Universidade Federal de Uberlândia - UFU

Faculdade de Computação - FACOM

Programa de Pós-graduação em Ciência da Computação

Caracterização de Imagens utilizando Redes Neurais Artificiais

Autor: Eduardo Ferreira Ribeiro

Orientadora: Profª. Drª. Celia A. Zorzo Barcelos

Uberlândia – Junho de 2009

FACULDADE DE COMPUTAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eduardo Ferreira Ribeiro

Caracterização de Imagens utilizando Redes Neurais Artificiais

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.
Área de concentração: Banco de Dados.

Orientadora:
Prof^a Dr^a Celia Aparecida Zorzo Barcelos

Uberlândia – Junho de 2009

Eduardo Ferreira Ribeiro

Caracterização de Imagens utilizando Redes Neurais Artificiais

Dissertação de Mestrado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.
Área de concentração: Banco de Dados.
Aprovação em 17 de março de 2009.

Banca Examinadora:

Orientadora: _____
Drª. Celia Aparecida Zorzo Barcelos – UFU

Avaliador 1: _____
Drª. Agma Juci Machado Traina – USP

Avaliador 2: _____
Drª. Gina Maira Barbosa de Oliveira – UFU

Uberlândia – Junho de 2009

Existem pessoas em nossas vidas
que nos deixam felizes pelo simples fato
de terem cruzado o nosso caminho. Algumas percorrem ao nosso lado,
vendo muitas luas passarem, mas outras apenas vemos entre um passo e outro.
A todas elas chamamos de amigo. Há muitos tipos de amigos. Talvez cada folha de uma árvore caracterize um deles. Os primeiros que nascem do broto é o amigo pai e a amiga mãe. Mostram o que é ter vida. Depois vem o amigo irmão, com quem dividimos o nosso espaço para que ele floresça como nós. Passamos a conhecer toda a família de folhas, a qual respeitamos e desejamos o bem. O destino ainda nos apresenta outros amigos, os quais não sabíamos que iam cruzar o nosso caminho. Muitos desse são designados amigos do peito, do coração. São sinceros, são verdadeiros. Sabem quando não estamos bem, sabem o que nos faz feliz... Mas também há aqueles amigos por um tempo, talvez umas férias ou mesmo um dia ou uma hora. Esses costumam colocar muitos sorrisos na face, durante o tempo que estamos por perto. O tempo passa, o verão se vai, o outono se aproxima, e perdemos algumas de nossas folhas. Algumas nascem num outro verão e outras permanecem por muitas estações. O que nos deixa mais felizes é quando as folhas que caíram continuam por perto, continuam alimentando as nossas raízes com alegria. Lembranças de momentos maravilhosos enquanto cruzavam o nosso caminho. Falando em perto, não podemos nos esquecer dos amigos distantes, que ficam nas pontas dos galhos, mas que quando o vento sopra, aparecem novamente entre uma folha e outra. Dedico este trabalho a você,
folha da minha árvore, simplesmente porque cada pessoa que passa
em nossa vida é única. Sempre deixa um
pouco de si e leva
um pouco de nós.
Há os que levaram
muito, mas há
os que não deixaram
nada. Esta é a maior
responsabilidade
de nossa vida e a prova
evidente de que duas almas não se encontram por acaso.

Agradecimentos

Agradeço primeiramente a Deus. Por vezes sinto que Ele, através do meu espírito está me ajudando, dando-me sabedoria, instigando-me a conhecer, convidando-me a me completar como ser humano, estimulando-me a fazer de mim uma expressão de Sua criação.

Aos meus pais José Ribeiro e Selmita e à minha madrasta Ana que eu tanto amo e que me educaram e fizeram de tudo para que eu apenas me preocupasse com os estudos.

À minha orientadora Professora Dra. Celia Aparecida Zorzo Barcelos, pela oportunidade de trabalhar com ela, pela paciência, pelos conselhos, pela grande ajuda e em me proporcionar um dos grandes sonhos de infância que era conhecer os Estados Unidos. Celia, muito obrigado por tudo!

Ao grande amigo, orientador, e colaborador deste trabalho, Professor Marcos Aurélio Batista, pelo incentivo, confiança e amizade, por todo o apoio prestado ao longo dos últimos 4 anos, pela paciência com que atendeu às minhas dúvidas, pela motivação que me deu com os seus constantes incentivos para a consolidação do meu futuro. Enfim, pela segurança que me transmitiu e me permitiu ultrapassar o receio pela inexperiência que tinha na investigação científica.

Ao professor Dr. Eraldo Ribeiro do *Florida Institute of Technology*, pela oportunidade de me proporcionar estagiar em seu laboratório (*Computer Vision Group at Florida Tech*) em Melbourne - FL, Estados Unidos para complementar minhas pesquisas, pela hospitalidade e atenção. Tal estágio viabilizou o capítulo 8 deste trabalho.

Agradeço aos meus irmãos e a toda minha família, que mesmo estando distantes, sempre me apoiaram e me deram forças para continuar. À minha melhor e eterna amiga Rejane, à qual eu confiei todos os meus segredos e meus problemas e que mesmo estando afastada fisicamente continua em meu coração. Aos grandes amigos Douglas e Núbia pela amizade, confiança, força e incentivo a seguir o caminho da ciência.

Aos irmãos na fé e grandes amigos Pr. Wellignton Alves, Pra. Íris Lúcia, Alexandre, Ana Paula, Guilherme e Fernando.

Aos meus amigos do mestrado que se mostraram companheiros e, diretamente ou indiretamente, contribuíram para a realização deste trabalho: Jean Carlo, Liliane, Rodrigo , Robson, Stéfano, Tauller, Valquíria, Vinícius Borges e Walter Dias. Sei que ainda falta muita gente para ser citada, mas tenho a convicção de que somente pude chegar a este resultado porque fui muito ajudado.

Sou imensamente agradecido à professora Dra. Denise Guliato pela grande ajuda no fim do mestrado e à Fapemig e CNPq pelo apoio financeiro.

Quero agradecer este trabalho também a você, que está percorrendo estas linhas agora. Por algum motivo, estamos agora em contato. Aproveito a forma desses agradecimentos para lhe dizer que a vida é fantástica. Através dela podemos realizar muitas coisas. Esta dissertação exigiu muito de mim. Tive que saber ouvir, refazer, repensar. Tive que aprender a conhecer os meus limites e a pedir ajuda. Aprendi também, a ser ajudado com dignidade, sabendo que a ajuda é parte da vida. Penso que pude fazer você sentir o quão grato eu sou a todos os que me ajudaram a atingir meu objetivo. Há outra coisa ainda: Cultive a gratidão, ela é uma credencial que habilita o ser humano à evoluir.

Resumo

Em sistemas de Recuperação de Imagens Baseada em Conteúdo a representação das imagens desempenham um papel fundamental. Os resultados obtidos por esses sistemas dependem fortemente da escolha das características selecionadas para representar uma imagem. Trabalhos existentes na literatura evidenciam que técnicas inteligentes conseguem minimizar o *gap*-semântico existente entre o poder de interpretação limitado das máquinas e a subjetividade humana.

Neste trabalho é proposto o uso das redes neurais artificiais para caracterizar imagens neurosemânticamente à partir de uma caracterização inicial baseada em características de baixo nível (cor, forma e textura).

Testes em 3 bases de dados de naturezas diferentes, um de imagens mais gerais (*BD-12750*), um de texturas (*Vistex-167*) e outro de prédios (*ZuBuD*) exemplificam a aplicação do método como também mostram a eficácia do modelo.

Ainda é apresentada a aplicação do método proposto na caracterização neurosemântica de movimentos complexos em vídeos.

Palavras-chave: caracterização de imagens, tranformação de características, características de baixo nível, recuperação de imagens, redes neurais artificiais.

Abstract

Image representation in Content Based Image Retrieval systems is a fundamental task. The results obtained by these systems strongly depend on the choice of features selected to represent an image. Works in the literature show that intelligent techniques are used to minimize the semantic gap between the limited power of machine interpretation and human subjectivity.

In this work the use of artificial neural networks to characterize images in a high-level space from an initial characterization based on low-level features (color, shape and texture) is proposed.

Experiments on 3 databases of various kinds, one with general images (*BD-12750*), one with texture images (*Vistex-167*) and other with buildings (*ZuBuD*) are performed to exemplify the application of the method and to show the effectiveness of the model.

Furthermore, the application of the proposed method in the high-level characterization of complex motions patterns is presented.

Keywords: image characterization, feature transformation, low-level features, neural networks, image retrieval, neural networks.

Sumário

Lista de Figuras

Capítulo 1

Introdução

1.1 Considerações Iniciais

O uso de imagens em nossa comunicação é bastante comum desde o início dos tempos. As pinturas rupestres gravadas em abrigos ou cavernas, em suas paredes e tetos rochosos demonstram que o desejo de expressão através das imagens é inerente ao ser humano. É curiosa a obstinação da humanidade para eternizar os momentos de sua vida, numa tentativa de dividí-los com as gerações futuras. A história da fotografia mostra bem isso. Os caminhos que antecederam aos aparatos atuais de registro de imagens são construídos de pequenas descobertas, aparentemente insignificantes, que vão culminar em grandes achados.

No século IV aC os gregos já conheciam o princípio da câmara escura, a partir da observação de que os raios de luz solar, penetrando num recinto fechado e escuro, através de um orifício, projetavam na parede oposta, imagens do exterior.

Já no século XI dC a câmara escura, que continha os princípios elementares da câmara fotográfica moderna, foi usada pela primeira vez para fins práticos, para a observação de um eclipse solar por um astrônomo árabe.

Na Renascença, Leonardo da Vinci descreveu essa câmara fotográfica minuciosamente e seus contemporâneos pintores e projetistas usaram-na largamente como importante método para auxiliar na construção da imagem. Data desse período os primeiros modelos portáteis projetados pelo italiano Gerônimo Cardano, diferentes das primeiras câmaras escuras, imensos quartos capazes de abrigar um homem no seu interior. Aos poucos foram surgindo outros tipos de câmeras mais leves e com dispositivos de maior controle de entrada de luz no filme e também equipamentos de tamanhos e formatos variados, além de finalidades específicas. Aperfeiçoaram-se as lentes objetivas, surgiram filtros para diversas finalidades. Hoje, vemos educadores e escritores utilizarem imagens para ilustrações, engenheiros e arquitetos em seus projetos, médicos para efetuar diagnósticos, cineastas para contar histórias, repórteres para realçar informação textual e pessoas em geral para apenas guardarem bons momentos.

Com o constante aperfeiçoamento da tecnologia essas práticas aumentaram ainda mais. Nos últimos anos, o crescimento do número de imagens em meio digital foi espantoso sendo que seu custo de processamento e armazenamento decresceu consideravelmente. A facilidade de distribuição advinda da popularização da internet e o aumento do poder computacional

foram decisivos para que isso ocorresse. Diversas são as áreas que contribuem para a geração de imagens digitais. Nestas, podemos citar as áreas de entretenimento, bibliotecas digitais e educação.

1.2 Motivação

A cada dia gigabytes de imagens são geradas por centros militares e sociais. No entanto, ao mesmo tempo, uma quantidade considerável de informação vem perdendo sua utilidade, pois não é possível ter acesso a essa informação se, previamente, ela não tiver sido organizada para permitir busca e recuperação eficientes de dados. Tradicionalmente, as imagens são armazenadas em bases de dados utilizando informação textual. A recuperação é realizada através de comandos da linguagem de consulta, os quais para este tipo de dado apresentam limitações. Uma dessas limitações é o fato de que é praticamente impossível fazer a anotação manual para todas as imagens. Além disso, as feições visuais das imagens são difíceis de serem descritas com palavras e também dependem da subjetividade humana para serem descritas. Por exemplo, diferentes pessoas podem fornecer informações diferentes sobre a mesma imagem, gerando respostas não padronizadas para pesquisas sobre esses assuntos.

Como soluções para estes problemas, buscam-se métodos que estendem as técnicas tradicionais de recuperação em bancos de dados, para incluir também informações visuais, permitindo, por exemplo, consultas baseadas no conteúdo das imagens. A essa tecnologia, dá-se o nome de *Content-Based Image Retrieval* (CBIR) ou Recuperação de Imagens Baseada em Conteúdo (RIBC). Cabe salientar que imagens são dados não-estruturados, que consistem de vetores com intensidades de pixel sem nenhum significado inerente. Uma das características do CBIR é a de extrair informações a partir das imagens, informações estas que venham a caracterizar seu conteúdo, utilizando principalmente atributos ou características de baixo nível como cor, textura e forma ou uma combinação destas para refletirem a semântica inerente da imagem.

Devido ao fato de que a percepção humana é o ponto de referência para a comparação da eficiência dos sistemas de recuperação, o desenvolvimento de um "sistema ideal"/ se torna cada vez mais desafiador. Da mesma maneira que a visão humana é considerada um processo inteligente, capaz de extrair padrões das imagens que as caracterize, assim como fazer inferências e generalizações através desses padrões associados a uma informação prévia, um sistema de recuperação ideal também deve ser capaz de extrair tais padrões e fazer tais inferências semânticas. Deste modo as propriedades de uma rede neural, tais como aprender através de exemplos, generalização de redundâncias e tolerância a falhas, proporcionam fortes incentivos para a escolha dessas redes como uma alternativa apropriada para uma modelagem de um sistema de recuperação baseado em semântica de alto nível ou neurosemântica. No entanto, em muitos sistemas atuais, os conceitos semânticos de alto nível são apresentados apenas com palavras chave [Zhang and Izquierdo 2007], o que pode limitar e comprometer o sucesso da consulta.

Neste trabalho vamos utilizar a força da estrutura das redes neurais e sua habilidade de adaptação e aprendizagem para a formação de vetores de características neurosemânticas. O uso do conhecimento adquirido pelo aprendizado das mesmas que generaliza os conceitos semânticos

da mente do usuário torna-se um aspecto importante para a redução do *gap*-semântico na recuperação de imagens baseada em conteúdo. Tal escolha também pode ser explicada pelo fato de que modelos de redes neurais lidam muito bem com dados imprecisos e situações não totalmente definidas, ou seja, uma rede treinada de maneira razoável tem a habilidade de apresentar soluções satisfatórias quando são apresentadas entradas que não estão presentes em dados já conhecidos por ela.

1.3 Objetivos

Essa pesquisa tem como objetivo fazer um levantamento bibliográfico de vários métodos que incluem a recuperação e a caracterização de imagens e as redes neurais e, além disso, propor uma forma de caracterização de imagens utilizando-se uma técnica da inteligência artificial, mais especificamente as redes neurais artificiais (rna's), para constituir fronteiras de decisão altamente não-lineares no espaço de características de baixo nível, formando um novo vetor de características neurosemânticas para mapear as características de baixo nível e salientar as mais importantes baseando-se em exemplos predefinidos.

Para isso, as rna's irão atuar como uma abstração matemática inspirada no cérebro humano para adquirir conhecimento através da experiência (treinamento da rede). A estrutura dessa rede é composta de várias unidades computacionais (neurônios artificiais) operando em paralelo, interconectadas total ou parcialmente. Cada neurônio artificial efetua certo número de operações simples e transmite seus resultados aos neurônios vizinhos, com os quais possui conexão [Russel and Norvig 2004]. O treinamento da rede é realizado retificando os pesos nas conexões para instituir as relações entre as características extraídas das imagens e as classes para gerar a melhor discriminação possível entre os padrões de diferentes classes. Assim, quando um novo padrão é apresentado à rede, esta indicará a classe que melhor o representa na camada de saída [Haykin 2001]. Após realizado este treinamento, os pesos da rede estarão prontos para serem usados na fase de caracterização neurosemântica. Na caracterização neurosemântica a rede neural treinada irá propagar as entradas (vetores de características de baixo nível) e utilizar a saída da rede para compor um vetor de características neurosemânticas que serão utilizados como parâmetro de similaridade entre as imagens para uma avaliação de performance.

1.4 Organização da Dissertação

Na intenção de expor com clareza os conceitos, as técnicas e os resultados obtidos neste trabalho, estruturamos o mesmo em duas partes. A parte 1 compreende os cápítulos 1,2,3,4 e 5 onde é feito um levantamento bibliográfico sobre o estado da arte da recuperação e caracterização de imagens, e uma descrição mais detalhada sobre as redes neurais artificiais com suas arquiteturas, parâmetros, métodos e algoritmos de treinamento. A parte 2 composta pelos capítulos 6,7 e 8 apresenta o modelo proposto por este trabalho com uma análise de resultados.

O **Capítulo 1** consiste das considerações iniciais e do contexto no qual se insere esse trabalho, da motivação para o seu desenvolvimento e dos objetivos a serem alcançados.

O **Capítulo 2** apresenta uma base teórica sobre o Processamento de Imagens e introduz uma visão geral sobre as várias concepções a respeito do uso das bases de imagens. A ênfase

da discussão nos princípios e métodos utilizados na recuperação de imagens baseada em conteúdo. Alguns problemas comuns e dificuldades dessas ferramentas são discutidos. Além disso, métodos de avaliação de sistemas de recuperação de imagens e algumas técnicas para redução do *gap*-semântico em sistemas de recuperação são explanados.

O **Capítulo 3** explana a extração de características expondo as principais técnicas utilizadas atualmente. Este capítulo está relacionado com técnicas de reconhecimento de padrões da área de Visão Computacional para a formação do vetor de características de baixo nível que permite a descrição das imagens. Além disso, são discutidos, ao final do capítulo, técnicas de seleção, transformação e redução de características.

No **Capítulo 4** as Redes Neurais Biológicas são introduzidas para explicar o anseio do homem em se basear na natureza para resolver problemas da vida cotidiana, dando assim uma base à explicação da implementação das Redes Neurais Artificiais apresentadas no capítulo seguinte.

O **Capítulo 5** apresenta os conceitos necessários para o entendimento das Redes Neurais Artificiais, bem como as arquiteturas, a representação do conhecimento e os algoritmos de treinamento necessários para a implementação do modelo proposto.

A segunda parte da dissertação é dedicada à descrição da técnica proposta, análise e comparações dos resultados obtidos, bem como propostas de trabalhos futuros:

No **Capítulo 6** o modelo proposto por esse trabalho é descrito e exemplificado, demonstrando todas as etapas do novo sistema de recuperação desde a fase de implementação e treinamento da rede neural até a fase de consulta, passando pela caracterização neurosemântica das imagens.

No **Capítulo 7** são descritos os testes baseados no modelo proposto e apresentados alguns resultados experimentais utilizando este método.

No **Capítulo 8**, uma breve introdução ao processamento de vídeos é feita, bem como a descrição de um algoritmo de caracterização de vídeos e a aplicação do modelo proposto para caracterizar os vídeos neurosemânticamente. Resultados e comparações com outros modelos são realizadas e uma conclusão bem como perspectivas para trabalhos futuros é realizada.

Finalmente as conclusões e a proposta de trabalhos futuros para a continuação dessa pesquisa são apresentadas no **Capítulo 9**.

Parte I

Revisão de Literatura

Capítulo 2

Recuperação de Imagens

Sistemas chamados de Recuperação de Imagens Baseada em Conteúdo ou *Content Based Image Retrieval* (CBIR) utilizam o conteúdo visual das imagens para buscar e recuperar imagens semelhantes a uma determinada imagem consulta em um dado banco de dados. O ponto chave desse sistema é a obtenção de dados discriminantes que corresponderiam a um possível julgamento humano da similaridade entre a imagem consulta e as outras pertencentes ao banco de imagens. Em sistemas CBIR, as imagens são indexadas por um vetor de características derivadas diretamente do conteúdo visual das mesmas. Esses atributos possuem informações chamadas por muitos autores de características de baixo nível como as cores, a textura, a forma e as relações espaciais que uma imagem possui. Este capítulo irá discutir vários tópicos sobre a recuperação de imagens focando a busca, a indexação e a recuperação das mesmas.

2.1 Conceitos Preliminares

Definir a palavra imagem pode se tornar uma tarefa árdua, especialmente se a atenção se voltar para as características pessoais, emocionais e culturais, que permitem a identidade própria desse termo. Assim como diz o antigo provérbio chinês: "Uma imagem vale mais que mil palavras..." na Ciência da Computação pode-se afirmar que uma imagem possui uma gama enorme de dados para serem interpretados tanto quantitativa quanto qualitativamente. Para cada abordagem a imagem pode ser definida e tratada sob diferentes aspectos. Do ponto de vista da arte, por exemplo, uma imagem pode ser a caracterização ou a reprodução do pensamento humano em uma tela, escultura, gravura, desenho ou pintura representada em algum objeto. Para a física uma imagem é um conjunto de pontos que convergem em um plano. Mas se for analisada ainda no domínio das idéias, uma imagem pode ser um apoio para que se desempenhem trocas de informações [de Holanda Ferreira 2004].

Sistemas de algoritmos de processamento de imagens devem considerar as características da percepção visual humana pois a ciência desse sistema e de determinadas técnicas como realce, restauração, e reconstrução de imagens de acordo com suas aplicações se tornam muito importantes por considerar critérios de fidelidade de resultados que dependem das características psicofísicas principalmente para aumentar os recursos de sistemas de processamento de imagens[Fischler and Firschein 1987].

2.1.1 Noções da Percepção Visual Humana

A visão é considerada, por muitos, um dos sentidos mais importantes para perceber o ambiente que cerca o ser humano. A sua capacidade de interpretação visual de grandes quantidades de dados desperta o interesse do desenvolvimento de técnicas e dispositivos, para que essas habilidades e sensibilidade se estendam cada vez mais.

No universo o trânsito de energia se faz, principalmente, através de radiações eletromagnéticas as quais oscilam sem necessitar de um meio para se propagarem. De acordo com cada meio de onde se propagam, suas amplitudes podem se modificar. Assim cada comprimento de onda caracteriza um espectro eletromagnético com nomes particulares, como os raios gama, infravermelho ou a luz visível [de Melo Aires 1999].

Plantas e animais desenvolveram em seu processo evolutivo, algumas estruturas para utilizarem essas radiações eletromagnéticas como fonte de informação. Esse tipo de transferência de energia pode ser recebida através de pigmentos carotenóides localizados em mecanismos biológicos específicos. Dá-se a essa estreita banda de freqüência capaz de excitar o sistema visual humano o nome de Luz Visível [de Melo Aires 1999].

Quando os raios luminosos, refletidos pela superfície dos objetos são recebidos pelo sistema visual humano, faz-se necessário o ajuste do foco e da luminosidade desses sinais. A retina codifica a informação visual em um padrão neuronal como uma câmara fotográfica que focaliza a luz no plano de um filme. A figura 2.1 apresenta um modelo de estrutura do olho humano. Nela pode-se observar que a luz entra no olho pela córnea, passa pela pupila que controla a quantidade da mesma e atravessa o cristalino. Ambas, córnea e cristalino, são espécies de lentes convexas e convergentes flexíveis sendo controladas pela tensão dos músculos ciliares. Depois disso, a luz atravessa o humor vítreo, substância gelatinosa para o preenchimento do globo ocular dando forma a ele, e atinge a retina. As células nervosas presentes na retina, também chamadas de fotoreceptores, convertem a imagem em sinais elétricos e enviam esses sinais ao cérebro através do nervo óptico [de Melo Aires 1999].

Dois tipos de fotorreceptores transformam o estímulo luminoso em estímulo elétrico: os bastonetes que respondem à iluminação fraca e são em menor quantidade (cerca de 6 e 7 milhões) e os cones, extremamente sensíveis às cores, respondendo à iluminação mais intensa, estes em maior quantidade (entre 75 e 150 milhões).

Após a informação ser enviada ao cérebro o mesmo armazena em uma memória compatível com o sistema em questão. Atualmente, várias teorias são discutidas a respeito da função da memória animal, mas pode-se concluir que a questão primordial é que o sistema, seja ele biológico ou digital, tenha acesso a essa informação para utilizá-la e manipulá-la de modo hábil quando necessário. Existem várias abordagens para explicar essa manipulação de informação pelo cérebro humano. Uma delas seria considerar que o cérebro leva em conta uma hierarquia taxonômica que associa semelhanças, ou coisas semelhantes e separa coisas não-semelhantes para realizar comparações entre a informação captada e alguns padrões de situações guardadas nessa base gerando, assim, uma resposta adequada para cada situação. Se essa informação for desconhecida ou nova, o cérebro irá rotulá-la na base de dados e enriquecê-la. Esse assunto será retornado ao longo desse trabalho.

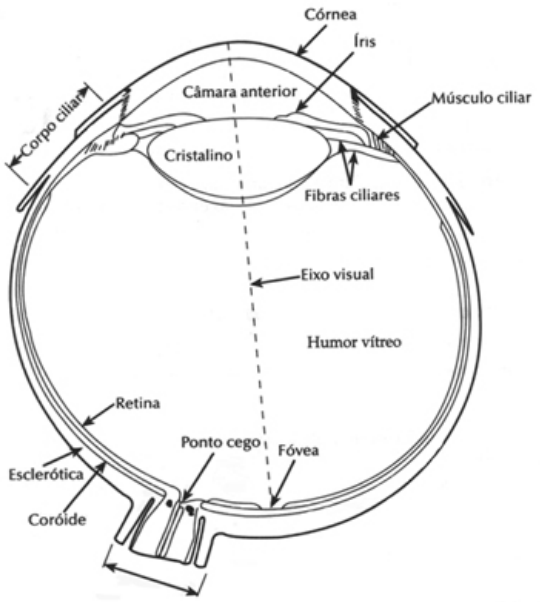


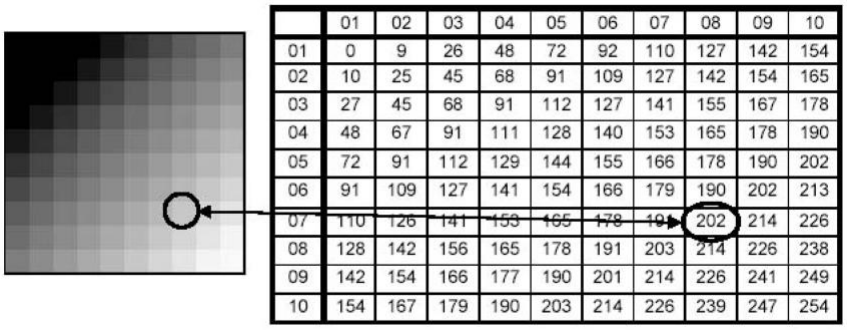
Figura 2.1: Estrutura do olho humano.

2.1.2 Fundamentos da Imagem Digital

Objetivos diferentes levam a modelos de representações de imagens diferentes. No processamento de imagens por exemplo usa-se geralmente o modelo matricial enquanto na computação gráfica usa-se o modelo vetorial.

A representação vetorial de um objeto é uma tentativa de representá-lo tão próximo quanto possível do real através de um vetor, procurando definir precisamente todas as posições, comprimentos e dimensões das entidades por sua natureza geométrica através da definição de uma função de elementos geométricos e parâmetros. Já a representação matricial pode ser definida como um conjunto de células colocadas em uma grade retangular de duas dimensões com coordenadas subsequentes e inteiras. Cada célula dessa matriz pode ser chamada também de "*pixel*" (abreviatura de "*picture elements*") e os objetos são representados utilizando adequadamente esses elementos da figura. Uma imagem pode ser representada matricialmente por um computador quando ela estiver discretizada ou digitalizada tanto no domínio espacial, através da matriz de ocorrências citadas anteriormente, como no das amplitudes através dos níveis de luminosidade de cada nível, ou seja, cada ponto dessa matriz bidimensional é dada pela função $f(x, y)$ que representa um nível de cinza para imagens monocromáticas [Castleman 1995]. A Figura 2.2 mostra um exemplo de uma imagem monocromática de tamanho 10×10 representando valores dentro da faixa de 255 níveis de cinza (como mostrado na matriz ao lado da imagem).

Para a representação de funções contínuas, os computadores digitais podem simulá-la. Para que essa simulação seja feita deve-se discretizar ou digitalizar essa função. Tomando valores pontuais x ao longo do intervalo de definição da função, obtendo seu valor correspondente $f(x)$ e armazenando-os em uma tabela por exemplo. Em uma imagem digital discretizar x chama-se



	01	02	03	04	05	06	07	08	09	10
01	0	9	26	48	72	92	110	127	142	154
02	10	25	45	68	91	109	127	142	154	165
03	27	45	68	91	112	127	141	155	167	178
04	48	67	91	111	128	140	153	165	178	190
05	72	91	112	129	144	155	166	178	190	202
06	91	109	127	141	154	166	179	190	202	213
07	110	126	141	153	165	178	191	202	214	226
08	128	142	156	165	178	191	203	214	226	238
09	142	154	166	177	190	201	214	226	241	249
10	154	167	179	190	203	214	226	239	247	254

Figura 2.2: *Pixels* de uma imagem.

amostragem e discretizar $f(x)$ chama-se quantização. Uma função contínua que se estende infinitamente geralmente é limitada a um intervalo do domínio para então ser discretizada, passando primeiro pelo processo de amostragem e depois pelo processo de quantização.

Ao se constituir o sinal analógico de uma imagem para o sinal digital, definindo seu espaço em uma matriz, temos a amostragem dessa imagem, na qual se torna computacionalmente manipulável. Quanto mais pixels uma imagem tiver, melhor é a sua resolução e qualidade. Os dispositivos responsáveis pela obtenção ou captura de imagens são denominados sistemas de imageamento. Para que esses sistemas operem corretamente é necessário que se faça uma digitalização da imagem, ou seja, transformar um sinal analógico, contínuo em um sinal digital, discreto. Um dos exemplos mais comuns de amostragem presente em muitos dispositivos de captura é a amostragem uniformemente espaçada, onde cada amostra é tomada em intervalos iguais. Existem também outros tipos de amostragem onde os intervalos podem ser irregulares ou mais espaçados. A figura 2.3 demonstra um exemplo de amostragem para alguns valores de resolução.

Figura 2.3: Amostragem de uma imagem.

No processo de quantização os tons contínuos da imagem são transformados em tons discretos armazenados em uma matriz por um código binário. Dependendo da precisão ou da

minimização da quantidade de bits, esse código é formulado visando o armazenamento ou a transmissão dos dados. Os métodos de compactação buscam otimizar os erros entre a imagem original e a discretizada procurando manter-se o mais fiel possível às suas características. A quantização de cor segue os mesmos passos para o plano da imagem e será melhor explicada na seção de cores, onde poderá ser encontrada sua definição, suas representações e como trazê-la para o mundo digital. A Figura 2.4 apresenta um exemplo da quantização para diferentes valores.

Um dos métodos de quantização mais comuns é a tomada do valor máximo da função e do valor mínimo da mesma, dividindo esse intervalo em partes constituintes iguais de acordo com o número de bits estabelecido para armazenar uma amostra. Logo o número de valores possíveis será de 2^{nbits}.

Imagens que possuem mais de uma banda de frequência, devem ser representadas por mais de uma função $f(x, y)$ como por exemplo as imagens coloridas, que apresentam (no modelo de cores RGB) uma função de intensidade para cada cor primária: azul, verde e o vermelho. A sobreposição destas três matrizes monocromáticas, compõem uma imagem colorida.

Figura 2.4: Quantização de uma imagem.

2.2 Bancos de Imagens

Para se explanar o tema de indexação e recuperação de dados visuais, obviamente, o primeiro passo é organizar as imagens em um banco de dados ou banco de imagens. Banco de imagens nada mais é que uma coleção de figuras em um repositório ou estrutura de dados específica como os bancos de dados. Bancos de dados podem ser definidos como conjuntos de dados com uma estrutura regular que organizam informação, onde as informações são agrupadas, de tal forma que seja possível localizar itens escolhidos. Uma das funções básicas de um banco de dados é facilitar o acesso a dados relevantes. Para implementar essa função em um banco de imagens, é necessária uma maneira efetiva de indexar as imagens e recuperá-las correspondendo à necessidade do usuário.

Algumas aplicações para banco de imagens já podem ser encontradas como, por exemplo, em banco de dados de imagens médicas por raios-X e tomografias computadorizadas, arquivos

de imagens de jornais, arquivos de impressões digitais, sensoriamento remoto, dados geográficos dentre vários outros. Em particular, o uso de banco de dados para aplicações médicas e imagens de satélites vêm chamando muita atenção em projetos de pesquisa, como por exemplo, a tecnologia criada pela Nasa chamada *NASA´s Earth Observing System* (EOS) [Plan et al. 1998] para análise e recuperação de imagens.

Outra importante aplicação para banco de imagens e tecnologias de recuperação está dentro da rede mundial de computadores ou *World Wide Web* (WWW) com suas bilhões de imagens. Os mecanismos de busca de imagens ainda são precoces se comparados a mecanismos de busca de texto, que são ferramentas robustas para a busca de informação nas páginas de texto da WWW (principalmente as páginas construídas em linguagem de marcação como a HTML). Devido à sua utilidade, não é de se assustar que as páginas mais visitadas atualmente são os portais que incluem alguma ferramenta de busca textual.

Como resultado disso, a necessidade de ferramentas efetivas para a busca e recuperação de imagens nos diferentes tipos de banco de imagens é imensa. As tecnologias presentes para realizar essas funções, ainda são imaturas e inadequadas para a maioria das aplicações práticas.

2.3 Descrições Textuais

A maneira mais tradicional de recuperação de imagens ao longo das ultimas décadas foi baseada em palavras-chave inseridas manualmente na ferramenta de recuperação. Essas palavras-chave descrevem o conteúdo da imagem e outras informações relevantes como, por exemplo, onde e quando a imagem foi criada. O próprio usuário formula questões textuais as quais se tornam palavras-chave no mecanismo de busca.

Existem, no entanto, algumas desvantagens nos métodos de descrição textual. A primeira delas, é que a descrição é feita manualmente. E para representar o conteúdo completo de uma imagem, o descritor humano deve prover uma descrição global para todas as características e relações espaciais dos objetos, além de outras relações com outros objetos da mesma. Esse processo de descrição pode se tornar impossível visto que algumas imagens possuem muitos detalhes. Este método também se torna impraticável à medida que o banco de dados aumenta em tamanho, pois as pessoas tendem a fazer anotações relevantes apenas nas imagens que as interessam. Em grandes e dinâmicos bancos de dados, como aqueles contendo índices das imagens armazenadas na WWW, esta aproximação ainda é um desafio. As descrições podem ser eventualmente mudadas como, por exemplo, alguns atributos previamente excluídos podem se tornar aspectos importantes e, conseqüentemente, a imagem precisa ser indexada novamente. Se isso for muito frequente, até a descrição de uma pequena coleção de imagens se torna uma tarefa entediante e desinteressante.

O segundo problema diz respeito à natureza rica e subjetiva que as imagens geralmente possuem. Como mencionado na seção anterior o provérbio que diz: "uma imagem vale mais que mil palavras" pode indicar a natureza incompleta das palavras para se descrever uma imagem. E mesmo se as informações forem descritas detalhadamente para cada imagem no banco de dados, há um problema com diferentes interpretações do conteúdo das imagens por diferentes pessoas. Se o banco de dados for grande, as descrições provavelmente serão feitas por várias pessoas e não somente por uma o que diversificaria ainda mais as interpretações

das imagens. O usuário teria que saber os termos exatos os quais o interpretador humano utilizou para que o sistema recupere a imagem que ele está procurando. Habitualmente, isso não ocorre, pois o usuário não irá possuir conhecimento sobre a formação do banco de dados, e menos ainda, as descrições concisas e consistentes para identificar um grupo de imagens.

No entanto, os problemas descritos acima não tornam a informação textual sobre o conteúdo da imagem obsoleta. Os métodos atuais de extração automática de atributos estão longe de serem perfeitos e nem sempre são capazes de extrair toda e qualquer informação para uma perfeita recuperação de imagens. Com isso, se a informação textual é eficaz, poderá servir de auxílio para a Recuperação de Imagens Baseada em Conteúdo.

Um exemplo do uso da informação textual como apoio a ferramentas de recuperação baseada em conteúdo é descrito por [Srihari 1995]. Seu sistema nomeado Piction utiliza legendas textuais de fotografias de jornais em conjunto com características visuais para a identificação de faces nas imagens. Uma ferramenta de busca de imagens na World Wide Web também poderia ser implementada usando informação textual disponível sobre o conteúdo da imagem. Tais fontes seriam as páginas HTML, especialmente a tag ALT, e o nome do arquivo da imagem, dentre outros. Um exemplo de sistema semelhante é o Google Images e esse tipo de procedimento será melhor explanado nas seções seguintes.

2.4 Recuperação de Imagens Baseada em Conteúdo

Como foi dito na introdução deste capítulo, na Recuperação de Imagens Baseada em Conteúdo (CBIR), as imagens são indexadas pelos atributos diretamente derivados do seu conteúdo visual. As características são sempre consistentes com a imagem e e podem ser extraídas automaticamente por um computador ao invés de descrições manuais.

Em sistemas CBIR, a informação extraída das imagens usualmente é formada por características de baixo nível, como as cores, texturas, formas e a estrutura apresentada. As características usadas deveriam, a princípio, ser correlatas a uma possível subjetividade de similaridade por um observador humano. Um dos problemas da Recuperação de Imagens Baseada em Conteúdo é que os atributos automaticamente extraídos apenas descrevem diferentes materiais, padrões, conteúdo de cor e formas, mas são inadequados para classificar o conteúdo semântico das imagens. Além disso, métodos automáticos geralmente se concentram em propriedades locais de imagens, sendo que a percepção humana das imagens é mais holística. O holismo (do grego *holos*, todo) é a idéia de que as propriedades de um sistema não podem ser explicadas apenas pela soma de seus componentes. A palavra foi cunhada por Jan Smuts por volta de 1920, governador britânico no sul da Índia, que assim a definiu: "É a tendência da natureza formar, através de evolução criativa, todos que são maiores que a soma de suas partes". O uso de exemplos para ponderar essas características se torna uma alternativa promissora para a recuperação de imagens suprimindo o problema da subjetividade humana em relação à semântica das mesmas [Hammerton 1998].

O reconhecimento automático de objetos é, em geral, um problema difícil e ainda não resolvido por inteiro. Uma solução para isso seria o desenvolvimento de algoritmos mais sofisticados para a extração de características. Por exemplo, poder-se-ia adicionar um detector de forma feito por encomenda para todos os diferentes objetos no quais o usuário estaria inter-

essado em localizar nas imagens. Uma importante aplicação para esse tipo de sistema é a detecção de faces [Rowley et al. 1998] [Moghaddam et al. 1998]. Infelizmente, essas aproximações ultrapassam os limites de nosso atual conhecimento nos métodos de visão computacional. Algoritmos adequados, na maioria dos casos, também são difíceis de implementar ou simplesmente não existem. Além disso, atributos complicados usualmente são computacionalmente caros, os quais podem se tornar um problema em um banco de dados suficientemente grande.

A Recuperação de Imagens Baseada em Conteúdo vem sendo objeto de pesquisa a um bom tempo. Como a demanda de aplicações práticas (como em sistemas de auxílio a diagnóstico médico) é evidente, e o paradigma de descrição textual possui sérias limitações, a pesquisa nessa área vem sendo bem ativa nos últimos anos. Um grande número de conferências focando esse tópico vem sendo organizado e muitos jornais e revistas especializadas têm publicado vários artigos sobre o CBIR. Para uma melhor visão compreensiva do assunto pode-se citar, por exemplo, os artigos [Silva et al. 2006], [Aigrain et al. 1996], [Rui et al. 1997], [Gupta and Jain 1997], [Liu et al. 2007] e [Deselaers et al. 2008].

2.4.1 Métodos de Consulta

A busca de imagens com as ferramentas CBIR são usualmente executadas utilizando uma imagem ou imagens consulta. A tarefa do sistema é recuperar as imagens mais semelhantes à(s) imagem(ns) referência. Esse método relativamente antigo é geralmente chamado de Consulta Através de Exemplo (*Query by Pictorial Example*) ou QBE. A figura 2.5 especifica uma típica arquitetura de um sistema de Recuperação de Imagens Baseada em Conteúdo. A recuperação começa quando o sistema apresenta uma seleção inicial de imagens de referência através de uma interface. O sistema pode apresentar alguma imagem representativa ou pode escolher aleatoriamente. O usuário então seleciona uma ou mais imagens relevantes, e a imagem selecionada é usada para recuperar as imagens mais semelhantes a ela dentro de um banco de dados.

Uma desvantagem nos sistemas de Consulta Através de Exemplo é que o sucesso da consulta depende consideravelmente da coleção inicial de imagens. Em um banco de dados grande, a seleção das imagens iniciais pode se tornar um problema, pois essa seleção deveria ter pelo menos uma imagem pertinente à desejada pelo usuário. No artigo de La Cascia [Cascia et al. 1998], este problema é chamado de problema da página zero. Além disso, sistemas QBPE claramente necessitam de uma definição sólida para a similaridade entre as imagens. Nos dias atuais o problema de página zero já foi contornado, visto que em muitos sistemas o usuário faz a seleção inicial levando a imagem escolhida por ele de fora para dentro do banco de dados a qual será comparada com todas as outras dentro do banco. Essa imagem é também chamada de imagem consulta. Após isso, o sistema aceita essa escolha e caracteriza a imagem formando um vetor de características que é armazenado em um banco de dados contendo as características extraídas bem como as próprias imagens.

O banco de imagens poderá ser a própria rede mundial de computadores (WWW), nesse caso o sistema deverá saber o endereço onde a imagem está localizada (indexação de imagens) para a sua recuperação. Após essa(s) imagem(ns) ser(em) caracterizada(s), os vetores de características presentes no banco de dados são comparados com o vetor consulta através de computação da similaridade. Esta similaridade pode ser feita de várias maneiras. As imagens

mais similares formarão um *ranking* (primeira mais semelhante, segunda mais semelhante e assim por diante) e são apresentadas ao usuário também através da interface do sistema.

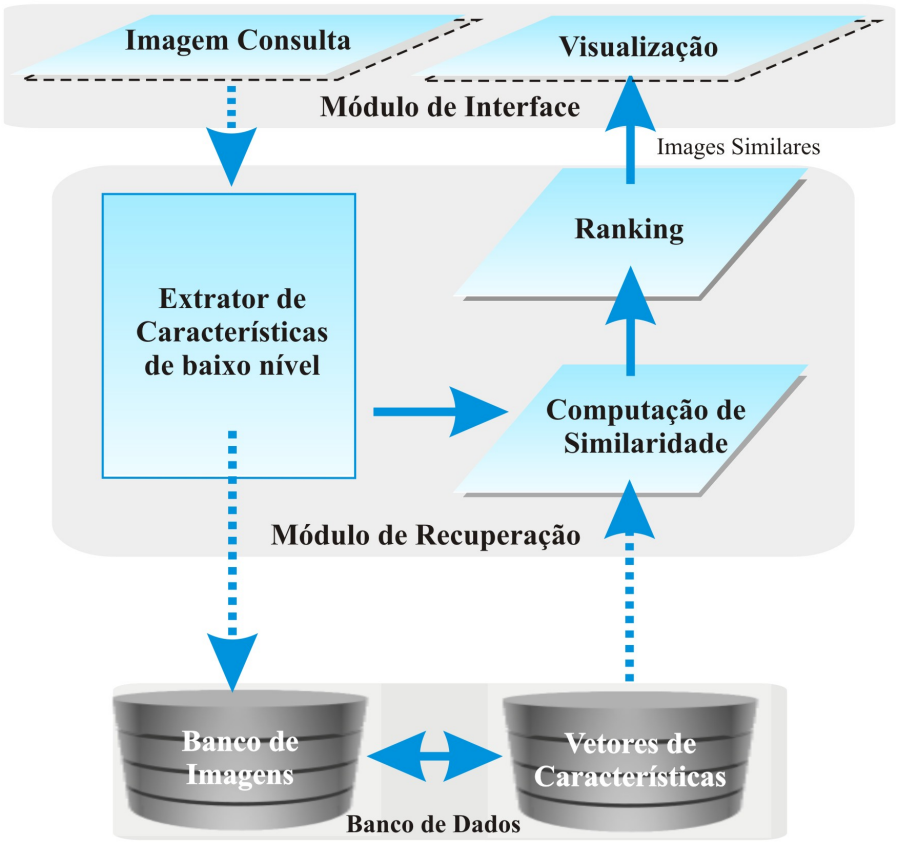


Figura 2.5: Arquitetura típica de um sistema de Recuperação de Imagens Baseada em Conteúdo.

O suporte de linguagem natural para a consulta em sistemas CBIR também podem ser úteis, especialmente para usuários inexperientes sem o conhecimento da implementação da ferramenta de recuperação. O usuário poderia então formular questões de banco de dados tradicionais, como "encontre carros vermelhos" ou "mostre-me imagens de borboletas coloridas", até mesmo se os bancos de imagens não tiverem nenhuma descrição textual das mesmas. O mapeamento entre a semântica da linguagem natural e as características visuais de baixo nível é, no entanto, uma questão difícil. Um exemplo de Recuperação de Imagens Baseada em Conteúdo utilizando linguagem natural é dada por [Harada et al. 1997].

2.4.2 Conceitos versus Características

Mesmo depois do considerável esforço nas pesquisas realizadas durante a última década, ainda existem problemas persistentes a respeito da Recuperação de Imagens Baseada em Conteúdo e, atualmente, ainda não existem sistemas tão robustos quanto os sistemas de busca baseados em descrição textual. A principal dificuldade se encontra no gap-semântico entre as

concepções semânticas de alto nível usadas pelos seres humanos para entender o conteúdo de uma imagem e as características de baixo nível usadas na visão computacional [Rui et al. 1999]. Para um computador, a extração do conteúdo semântico de uma imagem é uma tarefa árdua, pois muitos objetos possuem o mesmo conteúdo semântico, mas são totalmente diferentes na aparência visual, e muitos objetos semanticamente diferentes possuem similaridade visual [Gupta and Jain 1997]. Até mesmo a segmentação automática de objetos é uma tarefa difícil. Os humanos, por outro lado, possuem muita informação a priori sobre diferentes objetos, que são usadas prontamente pra reconhecê-los. Ou seja, essa informação é obtida através de uma experiência prévia, preferências pessoais ou interesses, assuntos pessoais, e o contexto em que as imagens são apresentadas. Este tipo de conhecimento é inerentemente difícil de se replicar em uma aplicação de visão computacional.

Geralmente, não existe um algoritmo de conversão direta de conceitos de alto nível para características de baixo nível. Por exemplo, considere uma imagem de um avião voando no céu. A imagem é composta de um avião como um objeto distinto e um fundo bem limpo e azul. Apesar do fundo da imagem ser claro, a cor do avião pode variar. Um avião não possui uma textura específica e pode diferir de acordo com o modelo. A forma depende substancialmente do ângulo da câmera e, então, o mesmo avião pode aparecer de várias formas em diferentes imagens. A cor, a textura e a forma possuem algumas informações úteis para encontrar imagens de avião. Mesmo assim, é difícil escolher os valores corretos para essas características para que possa haver uma consulta ótima e uma correspondência correta para a concepção semântica de um avião no céu. O sistema de recuperação poderia buscar imagens de navios em alto mar, por exemplo. As características simples comumente usadas na Recuperação de Imagens Baseada em Conteúdo não são suficientes para se fazer uma classificação e rotulação satisfatória em coleções de imagens gerais. Uma criança de três anos pode facilmente identificar todas as figuras de cachorro em um livro infantil, mas para computadores isso pode se tornar um problema [Flickner et al. 1995].

As diferenças entre a concepção semântica e as características de baixo nível se tornam uma limitação séria e podem reduzir a usabilidade das técnicas baseadas em conteúdo. Como resultado disso, muitas aplicações baseadas em conteúdo não são auto-suficientes e não podem ser aproveitadas de modo eficaz em todas as áreas de aplicação. Elas podem, todavia, servir de valiosas ferramentas semi-automáticas para fazer a recuperação de imagens até mesmo para uma grande escala e coleções de imagens vagas. Além disso, o *gap* entre o ser humano e o computador não é igualmente grande em todas as áreas de aplicação. Em alguns domínios não alteráveis e precisamente definidos com imagens bem distintas e previsíveis, a aproximação baseada em conteúdo com características simples pode trabalhar tão bem quanto se queira. Os grupos de pesquisa ainda não alcançaram um método reconhecidamente ideal, aplicável a vários domínios, pelo próprio desconhecimento de como as relações são feitas no cérebro humano. Assim sendo, muitas pesquisas estão sendo realizadas com áreas como a psicologia, arquitetura, educação e a neurociência caracterizando a interdisciplinaridade do tema.

2.4.3 A Subjetividade Humana

Outro problema na Recuperação de Imagens Baseada em Conteúdo é a subjetividade humana da percepção. Diferentes pessoas vêem e descrevem diferentes aspectos em uma imagem,

e assim, qualificam de modo diferente imagens iguais. Um usuário pode, por exemplo, estar interessado no conteúdo cromático de uma imagem, enquanto outro estar procurando por certa textura. Este assunto será melhor tratado no próximo capítulo sobre a extração de características.

A subjetividade também pode ser afetada no contexto semântico da imagem. As pessoas podem perceber a mesma imagem de várias maneiras, dependendo do contexto em que elas se encontram e do tempo da consulta. Os artistas são exemplos da representação da subjetividade. Através das artes e de algumas obras pode-se tirar várias interpretações, conlusões, sentimentos, etc., de acordo com a carga emocional do observador bem como sua cultura e seus conhecimentos prévios [Jorgensen 1998].

Um sistema estático de recuperação de imagens não pode, dessa forma, executar de maneira ótima com todas as possíveis consultas. O sistema de recuperação deve ser capaz de aprender através da interação com o usuário e ajustar seus parâmetros para presumir as expectativas do mesmo. Se o sistema de recuperação for baseado em várias características, pode-se escolher as mais relevantes para serem usadas na consulta corrente ou algum tipo de fator de pesos para controlar a contribuição de cada característica no resultado final. As redes neurais podem, através de exemplos de classes de padrões invariantes, serem usadas para ponderar esses atributos formando uma fronteira não linear no espaço de características.

2.4.4 Similaridade entre Imagens

Determinar o nível de similaridade entre diferentes imagens é uma tarefa elementar na recuperação de imagens. Obviamente, uma medida útil de similaridade deve produzir grandes valores para imagens similares e pequenos valores pra imagens diferentes (ou vice e versa dependendo do algoritmo usado). Como a base dos sistemas CBIR é recuperar imagens relevantes para o usuário, as medidas de similaridade entre imagens usadas pelo sistema de recuperação devem corresponder à noção de similaridade entre imagens para o usuário. Isso só será possível se houver uma grande afinidade entre as medidas de semelhança humana e da máquina. A validade dessa suposição é analisada por [Squire and Pun 1997].

Consultas de imagem podem ser vistas como procedimentos que "pareiam" a imagem consulta com a imagem mais similar ou com o conjunto de imagens mais semelhantes disponíveis no banco de dados. A partir disso, a performance do sistema de recuperação dependerá do critério de similaridade escolhido.

Uma medida comum para medir a similaridade entre imagens é a distância. Duas imagens podem ser comparadas pela medição da distância entre elas de acordo com alguma função satisfatória. Várias funções de distância foram desenvolvidas para a comparação de imagens. O livro de Rubner e Tomasi [Rubner 1999] e o artigo de Vito Di Gesù e Valery Starovoitov [Gesù and Starovoitov 1999], por exemplo, podem ser considerados boas referências sobre métricas, onde são classificadas e analisadas algumas das medidas de similaridade mais comuns.

Em uma implementação de banco de dados tradicional, o usuário faz consultas e somente os itens comparados na consulta são retornados. A comparação é uma operação fundamental em operação de banco de dados, que consiste em relacionar os itens do banco de dados com a consulta corrente e decidir se o item satisfaz os termos da consulta ou não. Com banco

de dados de imagens é difícil encontrar o critério correto de casamento de padrões binários para escolher as imagens desejadas. Assim, diferentes aproximações vêm sendo desenvolvidas para serem usadas em banco de dados de imagens. Ao invés de se emparelhar, as imagens são ordenadas por algum critério de similaridade escolhido [Santini and Jain 1996]. Isso resulta em uma permutação de todos os itens no banco de dados ordenados de acordo com a medida de similaridade usada como resposta para a questão. Um determinado número de imagens que formam esse ranking é, então, mostrado como resposta da consulta para o usuário. Uma definição de Consulta de Imagens Baseada em Conteúdo, adotada por [Smith 1997] é dada a seguir:

> **Consulta de Imagens Baseada em Conteúdo:** *Dado um banco de imagens D com N imagens e uma função de similaridade entre características $d(q,i)$, encontre as N primeiras imagens $i \in D$ com a maior similaridade $d(q,i)$ para a imagem consulta q.*

Por exemplo, se os objetos tiverem sido caracterizados por vetores de características de igual tamanho, a distância $d(q,i)$ é a distância euclidiana entre eles dada pela equação 2.1.

$$d(q,i) = (\sum_{k=1}^{n}(q_k - i_k)^2)^{1/2} \tag{2.1}$$

Quando a imagem é segmentada em regiões as medidas de similaridade podem ser feitas sob dois níveis:

Nível de Regiões É a comparação de 2 regiões baseando-se em seus atributos de baixo nível.

Nível de Imagens É a comparação global de duas imagens que podem conter diferentes números de regiões.

