web de acordo com os seguintes passos:

1. Primeiramente, o Robô captura uma imagem qualquer na Web e a envia para o Classificador.
2. em seguida, o Classificador executa o algoritmo descrito no Capítulo 5 com os limiares $L_1 = 0.04$ e $L_2 = 1$, definidos a partir do experimento da Seção 6.4.

O limiar L_1 representa a maior d_2 normalizada obtida entre os padrões de teste corretamente classificados sem gerar variações. Enquanto que o limiar L_2 representa a maior d_2 normalizada obtida entre os padrões de teste corretamente classificados. Além desses limiares - com o objetivo de excluir certos tipos de imagens indesejadas, como ícones, barras e imagens muito grandes - foram definidas as dimensões mínima e máxima para as imagens utilizadas pelo sistema, no caso 80 e 700 pixels, respectivamente.

A consulta implementada nesse protótipo é do tipo: “**select URL from IMAGENS where BMU = bmu and Dist <= d2 + similaridade and Dist >= d2 - similaridade**”. Em que “bmu” e “d2” são retornados pelo Classificador; “similaridade” é um valor entre 0.03 e 0.003, com passos de 0.03, definido pelo usuário através de uma Interface (ver Figura 6.16), utilizando

uma escala de 10 níveis; e as inequações envolvendo os atributos anteriores definem um raio de similaridade englobando imagens próximas à imagem de entrada. Os tipos de variações a serem geradas pelo sistema - orientação, escala, orientação+escala ou nenhuma - também são especificados na Interface, ilustrada na Figura 6.16.

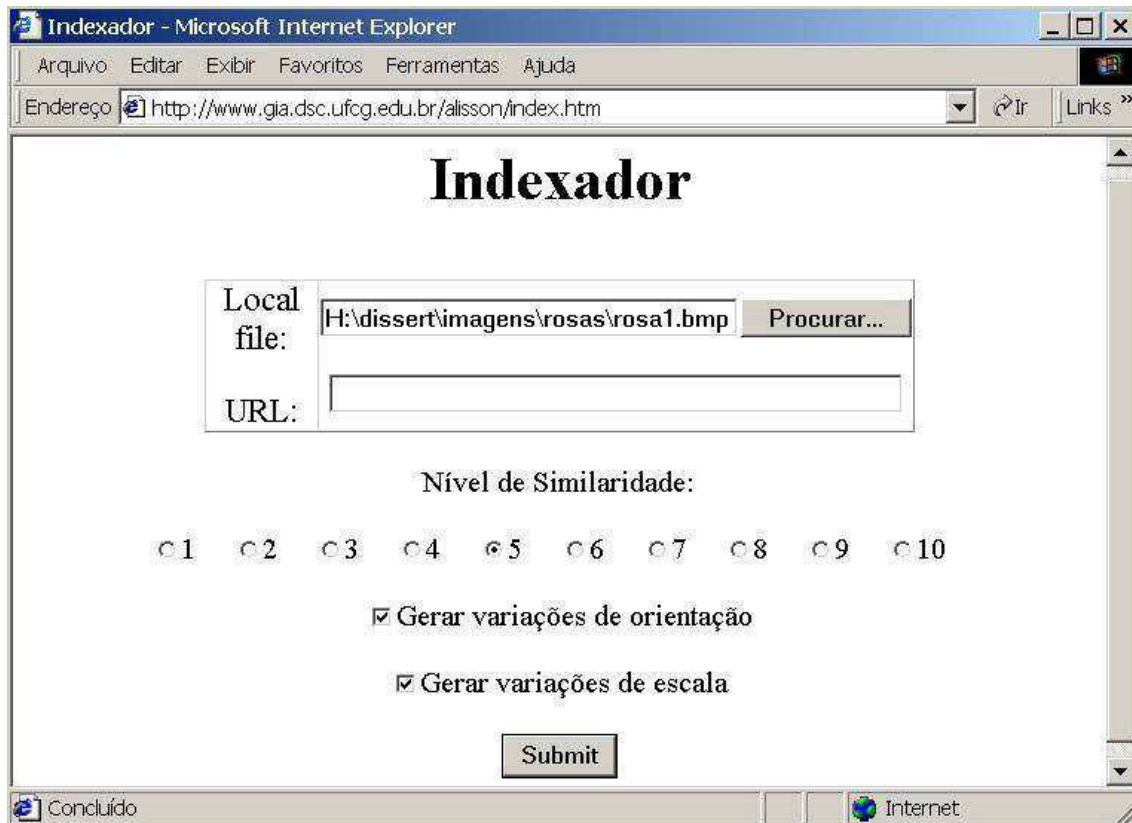


Figura 6.16: Interface do SRIBCW.