Várias pesquisas empregam a métrica de Minkowski para definir distância entre regiões. Sejam duas regiões representadas por dois vetores de características p-dimensionais $X = (x_1, x_2, ..., x_p)$ e $Y = (y_1, y_2, ...y_p)$, respectivamente. A métrica de Minkowsk é definida pela equação 2.4.4.

$$d(X,Y) = \left(\sum_{i=1}^{p} |x_i - y_i|^r \right)^{1/r} \tag{2.1}$$

Quando o valor de r for igual à 2, temos a distância Euclideana, quando for igual à 1 temos a distância de Manhatan (ou distância em quadras). Existem várias outras distâncias na recuperação de imagens, como a distância de Camberra, distância angular, coeficiente de Czekanowski, produto interno, coeficiente de dados, coeficiente cosseno e coeficiente de Jaccard. Algumas dessas distâncias podem ser encontradas em [Androutsos et al. 1998].

A similaridade global de duas imagens é mais difícil de ser medida. Há basicamente duas maneiras: comparação um-para-um (ou um-para-muitos) e comparação muitos-para-muitos.

Na comparação um-para-um cada região da imagem consulta é comparada com a região de melhor escolha feita na imagem a ser comparada. Logo a similaridade global é dada pela soma ponderada das distâncias entre cada região da imagem consulta e sua "melhor escolha" na imagem a ser comparada, onde o peso está relacionado com o tamanho da região.

Na comparação muitos-para-muitos todas as regiões da imagem consulta são comparadas com todas as regiões da imagem a ser comparada e vice e versa. Li et al [Edward 2002] propõem um esquema de comparação integrado de regiões (IRM) que permite comparar uma região de uma imagem com várias regiões de outra imagem, diminuindo assim o impacto de uma segmentação inexata. Cada comparação possui um valor (chamado valor significante), assim, uma matriz de significâncias de um conjunto de regiões é criada. A similaridade entre as duas imagens é medida por essa matriz.

2.5 Métodos de Avaliação

Não existe na literatura, um padrão comum para avaliação de um sistema de recuperação de imagens, mas existe uma proposta de uma série de medidas para a avaliação desses sistemas elaboradas em uma conferência chamada Text REtrieval Conference (TREC) [Deselaers et al. 2008].

Considere uma base de dados $\{i_0, ..., i_n, ...i_N\}$ onde i_n é uma imagem representada por seu vetor de características. Para recuperar imagens semelhantes a uma imagem consulta q, cada imagem i_n é comparada com a imagem consulta usando uma função distância $d(q, i_n)$ conforme citado no capítulo anterior. Assim, as imagens do banco de dados são ordenadas de acordo com essa função de modo que $d(q, i_{n_i}) \leq d(q, i_{n_{i+1}})$ resultando, para cada par de imagens i_{n_i} e $i_{n_{i+1}}$ na sequência $(i_{n_1}, ..., i_{n_i}, ...i_{n_N})$. Se uma diferente combinação de características é usada, as distâncias são normalizadas para estarem na mesma faixa de valores, então uma combinação linear das distâncias é usada para criar o *ranking*.

Várias formas de avaliação de sistemas CBIR´s têm sido propostas [Deselaers et al. 2008] baseados na precisão P e revocação R. Dada uma imagem consulta (q), a precisão é usualmente definida como a fração entre o número de imagens relevantes recuperadas em relação à q, e o número de imagens recuperadas para essa consulta, isto é:

$$P(q) = \frac{\text{Número de imagens relevantes recuperadas}}{\text{Número total de imagens recuperadas}}.$$

Por outro lado, a revocação é definida como a fração entre o número de imagens relevantes recuperadas em relação à q, e o número total de imagens relevantes no banco de dados para q, ou seja:

$$R(q) = \frac{\text{Número de imagens relevantes recuperadas}}{\text{Número total de imagens relevantes}}.$$

Quando o conjunto de imagens recuperadas é considerado até a posição $n_{A(q)}$ onde $n_{A(q)}$ é a quantidade de elementos do conjunto de imagens relevantes $A(q)$, a definição de revocação torna-se equivalente à definição de precisão conhecida como *R-Precision* [Baeza-Yates and Ribeiro-Neto Entretanto, a medida *R-Precision* é equivalente a medir a precisão até posição $n_{A(q)}$ do ranking, o que diz pouco sobre a performance global do sistema. Assim, convencionou-se a utilização da

curva precisão-revocação [Baeza-Yates and Ribeiro-Neto 1999] para a descrição e comparação de performance de sistemas de recuperação de informação. A maneira mais comum de resumir este gráfico em um único valor é a Média das Precisões Médias (*mean average precision* ou MAP).

Dada uma imagem consulta q a precisão média Ap é a média sobre os valores de precisão P_q após cada imagem relevante ser recuperada:

$$Ap(q) = \frac{1}{N_r} \sum_{n=1}^{N_r} P_q(R_n),$$

onde R_n é a revocação após a n-ésima imagem relevante recuperada e N_r é o número total de imagens relevantes à consulta q. Logo, a média das precisões MAP é a média dos valores de precisão média sobre todas as consultas:

$$MAP = \frac{1}{Q} \sum_{q \in Q} Ap(q),$$

onde Q é o conjunto de consultas q.

Uma vantagem da média das precisões é que a mesma contém aspectos tanto da precisão quanto da revocação e é sensível a todo o ranking.

Outro método utilizado é o erro de classificação ER onde é considerada apenas a imagem mais similar de acordo com a função distância aplicada [Deselaers et al. 2008]. Uma imagem consulta q só será considerada como classificada corretamente se a primeira imagem recuperada for relevante, ao contrário, a imagem não foi classificada corretamente. O cálculo de ER é dado pela seguinte equação.

$$ER = \frac{1}{Q} \sum_{q \in Q} \begin{cases} 0, & \text{se a imagem mais similar é relevante/pertence a classe correta;} \\ 1, & \text{caso contrário.} \end{cases}$$

Este método de avaliação é interessante quando o banco de dados está rotulado em classes, para bancos de dados de classes pré-definidas mas com imagens consultas selecionadas com suas correspondentes imagens relevantes, as classes podem ser definidas somente como *relevante* ou *irrelevante* [Deselaers et al. 2008].

Esta medida está diretamente relacionada com a precisão de um documento sendo usada como uma medida de performance adicional em muitas avaliações de sistemas de recuperação. A ER usada na avaliação dos resultados na seção 7.5.2 deste trabalho é igual a $1 - P(1)$, onde $P(1)$ é a precisão após o primeiro documento ser recuperado. Usando a ER um sistema de recuperação de imagens pode ser visto como um classificador do tipo "vizinho mais próximo" ou *nearest neighbor classifier* usando a mesma função distância e o mesmo sistema de recuperação [Deselaers et al. 2008].

2.6 Técnicas para redução do Gap-Semântico

A principal motivação para o estudo da Recuperação de Imagens Baseada em Semânticas é a redução do gap-semântico (ou hiato semântico) pois a principal questão é satisfazer a expectativa de um usuário em relação a um sistema de recuperação de imagens. Se considerarmos que conceitos semânticos como "um gato no muro", "uma sala cheia de pessoas" ou "meninas correndo", podem ser facilmente entendidos e procurados por uma criança, podemos ter a noção não só das expectativas dos usuários mas também da distância que separa o estado da arte da visão computacional e a capacidade do sistema perceptual humano.

Descritores obtidos através do processamento das características da imagem podem estar completamente dissociados da interpretação do usuário. Associar um significado à imagem é um dos grandes desafios da visão computacional. Logo, a pesquisa atual se concentra em diferentes métodos de associação de conceitos semânticos às características que podem ser computadas pelas técnicas atuais aplicadas aos pixels da imagem. Esta seção foca as técnicas usadas na derivação de semânticas de alto nível e são identificadas 3 categorias [Liu et al. 2007]:

- o uso de ontologia de objetos para definir conceitos de alto nível;
- a introdução do conceito do Retorno de Relevantes em um laço de recuperação para um aprendizado contínuo de intenções do usuário para suporte na recuperação de imagens em alto nível;
- e o uso de métodos de aprendizado supervisionado e não supervisionado para associar atributos de baixo nível com conceitos de consulta;

2.6.1 Ontologia de Objetos

Em alguns casos (objetos), a semântica de alto nível pode ser facilmente derivada de nossa linguagem diária, como por exemplo, a palavra "céu" pode ser caracterizada como uma região azul e uniforme que está na parte de cima de uma imagem. As ontologias podem descrever essas relações de maneira formalizada [Liu et al. 2007].

Ontologias são teorias sobre conteúdo, referentes a uma gama de objetos, suas propriedades e as relações que esses possuem entre si, os quais são possíveis dentro de um determinado domínio de conhecimento. Elas fornecem termos para descrevermos nosso conhecimento dentro de um domínio.

As imagens podem, então, ser classificadas em diferentes categorias pelo mapeamento de descritores de baixo nível (características) para semânticas de alto nível (palavras-chave) baseados em nosso conhecimento. Retomando ao exemplo "céu" podemos caracterizá-lo como uma região de cor azul clara (cor), uniforme (textura) e localizada na parte de cima da imagem (localização espacial) [Liu et al. 2007].

O uso de ontologias na recuperação de imagens ainda está na sua fase inicial, pois o mesmo funciona bem para imagens de cenas naturais de cores homogêneas mas ainda não estão bem definidos para imagens com textura além do mapeamento de outros atributos de baixo nível. Para banco de dados com coleções de imagens específicas, tais semânticas simples derivadas de ontologia de objetos podem funcionar bem. No entanto para grandes bancos de dados, com

grandes coleções de imagens e de vários conteúdos, ferramentas mais poderosas são requeridas para aprender as semânticas como, por exemplo, o aprendizado de máquina [Liu et al. 2007].

2.6.2 Retorno de Relevantes

A Recuperação de Imagens Baseada em Conteúdo também é, na maioria das aplicações tradicionais em visão computacional, usualmente automática e auto-alimentável. Para satisfazer exatamente o modelo de consulta que o usuário necessita a consulta de imagem pode se tornar um processo iterativo e interativo para a imagem ou as imagens desejadas.

O retorno de relevantes é um termo comum em recuperação baseada em texto para descrever uma forma de aprendizado supervisionado para ajustar as consultas subsequentes usando a informação fornecida pelo usuário [Salton and McGill 1986]. A essência do retorno de relevantes (ou *Relevance Feedback*) é a seguinte: depois de uma busca inicial por documentos da coleção, alguns documentos são recuperados e são apresentadas aos usuários certas informações descrevendo cada documento recuperado (por exemplo, título, resumos, tabelas de conteúdos ou uma combinação desses). O usuário então examina essas informações e identifica cada documento como sendo relevante ou irrelevante a sua consulta. Esse julgamento é retornado ao sistema de recuperação de informação para que se faça um ajuste na consulta inicial de tal modo que os termos da consulta presentes nos documentos relevantes sejam "promovidos" ou novos termos são adicionados à consulta e termos presentes nos documentos irrelevantes são penalizados. Outro processo de recuperação é então realizado e o processo é repetido até que o usuário esteja satisfeito ou até que nenhuma mudança adicional nos documentos recuperados seja observada [Elliott and Cashman 1973]. Na recuperação de imagens o princípio é o mesmo, a única diferença é o tipo de dado tratado, que neste caso serão as imagens fornecidas como *feedback* pelo sistema. Algumas medidas para melhorar ainda mais a performance desse tipo de sistema estão sendo estudadas como o uso de Algoritmos Genéticos aplicados ao retorno de relevantes [Silva et al. 2006].

2.6.3 Aprendizado de Máquina

Na maioria dos casos, para se derivar atributos semânticos de alto nível é necessário o uso de ferramentas formais como as técnicas de aprendizado supervisionado e não supervisionado de máquina. A meta do aprendizado supervisionado é prover uma medida de resultado (por exemplo, classe de categoria semântica) baseada em uma medida de entrada. No aprendizado não supervisionado, não há uma medida de resultado e a meta é descrever como os dados de entrada são organizados ou agrupados.

Aprendizado Supervisionado

Para tentar reduzir o gap-semântico existente entre o poder limitado de interpretação semântica pelas máquinas e a rica subjetividade do pensamento humano, alguns trabalhos baseados em semânticas de alto nível têm sido propostos através do uso de ferramentas formais como as técnicas de aprendizado supervisionado [Sethi et al. 2001].

O objetivo do aprendizado supervisionado é fornecer um resultado (por exemplo, uma categoria semântica à qual a consulta pertence) baseado em uma série de medidas de entrada. Algumas técnicas como o uso do *Support Vector Machine* (SVM) [Shi et al. 2004] podem ser utilizadas para aprender conceitos de alto nível (como categorias semânticas) através de características de baixo nível (como atributos de cor, forma e textura) [Liu et al. 2007].

Com teoria e aplicações bem fundamentadas, o SVM tem sido muito utilizado para reconhecimento de objetos, classificação de textos, etc. e é considerado um bom candidato para o aprendizado em sistemas de recuperação de imagens [Tong and Chang 2001]. O SVM foi originalmente desenvolvido para classificação binária e através dele pode-se encontrar um espaço n-dimensional (entre características de baixo nível, por exemplo) onde um hiperplano separador é construído. Dentre os possíveis hiperplanos (como mostrado na Figura 2.6), o plano separador ótimo (OSP) (mostrado na Figura 2.7) irá maximizar a distância entre o hiperplano e os pontos mais próximos de cada classe utilizando, assim, esse separador para classificar as imagens [Shi et al. 2004].

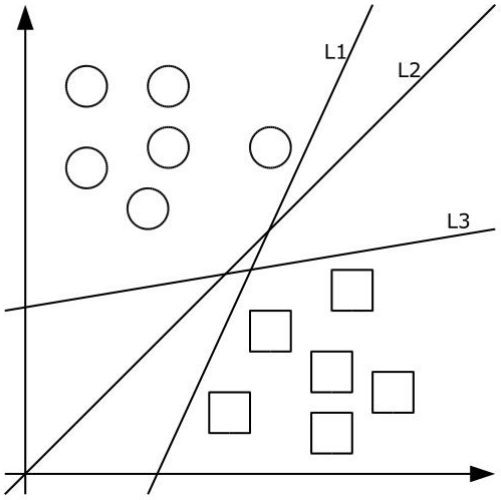


Figura 2.6: Existência de diversos separadores para fazer a separação de duas classes.

As Redes Neurais, em geral, também podem ser utilizadas para fazer a classificação de imagens. Nesse caso um grande número de dados de treinamento (características de baixo nível) é inserido na rede neural para estabelecer o vínculo entre os atributos de baixo nível das imagens e suas semânticas de alto nível (categorias) [Town and Sinclair 2000]. A classificação se dá na constituição de fronteiras de decisão não lineares no espaço de características de baixo nível, adequando a imagem consulta ao grupo em que ela mais se aproxima. Em [Gonzalez et al. 2006], Gonzales et. al descrevem a combinação de Redes Neurais e *Wavellets* para a recuperação de imagens em termos de seu conteúdo.

No artigo [Zhang and Izquierdo 2007], Zhang propõe uma forma de recuperação de imagens em alto nível, através da divisão das imagens de interesse em blocos de regiões uniformes e extraindo características de baixo nível (através de sete descritores: CLD, CSD, DCD, EHD, TGF, GLC e HSV) para montar vetores de características ponderados. A pesagem de cada termo é dada pelo cálculo da matriz de distâncias do vetor para o centróide do conjunto de

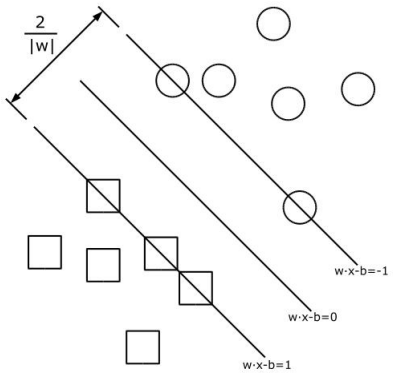


Figura 2.7: Hiperplano ótimo e amostras de treinamento.

vetores que caracterizam um conceito pré-escolhido. A recuperação é, então, feita através desses vetores ponderados.

Em [Ma and Wang 2005] também é proposto um sistema de recuperação de imagens através das redes neurais, onde cada imagem é caracterizada em baixo nível (através da Transformada Discreta do Co-seno) e associada à um ou mais conceitos presentes nesta. O vetores de características e seus respectivos rótulos são reunidos em um conjunto para treinar a rede neural e estabelecer o vínculo entre as características e os conceitos pré-definidos. Os resultados mostram que as redes neurais podem ser capazes de memorizar tais categorias mas, para imagens compostas por vários conceitos semânticos os resultados não são satisfatórios.

Algoritmos de aprendizado convencionais sofrem dois problemas: uma grande quantidade de amostras classificadas são necessárias para o treinamento e prover tais dados pode estar propenso à erros. Outro fator é que o conjunto de treinamento é fixado durante o aprendizado e o estágio de aplicação, consequentemente se o domínio de aplicação muda, novas amostras classificadas devem ser providas para assegurar a efetividade do classificador. Ou seja os mesmos requerem um grande número de dados de treinamento sendo computacionalmente extensivo além de retornar apenas imagens pertencentes a categorias pré-estabelecidas.

O modelo proposto por esse trabalho usa o mesmo princípio apresentado nessa seção mas emprega a rede neural em seu estágio final (após seu treinamento) para fazer um mapeamento não linear das características e montar um vetor de características genérico baseado nas semânticas intrínsecas de cada imagem, ou seja, o sistema irá fazer uma busca em imagens não utilizadas na fase de treinamento baseado no poder de generalização da rede. Esses aspectos semânticos dos vetores de neurosemânticos são adquiridos durante o treinamento da rede que generaliza e infere esses conceitos no banco de dados de treinamento e os usa então na transformação das características de baixo nível em características neurosemânticos.

Aprendizado Não Supervisionado

O aprendizado não supervisionado não possui medidas de saída. A principal tarefa é encontrar uma forma de organizar os dados de entrada ou clusterizá-los. A clusterização de imagens é uma técnica típica do reconhecimento de padrões e vem sendo amplamente abordada na recu-

peração de imagens e o objetivo final é agrupar um conjunto de imagens de forma a maximizar a similaridade dentro dos *clusters* e minimizar a similaridade entre *clusters* diferentes. Cada *cluster* resultante é associado a uma classe e as imagens de mesmo *cluster* serão supostamente similares umas às outras.

O algoritmo K-*means* (ou k-médias) pode ser usado para classificar as amostras baseando-se nos atributos de baixo nível. Desse modo, medidas estatísticas de variação em cada *cluster* são usadas para derivar uma série de mapeamentos entre os atributos de baixo-nível e a caracterização textual ótima (palavras-chave) de cada *cluster* correspondente. As regras de mapeamento derivadas podem ser usadas para indexar novas imagens adicionadas posteriormente ao banco. Nesse caso, a recuperação também ficará presa ao número de *clusters* escolhidos para realizar o aprendizado, tornando a recuperação de imagens limitada.

Capítulo 3

Caracterização de Imagens

3.1 Introdução

Sistemas de Recuperação de Imagens Baseada em Conteúdo (CBIR´s) se utilizam da hipótese de correspondência de uma dada imagem a outra a partir de seus atributos. Nesses sistemas as imagens são indexadas por um conjunto numérico de dados relevantes. Essa estrutura de dados também chamada de vetor de características abstrai a informação necessária para a representação destes atributos.

O número de características extraídas dependerá do tamanho do domínio da imagem, dos atributos que se deseja representar e dos algoritmos de extração escolhidos para caracterizar a imagem. Essa quantidade dará a dimensão do vetor que poderá ser armazenado em alguma tabela de Banco de Dados para facilitar e agilizar as próximas etapas da recuperação de imagens. Algumas considerações a respeito do vetor de características são feitas em [Loew 2000] que diz:

> *"Um vetor de características deve ser formado baseando-se na redução da dimensionalidade dos dados, bem como ressaltar os aspectos principais da informação visual para facilitar a percepção humana além de ser invariante às transformações das imagens."*

O tempo de processamento para a Recuperação de Imagens baseada em Conteúdo também pode ser considerado, pois há um gasto muito maior de se trabalhar com toda a informação da imagem do que com um vetor de dimensão reduzida. Por exemplo, em uma imagem de dimensões 256 x 256 teria-se que acessar 65536 pixels para colher toda a informação pertinente a essa imagem cada vez que um sistema CBIR fosse iniciado. Logo, a codificação e abstração de novos valores a partir de uma imagem original se tornam imprescindíveis para a caracterização de uma grande quantidade de informação.

Como foi citado no capítulo anterior, devido à percepção particular pertinente a cada ser humano não há uma única ou melhor forma para se representar estes atributos. As pessoas identificam uma imagem através de seus padrões morfológicos interpretando visualmente essa imagem, determinando e caracterizando os elementos e os associando a algum padrão pré-definido pelos seus cérebros. Por exemplo, na Figura 3.1 , diferentes pessoas poderiam estar

interessadas em diferentes aspectos como a cor, a textura, forma ou os objetos pertencentes à mesma.

Figura 3.1: Doll and Stuffed Animals, tela de Norman J. LaPrise ©

Verificando cada representação dada pela Figura 3.2, pode-se notar a evidência do exame intrínseco e pessoal da informação visual, a qual é ponderada e influenciada fortemente pelo exame visual subjetivo da imagem, pela sensação e pelas idéias dadas ao observador, através de seu poder de discernimento, seu entendimento analítico e característico. Se analisarmos em separado cada uma das imagens mostradas, podemos ter sensações, interesses e perspectivas diferentes.

Figura 3.2: Modelos de Representação dos atributos: Cor, Forma e Textura, respectivamente

Existem várias representações que caracterizam esses atributos sob diferentes perspectivas. A maioria das pesquisas atuais baseadas em sistemas CBIR focam na descrição matemática de aspectos da visão humana tais como cor, forma e textura. Algumas pesquisas focam na descrição de aspectos semânticos como objetos, ações, sentimentos mas, por enquanto, tem se mostrado uma tarefa bastante complexa e com poucos resultados práticos [Rui et al. 1999]. No trabalho de [Liu et al. 2007] é apresentada uma revisão do estado arte da exploração de características de alto nível (semânticas). À descrição matemática de características de baixo nível dá-se o nome de vetores de características. Um vetor de característica tem por finalidade representar os aspectos visuais significativos de uma imagem. Nesse capítulo serão apresentados

os atributos principais utilizados em sistemas CBIR bem como alguns métodos clássicos e conjunto de medições para a obtenção de seus respectivos valores populando assim, o vetor de características.

3.2 Atributos das Imagens

As características de uma informação visual podem incluir tanto atributos textuais (palavras, anotações, descrições, etc.) como características visuais (cor, textura, formas, localização espacial, etc.). O modelo proposto em [Rui et al. 1997] sugere a utilização de três bancos de dados para recuperação de imagens: a coleção de imagens, os atributos visuais e os atributos descritivos conforme mostrado na Figura 3.3. A coleção de imagens possuirá todas as imagens a serem utilizadas pelo sistema, principalmente para a visualização das figuras, e servirá também de base para as outras duas. O banco de atributos descritivos possui as palavras-chave e, geralmente, uma descrição textual livre de cada imagem. Esse banco de dados é utilizado em Sistemas de Recuperação de Imagens Baseada em Texto e sua indexação se dá por processos de Recuperação de Informação clássicos, portanto não será focado neste trabalho. O banco de atributos visuais conterá todas as características visuais extraídas de cada imagem. Essa informação será primordial para o funcionamento do Sistema de Recuperação de Imagens Baseada em Conteúdo. Ferramentas e técnicas de visão computacional são utilizadas para a obtenção desses dados.

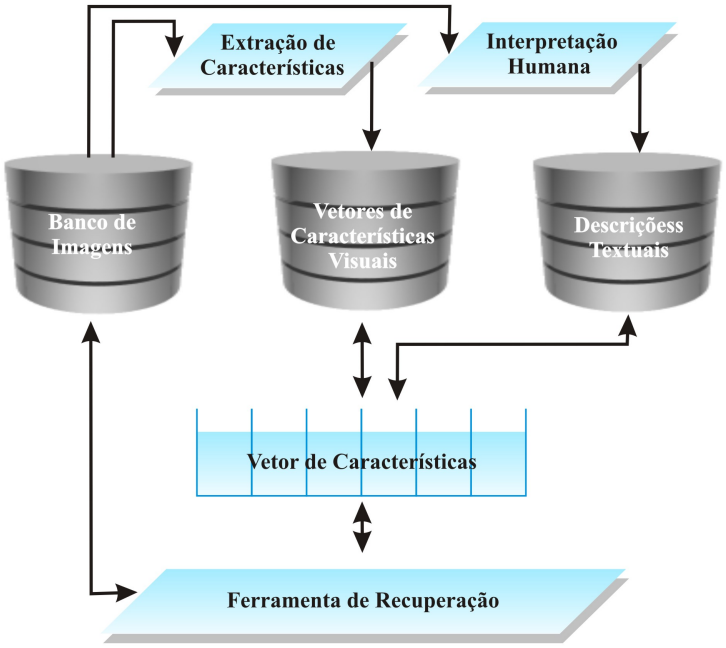


Figura 3.3: Arquitetura de um sistema de caracterização de imagens. Adaptado de [Rui et al. 1997]

Os atributos principais utilizados em sistemas CBIR são: cor, forma e textura. Estas características também chamadas de características de baixo nível estão intimamente interligadas com o conteúdo semântico dos dados envolvidos tanto que, na literatura de recuperação de informação, esses atributos são chamados também de "evidências semânticas" ou simplesmente "evidências" informações pelas quais são utilizadas na diferenciação de duas imagens. Definir quais dessas características devem ser incluídas no vetor e em que proporção elas serão utilizadas para a representação da informação visual ainda gera muita discussão. A cor por exemplo, é um atributo simples e considerada, por muitos autores, confiável para a representação de uma imagem. No entanto, em algumas situações a cor não constata a semelhança entre duas imagens que seriam subjetivamente parecidas. Para que isso não ocorra pode-se fazer uma junção entre os atributos e utilizá-los de maneira ponderada. A partir de uma caracterização adequada de uma imagem e da escolha de uma ferramenta adequada de recuperação é possível identificar alguns padrões principais na mesma ou realizar descrições sobre o conteúdo dela. Este trabalho propõe, através das redes neurais, uma alternativa para a escolha da proporção mais adequada para cada atributo.

Um vetor de características pode ser global ou local. Um vetor global usa as características visuais da imagem inteira, já um vetor local usa características de regiões ou objetos para descrever o conteúdo da imagem. Para se obter descrições (vetores) locais, uma imagem deve ser dividida em partes. A figura 3.4 apresenta um exemplo, onde a imagem é subdivida em partes onde cada objeto representa uma região. Em um sistema de recuperação baseado em regiões cada parte estaria ligada a um vetor de características independente. O caminho mais simples de dividir uma imagem é usar partições seguindo algum critério pré-definido, como por exemplo, blocos de mesmo tamanho. Um caminho mais complexo é empregar técnicas de segmentação[1] para se obter objetos mais significantes semanticamente (como pessoas, bola de futebol, carro, casa, etc.). Entretanto, a segmentação de objetos não é discutida nesse trabalho, visto que a segmentação de uma imagem em objetos semanticamente significativos é tão desafiante quanto o próprio problema do *gap-semântico*.

Figura 3.4: Exemplo de segmentação de objetos de uma imagem.

[1] Atualmente, segmentação automática de objetos para imagens de domínios gerais é uma tarefa de improvável sucesso.

Geralmente a primeira fase da extração de características para cada atributo é precedida de uma segmentação ou um pré-processamento. Para minimizar os erros e defeitos resultantes da baixa qualidade de uma imagem na extração de características, por exemplo, pode-se fazer uso de filtros ou equalizações de histograma. Alguns problemas comuns relacionados com a qualidade e a dificuldade de extrair padrões invariantes são os ruídos, distorções, translações, rotações, mudança de escala, etc. Todos esses problemas possuem alternativas para serem corrigidos. O bom funcionamento dos algoritmos de extração bem como a maximização da eficiência dos mesmos se torna um fator preponderante para a descrição invariante da imagem e, consequentemente, para o resultado da recuperação de imagens. Um exemplo disso é a detecção de bordas para a extração de formas: um algoritmo que detecta todas as bordas de uma imagem trará um maior benefício para a extração das características de forma de um dado objeto. Algumas dessas técnicas serão citadas ao longo deste capítulo.

3.3 Características de Imagens Baseadas em Aparência

A aproximação mais direta para se caracterizar uma imagem é usar os valores dos próprios *pixels* das imagens como características: as imagens são reduzidas a um tamanho comum e comparadas utilizando uma função distância. No artigo [Deselaers et al. 2008] é usada uma redução para uma janela 32×32. Foi observado que, para classificação e recuperação de imagens radiográficas este método serve como uma base razoável. Em [Keysers and Gollan 2007] diferentes métodos foram utilizados para comparar imagens diretamente atentando para deformações locais.

3.4 Cor

A cor é um dos atributos mais usados para a recuperação de imagens baseada em conteúdo por ser relativamente simples e ser invariante tanto ao tamanho da imagem quanto sua orientação, além disso o atributo cromático representa uma função significativa na representação das características agregadas aos objetos visualizados. Por exemplo, no estudo de solos, a cor diferencia quais as características químicas dos mesmos, como a terra roxa que está presente nos solos avermelhados, derivados principalmente de rochas básicas: basalto e diabásio [Gonzales and Woods 1992] [Rui et al. 1997].

A cor é a propriedade dos objetos de refletirem radiação eletromagnética com frequência e comprimento de onda característicos. A faixa visível do olho humano desse espectro eletromagnético varia aproximadamente de 400 nm a 700 nm. A figura 3.5 apresenta a denominação usual dada às diversas regiões do espectro eletromagnético.

As cores visíveis são o resultado da incidência de luz com diferentes comprimentos de ondas. A cor magenta, por exemplo, é uma das combinações de cor mais puras e resulta do aditamento de duas cores do espectro: o azul e o vermelho. Para fazer essa adição basta combinar a luz derivada de dois ou mais focos onde cada um deles lança de si a luz de cada uma daquelas cores, ou seja, a faixa visível resultante é definida pela soma das cores espectrais emitidas gerando um processo também chamado de processo aditivo [Foley et al. 1990].

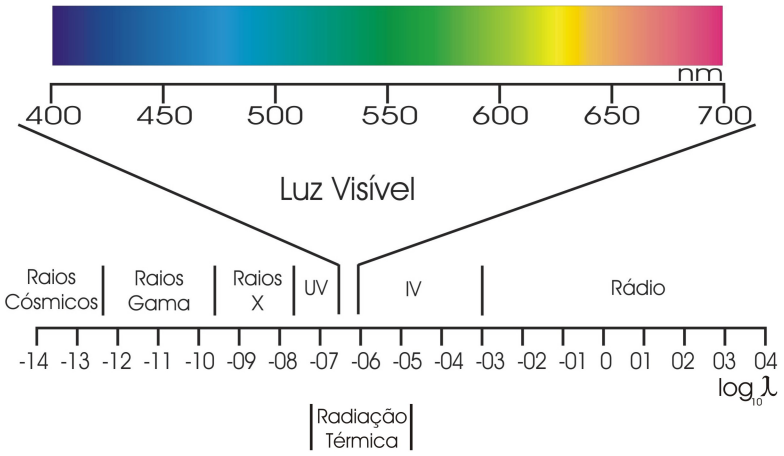


Figura 3.5: Espectro electromagnético representando as bandas de comprimento de onda principais e a banda correspondente à luz visível.

A luz quando emitida em um comprimento de onda preciso produz uma cor pura do espectro visível. Na verdade, com exceção de fontes de luz como os lasers, as fontes de luz emitem-na com uma dada energia em vários comprimentos de onda numa gama localizada à volta de um comprimento de onda dominante. Se a potência dada pela fonte for pequena, a cor passará a ser vista como um sombreado mais ou menos escuro, se a potência for elevada, a cor será percebida como uma cor viva e brilhante [Foley et al. 1990].

Existem vários sistemas e modelos para representar a combinação dessas faixas de luz. A maioria dos espaços de cores é definida em três dimensões, de modo que, cada cor é representada por três coordenadas. Além disso, os modelos de cores são classificados como uniforme/não-uniforme dependendo da diferença sobre o espaço de cores, de acordo com a percepção do observador [Colombo et al. 1999]. Existem, além dos modelos mais conhecidos como o RGB e o HSV, outros espaços de cores criados para fins mais específicos dentre esses pode-se citar os modelos *Mussel*, *CMY*, HLS, YIK e YcbCr , *CIE L*a*b*, *CIE L*u*v* e *HSI*. Todos esses modelos podem ser facilmente derivados dos dois modelos citados anteriormente. Abaixo são apresentados os dois modelos de cores utilizados neste trabalho e como é realizada a conversão de um modelo para o outro.

O Modelo de Cores RGB

O modelo de cores RGB é o mais comum de todos os modelos existentes e é composto basicamente de três cores também chamadas primárias: vermelho, verde e azul. A sigla RGB representa as iniciais dos nomes em inglês destas cores: *Red*, *Green* e *Blue*.

As cores descritas por esse modelo se dão pela adição destas três cores primárias, cada uma delas variando entre 0 e 1. O valor 1 corresponde à intensidade máxima em que a cor é representada pelo dispositivo gráfico e o valor 0 a intensidade mínima ou a não emissão de luz em alguns casos. A cor branca será o resultado da soma de todas as três cores primárias em seu valor máximo, e a cor preta será o resultado da ausência de cor [Gonzales and Woods 1992].

O espaço RGB de cor é construído através dos três parâmetros deste modelo e das intensi-

dades das três cores primárias deste, criando assim, um espaço tridimensional com três direções ortogonais (R, G e B). As cores deste espaço existem em um sub-espaço onde $0 \leq R \leq 1$, $0 \leq G \leq 1$ e $0 \leq B \leq 1$. Este sub espaço caracteriza um cubo de aresta unitária em que o vértice de coordenadas (0,0,0) representa a cor preta na qual coincide com a origem do espaço e o vértice representando a cor branca correspondente ao ponto de coordenadas (1,1,1), como na figura 3.6. A Figura 3.7 mostra uma imagem colorida e a visualização de seus canais de cor R, G e B.

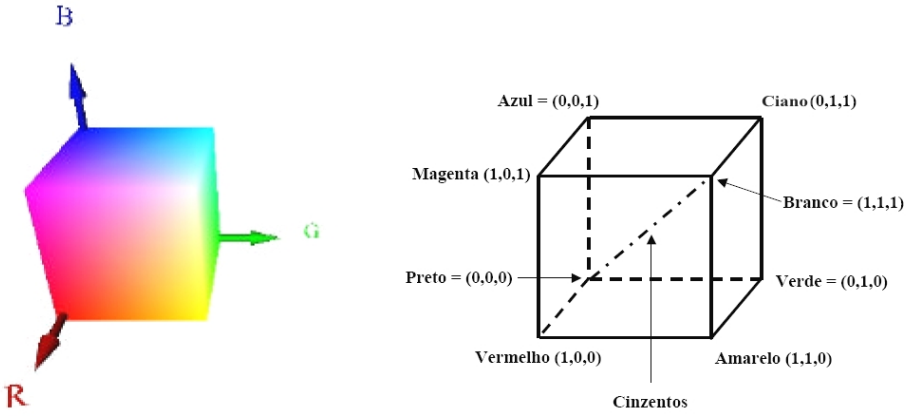


Figura 3.6: Espaço de Cores RGB.

Figura 3.7: (a) Imagem original; (b), (c) e (d) visualização dos canais R, G e B, respectivamente.

Habitualmente, em sistemas gráficos, ao invés de se exprimir os valores reais entre 0 e 1, é comum usar valores inteiros entre 0 e 255 para representar o grau de valor de cada componente de cor. Esse padrão foi adotado para facilitar o processamento nos primeiros dispositivos gráficos, pois a manipulação de dados inteiros de oito bits em relação a dados reais fracionários se torna mais simples e rápida. Para o olho humano, a discretização em 256 valores de intensidade é bastante para a distinção de cores, visto que este consegue distinguir entre um número mínimo de 16 intensidades na área do azul e um máximo de 23 intensidades na área do amarelo [Gonzales and Woods 1992].

O Modelo de Cores HSI

A sigla HSI representa as iniciais dos nomes em inglês das três variáveis do modelo: *Hue* que é componente de escolha da "tinta" em vigor, sendo dominada pela disposição angular de um ponteiro em um círculo de cores definida de 0 a 359, *saturation* que é a componente que determina a pureza da cor selecionada em *Hue* e *Intensity* que é desacoplada das informações de cor, correspondendo a uma representação da imagem em níveis de cinza.

A Figura 3.8 ilustra o espaço de cor HSI. A Figura 3.9 mostra uma imagem colorida e a visualização de seus canais de cor H, S e I.

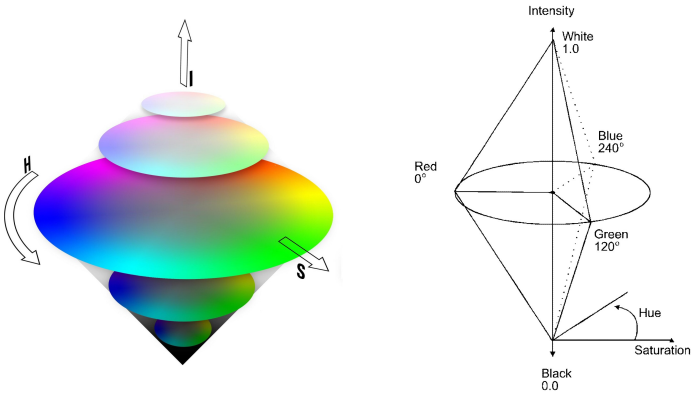


Figura 3.8: Espaço de Cores HSI.

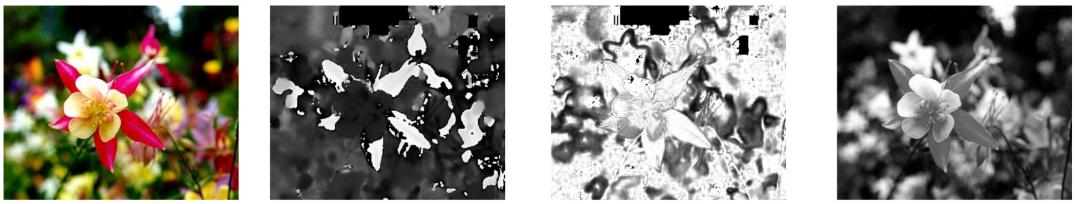


Figura 3.9: (a) Imagem original; (b), (c) e (d) visualização dos canais H, S e I, respectivamente.

Conversão RGB/HSI

As expressões 3.1, 3.2 e 3.3 são utilizadas para a conversão do sistema de cor RGB para o HSI. Elas obtêm valores para as componentes H, S e I no intervalo $[0, 1]$ a partir dos valores de R, G e B no mesmo intervalo.

$$I = \frac{1}{3}(R + G + B) \tag{3.1}$$

$$S = 1 - \frac{3}{(R+G+B)}\ min(R,G,B) \tag{3.2}$$

$$H = cos^{-1}\left[\frac{\frac{1}{2}\big[(R-G)+(R-B)\big]}{\sqrt{\big[(R-G)^2+(R-B)(G-B)\big]}}\right] \tag{3.3}$$

onde $H = 2\pi - H$ se $\frac{B}{I} > \frac{G}{I}$. Para normalizar H no intervalo $[0,1]$, calcula-se $H = \frac{H}{2\pi}$.

Não há, na literatura, um padrão para o uso de espaço de cores, sendo que a escolha irá depender da aplicação e do descritor utilizado. No entanto, na descrição das cores de uma imagem, umas das propriedades mais relevantes de um espaço de cor é a uniformidade, ou seja, a recuperação de imagens baseada na semelhança das cores requer distâncias no espaço de cores pertinentes à percepção humana. A indexação de imagens por cor foi introduzida por [Swain and Ballard 1991]. Ao longo do tempo, várias outras técnicas foram propostas para avaliar essa semelhança. A seguir são apresentados alguns descritores de cor comumente usados para descrever as informações de cores de imagens.

3.4.1 Momentos de Cores

Uma técnica bastante utilizada para caracterizar imagens em termos da distribuição das cores e que têm sido utilizados com sucesso em muitos sistemas CBIR, como o *QBIC* [Flickner et al. 1995 da *IBM* são os momentos de cor [Stricker and Orengo 1995]. As imagens são representadas no espaço de cor HSI (*Hue, Saturation and Intensity*) e cada canal de cor é interpretado por três medidas estatísticas.

Seja N o número de pixels da imagem e p_{ij} o valor do i-ésimo canal de cor para o j-ésimo pixel da imagem, a média, o desvio padrão e a inclinação são computados através das expressões 3.4, 3.5 e 3.6, respectivamente.

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \tag{3.4}$$

$$\sigma_i = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^2\right)} \tag{3.5}$$

$$S_i = \sqrt[3]{\left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^3\right)} \tag{3.6}$$

Estes três momentos são computados para cada componente de cor, totalizando nove momentos. A similaridade baseada nos momentos de cor é medida através da distância euclidiana. Em [Stricker and Orengo 1995] é afirmado que se duas imagens são similares, seus momentos de cores também o serão. Porém, se duas imagens têm apenas sub-regiões similares, os momentos das cores serão diferentes e a similaridade entre elas diminui.

3.4.2 Histograma de Cores

Como foi dito anteriormente, o resultado da diferença entre espaços de cores pode ser avaliado como uma distância entre os pontos de cores correspondentes. A maneira mais comum de se criar esse espaço de características é o uso de histogramas de cores.

O histograma de cores de uma imagem pode ser armazenado computacionalmente em uma estrutura de dados que armazena a quantidade de ocorrências de cada valor de cores na imagem dada. A dimensão dessa estrutura será correspondente à circunstancialidade de cores no espaço cromático usado para descrever a imagem. O histograma de cores é constituído por pilhas também chamadas de *bins*, uma para cada intensidade de cor do espaço quantizado, que clusterizam o número de vezes em que ocorrem os diferentes valores de luminância apresentados pelos pixels da imagem. Depois do histograma ser formado, ele pode ainda ser normalizado, dividindo-se o valor encontrado em cada pilha pelo número total de pixels contidos na imagem. A formação do descritor de cor baseado em histograma pode ser reduzido com relação ao espaço de armazenamento através da quantização de cores presentes no histograma e também através da formação de histogramas métricos o qual transforma os *bins* em *buckets*. Os *buckets* são normalizações obtidas das curvas de aproximações entre máximos e mínimos locais da função que os representa [Bueno 2001].

Dentre os problemas dos histogramas de cores podem-se destacar sua alta dimensionalidade (necessária para sua eficiência) e o fato destes não considerarem a localização espacial das cores. Assim, imagens muito diferentes podem ter representações semelhantes. Novamente pode ser adotada a divisão das imagens em regiões, computando uma representação para cada região. Porém, tal procedimento aumenta ainda mais a dimensionalidade e carga computacional necessária para comparar os descritores. Uma forma de se comparar esses descritores é calcular a distância euclidiana entre os dois histogramas ou utilizar a função de divergência Jensen-Shannon (JSD), dada pela equação 3.7.

$$d_{JSD}(H, H') = \sum_{m=1}^{M} H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H'_m + H_m} \tag{3.7}$$

Onde H e H' são os histogramas a serem comparados e H_m é o m-ésimo *bin* ou *bucket* do histograma H.

3.4.3 Vetor de Coerência de Cores

O vetor de coerência de cores (CCV) foi proposto por [Pass and Zabih 1996] e é uma forma de refinamento do histograma de cores, onde os *bins* do histograma são particionados baseando-

se na sua ocorrência. Isso é explicado pelo fato de existir uma variável oculta que indica qual a coerência espacial dos pixels.

De acordo com [Pass and Zabih 1996] o cálculo do vetor de coerência de uma imagem i é dado pelos seguintes passos:

- A imagem é suavizada sutilmente substituindo o valor do *pixel* pela média de seus vizinhos, utilizando uma vizinhança de oito.
- Cada pixel é rotulado como coerente e incoerente.
  Um *pixel* será coerente se ele fizer parte de uma região de coloração uniforme (de tamanho pré-definido k), caso contrário ele será incoerente.
- Para cada cor c_k o número de *pixels* coerentes α_{c_k}, e o número de *pixels* incoerentes β_{c_k} são computados.
- Cada componente do CCV é um par $(\alpha_{c_k}, \beta_{c_k})$, chamado par de coerência.
- O vetor de coerência é então definido como

$$CCV(i) = \Big((\alpha_{c_1}, \beta_{c_1}), (\alpha_{c_2}, \beta_{c_2}), \ldots, (\alpha_{c_n}, \beta_{c_n})\Big) \tag{3.8}$$

Pode-se notar que $(\alpha_{c_1} + \beta_{c_1}, \alpha_{c_2} + \beta_{c_2}, \ldots, \alpha_{c_n} + \beta_{c_n})$ é o próprio histograma de cor da imagem.

De acordo com [Pass and Zabih 1996], os vetores de coerência de cores produzem melhores resultados de recuperação que os histogramas de cores, especialmente para imagens com muitas regiões de cor uniforme ou muita textura principalmente quando aplicados ao espaço de cor *HSV*.

3.4.4 Correlograma de Cores

Um outro método para melhorar a performance de descrição do histograma de cores é o método de correlograma de cores proposto por [Huang et al. 1997]. O correlograma de cores é um histograma tridimensional que caracteriza a distribuição das cores e a correlação espacial entre pares de cores, o que facilita na descrição da distribuição global com a correlação local das cores. A primeira e a segunda dimensão do histograma representam as cores de qualquer par de *pixels* e a terceira dimensão, a distância espacial entre eles. Um correlograma pode ser visto como uma tabela indexada por pares de cores, onde a k-ésima entrada para (i, j) especifica a probabilidade de encontrar um *pixel* de cor j a uma distância k de um *pixel* de cor i na imagem. Seja H o conjunto de *pixels* de uma imagem e $H_{c(j)}$ o conjunto de *pixels* cuja cor é $c(j)$. Então seu correlograma é definido como:

$$\gamma_{i,j}^{k} = \Pr_{p_1 \in H_{c(i)}, p_2 \in H}\Big[p_2 \in H_{c(j)}, |p_1 - p_2| = k\Big] \tag{3.9}$$

onde $i, j \in \{1, 2, \ldots, N\}$, $k \in \{1, 2, \ldots, d\}$, e $|p_1 - p_2|$ é a distância entre os *pixels* p_1 e p_2, e Pr é a função probabilidade. Se considerarmos todas as possíveis combinações de pares de cores o tamanho do correlograma será muito grande ($O(N^2d)$), assim, uma versão simplificada chamada autocorrelograma é frequentemente usada. O autocorrelograma de cores captura somente a correlação espacial entre cores idênticas e assim reduz a dimensão para $O(Nd)$.

Comparado ao histograma de cores e ao vetor de coerência de cores, o autocorrelograma de cores produz melhores resultados, mas também é computacionalmente o mais caro devido à sua alta dimensionalidade [Huang et al. 1997].

3.4.5 Histogramas de Características Invariantes

Uma característica é dita invariante se respeitar a certas regras de transformações (quando essas transformações são aplicadas à imagem) [Siggelkow 2002]. A cor, por exemplo, não somente reflete o material da superfície, mas também varia consideravelmente com mudanças de iluminação, orientação da superfície, e visão geométrica das câmeras. Em alguns casos é interessante levar em conta esta variabilidade pois podem levar a alguma perda do poder de discriminação entre imagens. Tais invariâncias podem ser aplicadas em histogramas de características por integração, ou seja, uma certa função de descrição é integrada sobre o conjunto de todas as transformações consideradas. Representações invariantes de cores foram introduzidas na recuperação de imagens por conteúdo recentemente.

3.5 Forma

Os seres humanos tendem a perceber cenários como sendo compostos por objetos individuais, que podem ser melhor identificados por suas formas. Este atributo é utilizado por vários sistemas de recuperação de imagens pois, apesar de ser um dos aspectos mais difíceis de se caracterizar, o mesmo é considerado por muitos autores um dos melhores atributos para se representar e identificar um objeto. A contextualização da forma pode se tornar bastante controversa. Alguns autores definem a forma como um denominador comum entre objetos tridimensionais iguais ou semelhantes vistos sob diferentes aspectos [Lowe 1987].

Sob o ponto de vista da fenomenologia (ciência que estuda os fenômenos do cérebro humano), a forma significa um subconjunto de uma imagem, seja digital ou perceptual, dotada de algumas qualidades que permitem seu reconhecimento. Se a forma está intimamente relacionada com o reconhecimento de um objeto, a recuperação de imagens utilizando extratores de forma se torna plausível e indicada pra fazê-la [Attneave 1954].

Em imagens bidimensionais, a representação da forma pode se dar de várias maneiras. Usualmente duas noções de similaridade são empregadas. Observando a figura 3.10, pode-se perceber que as figuras situadas na primeira linha possuem uma distribuição de pixels semelhante, sendo semelhantes pela convenção de representação baseada em região. No entanto, se for observado o contorno desses objetos, pode-se verificar certa diferença entre os itens da primeira linha, e uma maior semelhança entre os objetos de cada coluna. Se houvesse uma consulta a partir de uma imagem exemplo, como a encontrada na primeira linha e primeira coluna, as imagens consideradas parecidas de acordo com critérios de similaridade baseada em

região seriam as da primeira linha. Se os critérios de similaridade fossem baseados em contorno as imagens mais semelhantes a essa seriam as da primeira coluna [Brandt 1999].

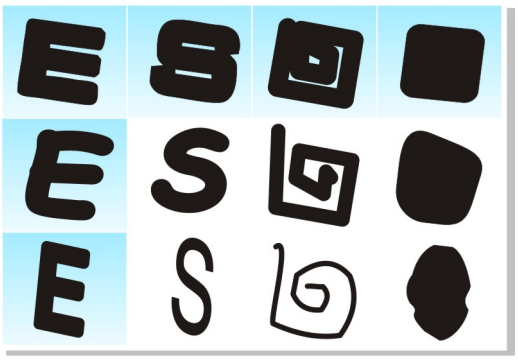


Figura 3.10: Exemplos de similaridade de forma baseada em contorno e região

Portanto, as técnicas de representação de forma podem ser divididas em descritores internos (baseada em regiões) e externos (baseadas em contornos), essas técnicas poderiam ainda ser distinguidas entre métodos no domínio do espaço e métodos no domínio de transformadas. Nas técnicas baseadas em regiões, todos os pixels dentro dos limites de uma forma são examinados para obter sua representação. A figura 3.11 apresenta a taxonomia dos métodos de descrição da forma com alguns exemplos de descritores que serão apresentados ao longo desta seção.

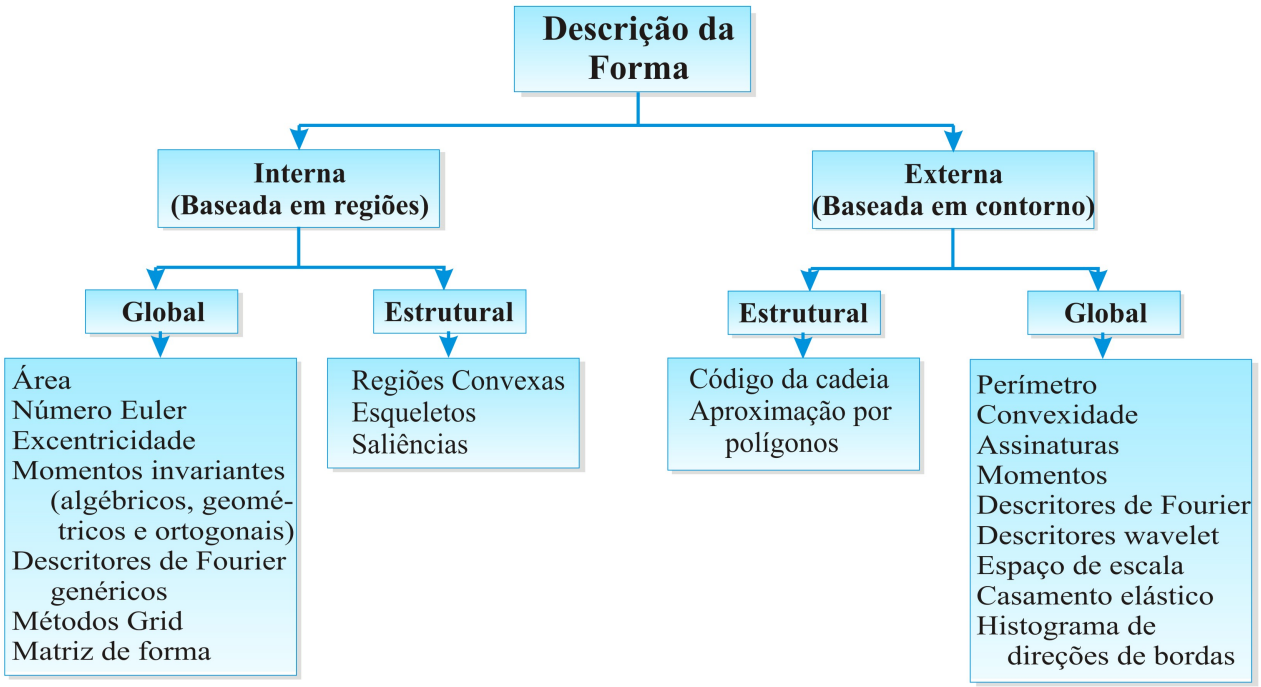


Figura 3.11: Classificação geral das técnicas de representação de formas.

3.5.1 Técnicas Baseadas em Regiões

Como foi dito na seção anterior, nas técnicas de extração de características de forma baseadas em regiões, todos os *pixels* dentro de uma região da imagem são levados em conta na representação de formas, ao invés de usar somente as informações da borda da imagem. Assim como os métodos baseados em contorno, os métodos baseados em regiões podem ser divididos em métodos globais e métodos estruturais.

Métodos Globais

Os métodos globais tratam a forma como um todo. Tais métodos medem a distribuição dos *pixels* de regiões, sendo menos prováveis de serem afetados por ruídos ou variações. Existem vários atributos escalares que podem ser derivados do interior de um objeto. Esses escalares podem ser combinados e um vetor de características pode ser criado. Alguns desses descritores são apresentados a seguir.

Momentos Invariantes As representações de momentos de regiões interpretam uma função normalizada de uma dada imagem em níveis de cinza como uma densidade da probabilidade de um valor aleatório. Dado que as representações de momentos combinam informação dentro de um objeto inteiro ao invés de um único ponto da borda, elas capturam algumas das propriedades globais que faltam em muitas representações puras baseadas em contorno: rotação, translação e escala. Objetos parecidos mas com dimensões diferentes, por exemplo, devem ser analisados como semelhantes independente de seu tamanho. Os primeiros sete termos dos momentos invariantes de Hu, derivados dos momentos centrais normalizados de segunda e terceira ordem, capturam bem as propriedades mais gerais da forma de um objeto e serão mostrados a seguir [Hu 1962]. Outro descritor bastante utilizados chamado de momentos ortogonais de Zernike também será mostrado.

Momentos Invariantes de Hu Considerados por muitos autores um dos mais populares métodos de descrição de forma por região, os momentos invariantes [Hu 1962], como o próprio nome já diz, também são invariantes às transformações geométricas citadas anteriormente. Para uma função bidimensional $f(x, y)$, os momentos de ordem p+q para p,q= 0,1,2... são dados pela fórmula 3.10.

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{3.10}$$

Um teorema de unicidade proposto por [Papoulis 1991] indica que se $f(x, y)$ for contínua por partes e possuir valores não nulos apenas em uma parte finita do plano, então os momentos de todas as ordens e a sequência de momentos (m_{pq}) é unicamente determinada por $f(x, y)$. Inversamente (m_{pq}) determinará $f(x, y)$ [Papoulis 1991] [Gonzales and Woods 1992]. Os momentos centrais podem ser expressos como mostrado na expressão 3.11.

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^p(y-\bar{y})^q f(x,y)dxdy \tag{3.11}$$

onde:

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ e } \bar{y} = \frac{m_{01}}{m_{00}}.$$

No caso de uma imagem digital discretizada, a equação 3.11 é substituída pelos somatórios da expressão 3.12.

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p(y-\bar{y})^q f(x,y) \tag{3.12}$$

Os momentos centrais normalizados, denotados por η_{pq} são definidos pela expressão 3.13.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{3.13}$$

sendo que:

$$\gamma = \frac{p+q}{2}.$$

O conjunto dos setes momentos invariantes pode ser derivado a partir dos segundos e terceiros momentos como em [Hu 1962]:

$\phi_1 = \eta_{2,0} + \eta_{0,2}$
$\phi_2 = (\eta_{2,0} + \eta_{0,2})^2 + 4\eta_{1,1}^2$
$\phi_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2$
$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} - \eta_{0,3})^2$
$\phi_5 = (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] + 3(\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})[3(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2]$
$\phi_6 = (\eta_{2,0} + \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3})$
$\phi_7 = (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] + (\eta_{3,0} - 3\eta_{1,2})(\eta_{2,1} + \eta_{0,3})[3(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2]$

Vários melhoramentos no cálculo dos momentos invariantes foram propostos ao longo do tempo. Em [Lin and Shen 1991] e [Lin 1991], por exemplo, são apresentadas novas maneiras de se calcular esses momentos invariantes de forma mais eficiente. Lin não só reformulou os momentos de Hu como introduziu uma nova ordem de momentos invariantes.

Momentos de Zernike Os momentos de Zernike utilizam funções de bases ortogonais e são menos sensíveis a ruídos que os momentos de Hu. Tais momentos são derivados dos polinômios de Zernike que formam um conjunto ortogonal sobre o interior de

um círculo unitário [Brandt 1999]. Assim, os momentos de Zernike bidimensionais [Long et al. 2003] são dados por:

$$A_{m,n} = \frac{m+1}{\pi} \int f(x,y)[V_{m,n}(x,y)]^* dx\ dy$$

onde,

$x^2 + y^2 \leq 1$
$m = 0, 1, 2, \ldots, \infty$
$f(x,y)$ é a função a ser descrita (uma imagem)
$V_{m,n}$ denota a função de base Zernike
$*$ denota o complexo conjugado de $V_{m,n}$
n é um inteiro que representa a dependência angular ou rotação

Para maiores detalhes sobre os Momentos de Zernike veja [Bin and Jia-Xiong 2002] e [Brandt 1999].

Existem ainda outros descritores nesta categoria como os descritores genéricos de Fourier [Rui et al. 1997], Métodos *Grid* e Matrizes de forma [Zhang and Lu 2004]. Outros descritores simples para esta categoria também chamados de descritores escalares internos podem ser usados. Alguns deles são apresentados a seguir.

Área Para uma imagem binária, a área A de um objeto pode ser facilmente calculada. Para imagens digitais a área é simplesmente o número de pixels que um objeto possui. Esta característica é naturalmente dependente da escala e do tamanho do objeto, mas é invariante à rotação e translação.

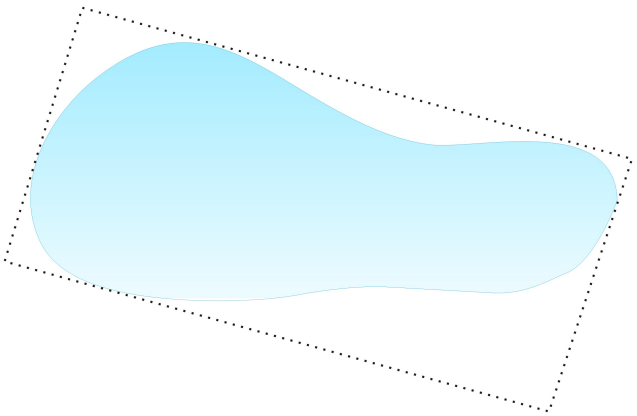


Figura 3.12: Um objeto com o menor retângulo de contorno que define a direção e excentricidade da forma.

Direção Se um objeto for alongado pode-se extrair dele uma característica chamada direção ou orientação do eixo maior. Esta característica pode ser definida como a direção do ângulo do maior lado do menor retângulo (ou elipse) que contém o objeto (como na

figura 3.12). A direção υ também pode ser calculada por momentos invariantes como em [Sonka et al. 2007] dada pela fórmula 3.15:

$$\upsilon = \frac{1}{2}\tan^{-1}\Big(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\Big). \tag{3.14}$$

Excentricidade Uma medida para a excentricidade pode ser baseada no quociente da divisão entre o maior e o menor eixo do retângulo (ou elipse) que contém o objeto. Nesse caso também há uma aproximação pelos momentos invariantes:

$$\epsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}}{A}, \tag{3.15}$$

onde A é a área do objeto [Brandt 1999].

Número de Euler O número de buracos (ou furos) de um objeto pode ser usado como uma característica. O conhecido número de Euler ξ [Gonzales and Woods 1992] pode ser definido como a diferença entre o número de componentes conectados N_c e o número de furos na região N_h, isto é:

$$\xi = N_c - N_h. \tag{3.16}$$

Retangularidade Uma medida de como uma região pode ser aproximada por um retângulo é chamada retangularidade [Brandt 1999]. Essa medida pode ser definida como a divisão entre a área da região do objeto e a área do menor retângulo que o encobre (como na Figura 3.12). Outras medidas baseadas nessa também são usadas (circularidade, elipsidade) [Sonka et al. 2007].

Compassidade A compassidade pode ser definida como a divisão do quadrado do perímetro P da fronteira pela área A da região, ou seja:

$$\varsigma = \frac{P^2}{A}. \tag{3.17}$$

Métodos Estruturais

Um tanto similar aos métodos estruturais de contorno, os métodos estruturais baseados em regiões decompõem estas em partes que são então usadas para as descrições. Os métodos estruturais baseados em regiões mais conhecidos são: fecho convexo e saliências explanados a seguir.

Fecho Convexo Uma região r é convexa se e somente se para quaisquer dois pontos $x_1, x_2 \in r$, o segmento de reta x_1x_2 estiver totalmente dentro da região r. O fecho convexo de uma região é a menor região convexa H que satisfaz a condição $R \subset H$. A diferença $H - R$ é chamada deficiência convexa D da região R [Zhang and Lu 2004]. Regiões e deficiências convexas podem ser descritas em forma de árvores, sendo estas representadas pelos nós folhas. A Figura 3.13 mostra uma forma composta de regiões, representada pelo método do fecho convexo.

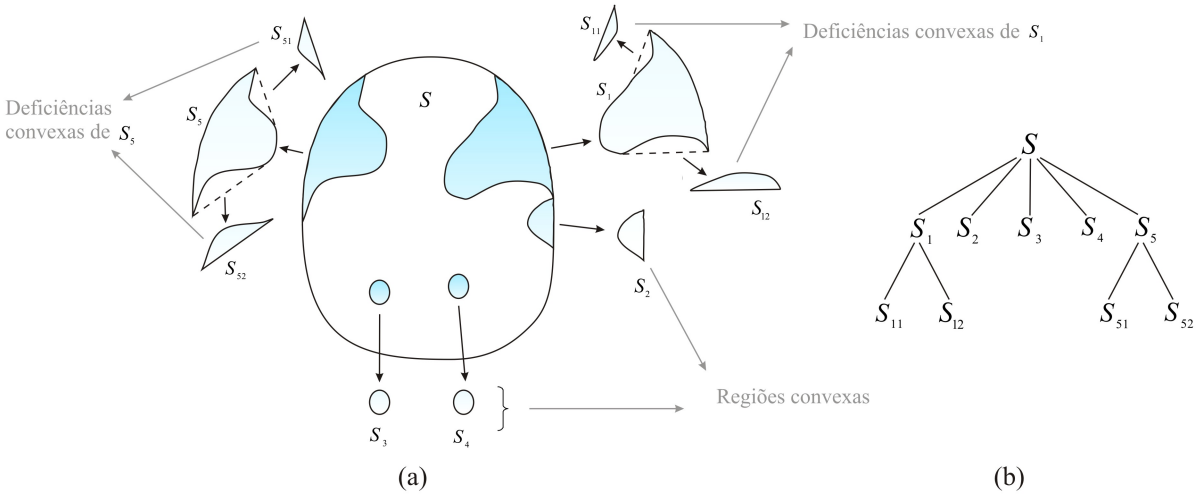


Figura 3.13: Fecho convexo: (a) Regiões e deficiências convexas (b) representação das regiões e deficiências convexas por árvore.

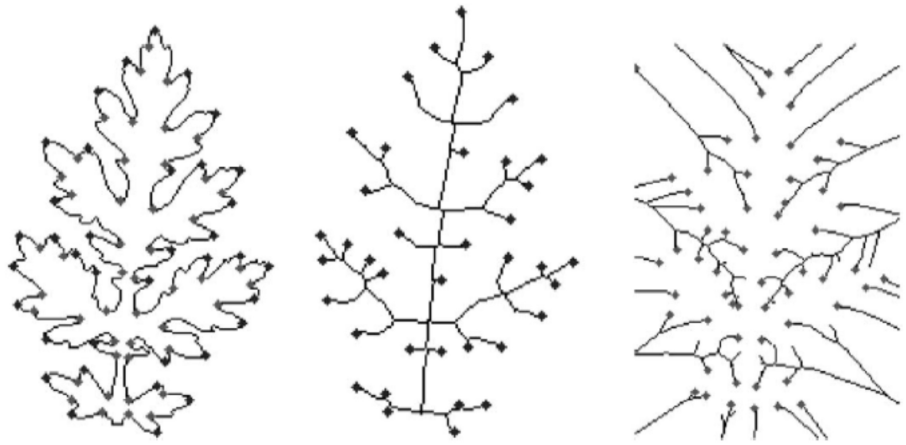
Saliências São usadas para descrever os pontos do contorno de alta curvatura ou as extremidades dos esqueletos de forma. Na Figura 3.14 temos três ilustrações de saliências de uma folha. Em (a) temos os pontos de saliência do contorno, em (b) temos as saliências do esqueleto interno e em (c) as saliências do esqueleto externo. Maiores detalhes sobre saliências pode ser encontrados em [Pedrosa et al. 2008].

3.5.2 Técnicas Baseadas no Contorno

Nas técnicas baseadas no contorno o foco principal é a informação das fronteiras de um objeto. Existem, basicamente, dois tipos de abordagens para a modelagem do contorno de formas: a abordagem contínua (global) e a abordagem discreta (estrutural).

Na abordagem global, um vetor de característica numérico derivado de toda a borda é usado para descrever a forma. Medidas de similaridade entre formas são usualmente tomadas por alguma métrica de distância.

Abordagens estruturais quebram a borda em segmentos, chamados primitivas, usando algum critério particular. A representação final é usualmente uma *string* ou um grafo. A medida de similaridade é feita por casamento de *strings* ou de grafos.



(a) (b) (c)

Figura 3.14: Pontos de saliências de uma folha; (a) saliências do contorno (b) e (c) saliências dos esqueletos internos e externos, respectivamente.

Os descritores baseados no contorno supõem que o contorno da imagem em análise tenha sido detectado previamente.

Métodos Globais

Histograma de Direção da Borda Os primeiros experimentos em extrações de características de forma foram feitos utilizando histogramas de direção da borda [Brandt 1999]. Seu sucesso provavelmente reside na facilidade de extração sem a necessidade de segmentação da imagem além de capturar a informação geral da forma da imagem e se tornar invariante a translação dos objetos na imagem. Esse método se baseia no princípio de que a imagem é pré-processada de modo a se obter seu mapa de bordas em várias direções. Para isso, o modelo de espaço de cores é transformado no modelo de cores HSV onde a variável Hue é desconsiderada por questões explicadas em [Brandt 1999]. As outras duas variáveis são convoluídas, ou seja, transformadas em uma nova imagem através de operadores de Sobel em oito direções (ver Figura 3.15).

A imagem resultante desta operação é então binarizada com um valor de *threshold* apropriado para cada variável. Os valores de *threshold* são, então, manualmente fixados para que sejam os mesmos em todas as imagens. Os histogramas de borda são calculados contando os pixels da borda em cada uma das oito direções ($H^o_0, H^o_{45}, H^o_{90}, H^o_{135}$, $H^o_{180}, H^o_{225}, H^o_{270}, H^o_{315}$). A invariância em relação à escala é obtida pela normalização do histograma com o respectivo número de pontos da borda de cada imagem [Brandt 1999].

Distribuição da Borda (Chord Distribution) Um *chord* é uma reta que une dois pontos do contorno. Seja uma imagem binária onde $b(x, y) = 1$ nos pontos de borda e $b(x, y) = 0$ nos outros pontos, a distribuição dos comprimentos e ângulos de todos os *chords* que ligam os pontos onde $b(x, y) = 1$ formam um descritor. Essa distribuição pode ser armazenada

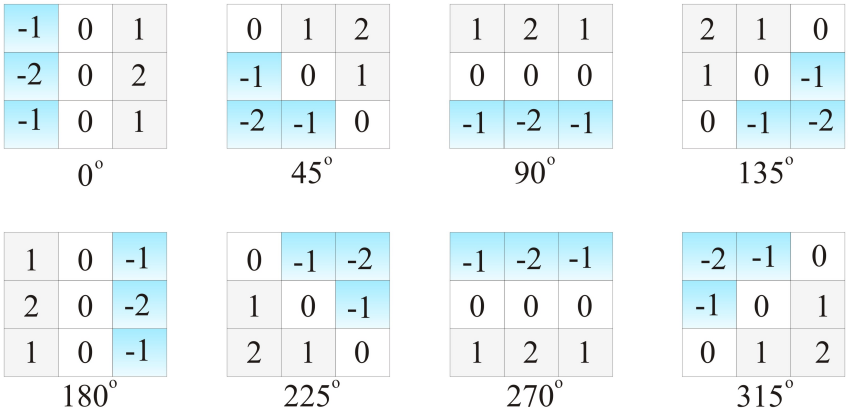


Figura 3.15: Operadores de Sobel em oito direções

em um histograma que poderá ser normalizado e, consequentemente, será invariante em relação à rotação e escala.

Assinatura de forma Uma assinatura representa uma forma por uma função unidimensional derivada dos pontos de borda [Zhang and Lu 2004]. Existem muitas assinaturas de forma: distância ao centro, ângulo tangente, curvatura, etc [Zhang and Lu 2004] . Assinaturas de formas normalizadas são invariantes a translação e escala, contudo, são sensíveis a ruído. A Figura 3.16 mostra uma forma e sua assinatura correspondente obtida pelo método da distância ao centro.

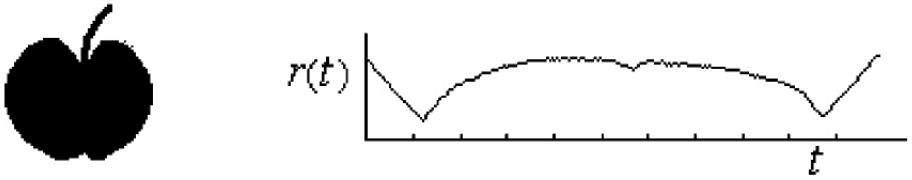


Figura 3.16: A forma de uma maçã e sua assinatura por distância ao centro [Zhang and Lu 2004].

Momentos de borda Os momentos de borda são usados para reduzir as dimensões das representações da forma. Por exemplo, seja uma borda representada por uma assinatura $z(i)$ descrita da mesma maneira explicitada no item anterior, o r-ésimo momento m_r e o momento central μ_r podem ser calculados como em [Zhang and Lu 2004] por:

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r \qquad e \qquad \mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i) - m_1]^r \tag{3.18}$$

onde N é o número de *pixels* do contorno. Momentos de borda normalizados como $\overline{m}_r = m_r/(\mu_2)^{r/2}$ e $\overline{\mu}_r = \mu_r/(\mu_2)^{r/2}$ são invariantes às transformações de rotação, translação e escala [Zhang and Lu 2004].

Transformadas espectrais Transformadas espectrais focam, principalmente, no problema de ruído e nas variações de borda analisando a forma no domínio espectral. Descritores espectrais incluem entre outros os descritores de *Fourier* e descritores *wavelet* e podem ser derivados de transformadas espectrais sobre assinaturas de formas unidimensionais ou derivados diretamente dos pontos do contorno [Zhang and Lu 2004]. Descritores de *Fourier* consistem na obtenção de um conjunto de coeficientes que representam uma dada borda.

Basicamente, existem duas maneiras de se descrever as bordas de uma imagem através dos descritores de Fourier. Uma maneira é utilizar as coordenadas cartesianas e o outro modo é representar as bordas como a função do ângulo das tangentes versus a distância a um ponto de referência.

A Figura 3.17 adaptada de [Gonzales and Woods 1992] mostra uma fronteira digital de n pontos no plano xy. Começando de um par de pontos arbitrários sequenciais (x_0, y_0), (x_1, y_1), pode-se encontrar os pares ordenados (x_2, y_2), (x_3, y_3), ..., (x_{N-1}, y_{N-1}) ao longo da fronteira.

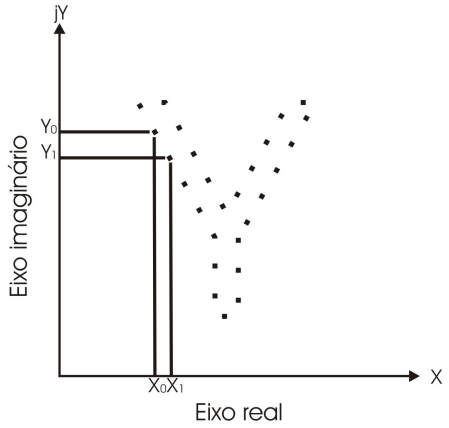


Figura 3.17: Uma fronteira digital e sua representação por uma seqüência complexa. Os pontos (x_0, y_0) e (x_1, y_1) são (arbitrariamente) os dois primeiros pontos da seqüência.

Essas coordenadas podem ser expressas na forma $x(k) = x_k$ e $y(k) = y_k$. Com essa notação, a própria fronteira pode ser representada como uma sequência de coordenadas $s(k) = (x(k), y(k))$, para k = 0,1,2,... $n - 1$. Além disso, cada par pode ser tratado como um número complexo como mostrado na expressão 3.19, para k = 0,1,2,...$N - 1$ [Gonzales and Woods 1992].

$$s(k) = x(k) + jy(k) \tag{3.19}$$

Resumindo: o eixo X é tido como o eixo real, enquanto o eixo Y como o eixo imaginário de uma sequência de números complexos. Embora a notação da seqüência tenha sido reformulada, a própria natureza da fronteira não foi alterada. Obviamente, essa representação possui uma grande vantagem: ela reduz um problema de duas dimensões a uma só dimensão.

Os descritores de Fourier podem ser computados a partir da transformada discreta de Fourier (DFT) de $s(k)$ que é definida na expressão 3.20, para $u = 0,1,2,...,n-1$.

$$a(u) = \frac{1}{N} \sum_{x=0}^{n-1} s(k) e^{-j2\pi(uk/N)} \tag{3.20}$$

Os coeficientes complexos $a(u)$ são chamados de Descritores de Fourier da fronteira.

Os descritores de Fourier apresentados acima são invariantes quanto ao ponto inicial da imagem. As invariâncias em relação à rotação, translação podem ser obtidas através de algumas modificações nos algoritmos. [Gonzales and Woods 1992] e [Rui et al. 1997] propõem algumas dessas mudanças.

O vetor de características pode então ser formado pelos N coeficientes e a distância euclidiana pode ser usada para medir a semelhança entre as imagens [Brandt 1999]. As transformadas espectrais produzem representações compactas, de fácil casamento e pouco sensíveis a ruído [Zhang and Lu 2004].

Descritores Escalares Externos Existem vários tipos de descritores escalares como por exemplo o chamado *aspect radio* que é a soma da maior distância vinda da reta perpendicular mais longa que separa dois pontos da borda em ambos os lados (como mostra a figura 3.18) dividido pelo tamanho da mesma. Outro descritor conhecido é a energia da borda (*Bending Energy*) que é a energia necessária para se dobrar uma vara até a forma desejada. Vários outros descritores escalares externos foram propostos ao longo dos tempos como o perímetro da borda, a convexidade, o casamento elástico, a circularidade, dentre outros e podem ser encontrados em [Zhang 2002], [Zhang and Lu 2004] e [Brandt 1999].

Métodos Estruturais

Outro ramo dos métodos de análise de bordas é a representação estrutural de forma. Com a abordagem estrutural, as formas são quebradas em segmentos de contorno chamados primitivas. Os métodos estruturais diferem entre si na seleção de primitivas e na organização destas para a representação de formas. Representações estruturais podem ser codificadas como uma *string* da forma:

$$S = s_1, s_2, \ldots, s_n$$

onde s_i poder ser um elemento de um código da cadeia, um lado de um polígono, etc. s_i pode conter atributos como comprimento, curvatura, energia da curvatura, orientação, etc.

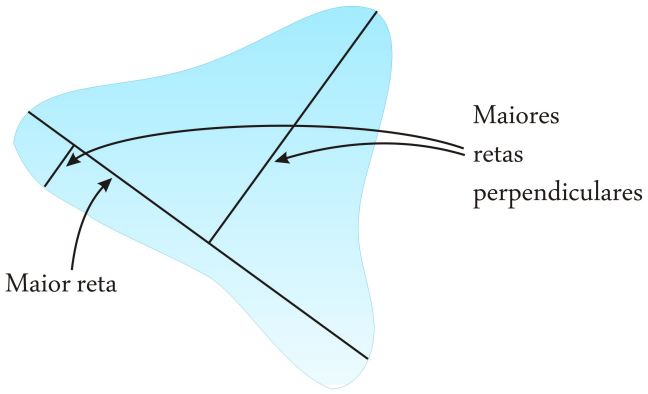


Figura 3.18: *Aspect Ratio*: Soma dos tamanhos das maiores retas perpendiculares à maior reta dividido pelo tamanho da maior reta.

Uma *string* S pode ser comparada com outras para verificar o grau de casamento (*matching*) ou usada como entrada de um analisador sintático de alto nível como autômatos e máquina de *Turing*, que verificam se a forma descrita pela *string* S pertence a uma determinada categoria.

A seguir alguns métodos estruturais são apresentados:

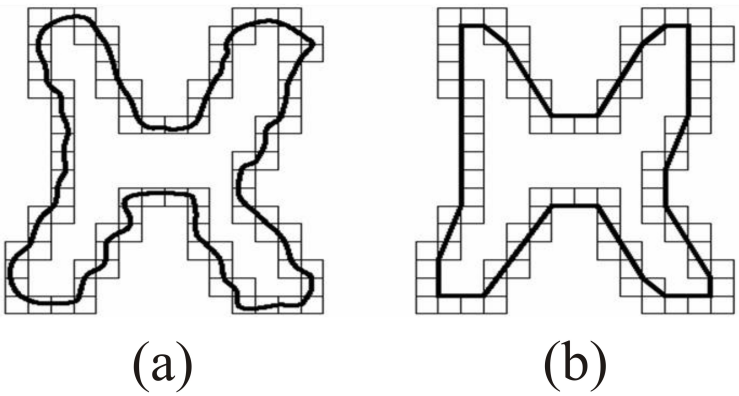


Figura 3.19: Aproximação poligonal. Contorno sobreposto por pequenos quadrados (a) e proximação por polígono traçada internamente às células (b).

Aproximação por polígonos Uma fronteira digital pode ser aproximada por um polígono. Na aproximação por polígonos o objetivo é encontrar um polígono com o menor número de segmentos, em acordância com a precisão escolhida.

O polígono resultante deve ter um perímetro mínimo, ou seja, dada uma fronteira conexa S, alguma técnica deve ser utilizada para que seja gerado um polígono que possua forma semelhante à original e o menor número possível de lados. Frequentemente é utilizada a sobreposição da fronteira com um conjunto de células de tamanhos pré-definidos de maneira que o polígono possa ser traçado dentro das células [Zhang 2002]. A Figura 3.19 ilustra tal procedimento.

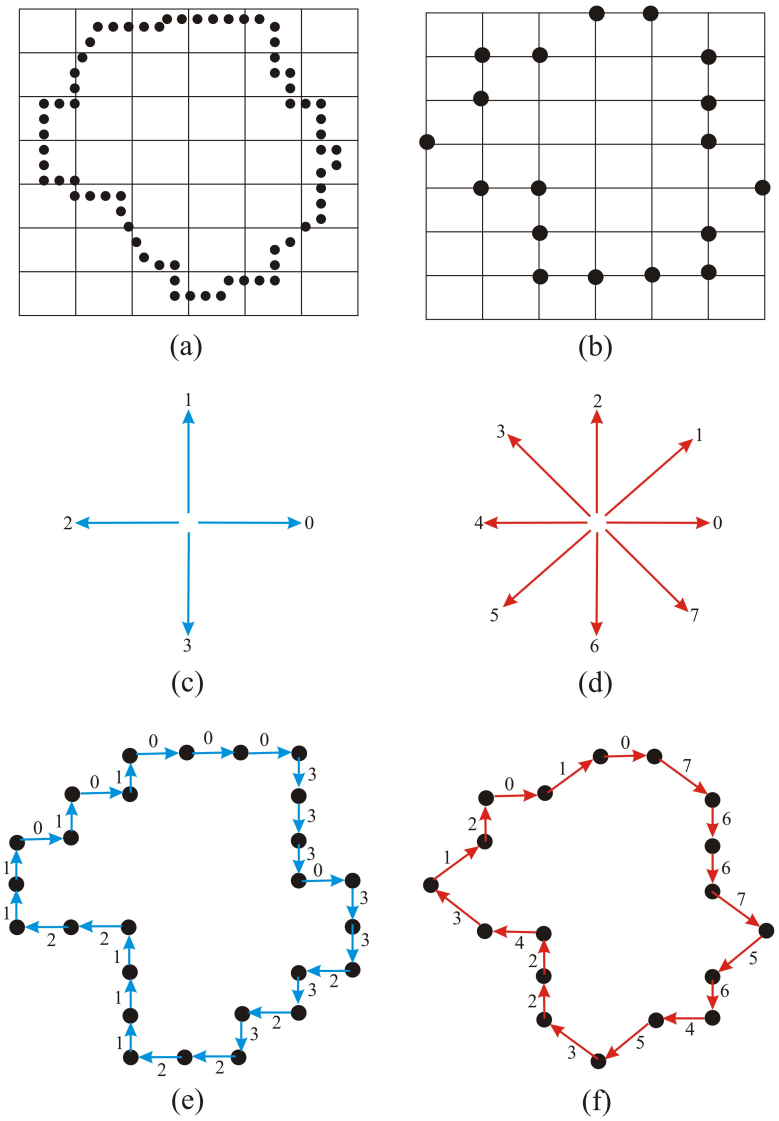


Figura 3.20: Código da Cadeia. Contorno sobreposto por uma grade (a); aproximação dos pontos do contorno para os pontos de intersecção da grade (b);

Código da Cadeia Na representação por códigos da cadeia, o contorno é representado por uma sequência conectada de segmentos de reta com comprimento e direções específicas. Neste método a borda da imagem é incluída em uma grade, onde um dado ponto de borda é aproximado para o ponto da grade mais próxima, obtendo assim uma borda amostrada como ilustrado na Figura 3.20(a) e (b). O tamanho da grade que sobrepõe o contorno da imagem original determina a dimensão do código. De um ponto de partida selecionado, um código da cadeia pode ser gerado seguindo os pontos de bordas usando 4, 8 ou N ($N > 8$; $N = 2^k, k \in \mathbb{N}$) direções como ilustrado na figura 3.20(c) e (d). Cada segmento é codificado de acordo com um número que corresponde à sua direção, e esta sequência (no sentido horário) forma o chamado código da cadeia como mostrado na figura 3.20(e) e (f).

A prinípio o código irá variar de acordo com o ponto inicial escolhido mas existem alguns métodos para tornar o algoritmo invariante. Uma maneira de fazer isso é tratar o código gerado como uma seqüência circular de números, sendo o ponto inicial dado de modo que o número formado seja de menor magnitude possível. Outras formas de normalização são apresentadas em [Zhang and Lu 2004].

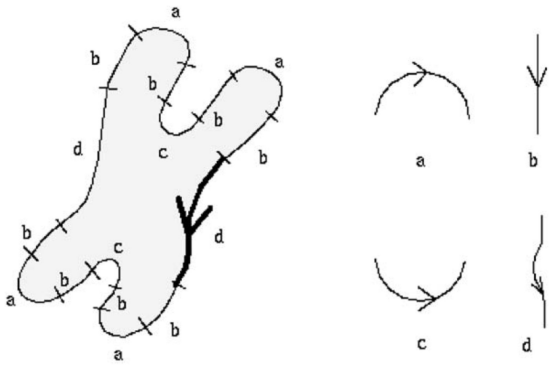


Figura 3.21: Analise sintática da forma de um cromossomo [Zhang and Lu 2004]. À direita temos os elementos primitivos. À esquerda temos o cromossomo representado a partir dos elementos primitivos;

Análise Sintática A análise sintática parte do suposto que a composição de uma forma é análoga à composição de uma linguagem. Nos métodos sintáticos as formas são representadas por primitivas e seus relacionamentos descritos por gramática [Zhang and Lu 2004]. Essas primitivas podem ser representadas por polinômios de 2ª ordem (círculos, parábolas) e cada primitiva têm um grau de curvatura, concavidade, etc. Por exemplo, o cromossomo da Figura 3.21 pode ser representado por suas primitivas (a, b, c, d) e descrito pela *string*:

$$S = dbabcbabdbabcbab.$$

Neste caso a similaridade entre formas é medida por métodos de casamento de *strings*. Maiores detalhes sobre métodos sintáticos para descrição e casamento de formas podem ser encontrados em [Fu 1974].

3.6 Textura

Uma considerável abordagem para a descrição de imagens é a quantificação de seu conteúdo de textura pois a grande maioria das superfícies naturais as exibem [Gonzales and Woods 1992]. A textura é uma propriedade inata de todas as superfícies e essa propriedade pode ser facilmente usada para representar diferentes matérias. Uma imagem pode ser considerada como sendo composta de um número de regiões com diferentes padrões de textura e as propriedades de textura dessas regiões podem ser usadas na recuperação de imagens baseada em conteúdo. Ainda não existe uma definição formal e precisa a respeito da textura, mas alguns autores acor-

dam em definí-la como sendo a variação na intensidade das imagens que formam certos padrões repetitivos [Tuceryan and Jain 1993]. A imagem 3.22 mostra alguns exemplos de texturas.

Figura 3.22: Exemplos de imagens naturais com textura.

Esses padrões podem ser o resultado das propriedades físicas da superfície do objeto como a suavidade, rugosidade e regularidade, ou ser o resultado de diferenças de reflexão através da absorção ou não da luz na superfície. Os descritores de textura fornecem medidas para essas propriedades. Embora seja fácil para um ser humano reconhecer uma textura prontamente e subjetivamente, para sistemas automáticos, essa tarefa se torna um pouco mais complicada e necessita de algoritmos mais complexos. As três principais abordagens utilizadas na recuperação de imagens para a formação do vetor de características são a estatística, a estrutural, também tratada por alguns autores de abordagem geométrica ou sintática, e a espectral também chamada de métodos de processamento de sinais. De modo geral as abordagens estatísticas caracterizam a textura como suave, áspera, granular, e assim por diante. As técnicas estruturais tratam de arranjos de primitivas da imagem, como a descrição de textura baseada em linhas paralelas regularmente espaçadas. As técnicas espectrais baseiam-se em propriedades

do espectro de transformadas sendo usadas basicamente na detecção de periodicidade global em uma imagem através da identificação de picos de alta energia no espectro. As subseções seguintes tratarão mais especificamente sobre cada uma dessas técnicas de descrição.

3.6.1 Abordagens Estatísticas

As propriedades das texturas são descritas por medidas estatísticas tais como: contraste, correlação, entropia, uniformidade, densidade, aspereza, rugosidade, regularidade, linearidade, direcionalidade e frequência. Estas medidas são fortemente baseadas nos aspectos da percepção humana de textura. Devido a grande popularidade dos métodos estatísticos na descrição de textura,a seguir alguns deles serão demonstrados.

Matriz de co-ocorrência Na década de 70, Robert M. Haralick propôs a representação de textura através da matriz de co-ocorrência de níveis de cinza [Aksoy and Haralick 1998]. Esta matriz é construída baseando-se na orientação e distância entre os pixels da imagem e extraindo estatísticas significantes da matriz como uma representação de textura. Seja z uma variável aleatória denotando a intensidade discreta de uma imagem e seja $p(z_i)$, $i = 1, 2, 3, ..., l$ o histograma correspondente, onde l é o número de níveis distintos de intensidade que se queira representar. Então o n-ésimo momento de $z = (z_1, z_2, ...z_l)$ em torno da média é dado pela expessão 3.21.

$$\mu_n(z) = \sum_{i=1}^{L} (z_i - m)^n p(z_i) \tag{3.21}$$

em que m é o valor médio de z (a intensidade média) demonstrado na expressão 3.22.

$$m = \sum_{i=1}^{L} (z_i) p(z_i) \tag{3.22}$$

Pode-se notar, a partir da equação 3.21 que $\mu_0 = 1$ e $\mu_1 = 0$. O segundo momento (também chamado de variância e denotado por $\sigma^2(z)$) possui uma importância particular para a descrição de textura, sendo uma medida de contraste de nível de cinza que pode ser usada no estabelecimento de descritores de suavidade relativa. Por exemplo, a medida apresentada pela equação 3.23 é 0 para áreas de intensidade constante ($\sigma^2(z) = 0$ se todos os z_i's possuírem o mesmo valor) e se aproxima de 1 para grandes valores de $\sigma^2(z)$.

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \tag{3.23}$$

O terceiro momento é uma medida de anti-simetria do histograma, enquanto o quarto momento é uma medida de seu achatamento ou planaridade. O quinto e os outros mo-

mentos mais altos não são tão facilmente relacionados ao formato do histograma, mas fornecem informação quantitativa adicional para o conteúdo da textura.

Medidas de textura calculadas apenas a partir do histograma não denotam a informação sobre a posição relativa dos pixels em relação uns aos outros. Um modo de se trazer essa informação para o vetor de características é considerar não apenas a distribuição de intensidades, mas também as posições dos pixels que possuem valores de intensidade iguais ou semelhantes.

Seja P um operador de posição e seja $\mathbf{A}$ uma matriz $k \times k$ cujo elemento a_{ij} seja o número de vezes que pontos com o nível de cinza z_i ocorram (na posição especificada por P) relativamente a pontos com nível de cinza z_j, com $1 \leq i,\ j \leq k$. Por exemplo, considere uma imagem com três níveis de cinza, z_1=0, z_2=1 e z_3=2, dispostos em uma matriz:

$$\begin{array}{ccccc} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}$$

A definição do operador de posição P como "um *pixel* à direita e um *pixel* abaixo" leva à seguinte matriz $\mathbf{A}$.

$$\mathbf{A} = \begin{pmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{pmatrix}$$

sendo que, por exemplo, a_{11} (acima e à esquerda) é o número de vezes que um ponto com nível de cinza $z_1 = 0$ aparece em um pixel abaixo e um à direita do pixel com o mesmo nível de cinza, enquanto que a_{13} (acima e à direita) é o número de vezes que um ponto com nível de cinza $z_1 = 0$ aparece em um pixel abaixo e um à direita do pixel com o nível de cinza igual a $z_3 = 2$. O tamanho de $\mathbf{A}$ é estritamente determinado pelo número de níveis de cinza diferentes em uma dada imagem. Por isso, a aplicação dos conceitos discutidos nessa seção usualmente requer que as intensidade sejam requantizadas em um número menor de níveis de cinza, de maneira a diminuir a dimensionalidade do vetor de características.

Seja n o número total de pares de pontos em uma imagem que satisfaçam P (no exemplo anterior, n=16). Se uma matriz $\mathbf{C}$ for formada dividindo-se cada elemento de $\mathbf{A}$ por n, então c_{ij}, será uma estimativa da probabilidade conjunta de que um par de pontos satisfazendo P possuirá os valores (z_i,z_j). A matriz $\mathbf{C}$ é chamada de *matriz de co-ocorrência de níveis de cinza*. Uma vez que $\mathbf{C}$ depende de P, a presença de uma dada textura pode ser detectada através da escolha de um operador de posição apropriado. Por exemplo, o operador usado no caso anterior é sensível às bandas de intensidade constante e inclinadas a -45º. (Note que o maior valor em $\mathbf{A}$ era $a_{11} = 4$, parcialmente devido a

uma faixa de pontos de intensidade 0 e de inclinação -45º.) De maneira mais genérica, o problema é analisar uma dada matriz **C** para categorizar a textura de uma região sobre a qual **C** foi calculada. Uma vez computada a matriz de co-ocorrência, várias características de textura podem ser extraídas desta. [Aksoy and Haralick 1998] propôs um conjunto de 14 estatísticas obtidas da matriz de co-ocorrência para caracterizar texturas. Destas as mais utilizadas na recuperação de imagens são as listadas na tabela 3.1, onde μ_i e μ_j são as médias e σ_i e σ_j são os desvios padrões de P.

Característica	Equação
Contraste	$f_1 = \sum_i \sum_j (i-j)^2 P(i,j)$
Energia	$f_2 = \sum_i \sum_j P^2(i,j)$
Entropia	$f_3 = \sum_i \sum_j P(i,j) - \log P(i,j)$
Homogeneidade	$f_4 = \sum_i \sum_j \frac{P(i,j)}{(1+\|i-j\|)}$
Correlação	$f_5 = \frac{1}{\sigma_i \sigma_j} \sum_i \sum_j (i-\mu_i)(j-\mu_j) P(i,j)$

Tabela 3.1: Características deriváveis da matriz de co-ocorrência.

Características extraídas da matriz de co-ocorrência apresentam boa capacidade de discriminação de texturas. A principal desvantagem é o seu alto custo computacional.

Vizinhança de Textura Um bom método para se descrever a textura de uma imagem é o chamado descritor de vizinhança de textura [Laaksonen et al. 2000]. Nesse método a imagem é representada em níveis de cinza e um descritor de dimensão oito é gerado da seguinte forma: se o valor de um pixel na vizinhança de oito é maior que o pixel central então, o contador direcional indicando a respectiva direção é incrementado. Isto é feito para todos os *pixels* na imagem e em seguida o descritor é normalizado.

Características Tamura As características Tamura [Tamura et al. 1978] combinam aspectos de rugosidade, contraste, direcionalidade, semelhança de linhas, regularidade e aspereza. Estas medidas foram projetadas em conformidade com estudos psicovisuais acerca da percepção humana de textura. Os três primeiros componentes das características Tamura foram usados em dois *CBIRs* populares: o *QBIC* [Flickner et al. 1995] e o *Photobook* [Pentland et al. 1996]. A computação destas três características são dadas a seguir. As demais medidas (semelhança de linhas, regularidade e aspereza) podem ser vistas em [Tamura et al. 1978].

Rugosidade - é uma medida da granularidade da textura. Inicialmente, para cada *pixel* (x, y) são calculadas as médias $A_k(x, y)$ dos níveis de cinza em cada vizinhança $2^k, k = 2, 3, ..., L$. Posteriormente, para cada *pixel* e cada vizinhança é calculada a diferença entre pares de médias vizinhas não-sobrepostas nas direções horizontal h e vertical v,

$$\begin{aligned} E_{k,h}(x,y) &= \left| A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y) \right| \\ E_{k,v}(x,y) &= \left| A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1}) \right| \end{aligned} \tag{3.23}$$

Depois disso, é calculado para cada *pixel* o tamanho da vizinhança S_{best} que maximiza o valor de E,

$$S_{best}(x,y) = 2^k \tag{3.24}$$

Finalmente, a rugosidade é definida como a média dos valores de S_{best},

$$F_{rugosidade} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j) \tag{3.25}$$

onde $m \times n$ corresponde à dimensão da imagem.

Contraste - é calculado pela seguinte formula:

$$F_{contraste} = \frac{\sigma}{\alpha_4^{1/4}} \tag{3.26}$$

onde $\alpha_4 = \mu_4/\sigma^4$ é o curtose , μ_4 é o quarto momento sobre a média, e σ^2 é a variância. Esta fórmula pode ser aplicada à imagem inteira ou a regiões.

Direcionalidade - para computar a direcionalidade é feita uma convolução entre a imagem com os operadores direcionais $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ e $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$, e em seguida é calculado um vetor gradiente para cada *pixel*. A magnitude $|\Delta G|$ e o ângulo θ de cada vetor são dados por:

$$\begin{aligned} |\Delta G| &= \big(|\Delta_H| + |\Delta_V|\big)/2 \\ \theta &= \tan^{-1}(\Delta_V/\Delta_H) + \pi/2 \end{aligned} \tag{3.26}$$

onde Δ_H e Δ_V são as diferenças horizontais e verticais da convolução.

Em seguida é construído um histograma dos valores de θ (histograma de direções), considerando somente os valores de magnitude $|\Delta G|$ maior que um limiar T. Este histograma H_D apresenta fortes picos para imagens altamente direcionais e é relativamente aplanado para imagens sem fortes orientações. O histograma inteiro é então resumido por um valor de direcionalidade por:

$$F_{direcionalidade} = \sum_{\forall p \in P} \sum_{\phi \in w_p} (\phi - \phi_p)^2 H_D(\phi) \tag{3.27}$$

onde P é o conjunto de picos do histograma, w_p é o conjunto de intervalos direcionais entre dois vales que delimitam um pico, e ϕ_p é a direção onde se encontra o pico.

***Características* Wold** A decomposição *Wold* [Liu and Picard 1996] provê outra abordagem para a descrição de textura em termos de propriedades perceptivas. As três componentes *Wold*, harmonia, *evanescence* e indeterminismo, correspondem a periodicidade, direcionalidade e randomicidade de texturas, respectivamente. Texturas periódicas têm um forte componente harmônico, texturas altamente direcionais têm um forte componente *evanescence*, e texturas menos estruturadas tendem a ter um forte componente indeterminístico.

A decomposição *Wold* bidimensional permite que um campo randômico discreto e homogêneo $\{y(m,n),(m,n\in\mathbb{Z})\}$ seja decomposto em três componentes mutuamente ortogonais:

$$y(m,n)=u(m,n)+d(m,n)=u(m,n)+h(m,n)+e(m,n) \tag{3.28}$$

onde $u(m,n)$ é o componente indeterminístico, e $d(m,n)$ é o componente determinístico que pode ser decomposto no componente harmônico $h(m,n)$ e no componente *evanescence* $e(m,n)$. No domínio da frequência temos que:

$$F_y(\xi,\eta)=F_u(\xi,\eta)+F_d(\xi,\eta)=F_u(\xi,\eta)+F_h(\xi,\eta)+F_e(\xi,\eta) \tag{3.29}$$

onde $F_y(\xi,\eta),F_u(\xi,\eta),F_d(\xi,\eta),F_h(\xi,\eta),F_e(\xi,\eta)$ são as funções de distribuição espectral de $\{y(m,n)\}$, $\{u(m,n)\}$, $\{d(m,n)\}$, $\{h(m,n)\}$ e $\{e(m,n)\}$, respectivamente.

No domínio espacial, os três componentes ortogonais podem ser obtidos pela estimação de probabilidade máxima, que consiste em ajustar um processo auto-regressivo de alta ordem, minimizando uma função custo e resolvendo um conjunto de equações lineares. No domínio da frequência, componentes *Wold* podem ser obtidos por limiares globais de magnitudes do espectro de Fourier das imagens.

Descritor Global de Textura Em [Deselaers et al. 2008] um descritor de características de textura é descrito como a soma de vários descritores. Estes descritores são muito utilizados em imagens médicas e não em imagens comuns, por isso não serão detalhados. Eles são citados a seguir, para um melhor entendimento procurar em [Deselaers et al. 2008].

Dimensão Fractal Um fractal (conhecida inicialmente como curva monstro) é um objeto geométrico que pode ser dividido em partes, cada uma das quais semelhante ao objeto original. O mesmo possui medidas númericas ou estatísticas que são preservadas em diferentes escalas. As definições de fractais geralmente implicam em alguma forma de auto-similaridade estatística (mesmo a dimensão fractal é uma medida numérica preservada em diferentes escalas). A dimensão fractal mede a rugosidade de uma superfície e pode ser calculada de várias formas como o método de *BoxCounting*. Mais detalhes deste método pode ser visto em [Deselaers et al. 2008].

Aspereza A aspereza caracteriza o tamanho das particulas de uma imagem. Uma forma de se medir é fazer o cálculo da variância na imagem [Deselaers et al. 2008].

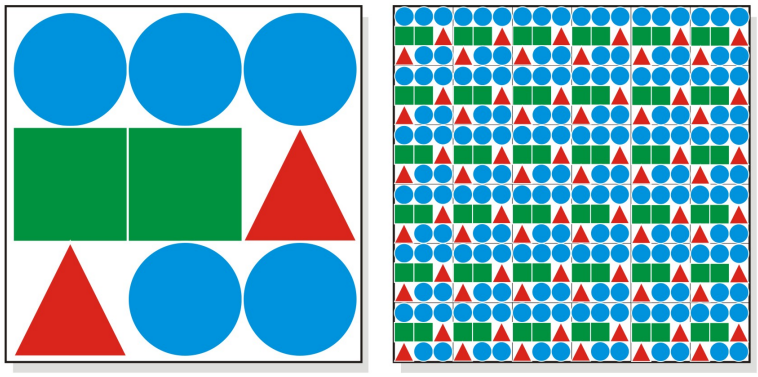
Entropia É uma medida da organização do conteúdo de informação de uma imagem.

Diferenças Espaciais de Níveis de cinza Descreve a relação de brilho dos *pixels* com seus vizinhos. Pode ser calculada pela matriz de co-ocorrência da mesma forma citada nessa seção.

Função de AutoCorrelação Circular de Moran Mede a rugosidade da textura. Para o cáculo, uma série de funções de correlação é usada. Mais detalhes em [Deselaers et al. 2008].

3.6.2 Abordagens Estruturais

Segundo foi mencionado no início desta seção, uma segunda abordagem de representação de textura baseia-se em conceitos estruturais. Por exemplo, suponha que exista um regra $S \rightarrow aS$, que indica que o símbolo S pode ser reescrito como aS ou de acordo com suas derivações. A cadeia $aaaS$ seria o resultados de três aplicações dessa regra por exemplo. A partir dessa idéia pode-se gerar um esquema adicionando mais regras a ele. Por exemplo, seja o conjunto de valores: a,b e c. Se a representar um "círculo azul à direita", b representar um "triângulo vermelho abaixo" e c representar um "quadrado verde à esquerda" e estabelecendo as seguintes regras: $S \rightarrow bA$, $A \rightarrow cA$, $A \rightarrow c$, pode-se gerar uma cadeia do tipo $aaabccbaa$ que corresponde a uma matriz 3x3 de símbolos como mostrado na figura 3.23(a). Maiores padrões de textura, como aquele mostrado na figura 3.23(b) podem ser gerados facilmente da mesma maneira. (Pode-se notar que, no entanto, essas regras também podem gerar estruturas que não sejam retangulares.) Além disso esta abordagem é limitada, dado que a identificação automática destas primitivas é um problema difícil, e também porque esta é somente aplicada a texturas estritamente uniformes, o que é muito raro em imagens reais. Métodos estruturais para descrição de textura incluem operadores morfológicos, grafos de adjacência e diagramas de Voronoi [Gonzales and Woods 1992].



(a) (b)

Figura 3.23: (a)Exemplo de padrão gerado pelas regras $S \rightarrow bA$, $A \rightarrow cA$, $A \rightarrow c$; (b) padrão de textura bidimensional gerado pelas mesmas regras.

3.6.3 Abordagens Espectrais

A adaptação do espectro de Fourier é uma das técnicas mais utilizadas para descrever a orientação de padrões periódicos ou quase periódicos em uma imagem. Esses padrões globais de textura, embora sejam facilmente identificados no espectro como concentrações de agrupamentos de alta energia, são, em geral, difíceis de se detectar com métodos espaciais devido à natureza local dessas técnicas.

[Gonzales and Woods 1992] diz que as três principais características do espectro de Fourier para a descrição de texturas são: os picos resultantes do espectro os quais indicam a direção dos padrões de textura, a posição dos picos no plano da frequência que fornecem o período espacial fundamental dos padrões e a eliminação de quaisquer componentes periódicos através de filtros deixando os elementos não periódicos na imagem, que podem ser descritos por técnicas estatísticas.

Além da abordagem através dos espectros de Fourier existem pesquisas que indicam o uso de filtros de Gabor [Tuceryan and Jain 1993] e modelos de *Wavelets* [Ma and Manjunath 1995]. Estudos mostram que esses sistemas de textura utilizando Gabor e multiresolução simultânea apresentam-se como uma boa técnica em sistemas de recuperação de imagens baseada em conteúdo [Ma and Manjunath 1995] [Picard and Minka 1995]. A diferença entre a análise multiresolução *Wavelets* para a análise dos espectros de Fourier é que a análise em *Wavelets* não é feita de acordo com a frequência e sim segundo a escala. Os algoritmos *Wavelets* processam dados em diferentes escalas e resoluções, possibilitando uma visão tanto global da imagem quanto os detalhes da mesma. Ainda há uma grande discussão a respeito de que família *Wavelets* utilizar para a análise de textura visto que algumas *Wavelets* obtêm melhores resultados na análise de texturas específicas. Outras objeções como a complexidade computacional também interferem nessa decisão. Esses dois importantes descritores serão melhor descritos a seguir:

***Características da Filtragem* Gabor** O filtro *Gabor* é bastante usado para extrair características de imagem, especialmente de textura. Existe muitas propostas de caracterização de textura com base na filtragem *Gabor*, no entanto estas seguem a mesma idéia básica.

Uma função *Gabor* bidimensional é definida como:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{3.30}$$

onde σ_x e σ_y são os desvios padrões dos núcleos Gaussianos ao longo das direções x e y. Então um conjunto de filtros *Gabor* podem ser obtidas por dilatações e rotações de $g(x,y)$:

$$\begin{aligned} g_{mn}(x,y) &= a^{-m}g(x',y') \\ x' &= a^{-m}(x\cos\theta + y\sin\theta) \\ y' &= a^{-m}(-x\sin\theta + y\cos\theta) \end{aligned} \tag{3.30}$$

onde $a > 1$, $\theta = n\pi/K$, $n = 0, 1, \ldots, K-1$, e $m = 0, 1, \ldots, S-1$. K e S são os números de orientações e escalas.