Os primeiros resultados foram obtidos após 32 horas de repetição dos passos 1 e 2 apresentados anteriormente, obtendo-se aproximadamente 2320 URLs de imagens classificadas.

A Figura 6.17 apresenta o resultado de uma consulta sem gerar variações da imagem de entrada e com nível de similaridade igual a 0.003. Note que apenas imagens muito similares à imagem de entrada foram recuperadas. Os valores abaixo de cada imagem retornada pelo sistema indicam a d_2 (normalizada) ao neurônio vencedor, que nesse caso é a BMU 30.

A Figura 6.18 apresenta uma amostra das imagens indexadas pela BMU 30⁹, com suas respectivas distâncias Euclidianas ao quadrado. O Apêndice A apresenta outras amostras de imagens indexadas no BD.

⁹A BMU 30 indexa, atualmente, 110 imagens.

Imagem de entrada:



Resultado da pesquisa:



Figura 6.17: Recuperação com similaridade 0.003 e sem variações.

A Figura 6.19 apresenta o resultado de uma consulta ao se gerar variações de orientação e escala da imagem de entrada e com nível de similaridade igual a 0.03. As imagens mais próximas à imagem de entrada são apresentadas na segunda linha da figura (da sexta à décima imagem).

A recuperação de diversas imagens diferentes da imagem de entrada, como por exemplo, a oitava imagem (terceira imagem da segunda linha), deve-se a problemas durante a fase de treinamento: imagens de classes diferentes foram mapeadas em uma mesma BMU (ver as BMUs 5 e 13 no Apêndice A). A Seção 7.3.1 discute como esse tipo de problema pode ser solucionado.

O fato de haver poucas imagens indexadas por uma determinada BMU¹⁰ também pode causar a recuperação de imagens inadequadas. A Figura 6.20 apresenta todas as imagens indexadas pela BMU 9. Note que as distâncias Euclidianas ao quadrado, abaixo de cada imagem da Figura 6.20, são muito distantes umas das outras.

Uma vez que é praticamente impossível o acesso a todas as classes de imagens existentes na Web, durante a fase de treinamento, o SRIBC proposto torna-se mais apropriado para aplicações com um número “limitado” de classes, como por exemplo, recuperação imagens de áreas específicas, como medicina e sensoriamento remoto; proibição de acesso a sites com conteúdo pornográfico, etc. A Figura 6.21 mostra o resultado do sistema ao se tentar

¹⁰Possivelmente porque o Robô não teve tempo suficiente para coletar um grande número de URLs de imagens até a conclusão desta dissertação.

Imagem de treinamento:



Imagens do BD:











			
0,0162685	0,03192437	0,0329814	0,03695439
			
0,03708972	0,03731524	0,03739578	0,0390985
			
0,03997863	0,04028462	0,0419316	0,04239256

Figura 6.18: Amostra das imagens indexadas pela BMU 30.

Imagem de entrada:



Resultado da pesquisa:



Figura 6.19: Recuperação com similaridade 0.03 e variações.






				
0,2078077	0,221523	0,472563	0,5195442	0,6029886

Figura 6.20: Imagens indexadas pela BMU 9.

recuperar uma imagem de uma classe muito diferente das utilizadas durante o treinamento.

6.6 Considerações Finais

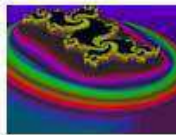
Este capítulo relatou os experimentos que foram fundamentais na formulação da melhor estratégia de classificação e nos testes de performance do sistema proposto. Tais experimentos foram realizados em um computador convencional (com 900MHz de processador e 512M de RAM) com o auxílio das seguintes bibliotecas do **Matlab** (<http://www.mathworks.com>):

- *Image Processing*;
- *NetLab* (<http://www.ncrg.aston.ac.uk/netlab/down.php>);
- *SOM Toolbox* (<http://www.cis.hut.fi/projects/somtoolbox/download/>) e
- *GH-SOM Toolbox* (<http://www.ai.univie.ac.at/~elias/ghsom/download.html>).

Além dessas funções, muitas outras tiveram que ser implementadas ou adaptadas.

O próximo capítulo apresenta as conclusões desta dissertação, apontando as principais contribuições, os objetivos atingidos e os possíveis trabalhos futuros.

Imagem de entrada:



Resultado da pesquisa:



d2=0.387415



d2=0.3896986



d2=0.392674



d2=0.393179



d2=0.3963095



d2=0.4126355



d2=0.4163807

Figura 6.21: Classe desconhecida.

Capítulo 7

Conclusão

Este capítulo apresenta um sumário dos principais pontos discutidos nesta dissertação, bem como as contribuições da pesquisa desenvolvida e sugestões de trabalhos futuros.

7.1 Sumário da Dissertação

No Capítulo 1, apresentamos a motivação para o desenvolvimento de Sistemas de Recuperação de Imagens Baseada em Conteúdo e as principais limitações existentes nas soluções atuais. Nesse contexto, definimos os principais objetivos desta pesquisa, que foram: (i) apresentar um estudo detalhado sobre o funcionamento dos Mapas Auto-Organizáveis, possibilitando um melhor entendimento da técnica; (ii) propor uma estratégia de classificação/indexação baseada em GH-SOMs e imagens log-polar, com vistas a possibilitar uma recuperação e representação de imagem alternativas àquelas tradicionalmente utilizadas em SRIBCs; e (iii) uma vez concluído o objetivo anterior, desenvolver um SRIBC, aplicá-lo em um problema de recuperação de imagens da World Wide Web e analisar seu desempenho.

No Capítulo 2, foi apresentado um estudo sobre os Mapas Auto-Organizáveis, um tipo de rede neural não-supervisionada inspirada nos mapas topologicamente ordenados do cérebro. Uma ênfase maior foi dada aos mapas de Kohonen, um modelo simples de SOM que possibilita a compressão dos dados de entrada (*quantização vetorial*). Além do modelo de Kohonen, também foi discutido o funcionamento do GH-SOM, um modelo composto por SOMs independentes organizados em forma de árvore. O GH-SOM, além de conservar as principais propriedades do modelo de Kohonen, define automaticamente as dimensões dos

SOMs que compõem sua hierarquia e, por se tratar de um modelo hierárquico, possui tempo de resposta menor que um mapa de Kohonen. A principal contribuição desse capítulo é a possibilidade de ser utilizado como material didático para futuros estudantes que darão continuidade à pesquisa.

No Capítulo 3, foram apresentadas as principais características dos SRIBCs. Nesse tipo de sistema, os atributos visuais, tais como cor, forma e textura, são extraídos de cada imagem - com o objetivo de obter representações reduzidas, bem como invariâncias a determinadas propriedades - e armazenadas em um vetor denominado “vetor de características”. Em seguida, os vetores de características são organizados em algum tipo de estrutura de indexação, com o objetivo de reduzir o espaço de busca. Dessa forma, foram brevemente descritas algumas das principais técnicas utilizadas tanto para extração e representação de características visuais (i.e., Histogramas de Cor, Descritores de Fourier, Momentos Invariantes e Filtros de Gabor), quanto para indexação de dados multidimensionais (i.e., estruturas derivadas da B-Tree e H-SOMs).

No Capítulo 4 foram analisados quatro trabalhos diretamente relacionados com a pesquisa desenvolvida. Tais trabalhos foram divididos de acordo com o tipo de Mapa Auto-Organizável utilizado para indexar imagens em SRIBCs, ou seja, SOM, H-SOM e TS-SOM. Tais sistemas, apesar de resolverem muitos dos problemas existentes nas estruturas de indexação tradicionais, não fazem uso de importantes características dos SOMs (como a quantização vetorial, por exemplo), as quais poderiam melhorar a performance do SRIBC.

No Capítulo 5, foi proposto um SRIBC que utiliza um GH-SOM, como estrutura de indexação, e uma representação de imagem log-polar, inspirada na retina humana. Inicialmente, foi descrito o funcionamento da representação log-polar que, por possuir a propriedade de conversão de mudanças de escala e orientação no espaço de entrada em translações no espaço log-polar, facilita o reconhecimento de objetos em diferentes escalas e orientações. A partir dessa representação, foi apresentada uma nova estratégia de classificação, na qual apenas as imagens mais representativas de cada classe são utilizadas na fase de treinamento. A principal vantagem dessa estratégia é a redução significativa do tempo de treinamento e da árvore de indexação.