Dada uma imagem $i(x, y)$, sua transformada de Gabor é definida como:

$$W_{mn}(x, y) = \int i(x, y) g^*_{mn}(x - x_1, y - y_1) dx_1 dy_1 \tag{3.30}$$

onde $*$ indica o complexo conjugado. Então a média μ_{mn} e o desvio padrão σ_{mn} da magnitude de $W_{mn}(x, y)$, isto é, $f = [\mu_{00}, \sigma_{00}, \ldots, \mu_{mn}, \sigma_{mn}, \ldots, \mu_{S-1K-1}, \sigma_{S-1K-1}]$ podem ser usados para representar as características de textura de uma imagem.

***Características da Transformada* Wavelet** Similar à filtragem Gabor, a transformada *wavelet* [Mallat 1992] produz uma abordagem multi-resolução para a análise e classificação de textura. A transformada *wavelet* decompõe um sinal com uma familia de funções bases $\Psi_{mn}(x)$ obtidas através de translações e rotações de uma *wavelet mãe* $\Psi(x)$, isto é,

$$\Psi_{mn}(x) = 2^{-m/2} \Psi(2^{-m} x - n) \tag{3.31}$$

onde m e n são os parâmetros de dilatação e translação, respectivamente. Um sinal $f(x)$ pode ser representado como:

$$f(x) = \sum_{m,n} c_{mn} \Psi_{mn}(x) \tag{3.32}$$

A computação da transformada *wavelet* de um sinal 2D envolve filtragem recursiva e sub-amostragem. Em cada nível, o sinal é decomposto em quatro sub-bandas de frequência, LL, LH, HL e HH, onde L denota frequência baixa e H denota frequência alta. Os dois tipos principais de transformada *wavelet* usadas para análise de textura são a transformada *wavelet* estruturada em pirâmide (*pyramid-structured wavelet transform* - PWT) e a transformada *wavelet* estruturada em árvore (*tree-structured wavelet transform* - TWT). A PWT decompõe recursivamente a banda LL. Contudo, para algumas texturas a informação mais importante frequentemente aparece nos canais de média frequência. Para superar este problema, a TWT decompõe outras bandas tais como LH, HL ou HH quando necessário.

Depois da decomposição, vetores de características podem ser construídos usando a média e o desvio padrão da distribuição de energia de cada sub-banda em cada nível.

3.7 Técnicas de Transformação, Seleção e Redução de Características

Com o avanço da tecnologia dos bancos de dados, cada vez mais computadores acumulam dados, processam dados, e fazem uso desses dados. Com isso, pesquisadores e desenvolvedores necessitam utilizar esses dados de forma eficaz, principalmente nas áreas de exploração, análise e visualização de dados. A necessidade de analisar grandes bancos de dados é o problema fundamental da redução de dados: Como descobrir representações compactas de dados n-dimensionais? Como julgar a similaridade? As representações mentais que os seres humanos possuem do mundo são formadas pelo processamento de um grande número de entradas sensoriais, por exemplo, a intensidade dos *pixels* das imagens, o poder espectral dos sons, os ângulos de junção dos corpos articulados. Enquanto estímulos complexos desta forma podem ser representados em um vetor de espaço altamente dimensional, eles podem tipicamente possuir uma descrição muito mais compacta [Roweis and Saul 2000]. Apesar de muitos métodos de aprendizado tentarem selecionar, extrair, construir características, tanto a análise quanto os estudos experimentais indicam que muitos algoritmos escalam pobremente os domínios com um grande número de informações irrelevantes e redundantes [Liu and Motoda 1998]. Em alguns casos estas características sofrem do problema da dimensionalidade, ou seja, o número de amostras de treinamento exigido para que um classificador, por exemplo, tenha um desempenho satisfatório é dado por uma função exponencial da dimensão do espaço de características, portanto, se esse espaço de características for muito grande o número de amostras deverá ser exponencialmente maior, o que torna a solução do problema inviável [Jain et al. 2000].

A tarefa de identificar quais características funcionarão eficazmente para um algoritmo representar uma variável em um espaço se torna um aspecto importante e possui um impacto considerável na eficácia dos resultados em sistemas de classificação e recuperação [Raymer et al. 2000]. Uma escolha ótima pode não ser intuitiva e não existe uma regra definida para escolher quais recursos irão proporcionar uma discriminação entre classes. Características que funcionam pobremente separadas podem ocasionalmente trabalhar bem se pareadas com outras características. No entanto, medir e representar todas as possíveis combinações de caracaterístas pode se tornar um trabalho dispendioso e inviável [Raymer et al. 2000].

Como resultado disso, métodos de extração, seleção e transformação de características têm se tornado importantes ferramentas para o reconhecimento de padrões, análise exploratória de dados e recuperação de informação e imagens.

Como foi dito no início deste capítulo, a extração de característas é um processo em que um objeto do mundo real (seja uma imagem, um som, um modelo, etc) é representado numericamente em um conjunto de dados que resume e abstrai o conhecimento necessário para a descrição dos atributos do objeto para um devido fim (seja a sua classificação, a sua comparação com outros objetos ou a sua descrição pura).

A transformação de características é um processo onde um novo conjunto de características é criado com base em um outro vetor de características. Este processo pode ser realizado através de um mapeamento linear ou não linear das características, e o conjunto resultante pode estar em outro espaço dimensional.

Seja um conjunto de características $A_1, A_2, ..., A_n$ a transformação de características é um

processo que irá extrair um conjunto de novas características $B_1, B_2, ..., B_m$ $(m < n)$, onde $B_i = F_i(A_1, A_2, ..., A_n)$ e F é uma função de mapeamento [Liu and Motoda 1998]. Um exemplo de aplicação seria, $B_1 = c_1A_1 + c2A_2$ onde c_1 e c_2 seriam coeficientes quaisquer.

A extração e a transformação de características podem ser vistas, sob outro ponto de vista, como sendo de mesma natureza, pois as técnicas de extração, assim como as funções de transformação, utilizam como base um conjunto de dados e transformam esses dados em um novo conjunto de características. Por exemplo, uma imagem pode ser vista como um conjunto de dados e um histograma de cores pode ser visto como um descritor. Se tomarmos por base que a mesma imagem pode ser considerada como um conjunto de características (disposta em uma matriz) o mesmo descritor (histograma de cores) pode ser visto como uma função de transformação de características.

Além disso existe também a seleção de características que não irá criar nenhuma nova característica, mas irá formar um novo conjunto com as características selecionadas do conjunto original reduzindo o espaço de características original. A seleção e a transformação de características não são áreas totalmente independentes, elas podem ser vistas como dois lados do problema de representação. Considere um conjunto de características como uma linguagem de representação. Em alguns casos onde a linguagem possui mais características que o necessário a seleção de características pode ajudar na simplificação desta linguagem. Em outros casos onde a linguagem não é suficiente para descrever o problema, a transformação de características pode enriquecer a linguagem, clareando os pontos em que a mesma é falha. Onde e como essas duas ferramentas podem ser usadas irá depender da aplicação.

As técnicas de seleção de características podem ser divididas em três áreas principais, de acordo com o tipo de interação com algoritmos de aprendizado de máquina: *wrapper*, *embedded* e filtro [Blum and Langley 1997]. Os métodos *embedded* são baseados em árvores de decisão e realizam uma busca gulosa nessas árvores, utilizando uma função de avaliação que irá selecionar as características de maior discriminação entre as classes [Blum and Langley 1997]. O método *wrapper* é baseado em ferramentas de aprendizado de máquina e tem como objetivo encontrar um subconjunto de características baseadas em uma população inicial, avaliando o resultado obtido pela "caixa preta" e utilizando o resultado para selecionar um novo conjunto de características [Blum and Langley 1997]. O modelo de filtro tem como princípio básico selecionar caracteríticas levando em consideração as propriedades próprias do conjunto de dados, iniciando com um estado pré-determinado, percorrendo o espaço de busca e avaliando cada subconjunto de características utilizando um critério de avaliação específico [Blum and Langley 1997]. Esses métodos não serão aprofundados nesta seção por não fazerem parte do escopo desse trabalho, mais informações sobre técnicas de seleção de caracteríticas podem ser encontradas em [Blum and Langley 1997], [Liu and Yu 2005] e [Dy and Brodley 2004].

Devido à importância da seleção e tranformação de características no reconhecimento de padrões e pelo fato de não haver regras bem definidas para que essa tarefa funcione bem em todos os contextos, uma enorme gama de algoritmos e funções vem sendo propostos. Porém, não existe uma técnica bem conhecida na literatura de reconhecimento de padrões que combine transformação de características e teoria da informação e que seja genérica o suficiente para lidar com questões de diversas áreas do conhecimento. Assim, nas seções seguintes, algumas técnicas de transformação mais utilizadas na literatura serão explanadas.

3.7.1 Transformações Lineares

Esses algoritmos são projetados para operar quando o espaço é fixo de forma quase linear no espaço de alta dimensão [Ashutosh Saxena and Mukerjee 2004] e podem ser implementados pela manipulação de matrizes. A seguir alguns métodos de transformação linear serão discutidos.

Análise dos Componentes Principais - PCA A análise dos componentes principais ou *Principal Component Analysis* (PCA) envolve uma transformação ortogonal linear que escolhe um novo sistema de coordenadas para o conjunto de dados [Errity and McKenna 2007]. Em essência, a PCA procura reduzir a dimensão dos dados pela procura de algumas combinações lineares ortogonais (os componentes principais) das variáveis originais com a maior variância. A primeira componente principal (PC) é a combinação linear com a maior variância, a segunda PC é a combinação com a segunda maior variância e assim por diante [Fodor 2002]. O PCA pode ser calculado computando os autovalores e autovetores da matriz de covariância de um conjunto de dados. Os autovetores com os maiores autovalores correspondem às dimensões que possuem a mais poderosa correlação no banco de dados. O banco de dados D-dimensional é então projetado sobre os d autovetores com os maiores autovalores produzindo um conjunto de dados d-dimensional, onde $d < D$ [Errity and McKenna 2007]. A figura 3.24 apresenta um exemplo com um conjunto de valores representados em duas dimensões onde f'_1 está na dimensão onde a variância dos dados é máxima, f'_2 está em uma distância ortogonal onde a variância remanescente é máxima e assim por diante [Cunningham 2007].

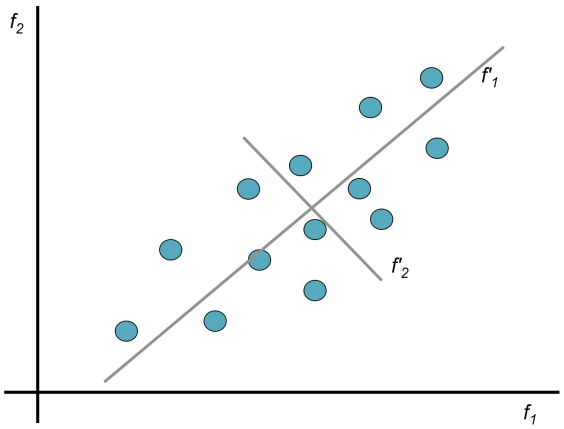


Figura 3.24: Neste exemplo de PCA em 2 dimensões o espaço de características é transformado para f'_1 e f'_2 em que a variância na direção de f'_1 é máxima [Cunningham 2007].

Análise dos Fatores Principais - FPA A análise de fatores principais ou *Principal Factor Analysis* (FPA) é um método linear usado para descrever a variabilidade sobre variáveis observadas em termos de algumas poucas variáveis não observadas chamadas fatores. As variáveis observadas são modeladas como combinações lineares dos fatores mais os termos de erro. A informação obtida através das interdependências pode ser usada depois para

reduzir o conjunto de variáveis no banco de dados. A análise dos fatores principais pode ser relacionada com a análise dos componentes principais se os "erros" no modelo de análise dos fatores forem atribuídos como todos possuindo a mesma variância [Fodor 2002].

Análise dos Componentes Independentes - ICA A análise dos componentes independentes ou *Independent Component Analysis* (ICA) tem como objetivo reduzir a redundância dos componentes dos vetores de características originais por transformações lineares, mas não necessariamente ortogonais. Enquanto o PCA remove as dependências de segunda ordem, o ICA remove também dependências de altas ordens (minimizando a informação mutual entre os componentes dos vetores de características). O modelo de dados do ICA linear é dado pela seguinte expressão:

$$x = As,$$

onde x é o vetor de características original, s é um vetor de valores aleatórios (independentes) e A é a matriz de valores combinados. Somente x é observável, e o objetivo é estimar tanto A quanto s procurando encontrar um conjunto s que seja estatísticamente independente. Existem vários modelos de geração para encontrar esse conjunto e esses modelos somente serão válidos se as componentes de s forem não gaussianas, esta é a principal diferença entre o ICA e o FPA [Comon 1994].

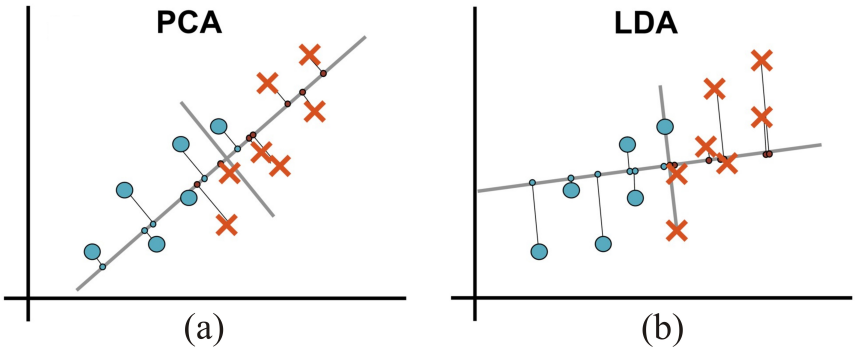


Figura 3.25: Em (a) pode-se observar que o PCA não provê uma boa separação quando os dados são divididos em classes. Em (b) o LDA procura a projeção que maximiza a separação dos dados [Cunningham 2007].

Análise de Discriminantes Lineares - LDA A análise de discriminantes lineares ou *Linear Discriminant Analysis* (LDA) é um método baseado em aprendizado supervisionado de máquina que tenta encontrar a combinação linear de um conjunto de características que melhor separa duas ou mais classes de objetos. A combinação dos resultados pode ser utilizada como um classificador linear ou, mais comumente, como ferramenta de redução de dimensionalidade. Em situações onde os rótulos das classes estão disponíveis é mais interessante descobrir uma transformação que enfatize a separação dos dados ao invés de descobrir dimensões que maximizem a variância dos dados como no PCA [Cunningham 2007]. Essa distinção é ilustrada na figura 3.25. Em um cenário 2D, o PCA projeta os dados em uma única dimensão que maximiza a variância, no entanto,

as duas classes não são separadas nessa dimensão. Em contraste, o LDA descobre uma projeção em que as duas classes são melhor separadas. Uma melhor explanação a respeito de como se encontrar essa projeção pode ser encontrada em [Cunningham 2007].

Além das técnicas apresentadas nesta seção, existem outros algoritmos, alguns derivados dos apresentados aqui que também trabalham com a transformação linear de características. Maiores informações sobre esses e outros algoritmos de transformação de características podem ser encontradas em [Errity and McKenna 2007], [Cunningham 2007], [Fodor 2002] e [Comon 1994].

3.7.2 Transformações Não-Lineares

Os métodos de trasnformação não-linear podem ser classificados de forma geral em dois grupos: aqueles que realmente provêm um mapeamento (tanto para o espaço de alta dimensão quanto para o espaço de baixa dimensão), e aqueles que apenas provêm uma visualização, como por exemplo na projeção de superfícies de três dimensões em duas. Alguns exemplos de superfícies muito utilizadas para testar esses métodos são mostrados na figura 3.26. Tipicamente aqueles métodos que apenas provêm um visualização são baseados na proximidade dos dados, ou seja, são baseados em medidas de distância, mas nada impede que esse conceito seja extendido para vetores de características de tamanhos maiores. Outros métodos são relacionados com os métodos lineares apresentados na seção anterior os quais possuem algumas diferenças que os caracterizam como sendo transformações não-lineares. A seguir alguns métodos principais utilizados na literatura serão discutidos.

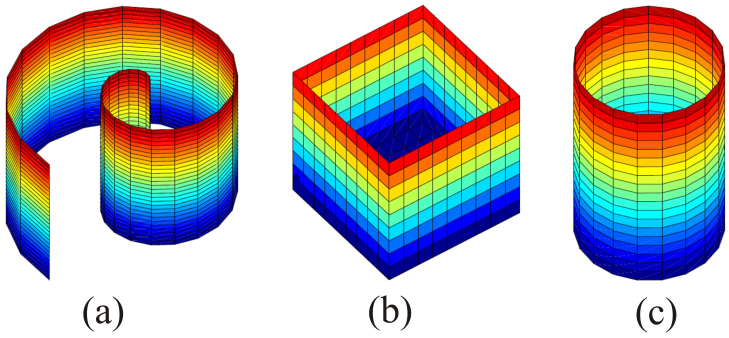


Figura 3.26: Exemplo de três superfícies (*manifolds*) típicas: (a) o rolo suíço (*swiss roll*), (b) o cubo aberto (*open box*) e (c) o cilindro (*cylinder*). [Cunningham 2007] [Lee et al. 2002].

Isomapas A técnica de transformação por isomapas ou *Isomaps* é uma técnica não supervisionada que não se limita a projeções lineares. Isso se deve ao fato de a mesma não fazer uso da distância Euclidiana, por exemplo, e utilizar outras métricas como a distância geodésica. Técnicas de projeção linear encontram dificuldades em projetar estruturas não lineares como por exemplo o espiral ilustrado na figura 3.27. O espiral está em um espaço bidimensional, mas sua dimensão intrínseca não passa de uma: somente um parâmetro é suficiente para descrever o espiral [Lee et al. 2002]. Infelizmente, a projeção de duas

dimensões para uma não é uma tarefa trivial pois a espiral necessita ser "desenrolada" em uma linha reta. Esse desdobramento é difícil para técnicas lineares pois o par distância Euclideana após projeção projeta um "atalho" entre os dois pontos como na figura 3.27 (b), no entanto, o que deve ser feito é utilizar uma medida de distância ao longo da espiral como na figura 3.27 (c).

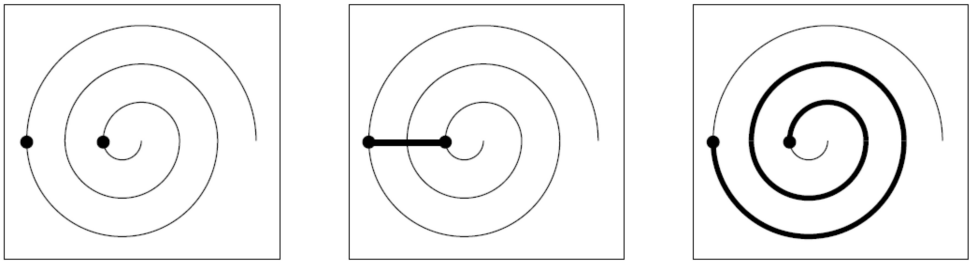


Figura 3.27: (a) dois pontos em um espiral; (b) distância Euclidiana entre os dois mesmos pontos; (c) distância curvilínea ou geodésica [Lee et al. 2002].

Essa distância, também chamada de distância geodésica é aproximada pelo isomapa da seguinte maneira: Primeiro a vizinhança de cada ponto é calculada. Por exemplo, a vizinhança de um ponto poderá ser os k pontos mais próximos ou o conjunto de pontos inclusos em um círculo de raio ϵ (onde k e ϵ são constantes pré-determinadas). Uma vez que os vizinhos mais próximos são conhecidos, um grafo é construído pela ligação entre todos os pontos de vizinhança (onde cada ponto será um vértice do grafo). A seguir, cada aresta desse grafo é rotulada pela distância entre os seus vértices (pontos de ligação). Finalmente a distância geodésica entre dois pontos é aproximada pela soma dos tamanhos das arestas sobre a menor distância de ligação entre esses dois pontos [Lee et al. 2002]. Vários são os algoritmos que calculam esse menor caminho em grafos como por exemplo o algoritmo de Dijkstra [Dijkstra 1959].

Após descobrir a distância geodésica entre todos pontos uma matriz de distâncias é criada, então os autovalores e autovetores dessa matriz são encontrados e os p maiores autovalores darão as coordenadas dos pontos de referência na projeção do espaço p-dimensional.

Locally Linear Embedding (LLE) Apesar do nome citar a palavra linear o algoritmo *Locally Linear Embedding* (LLE) [Roweis and Saul 2000] é uma técnica que utiliza métodos não lineares baseados em intuições geométricas. O nome *Locally Linear Embedding* se deve, então, ao fato que os pontos representados em uma superfície e suas vizinhanças estão em uma malha em que a geometria local é caracterizada por coeficientes lineares que constroem o ponto. Suponha que o conjunto de dados consista de N vetores de valores reais X_i, cada um com dimensionalidade n, amostrados de alguma superfície e seja Y_i as coordenadas internas globais da superfície (coordenadas no espaço de baixa dimensão). O algoritmo é realizado da seguinte maneira: Primeiro os vizinhos mais próximos a cada ponto X_i são calculados (como mostrado na sub-seção anterior). Após isso, os pesos W_{ij} que melhor reconstroem linearmente X_i a partir de seus vizinhos é calculado. Finalmente,

os vetores de baixa dimensão Y_i são calculados pelo resultado da reconstrução de X_i por W_{ij} [Roweis and Saul 2000]. A Figura 3.28 mostra um exemplo da aplicação do LLE em uma superfície.

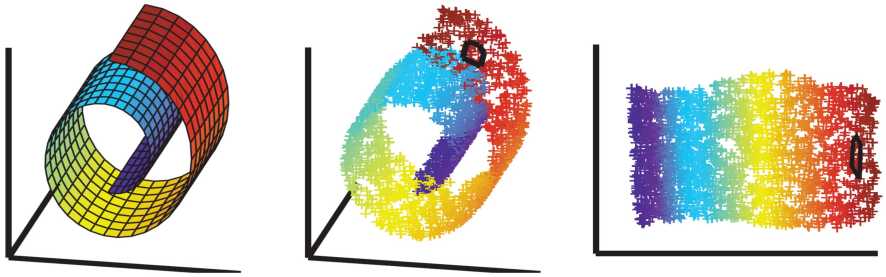


Figura 3.28: Aplicação do método de transformação LLE: (a) superfície tri-dimensional (rolo suiço), (b) amostra dos dados da superfície, (c) resultado da projeção da superfície utilizando o LLE [Roweis and Saul 2000].

Redes Neurais (Auto-Encoders) As redes neurais artificiais são ferramentas estatísticas poderosas para a representação de mapeamentos de várias variáveis de entrada para várias variáveis de saída. Elas podem ser vistas como circuitos com unidades de alta interconectividade com pesos de conexões ajustáveis. Para melhor entendimento desta seção, é sugerido que se leia os próximos dois capítulos onde é explicado passo a passo a concepção e o funcionamento de uma rede neural.

Para a implementação desta técnica é utilizada uma arquitetura de rede particular chamada rede auto-associativa onde o padrão de saída da rede neural deve ser idêntico ao padrão de entrada, o que significa que a primeira e a ultima camada possuirão o mesmo número de neurônios [Golinval 2004]. O objetivo dessa rede é organizar uma codificação/compressão dos dados entre a camada de entrada e a camada do meio da rede e fazer uma descompressão/decodificação dos dados entre a camada do meio e a camada de saída da rede. Para isso a camada do meio deverá possuir um número menor de neurônios que as camadas de entrada/saída. A Figura 3.29 ilustra o uso das redes neurais autoassociativas para o mapeamento não linear. O número de camadas entre a camada do meio e a camada de entrada e entre a camada do meio e a camada de saída irá depender da aplicação.

O algoritmo de treinamento e uso das redes autoaussociativas para o mapeamento não linear de características funciona da seguinte forma: Um conjunto de dados de treinamento é selecionado onde cada objeto é representado por um vetor. Cada vetor é então inserido na rede e irá gerar um vetor de saída. A rede é então treinada, retificando seus pesos, até que o vetor de entrada seja igual ao vetor de saída. Quando isso ocorrer, o vetor é inserido novamente na rede neural e são colhidos da camada do meio, os valores resultantes da camada de codificação. Esses valores, serão, deste modo, a nova representação do objeto em questão.

Foi comprovado por [Baldi and Hornik 1989] que, quando a camada do meio for menor

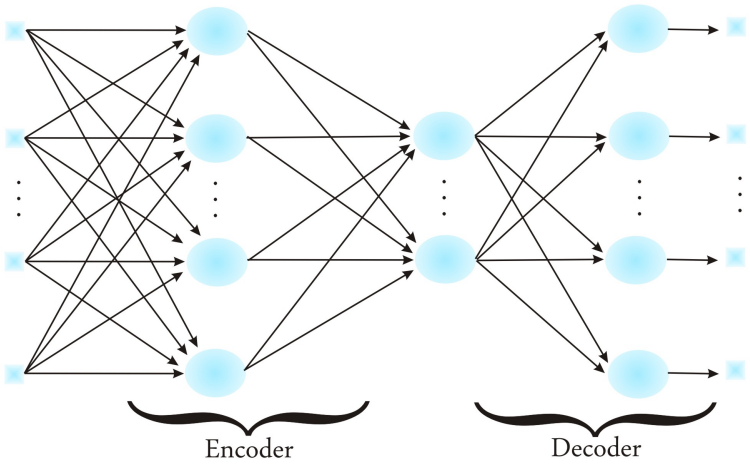


Figura 3.29: Rede neural autoassociativa com suas funções de projeção (encoder) e expansão (decoder).

> que as camadas de entrada/saída e forem usadas somente funções de ativação lineares, a rede neural irá atuar como um compressor do tipo PCA. Por isso alguns autores costumam chamar esse tipo de transformação (quando utilizados outros tipos de funções de ativação) de Análise de Componentes Principiais não Linear ou *nonlinear PCA*.

O modelo proposto por esse trabalho se baseia no método de transformações de características por redes neurais, mas utiliza redes heteroassociativas fazendo com que o mapeamento, utilize como suporte, exemplos, para reforçar as conexões onde as características sobressaem às outras. Esse mapeamento não-linear se baseia no aprendizado neural feito através de exemplos pré-definidos. Tais exemplos servirão como suporte para dar à transformação o caráter subjetivo do ser humano na análise das imagens.

Capítulo 4

Redes Neurais - Inspiração Biológica

4.1 Introdução

Paul Broca (1824 -1880) fez amplos estudos no que diz respeito à afasia, uma espécie de deficiência da fala em pacientes com danos cerebrais e descobriu que o cérebro é o responsável fundamental pelo trabalho de cada órgão. Nesse momento Paul descobriu que o cérebro era formado por células nervosas, também chamadas de neurônios. O estudo do cérebro não se limitou apenas na formação biológica do mesmo mas também nos fenômenos causados por ele. Isso fez com que surgissem várias áreas relacionadas à aplicação do entendimento para saber e compreender a mente humana entre elas a Neurociência e Psicologia.

Hermann Von Helmontz (1821 -1894) em parceria com seu aluno Wilhelm Wond (1832 - 1921) publicou em seu livro "Handbook of Physiological Optcs" vários estudos sobre a visão humana. Este produto é considerado um dos primeiros indícios da origem da Psicologia Científica. Alguns biólogos da época estavam interessados apenas em pesquisar fisicamente os animais sem antes estudar a parte introspectiva dos mesmos desenvolvendo uma metodologia objetiva para seus comportamentos. O chamado Movimento Behaviorista comandado por John Watson se preocupava em estudar somente as medidas objetivas das percepções ou estímulos pertinentes a um animal e suas reações ou respostas. Esse movimento obteve um grande avanço com ratos e pombos, mas quase nada esclareceu a respeito das construções mentais mais complexas advindas dos seres humanos. Mesmo assim, exerceu uma forte influência sobre a psicologia nos Estados Unidos entre 1920 e 1960 [Russel and Norvig 2004].

A psicologia cognitiva foi um grande marco no entendimento do processamento de informações através do cérebro. William James (1842 - 1910) realizou alguns trabalhos sobre a percepção humana como uma forma de inferência lógica inconsciente, caracterizando o ponto de vista cognitivo. Como se era de esperar, vários cientistas da época como Frederic Bartlett (1886 - 1969) e seu também aluno Kenneth Craick (1943) bateram de frente ao Behaviorismo tentando provar que os termos "mentais" como as convicções e metas eram tão científicos quanto os fenômenos físicos como os gases, por exemplo. Por volta de 1958 nos Estado Unidos, o crescimento da modelagem e a criação de novos computadores induziram a concepção do campo da ciência cognitiva. Essa ciência mostra que os modelos computacionais podem servir de base para tratar a psicologia da memória, a linguagem e o pensamento lógico. Além disso, a

caracterização do comportamento social humano e dos mecanismos que governam atitudes pró-sociais e anti-sociais é uma tarefa desafiadora que requer pesquisa interdisciplinar, incluindo os campos da neurociência, psicologia, neurologia comportamental, psiquiatria, antropologia, sociologia e biologia evolutiva, entre outros. O comportamento social humano é estruturado com base em interações complexas entre fatores neurobiológicos, culturais e econômicos.

Atualmente para se modelar um sistema inteligente, pode-se basear em dois paradigmas fundamentais: O conexionismo e o simbolismo nascidos em 1956 no "Darthmouth College". A Inteligência Artificial simbólica tenta simular o comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal. Já a Inteligência Artificial Conexionista acredita que construindo-se um sistema que simule a estrutura do cérebro, este sistema apresentará inteligência, ou seja, será capaz de aprender, assimilar, errar e aprender com seus erros. Essa ultima será melhor abordada ao longo desse capítulo por tratar-se do assunto primordial deste trabalho. Alguns autores divergem em relação a essa divisão, pois novas áreas da Inteligência Artificial têm sido exploradas como a Inteligência Artificial Evolutiva ou Computação Evolutiva que começa surgiu no final dos anos 60, impulsionada pelos estudos de John Holland que começou a investigar a possibilidade de incorporar os mecanismos naturais de seleção e sobrevivência para a resolução de problemas de inteligência Artificial, os quais já tinham encontrado solução na natureza mas não apresentavam uma abordagem satisfatória em sistemas computacionais.

4.2 Modelos Neuronais Biológicos

A mente humana é um sistema cognitivo composto por unidades biológicas (neurônios) interconectados entre si por uma rede de sinapses cooperativas para processar a informação de forma distribuída. Com isso os estados e fenômenos mentais descritos anteriormente surgem como resultado da cooperação global da atividade distribuída de células neuronais no cérebro [Hertz and Krogh 1991].

O conhecimento das áreas mais específicas do pensamento ainda é um campo muito sombrio, o que pode caracterizar as manifestações do pensamento de forma simplificada. A mente humana pode realizar fenônemos e atividades diversas, estudar e compreender esses acontecimentos pode se tornar um grande desafio. Assim como a criatividade, o delírio e o pensamento desorganizado, a bagagem emocional, a memória e os padrões pré-definidos pela mente humana dentre outros formam o sistema nervoso. É no sistema nervoso que as escolhas são feitas e será baseado nesses fenômenos cognitivos formados por esse sistema que o modelo proposto por esse trabalho se desenvolverá.

4.3 O Sistema Nervoso

Camilo Golgi foi um dos precursores do estudo do sistema nervoso. Em 1875 criou um método para a análise e estudo de um neurônio solitariamente através da coloração do mesmo. Essa experiência deu um grande impulso para vários estudos na área mas ainda não revelava o obscuro campo das muitas redes de estruturas neuronais [Paradiso 2002].

O estudioso espanhol Santiago Ramón y Cajal, seguindo a sombra de Golgi adotou a noção de sistema nervoso falando sobre a interligação das células por estímulos nervosos, também chamados de sinapses. Descreveu também várias estruturas neuronais clareando a idéia de que essas células trabalhavam em conjunto, cada uma com suas especialidades para uma estrutura global de propósitos mais gerais.

Pode-se concluir, a partir disso, que o estudo do cérebro humano foi e ainda é um dos grandes desafios para os cientistas. Por volta de 335 a.C. Aristóteles disse: "De todos os animais, o homem é o que possui o maior cérebro em proporção ao seu tamanho" (hoje, sabe-se que algumas espécies de golfinhos e baleias têm cérebros relativamente maiores). Naquela época acreditava-se que o sentimento íntimo que avisa o que se passa pelo ser humano, dando a ele conhecimento de suas ações estava localizado em outros órgãos do corpo humano como o coração. Apenas em meados do século XVIII o cérebro foi amplamente conhecido como sede da consciência [Russel and Norvig 2004].

O cérebro pode ser considerado como um dos principais componentes do corpo humano pois rege várias funções deste como o controle muscular, a velocidade e o equilíbrio de vários órgãos dentre outros através de um sistema complexo e muito estudado chamado sistema nervoso. Ele é formado basicamente por dois tipos de células: os neurônios e as células neuróglias ou simplismente glia (grego para "cola"). A função da glia ainda é um campo de muito estudo pois há muito a ser aprendido e descoberto sobre ela. Alguns cientistas sustentam a hipótese que a glia tem a função de somente dar sustentação aos neurônios e auxiliar o seu funcionamento [Paradiso 2002].

4.3.1 O Neurônio

Existem aproximadamente 10 000 000 000 a 100 000 000 000 neurônios no cérebro humano. Eles podem ser de diversos tipos. Na Figura 4.1 apresenta-se o modelo simplificado de um único neurônio real. O neurônio possui um corpo celular chamado soma constituído de núcleo e pericário, que dá suporte metabólico à toda célula.

Os dendritos são ramificações nervosas (arborizações terminais) que emergem do pericário e do final do axônio e conduzem sinais elétricos das extremidades para o corpo celular, sendo, na maioria das vezes, responsáveis pela comunicação entre os neurônios através das sinapses. O axônio é uma ramificação maior (fibra nervosa) e única que aparece no soma. É responsável pela condução do impulso nervoso para o próximo neurônio, podendo ser revestido ou não por mielina (bainha axonial) ou por células gliais especializadas. As extremidades do axônio são conectadas com os dentritos de outros neurônios caracterizando a sinapse.

4.3.2 As Sinapes

Algumas substâncias geradas em determinadas regiões do cérebro e liberadas em áreas distantes e separadas caracterizam a modulação dopaminérgica, uma espécie de ativação da sinapse. Como algumas dessas substâncias pode-se citar as catecolaminas, a norepinefrina, epinefrina e a dopamina. Elas não agem nos canais iônicos das células, mas ativam os mensageiros dentro das mesmas durante mais tempo que as outras substâncias.

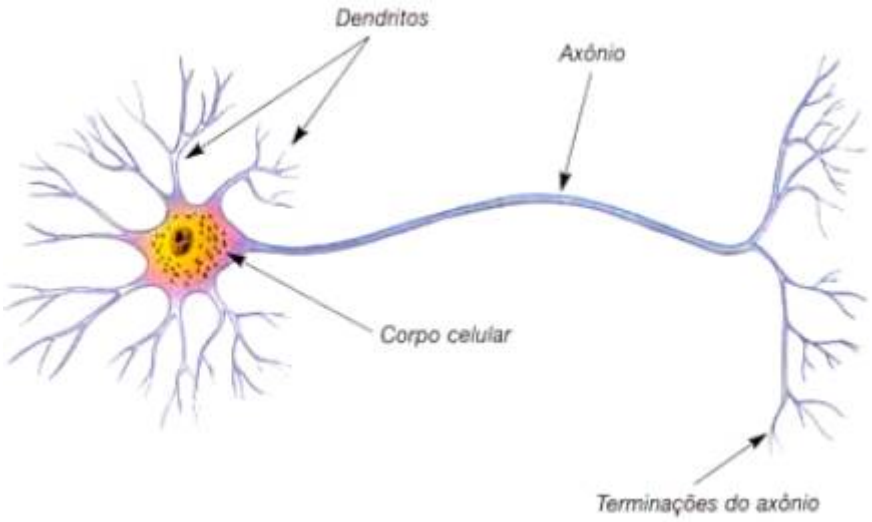


Figura 4.1: Representação de um Neurônio.

Uma vez que essas moléculas do neurotransmissor são liberadas de uma célula como resultado do disparo de um potencial de ação, elas se ligam a receptores específicos na superfície da célula pós-sináptica. A partir disso os neuroreguladores insinuam uma função reguladora que modula as características operacionais dos neurônios receptores, isto em suas respostas aos neurotransmissores.

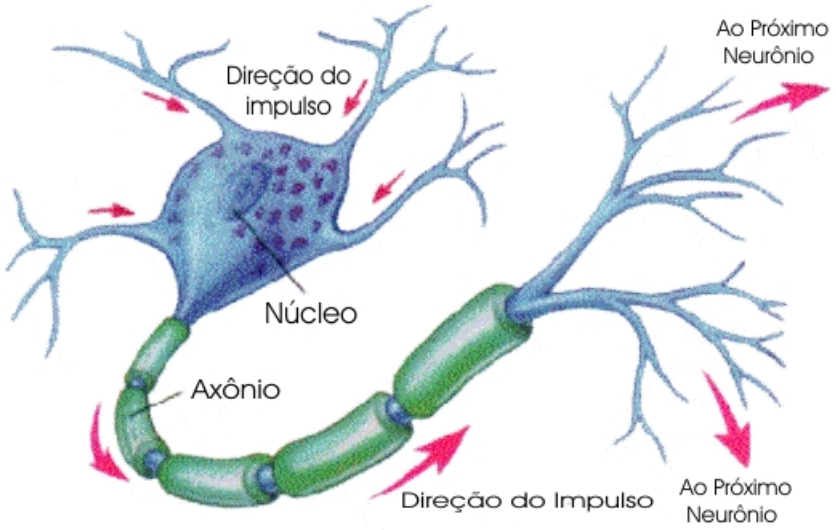


Figura 4.2: Diagrama de um neurônio e seus impulsos nervosos.

Potencial de Ação

Existem dentro e fora das células diferentes concentrações de Na^+ e K^-. Isso pode ocasionar variações muito rápidas de voltagem de repouso da célula. Quando há alguma perturbação da

membrana da célula irá ocorrer esse fenômeno de diferença de potencial (uma sucessão de eventos fisiológicos que ocorrem através da membrana). Tais fenômenos importantes para o funcionamento de um neurônio, em conjunto, produzem aquilo que se pode chamar Potencial de Ação. Este potencial pode ser ocasionado por uma estimulação química, elétrica, de calor, etc.

O potencial a ser gerado age inicialmente em uma extremidade no axônio e se propaga em uma única direção, não retornando pelo caminho já percorrido. Consequentemente, os potenciais de ação são unidirecionais - ao que pode se chamar de condução ortodrômica. Com essa propagação a sinapse é gerada para a liberação de moléculas e neurotransmissores. Esses neurotransmissores colam na membrana de outro neurônio desencadeando o mesmo processo dando continuação ao fenômeno.

4.4 Representação dos Modelos Neuronais

Por volta do Século XIX dois cientistas chamados Helmholtz e Mach estudaram muitos fenômenos relacionados à visão humana. Estavam interessados no estudo da ilusão de ótica, pois propunham a conjectura de que na retina humana as células são excitadas pela luz que converge para uma região central de onde se diverge para as demais regiões periféricas.

Após um século foi descoberto, através de uma espécie de caramujo chamado Limulos [Hartline 1957] que alguns vertebrados possuem uma estrutura chamada "on-center/off-surround", onde um neurônio se comunica com seus vizinhos através de excitações imediatas. Isso leva a uma certa competição entre algumas espécies de neurônio que se localizam fora dessa vizinhança e é essencial para o neurodesenvolvimento dessas células. Este processo chama-se facilitação. Em 1949 o cientista Donald Hebb propõe a "Lei de Hebb" para explicar esse mecanismo, que diz: "A intensidade de uma conexão sináptica entre dois neurônios aumentam quando os dois neurônios estão excitados simultaneamente".

Algumas substâncias chamadas de forma genérica "fatores de crescimento neurais" são expelidas pelos neurônios excitados promovendo esse aumento sináptico. Isso é um fator preponderante para a seleção natural das mesmas [Paradiso 2002].

A competição citada anteriormente caracteriza a formação de uma espécie de cadeia neural denominada mapa, formada basicamente por duas folhas de neurônios chamados domínio e imagem, de modo que padrões semelhantes de ativação do domínio são projetados para os neurônios vizinhos da imagem.

O modelo de neurônio pode ser representado pela soma de todos os seus estímulos de entrada multiplicados pelos pesos das forças de sinapse de entrada. A excitação ou inibição do neurônio é decidida através de um valor limiar fornecido antecipadamente. Se a soma dos estímulos for maior que este limite pode-se dizer que o neurônio está excitado, se ocorrer o contrário, ou seja se a soma destes estímulos for menor, pode-se dizer que ele está inibido.

A transmissão de informação entre os neurônios irá depender principalmente dos neurotransmissores (abundância, sensibilidade da membrana). Logo se a intensidade que um neurônio pode inibir ou excitar outro for modificada através dos valores dessas conexões, pode-se modificar também o comportamento da rede formada por esses neurônios. Isso se chama aprendizado de rede. Todos esses conceitos são utilizados pela Ciência da Computação e mais especifica-

mente pela Inteligência Artificial para a implementação das Redes Neurais Artificiais apresentadas no próximo capítulo.

Capítulo 5

Redes Neurais Artificiais

5.1 Introdução

Atualmente acredita-se que as redes neurais são parte fundamental no processamento das idéias e pensamentos do ser humano, baseados nisso, os primeiros trabalhos de Inteligência Artificial surgiram com o objetivo de criar, implementar e utilizar de forma prática as Redes Neurais Artificiais. Esse campo da Ciência da Computação e da Inteligência Artificial também chamado de Conexionismo ou Neurocomputação é visto como um desenvolvimento recente, mas sua origem se dá nos primórdios da Ciência da Computação, psicologia e filosofia.

Já no início do século XX, Alan Turing e John Von Newman asseguraram que a inteligência podia ser representada pela matemática booleana. Esse pensamento ganhou força com os artigos de Warren McCulloch, médico, filósofo, matemático e poeta, juntamente com o estatístico Walter Pitts, publicados por volta de 1943 os quais já descreviam um modelo neuronal simplificado em termos de operações lógicas booleanas. Este modelo de neurônio booleano era intuitivamente simples, mas com base nesse modelo, McCulloch conseguiu implementar uma cadeia de funções, onde eram aplicadas às entradas um valor eventual, provocando uma soma devidamente ajustada (ou subtrações no caso de sinapses inibitórias), tendo como resultado uma única saída: pulso, quando a soma exceder um dado limiar, ou não pulso, caso contrário. Vários outros artigos e livros foram publicados desde então, por certo tempo, mas não alcançaram muitos resultados.

Em 1949, um neuropsicólogo chamado Donald Hebb afirmou através de seu livro de nome "The Organization of Behavior" (A Organização do Comportamento) a idéia de que a dependência psicológica clássica encontra-se presente em qualquer parte dos animais, pois esta é uma qualidade dos neurônios individuais. Apesar dessas idéias não serem totalmente originais, Hebb obteve grande destaque por ser o pioneiro no que diz respeito às leis de aprendizagem particulares para as sinapses dos neurônios. Este audacioso precedente serviu de influxo para que vários outros pesquisadores seguissem a mesma idéia. Ainda que muito tenha sido analisado, estudado e divulgado nos anos subsequentes (entre as décadas de 1940 e 1950), estes se consagraram mais como apoio para incremento futuro que para o próprio desenvolvimento. Também neste período de tempo o primeiro neurocomputador foi criado. Cognominado Snark, essa máquina desenvolvida por um cientista chamado Marvin Minsky, operava com êxito de acordo com um ponto

de partida técnico, adaptando seus pesos de forma automática. Apesar desse neurocomputador não convir para efetuar qualquer função de processamento de informação interessante, ele serviu de inspiração para as idéias de estruturas que o sucederam [Luger 2004] [Haykin 2001].

Por volta de 1957 foi apresentado o que hoje se considera o primeiro neurocomputador de sucesso. Chamada de Mark I Perceptron, essa máquina foi construída por Charles Wightman e Frank Rosenblatt, que devido à sagacidade de seus estudos, suas colaborações técnicas e de seu caráter moderno de pensar, é considerado por muitos como o criador da neurocomputação moderna. Seu objetivo inicial para a concepção do Perceptron era o reconhecimento de padrões [Russel and Norvig 2004]. No entanto, Minsky e Papert analisaram matematicamente o Perceptron e demostraram que redes de uma camada não são capazes de solucionar problemas que não sejam linearmente separáveis. Como não acreditavam na possibilidade de se construir um método de treinamento para redes com mais de uma camada, eles concluíram que as redes neurais seriam sempre suscetíveis a essa limitação.

Posteriormente a Rosenblatt, um professor de engenharia elétrica da Universidade de Stanford chamado Bernard Widrow, com o auxílio de alguns de seus alunos, criou um novo tipo de componente de processamento de redes neurais chamado Adaline municiado com uma intensa lei de aprendizado, que perdura até os dias atuais. Além disso, Widrow instituiu a primeira corporação de hardware de neurocomputadores e componentes [Haykin 2001].

Excepcionalmente, os anos seguintes foram abalizados por um ânimo "descomedido" de vários pesquisadores, os quais começaram a publicar artigos e livros com anseios audaciosos através de presciências inseguras para a época, dissertando sobre instrumentos tão poderosos quanto o cérebro humano que seriam concebidos em um breve espaço de tempo. Isso tirou quase toda a qualidade crível dos estudos deste campo e ocasionou execrações por parte dos técnicos de outras áreas [Haykin 2001].

Os anos entre 1967 e 1982 foram caracterizados pelas pesquisas taciturnas, pois somente alguns estudos foram divulgados por causa dos acontecimentos antecedentes. Contudo, tanto os pesquisadores daquela época quanto todos aqueles que acompanharam tais estudos no transcorrer de treze anos conseguiram, outra vez, constituir uma área palpável para o renascimento do campo das redes neurais artificiais [Luger 2004].

Entusiasmados e encorajados pelos novos experimentos, muitos pesquisadores passaram, a partir dos anos 80, a publicar várias propostas para a pesquisa e desenvolvimento de redes neurais assim como suas aplicações. Em 1983 um administrador de programas da DARPA (Defense Advanced Research Projects Agency) chamado Ira Skurnick passou a estudar a neurocomputação e em conjunto com alguns projetistas fundou o primeiro centro de pesquisas em neurocomputação. Esta ação não somente abriu as portas para a neurocomputação, como também forneceu à DARPA o status de uma das líderes mundiais em se tratando de tecnologias emergentes [Haykin 2001].

Um outro cientista que se destacou nessa mesma época foi o físico John Hopfield, que também se interessou pela neurocomputação e escreveu vários artigos, principalmente chamando a atenção para as propriedades associativas das RNA's. Tais publicações percorreram o mundo todo determinando a vontade de vários cientistas, matemáticos, e tecnólogos altamente qualificados a se unirem nesta nova área emergente [Haykin 2001] [Russel and Norvig 2004].

Não obstante a toda a influência de Hopfield sobre os novos pesquisadores, em 1986 este campo obteve uma maior adesão de novos cientistas com a publicação do livro "Parallel Dis-

tributed Processing" (Processamento Distribuído Paralelo) escrito por David Rumelhart e James McClelland. As novas ponderações de David Rumelhart, Geoffrey Hinton e James McClelland, em 1986, ocasionaram um novo julgamento sobre os perceptrons dando enfoque na sistematização de processamento da informação. Eles avaliaram que as críticas da década de 70 foram falhas de interpretação que acabaram por destruir a confiabilidade da análise do perceptron naquela época [Luger 2004] [Haykin 2001].

Finalmente, em 1987 ocorreu a primeira conferência de redes neurais em tempos modernos chamada IEEE - International Conference on Neural Networks. Nesse encontro também foi formada a primeira associação de estudiosos em redes neurais, chamada International Neural Networks Society (INNS). Diante desses acontecimentos transcorreu-se a fundação do INNS journal em 1989, acompanhado do Neural Computation e do IEEE Transactions on Neural Networks em 1990 [Luger 2004] [Haykin 2001].

Desde 1987, várias universidades proclamaram o desenvolvimento de institutos de pesquisa e programas de ensino em neurocomputação. Atualmente muitas divisões de análise da ciência procuram respostas a questões características pela reflexão da natureza, e como a mesma pode determinar a solução de tais problemas alcançando, frequentemente, condições ótimas ou quase-ótimas para os sistemas desenvolvidos, sem a necessidade de ajuda ou "intervenção" por alguma entidade controladora. Em consequência disso, vários estudos de algoritmos genéticos, busca evolucionária, autômatos celulares, redes neurais, entre outros incitáveis ramos são realizados para levar até a computação e a matemática a eficácia e o vigor da maneira auto-ajustável e a transformação progressiva da natureza biológica [Haykin 2001]. Baseado nessa eficácia da maneira auto-ajustável da natureza biológica dos neurônios, este capítulo apresenta alguns fundamentos sobre as Redes Neurais Artificiais utilizados na obtenção do modelo proposto.

5.1.1 Sumário de Notações Utilizadas

A seguinte notação será utilizada ao longo desse capítulo.

x_i y_i
: Ativações das unidades X_i e Y_j respectivamente:
Para unidades de entrada X_i;
x_i = sinal de entrada;
Para outras unidades Y_j;
$y_j = f(y\_in_j)$.

w_{ij}
: Pesos nas conexões vindas de X_i para a unidade Y_j;

b_j
: Bias na unidade Y_j.

$f(y\_in_j)$
: Entrada da rede para a unidade Y_j:
$net(y\_in_j) = b_j + \sum_i x_i w_{ij}$.

W

Matriz de pesos:
$W = w_{ij}$.

$W.j$

Vetor de Pesos:
$W_{.j} = (w_{1j}, w_{2j}, w_{3j}, ..., w_{nj})^T$;

σ_j

Função de ativação para o neurônio Y_j.

c

Vetor de entrada durante o treinamento:
$c = (f_1, f_2, f_3, ..., f_n)$.

A

Vetor de saída alvo, durante o treinamento:
$A = (a_1, a_2, a_3, ..., a_m)$.

x

Vetor de entrada durante a fase de testes:
$x = (x_1, x_2, x_3, ..., x_n)$.

5.2 Neurônio Artificial

De acordo com [Luger 2004] e [Haykin 2001] as redes neurais são compostas de nós ou unidades conectadas por vínculos orientados chamados neurônios artificiais. Um neurônio artificial (exemplificado na figura 5.1) é uma unidade de processamento de informação fundamental para uma rede neural e é basicamente formado pelos seguintes elementos:

Sinais de Entrada, x_i ou f_i (dependendo do estágio da rede). Esses dados podem ser provenientes do ambiente, ou da ativação de outros neurônios. Os valores de entrada diferem quanto ao intervalo permitido de acordo com o modelo implementado, mas geralmente as entradas são valores, do conjunto (0,1) ou (-1,1), ou números reais quaisquer.

Um *conjunto de sinapses* ou *elos de conexão*, cada uma caracterizada por um *peso* ou *força própria*. Especificamente um sinal x_i na entrada de sinapse i conectada ao neurônio j é multiplicado pelo peso sináptico w_{ij}. É importante notar a maneira com que são escritos os índices dos pesos sinápticos w_{ij}. O primeiro índice se refere ao terminal de entrada da sinapse a qual o peso se refere e o segundo índice se refere ao neurônio em questão. O peso sináptico de um neurônio artificial pode estar em um intervalo que inclui tanto valores positivos quanto negativos.

Um *somador* ou *nível de ativação* $\sum_i x_i w_{ij}$ para somar os dados de entrada, ponderados pelas respectivas sinapses do neurônio ou seja, a força cumulativa de seus sinais de entrada escalados pelo peso da conexão w_{ij} ao longo da linha de entrada.

Uma função de ativação $\sigma(.)$ é usada para restringir a amplitude da saída de um neurônio. Esta função calcula o estado final ou de saída do neurônio determinando quanto o nível de

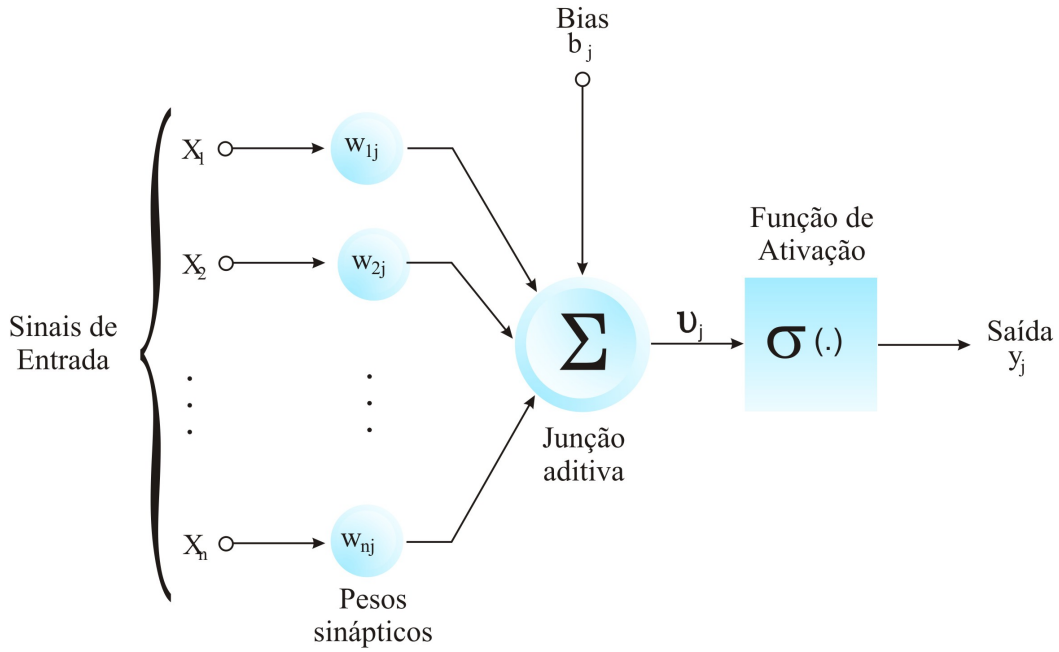


Figura 5.1: Modelo não-linear de um neurônio (adaptado de [Haykin 2001]).

ativação do neurônio está abaixo ou acima de um valor limiar. O objetivo do limiar é reproduzir o estado ativo/inativo dos neurônios biológicos. Característicamente, o intervalo normalizado da amplitude da saída de um neurônio é um intervalo unitário fechado [0,1] ou alternativamente [-1,1], no entanto, existem diversos outros modelos de funções de ativação. Três escolhas para $\sigma(.)$ são mostradas na figura 5.2: a função de limiar, a função de limiar por partes e a função sigmóide (igualmente conhecida como função logística). A função sigmóide tem a conveniência de ser diferenciável, o que facilita na fase de aprendizagem dos pesos como será mostrado nas seções posteriores. Todas as funções possuem um limiar (seja ele permanente ou temporário) em zero; o peso de desvio w_{0j} define o limite real para a unidade, no sentido de que a unidade é ativada quando a soma ponderada de entradas reais $\sum_i x_i w_{ij}$ (ou seja, excluíndo-se a entrada de desvio) excede w_{0j} [Haykin 2001] [Russel and Norvig 2004].

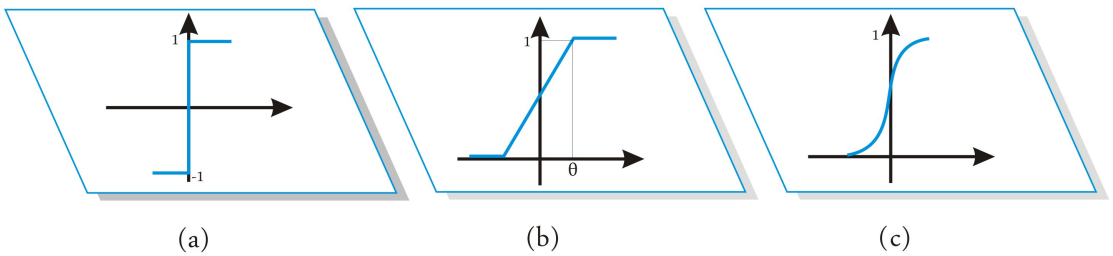


Figura 5.2: (a) Função de limiar (degrau). (b) Função de limiar por partes (rampa). (c) Função sigmóide).

O modelo neuronal da figura 5.1 inclui ainda um bias inserido de modo externo, representado

por b_j. O bias b_j tem como finalidade acrescentar ou enfraquecer a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo concomitantemente.

Em termos matemáticos, pode-se escrever um neurônio j pelo par de equações 5.0 e 5.1.

$$u_j = \sum_{i=1}^{n} x_i w_{ij} \tag{5.0}$$

$$v_j = u_j + b_j \tag{5.1}$$

onde $(x_1, x_2, ..., x_n)$ são os sinais de entrada; $(w_{1j}, w_{2j}, ..., w_{3j})$ são os pesos sinápticos do neurônio j; y_j é a saída do combinador linear devido aos sinais de entrada; b_j é o bias; $\sigma(.)$ é a função de ativação e y_j é o sinal de saída do neurônio. Como já foi dito o uso do bias b_j tem o efeito de aplicar uma tranformação afim à saída u_j do combinador linear no modelo da figura 5.1 como mostrado pela equação 5.2.

$$y_j = \sigma(u_j + b_j) \tag{5.2}$$

Em especial, de acordo com o bias b_j positivo ou negativo, a relação entre o campo local induzido ou potencial de ativação v_j do neurônio j e a saída do combinador linear u_j, é modificada na forma ilustrada na figura 5.3. Nota-se pela figura que como o resultado desta tranformação afim, o gráfico v_j em função de u_j não passa mais pela origem.

O bias b_j, é um parâmetro externo do neurônio artificial j. Pode-se considerar a sua presença como na equação 5.1. De forma similar, pode-se formular a combinação das equações 5.0 até 5.2 tal como nas equações 5.3 e 5.4 apresentadas a seguir onde $b_j = x_0$.

$$v_j = \sum_{i=0}^{n} x_i w_{ij} \tag{5.3}$$

$$y_j = \sigma(v_k) \tag{5.4}$$

Na expressão adiciona-se uma nova sinapse. A sua entrada será $x_0 = 1$ e o seu peso será $w_{0j} = b_j$. A equação 5.6 apresenta a multiplicação da matriz de pesos para o cálculo da entrada da rede para a unidade Y_j. Se os pesos de conexão de uma rede neural estão armazenados em um vetor $W = w_{ij}$, a entrada da rede para a unidade Y_j é o produto dos vetores $x = (x_0, x_1, x_2, ..., x_n)$ e $W.j$ (a j-ésima coluna da matriz de pesos).

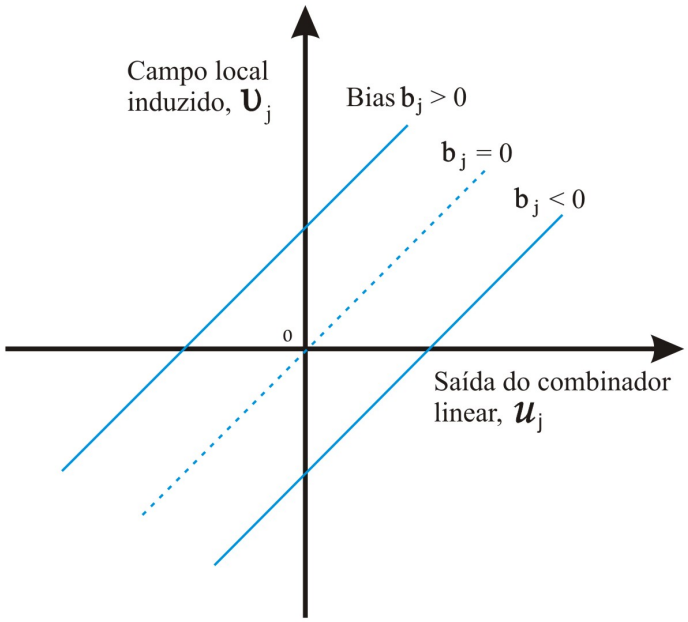


Figura 5.3: Transformação afim produzida pela presença de um bias; note que $v_j = b_j$ em $u_j = 0$.

$$y\_in_j = xW.j \tag{5.5}$$

$$= \sigma(\sum_{i=0}^{n} x_i w_{ij}) \tag{5.6}$$

Além dessas propriedades dos neurônios individuais, uma rede neural é também caracterizada por propriedades globais, tais como a topologia da rede que forma o padrão de conexões entre os neurônios individuais sendo fonte primária do viés indutivo da rede. Ou seja, o algoritmo de aprendizagem utilizado e o esquema de codificação correspondem à interpretação que se dá aos dados fornecidos para a rede e ao resultado do seu processamento. Os principais exemplos destas características serão apresentados ao longo desse capítulo.

5.3 Arquiteturas de Rede

O modo como são organizadas as conexões entre os neurônios de uma rede neural bem como os algoritmos de aprendizagem utilizados para o treinamento da mesma estão intimamente ligadas com a estrutura escolhida para a representação da mesma. Pode-se dizer então que os algoritmos ou regras de aprendizagem utilizados nos projetos de redes neurais são estruturados. [Russel and Norvig 2004] diz que as estruturas das redes neurais podem se dividir em duas: redes acíclicas ou redes de alimentação direta e redes cíclicas ou redes recorrentes. Uma rede

de alimentação direta representa uma função de sua entrada atual; desse modo, ela não tem nenhum estado interno além dos pesos propriamente ditos. Por outro lado, uma rede recorrente utiliza suas saídas para alimentar de volta suas próprias entradas. Isso significa que os níveis de ativação da rede formam um sistema dinâmico que pode atingir um estado estável ou exibir oscilações, ou mesmo apresentar um comportamento caótico. Além disso, a resposta da rede a uma determinada entrada depende de seu estado inicial, que pode depender de entradas anteriores. Consequentemente, redes recorrentes (diferentes das redes de alimentação direta) podem admitir memórias de curto prazo. Isso as torna mais interessantes como modelos do cérebro, mas também mais difíceis de compreender pois possuem vários desdobramentos. A seguir será descrito um exemplo simples sobre a asserção em que uma rede de alimentação direta representa em função de suas entradas.

A figura 5.4 apresenta um exemplo de rede neural simples possuindo duas unidades de entrada, duas unidades ocultas e uma unidade de saída. (Para manter a simplicidade, foram omitidas as unidades de desvio (bias) nesse exemplo.) Dado um vetor de entrada $x = (x_1, x_2)$, a rede irá calcular o resultado de acordo com a equação 5.4 mostrada a seguir.

$$\begin{aligned} y_1 &= \sigma_1(w_{11}x_1 + w_{12}x_2) \\ y_2 &= \sigma_2(w_{21}x_1 + w_{22}x_2) \\ y_3 &= \sigma(w_{13}y_1 + w_{23}y_2) \\ y_3 &= \sigma(w_{13}\sigma_1(w_{11}x_1 + w_{12}x_2) + w_{23}\sigma_2(w_{21}x_1 + w_{22}x_2)) \end{aligned} \tag{5.4}$$

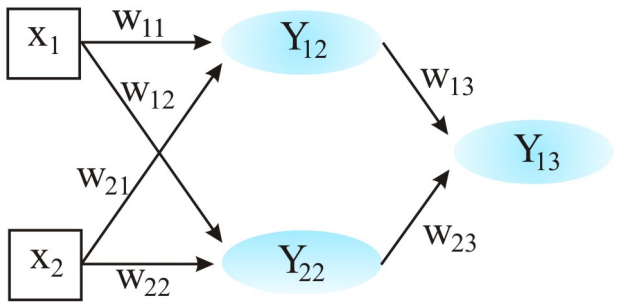


Figura 5.4: Exemplo de uma rede neural simples com duas entradas, uma camada oculta de duas unidades e uma saída.

Ou seja, expressando a saída de cada unidade oculta como uma função de suas entradas, a expressão mostra que a saída da rede como um todo y_3 é uma função das entradas da rede. Além disso, nota-se que os pesos na rede atuam como parâmetros dessa função; escrevendo-se w para o conjunto de parâmetros (entradas e pesos), a rede calcula uma função $h_w(i)$ onde i é o neurônio em questão. Ajustando os pesos, muda-se a função que a rede representa. É assim que a aprendizagem ocorre em redes neurais.

Generalizando, dado um vetor de entrada $(x_0, x_1, x_2, ..., x_{n-1})$ na camada de entrada (0-

ésima camada), o sinal de saída do j-ésimo neurônio na ultima camada é dado pela Equação 6.0.

$$y_j = \sigma\left(\sum_{i=0}^{m} w_{(i,j)}\sigma\left(...\sigma\left(\sum_{i=0}^{n} w_{(i,1)}x_i\right)...\right)\right) \tag{5.5}$$

Onde n é o número de neurônios da camada de entrada (camada 1) e m é o número de neurônios da camada de saída (camada j).

Uma rede neural pode ser usada para classificação ou regressão. No caso de classificação booleana com saídas contínuas (por exemplo, com unidades sigmóides), é tradicional ter uma única unidade de saída, com um valor acima de 0,5 interpretado como uma classe e um valor abaixo de 0,5 como a outra. No caso da classificação em k categorias, pode-se dividir o intervalo da única unidade de saída em k porções, embora seja mais comum ter k unidades de saída separadas, com o valor de cada uma representando a probabilidade relativa dessa classe, de acordo com a entrada atual.

As redes de alimentação direta normalmente estão organizadas em camadas, de tal forma que cada unidade receba apenas a entrada de unidades situadas na camada imediatamente precedente. A seguir serão abordadas as redes de uma única camada que não possuem nenhuma camada oculta e redes de várias camadas, que têm uma ou mais camadas ocultas.

5.3.1 Redes Alimentadas Adiante com Camada Única (Perceptrons)

Em uma rede neural em camadas todas as entradas são conectadas diretamente às saídas nas quais são independes entre si sendo afetadas por pesos também distintos. Em outras palavras, esta rede é estritamente do tipo alimentação direta ou acíclica. A figura 5.5 ilustra esse tipo de arquitetura para quatro nós tanto na camada de entrada quanto na camada de saída. Esta rede é chamada de rede de camada única, sendo que a resignação "camada única" se refere à camada de saída de nós computacionais (neurônios). Não se conta a camada de entrada de nós de fonte, pois lá não é realizada qualquer computação. Examinando a equação 5.3, vemos que o perceptron de limiar retorna 1 se e somente se a soma ponderada de suas entradas (incluindo o desvio) é positiva:

$$\sum_{j=0}^{m} w_j x_j > 0.$$

A equação $\sum_{j=0}^{m} w_j x_j > 0$ irá definir um hiperplano no espaço de entradas. Em geometria, um hiperplano é um subespaço linear, afim ou projetivo de codimensão 1. Por exemplo, em um espaço tridimensional um hiperplano é o plano habitual e em um espaço bidimensional, um hiperplano é uma reta. O perceptron irá retornar 1 apenas se a entrada estiver em um lado desse hiperplano e retornará -1 se a entrada estiver em qualquer ponto do outro plano. Por isso, o perceptron de limiar também é chamado de separador linear. A figura 5.6 representa um hiperplano de duas dimensões (uma linha em um espaço de duas dimensões) para as representações de perceptrons das funções E e OU de duas entradas. Os círculos preenchidos em azul

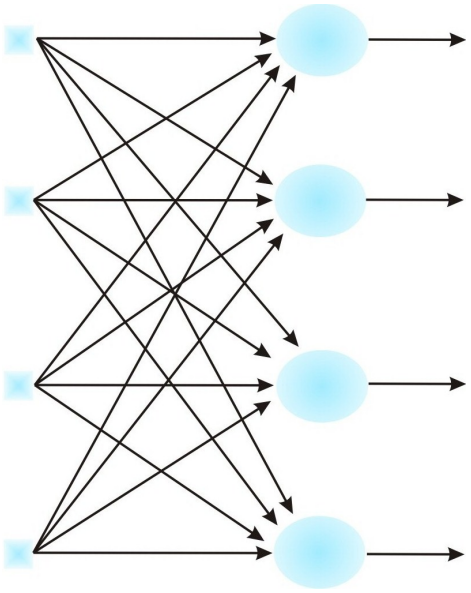


Figura 5.5: Rede alimentada adiante ou acíclica com uma única camada de neurônios.

representam um ponto no espaço onde o valor da função é um, e os círculos vazios mostram um ponto onde o valor é zero. O perceptron pode ser usado nesse caso pois a reta separa todos os círculos preenchidos de todos os círculos vazios. Essas funções representadas pelo perceptron são chamadas de linearmente separáveis.

Os perceptrons foram inicialmente saudados com entusiasmo. No entanto, Nills Nilson (1965) e outros analisaram as limitações de seu modelo. Eles demonstraram que os perceptrons não poderiam resolver uma certa classe de problemas, na qual os pontos de dados não eram linearmente separáveis. A figura 5.6(C) demonstra um exemplo em que o perceptron não conseguiria aprender a função de ativação, pois a mesma não é linearmente separável. Embora várias melhorias do modelo, incluindo os perceptrons multicamadas estivessem sendo concebidas naquele tempo, Marvin Minsky e Seymour Papert, no seu livro Perceptrons (1969) argumentavam que o problema de separabilidade linear não poderia ser suplantado por qualquer configuração de rede de perceptrons. Hoje se sabe que o problema da função XOU pode ser resolvido utilizando redes neurais, pois outras arquiteturas de redes ditas "mais sofisticadas" como as redes multicamadas oferecem subsídios para resolver problemas com uma maior granularidade de entradas, ou seja, um maior nível de detalhes dos dados existentes [Luger 2004].

Apesar de possuir um poder de expressão restrito, os perceptrons de limiar possuem várias vantagens. Uma delas é o seu algoritmo de aprendizagem simples que pode adequar um perceptron de limiar a qualquer conjunto de treinamento linearmente separável.

Algoritmo de Treinamento do Perceptron

Frank Rosenblatt (1958) formulou um algoritmo de treinamento para esse tipo de rede de camada única usando uma fórmula simples de aprendizado supervisionado. Assim como a maioria dos algoritmos de aprendizagem para redes neurais, esse procedimento ajusta os pesos para minimizar alguma medida do erro no conjunto de treinamento. Com isso, a aprendizagem

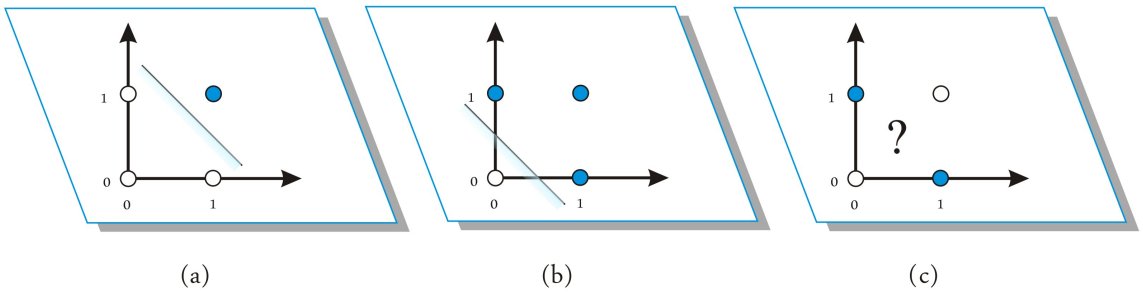


Figura 5.6: Separabilidade linear em perceptrons de limiar. Os pontos azuis indicam um ponto no espaço de entrada em que o valor da função é 1, e os pontos brancos indicam um ponto onde o valor é 0. O perceptron retorna 1 na região sobre o lado não-sombreado da linha. Em (c), não existe nenhuma reta desse tipo que classifique corretamente as entradas.

é formulada como uma busca de informação no espaço de pesos. Após tentar resolver uma ocorrência do problema, um professor fornece a ele um resultado correto. O perceptron modifica, então seus pesos de modo a reduzir o erro. Uma regra, mais conhecida como regra delta, é usada para corrigir esses pesos. Tal algoritmo será explicitado a seguir.

Dado um vetor de entradas $c = (f_1, f_2, ..., f_n)$ e um vetor alvo $A = (a_1, a_2, a_3, ..., a_m)$, um limiar θ e um bias fixo b e uma constante de aprendizado α, o algoritmo se dá de acordo com os passos descritos no algoritmo 1.

Note que somente os pesos conectados a unidades ativas ($x_i \neq 0$) são atualizados. Além disso, os pesos são atualizados somente para padrões que não produzem o valor correto para y. Isto quer dizer que quanto mais padrões produzem respostas corretas, menos aprendizado ocorre. Esse processo tem como decorrência a produção de um conjunto de pesos que minimiza o erro médio sobre todo o conteúdo de treinamento, em várias iterações também chamadas de épocas. As épocas são repetidas até ser alcançado algum critério de parada, na maioria das vezes quando todas as saídas reais estiverem de acordo com a saída desejada. Se existir um conjunto de pesos que forneça a saída ajustada para todos os componentes do conjunto de treinamento, o algoritmo de aprendizagem irá aprendê-lo [Haykin 2001].

Uma forma mais direta de generalizar o perceptron é substituir a sua função de limiar abrupto por outros tipos de função de ativação como funções contínuas. A figura 5.2 mostra o gráfico de algumas funções de ativação comuns: uma função de limiarização bipolar (figura 5.2 (a)) parecida com a usada na regra explicada anteriormente do perceptron, uma função linear e outra sigmóide. As funções sigmóides são assim chamadas por terem seu gráfico em forma "S", como da figura 5.2 (c). Uma função de ativação sigmóide comum chamada de função logística é dada pela expressão 5.6:

$$f(net) = 1/(1 + e^{-\lambda * net}) \tag{5.6}$$

onde $net = \sum w_i x_i$.

Como o caso das funções previamente definidas, x_i é a entrada na componente do vetor de entrada i, w_i é o peso na componente do vetor de entrada i e λ é um "parâmetro de achatamento"

Algoritmo 1 Algoritmo de treinamento do perceptron.

{PASSO 0};
Inicialize pesos e bias; {Para simplicidade atribua valor 0 para pesos e bias.};
Inicialize constante de aprendizado α; {Para simplicidade α pode ser atribuído ao valor 1.};
{PASSO 1};
while Enquanto condição de parada é falsa, faça os passos 2-6; **do**
  {PASSO 2};
  for Cada par de treinamento $c : A$, faça os passos 3-5; **do**
    {PASSO 3};
    Atribua os valores de entrada:
    $x_i = f_i$;
    {PASSO 4};
    Calcule a resposta da unidade de saída:
    $y\_in = b + \sum_i x_i w_i$.

$$y = \begin{cases} 1, & \text{se } y\_in > \theta \\ 0, & \text{se } -\theta < y\_in < \theta \\ -1, & \text{se } y\_in < \theta \end{cases}$$

    {PASSO 5};
    Atualize os pesos e bias se um erro ocorrer para este padrão.
    if $y \neq A$ **then**
      $w_i(novo) = w_i(antigo) + \alpha A x_i$
      $b(novo) = b(antigo) + \alpha A$
    else
      $w_i(novo) = w_i(antigo)$
      $b(novo) = b(antigo)$
    end if
  end for
  {PASSO 6};
  Condição de parada;
  if Nenhum peso foi modificado no passo dois **then**
    Pare e salve os pesos;
  else
    Continue;
  end if
end while

usado para ajustar a curva sigmóide. Conforme λ aumenta, a sigmóide se aproxima de uma função de limiarização linear sobre 0,1; quando λ se aproxima de 1 a função se assemelha a uma reta [Luger 2004].

Pode julgar-se o conjunto de valores dos elementos de entrada para uma rede como determinando um espaço. Cada parâmetro dos dados de entrada se relaciona a uma dimensão,

com cada valor de entrada determinando um ponto no espaço. O problema de aprender uma classificação binária dos exemplos de treinamento se reduz àquele de separar estes pontos em dois grupos. Para um espaço de n dimensões, uma classificação é linearmente separável se as classes puderem ser separadas por um hiperplano de dimensão n-1. A figura 5.7 apresenta um gráfico explicitando a busca pelo vetor peso ideal através da regra delta em um espaço de características bidimensional.

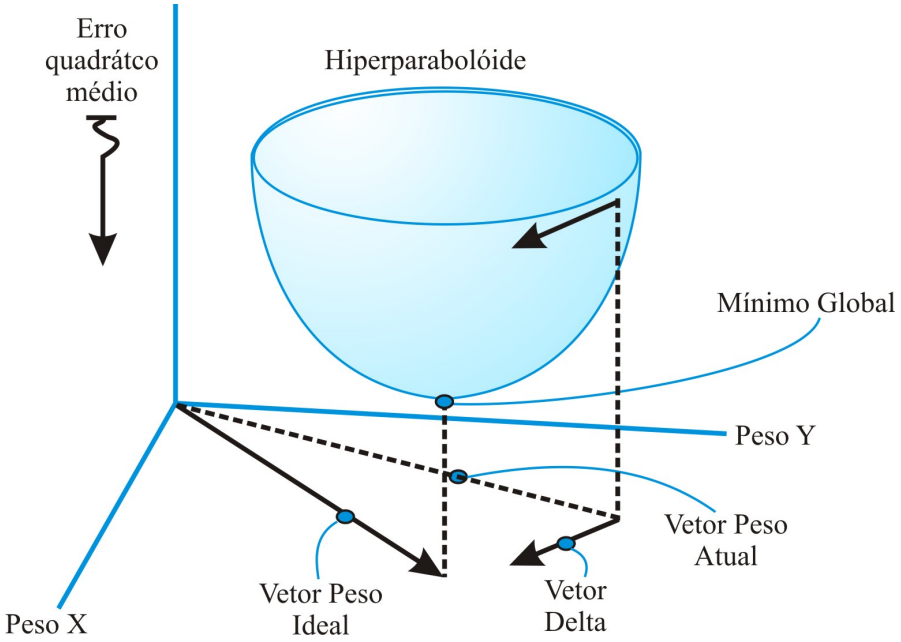


Figura 5.7: Regra Delta

Como consequência da restrição da separabilidade linear das redes perceptron, as análises se voltaram para as arquiteturas simbólicas, abafando o avanço na tecnologia conexionista. No entanto, trabalhos seguintes nas décadas de 70 e 80 divulgaram que esses problemas eram solucionáveis, alguns através das redes neurais de alimentação direta de várias camadas apresentadas a seguir.

5.3.2 Redes Alimentadas Diretamente com Múltiplas Camadas

Esse segundo tipo de arquitetura de rede de alimentação direta se difere do perceptron principalmente pela presença de uma ou mais camadas ocultas possuindo, nestas, unidades neuronais também chamadas de unidades ocultas ou neurônios ocultos. O papel fundamental dos neurônios ocultos é participar de maneira útil entre as unidades de entrada e de saída de uma rede neural. Quando se adiciona camadas ocultas em uma rede pode-se elevar o espaço de hipóteses que a rede pode representar. Em um sentido bem livre, a rede obtém uma perspectiva global que, apesar de sua conectividade podendo, com uma única camada oculta suficientemente grande pode representar qualquer função contínua de entradas com exatidão arbitrária e com duas camadas até mesmo funções descontínuas. No entanto, ainda não é

possível, para qualquer estrutura de rede específica, assinalar precisamente que funções podem e que funções não podem ser representadas por tal rede [Luger 2004].

Os nós de entrada da rede municiam os elementos do padrão de ativação, ou seja, o vetor de entradas que formam os sinais aplicados aos neurônios ou nós computacionais da segunda camada, que é a primeira camada oculta. Os sinais de saída da segunda camada servirão de entradas para a terceira camada e assim por diante para o resto da rede. Característicamente, os neurônios em cada camada da rede possuem como suas entradas exclusivamente os sinais de saída da camada antecedente. O conjunto de sinais de saída dos neurônios da camada de saída da rede compõe a resposta global da rede para o padrão de ativação dado pelos nós de fonte da camada de entrada. O grafo arquitetural na figura 5.8 elucida o diagrama de uma rede neural de múltiplas camadas alimentada adiante para o caso de uma única camada oculta. De maneira mais concisa, a rede na figura 5.8 pode ser reportada como uma rede 10-4-2, pois a mesma possui 10 neurônios de fonte, 4 neurônios ocultos e 2 neurônios de saída. Como um outro exemplo, uma rede alimentada adiante com n nós de fonte, $k1$ neurônios na primeira camada oculta, $k2$ neurônios na segunda camada oculta e m neurônios na camada de saída é referida como uma rede $n - k1 - k2 - m$ [Haykin 2001].

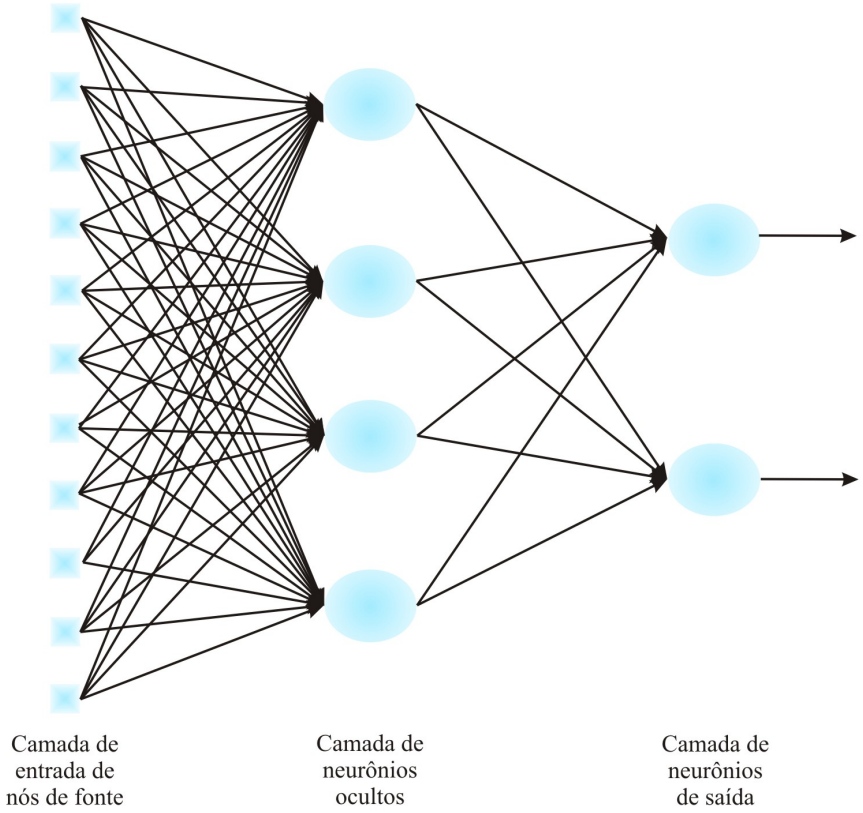


Figura 5.8: Rede alimentada adiante ou acíclica totalmente conectada com uma camada oculta e uma camada de saída.

A rede neural da figura 5.8 é dita totalmente conectada, pois cada um dos nós de uma camada da rede está conectado a todos os nós da camada imediatamente seguinte. Todavia, se

alguns dos elos de comunicação (conexões sinápticas) estiverem faltando na rede, diz-se que a rede é parcialmente conectada [Haykin 2001].

Os algoritmos de aprendizagem para a rede de várias camadas são similares ao algoritmo de aprendizado de perceptrons explicitados anteriormente. A diferença fundamental é que, enquanto o erro na camada de saída é claro, o erro nas camadas ocultas parece incompreensível, porque os dados de treinamento não informam quais valores os nós ocultos devem possuir. Para contornar esse problema pode-se efetuar a propagação de retorno do erro da camada de saída para as camadas ocultas. O processo de propagação de retorno também chamado de *back-propagation* insurge diretamente de uma derivação do gradiente de erro global. A abordagem adotada pelo algoritmo é iniciar a camada de saída e disseminar o erro retroativamente através das camadas ocultas. Quando se analisa a aprendizagem com a regra delta, pode-se notar que toda a informação necessária para atualizar os pesos de um neurônio se encontra localmente neste neurônio, exceto pela parcela de erro. Para nós de saída, esta parcela é facilmente calculada pela diferença entre os valores de saída desejado e real. Para nós em camadas ocultas, é consideravelmente mais difícil determinar o erro para o qual um nó é responsável [Haykin 2001]. A seguir será descrito esse algoritmo de retropropagação do erro.

5.3.3 Algoritmo para Treinamento de Redes com Múltiplas Camadas: *Backpropagation*

O treinamento de uma rede neural de múltiplas camadas por retropropagação envolve três estágios: a propagação dos padrões de entrada pela rede, a retropropagação do erro associado e o ajuste dos pesos [Fausett 1994]. A seguir é apresentada a nomenclatura que será utilizada no algoritmo de treinamento para a retropropagação da rede.

Sumário de Notações Utilizadas

A seguinte notação será utilizada ao longo dessa seção.

x

Vetor de entrada durante a fase de testes:
$x = (x_1, x_2, x_3, ..., x_n)$.

A

Vetor de saída alvo, durante o treinamento:
$A = (a_1, a_2, a_3, ..., a_m)$.

Δw_{ij}

Mudança em w_{ij}:
$\Delta w_{ij} = [w_{ij}(novo) - w_{ij}(antigo)]$;

α

Constante de aprendizado.

δ_k

Porção de ajuste do peso para correção do erro em w_{jk} que corresponde a um erro na

saída Y_k; Além disso, é a informação sobre o erro na unidade Y_k que é propagada de volta para as unidades que alimentam a unidade Y_k.

δ_j

Porção de ajuste do peso para correção do erro em v_{jk} que corresponde a retropropagação do erro da camada de saída para a unidade oculta Z_j.

v_{0j}

Bias da camada oculta j.

Z_j

Unidade oculta j.
A entrada da rede para Z_j é denotada por z_in_j:
$z\_in_j = v_{0j} + \sum_i x_i v_{ij}$.
O sinal de saída (ativação) de Z_j é denotado z_j:
$z_j = \sigma_j(z\_in_j)$.

w_{0k}

Bias da camada de saída k.

Y_k

Saída da unidade k.
A entrada da rede para Y_k é denotada por y_in_k:
$y\_in_k = w_{0k} + \sum_j z_j w_{jk}$.
O sinal de saída (ativação) de Y_k é denotado y_k:
$y_k = \sigma_k(y\_in_k)$.

Como pode-se observar na figura 5.9, durante a propagação, cada unidade de entrada (X_i) recebe um sinal de entrada e propaga esse sinal para cada uma das unidades ocultas $Z_1, ..., Z_p$. Cada unidade oculta, então, calcula sua ativação e envia seus sinais (z_j) para cada unidade de saída. Cada unidade de saída (Y_k) calcula sua ativação y_k para formar a resposta da rede dado o padrão de entrada [Fausett 1994].

Durante o treinamento, cada unidade de saída compara seu valor de saída real y_k com o correspondente valor de saída ideal a_k para determinar o erro associado àquele padrão naquela unidade. Baseado nesse erro, o fator $\delta_k (k = 1, ..., m)$ é calculado. δ_k é usado para distribuir o erro da unidade de saída Y_k de volta a todas as unidades da camada anterior (unidades ocultas conectadas à Y_k). Este fator também é usado (mais tarde) para atualizar os pesos entre a camada de saída e a camada oculta. De maneira semelhante, o fator $\delta_j (j = 1, ..., p)$ é calculado para cada unidade da camada oculta Z_j. Não é necessário propagar o erro de volta para a camada de entrada, mas δ_j é usado para atualizar os pesos entre a camada oculta e a camada de entrada [Fausett 1994].

Depois de todos os fatores δ serem determinados, os pesos de todas as camadas são ajustados simultâneamente. O ajuste do peso w_{jk} (vindo da unidade oculta Z_j para a unidade de saída y_k) é baseado no fator δ_k e no sinal de ativação z_j da camada oculta Z_j. O ajuste de pesos v_{ij} (vindo da unidade de entrada X_i para a unidade oculta Z_j) é baseado no fator δ_j e no sinal de ativação x_i da unidade de entrada.

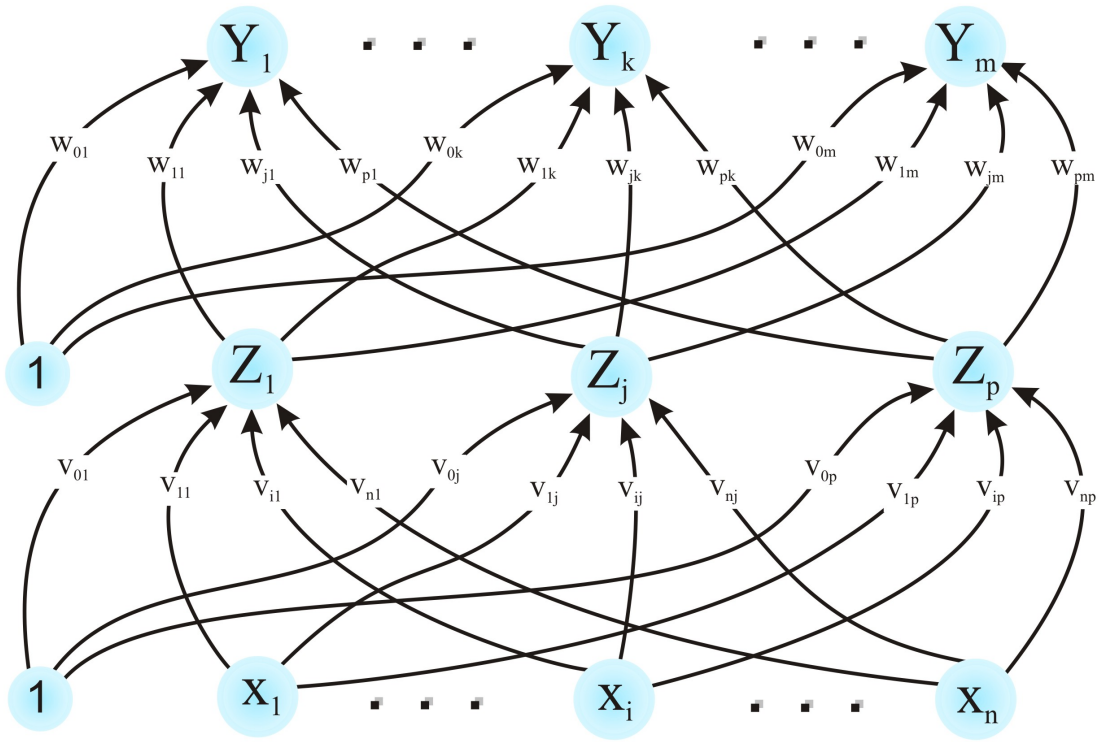


Figura 5.9: Rede Neural do tipo *Backpropagation* com uma camada oculta (Adaptado de [Fausett 1994]).

Função de ativação

Uma função de ativação para uma rede *backpropagation* deve possuir as seguintes características: ser contínua, diferenciável, monotonicamente decrescente e, para fins práticos, ser fácil de calcular. Para as funções de ativação mais usadas, o valor da derivada (como um valor particular da variável independente) pode ser expresso em termos dos valores da função. Usualmente, é esperado que a função seja saturada, ou seja, que possua uma aproximação de valores máximos e mínimos finitos assintóticamente.

Uma das mais típicas funções de ativação é a função sigmóide binária, que possui faixa de (0,1) e é definida como:

$$\sigma_1(x) = \frac{1}{1 + exp(-x)},$$

com

$$\sigma_1'(x) = \sigma_1(x)[1 - \sigma_1(x)].$$

Esta função é ilustrada pela figura 5.10(a).

Outra função de ativação conhecida é a função sigmóide bipolar, que possui a faixa (-1,1) e é definida como:

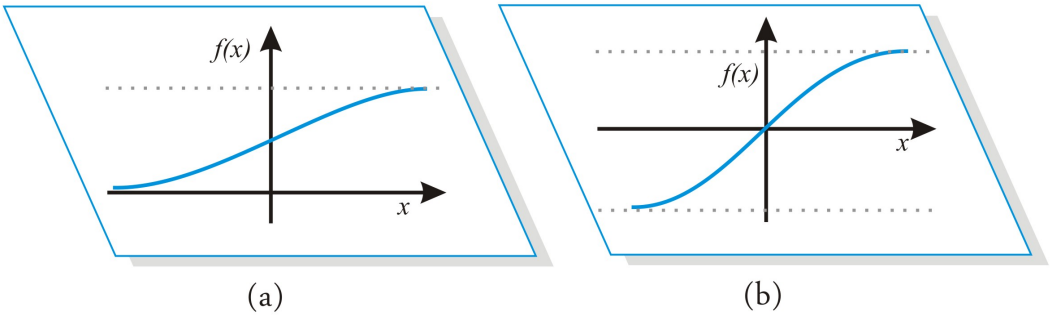


Figura 5.10: Função sigmóide binária (a) com faixa (0,1) e função sigmóide bipolar (b) com faixa (-1,1).

$$\sigma_2(x) = \frac{2}{1 + exp(-x)} - 1,$$

com

$$\sigma_2'(x) = \frac{1}{2}[1 + \sigma_2(x)][1 - \sigma_2(x)].$$

Esta função é ilustrada pela figura 5.10(b).

Algoritmo de Treinamento *Backpropagation*

Qualquer uma das funções de ativação mostradas na seção anterior pode ser usada no algoritmo de retropropagação padrão apresentado a seguir. A forma dos valores utilizados (especialmente os valores de saídas alvo) é um importante fator na escolha da função apropriada. Note que, por causa da simples relação entre o valor da função e sua derivada, nenhuma avaliação adicional da exponencial é requerida para calcular as derivadas requeridas durante a fase de retropropagação do algoritmo. O algoritmo é dado pelo algoritmo 2.

Note que na implementação deste algoritmo, vetores distintos podem ser usados para os deltas das unidades de saída (Passo 6, δ_k) e para os deltas das unidades ocultas (Passo 7, δ_j).

Uma época (ou iteração) é um ciclo sobre todo o conjunto de vetores de treinamento. Habitualmente, muitas épocas são requeridas para o treinamento de uma rede *backpropagation*. O algoritmo mostrado aqui atualiza os pesos depois que cada padrão de treinamento é apresentado. Uma variação comum é a atualização em lotes, em que as atualizações dos pesos são acumuladas sobre uma época inteira (ou sobre algum outro número de apresentação de padrões) para então ser aplicada. Embora esse algoritmo forneça uma solução para o problema da aprendizagem em redes multicamadas, o mesmo possui suas próprias dificuldades. Como no caso da subida de encosta onde o algoritmo pode convergir para mínimos locais, dando uma aproximação não ideal ou até mesmo errônea para a taxa de erro. Além disso, o custo computacional da retropropagação pode ser elevado, especialmente quando a rede converge lentamente [Haykin 2001] [Fausett 1994].

Algoritmo 2 Algoritmo de treinamento *Backpropagation.*

{PASSO 0 - Inicialize pesos e bias};
Inicialize pesos e bias;
Inicialize constante de aprendizado α;
{PASSO 1};
while Enquanto condição de parada é falsa, faça os passos 2-9; **do**
  {PASSO 2};
  for Cada par de treinamento, faça os passos 3-8; **do**
    Propagação da rede:
    {PASSO 3};
    Cada unidade de entrada $(X_i, i = 1, ..., n)$ recebe um sinal de entrada x_i e distribui esse sinal para todas as unidades da camada seguinte (unidades ocultas).
    {PASSO 4};
    Cada unidade oculta $(Z_j, j = 1, ..., p)$ soma seus sinais de entradas ponderados:
    $z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$;
    Aplique sua função de ativação para calcular o sinal de saída:
    $z_j = \sigma(z\_in_j)$;
    Distribua este sinal para todas as unidades da camada seguinte (unidades de saída).
    {PASSO 5};
    Cada unidade de saída $(Y_k, k = 1, ..., m)$ soma seus sinais de entradas ponderados:
    $y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$;
    Aplique sua função de ativação para calcular o sinal de saída:
    $y_k = \sigma(y\_in_k)$;
    Retropropagação do erro.:
    {PASSO 6};
    Cada unidade de saída $(Y_k, k = 1, ..., m)$ recebe um valor de saída alvo relacionado ao padrão de treinamento e computa o fator de erro:
    $\delta_k = (a_k - y_k)\sigma'(y\_in_k)$;
    Calcule seu termo de correção de pesos (utilizado para atualizar w_{jk} mais tarde):
    $\Delta w_{jk} = \alpha \delta_k z_j$;
    Calcule seu termo de correção do bias (utilizado para atualizar w_{0k} mais tarde):
    $\Delta w_{0k} = \alpha \delta_k$;
    Envie δ_k para as unidades da camada anterior.
    {PASSO 7};
    Cada unidade oculta $(Z_j, j = 1, ..., p)$ soma suas entradas delta (vindas da camada seguinte):
    $\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$;
    Multiplique o resultado pela derivada da sua função de ativação para calcular o fator de erro:
    $\delta_j = \delta\_in'_\sigma(z\_in_j)$;
    Calcule seu termo de correção de pesos (utilizado para atualizar v_{ij} mais tarde):
    $\Delta v_{ij} = \alpha \delta_j x_i$;
    Calcule seu termo de correção do bias (utilizado para atualizar v_{0j} mais tarde):
    $\Delta v_{0j} = \alpha \delta_j$;
    Atualização dos pesos:
    {PASSO 8};
    Cada unidade de saída $(Y_k, k = 1, ..., m)$ atualiza seus bias e pesos $(j = 0, ..., p)$:
    $w_{jk}(novo) = w_{jk}(antigo) + \Delta w_{jk}$;
    Cada unidade oculta $(Z_j, j = 1, ..., p)$ atualiza seus bias e pesos $(i = 0, ..., n)$:
    $v_{ij}(novo) = v_{ij}(antigo) + \Delta v_{ij}$;
  end for
  {PASSO 9};
  Condição de parada;
end while

5.3.4 Procedimento de Aplicação na Fase de Testes.

Após a rede estar treinada, a mesma será aplicada na fase de teste utilizando somente a fase de propagação da rede do algoritmo de treimanento. O procedimento de aplicação é dado no algoritmo 3.

Algoritmo 3 Aplicação da Rede Neural na fase de Testes.

1: {PASSO 0};
2: Carregue os pesos (vindos do treinamento da rede);
3: {PASSO 1};
4: **for** Para cada vetor de entrada, faça os passos 2-4; **do**
5: {PASSO 2};
6: **for** i=1,..,n: **do**
7: Atribua a unidade de ativação x_i;
8: **end for**
9: {PASSO 3};
10: **for** j=1,..,p: **do**
11: $z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$;
12: $z_j = \sigma(z\_in_j)$;
13: **end for**
14: {PASSO 4};
15: **for** k=1,..,m: **do**
16: $y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$;
17: $y_k = \sigma(y\_in_k)$;
18: **end for**
19: **end for**

5.4 Escolha de Atributos de Treinamento de uma Rede Neural

5.4.1 Incialização de Pesos e Bias

A escolha inicial dos pesos irá influenciar na convergência do erro para um mínimo local ou global assim como na velocidade desta convergência. A atualização dos pesos irá depender tanto da derivada da função de ativação das camadas posteriores e da ativação das unidades anteriores. Por esta razão, é importante permitir escolhas de pesos iniciais de modo que tanto as ativações quanto suas derivadas sejam iguais a 0. Os valores dos pesos iniciais não podem ser muito grandes ou os sinais de entrada para cada unidade oculta ou de saída provavelmente cairão em uma região onde a derivada da função sigmóide possui um valor muito pequeno (também chamada de região de saturação). Por outro lado, se os valores iniciais forem muito pequenos, a entrada da rede para as unidades ocultas e de saída serão bem próximas a 0, o que causará um treinamento extremamente lento [Fausett 1994].

Um procedimento comum é inicializar os pesos (e bias) com valores aleatórios entre -0.5 e 0.5 (ou entre -1 e 1). Estes valores podem ser negativos e positivos pois os pesos finais após o treinamento também pertencerão a ambos.

5.4.2 Incialização da constante de aprendizado

À medida que o treinamento evolui, os pesos sinápticos podem passar a assumir valores maiores, forçando a operação dos neurônios na região onde a derivada da função de ativação é muito pequena. Como o erro retropropagado é proporcional a esta derivada, o processo de treinamento tende a se estabilizar, levando a uma paralisação da rede sem que a solução tenha sido encontrada. Isto pode ser evitado pela aplicação de uma taxa de aprendizagem menor. Teoricamente, o algoritmo de aprendizado exige que a mudança nos pesos seja infinitesimal [Rumelhart and McClelland 1986]. Entretanto, a alteração dos pesos nessa proporção é impraticável, pois implicaria em tempo de treinamento infinito. Em vista disso, é recomendável que a taxa de aprendizado assuma valor maior no início do treinamento e, à medida em que se observe decréscimo no erro da rede, essa taxa também seja diminuída. Diminuindo progressivamente a taxa de atualização dos pesos, o gradiente decrescente está apto a alcançar uma solução melhor.

5.4.3 Número de iterações e condição de parada

A principal motivação para a aplicação de uma rede *backpropagation* é atingir um balanço entre as respostas corretas para os dados de treinamento e também boas respostas para novos padrões de entrada na fase de testes (ou seja, um balanceamento entre a memorização e a generalização). Portanto, não é necessáriamente vantajoso se treinar uma rede neural até que se atinja um erro total muito pequeno.

5.4.4 Número de pares de treinamento

A relação entre o número de padrões de treinamento P, o número de pesos a serem treinados W, e a acurácia da classificação esperada é dada pela seguinte regra. A questão a ser respondida é "Sob quais circustâncias eu posso assegurar que uma rede que é treinada para classificar corretamente uma dada porcentagem de padrões de teste corretamente, também irá classificar corretamente padrões de teste vindos do mesmo espaço de amostras?". Especificamente, se a rede é treinada para classificar a fração de $1-(e/2)$ de padrões de treinamento corretamente, onde $0 < e \leq 1/8$, posso assegurar que a rede irá classificar $1-e$ padrões de teste corretamente? A resposta é que se há um número suficiente de padrões de treinamento, a rede será capaz de generalizar como desejado (classificar padrões de teste corretamente). A quantidade suficiente de padrões de treinamento é determinada pela condição:

$$\frac{W}{P} = e,$$

ou

$$P = \frac{W}{e}.$$

Por exemplo, se e=0.1, uma rede neural com 100 pesos requer 1000 padrões de treinamento para assegurar uma classificação correta de 90% dos padrões de teste, assumindo que a rede foi treinada para classificar corretamente 95% dos padrões de treinamento.

5.4.5 Número de camadas ocultas

Para uma rede neural com mais de uma camada oculta, uma pequena modifcação do algoritmo proposto na seção anterior, deve ser realizada. O cálculo dos δ´s é repetido para cada camada oculta adicional no turno, somando sobre os δ´s paras as unidade na camada anterior para alimentar na camada atual onde o δ é então calculado. Tomando como referência o algoritmo, o passo 4 é repetido para cada camada oculta na fase de propagação da rede e o passo 7 é repetido para cada camada oculta na fase de retropropagação da rede. Alguns autores afirmam que uma rede com uma única camada oculta é suficiente para a rede se aproximar qualquer mapa de entradas aos seus padrões de saída para um valor arbitrário de acurácia. No entando, duas ou mais camadas ocultas podem tornar o treinamento mais fácil em alguns casos [Fausett 1994].

O número de unidades de processamento das camadas de entrada e saída é usualmente determinado pela aplicação. Com relação às camadas ocultas, a relação não é tão óbvia. O ideal é utilizar um número razoável de unidades ocultas para que a generalização não fique prejudicada. Se o número de neurônios ocultos for muito grande, a rede acaba memorizando os padrões apresentados durante o treinamento. Contudo, se a arquitetura das camadas ocultas possuir unidades de processamento em número inferior ao necessário, o algoritmo backpropagation pode não conseguir ajustar os pesos sinápticos adequadamente, impedindo a convergência para uma solução. O empirismo ainda é a melhor indicação para a definição da topologia de um modelo conexionista. O número de camadas intermediárias e de neurônios geralmente é obtido por tentativa-e-erro, sendo a experiência a principal heurística do projetista da RNA. Todavia, deve-se levar em consideração que com um número muito pequeno de neurônios a rede pode não ter recursos suficientes para aprender. Além disso, a utilização de um número muito grande de neurônios e conexões pode fazer a rede perder sua capacidade de interpolação, pois ela decora os exemplos, ficando assim "cristalizada".

5.4.6 Representação do Conhecimento

Uma das principais características das redes neurais é a representação do conhecimento tanto que o termo "conhecimento" é utilizado na própria definição de redes neurais. Esse assunto é muito discutido entre os neurocientistas e será abordado nessa seção iniciando com a seguinte definição do termo dada por [Fischler and Firschein 1987]:

> *"Conhecimento se refere à informação armazenada ou a modelos utilizados por uma pessoa ou máquina para interpretar, prever e responder apropriadamente ao mundo exterior."*

As características fundamentais da representação do conhecimento podem se dividir em duas: qual informação é legitimamente tornada expressa e como a informação é transformada em uma sequência de sinais conforme um determinado código para seu uso posterior. Deste modo, pelo seu próprio caráter, a representação do conhecimento é orientada a uma determinada finalidade. Como foram citadas nos capítulos anteriores, as soluções de problemas através de aplicações de máquinas "inteligentes" no mundo real dependem de uma boa representação do conhecimento [Woods 1964]. Isso ocorre nas redes neurais, que não deixam de ser, também, uma classe representativa dessas máquinas "inteligentes". Contudo, por causa da diversidade de representação deste conhecimento através da rede desde suas entradas até os parâmetros internos, o projeto de uma rede neural pode se tornar bem laborioso e o uso de técnicas incorretas pode atrapalhar na solução final tornando-a, muitas vezes, insatisfatória.

O aprendizado do mundo (ambiente) em que a rede está implantada se torna um aspecto importante nesse contexto assim como a manutenção desse padrão de modo consistente para o mundo real de maneira a alcançar as metas especificadas da aplicação de interesse. O conhecimento do mundo, de acordo com [Haykin 2001], é composto de duas classes de informação:

1. O estado conhecido do mundo, constituído pelos fatos sobre o que é e o que era conhecido; esta forma de conhecimento é chamada de informação prévia.