No Capítulo 6 foram relatados os principais experimentos realizados ao longo desta pesquisa. A partir de dois experimentos iniciais, foi concluído que é mais adequado incor-

porar invariâncias em Mapas Auto-Organizáveis *por espaço de características* ao invés de incorporá-las *por treinamento*. Nessa última técnica, não há como impedir o agrupamento de padrões com base em características como orientação, escala, cor, etc., já que os Mapas Auto-Organizáveis utilizam aprendizagem não-supervisionada. O terceiro experimento comprovou que o GH-SOM é equivalente ao SOM convencional, com respeito às taxas de reconhecimento, além de possuir um tempo de busca reduzido. Esses três experimentos iniciais foram fundamentais para a formulação da melhor estratégia de classificação. Em seguida, foram realizados experimentos comparativos utilizando imagens log-polares, histogramas de cores e formas. Dessa forma, foi possível comparar a representação de imagem utilizada no sistema proposto com as representações mais comuns da abordagem tradicional. Finalmente, demonstrou-se o funcionamento do sistema proposto na recuperação de imagens da World Wide Web.

Diante dos resultados obtidos, consideramos que os principais objetivos desta dissertação foram alcançados. Além disso, esta pesquisa resultou em quatro publicações [Batista and Gomes, 2003a; Batista and Gomes, 2003b; Batista and Gomes, 2003c; Batista et al., 2004] decorrentes dos principais experimentos realizados.

7.2 Contribuições

Como vimos no Capítulo 3, as estruturas de indexação baseadas na B-Tree possuem alguns problemas, como sobreposição de MBRs (ex., R-Tree e TV-Tree), armazenamento de regiões vazias (ex., KDB-Tree) e ineficiência ao lidar com altas dimensões (ex., KDB-Tree e R-Tree). Tais problemas podem ser resolvidos utilizando-se estratégias baseadas em Mapas Auto-Organizáveis, como ocorre com os sistemas descritos no Capítulo 4. Entretanto, tais sistemas não fazem uso de todas as vantagens proporcionadas por esse tipo de rede neural. Os sistemas propostos por Laaksonen *et al.* [Laaksonen et al., 1999b; Laaksonen et al., 1999a] e del-Solar e Navarrete [del Solar and Navarrete, 2002], por exemplo, apesar de utilizarem TS-SOMs - um tipo de H-SOM baseado no modelo de Kohonen - utilizam mais neurônios que o número total de imagens, ou seja, não realizam quantização vetorial para reduzir o espaço de entrada. Já o modelo de Zhang e Zhong [Zhang and Zhong, 1995] utiliza a auto-organização apenas na base da hierarquia. As outras camadas são geradas a partir da média dos vetores

de pesos dos neurônios da camada anterior.

O sistema proposto nesta dissertação possui três novidades com relação aos sistemas descritos no Capítulo 4:

1. Ao invés de treinar o mapa com toda base de imagens, um pequeno conjunto de imagens de cada classe é escolhido para representá-las, permitindo uma redução significativa do tempo de treinamento, da árvore de indexação e, conseqüentemente, do tempo de recuperação;
2. Ao invés de utilizar características visuais do tipo cor, forma e textura - que podem causar a perda de informações valiosas da imagem - é utilizada uma representação de imagem equivalente à imagem original, inspirada na retina humana. Esse tipo de representação, além de facilitar o reconhecimento de imagens independentemente de orientação e escala, permite uma compactação da imagem original. Como utilizamos a representação log-polar com 256 níveis de cinza, não é possível recuperar imagens com base na “cor”. No entanto, em trabalhos futuros, a representação log-polar colorida poderá ser utilizada.
3. Ao invés de utilizar um SOM cuja arquitetura deve ser previamente definida, é utilizado um GH-SOM, no qual as dimensões dos SOMs, bem como a profundidade da hierarquia são determinadas automaticamente durante a fase de treinamento. Dessa forma, pouco tempo é gasto na obtenção da arquitetura mais adequada.

Além disso, demonstrou-se que o sistema proposto pode ser adaptado para a busca de imagens na Web (em aplicações com um número limitado de classes) com a inclusão de um Robô capaz de capturar imagens de forma automática em tal ambiente.

7.3 Trabalhos Futuros

Esta seção apresenta algumas sugestões de trabalhos futuros com respeito à obtenção de uma melhor performance e a novas aplicações do sistema proposto.

7.3.1 SOMs *com consciência* e LVQ

Mesmo utilizando mais neurônios que o número de imagens de treinamento no GH-SOM do RIBCW, alguns padrões - nem sempre da mesma classe - foram agrupados em um mesmo neurônio. Isto é, vários neurônios foram sub-utilizados durante o treinamento. Isso também se refletiu durante os testes do sistema, ou seja, com 2320 imagens classificadas, os neurônios 1 e 3, por exemplo, não indexavam nenhuma imagem. Já o neurônio 99 indexava 227 imagens. Existem pelo menos duas formas de amenizar este tipo de problema:

1. utilizando SOMs *com consciência* (discutido no Capítulo 4), os quais tentam utilizar uniformemente todos os neurônios do mapa;
2. utilizando LVQ (Learning Vector Quantization) [Kohonen, 1990], técnica de aprendizagem supervisionada que utiliza os rótulos de classe para ajustar os vetores de pesos do SOM. considere, por exemplo, um padrão de treinamento x e um vetor de pesos W que representa tanto x , da classe X , quanto um outro padrão de treinamento y da classe Y .¹ Se os rótulos de x e de W concordarem (W possui o rótulo X), o vetor W é movido em direção ao vetor de treinamento x . Se, por outro lado, os rótulos de x e de W discordarem (W possui o rótulo Y), o vetor W é afastado do vetor de treinamento X . Ou seja:

- Se $Rótulo(W) = Rótulo(x)$, $W(n+1) = W(n) + \alpha[x - W(n)]$
- Senão, $W(n+1) = W(n) - \alpha[x - W(n)]$, onde $0 < \alpha < 1$

Portanto, um trabalho futuro desta pesquisa é melhorar a performance do sistema utilizando os algoritmos acima conjuntamente, já que, mesmo após um treinamento *com consciência*, é possível que existam padrões de classes diferentes mapeados em um mesmo neurônio. Dessa forma, o LVQ poderia ser utilizado para um ajuste final dos SOMs com consciência.

¹Observe que W possui apenas um rótulo de classe, decorrente do padrão de treinamento mais próximo a ele.

7.3.2 Aprendizagem Incremental

No sistema proposto nesta dissertação, as classes de imagens precisam ser definidas em uma etapa anterior ao treinamento. Após o treinamento, novas imagens podem ser adicionadas ao sistema de acordo com o escopo de classes escolhido. Entretanto, é altamente desejável que a aprendizagem seja incremental, ou seja, se após um número n de variações de orientação e escala do padrão de entrada não houver o casamento com nenhuma das classes conhecidas, o padrão desconhecido poderia ser aprendido; representando, portanto, uma nova classe.

É importante notar que, na aprendizagem incremental, não há distinção entre a fase de treinamento e a fase de uso, uma vez que o aprendizado é realizado de forma contínua. Algumas redes neurais Hebbianas, derivadas dos Mapas Auto-Organizáveis - como o ITPM [Millán, 1997] e o ITM [Jockusch and Ritter, 1999], por exemplo - realizam esse tipo de aprendizado, inserindo novas unidades sempre que é necessária uma melhor cobertura do espaço de entrada. No entanto, tais redes neurais não seriam eficientes em SRIBCs com um grande número de imagens, uma vez que a busca é realizada de forma seqüencial. Portanto, sugerimos a implementação de um novo tipo de rede neural que seja tanto incremental quanto hierárquica.

7.3.3 Representação Log-Polar *versus* Texturas

Apesar de terem sido realizados experimentos comparativos entre a representação log-polar (utilizada pelo sistema proposto) e histogramas de cores e formas (utilizados pela abordagem tradicional), a textura não foi utilizada nessa pesquisa. A principal razão para a exclusão dessa característica visual foi a complexidade das técnicas utilizadas para a sua extração/representação, como por exemplo, as Wavelets. Sugerimos, portanto, a realização de um estudo comparativo entre a representação log-polar e as texturas, em recuperação de imagens baseada em conteúdo.

7.3.4 Bloqueio de Conteúdos Proibidos

Outro trabalho futuro seria a adaptação e utilização do sistema proposto na proibição de acesso a sites com determinados tipos de conteúdos. Ao se tentar entrar em um site com fotos pornográficas, por exemplo, o sistema capturaria algumas dessas fotos, as classificaria e, em

vez de retornar URLs com imagens similares, impediria o acesso a esse site e o acrescentaria em uma lista “sites proibidos”.

7.3.5 Sistemas de Apoio à Decisão

Como vimos no Capítulo 1, uma aplicação muito importante de RIBC é em sistemas de apoio à decisão, como a identificação de suspeitos (reconhecimento de faces) e o suporte a diagnósticos médicos (reconhecimento de imagens médicas). Dessa forma, sugerimos a adaptação do sistema proposto - provavelmente agregando outros tipos de características visuais, além da representação log-polar - para este tipo de problema.

Bibliografia

- [Alexandrov et al., 1995] Alexandrov, A., Ma, W., Abbadi, A., and Manjunath, B. (1995). Adaptive filtering and indexing for image databases. In *SPIE Conference on Storage and Retrieval for Image and Video Databases*, pages 12–23.
- [Barnard and Casasent, 1991] Barnard, E. and Casasent, D. (1991). Invariance and neural nets. *IEEE Transactions on Neural Networks*, 2:498–508.
- [Batista and Gomes, 2003a] Batista, L. and Gomes, H. (2003a). Application of growing hierarchical self-organizing map in handwritten digit recognition. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBIGRAPI'03)*.
- [Batista and Gomes, 2003b] Batista, L. and Gomes, H. (2003b). Scale and orientation invariant object recognition using self-organizing maps. In *Conferência Latino Americana de Informática (CLEI'03)*.
- [Batista and Gomes, 2003c] Batista, L. and Gomes, H. (2003c). Um método para classificação e recuperação de imagens da world wide web utilizando mapas auto-organizáveis. In *Simpósio Brasileiro de Automação Inteligente (SBAI'03)*, pages 66–71.
- [Batista et al., 2004] Batista, L., Gomes, H., Almeida, F., and Silva, A. (2004). Content-based image retrieval combining growing hierarchical self-organizing maps and a log-polar representation. In *Simpósio Brasileiro de Redes Neurais (SBRN'04)*.
- [Bayer and McCreight, 1972] Bayer, R. and McCreight, E. (1972). *Organization and Maintenance of Large Ordered Indexes*, volume 1. Acta Informatica.
- [Beale and Jackson, 1990] Beale, R. and Jackson, T. (1990). *Neural Computing: an Introduction*. Institute of Physics Publishing, Bristol.

- [Brice and Fennema, 1970] Brice, C. and Fennema, C. (1970). Scene analysis using regions. *Artificial Intelligence*, 1:205–226.
- [Brodatz, 1966] Brodatz, P. (1966). *Texture: A Photographic Album for Artists and Designers*. Dover publications, New York.
- [Brown and Gruenwald, 1998] Brown, L. and Gruenwald, L. (1998). Tree-based indexes for image data. *Journal of Visual Communication and Image Representation*, 9(4):300–313.
- [Chien, 1993] Chien, C. (1993). Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686.
- [del Solar and Navarrete, 2002] del Solar, J. R. and Navarrete, P. (2002). Faceret: An interactive face retrieval system based on self-organizing maps. *Lecture Notes in Computer Science*, 2383:157–164.
- [DeSieno, 1988] DeSieno, D. (1988). Adding a conscience to competitive learning. *IEEE International Conference Neural Networks*, 1:117–124.
- [Dittenbach et al., 2000] Dittenbach, M., Merkl, D., and Rauber, A. (2000). The growing hierarchical self-organizing map. In *Int. Joint Conference on Neural Networks*, volume 6, pages 15–19.
- [Falcão, 2003] Falcão, A. (2003). Fundamentos de processamento de imagem digital. <http://www.dcc.unicamp.br/~cpg/material-didatico/mo815/9802/curso/curso.html>.
- [Friedman, 1977] Friedman, J. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, 26:404–408.
- [Gomes and Fisher, 2001] Gomes, H. and Fisher, R. (2001). Learning and extracting primal-sketch features in a log-polar image representation. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBIGRAPI)*, pages 338–345.
- [Gomes and Fisher, 2003] Gomes, H. and Fisher, R. (2003). Learning-based versus model-based log-polar feature extraction operators: a comparative study. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBIGRAPI)*, pages 299–306.

- [Gomes et al., 1998] Gomes, H., Fisher, R., and Hallam, J. (1998). A retina-like image presentation of primal sketch features extracted using a neural network approach. In *Noblesse Workshop on Non-Linear Model Based Image Analysis*, pages 251–256.
- [Gonzalez and Woods, 1992] Gonzalez, R. and Woods, R. (1992). *Digital Image Processing*. Addison-Wesley Publishing Company, Inc, Massachusetts.
- [Grossberg, 1980] Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, 87:1–51.
- [Guttman, 1984] Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD International Conference on Management of Data*, pages 47–57.
- [Haykin, 2001] Haykin, S. (2001). *Redes Neurais - Princípios e Prática*. Bookman, Porto Alegre.
- [Hebb, 1949] Hebb, D. (1949). *The organisation of behavior*. Wiley, New York.
- [Hopfield, 1982] Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *National Academy of Sciences of the USA*, pages 2554–2588.
- [Hu, 1962] Hu, M. (1962). Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, IT-8.
- [Hubel and Wiesel, 1962] Hubel, D. and Wiesel, T. (1962). Integrative action in the cat's lateral geniculate body. *Journal of Physiology*, 155:385–398.
- [Hubel and Wiesel, 1977] Hubel, D. and Wiesel, T. (1977). Ferrier lecture functional architecture of macaque visual cortex. In *R. Soc. Lond.*, volume 198, pages 1–59.
- [Jockusch and Ritter, 1999] Jockusch, J. and Ritter, H. (1999). An instantaneous topological mapping model for correlated stimuli. In *Int. Joint Conference on Neural Networks*, page 445.
- [Kaas, 1983] Kaas, J. (1983). What, if anything, is si? organization of first somatosensory area of cortex. *Physiol Reviews*, 63:206–231.