2. As observações (medidas) do mundo, obtidas através de sensores implementados para examinar o espaço em que a rede neural irá atuar. Habitualmente, estas observações são intrinsecamente ruidosas, sendo sujeitas a falhas por vários motivos como o ruído do sensor ou até mesmo deficiências no sistema. Em todo caso, as observações que são assim obtidas oferecem o conjunto de informações de onde são extraídos os exemplos aproveitados no treinamento da rede neural.

Os exemplos acima citados também podem se subdividir em dois: rotulados ou não rotulados. No caso dos exemplos rotulados, cada exemplo já é anteriormente rotulado, ou seja, já é previamente associado a uma resposta esperada (saída-alvo). No caso de exemplos não rotulados, os mesmos são constituídos de ocorrências distintas dos próprios sinais de entrada. De qualquer forma, o conjunto de exemplos, sejam eles rotulados ou não, representa o conhecimento acerca do ambiente de interesse que uma rede neural possui para poder aprender através do treinamento. Caracteristicamente, os exemplos (amostra de treinamento) consistem de uma gama de dados simbólicos de uma circunstância do mundo real. Uma diferença fundamental entre o projeto de uma rede neural e o de sua parte oposta (ou complementar), o processamento de informação clássico (classificação de padrões), é que neste último, habitualmente o processamento é realizado, a princípio, através da formulação de um modelo matemático das observações do ambiente. Além disso, há também uma validação deste modelo com dados reais, e então, uma estruturação do projeto com base no mesmo. Já o projeto de uma rede neural, o processamento ocorre exatamente o contrário, pois o mesmo é baseado literalmente nos dados do mundo real, consentindo-se que o conjunto de dados fale por si mesmo. Com isso, a rede neural não apenas aprovisiona o modelo subentendido do ambiente no qual ela está inserida, como também atinge o desempenho do processamento da informação na qual se está interessado.

No entanto, o tema da representação do conhecimento no interior de uma rede artificial é muito complicado. Apesar disso existem quatro regras para a representação do conhecimento que são de senso comum por muitos autores [Anderson and Rosenfeld 1988]:

1ª Regra: Entradas semelhantes de classes semelhantes habitualmente necessitam produzir resultados similares dentro da rede, assim sendo devem ser rotuladas como pertencentes ao mesmo conjunto.

Há uma gama enorme de medidas para determinar a "similaridade" entre entradas. Uma medida de similaridade comumente usada é baseada no conceito de distância euclidiana. Essa distância pode ser especificada da seguinte maneira: seja um vetor unidimensional x_i:

$$x_i = [x_{i1}, x_{i2}, ..., x_{im}]^T$$

onde os elementos deste vetor são todos números reais; o índice superior T indica a trasposição matricial. O vetor x_i define um ponto em um espaço de dimensão m chamado espaço euclidiano e representado por $\mathbb{R}^m$. A distância euclidiana entre um par de vetores x_i e x_j é definida pela expressão 5.5 apresentada a seguir:

$$\begin{aligned} d(x_i, x_j) &= \|x_i - x_j\| \\ &= [\sum_{k=1}^{m}(x_{ik} - x_{jk})^2]^{1/2} \end{aligned} \tag{5.5}$$

onde x_{ik} e x_{jk} são os $k-$ésimos elementos dos vetores de entrada x_i e x_j respectivamente.

De modo apropriado, a semelhança entre as entradas atribuídas pelos vetores x_i e x_j é diretamente proporcional à distância euclidiana d(x_i, x_j). Quanto mais próximos entre si encontram-se os elementos individuais dos vetores de entrada x_i e x_j, menor será a distância euclidiana d(x_i, x_j), e, deste modo, maior será a similaridade entre os vetores x_i e x_j. A regra 1 exige que, se os vetores x_i e x_j são similares, eles devem ser atribuídos à mesma categoria [Haykin 2001].

2ª Regra: Devem ser atribuídas como pertencentes a categorias diferentes as entradas que possuem uma distância superior à definida pelo arquiteto da rede.

Essa regra diz respeito ao oposto da regra um, pois deve-se também observar, além das semelhanças, as diferenças entre os ítens na rede, os quais devem pertencer a classes separadas.

3ª Regra: Se uma característica particular é mais significante, logo precisa conter um número considerável de neurônios envolvidos na representação daquele componente na rede.

Para elucidar melhor essa regra, analise o seguinte exemplo: uma aplicação de radar envolvendo o rastreamento de um alvo (uma aeronave) na presença de perturbações (reflexões de radar por alvos indesejáveis como edifícios, árvores e formações meteorológicas).

A performance da detecção deste sistema de radar é medida em termos de duas possibilidades:

Probabilidade de detecção, definida como a probabilidade de o sistema decidir que o alvo está presente, quando ele realmente está.

Probabilidade de alarme falso, definida como a probabilidade de o sistema decidir que um alvo está presente, quando na realidade ele não está.

De acordo com o critério de Neyman-Pearson, a possibilidade de detecção é elevada ao máximo e sujeita à ressalva de que a probabilidade de alarme falso não ultrapasse um determinado valor. Nesta aplicação, a presença real de um alvo no sinal recebido representa uma característica importante da entrada. Na verdade, a Regra 3 afirma que deve haver um grande número de neurônios envolvidos na tomada de decisão se um alvo está presente, quando ele realmente estiver. Pelo mesmo motivo, deve haver um número muito grande de neurônios envolvidos na tomada de decisão se a entrada consiste apenas de perturbações, quando realmente este for o caso. Em ambas as situações o grande número de neurônios assegura um elevado grau de precisão na tomada de decisão e tolerância a falhas ou a sinais ruidosos de entrada.

4ª Regra: Informação prévia e invariâncias devem ser incorporadas no projeto de uma rede neural, simplificando com isso o projeto de rede por não ter que aprendê-las.

A Regra 4 é particularmente importante porque a aderência adequada a ela resulta em uma rede neural com uma estrutura especializada (restrita). Isto é altamente desejável por várias razões [Russo 1991]:

1. Sabe-se que as redes biológicas visuais e auditivas são muito especializadas.

2. Uma rede neural com estrutura especializada normalmente tem um número menor de parâmetros livres disponíveis para ajuste do que uma rede totalmente conectada. Consequentemente, a rede especializada requer um menor conjunto de dados para treinamento, aprende mais rápido e frequentemente generaliza melhor.

3. A taxa de transmissão de informação através da rede especializada (i.e. a produtividade da rede) é acelerada.

4. O custo de construção de uma rede especializada é reduzido por causa do seu tamanho menor, quando comparado com a rede totalmente conectada equivalente.

Parte II

Nossa Contribuição

Capítulo 6

Modelo Proposto

Conforme foi mencionado no capítulo inicial, o objetivo deste trabalho é a concepção de um modelo que, auxiliado pelas redes neurais artificiais, irá caracterizar imagens. Assim o sistema implementado tentará usar a representação do conhecimento e das semânticas de alto nível através de exemplos para transformar características de baixo nível em características neurosemânticas na tentativa de reduzir o gap-semântico existente. Uma visão geral do modelo proposto será apresentada na próxima seção.

6.1 Visão Geral

O modelo proposto pode ser dividido em três etapas básicas: a fase de treinamento, a fase de caracterização e a fase de consulta. Este capítulo descreve essas etapas e a Figura 6.1 mostra uma visão geral do sistema proposto.

A primeira etapa consiste no treinamento da rede, onde é escolhido o banco de dados de imagens de treinamento, a definição dos descritores de baixo nível que irão representar essas imagens, a escolha da arquitetura da rede bem como seus parâmetros e o treinamento da mesma. Esses procedimentos são feitos de forma *off-line*, ou seja, serão realizados antes do sistema estar pronto para o uso.

Após a rede ser treinada, seus parâmetros serão armazenados e essa mesma rede (sem as funções de ativação) servirá como suporte para as fases 2 e 3. Nas seções seguintes a descrição e a formulação matemática para esses procedimentos serão apresentados.

Na segunda fase do modelo, também chamada de fase de caracterização neurosemântica, é escolhido um banco de testes, cujas imagens serão representados pelos mesmos descritores de características da fase anterior. Os vetores resultantes desta caracterização servirão de entrada da rede que processará os mesmos e retornará os vetores neurosemânticos para a caracterização das imagens. Esse vetores serão armazenados em uma estrutura de dados e servirão como suporte para a fase 3. Tais procedimentos também são realizados de forma *off-line* e serão detalhados nas seções seguintes.

A terceira e última fase do modelo, também chamada de fase de recuperação, é a fase onde haverá a interação com o usuário. Essa fase é formada basicamente por um módulo de CBIR

que, após receber uma imagem consulta do usuário, caracterizará essa imagem neurosemanticamente e então passará à análise de similaridade da imagem consulta com as outras imagens do banco de dados retornando ao usuário um *ranking* de imagens mais similares. Este modulo também será explanado nas seções seguintes.

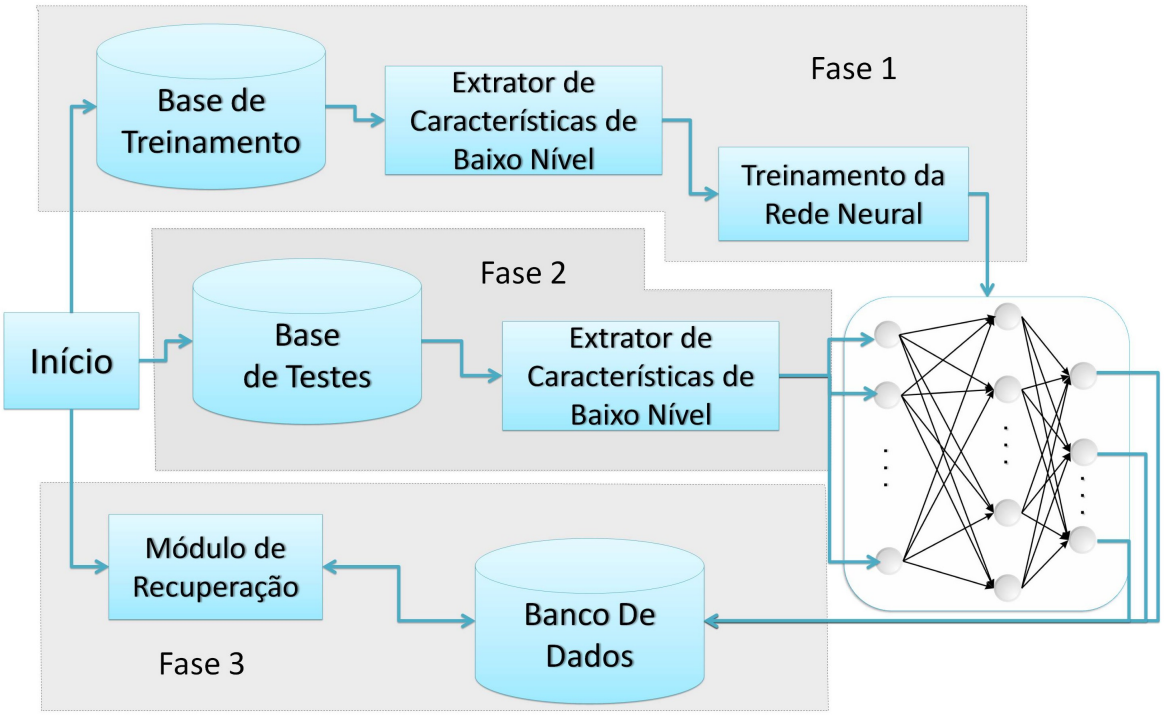


Figura 6.1: Fluxograma que representa uma visão geral do modelo proposto.

6.1.1 Sumário de Notações Utilizadas

Ao longo deste capítulo denotaremos por:

Itr_j
O conjunto j de imagens de treinamento.

k
O número de grupos de imagens de treinamento.

n
O número de imagens em cada grupo de imagens de treinamento.

I_i^j
Imagem i pertencente ao grupo j.
$Itr_i = \{I_0^j, I_1^j, I_2^j, I_{n-1}^j\}$, $j = 1, ...k$

Ctr

O conjunto de características de baixo nível.

c_i^j

O vetor de caraterísticas da i-ésima imagem do j-ésimo grupo.
$Ctr = \{c_0^1, c_1^1, ..., c_{kn-1}^1, c_0^2, c_1^2, ..., c_{kn-1}^2, c_0^k, c_1^k, ..., c_{kn-1}^k\}$

p

O número de valores que representam as características de baixo nível.

f_l^{ij}

Característica de baixo nível da imagem i do grupo j na l-ésima posição.
$c_i^j = (f_0^{ij}, f_1^{ij}, f_2^{ij}, ..., f_{p-1}^{ij})$

A_i^j

Vetor de saída alvo durante o treinamento para a imagem i do grupo j.

a_l^{ij}

A saída do l-ésimo neurônio para a imagem i do grupo j.
$A_i^j = (a_1^{ij}, a_2^{ij}, a_3^{ij}, ..., a_k^{ij})$

G

A Rede Neural implementada para o modelo proposto. (Pode ser vista também como $G(c)$: função de transformação na fase de caracterização neurosemântica.)

$s_j^{(q)}$

O sinal de saída do j-ésimo neurônio na q-ésima camada.

$w_{ij}^{(q)}$

O peso de conexão vindo do i-ésimo neurônio na camada $(q-1)$ para o j-ésimo neurônio na q-ésima camada.

$S^t(c)$

A função de propagação do vetor de características de baixo nível c pela rede na iteração t.

Ite

O conjunto de imagens de testes.

N_i

O vetor de características neurosemântico da imagem i.
$N_i = (s_1, s_2, s_3, ..., s_k)$

6.2 Fase 1: Treinamento da Rede Neural

6.2.1 O Banco de Imagens de Treinamento

Para que a rede neural possa representar bem as semânticas das imagens, generalizando as redundâncias e ressaltando seus aspectos principais, a mesma deve ser treinada com imagens exemplo. Para isso, ao invés de treinar a rede com todas as imagens existentes (o que se torna quase impossível visto o crescente número de base de dados principalmente na rede mundial de computadores), um relativo e relevante conjunto de imagens será escolhido e divididas em classes de modo que englobe o maior número possível de representações do mundo real. Essas representações, também chamadas de informação prévia, serão selecionadas em um banco de dados de treinamento Itr chamado banco de dados de treinamento e deverão ser divididos em k grupos bem definidos e distintos onde $Itr_j = \{I_0^j, I_1^j, I_2^j, I_{n-1}^j\}$, $j = 1, ...k$, e:

$$Itr = \bigcup_{i=1}^{k} \ Itr_j, \ Itr_j \cap Itr_l = \emptyset, \ \forall j \neq l$$

Para essa divisão do banco de treinamento, deve-se então aplicar as regras básicas apresentadas na seção 5.4.6 onde as imagens semelhantes serão agrupadas através da análise das distâncias euclidianas entre as mesmas, ou visualmente através de uma pesquisa com usuários do sistema (pesquisa cognitiva), em que os mesmos separarão as imagens do banco de imagens de treinamento em grupos distintos pelo exame intrínseco das figuras.

Considerando um banco de imagens de treinamento pré-classificado, o próximo passo consiste na caracterização das imagens por descritores de baixo nível, tais como apresentados no capítulo 3. Estes descritores podem ser de qualquer forma, desde que atendam as especificações descritas na seção 5.4.6. Para a implementação e verificação do modelo proposto optamos por utilizar três descritores (cor, forma e textura) explanados no próximo capítulo.

Portanto, cada uma das n imagens do conjunto Itr será caracterizada em baixo nível. Desta forma Itr será representado pelo conjunto $Ctr = \{c_0^1, c_1^1, ..., c_{kn-1}^1, c_0^2, c_1^2, ..., c_{kn-1}^2, c_0^k, c_1^k, ..., c_{kn-1}^k\}$, onde c_i^j é o vetor de características que representa a imagem I_i^j e $c_i^j = (f_0^{ij}, f_1^{ij}, f_2^{ij}, ..., f_{p-1}^{ij})$ onde p é o número de valores representando as características de baixo nível. A figura 6.2 exemplifica a formação do vetor de características.

A equação 6.0 representa a função que transforma o conjunto de imagens de treinamento Itr em um conjunto de características de baixo nível Ctr.

$$\begin{aligned} \Im_1 : Itr &\longrightarrow Ctr \\ I_i^j &\longrightarrow \Im(I_i^j) = c_i^j \\ \Im_1 : Itr \subset \mathbb{N}^\alpha \times \mathbb{N}^\beta \times 3 &\longrightarrow Ctr \subset \mathbb{R}^p \end{aligned} \qquad (6.\text{-}2)$$

onde a função $\Im_1$ representa os extratores de características de baixo nível. Esse vetor de características de baixo nível não será usado como parâmetro de verificação de similaridade como ocorre em sistemas de recuperação tradicionais, consequentemente, não será armazenado no banco de dados do sistema. As características serão utilizadas única e exclusivamente para servirem como entradas da rede que irá gerar, através delas, um novo vetor neurosemântico

que procura representar o conhecimento subjetivo humano através dos exemplos pré-definidos na fase de treinamento.

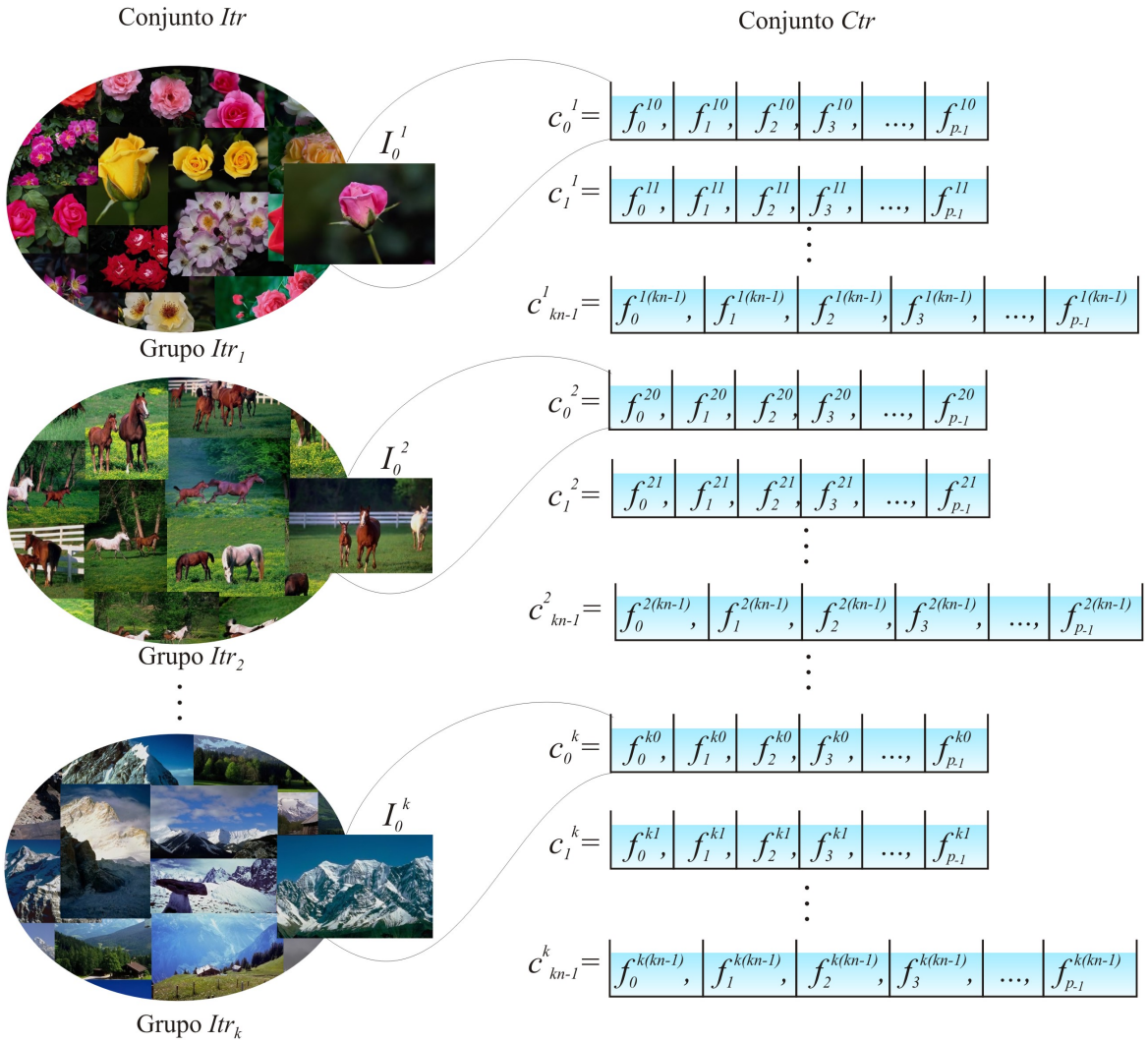


Figura 6.2: Exemplo do processo de extração de características para a formação do vetor de características de baixo nível.

Durante o treinamento da rede neural (estipulado aqui como uma função G) cada vetor de características c_i^j é associado a um vetor alvo ou vetor ideal: $A_i^j = (a_1^{ij}, a_2^{ij}, a_3^{ij}, ..., a_k^{ij})$, onde k é o número de grupos. A função G é definida de modo que sua aplicação em c_i^j produz a um resultado similar a A_i^j, onde:

$$a_l^{ij} = \begin{cases} -1, & l \neq j; \\ 1, & l = j. \end{cases}$$

A figura 6.3 exemplifica o processo de rotulação das imagens através das saídas alvo de acordo com as restrições da Equação 6.-1.

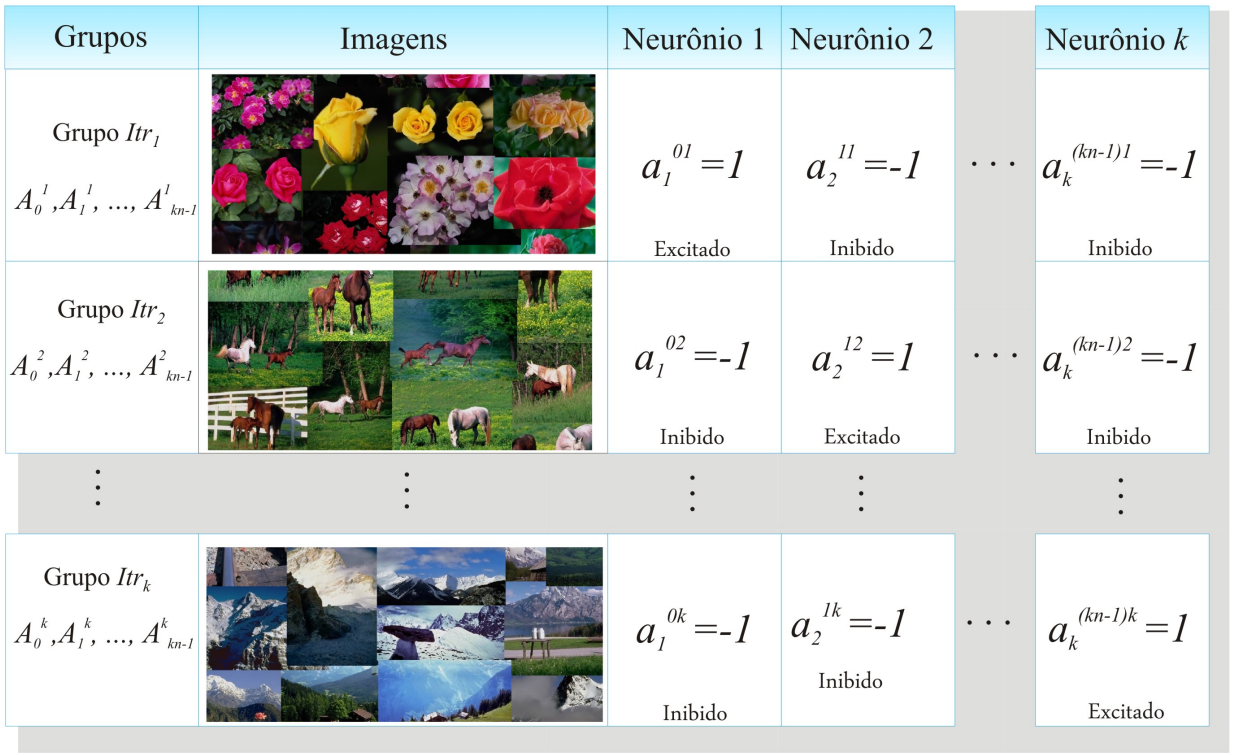


Figura 6.3: Exemplo de classificação de imagens.

Cada imagem I_i do subconjunto Itr_j possui um conjunto de saídas alvo A_i^j. Portanto, cada conjunto dessas imagens de Itr_j possuem um conjunto k de saídas alvo distinto. Logo, o número de neurônios será correspondente ao número de classes k de Itr. Considerando este tipo de rotulação a matriz de saídas alvo sempre será uma matriz quadrada com a diagonal principal 1 e -1 nas demais posições da matriz.

6.2.2 Implementação da Rede Neural G

Para este modelo foi considerada uma rede neural de múltiplas camadas totalmente conectada e de alimentação direta (*feedforward*) como mostrado na Figura 6.4. Todas os valores de c_i^j procedentes do vetor de características extraídas das imagens de treinamento serão conectadas às entradas de cada unidade as quais são independentes entre si e ponderadas por pesos também distintos.

A soma ponderada dos valores do vetor de características e do bias (convencionado como 1 no início do treinamento) irá retornar um valor e o conjunto desses valores nas saídas da rede resultarão em um vetor s, também chamado de "vetor real". Logo $s_j^{(q)}$ denotará o valor de saída

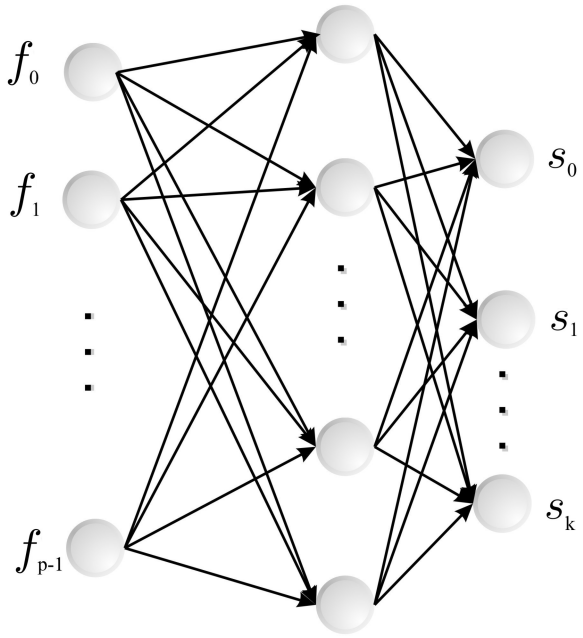


Figura 6.4: Exemplo de rede neural utilizada no modelo proposto.

do j-ésimo neurônio na q-ésima camada de saída e $w_{ij}^{(q)}$ o peso de conexão vindo do i-ésimo neurônio na camada $(q-1)$ para o j-ésimo neurônio na q-ésima camada.

Como foi mencionado no capítulo 5, durante o treinamento, as saídas $s_j^{(q)}$ são afetadas por uma função de ativação $\sigma(net_j^{(q)})$ em que net é gerada pela equação:

$$net_j^{(q)} = \sum_{i=0}^{n_{q-1}} w_{ij}^{(q)} s_i^{q-1}$$

onde $net_j^{(q)}$ corresponde ao nível de ativação do neurônio, n_{q-1} é o número de neurônios na camada $(q-1)$ e σ é a função sigmoidal de ativação dada pela Equação 6.-2. A sua derivada (dada pela equação 6.-1) é utilizada para o treinamento da rede na retificação dos pesos (conforme mostrado no capítulo 5).

$$\sigma(net) = \frac{2}{1+exp(-net)} - 1, \quad (6.\text{-}2)$$

$$\sigma'(net) = \frac{1}{2}[1+\sigma(net)][1-\sigma(net)]. \quad (6.\text{-}1)$$

Treinamento da Rede

Depois de inseridas as entradas e inicializados os pesos com um valor aleatório e o peso do bias com o valor 1, o algoritmo irá iniciar seu aprendizado. Dado um vetor de entrada

$(f_0^{ij}, f_1^{ij}, ..., f_{p-1}^{ij})$ de uma imagem I_i^j na camada de entrada (0-ésima camada), o sinal de saída do r-ésimo neurônio na l-ésima camada na iteração t é dado pela Equação 6.0.

$$s_{rl}^t = \sigma\left(\sum_{m_l=0}^{n_{(l-1)}} w_{(m_l,r,l)}^{(t)} \sigma\left(...\sigma\left(\sum_{i=0}^{n_0} w_{(i,m_1,1)}^{(t)} f_i\right)...\right)\right) \tag{6.0}$$

Para cada imagem I_i^j, uma seqüencia de funções S^t são aplicadas às características de baixo nível c_i^j da imagem I_i^j, ou seja $S^t(c_i^j)$ gerará um vetor de saídas reais $N_{ij}^t = (s_{1L}^t, s_{2L}^t, s_{3L}^t, ..., s_{kL}^t)$ e equivalente à $S^t(f_0, f_1, f_2, ..., f_p) = (s_{1L}^t, s_{2L}^t, s_{3L}^t, ..., s_{kL}^t)$ onde s_{jL}^t é dado pela Equação 6.0 em que $l = L$ e L equivale à última camada da rede.

Na função $S^t(c_i^j)$ é aplicada uma regra de aprendizado utilizando as saídas da rede s_{jL}^t para a geração de novos pesos em G. Neste modelo consideramos o uso de uma regra bastante conhecida na literatura que é a regra da retropropagação utilizando como suporte a regra delta generalizada. Os algoritmos para esse tipo de aprendizado são apresentados no Capítulo 5 dessa dissertação e a figura 6.5 exemplifica o treinamento da rede neural.

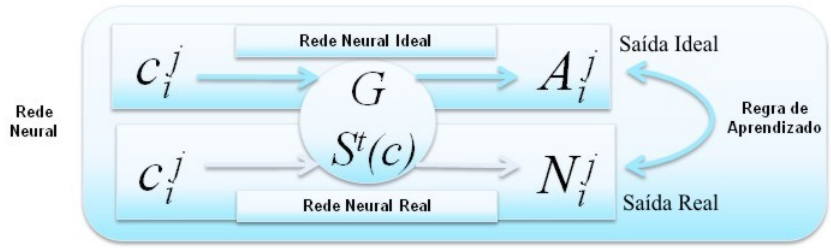


Figura 6.5: Ilustração da aplicação da regra de aprendizado no treinamento da rede neural.

A cada instante (ou iteração) t é aplicada a função (S^t) no conjunto de características c (propagação da rede) e a regra de aprendizado será utilizada para corrigir os pesos de modo a se obter o seguinte objetivo:

$$\lim_{t\to\infty}(S^t) = G$$

Ou seja até que $\lim_{t\to\infty}(S^t(c_i^j)) = A_i^j$ para todas as imagens de Ctr. Assim a regra de aprendizado será aplicada até que todas as entradas estiverem de acordo com as saídas desejadas ou até que se atinja um nível suficiente de corretude em relação aos rótulos dados pelas imagens do conjunto Itr. Quando esse critério for satisfeito a rede G estará pronta pra ser utilizada na fase de caracterização neurosemântica. Nessa fase, as funções de ativação serão retiradas para que as saídas da rede não se restrinjam apenas ao intervalo definido pela função sigmóide. O uso dessa rede bem com as modificações nas estruturas das funções utilizadas na fase de treinamento serão demonstradas na próxima seção.

6.3 Fase 2: Caracterização Neurosemântica

Após a rede estar treinada e sua memória estar carregada de informações pertinentes à fase de treinamento, a próxima etapa será utilizar essas informações em outras imagens de um outro banco de dados (chamado *Ite*). A fase 2 da figura 6.1 apresenta a ilustração do uso da rede neural para a caracterização das imagens neurosemanticamente no banco de testes *Ite*. Para isso, cada imagem I_i desse banco será caracterizada em baixo nível (da mesma forma que na fase de treinamento) formando o vetor $(C_i = (f_0^i, f_1^i, f_2^i, ..., f_{p-1}^i))$ que servirá de entrada da rede neural G.

A rede neural empregada para a formação do vetor de características neurosemântico será a mesma do treinamento, sendo que os pesos serão inicializados com os valores armazenados ao fim da fase de treinamento. Testes com essa rede mostraram que a retirada da função de ativação aumenta o grau de diferenciação das imagens, facilitando a discriminação entre as mesmas. Logo a propagação dos sinais nessa nova rede sem a função de ativação irá gerar um vetor $N_i = (s_1, s_2, s_3, ..., s_k)$, mas, desta vez as saídas procedentes da rede neural não servirão para o reajuste dos pesos e sim para a formação de um vetor de características, convencionado nesse trabalho como vetor de características de alto nível ou neurosemântico. O número de características a serem comparadas dependerá do número de neurônios utilizados na rede, ou seja, o número de classes utilizadas na concepção do projeto da rede neural. A equação 6.1 representa a função que transforma o conjunto de imagens de teste em um conjunto de características neurosemânticas.

$$\begin{aligned} \Im_2 : Cte &\longrightarrow N \\ C_i : \Im_2(C_i) &= N_i \\ G : Ite &\longrightarrow Nh \\ G : \Im_2 &\circ \Im_1 \\ \Im_2 : Nh \subset \mathbb{R}^k &\longrightarrow Cte \subset \mathbb{R}^p \end{aligned} \tag{6.-3}$$

Onde Nh é o conjunto de características de neurosemânticas, a função $\Im_1$ representa os extratores de características de baixo nível e a função G é a função de transformação que inclui a rede neural modificada.

6.4 Fase 3: Processo de Recuperação

Após as imagens do banco de dados de testes *Ite* serem caracterizadas pela rede, o sistema estará pronto para ser utilizado na fase 3 da figura 6.1. O modelo completo do sistema de recuperação é semelhante ao modelo típico de sistema de Recuperação de Imagens Baseada em Conteúdo onde o usuário inserirá no sistema uma imagem consulta Q e a mesma passará por um extrator de características de baixo nível formando o vetor $c_q = (f_0^q, f_1^q, f_2^q, ..., f_{p-1}^q)$ que servirá de entrada para a rede neural G formando o vetor de características $N_q = (s_1^q, s_2^q, s_3^q, ..., s_k^q)$ neurosemânticas para representar a imagem consulta Q. Esse vetor N_q relacionado à imagem consulta Q é então comparado aos outros vetores $N_i = (s_1^i, s_2^i, s_3^i, ..., s_k^i)$ relacionados à cada

imagem I_i do banco de testes Ite , a métrica de distância utilizada entre os vetores da imagem consulta com uma imagem I do banco de testes foi:

$$d(Q, I_i) = (\sum_{j=1}^{k} (s_j^q - s_j^i)^2)^{1/2}$$

Onde s_j^q e s_j^i correspondem ao valor da j-ésima dimensão do espaço de características neurosemânticas de Q e I respectivamente. As menores distâncias entre o vetor da imagem consulta e os vetores do banco de dados formarão um ranking de imagens mais semelhantes que são retornadas ao usuário como resultado da consulta. A figura 6.6, mostra com mais detalhes, o modelo geral do sistema de recuperação proposto para o processo de recuperação.

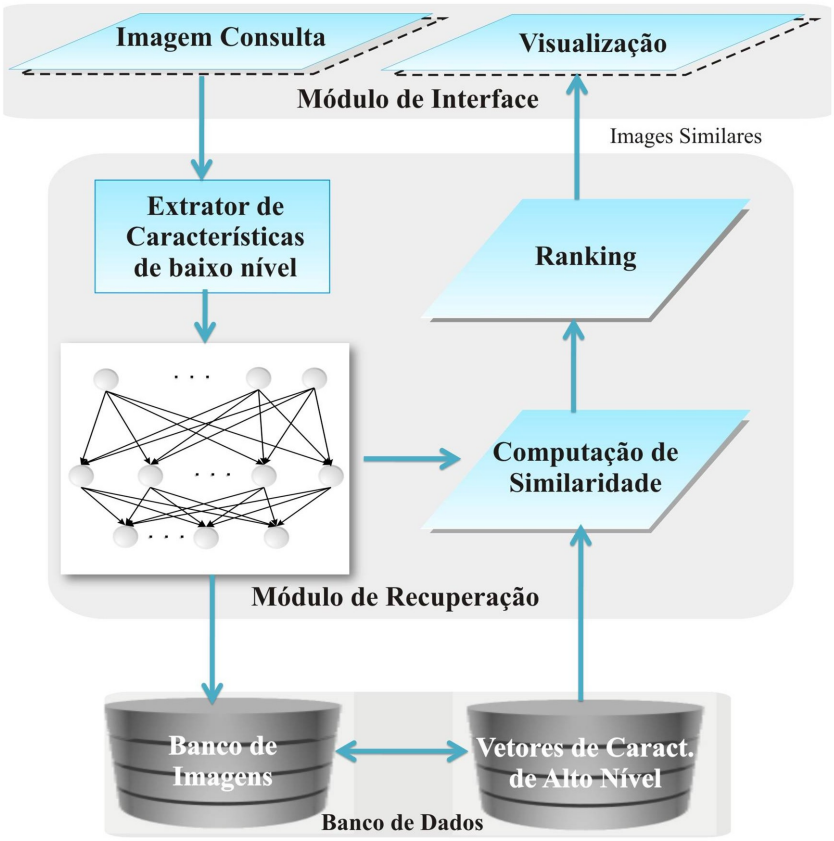


Figura 6.6: Projeto conceitual do modelo proposto para a recuperação de imagens utilizando Redes Neurais Artificiais.

Esse modelo poderá ser adaptado para qualquer tipo de caracterização em baixo nível bem como para um diferente número de entradas desde que as mesmas sejam fixadas e o mesmo procedimento seja adotado durante a fase de consulta. A seguir é apresentado o algoritmo completo para o modelo proposto.

Algoritmo 4 Modelo Proposto

Dados dois Bancos de Dados Itr e Ite, bancos de dados de treinamento e de testes respectivamente;
{/*Fase de Treinamento*/}
for Cada grupo Itr_j **do**
  for Cada imagem I_i^j de Itr_j **do**
    for Cada característica p de I_i^j **do**
      {/*Caracterização em Baixo Nível*/}
      $f_p^{ij} \leftarrow$ Extrair características. (Cor, Forma, Textura, etc.);
      $c_i^j \oplus \leftarrow f_p^{ij}$
    end for
  end for
  $Ctr_j \oplus \leftarrow c_i^j$
end for
for Cada c_i^j de Ctr_j **do**
  Associe c_i^j a uma saída alvo A_i^j;
end for
t $\leftarrow$ 0; {Iteração t}
while Erro da rede não é satisfatório. **do**
  t $\leftarrow$ t+1;
  for Cada grupo Itr_j **do**
    for Cada imagem I_j^i de Itr_j **do**
      Propague c_i^j na rede neural e obtenha o conjunto de saídas N_{ij}^t;
      Retifique os pesos da rede neural G com as entradas c_i^j e seus respectivos alvos A_i^j para fazer s_j^t próximo a_j;
    end for
  end for
end while
Salve G
{/*Caracterização Neurosemântica*/}
for Cada imagem Ite_i **do**
  for Cada caracacterística p de Ite_i **do**
    $f_p^i \leftarrow$ Extrair características. (Cor, Forma, Textura, etc.);
    $c_i \oplus \leftarrow f_p^i$
  end for
  Propague c_i na rede neural e obtenha o conjunto de saídas N_i
  Armazene o conjunto de vetores N_i como vetores de características de neurosemântica correspondentes ao conjunto de imagem Ite_i;
end for
{/*Processo de Recuperação*/}
Dada uma imagem consulta q;
for Cada característica p **do**
  $f_q \leftarrow$ Extrair características. (Cor, Forma, Textura, etc.);
  $c_q \oplus \leftarrow f_q$;
end for
Propague c_q na rede neural e obtenha o conjunto de saídas N_q;
Compare N_q como os outros vetores da base de dados Ite usando uma função distância;
Mostre o *ranking* de imagens;

Capítulo 7

Resultados Experimentais

Este capítulo é dedicado às avaliações experimentais de performance para a caracterização das imagens neurosemânticamente proposta neste trabalho. Para isso foi implementado um sistema CBIR e utilizado como ferramenta para os testes em várias bases de dados. Os experimentos aqui realizados têm os seguintes objetivos: avaliar os resultados obtidos pelo modelo proposto em uma base de imagens grande e diversificada, analisar a eficiência da redução e transformação das características de baixo nível que serviram de entrada para a rede neural e comparar nosso método com outros descritores apresentados em [Deselaers et al. 2008].

Para efeito de comparação dos resultados obtidos foram utilizados como banco de imagens de treinamento: *Corel-1000* e como bancos de dados de teste: *BD-12750*, *Vistex-167* e *ZuBuD*. Tais bases serão detalhadas a seguir. Exemplos de resultados visuais serão mostrados bem como os resultados obtidos com as medidas de avaliação apresentadas na seção 2.5 deste trabalho. A seguir alguns testes realizados serão descritos para ilustrar a a boa performance da proposta de caracterização apresentada nesse trabalho.

7.1 Banco de Imagens de Treinamento

Conforme explicitado no capítulo anterior, para a implementação do modelo proposto, a primeira etapa consiste na escolha das imagens para o treinamento da rede. O banco de dados escolhido foi o banco de imagens *Corel-1000* (também conhecido com *Wang Database*) que é um subconjunto do banco de dados da *Corel Corporation*, contendo 1000 imagens com resolução de 384×256 e também 256×384 *pixels*. Este banco de imagens (que pode ser obtido em [Corel Database]) possui 100 imagens de 10 classes distintas, são elas: África, praia, edifícios, ônibus, dinossauros, elefantes, flores, comidas, cavalos e montanhas. A Figura 7.1 contém uma amostra deste banco de dados.

Como o banco possui 10 classes distintas, a rede neural a ser implementada deverá possuir 10 neurônios de saída que representarão essas 10 classes neste banco de dados em específico.

7.2 Caracterização em Baixo Nível

Os descritores empregados para a formação dos vetores de características de baixo nível e utilizados como suporte no treinamento da rede neural bem como na caracterização neurosemântica foram: momentos de cores globais [Stricker and Orengo 1995], vizinhança de textura [Laaksonen et al. 2000] e histogramas de direção da borda [Brandt 1999] totalizando 25 posições nos vetores de entrada da rede neural. Esses descritores foram detalhados no Capítulo 3 desta dissertação.

Os vetores resultantes da caracterização em baixo nível não serão utilizados como parâmetro de similaridade como ocorre em sistemas tradicionais de recuperação, consequentemente, eles não serão armazenados na base de dados. Essas características serão mapeadas pela rede neural, que irá transformá-las em um novo vetor chamado vetor de características neurosemânticas.

7.3 Parâmetros da Rede Neural

A rede neural implementada neste trabalho é uma rede neural de múltiplas camadas e de alimentação direta. O treinamento utilizado foi o treinamento por retropropagação do erro mostrado no Capítulo 5 desta dissertação. A rede neural possui 26 neurônios na camada de entrada (bias mais vetor de características de baixo nível), 18 neurônios na camada intermediária e 10 neurônios na camada de saída (características neurosemânticas), i.e., a rede neural implementada possui uma estrutura do tipo $26 - 18 - 10$. Para o treinamento da rede a função de ativação utilizada foi a função sigmoidal simples e a taxa de aprendizado utilizada foi de 0.003.

7.4 Bancos de Imagens de Teste

Na condução dos experimentos foram utilizados três bancos de dados de teste. Uma descrição seguida de uma amostra de cada um destes bancos de dados é apresentada a seguir:

BD-12750 Este banco de dados foi construído pelo grupo de pesquisa *VIPIGRAF* da Universidade Federal de Uberlândia e possui 12750 imagens reais cobrindo uma ampla variedade de categorias, tais como, texturas uniformes, nuvens, pôr-do-sol, oceanos, paisagens, edifícios, mamíferos, peixes, aves, répteis, árvores, flores, motocicletas, carros, aviões, faces humanas, etc. O banco de dados está em construção e até o momento não é agrupado em categorias. A Figura 7.2 contém uma amostra desta coleção.

ZuBuD O banco de imagens "*Zurich Buildings Database for Image Based Recognition*" (ZuBuD) é um banco de dados criado pelo Instituto Federal de Tecnologia de Zurique e pode ser encontado em [ZuBuD Database]. O banco de dados consiste de 1005 imagens divididas em 201 categorias e cada categoria é representada por um prédio onde cada imagem dessa categoria é uma fotografia deste prédio visto de ângulos diferentes e em alguns casos sob diferentes condições climáticas ou tiradas por diferentes câmeras fotográfica. Dada uma imagem consulta, somente imagens mostrando exatamente o mesmo prédio são consideradas como relevantes àquela consulta. Algumas imagens exemplo desta base são mostradas na Figura 7.3.

Vistex-167 O banco de imagens *Vistex-167* é um banco de dados de 167 imagens coloridas, com resolução de 128×128 *pixels*, de texturas homogêneas. O banco de dados é composto de 19 categorias, variando de 3 a 20 imagens cada e pode ser obtido em [Vistex Database]. A Figura 7.4 contém uma amostra deste banco de dados.

7.5 Resultados Experimentais

Os experimentos foram divididos de acordo com as bases de dados escolhidas. As seções seguintes descrevem os resultados obtidos para cada uma dessas bases.

7.5.1 Resultados para a base *BD-12750*

Estes experimentos podem ser vistos como os mais importantes, visto que é aplicado na base de dados *BD-12750* a qual reflete a diversidade de imagens que compõem o grande repositório de imagens que é a *web*. Por não ser uma base clusterizada, não foi possível traçar curvas de precisão-revocação, mas foram feitas várias consultas e os resultados visuais serão mostrados e analisados a seguir.

O primeiro experimento para essa base de imagens explora a performance da rede neural na caracterização de imagens quando o procedimento é realizado com duas arquiteturas de redes neurais. A Figura 7.5 apresenta resultados quando o procedimento proposto é realizado utilizando uma rede neural de camada única (a) e utilizando a rede neural de múltiplas camadas proposta neste capítulo (b). O banco de dados de treinamento e de teste são os mesmos para ambos os casos. Nos experimentos foram omitidas as primeiras imagens retornadas no *ranking* visto que são as próprias imagens consulta, quando as mesmas pertencerem ao banco de dados. Analisando os resultados obtidos pela rede neural de camada única (Figura 7.5 (a)) pode-se notar que apenas 7 das 15 primeiras imagens retornadas pertencem à mesma categoria da imagem consulta. Já nos resultados obtidos utilizando a rede neural de múltiplas camadas (Figura 7.5 (b)) pode-ser observar que 12 das 15 primeiras imagens retornadas pertencem à mesma categoria da imagem consulta. É observado, então, que a rede neural multicamadas apresenta resultados mais satisfatórios que a rede de camada única, pois a rede multicamadas possui, entre outros benefícios, uma melhor probabilidade de generalização. Um dos fatos que ajudam nessa tarefa é a eliminação de redundâncias na representação de informação que torna o sistema mais tolerante às imperfeições de recuperação. Todos os outros experimentos mostrados nesse capítulo utilizam a rede neural de múltiplas camadas explanada nas seções anteriores.

No segundo experimento vamos ilustrar o resultado da recuperação considerando a caracterização das imagens (tanto consulta como o banco) feita de duas maneiras: a primeira utilizando a caracterização em baixo nível (cor, forma e textura) e a segunda utilizando a caracterização neurosemântica via redes neurais. Os resultado obtidos são mostrados na figura 7.6. A figura 7.6 - (a) apresenta os resultados da busca quando a consulta é caracterizada pelas mesmas características de baixo nível utilizadas para formar os vetores de entrada da rede neural (*FOriginal*) (9 momentos de cores, 8 características das vizinhanças de textura e 8 posições do histograma de direção da borda, totalizando 25 características) e a Figura 7.6 - (b) apresenta os

resultados obtidos pelo método proposto utilizando os vetores de características neurosemânticas (*FProposed*) (10 características). Analisando os 16 primeiros resultados retornados na figura 7.6 - (a) nota-se que apenas a terceira imagem retornada pertence à mesma categoria da imagem consulta (flores). Já na imagem 7.6 - (b) pode-se notar que das 16 primeiras imagens retornadas 12 imagens pertencem à mesma categoria da imagem consulta. Esse exemplo ilustra que a caracterização neurosemântica via redes neurais apresenta melhores resultados que as características de baixo nível e ainda utiliza um menor número de características para representar a mesma imagem.

O terceiro experimento ilustra a performance do modelo proposto para consultas de diferentes categorias e é mostrado na figura 7.7. Dez diferentes imagens consulta foram utilizadas e os resultados mostram a adaptação neurosemântica do modelo proposto em diferentes níveis de requerimentos do usuário, visto que nenhuma imagem deste banco foi utilizada durante a fase de treinamento. Analisando os resultados obtidos, pode-se notar que a distribuição de cor e as transformações geométricas foram preservadas. Considerando que a busca por conceitos semânticos é altamente complexa, os resultados se mostraram efetivos diante da dificuldade de localizar atributos comuns a todas as imagens.

Os resultados do quarto experimento dentro do banco de dados *BD-12750* são apresentados na figura 7.8 onde as imagens consulta são imagens para a categoria faces. Pode-se notar que, apesar de não ter sido utilizada nenhuma imagem de face na fase de treinamento, quase todas as imagens retornadas paras as consultas em níveis de cinza são imagens de faces. Se considerarmos que, diante da diversidade de imagens que compõem o banco, o retorno de imagens de face para consultas de imagens de face é relevante (independente do indivíduo) os resultados se mostram promissores. Para as imagens em tons de cinza pode-se notar que a distribuição de cor foi predominante principalmente em relação ao fundo da imagem. Pode-se observar que nas 4 primeiras consultas, algumas imagens retornadas como resultado são imagens de motos e peixes em tons de cinza com o fundo branco. Para as imagens de faces coloridas a precisão dos resultados (considerando todas as imagens de faces do banco como sendo relevantes para esse tipo de consulta) não foi tão boa quanto para imagens em tons de cinza mas, ainda assim, foram retornadas várias imagens de faces coloridas.

O quinto experimento mostrado na figura 7.9 apresenta resultados obtidos quando a imagem consulta é uma imagem danificada e/ou rotacionada. Em todos os resultados apresentados, as 16 primeiras imagens retornadas pertencem a categoria mostrando uma acurácia de 100% do método. Este experimento mostra a robustez do sistema proposto mesmo quando a imagem consulta contém estragos. Observando, em particular, a consulta do avião rotacionado, pode-se notar que a mesma imagem, sem o rotacionanemto nem os estragos aparece na sexta posição do *ranking*.

Para examinar o modelo proposto de uma forma mais genérica, nós conduzimos experimentos com 30 consultas randômicas utilizando o conjunto de vetores de características de baixo nível e o conjunto de vetores de características neurosemânticas. A curva *FOriginal* (em azul) está relacionada aos resultados obtidos com os vetores de características de baixo nível utilizados como entrada da rede. A curva *FProposed* (em vermelho) está relacionada aos resultados obtidos pelo método proposto utilizando os vetores de características neurosemânticas. Como a base de dados não é clusterizada optamos por adaptar a curva de precisão-revocação, obtendo uma nova medida, onde o gráfico representa o número de imagens relevantes recuper-

adas pelo número de imagens recuperadas. Neste caso são analisadas as imagens retornadas nas primeiras 20 posições do ranking. Pode-se notar pela figura 7.10 que no modelo proposto houve uma melhora de 20% nos resultados comparados com a curva obtida pelos resultados utilizando as características de baixo nível. Na curva *FProposed* há um bom "balanceamento" tanto para baixos quanto para altos níveis de revocação. A curva *FOriginal* apresenta um declive grave no topo do ranking que afeta negativamente na inspeção visual dos resultados o que, em situações reais de busca, não satisfaz a maioria dos interesses dos usuários.

O principal objetivo destes experimentos foi demonstrar que o modelo proposto trabalha bem com uma grande diversidade de imagens das mais variadas categorias. No entanto não foi encontrada uma base de dados suficientemente grande como a base *BD-12750* e tão diversificada como esta que afim de ilustrar medidas de precisão-revocação fosse classificada. Por isso não foi possível a realização de testes utilizando as medidas de avaliação apresentadas na seção 2.5 dessa dissertação. No entanto, escolhemos duas bases pré-classificadas, que apesar de serem pequenas e não diversificadas podem ilustrar a acurácia do método utilizando essas medidas de avaliação clássicas. A seguir os resultados obtidos para estas duas bases são demonstrados.

7.5.2 Resultados para a base *ZuBuD*

Esta seção apresenta a avaliação do modelo proposto na base de dados *ZuBuD*. Optamos por utilizar esse banco, pois, o artigo [Deselaers et al. 2008] implementa e avalia vários descritores de imagens, alguns detalhados no capítulo 3 dessa dissertação. Com isso foi possível comparar a taxa de erro dos vários descritores apresentados no artigo com o modelo apresentado nesse trabalho. Dentre esses descritores está o conjunto de descritores utilizados como entrada na rede (*FOriginal*) e o descritor obtido pelo modelo proposto (*FProposed*). A figura 7.11 apresenta o gráfico com as taxas de erros para cada um desses descritores. No erro de classificação ER a imagem consulta é retirada do banco de imagens e é considerada apenas a imagem mais similar de acordo com a função distância aplicada [Deselaers et al. 2008]. Uma imagem consulta q só será considerada como classificada corretamente se a primeira imagem recuperada for relevante, ao contrário, a imagem não foi classificada corretamente. O cálculo de ER é dado pela seguinte equação:

$$ER = \frac{1}{Q} \sum_{q \in Q} \begin{cases} 0, & \text{se a imagem mais similar é relevante/pertence a classe correta;} \\ 1, & \text{caso contrário.} \end{cases}$$

onde Q é o conjunto de consultas q.