- [Keyes and Winstanley, 1999] Keyes, L. and Winstanley, A. (1999). Fourier descriptors as a general classification tool for topographic shapes. In *Irish Machine Vision and Image Processing Conference*, pages 193–203.
- [Keyes and Winstanley, 2001] Keyes, L. and Winstanley, A. (2001). Using moment invariants for classifying shapes on large scale maps. *Computers, Environment and Urban Systems*, 25.
- [Knudsen et al., 1987] Knudsen, E., du Lac, S., and Esterly, S. (1987). Computational maps in the brain. *Annual Review of Neuroscience*, 10:41–65.
- [Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.
- [Kohonen, 1989] Kohonen, T. (1989). *Self-Organization and Associative Memory*, volume 3rd. Springer-Verlag, Tiergartenstrasse 17, D-6900 Heidelberg, Germany.
- [Kohonen, 1990] Kohonen, T. (1990). The self-organizing map. *IEEE*, 78:1464–1480.
- [Koikkalainen, 1994] Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence*, pages 211–215.
- [Koikkalainen and Oja, 1990] Koikkalainen, P. and Oja, E. (1990). Self-organizing hierarchical feature maps. In *International Joint Conference on Neural Networks*, pages 279–284.
- [Laaksonen et al., 1999a] Laaksonen, J., Koskela, M., and Oja, E. (1999a). Application of tree structured self-organizing maps in content-based image retrieval. In *9th International Conference on Artificial Neural Networks*.
- [Laaksonen et al., 1999b] Laaksonen, J., Koskela, M., and Oja, E. (1999b). Picsom: Self-organizing maps for content-based image retrieval. *IEEE International Joint Conference on Neural Networks*, 4:2470–2473.
- [Lin et al., 1994] Lin, K., Jagadish, H., and Faloutsos, C. (1994). The tv-tree: An index structure for high-dimensional data. *Very Large Data Bases*, 3:517–542.

- [Manjunath and Ma, 1998] Manjunath, B. and Ma, W. (1998). A texture thesaurus for browsing large aerial photographs. *Journal of the American Soc. for Inf. Science*, 49(7):633–648.
- [Marr, 1982] Marr, D. (1982). *Vision*. W. Freeman.
- [Martins et al., 2002] Martins, M., Guimarães, L., and Fonseca, L. (2002). Texture feature neural classifier for remote sensing image retrieval systems. In *XV Brazilian Symp. on Computer Graphics and Image Proc.*, pages 90–96.
- [McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–13.
- [Miikkulainen, 1990] Miikkulainen, R. (1990). Script recognition with hierarchical feature maps. *Royal Society London*, 2.
- [Millán, 1997] Millán, J. (1997). Incremental acquisition of local networks for the control of autonomous robots. In *7th International Conference on Artificial Neural Networks*, pages 739–744.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT Press.
- [Phillips et al., 1998] Phillips, P., Wechsler, H., Huang, J., and Rauss, P. (1998). The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 5:295–306.
- [Pratt, 1991] Pratt, W. (1991). *Digital Image Processing*, volume 2. Wiley-Interscience.
- [Robinson, 1981] Robinson, J. (1981). The k-d-b-tree: A search structure for large multidimensional dynamic indexes. In *ACM SIGMOD International Conference on Management of Data*, pages 10–18.
- [Rosa et al., 2002] Rosa, N., Santos, R., Bueno, J., Traina, A., and Traina, C. (2002). Sistema de recuperação de imagens similares em um hospital universitário. In *VIII Congresso Brasileiro de Informática em Saúde*.
- [Rosenblatt, 1957] Rosenblatt, F. (1957). The perceptron: A perceiving and recognizing automaton. Technical report, Project PARA, Cornell Aeronautical Lab.

- [Rui et al., 1999] Rui, Y., Huang, T., and Chang, S. (1999). Image retrieval: current techniques, promising directions, and open issue. *Journal of Visual Comm. and Image Representation*, 10:39–62.
- [Rumelhart et al., 1986] Rumelhart, D., Hinton, G., and Williams, R. (1986). *Parallel Distributed Processing*, volume 1. MIT Press.
- [Smith and Chang, 1997] Smith, J. and Chang, S. (1997). Safe: A general framework for integrated spatial and feature image search. In *Workshop on Multimedia Signal Processing*, pages 301–306.
- [Subrahmanian, 1998] Subrahmanian, V. (1998). *Principles of Multimedia Database Systems*. Morgan Kaufmann Publishers, Inc.
- [Suga and Tsuzuki, 1985] Suga, N. and Tsuzuki, K. (1985). Inhibition and level-tolerant frequency tuning representation in the auditory cortex of the mustached bat. *Neurophysiology*, 47:225–255.
- [Suganthan, 2002] Suganthan, P. (2002). Shape indexing using self-organizing maps. *IEEE Transactions on Neural Networks*, 13(4).
- [Tamura et al., 1976] Tamura, H., Mori, S., and Yamawaki, T. (1976). Texture features corresponding to visual perception. *IEEE Transactions on System, Man and Cybernetics*, SMC-6(4):460–473.
- [Vasconcelos, 2000] Vasconcelos, N. (2000). Mapas auto-organizativos e aplicações. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro (UFRJ).
- [von der Malsburg, 1973] von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14:85–100.
- [Willshaw and von der Malsburg, 1976] Willshaw, D. and von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Royal Society London*, B. 194:431–445.

[Xu et al., 2000] Xu, K., Georgescu, B., Comaniciu, D., and Meer, P. (2000). Performance analysis in content-based retrieval with textures. In *International Conference on Pattern Recognition*, pages 4275–4278.

[Zhang and Zhong, 1995] Zhang, H. and Zhong, D. (1995). A scheme for visual feature based image retrieval. In *SPIE Conf. on Storage and Retrieval for Image and Video Database*, pages 36–46.

Apêndice A

Amostras de imagens indexadas

Este apêndice apresenta amostras de imagens indexadas por algumas BMUs do sistema implementado. A base de imagens - utilizada pelo segundo estudo de caso, no Capítulo 6 - possui atualmente 2320 imagens, as quais são indexadas por 128 neurônios (do tipo “folha”) de um GH-SOM.

- A Figura A.1 apresenta as 12 primeiras imagens das 22 indexadas pela BMU 5. Durante o treinamento, foram mapeadas duas imagens de classes diferentes, ou seja, das classes “gato” e “flor”.
- A Figura A.2 apresenta as 12 primeiras imagens das 60 indexadas pela BMU 13. Durante o treinamento, foram mapeadas duas imagens de classes diferentes, ou seja, das classes “urso” e “gato”.
- A Figura A.3 apresenta as 12 primeiras imagens das 95 indexadas pela BMU 16. Durante o treinamento, foi mapeada uma imagem da classe “face”.
- A Figura A.4 apresenta as 12 primeiras imagens das 22 indexadas pela BMU 40. Durante o treinamento, foi mapeada uma imagem da classe “flor”.

Imagens de treinamento:



Imagens do BD:













			
0,1040646	0,1107869	0,1950989	0,2195561
			
0,2570656	0,2931115	0,3007134	0,3137358
			
0,3857773	0,5435663	0,6214053	0,6874714

Figura A.1: Amostra de imagens indexadas pela BMU 5.

Imagens de treinamento:



Imagens do BD:













			
0,02728352	0,02731855	0,03316936	0,03716032
			
0,04039749	0,04203737	0,04599509	0,04630184
			
0,04875657	0,05355782	0,05372273	0,06150255

Figura A.2: Amostra de imagens indexadas pela BMU 13.

Imagem de treinamento:



Imagens do BD:













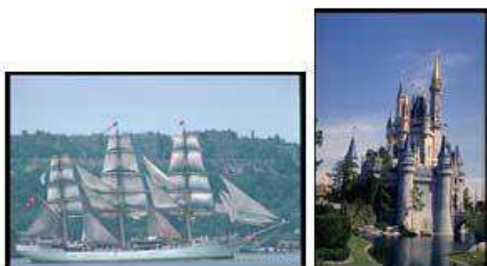
			
0,07221338	0,07634314	0,07773742	0,07876797
			
0,07876797	0,07913732	0,07940373	0,07999536
			
0,08252492	0,08567317	0,08578682	0,08670983

Figura A.3: Amostra de imagens indexadas pela BMU 16.

Imagens de treinamento:



Imagens do BD:













			
0,03479588	0,04235411	0,04260166	0,05776941
			
0,06310087	0,07003404	0,07071794	0,07985649
			
0,1346432	0,1422885	0,1425768	0,1458377

Figura A.4: Amostra de imagens indexadas pela BMU 40.