Pode-se notar que apesar da taxa de erro apresentada pelo modelo proposto não estar entre as melhores, a rede melhorou em 11% com relação ao vetor obtido pelos descritores de características de baixo nível que serviram de entrada para a rede. Acreditamos que essa taxa não foi satisfatória para avaliar o método nesse banco por se tratar de imagens muito específicas e bem similares e o nosso método foi proposto para generalizar a representação de imagens dos mais variados tipos conforme foi ilustrado nos resultados da seção anterior.

Deselaers et. al. também apresenta no artigo [Deselaers et al. 2008] a média das precisões para todas as consultas. Os resultados obtidos por essa medida são apresentados na figura 7.12.

Pode-se observar que o modelo proposto está entre os 5 melhores descritores avaliados. Apesar da taxa de erro não ter sido tão satisfatória consideramos que a aplicação do método proposto nesse banco de imagens obteve bons resultados visto que a precisão média demonstra aspectos tanto da precisão quanto da revocação e é sensível a todo o *ranking* ao contrário da taxa de erro que analisa somente a primeira imagem retornada.

A Figura 7.13 ilustra resultados de recuperação para três imagens consulta dentro da base de dados *ZuBuD*. Pode-se notar na primeira consulta que foram retornadas todas as imagens pertencentes à mesma categoria da consulta já nas primeiras posições. As imagens seguintes não pertencem à mesma categoria da consulta mas são visualmente bem similares, inclusive, as duas ultimas imagens da primeira consulta (7ª e 8ª imagens retornadas), apesar de não estarem classificadas no banco como sendo da mesma categoria da imagem consulta, são fotografias da mesma casa em ângulos diferentes.

A Figura 7.14 demonstra resultados que interferem negativamente na taxa de erro pois, apesar de as imagens retornadas possuirem aspectos de cor e forma similares, a primeira imagem retornada no *ranking* não pertence à mesma classe da imagem consulta. Analisando os resultados, pode-se observar que nas demais posições do *ranking* existem imagens de mesma categoria da imagem consulta, o que pode explicar a melhora da performance da caracterização neurosemântica pelo cálculo da precisão média.

7.5.3 Resultados para a base *Vistex-167*

Esta seção descreve os experimentos realizados na caracterização neurosemântica da base de dados *Vistex-167*. Nesta abordagem avaliamos a eficiência do método proposto utilizando as curvas de precisão-revocação explanadas na seção 2.5 dessa dissertação e o resultado é mostrado na figura 7.15. A curva *FOriginal* (em azul) está relacionada aos resultados obtidos com os vetores de características de baixo nível e a curva *FProposed* (em vermelho) está relacionada aos resultados obtidos pelo método proposto utilizando os vetores de características neurosemânticas. Pode-se notar que houve uma melhora de aproximadamente 5% no modelo proposto para esse banco além de uma redução de 60% na dimensionalidade do vetor, fato que, para grandes bancos de imagens, pode significar uma grande redução no tempo de processamento.

A Figura 7.16 ilustra resultados de recuperação para três imagens consulta dentro da base de dados *Vistex-167*. O *ranking* é formado pela imagem consulta e pelas 8 primeiras imagens mais similares de acordo com o critério de similaridade apresentado no capítulo anterior. Os resultados mostram que várias imagens retornadas pertencem à mesma categoria da imagem consulta, e mesmo aquelas imagens que não pertencem à mesma categoria da imagem consulta possuem aspectos de cor, forma e textura bem similares a elas.

7.6 Considerações Finais

Nesse capítulo o modelo proposto foi avaliado sob diferentes aspectos e em várias bases de dados. A técnica proposta utilizou como base de treinamento o banco de dados *Corel-1000* por ser um banco de dados já classificado e que cobre uma razoável faixa de categorias. Estudos sobre o impacto da formação dessa base de dados, bem como o número de classes e imagens que

a compõe serão estudados em trabalhos futuros. Atualmente as bases de dados clusterizadas são muito específicas e a base *Corel-1000* foi a que melhor satisfez os requisitos do modelo.

O modelo proposto foi testado em 3 bases de dados diferentes: *BD-12750*, *ZuBuD* e *Vistex-167*. Como a base de imagens *BD-12750* não é clusterizada, não foi possível fazer uma análise global dos resultados, mas várias consultas foram realizadas e demonstraram que o modelo funciona bem para esse tipo de base composta pelas mais variadas categorias. Para o banco de dados *ZuBuD* utilizamos como parâmetro de comparação outros descritores apresentados pelo artigo [Deselaers et al. 2008]. Os resultados mostraram que no modelo proposto houve uma melhora em relação às características originais que serviram de entrada para a rede neural tanto na taxa de erro quanto na precisão média. Analisando a precisão média pôde-se observar que, apesar de a base de dados ser muito específica (apenas prédios) o modelo proposto ficou entre os 5 melhores descritores apresentados. Todos os resultados mostraram informações úteis para a investigação futura de alguns aspectos que podem ser estudados. A aplicação do modelo proposto para outros descritores, por exemplo, como os apresentados em [Deselaers et al. 2008] pode dar uma direção no uso das redes neurais na transformação de qualquer tipo de característica.

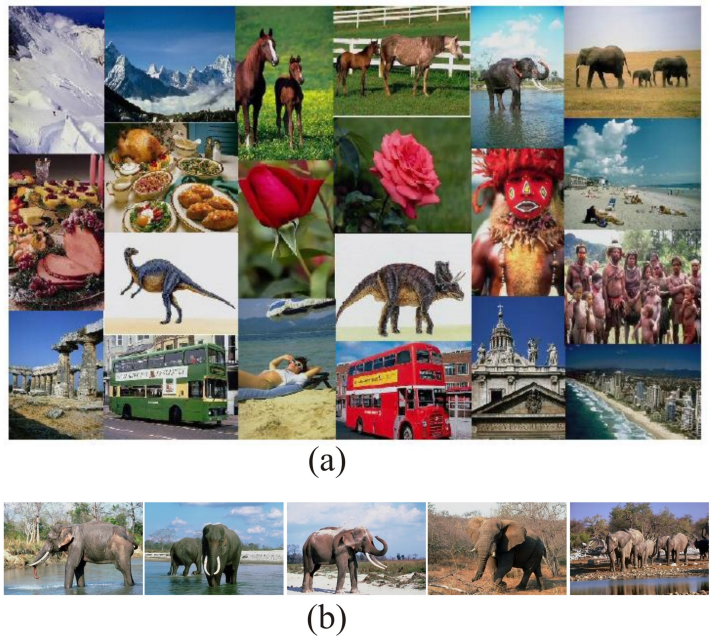


Figura 7.1: (a) Amostra do banco de dados Corel-1000. (b) Exemplo de um conjunto de imagens da base pertencentes a uma mesma categoria (elefantes).

Figura 7.2: Amostra do banco de dados BD-12750.

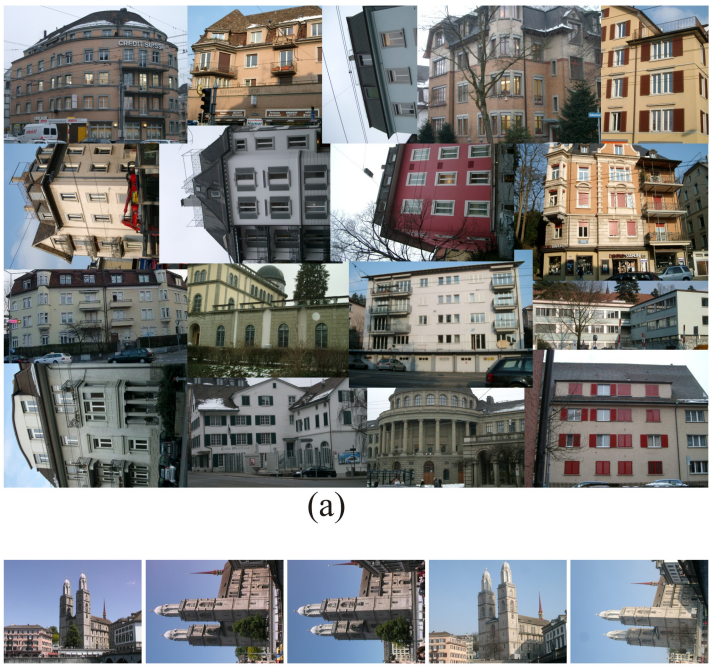


Figura 7.3: (a) Amostra do banco de dados *ZuBuD*. (b) Exemplo de um conjunto de imagens da base pertencentes a uma mesma categoria.

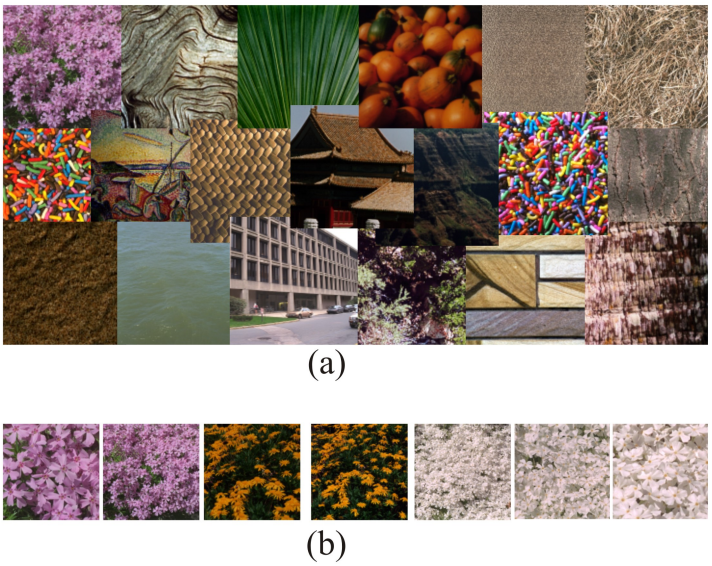
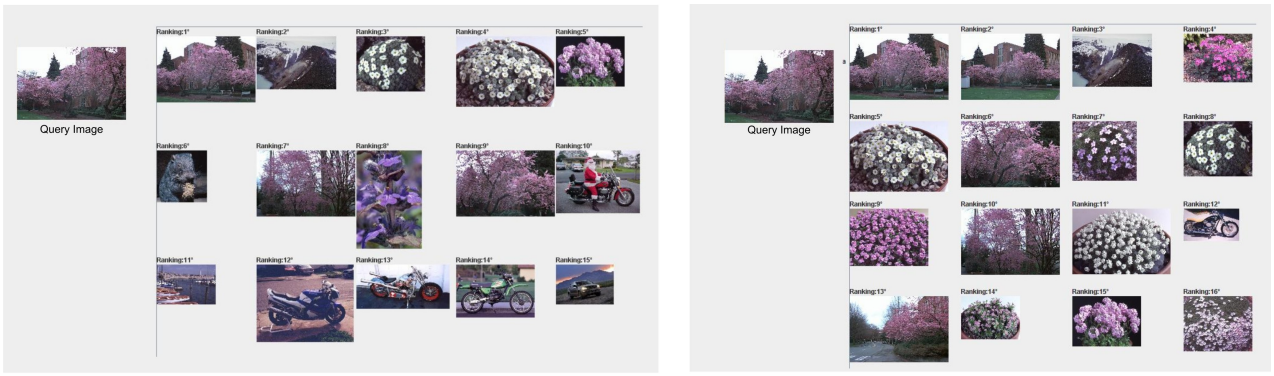
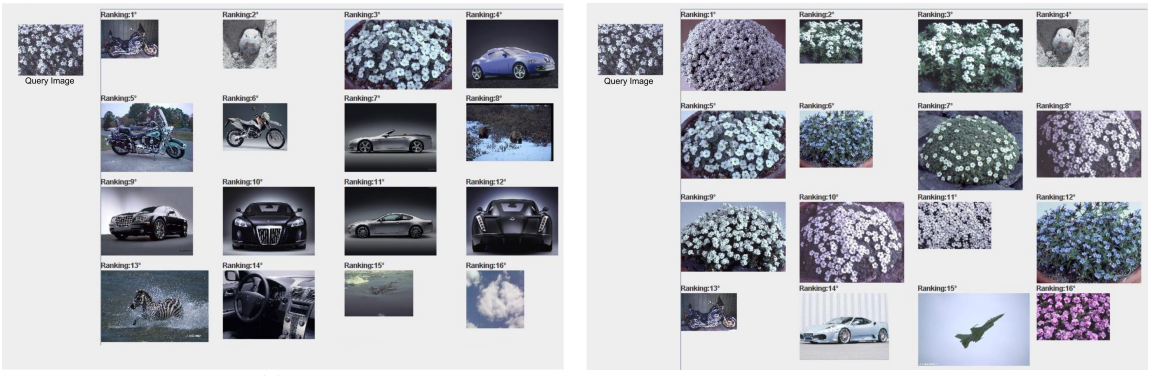


Figura 7.4: (a) Amostra do banco de dados *Vistex-167*. (b) Exemplo de um conjunto de imagens da base pertencentes a uma mesma categoria (flores).



(a) (b)

Figura 7.5: *Ranking* das primeiras imagens obtidas utilizando uma rede neural simples (a) e uma rede neural de múltiplas camadas para formar o vetor de características neurosemânticas (b).



(a) (b)

Figura 7.6: *Ranking* das primeiras imagens obtidas utilizando características de baixo nível (*FOriginal*) (a) e utilizando os vetores de característica neurosemânticas obtidos pelo modelo proposto (*FProposed*) (b).

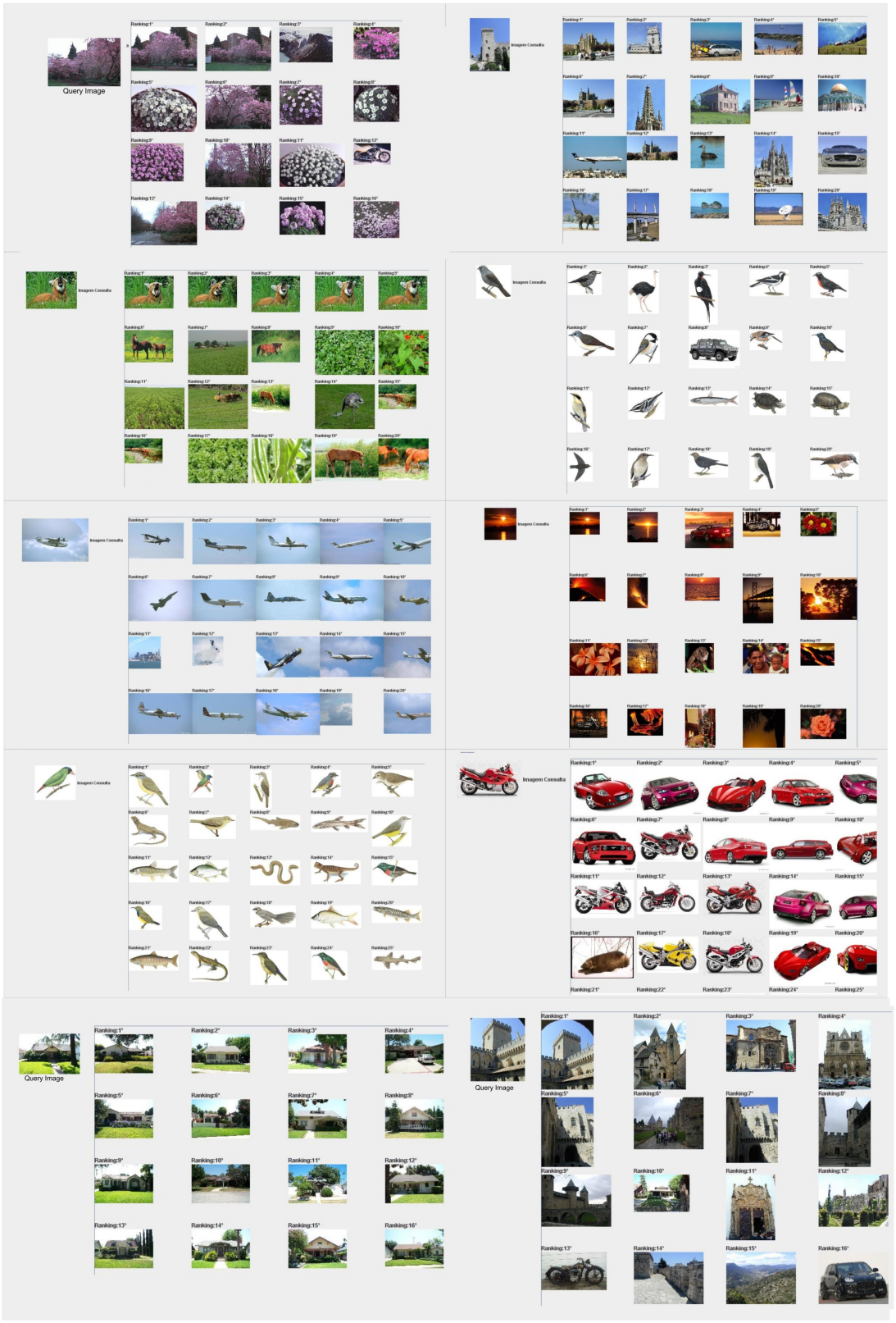


Figura 7.7: Resultados de busca obtidos pelo modelo proposto na base de imagens *BD-12750* para diferentes categorias.

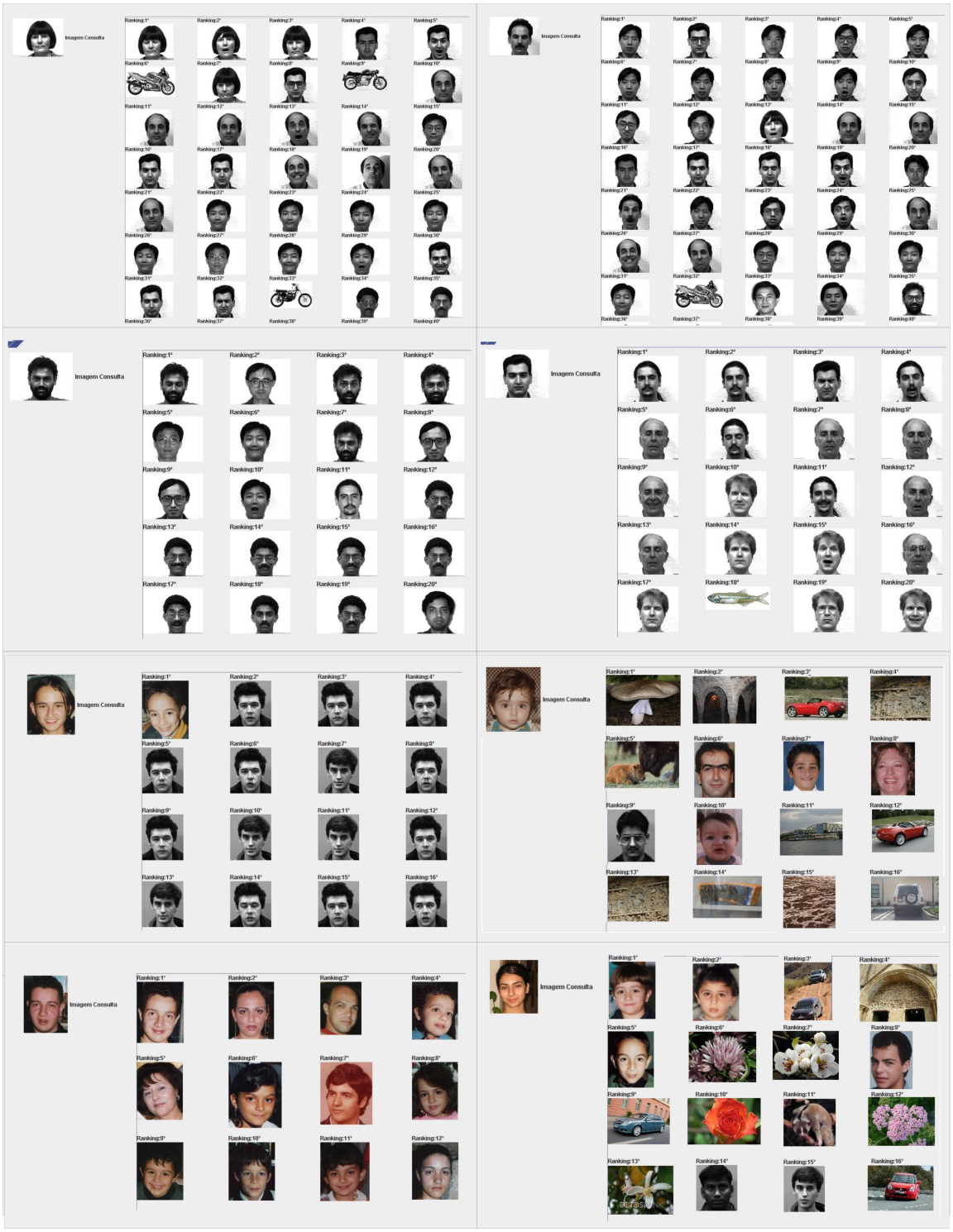


Figura 7.8: Resultados de busca de faces obtidos pelo modelo proposto na base de imagens *BD-12750*.

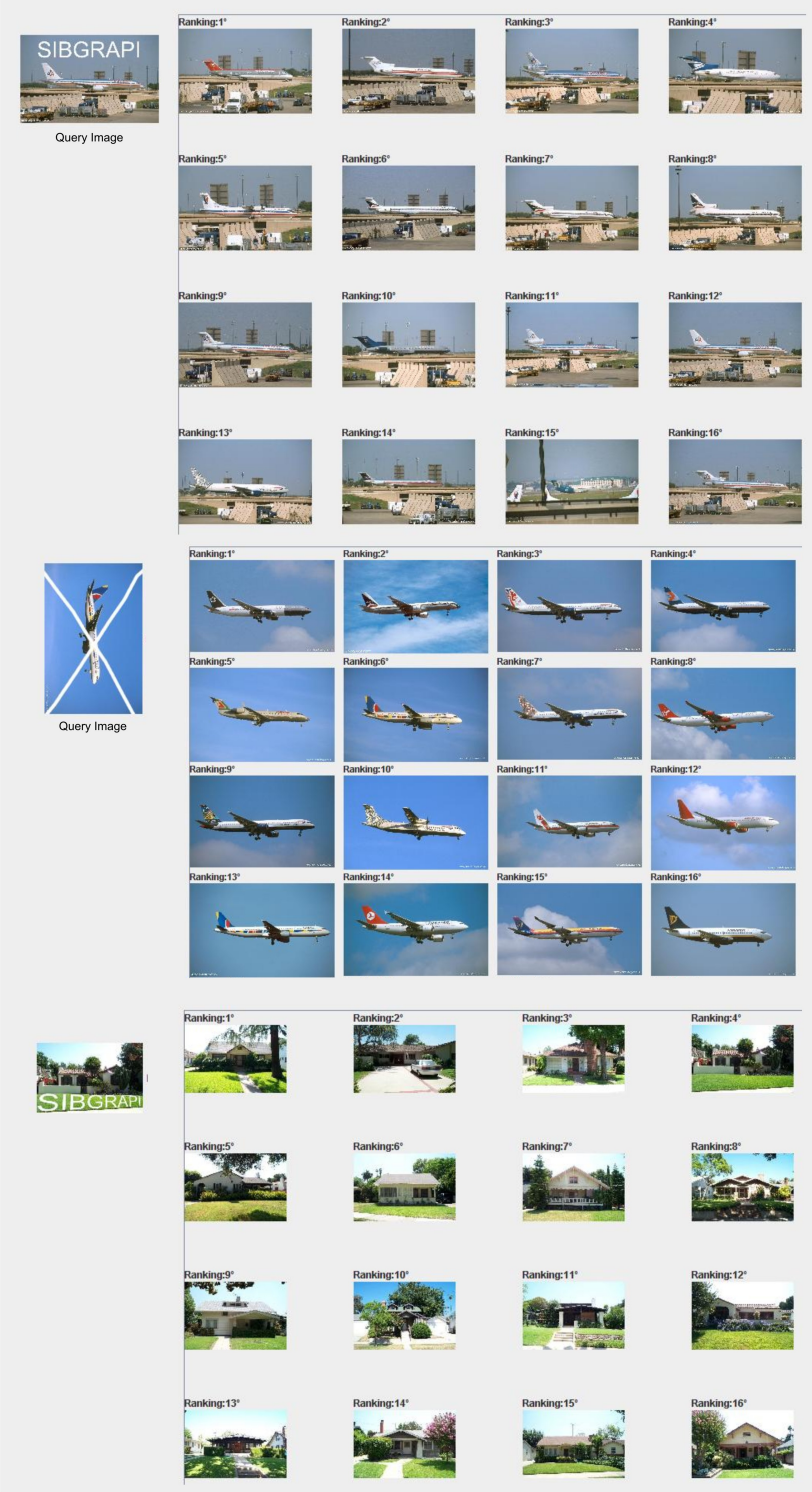


Figura 7.9: Resultados de busca obtidos pelo modelo proposto na base de imagens *BD-12750* para imagens modificadas/danificadas.

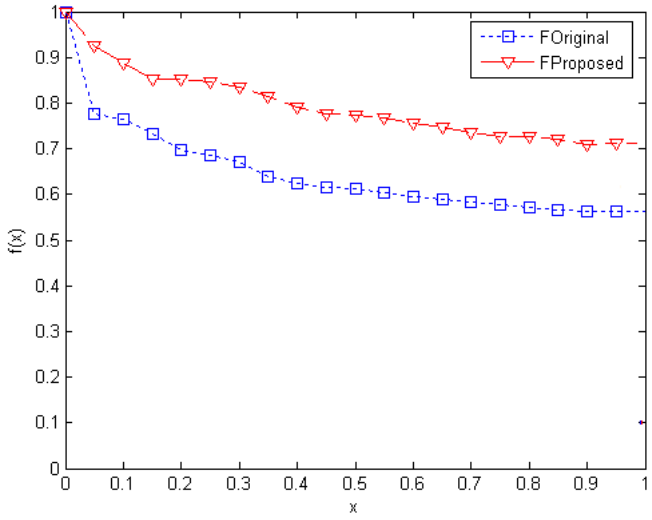


Figura 7.10: Curvas resultantes dos vetores de característica de baixo nível (*FOriginal*) e dos vetores de características neurosemânticas *FProposed*

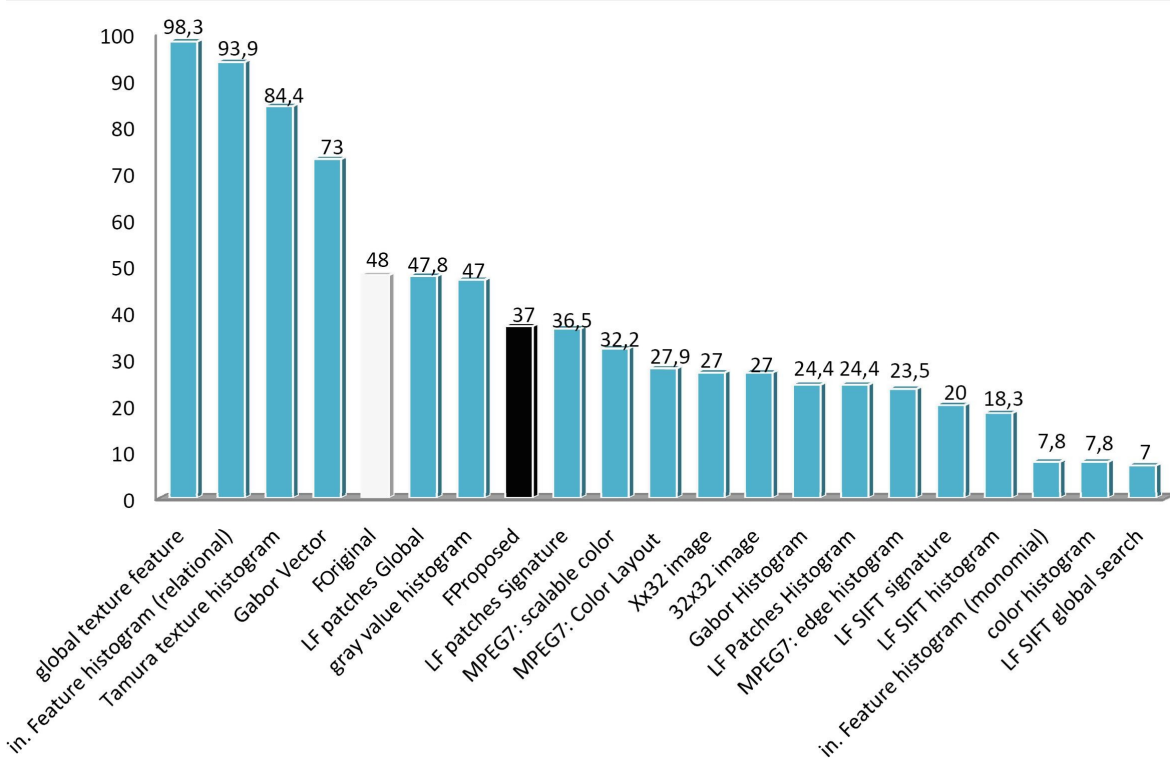


Figura 7.11: Taxa de erro de classificação para vários descritores no banco de dados *ZuBuD*.

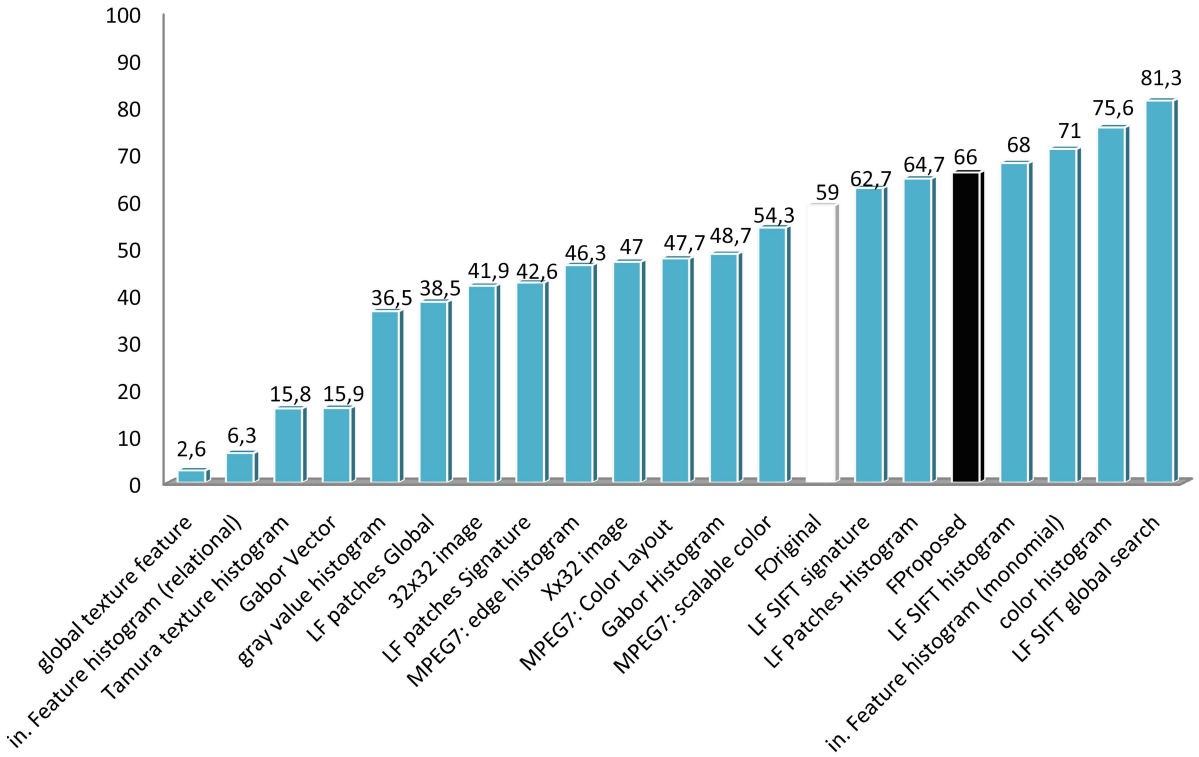


Figura 7.12: Precisão média para vários descritores no banco de dados *ZuBuD*.

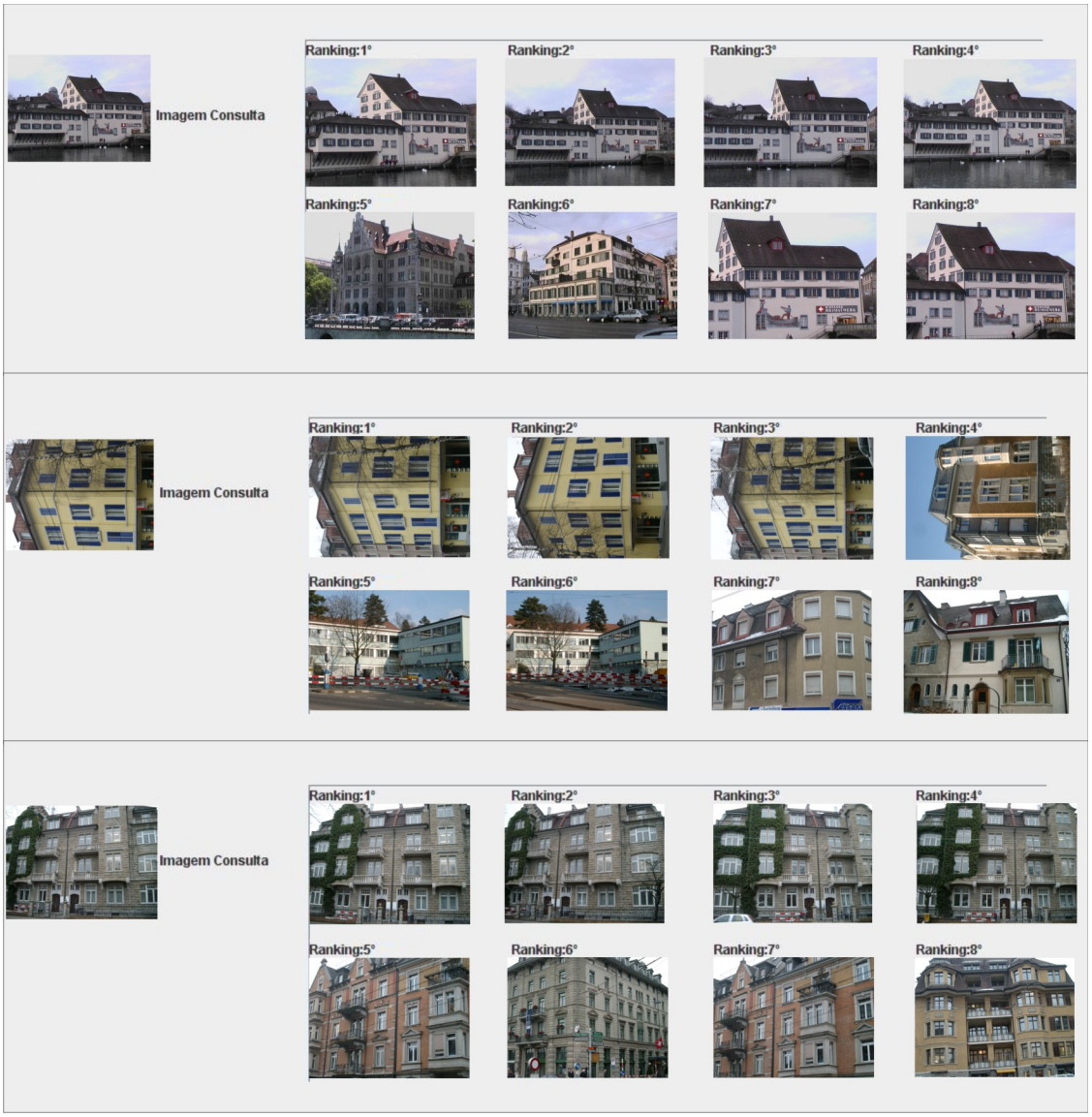


Figura 7.13: Resultados de busca obtidos pelo modelo proposto na base de imagens *ZuBuD*.

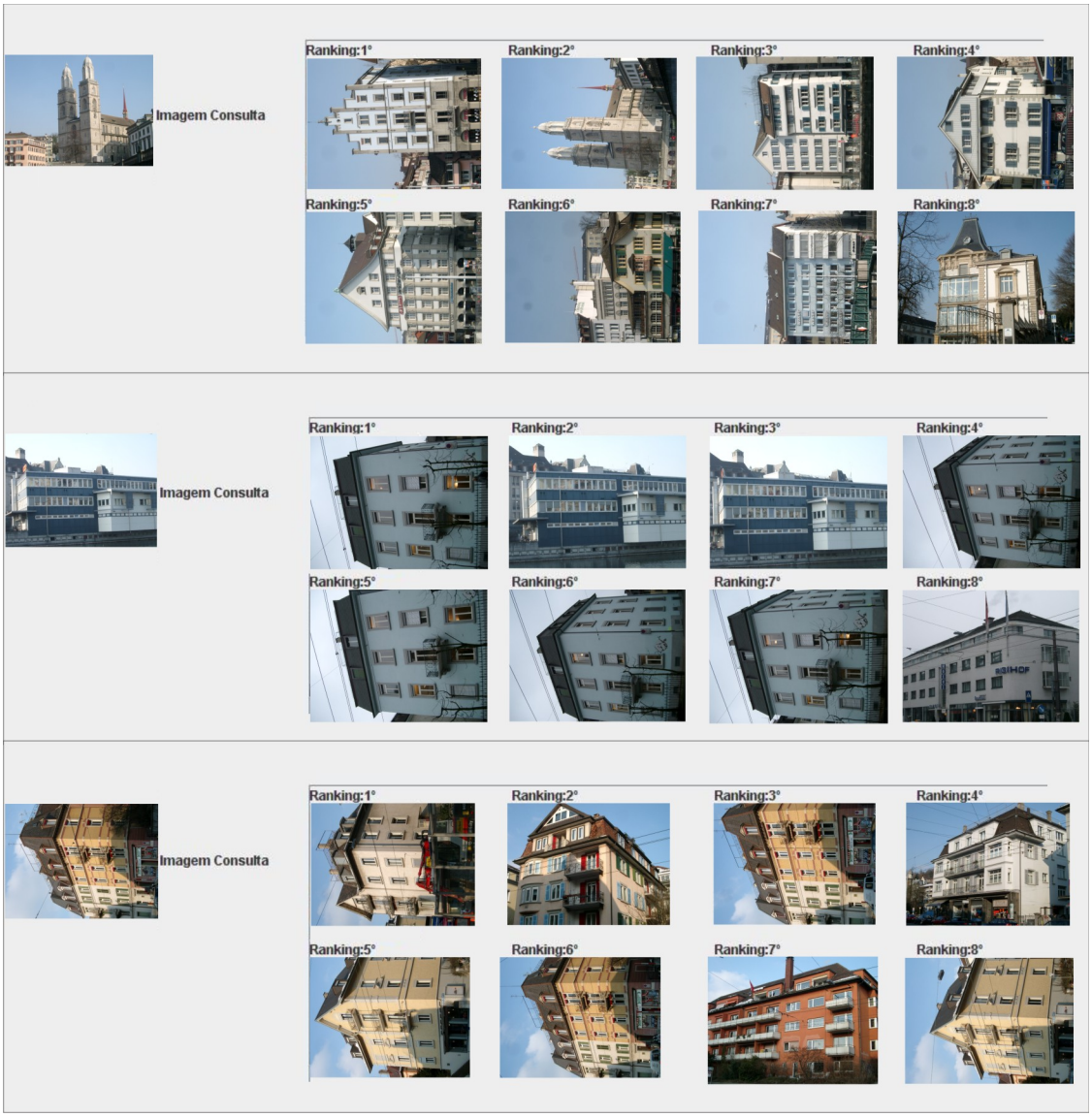


Figura 7.14: Resultados de busca obtidos pelo modelo proposto na base de imagens *ZuBuD* que interferem negativamente no cálculo da taxa de erro de classificação.

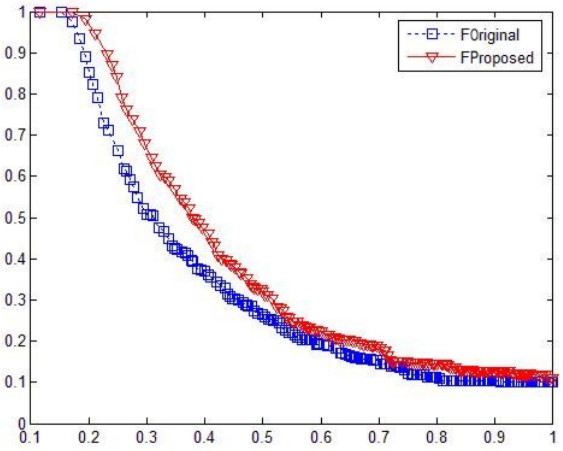


Figura 7.15: Curvas de precisão-revocação obtidas pelos resultados das características de baixo nível (*FOriginal*) e dos vetores de características neurosemânticas (*FProposed*) no banco de dados *Vistex-167*.

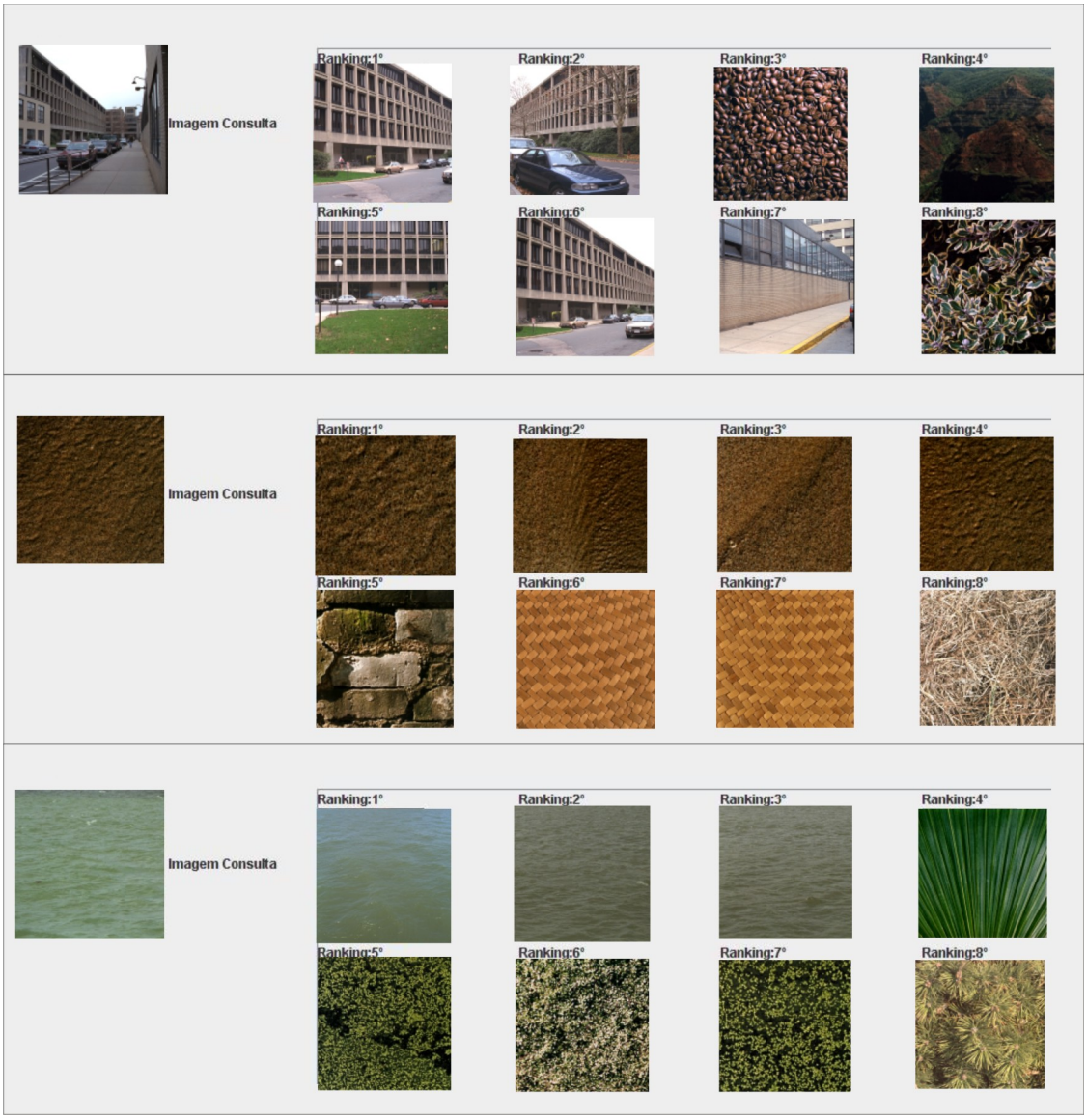


Figura 7.16: Resultados de busca obtidos pelo modelo proposto na base de imagens *Vistex-167*.

Capítulo 8

Aplicação: Caracterização Neurosemântica de Padrões de Movimentos Complexos utilizando Redes Neurais

8.1 Introdução

Imagine a cena de um parque em um domingo ensolarado, onde pessoas caminham pela grama, jogam futebol e conversam entre si enquanto caminham. Essa cena pode ser descrita de diversas maneiras: através de palavras, como feito anteriormente ou através de pinturas e fotografias digitais que, muitas vezes, podem não descrever alguns aspectos da cena como o movimentos que as pessoas realizam. Fazer essas descrições de forma automática através de um computador pode-se tornar uma tarefa laboriosa e muitas vezes ineficiente. Uma das maneiras de se tratar esse assunto é através do processamento digital de vídeos. Um vídeo digital é a representação digital de um sinal analógico de vídeo. Essa amostragem pode ser feita quadro a quadro, onde cada quadro ou *frame* é uma imagem digital dada por uma função $f(x, y)$ em um dado instante t. Logo um vídeo digital pode ser representado por um conjunto de *frames* dispostos em forma de um cubo dado pela função $f(x, y, t)$. A Figura 8.1 apresenta um exemplo de *frames* que formam a sequência de uma ação onde um cilindro de vidro é lançado na água.

Um dos problemas explorados em processamento de vídeos atualmente é a tarefa de identificação de diferentes ações dos seres humanos em uma sequência de vídeos. O reconhecimento de ações humanas é um componente chave em muitas aplicações de visão computacional, entre elas pode-se destacar: detectar atividades relevantes em vigilância baseada em vídeos, interface homem computador, reconhecimento de gestos, análise de eventos esportivos como também coreografias de dança, indexação e recuperação de vídeos [Blank et al. 2005]. No entanto, ainda é um problema desafiador para os pesquisadores desenvolver algoritmos para alcançar um nível satisfatório de classificação de vídeos onde há oclusão, fundos confusos, movimentos de câmera, mudança de foco, e variações geométricas dos objetos [Niebles et al.].

Os métodos de reconhecimento de ações podem ser divididos em duas categorias: métodos baseados em modelos e métodos baseados em características. As aproximações baseadas em modelo recorrem tanto para a adaptação de uma estrutura pré-definida - tradicionalmente

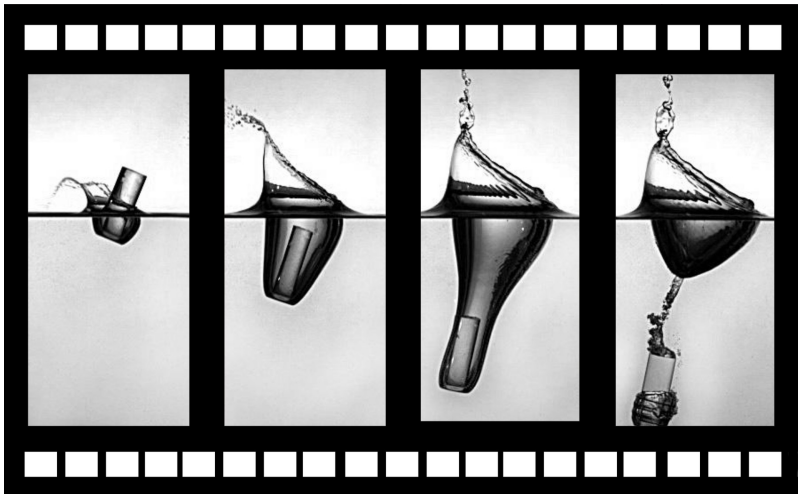


Figura 8.1: Exemplos de *frames* de um vídeo.

um esqueleto humano - para definir um movimento, quanto para o casamento com modelos de movimentos pré-definidos. Esses métodos funcionam bem, mas são limitados pelo fato de que modelos antropométricos[1] explícitos são requeridos [Goodhart et al. 2008]. As aproximações baseadas em características são inerentemente mais gerais pois examinam os dados brutos dos pixels de cada frame, no entanto são muito sensíveis a ruídos e transformações [Goodhart et al. 2008]. Outro problema no reconhecimento de ações é a variação intra-classes que produzem uma grande confusão para os classificadores como por exemplo duas pessoas que correm com estilos diferentes, em lugares diferentes ou vistas de diferentes pontos de vista. A variação inter-classes também é um problema pois alguns movimentos similares como andar e correr, podem confundir os classificadores.

Vários algoritmos são usados para descrever esses dados e suprimir esses problemas. Neste capítulo vamos explorar um desses algoritmos para a extração de características de vídeo e aplicaremos o método proposto por esse trabalho para transformá-las em características de neurosemânticas na tentativa de reduzir o erro de classificação de movimentos humanos complexos. A seguir esse descritor também chamado de descritor de características espaço-temporais bem como a formação do vetor de características que servirá como suporte da rede neural na aplicação do modelo proposto serão apresentados. Após isso, é discutida a adaptação do modelo proposto apresentado no Capítulo 6 para a caracterização de vídeos seguida de uma avaliação experimental. Nessa adaptação as redes neurais serão treinadas com vídeos-exemplo para dar à rede o caráter semântico necessário para a representação de movimentos complexos. Após a rede ser treinada, os pesos serão armazenados e essa rede (sem as funções de ativação) será utilizada para caracterizar neurosemânticamente os vídeos, para que os vetores resultantes dessa caracterização sejam utilizados na classicação ou recuperação de padrões de movimentos complexos. Cada umas dessas fases será explicitada nas seções seguintes.

[1] Antropometria (do Grego ánthropos: homem + métron) é o conjunto de técnicas utilizadas para medir o corpo humano ou suas partes.

8.2 Extração de Características Espaço-Temporais

Características espaço temporais (também chamados cubóides) são pequenas partes (pequenos sub-vídeos) extraídas localmente dos vídeos em pontos de interesse. Os pontos de interesse são pontos encontrados onde há a configuração de um movimento, como por exemplo um olho abrindo, uma flexão do joelho, ou a pata de um rato se movendo rapidamente para frente e para trás [Dollar et al. 2005].

A Figura 8.2 apresenta a ilustração da retirada de cubóides de um vídeo em pontos de interesse para o reconhecimento do comportamentos de ratos. Esses comportamentos são descritos em termos dos tipos e das localizações dos cubóides. No reconhecimento de ações humanas, esses cubóides são utilizados para descrever os movimentos realizados durante uma ação (como por exemplo: andar, correr, acenar, etc.).

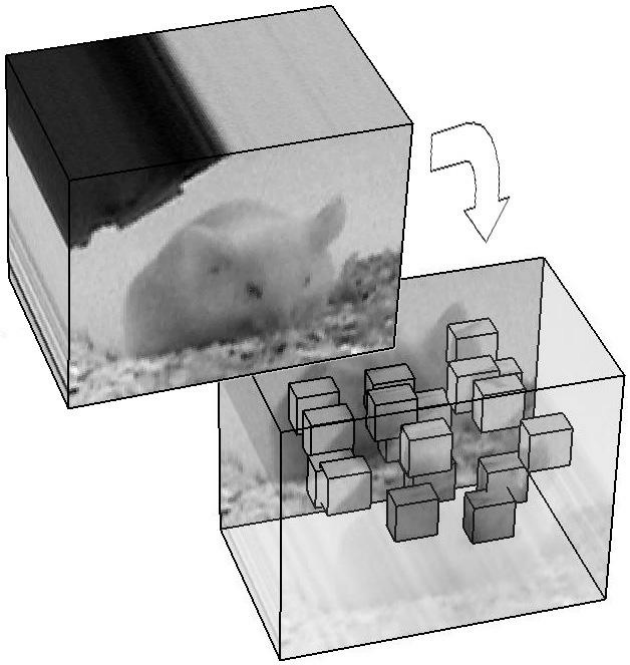


Figura 8.2: Visualização dos cubóides extraídos para o reconhecimento de comportamentos de ratos(Retirado de [Dollar et al. 2005]).

A motivação para o uso dos cubóides é que movimentos similares podem produzir cubóides similares. Por exemplo, um olho abrindo pode ser caracterizado de forma arbitrária independente da aparência global (quando analisamos o vídeo inteiro não importa onde o movimento ocorre), postura (não importa como ocorre a ação, se ela ocorre o cubóide será detectado), oclusão (não importa se parte do vídeo está ocluso, os outros cubóides detectados podem descrever a mesma ação) e assim por diante. Isso pode ser explicado pelo fato de que os cubóides estarão localizados apenas nos pontos de interesse, independente do fundo do vídeo ou se partes do mesmo estão oclusos por um objeto.

A Figura 8.3 mostra um exemplo de comparação entre dois vídeos de ratos executando uma ação (se limpando). Considere a figura como sendo dividida em duas colunas. A primeira coluna

apresenta no topo um frame do vídeo explicitando onde cada um dos seis cubóides extraídos está localizado. Em cada coluna, cada linha apresenta os seis cubóides representados por oito pequenos *frames*. Pode-se observar comparando os três primeiros cubóides (três primeiras linhas) de cada vídeo que, apesar dos dois ratos estarem em posições diferentes esses três cubóides mostrados nas três primeiras linhas parecem semelhantes.

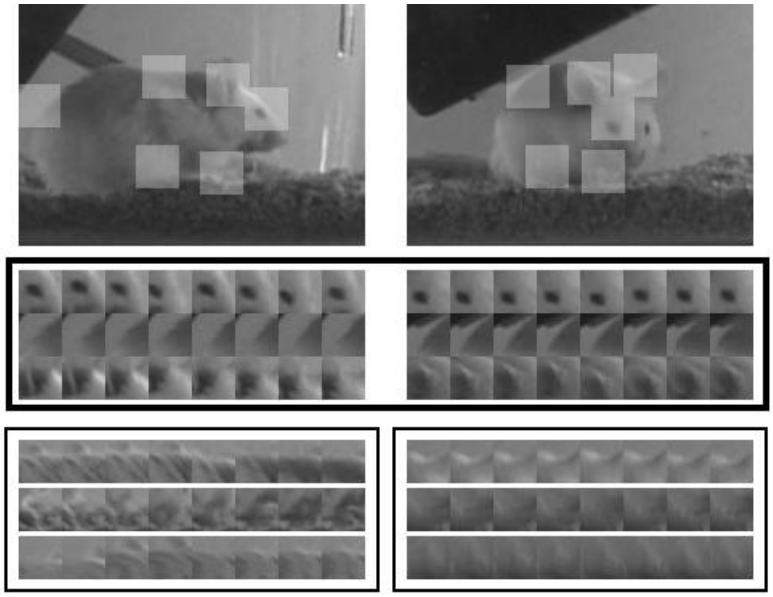


Figura 8.3: Exemplo de seis cuboides extraídos para cada um das duas sequências de ratos se limpando. Cada imagem representa a sequência original. O conjunto de cubóides extraídos é mostrado abaixo de cada imagem e cada cubóide é representado por oito pequenos frames no respectivo tempo (retirado de [Dollar et al. 2005]).

Vários métodos podem ser utilizados para se extrair os pontos de interesse no domínio espacial e uma avaliação e comparação destes métodos podem ser encontrados em [Schmid et al. 2000]. Neste trabalho, para a aplicação do método proposto utilizamos a aproximação utilizada por [Dollar et al. 2005] para encontrar esses pontos de referência onde uma função resposta é calculada em cada ponto da imagem e os pontos de interesse serão correspondentes aos pontos de máximo local. Essa função resposta pode ser calculada do seguinte modo: primeiramente é aplicado um detector de bordas nos frames e detectados nos mesmos as saliências das bordas. Essas saliências podem ser calculadas por vários algoritmos como em [Forstner and Gulch]. Para cada conjunto de frames em sequência essas saliências são identificadas e uma função resposta é calculada para identificar a mudança de posição entre uma saliência de um frame com as mesmas saliências dos frames posteriores. As maiores respostas retornaram o conjunto de saliências que serão selecionadas para formar os cubóides.

Como foi dito anteriormente, cada ponto de interesse servirá de base para a extração do cubóide de tamanho predefinido, que conterá os valores dos *pixels* chamados de *pixels* espaço-temporais. O tamanho do cubóide é atribuído de maneira a conter o maior volume de dados que contribuirão para a aplicação. Para efeito de ilustração, os vídeos utilizados para avaliar

a aplicação do modelo proposto na caracterização neurosemântica são vídeos de dimensão = com uma taxa de 50 frames por segundo. Desta forma optamos por escolher para os cubóides extraídos janelas de tamanho 20×20 com uma profundidade de 7 *frames*.

Após a extração deste cubóides, a formação do vetor de características pode ser feita de várias formas. A mais simples dela é "achatar" o cubóide colocando cada valor dos frames que formam o cubóide em sequência em um vetor. Por exemplo, seja um cubóide formado por conjunto de frames $\{F_1, F_2, ..., F_o\}$, onde o é o número de frames que compõe cada cubóide. O k-ésimo frame F_k é constituído por uma matriz $n \times m$ de *pixels*:

$$F_k = \begin{pmatrix} a_{11}^k & \cdots & a_{1n}^k \\ a_{21}^k & \cdots & a_{2n}^k \\ \vdots & \ddots & \vdots \\ a_{m1}^k & \cdots & a_{mn}^k \end{pmatrix}$$

onde $k = 1, 2...o$. Para montar o vetor de características pode-se transformar o conjunto de matrizes F_k em vetores e concatená-los na ordem em que eles estão dispostos no vídeo. O vetor resultante poderá ser escrito da seguinte forma:

$$(a_{11}^1, ..., a_{1n}^1, a_{21}^1, ..., a_{2n}^1, ...a_{m1}^1, ...a_{mn}^1, ...,$$
$$a_{11}^k, ..., a_{1n}^k, a_{21}^k, ..., a_{2n}^k, ...a_{m1}^k, ...a_{mn}^k, ...,$$
$$a_{11}^o, ..., a_{1n}^o, a_{21}^o, ..., a_{2n}^o, ...a_{m1}^o, ...a_{mn}^o).$$

Existem outras maneiras de se extrair características desses cubóide, pode-se por exemplo fazer um histograma dos valores dos pixels deste cubóide. Porém, tais métodos são muito sensíveis a pequenas transformações e o principal objetivo desta caracterização é criar uma forma de deixar o cubóide invariante a essas transformações, como por exemplo, translação, pequenas mudanças de iluminação, etc. Para tentar obter o equlíbrio certo entre a invariância e o poder discriminativo dos vetores de características extraídos dos cubóides, Dollar *et. al* [Dollar et al. 2005] projetou alguns extratores de características desses cubóides. Dois desses extratores foram utilizados nesse trabalho: o gradiente do brilho e *optical flow*. Os algoritmos para a implementação destes métodos podem ser encontrados em [Dollar et al. 2005].

Para este trabalho optamos por extrair 25 cubóides para cada vídeo e para cada cubóide foram extraídos dois vetores de características (gradiente e *optical flow*). Esses dois vetores referentes à um cubóide foram concatenados formando 25 vetores de características para cada vídeo. No entanto, para a aplicação do modelo proposto precisamos representar cada vídeo em um único vetor e, desse modo, surge uma questão: O que fazer com os 25 vetores de cada vídeo para colocá-los em um único vetor? Vale ressaltar que cada vetor está associado à um cubóide e que esse cubóide pode estar localizado em qualquer lugar do vídeo (como pode-se observar na Figura 8.2). Vários métodos podem ser utilizados para ordenar esses vetores [Dollar et al. 2005]. A seguir é apresentada a técnica utilizada para a obtenção dos vetores de características para este trabalho. Essa técnica monta uma espécie de dicionário de cubóides. Nesse caso o dicionário de cubóides é uma compilação de cubóides quantizados em uma certa ordem.

Seja um conjunto de n vídeos (tomando como exemplo a Figura 8.4 suponha um conjunto

de 3 vídeos). Em cada vídeo, são localizados t pontos de referência formando $t \times n$ cubóides (no caso do nosso exemplo 9 cubóides, 3 de cada vídeo). A localização desses pontos de referência é realizada por algoritmos específicos encontrados em [Dollar et al. 2005].

Após a extração dos cubóides é então realizada a extração dos vetores de características desses cubóides (conforme explicitado anteriormente). Os vetores resultantes serão, então, clusterizados em k grupos (no caso do exemplo da Figura 8.4 o número de grupos é 3). O número de grupos irá definir quantas posições o vetor resultante irá possuir, pois cada posição desse vetor irá se referir ao número de cubóides que está inserido naquele grupo para cada vídeo.

Após separar os cubóides em grupos, retorna-se então à cada vídeo para identificar qual cubóide estava em qual grupo, quantizando em um novo vetor de características os cubóides daquele vídeo que estava naquele grupo específico. Como resultado dessa quantização teremos uma espécie de histograma de cubóides onde cada posição identifica o grupo em questão e os *bins* representam o número de cubóides em cada grupo. Após feito isso com todos os vídeos, os mesmos estarão caracterizados em um vetor k-dimensional e prontos para serem utilizados. Para efeito de padronização esses vetores de características espaço-temporais provenientes do dicionário de cubóides serão chamados a partir de agora de vetores de características de baixo nível.

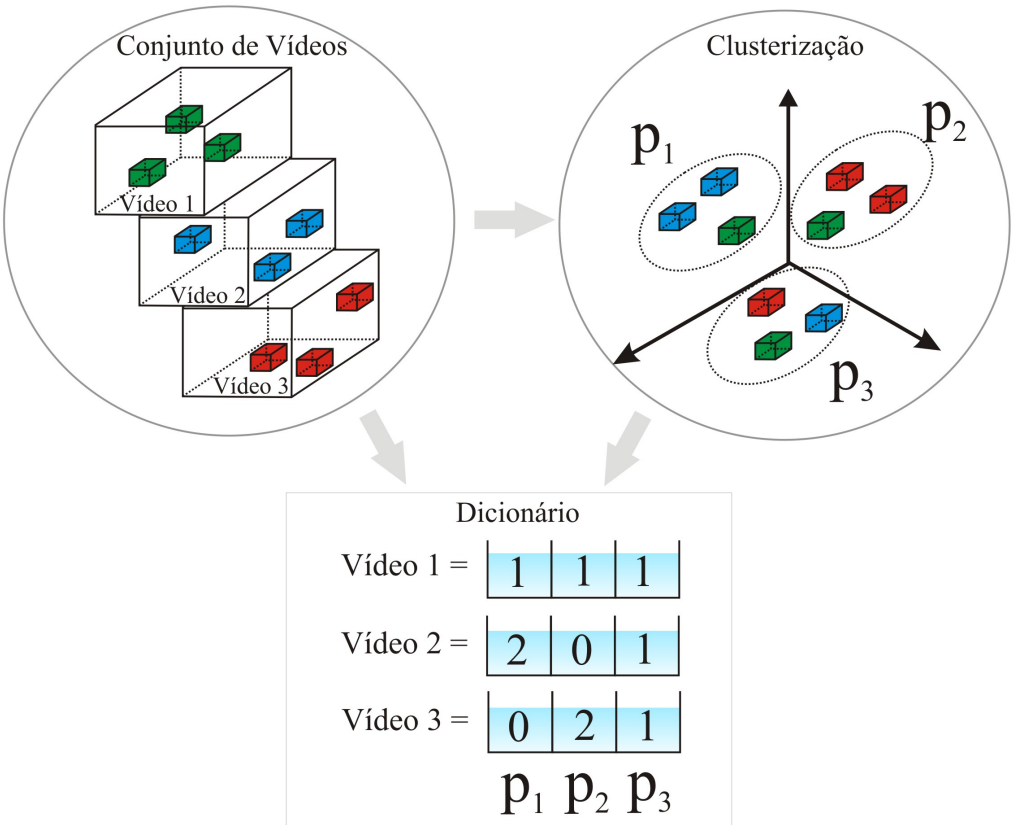


Figura 8.4: Ilustração da formação do conjunto de características de um conjunto de vídeos.

8.3 Aplicação do Modelo Proposto para a Caracterização Neurosemântica de Imagens na Caracterização de Vídeos

O principal objetivo dessa aplicação é tentar incorporar informações semânticas nos vetores de características de baixo nível para serem utilizados no reconhecimento de movimentos. Aqui, nós utilizamos a intuição de que os pesos da rede neural treinada para classificar padrões de movimentos possam ser utilizados para realizar um mapeamento não linear de um espaço de características de baixo nível para um espaço de características neurosemanticamente ponderado. Portanto, assim como na caracterização de imagens, a rede neural irá funcionar como uma função de transformação de características.

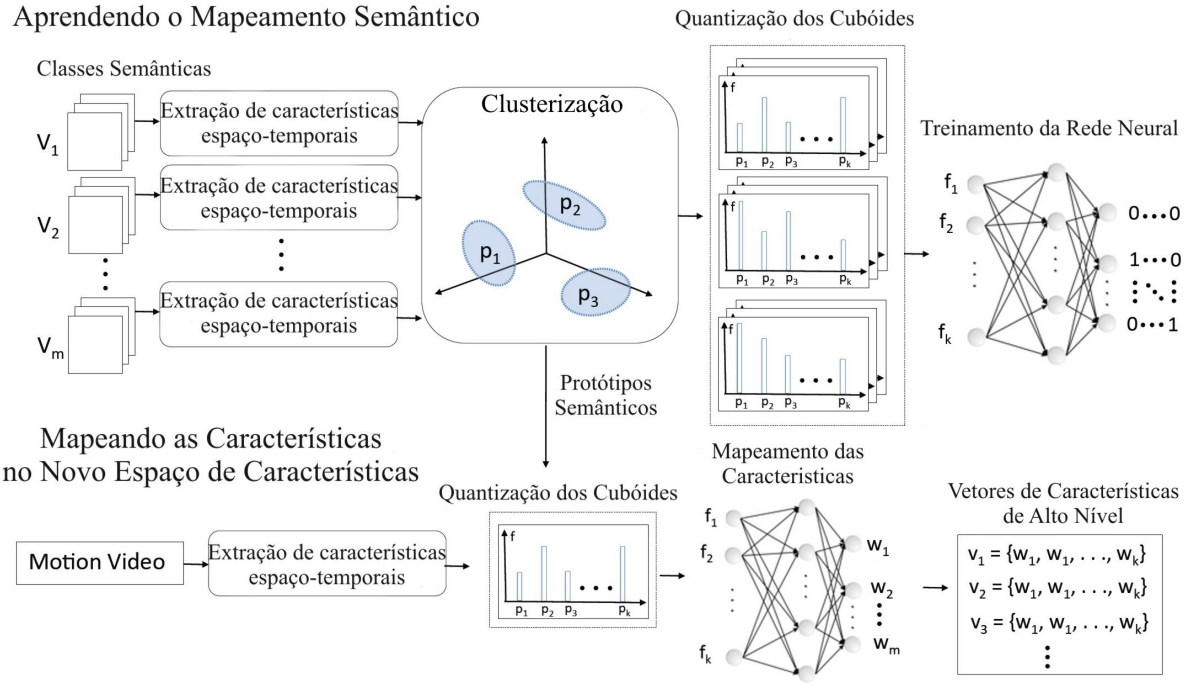


Figura 8.5: Os dois principais processos na proposta de adaptação. Na fase de treinamento (em cima), uma rede neural é treinada utilizando as sequências de vídeos pré-classificadas. No processo de caracterização neurosemântica (em baixo) são utilizados os protótipos e os pesos da rede neural para produzir um mapeamento das características de baixo nível extraídas.

Uma visão geral da adaptação do modelo proposto pode ser vista na figura 8.5 que consiste em duas fases. Primeiramente um conjunto de vídeos pré-classificados $\{V_1, V_2, ...V_n\}$ representando diferentes categorias semânticas é selecionado para a fase de treinamento da rede ou seja, para o aprendizado do mapeamento semântico das categorias. A idéia aqui é aprender uma representação implícita das categorias de movimentos usando a rede neural. Essa representação implícita será então usada para mapear as características de baixo nível dos vídeos contendo movimentos complexos em um espaço de características neurosemânticas durante a segunda fase do modelo. Em seguida os vetores de características neurosemânticas resultantes dessa

aplicação serão utilizados para a classificação de movimentos humanos em vídeos. A seguir os passos para a obtenção do método apresentado na figura 8.5 são descritos.

Após selecionados os vídeos da fase de treinamento e clusterizados em grupos $\{p_1, p_2, ..., p_k\}$ por um classificador, as características de baixo nível referentes a cada vídeo são extraídas de acordo com os métodos apresentados na seção anterior formando o vetor de características $(f_1, f_2, ..., f_k)$. Os vetores resultantes dessa caracterização em baixo nível serão utilizados para o treinamento da rede neural implementada e treinada da mesma forma explicitada na seção 6.2 dessa dissertação.

Após a rede neural estar treinada, os pesos serão salvos, as funções de ativação retiradas e um outro grupo distinto de vídeos de testes será escolhido. Este conjunto será caracterizado da mesma forma utilizada nos vídeos de treinamento. Após essa caracterização, os vetores de características passarão pela rede neural treinada que irá transformar esses vetores de características de baixo nível em novos vetores de característica neurosemânticas, que poderão ser utilizados tanto para classificação quanto para recuperação.

8.4 Avaliação Experimental

Assim como na avaliação da caracterização de imagens, um protótipo de sistema foi implementado para estudar a avaliação de performance das redes neurais na caracterização de vídeos. Esta seção descreve os testes realizados de acordo com o modelo proposto e apresenta alguns resultados experimentais para ilustrar a eficiência do método considerado.

Para os experimentos foi escolhido um banco de 90 vídeos de baixa resolução (180×144, com 50 fps) com sequências de atividades humanas. Esses vídeos de atividades humanas foram extraídos de [Blank et al. 2005] que se tornou um banco de dados de teste padrão para métodos de reconhecimento de ações similares. Existem nesse banco vídeos com 9 indivíduos, cada um efetuando 10 ações naturais descritas em inglês como: *"run"* (correr), *"walk"* (andar), *"skip"* (saltar de um pé só), *"jumping-jack"* ou *"jack"* (fazer polichinelo), *"jump-forward-on-two-legs"* ou *"jump"* (saltar com os dois pés juntos), *"jump-in-place-on-two-legs"* ou *"pjump"* (saltar parado com os dois pés juntos), *"gallopsideways"* ou *"side"* (correr de lado), *"wave-two-hands"* ou *"wave2"* (acenar com duas mãos), *"wave-one- hand"* ou *"wave1"* (acenar com uma mão), e *"bend"* (curvar-se). Tais vídeos foram filmados com câmera estática e em um fundo simples. Alguns *frames* exemplo para cada movimento são mostrados na figura 8.6 (a) e algumas sequências de frames de alguns movimentos são mostradas em 8.6 (b).

Utilizamos como entrada da rede os vetores provenientes dos descritores dos cuboides descritos na seção 8.2, e após realizados vários testes, escolhemos os descritores gradiente e *optical flow*. Para a formação do dicionário de cubóides utilizamos a técnica de agrupamento k-médias (ou *k-means*). Este algoritmo pode ser encontrado em vários artigos da literatura. O número de grupos utilizados foi escolhido de forma empírica e resultou em k=800 tanto para a característica *Optical Flow* como também para a característica gradiente, após concatenados esses vetores formaram um vetor único de 1600 posições. As posições iniciais dos centros dos clusteres foram inicializadas aleatoriamente.

Portanto, a rede neural implementada para esta aplicação possui 1601 neurônios na camada de entrada (bias e características de baixo nível), 80 neurônios na camada oculta e 10 neurônios

na camada de saída (características neurosemânticas), i.e. a rede neural implementada possui uma estrutura do tipo 1600-80-10. Para o treinamento da rede a função de ativação utilizada foi a função sigmóidal simples e a taxa de aprendizado utilizada foi escolhida empiricamente e possui o valor 0.003.

Para treinar e testar a rede optamos por utilizar a estratégia "leave one out", visto que foi constatado que o número de vídeos (90) é insuficiente para ser dividido em base de treinamento e de testes. Esta estratégia é adotada por vários sistemas, inclusive de reconhecimento de ações humanas. A estratégia "leave one out" consiste na seleção de um conjunto de vídeos da base de dados para o treinamento retirando um elemento para testar o sistema final. No caso desta aplicação, são retiradas todos os vídeos de movimentos de um mesmo indivíduo enquanto todos os vídeos restantes são utilizadas para treinar a rede. Como foi dito anteriormente, a base de dados possui 90 vídeos, 9 indivíduos realizando 10 ações diferentes. Portanto em cada etapa, 80 vídeos são utilizados para treinar a rede e os 10 restantes são caracterizados neurosemanticamente. Esse passo é repetido até que todos os vídeos estejam caracterizados.

Para comparar os resultados optamos por utilizar a estratégia utilizada na literatura. Utilizamos o resultado da rede neural (vetores de características neurosemânticas) em um classificador simples, chamado algoritmo dos k vizinhos mais próximos (k-*nearest neighbor*) onde as tuplas de treinamento são os vetores "ideais" de saída da rede neural. O resultado desse classificador foi sumarizado em uma tabela de acurácias sobre cada movimento também chamada de matriz de confusão. A matriz de confusão oferece uma medida efetiva do modelo de classificação, ao mostrar o número de classificações corretas versus as classificações preditas para cada classe, sobre um conjunto de exemplos. O número de acertos, para cada classe, se localiza na diagonal principal da matriz e representa a acurácia para aquela classe, os demais elementos representam erros na classificação. A soma dos valores da diagonal principal dividida pelo número de classes representa a acurácia do classificador. A acurácia de uma estimativa é uma medida da correlação entre o valor resultante da aplicação do classificador e os valores das fontes de informação (resultados que deveriam ser apresentados), ou seja, mede o quanto a estimativa que obtivemos é relacionada com o "valor ideal" do classificador. Ela nos informa o quanto o valor estimado é "bom", ou seja, quanto o valor estimado é "próximo" do valor real e nos dá a "confiabilidade" daquela estimativa ou valor. Portanto, a acurácia de um classificador ideal deve ser igual à 100% e a matriz de confusão desse classificador ideal deve possuir todos os valores da diagonal principal iguais a 1 e os outros elementos iguais a 0 uma vez que ele não comete erros. Medir adequadamente o desempenho do método proposto através da matriz de confusão assume um papel importante neste trabalho, uma vez que o objetivo consiste em construir um método com alta taxa de acerto em novos exemplos.

A Figura 8.7 (a) apresenta os resultados obtidos pelo modelo proposto por este trabalho. A Figura 8.7 (b),(c) e (d) ilustra os resultados apresentados por Goodhart *et al.* [Goodhart et al. 2008], Scovanner *et al.* [Scovanner et al. 2007] e Niebles and Fei-Fei [Niebles and Fei-Fei 2007], respectivamente. Pode-se notar que os resultados obtidos pelo modelo proposto são promissores. A maioria das classes obteve uma melhor performance em nosso método. Em particular, os movimentos "bend", "jack", "pjump" e "walk" foram classificados com uma acurácia de 100%. Pode-se notar que o método proposto foi bom tanto quando o individuo fazia movimentos em uma posição ou quando ele se movia de um lado para o outro da cena.

Além disso, pode ser observado que foi obtida uma performance razoável para a maioria das

ações com exceção dos movimentos "jump", "run" e "skip". Essas duas últimas ações são muito similares entre si, pois o indivíduo atravessa o cenário de uma forma bem similar. O movimento "skip" é o movimento mais difícil de classificar, tanto que em alguns trabalhos (como na figura 8.7 (d)) ele nem é inserido na tabela de resultados. O método obteve respostas confusas onde as ações são muito similares. De um modo geral, o sistema proposto classificou corretamente 85.7% das sequências de teste, além de reduzir em 99,37% a dimensionalidade dos dados (de 1600 características para 10 características) o que significa uma grande redução no tempo de processamento.

A tabela 8.1 apresenta o resultado da acurácia do sistema proposto em comparação com as acurácias obtidas pelos resultados de Goodhart *et al.* [Goodhart et al. 2008] , Scovanner *et al.* [Scovanner et al. 2007] e Niebles and Fei-Fei [Niebles and Fei-Fei 2007]. Os resultados demonstram que o método proposto tem uma melhor performance que os outros apresentados na literatura.

Métodos	**Acurácia (%)**
Modelo Proposto	**85,7**
Goodhart *et al.* [Goodhart et al. 2008]	84,6
Scovanner *et al.* [Scovanner et al. 2007]	82.6
Niebles and Fei-Fei [Niebles and Fei-Fei 2007]	72,8

Tabela 8.1: Comparação dos diferentes métodos utilizando o a base de dados Weizmann Human Action.

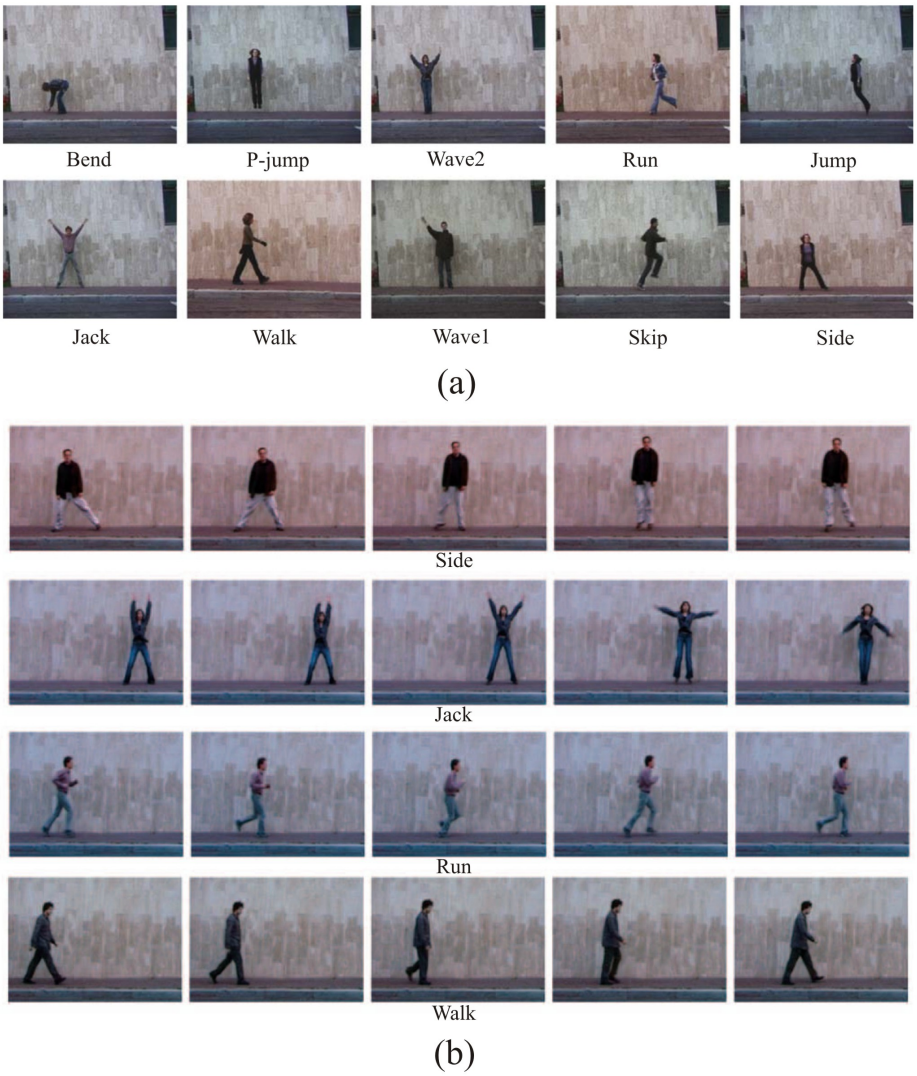


Figura 8.6: (a) Exemplos de frames para cada um dos movimentos do banco de vídeos *Weizmann Human Action* [Blank et al. 2005]. (b) Exemplos de sequências de frames de alguns movimentos.

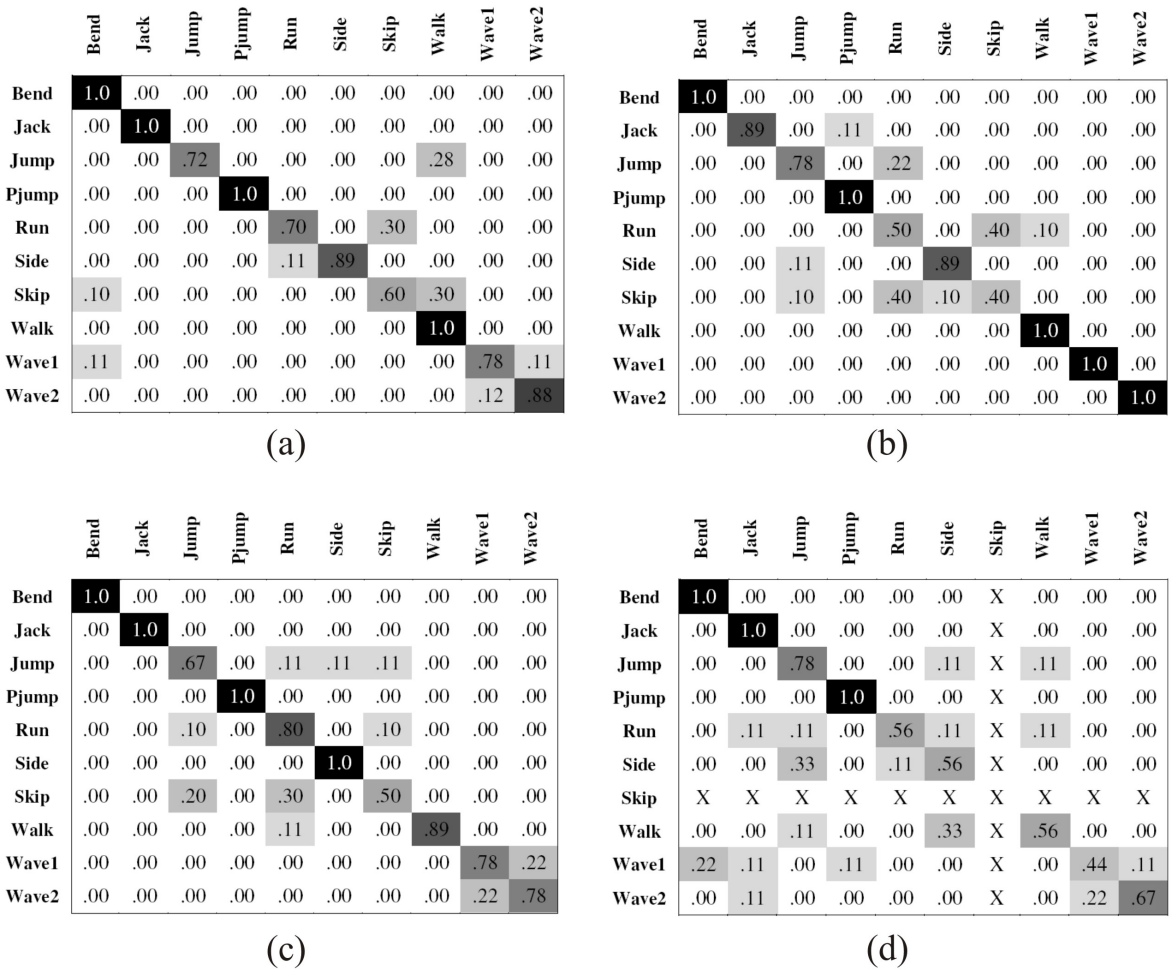


Figura 8.7: Matrizes de confusão obtidas pelo método proposto (a) e pelos métodos observados em Goodhart *et al.* [Goodhart et al. 2008] (b), Scovanner *et al.* [Scovanner et al. 2007] (c) e Niebles and Fei-Fei [Niebles and Fei-Fei 2007] (d).

Capítulo 9

Conclusão e Trabalhos Futuros

Apesar dos avanços nos últimos anos a pesquisa em recuperação e caracterização de imagens ainda possui grandes desafios a serem alcançados. Um deles é a extração de um número suficiente de características para representar uma imagem que reduza a dimensionalidade dos dados originais e possa realmente diferenciar imagens que seja perceptualmente diferentes. Além disso essas características devem "idealmente" descrever as imagens baseando-se na percepção humana e na sua subjetividade, onde devem ser considerados os aspectos mais importantes para que a recuperação satisfaça o objetivo da consulta. Baseando-se nessas necessidades, diversas pesquisas estão sendo realizadas buscando-se extratores de características, técnicas de seleção e transformação de características e formas de análise de similaridade entre imagens. A complexidade do problema faz com que os pesquisadores busquem informações de áreas, como a psicologia e a neurociência, para tentar entender um pouco do funcionamento do cérebro e tentar reproduzir essas asserções no sistemas de recuperação e caracterização de imagens. Assim, a escolha das redes neurais para a caracterização neurosemântica de imagens se torna um aspecto importante pelo fato de que modelos de redes neurais lidam muito bem com dados imprecisos e situações não totalmente definidas.

Neste trabalho foram estudadas técnicas de extração de características para descrever e representar as imagens e proposto o uso das redes neurais para caracterizar imagens neurosemânticamente. Em vista disso uma visão geral sobre as redes neurais e a necessidade do aprendizado para que um sistema inteligente possa ser considerado como tal foi dada. Enfocou-se o aprendizado neural como sendo uma forma robusta de aquisição de conhecimentos, que tem demonstrado um grande potencial em Processamento de Imagens e Visão Computacional.

A partir disto foi proposto o modelo de um sistema de caracterização e recuperação de imagens para mapear características de baixo nível gerando um novo espaço de características na tentativa de suprimir a subjetividade humana e reduzir assim a discrepância entre o poder limitado de descrição das características de baixo nível e a descrição de alto nível feita por interpretadores humanos.

A técnica proposta foi testada em várias bases de dados, sendo uma delas com mais de 10.000 imagens cobrindo uma grande faixa de categorias semânticas. Para melhor avaliar a proposta, comparações entre os resultados obtidos pela caracterização das imagens em baixo nível e o conjunto resultante da caracterização via redes neurais (neurosemântica) foram realizados e

mostram que o método proposto pode refinar os resultados. Comparações entre os resultados obtidos pelo método proposto e os resultados obtidos por outros descritores explanados em [Deselaers et al. 2008] foram feitas, e demonstram a potencialidade do modelo proposto. Em geral, o método proposto apresentou bons resultados e acreditamos que o mesmo possa ser uma solução robusta para a aquisição de conhecimento tornando a transformação e redução de características mais eficiente.

Como trabalhos futuros pretende-se testar e avaliar o modelo proposto em um banco de dados genérico, como o banco de dados *BD-12750*, mas que seja classificado para apurar o quanto a rede conseguiu generalizar os conceitos semânticos durante a fase de treinamento. A partir disso pretende-se estudar o impacto da base de treinamento de acordo com o número de classes e quais classes devem ser representadas para que o sistema abranja ainda mais características neurosemânticas. Pretende-se também fazer a verificação da dependência do modelo com outros tipos de caracterizações em baixo nível para melhorar ainda mais acurácia do método. Uma outra abordagem a ser estudada seria a utilização de mapas auto-organizáveis (com o auxílio de algorítmos de agrupamentos não supervisionados) para que a rede seja constantemente retreinada com novas imagens, aumentando o número de classes e melhorando as já existentes.

Pretende-se também fazer testes com o uso da similaridade por regiões explicadas no capítulo de Recuperação de Imagens, tais técnicas podem melhorar ainda mais os resultados pela caracterização de baixo nível das imagens.

A realimentação por relevantes poderá também ser incorporada futuramente ao sistema para que o usuário transforme suas necessidades em uma consulta devidamente formulada diminuindo ainda mais o gap-semântico.

Neste trabalho o método proposto foi extendido na aplicação da caracterização neurosemântica de vídeos de movimentos complexos. Nessa proposta a rede neural foi treinada para representar movimentos complexos representados por características-espaço temporais e utilizada para caracterizar vídeos de uma base de dados muito utilizada na literatura. Demonstramos através de experimentos utilizando técnicas de avaliação presentes na literatura que o método proposto apresenta resultados bem satisfatórios. Fizemos ainda, a comparação dos nossos resultados com resultados de publicações recentes de reconhecimento de movimentos e concluímos que o método proposto na caracterização de vídeos é bem auspiciosa.

Apesar dos experimentos serem limitados à uma base com categorias predefinidas (andar, correr, saltar, etc.), acreditamos que o modelo proposto pode ser aplicado de uma forma mais geral principalmente na recuperação de vídeos com várias categorias de movimentos. Devido à complexidade do problema de reconhecimento de movimentos a partir de sequências de vídeo perante a atual tecnologia, é de se esperar que ainda haja muito trabalho a ser feito. Como trabalhos futuros, pretendemos montar um banco de vídeos mais geral e aplicar o método proposto para, dado um vídeo consulta, recuperar os vídeos mais similares baseados nos vetores de características neurosemânticos obtidos e fazer estudos para aplicar o modelo proposto à outros extratores de características de vídeos.

Referências Bibliográficas

[Aigrain et al. 1996] Aigrain, P., Zhang, H., and Petkovic, D. (1996). Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review. *Multimedia Tools and Applications*, 3(3):179–202.

[Aksoy and Haralick 1998] Aksoy, S. and Haralick, R. (1998). Textural features for image database retrieval. *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries in conjunction with CVPR'98*, pages 45–49.

[Anderson and Rosenfeld 1988] Anderson, J. A. and Rosenfeld, E., editors (1988). *Neurocomputing: foundations of research*. MIT Press, Cambridge, MA, USA.

[Androutsos et al. 1998] Androutsos, D., Plataniotis, K. N., and Venetsanopoulos, A. N. (1998). Distance Measures for Color Image Retrieval. In *ICIP (2)*, pages 770–774.

[Ashutosh Saxena and Mukerjee 2004] Ashutosh Saxena, A. G. and Mukerjee, A. (2004). Nonlinear dimensionality reduction by locally linear isomaps. *Lecture notes in computer science*, 3316:1038–1043.

[Attneave 1954] Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review*, 3(61):83–193.

[Baeza-Yates and Ribeiro-Neto 1999] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

[Baldi and Hornik 1989] Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: learning from examples without local minima. *Neural Netw.*, 2(1):53–58.

[Bin and Jia-Xiong 2002] Bin, Y. and Jia-Xiong, P. (2002). Invariance analysis of improved Zernike moments. *Journal os optics A: Pure ans Applied Optics*, 4:606–614.

[Blank et al. 2005] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. volume 2, pages 1395–1402.

[Blum and Langley 1997] Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artif. Intell.*, 97(1-2):245–271.

[Brandt 1999] Brandt, S. (1999). Use of Shape Features in Content-Based Image Retrieval. Master's thesis, Helsinky University of Technology, Espoo, Finlândia.

[Bueno 2001] Bueno, J. M. (2001). Suporte à Recuperação de Imagens Médicas Baseada em Conteúdo através de Histogramas Métricos. Master's thesis, Universidade de São Paulo, USP - São Carlos.

[Cascia et al. 1998] Cascia, M. L., Sethi, S., and Sclaroff, S. (1998). Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web. Technical Report 1998-004.

[Castleman 1995] Castleman, K. (1995). *Digital Image Processing*. Upper Saddler River: Prentice Hall, New Jersey, 1st edition edition.

[Colombo et al. 1999] Colombo, C., Bimbo, A. D., and Pala, P. (1999). Semantics in Visual Information Retrieval. *IEEE MultiMedia*, 6(3):38–53.

[Comon 1994] Comon, P. (1994). Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314.

[Corel Database] Corel Database. Corel Corporation, Corel Gallery 3.0. Available in James Z. Wang's Research Group: `http://wang.ist.psu.edu/ jwang/test1.tar`.

[Cunningham 2007] Cunningham, P. (2007). Dimension Reduction. Technical report, University College Dublin.

[de Holanda Ferreira 2004] de Holanda Ferreira, A. B. (2004). *Novo Dicionário Aurélio da Língua Portuguesa*. Number 1. São Paulo / SP, 1 edition.

[de Melo Aires 1999] de Melo Aires, M. (1999). *Fisiologia*. Rio de Janeiro, RJ, 2 edition.

[Deselaers et al. 2008] Deselaers, T., Keysers, D., and Ney, H. (2008). Features for image retrieval: an experimental comparison. *Inf. Retr.*, 11(2):77–107.

[Dijkstra 1959] Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs. In *Numeriskche Mathematik*, volume 1, pages 269–271.

[Dollar et al. 2005] Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. pages 65–72.

[Dy and Brodley 2004] Dy, J. G. and Brodley, C. E. (2004). Feature Selection for Unsupervised Learning. *J. Mach. Learn. Res.*, 5:845–889.

[Edward 2002] Edward, B. L. (2002). DPF — A Perceptual Distance Function for Image Retrieval.

[Elliott and Cashman 1973] Elliott, R. W. and Cashman, L. E. (1973). An experimental comparison of relevance-feedback techniques. In *ACM'73: Proceedings of the annual conference*, pages 256–261, New York, NY, USA. ACM Press.

[Errity and McKenna 2007] Errity, A. and McKenna, J. (2007). A Comparative Study of Linear and Nonlinear Dimensionality Reduction for Speaker Identification. pages 587–590.

[Fausett 1994] Fausett, L. V. (1994). *Fundamentals of Neural Networks*. Prentice Hall.

[Fischler and Firschein 1987] Fischler, M. A. and Firschein, O. (1987). *Intelligence: the eye, the brain, and the computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Flickner et al. 1995] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by Image and Video Content: The QBIC System. *Computer*, 28(9):23–32.

[Fodor 2002] Fodor, I. (2002). A Survey of Dimension Reduction Techniques. Technical Report UCRL-ID-148494, LLNL.

[Foley et al. 1990] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990). *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Forstner and Gulch] Forstner, W. and Gulch, E. A fast operator for detection and precise location of distinct points, corners and centres of circular features.

[Fu 1974] Fu, K. S. (1974). *Syntactic Methods in Pattern Recognition*. Academic Press, N.Y.

[Gesù and Starovoitov 1999] Gesù, V. D. and Starovoitov, V. (1999). Distance-based functions for image comparison. *Pattern Recogn. Lett.*, 20(2):207–214.

[Golinval 2004] Golinval, G. E. J.-C. (2004). Feature extraction using auto-associative neural networks. *Smart materials and structures*, 13(1):211–219.

[Gonzales and Woods 1992] Gonzales, R. C. and Woods, R. E. (1992). *Digital Image Processing; Trad. Roberto Marcondes Cesar Júnior and Luciano Fontoura Costa*. Addison Wesley, University of Tennessee Perceptics Corporation, 2 edition.

[Gonzalez et al. 2006] Gonzalez, A. C., Sossa, J. H., Felipe, E. M., and Pogrebnyak, O. (2006). Wavelet transforms and neural networks applied to image retrieval. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 909–912, Washington, DC, USA. IEEE Computer Society.

[Goodhart et al. 2008] Goodhart, T., Yan, P., and Shah, M. (2008). Action recognition using spatio-temporal regularity based features. pages 745–748.

[Gupta and Jain 1997] Gupta, A. and Jain, R. (1997). Visual Information Retrieval. *Communications of the ACM*, 40(5):70–79.

[Hammerton 1998] Hammerton, J. A. (1998). Holistic Computation: Reconstructing a muddled concept. *Connection Science*, 10(1):3–19.

[Harada et al. 1997] Harada, S., Itoh, Y., and Nakatani, H. (1997). Interactive image retrieval by natural language. *Optical Engineering*, 36:3281–3287.

[Hartline 1957] Hartline, H., R. (1957). *Inhibitory Interactions of Receptor Units in the Eye of Limulus*, volume v. 40, pages 351–376.

[Haykin 2001] Haykin, S. (2001). *Redes Neurais: Princípio e Prática*. 2 edition.

[Hertz and Krogh 1991] Hertz, J. and Krogh, A. (1991). *Introduction to the Theory of Neural Computation*, volume 1. Perseus Books, Santa Fe Institute, Cambridge Massachusetts.

[Hu 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants, computer methods in image analysis. *IRE Transactions on Information Theory*, 8.

[Huang et al. 1997] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image Indexing Using Color Correlograms. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 762, Washington, DC, USA. IEEE Computer Society.

[Jain et al. 2000] Jain, A., Duin, R., and Mao, J. (2000). Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37.

[Jorgensen 1998] Jorgensen, C. (1998). Attributes of images in describing tasks. *Inf. Process. Manage.*, 34(2-3):161–174.

[Keysers and Gollan 2007] Keysers, D. and Gollan, C. (2007). Deformation Models for Image Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1422–1435. Student Member-Deselaers, Thomas and Member-Ney, Hermann.

[Laaksonen et al. 2000] Laaksonen, J., Oja, E., Koskela, M., and Br, S. (2000). Analyzing low-level visual features using content-based image retrieval. In *International Conference on Neural Information Processing (ICONIP*, pages 1333–1338.

[Lee et al. 2002] Lee, J. A., Lendasse, A., and Verleysen, M. (2002). Curvilinear distance analysis versus isomap. In *Proceedings of ESANN 2002, 10th European Symposium on Artificial Neural Networks*, pages 185–192.

[Lin 1991] Lin, B.-C. (1991). A new computation of geometric moments. *Pattern Recognition*, 1(26):109–113.

[Lin and Shen 1991] Lin, B.-C. and Shen, J. (1991). Fast computation of moment invariants. *Pattern Recognition*, 8(24):807–813.

[Liu and Picard 1996] Liu, F. and Picard, R. W. (1996). Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. on Pattern Analysis and Machine Learning*, 18(7):184–189.

[Liu and Motoda 1998] Liu, H. and Motoda, H. (1998). Feature Transformation and Subset Selection. *IEEE Intelligent Systems*, 13(2):26–28.

[Liu and Yu 2005] Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502.

[Liu et al. 2007] Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282.

[Loew 2000] Loew, M. H. (2000). Feature Extraction - SPIE Handbook on Medical Imaging: Medical Image Processing and Analysis. 2.

[Long et al. 2003] Long, F., Zhang, H., and Feng, D. (2003). Fundamentals of Content-Based Image Retrieval. *Multimedia information retrieval and management – technological fundamentals and applications, Springer-Verlag*, pages 1–26.

[Lowe 1987] Lowe, D. G. (1987). Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31(3):355–395.

[Luger 2004] Luger, G. F. (2004). *Inteligência Artificial - Estruturas e Estratégias para a Resolução de Problemas Complexos; Trad. Paulo Engel.* University of New Mexico at Albuquerque, 4 edition.

[Ma and Manjunath 1995] Ma, W. and Manjunath, B. (1995). Image indexing using a texture dictionary.

[Ma and Wang 2005] Ma, X. and Wang, D. (2005). Semantics modeling based image retrieval system using neural networks. In *ICIP (1)*, pages 1165–1168.

[Mallat 1992] Mallat, S. G. (1992). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:674–693.

[Moghaddam et al. 1998] Moghaddam, B., Wahid, W., and Pentland, A. (1998). Beyond Eigenfaces: Probabilistic Matching for Face Recognition. In *Proc. of Int'l Conf. on Automatic Face and Gesture Recognition (FG'98)*, pages 30–35, Nara, Japan.

[Niebles and Fei-Fei 2007] Niebles, J. and Fei-Fei, L. (2007). A Hierarchical Model of Shape and Appearance for Human Action Classification. pages 1–8.

[Niebles et al.] Niebles, J., Wang, H., and Fei-Fei, L. Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *International Journal of Computer Vision.*

[Papoulis 1991] Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*, volume 1. McGraw-Hill Companies, 3 edition.

[Paradiso 2002] Paradiso, B. W. C. . M. F. B. . M. A. (2002). *Neurociências: Desvendando o Sistema Nervoso.* 2004 edition.

[Pass and Zabih 1996] Pass, G. and Zabih, R. (1996). Histogram Refinement for Content-Based Image Retrieval. In *IEEE Workshop on Applications of Computer Vision*, pages 96–102.

[Pedrosa et al. 2008] Pedrosa, G. V., Santos, C. F., Batista, M. A., Fernandes, H. C., and Barcelos, C. A. (2008). An Effective Salience-Based Algorithm for Shape Matching. In *ACIVS '08: Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 818–827, Berlin, Heidelberg. Springer-Verlag.

[Pentland et al. 1996] Pentland, A., Picard, R., and Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254.

[Picard and Minka 1995] Picard, R. W. and Minka, T. P. (1995). Vision Texture for Annotation. *Multimedia Systems*, 3(1):3–14.

[Plan et al. 1998] Plan, N., NASA, and NPD, D. (1998). National Aeronautics and Space Administration.

[Raymer et al. 2000] Raymer, M., Punch, W., Goodman, E., Kuhn, L., and Jain, A. (2000). Dimensionality reduction using genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 4(2):164–171.

[Roweis and Saul 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326.

[Rowley et al. 1998] Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

[Rubner 1999] Rubner, Y. J. (1999). *Perceptual metrics for image database navigation.* PhD thesis. Adviser-Carlo Tomasi.

[Rui et al. 1999] Rui, Y., Huang, T., and Chang, S. (1999). Image retrieval: current techniques, promising directions and open issues.

[Rui et al. 1997] Rui, Y., Huang, T. S., and Chang, S.-F. (1997). Image retrieval: Past, present, and future. In *International Symposium on Multimedia Information Processing.*

[Rumelhart and McClelland 1986] Rumelhart, D. and McClelland (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2. The MIT Press, Cambridge, MA.

[Russel and Norvig 2004] Russel, S. and Norvig, P. (2004). *Inteligência Artificial; trad. Vandenberg D. de Souza.* Elsevier, 2 edition.

[Russo 1991] Russo, A. (1991). Neural Networks for Sonar Signal Processing. In nº 8, T., editor, *IEEE Conference on Neural Networks for Ocean Egineering*, Washington, DC.

[Salton and McGill 1986] Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

[Santini and Jain 1996] Santini, S. and Jain, R. (1996). Similarity Matching. *Lecture Notes in Computer Science*, 1035:571.

[Schmid et al. 2000] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172.

[Scovanner et al. 2007] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 357–360, New York, NY, USA. ACM.

[Sethi et al. 2001] Sethi, I. K., Coman, I. L., and Stan, D. (2001). Mining association rules between low-level image features and high-level concepts. *Proceedings of the SPIE Data Mining and Knowledge Discovery*, 3:279–290.

[Shi et al. 2004] Shi, R., Feng, H., Chua, T.-S., and Lee, C.-H. (2004). An adaptive image content representation and segmentation approach to automatic image annotation. *International Conference on Image and Video Retrieval (CIVR)*, pages 545–554.

[Siggelkow 2002] Siggelkow, S. (2002). *Feature Histograms for Content-Based Image Retrieval.* PhD thesis, Albert-Ludwigs-Universität Freiburg, Fakultät für Angewandte Wissenschaften, Germany.

[Silva et al. 2006] Silva, S. R., Barcelos, C. A. Z., and Batista, M. A. (2006). The Effects of Fitness Functions on Genetic Algorithms applied to Relevance Feedback in Image Retrieval. *Semantic Multimodal of Digital Media*, Cost 292:21–23.

[Smith 1997] Smith, J. R. (1997). *Integrated Spatial and Feature Image Systems: Retrieval , Analysis and Compression.* PhD thesis, Columbia University, New York.

[Sonka et al. 2007] Sonka, M., Hlavac, V., and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision.* Thomson-Engineering.

[Squire and Pun 1997] Squire, D. and Pun, T. (1997). A comparison of human and machine assessments of image similarity for the organization of image databases.

[Srihari 1995] Srihari, R. K. (1995). Automatic Indexing and Content-Based Retrieval of Captioned Images. *Computer, IEEE Computer Society Press, Los Alamitos, CA, USA*, 28(9):49–56.

[Stricker and Orengo 1995] Stricker, M. A. and Orengo, M. (1995). Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392.

[Swain and Ballard 1991] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *Int. J. Comput. Vision*, 7(1):11–32.

[Tamura et al. 1978] Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-8(6):460–473.

[Tong and Chang 2001] Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA. ACM Press.

[Town and Sinclair 2000] Town, C. and Sinclair, D. (2000). Content based image retrieval using semantic visual categories.

[Tuceryan and Jain 1993] Tuceryan, M. and Jain, A. K. (1993). Texture analysis. pages 235–276.

[Vistex Database] Vistex Database. Vision Texture Database. Massachusetts Institute of Technology. Media Laboratory. Available in `ftp://whitechapel.media.mit.edu/pub/VisTex/`.

[Woods 1964] Woods, W. A. (1964). Important Issues in Knowledge Representation. In *Proceedings of the IEEE*, volume 74, pages 1322–1334.

[Zhang 2002] Zhang, D. (2002). Image Retrieval Based on Shape. Phd thesis, Faculty of Information Technology, Monash University.

[Zhang and Lu 2004] Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19.

[Zhang and Izquierdo 2007] Zhang, Q. and Izquierdo, E. (2007). Combining Low-Level Features for Semantic Inference in Image Retrieval. *Eurassip - Journal on Advances in Signal Processing*, April.

[ZuBuD Database] ZuBuD Database. Zurich Buildings Database for Image Based Recognition. Media Laboratory. Swiss Federal Institute of Technology. Available in `http://www.vision.ee.ethz.ch/ZuBuD`.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo