Uso de Técnicas de Aprendizagem para Classificação e Recuperação de Imagens

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Fábio Augusto Faria e aprovada pela Banca Examinadora.

Campinas, 11 de junho de 2010.

Ricardo Torres

Ricardo da Silva Torres (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Faria, Fábio Augusto
F225u Uso de técnicas de aprendizagem para classificação e recuperação de imagens/Fábio Augusto Faria -- Campinas, [S.P. : s.n.], 2010.

Orientador : Ricardo da Silva Torres
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1.Processamento de imagens. 2.Análise de imagem. 3.Sistemas de recuperação de informação. 4.Programação genética (Computação). 5.Aprendizagem. I. Torres, Ricardo da Silva. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Use of learning techniques for image classification and retrieval

Palavras-chave em inglês (Keywords): 1. Image processing. 2. Image analysis. 3. Information storage and retrieval systems. 4. Genetic programming (Computation). 5. Learning.

Área de concentração: Recuperação de Informação / Processamento de Imagens

Titulação: Mestre em Ciência da Computação

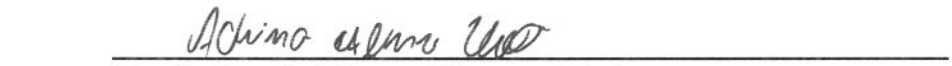
Banca examinadora: Prof. Dr. Ricardo da Silva Torres (IC-UNICAMP)
Prof. Dr. Adriano Veloso (UFMG)
Prof. Dr. Anderson de Rezende Rocha (IC-UNICAMP)

Data da defesa: 19/03/2010

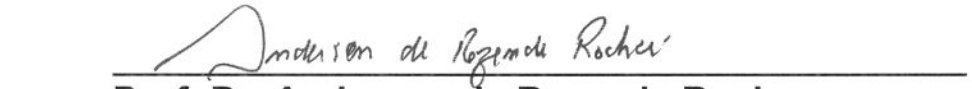
Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

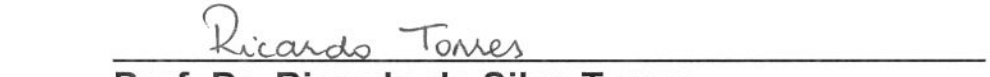
Dissertação Defendida e Aprovada em 19 de março de 2010, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Adriano Alonso Veloso
DCC / UFMG



Prof. Dr. Anderson de Rezende Rocha
IC / UNICAMP



Prof. Dr. Ricardo da Silva Torres
IC / UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Uso de Técnicas de Aprendizagem para Classificação e Recuperação de Imagens

Fábio Augusto Faria[1]

Junho de 2010

Banca Examinadora:

- Ricardo da Silva Torres (Orientador)
- Adriano Alonso Veloso
  Universidade Federal de Minas Gerais
- Anderson de Rezende Rocha
  Universidade Estadual de Campinas
- Alexandre Xavier Falcão (Suplente)
  Universidade Estadual de Campinas
- Marcos André Gonçalves (Suplente)
  Universidade Federal de Minas Gerais

[1]Suporte financeiro de: Bolsa do CNPq (processo 380965/2008-9), Bolsa FAPESP (processo 2008/57016-8).

Resumo

Técnicas de aprendizagem vêm sendo empregadas em diversas áreas de aplicação (medicina, biologia, segurança, entre outras). Neste trabalho, buscou-se avaliar o uso da técnica de Programação Genética (PG) em tarefas de recuperação e classificação de imagens. PG busca soluções ótimas inspirada pela teoria de seleção natural das espécies. Indivíduos mais aptos (melhores soluções) tendem a evoluir e se reproduzir nas gerações futuras.

As principais contribuições deste trabalho são: implementação de um classificador de imagens utilizando PG para combinar evidências visuais (descritores de imagens) e assim, obter melhores resultados com relação à eficácia de classificação; Comparação de PG com outras técnicas de aprendizagem em tarefas de recuperação de imagens por conteúdo; Uso de regras de associação para recuperação de imagens.

Abstract

Learning techniques have been used in several applications (medicine, biology, surveillance systems, e.g.) This work aims to evaluate the use of the Genetic Programming (GP) learning technique for image retrieval and classification tasks. This technique is a problem-solving system that follows principles of inheritance and evolution, inspired by the idea of Natural Selection. The space of all possible solutions is investigated using a set of optimization techniques that imitate the theory of evolution.

The main contributions of this work are: proposal of classifier implementation using GP to combine visual evidences (image descriptors) to be used in image classification tasks; comparison of GP with other learning techniques in content-based image retrieval tasks.

Agradecimentos

Eu gostaria de agradecer ao meu pai pelo apoio, pela confiança depositada em todos os momentos, por ser essa pessoa tão importante em minha vida e que sou grato todos os dias (um grande amigo); aos meus irmãos Vitor e Sérgio pela amizade verdadeira que temos; à minha namorada Ariadne pelo carinho, paciência e compreensão nos períodos difíceis; aos meus colegas, em especial ao meu amigo Tripodi, do Laboratório de Sistema de Informação (LIS) do Instituto de Computação da Unicamp pelo ambiente descontraído e pelas colaborações que tivemos durante o período e a todos os amigos por existirem, pois sem eles eu nada seria. Agradeço ao orientador Prof. Ricardo da Silva Torres pela paciência, compreensão e ensinamentos, à Prof. Claudia Medeiros pelo apoio, sabedoria e pela oportunidade de fazer parte de seu grupo de pesquisas, aos pesquisadores Eduardo Valle, Prof. Anderson de Rezende Rocha, Adriano Veloso e Humberto Mossri de Almeida pela grande ajuda nos experimentos realizados. Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Projeto BioCore) e à Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP pelo apoio financeiro.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Nos dias atuais, com o avanço das tecnologias de aquisição e armazenamento de imagens, juntamente com o uso da internet, verifica-se o surgimento de grandes coleções de imagens. Além disso, a redução de custos associado ao armazenamento e processamento de grandes coleções de imagens fizeram aumentar a utilização de sistemas computacionais inteligentes, os quais fazem uso de técnicas de aprendizagem.

Técnicas de aprendizagem são empregadas na área de processamento de imagem na análise e reconhecimento de padrões, classificação e recuperação de imagens [4]. Estas técnicas têm sido utilizadas em diversas aplicações. Na área médica, a tarefa de classificação, por exemplo, é empregada em vários tipos de imagens (raio-x, tomografias computadorizadas, entre outras) [5, 7, 13]. Em [5], técnicas de aprendizagem são utilizadas na detecção/classificação de tumor maligno e benigno. Já em [6], são usadas para determinar idade gestacional em recém-nascidos. Em [67], são utilizadas para auxiliar no diagnóstico automático de exames médicos. Na área de segurança, técnicas de aprendizagem são usadas para reconhecimento biométricos (digitais e íris) [10, 42, 80]. Na Biologia, verifica-se seu uso em sistemas para mapeamento de genomas, modelagem e simulação, desenvolvimento e exploração de bases de dados de imagens de seres vivos, visando a identificação e classificação de espécies (e.g., [70]), entre outras. Outros trabalhos também estão sendo feitos para classificar e/ou reconhecer imagens de frutos e imagens de sensoriamento remoto [25, 60].

As tarefas de classificação e recuperação são aplicadas a uma área chamada Mineração de Imagem. Esta área consiste na extração automática de padrões ou relacionamentos entre dados de imagens que podem resultar em informações úteis a usuários [78]. Esta é uma área interdisciplinar que envolve pesquisa em visão

computacional, processamento de imagem, mineração de dados, banco de dados e inteligência artificial. O desafio da pesquisa é utilizar representação de baixo nível (propriedades de *pixel*, como cor, textura e forma de objetos) contida em uma imagem para identificar objetos e relacionamentos em alto nível, ou seja, extrair conhecimento implícito contido em grandes bases de imagens [78].

Em geral, em tarefas de classificação supervisionada de imagem assume-se que cada imagem pertence a uma classe definida. O problema consiste em treinar um classificador de forma que este indique a classe correta para uma dada imagem de teste [78].

O processo de recuperação de imagens, por sua vez, consiste em buscar em coleções de imagens aquelas que sejam de interesse para o usuário. Uma das técnicas mais utilizadas recentemente é a recuperação de imagens por conteúdo (do inglês *Content-based Image Retrieval* - CBIR [16]). CBIR consiste em recuperar as imagens mais similares de uma coleção em relação a um padrão de consulta (uma imagem por exemplo) levando-se em conta propriedades visuais como cor, forma e textura.

Tanto no domínio de CBIR quanto no de classificação, descritores de imagens são usados para caracterizar propriedades visuais. Um descritor é caracterizado por duas funções: (1) um algoritmo de extração de característica que codifica as propriedades visuais (cor, textura e forma) em um vetor de características; e (2) uma medida de similaridade (função de distância) que calcula a similaridade entre duas imagens utilizando os vetores de características correspondentes. Diferentes descritores potencialmente fornecem diferentes informações que se complementam. Cada descritor pode caracterizar diferentemente propriedades visuais distintas (cor, textura e forma) ou até uma mesma propriedade visual (os descritores BIC [65] e JAC [73] caracterizam propriedade visual de cor de maneiras diferentes). Certamente um descritor pode ser mais eficaz para uma base de imagens que para outra, mas nenhum descritor é "perfeito" para qualquer coleção.

O uso de técnicas de aprendizagem para combinar diferentes evidências visuais caracterizados por descritores tem o objetivo de melhorar o desempenho em tarefas de recuperação e classificação de imagens, explorando as vantagens de bons descritores e minimizando eventuais efeitos negativos de descritores ruins.

Uma técnica que vem sendo estudada em diversas áreas como mineração de dados, processamento de sinais, evolução interativa, classificação e recuperação de imagens é a programação genética (PG) [8, 17, 21, 41, 77]. Esta técnica da área de inteligência artificial busca soluções ótimas se baseando na seleção natural das espécies. Ou seja,

os indivíduos mais aptos (melhores soluções) tendem a se reproduzir e evoluir em gerações futuras.

A escolha da programação genética é motivada pelos bons resultados obtidos em estudos realizados recentemente [3, 17, 19, 26, 76]. Por exemplo, no trabalho de Torres et al. [17], essa técnica é empregada na combinação dos valores de similaridade obtidos a partir de descritores de imagens, gerando uma função de similaridade mais eficaz. Zhang et al. [76] recentemente propuseram um mecanismo de classificação de texto utilizando técnicas de programação genética para obter melhores funções de similaridade. Em [3, 20], PG é usada para otimizar as funções de ordenação em sistemas de busca por documentos textuais.

As principais contribuições desse trabalho de mestrado são:

- Adaptação do modelo de classificação de documentos textuais, baseado em Programação Genética, proposto por Zhang et al. [77] visando à classificação de imagens;
- Modelagem da técnica de ordenação de documentos textuais, baseado em regras de associação, para imagens proposto por Veloso et al. [72]. Esta técnica foi utilizada como técnica de referência nos experimentos realizados;
- Comparação da PG com outras técnicas de aprendizagem em problemas de classificação e recuperação de imagens;
- Implementação parcial de um sistema de classificação de imagens usando programação genética.

O presente trabalho está contido em um projeto mais amplo, o *Bio-CORE* [53], proposto no âmbito da Universidade Estadual de Campinas (UNICAMP) e que conta com pesquisadores nas áreas de Ciência da Computação e Biologia. O objetivo geral é fornecer aos cientistas que trabalham com registros e reconhecimento de espécies, um sistema que apoia busca exploratória multimodal, em fontes de dados heterogêneas. Essas fontes incluem imagens, dados geográficos e metadados referentes à descrição de habitat e ecossistemas. Assim, pretende-se estabelecer maior abrangência na pesquisa, propondo e implementando um sistema de classificação de imagens de seres vivos (por exemplo, classificação de espécies de insetos utilizando imagens de asas de moscas).

Neste trabalho de mestrado algumas colaborações foram realizadas, gerando resultados na forma de publicações:

- Uma delas foi uma parceria com colegas da Universidade Federal de Minas Gerais (UFMG) que originou um artigo aceito na *11th ACM SIGMM International Conference on Multimedia Information Retrieval* (MIR 2010) intitulado *Learning to Rank for Content-Based Image Retrieval* [23];

- Um artigo, em parceria com colegas do Laboratório de Informática Visual (LIV) do Instituto de Computação da UNICAMP, foi publicado nos anais do *XXII Brazilian Symposium on Computer Graphics and Image Processing* (SIBGRAPI 2009) intitulado *A comparative study among pattern classifiers in interactive image segmentation* [64];

- Uma parceria com colegas do Laboratório de Sistemas de Informação (LIS) do Instituto de Computação da UNICAMP resultou em artigo aceito para *International Geoscience and Remote Sensing Symposium* (IGARSS 2010) intitulado *A Genetic Programming Approach for Coffee Crop Recognition* [59];

- Uma outra parceria com colega da Universidade Estadual Paulista (Unesp-Bauru) resultou em um artigo aceito para *17th International Conference on Systems, Signals and Image Processing* (IWSSIP'10) intitulado *Multimodal Pattern Recognition Through Particle Swarm Optimization* [22].

Esse texto está organizado da seguinte forma: o Capítulo 2 apresenta trabalhos e conceitos relacionados; o Capítulo 3 descreve o modelo de classificação usando programação genética e experimentos realizados; o Capítulo 4 descreve experimentos realizados visando à comparação de PG com outras técnicas de aprendizagem na recuperação de imagens por conteúdo; finalmente, o Capítulo 5 apresenta as conclusões deste trabalho e suas possíveis extensões.

Capítulo 2

Trabalhos e Conceitos Relacionados

Este capítulo apresenta os conceitos básicos relacionados, bem como trabalhos correlatos. A seção 2.1 apresenta conceitos de recuperação de imagens por conteúdo (CBIR). A seção 2.2 aborda as técnicas de aprendizagem utilizadas nos experimentos de classificação e recuperação de imagens. A seção 2.3 apresenta conceitos relacionados ao problema de classificação de imagens. Por fim, a seção 2.4 apresenta conceitos relacionados ao problema de recuperação de imagens.

2.1 Descrição do Conteúdo Visual de Imagens

As técnicas de aprendizagem são usadas para classificação e recuperação de objetos a partir de um conjunto de medidas, atributos ou características, que podem ser quantitativas (área, comprimento) ou qualitativas (profissão, tipo sanguíneo). Os dados multimídias (imagem, som e vídeo) são geralmente semi-estruturados ou não-estruturados. Como a maior parte dos algoritmos e técnicas de aprendizagem foram desenvolvidos para processar dados estruturados, uma solução para tratar dados não-estruturados é extrair atributos dos dados originais. No caso de imagens, descritores são usados para extração de atributos (características).

2.1.1 Recuperação de imagens por Conteúdo

As definições abaixo, apresentadas em [16], discutem os principais conceitos de CBIR.

Definição 1 *Uma* ***imagem*** $\hat{I}$ *é um par* $(D_I, \vec{I})$*, onde:*

- D_I *é um conjunto finito de* pixels *(pontos em* $\mathbb{Z}^2$*, tal que,* $D_I \subset \mathbb{Z}^2$*), e*
- $\vec{I} : D_I \rightarrow \mathsf{D}'$ *é uma função que atribui a cada* pixel p *em* D_I *um vetor* $\vec{I}(p)$ *de valores em algum espaço arbitrário* D' *(por exemplo,* $\mathsf{D}' = \mathbb{R}^3$ *quando uma cor é atribuída a um* pixel *no sistema RGB).*

Definição 2 *Um* ***descritor simples*** *(ou simplesmente,* ***descritor****)* D *é definido como um par* (ϵ_D, δ_D)*, onde:*

- $\epsilon_D : \hat{I} \rightarrow \mathbb{R}^n$ *é uma função que extrai um* vetor de características $\vec{v}_{\hat{I}}$ *de uma* imagem $\hat{I}$*.*
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *é uma* função de similaridade *(por exemplo, baseada em uma medida de distância) que calcula a similaridade entre duas imagens a partir da distância entre seus* vetores de características *correspondentes.*

Definição 3 *Um* ***vetor de características*** $\vec{v}_{\hat{I}}$ *de uma imagem* $\hat{I}$ *é um ponto no espaço* $\mathbb{R}^n$*:* $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$*, onde* n *é a dimensão do vetor. A Figura 2.1(a) ilustra o uso de um descritor simples* D *para calcular a similaridade entre duas imagens* $\hat{I}_A$ *e* $\hat{I}_B$*. Primeiro, o algoritmo de extração* ϵ_D *é usado para computar os vetores de características* $\vec{v}_{\hat{I}_A}$ *e* $\vec{v}_{\hat{I}_B}$ *associados com as imagens. Depois, a função de similaridade* δ_D *é utilizada para o valor da similaridade* d *entre as imagens.*

Definição 4 *Um* ***descritor composto*** $\hat{D}$ *é um par* $(\mathcal{D}, \delta_{\mathcal{D}})$ *(veja Figura 2.1 (b)), onde:*

- $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *é um conjunto de* k *descritores simples pré-definidos.*
- $\delta_{\mathcal{D}}$ *é um função de similaridade que combina os valores de similaridade obtidos de cada descritor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

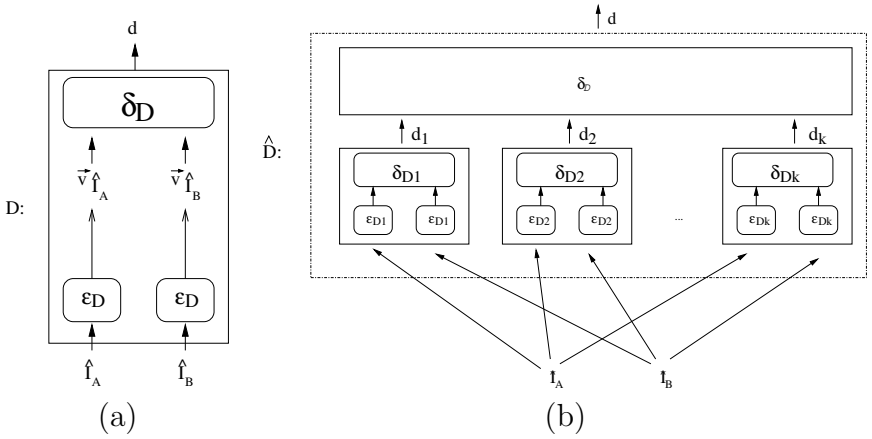


Figura 2.1: (a) O uso de um descritor simples D para computar a similaridade entre duas imagens e (b) um descritor composto. Retirado de [16].

2.2 Técnicas de Aprendizagem

Esta seção descreve as técnicas utilizadas neste trabalho para a tarefa de classificação e recuperação de imagem.

No domínio de CBIR, o uso de técnicas de aprendizagem para ordenar imagens busca aliviar o chamado problema de *gap* semântico: tradução de conceitos de alto nível em características de baixo nível, a partir do uso de descritores de imagens. Uma abordagem comum para tratar deste problema consiste no uso de métodos de aprendizagem para combinar diferentes descritores. Em [28, 62], estas abordagens consistem na atribuição de pesos para indicar a importância de um descritor. Basicamente, quanto maior o peso assumido, maior a importância do descritor. Frome et al. [28] aplicam uma formulação de margem maximal para aprender a combinação linear de distâncias elementares definas por triplas de imagens. Shao et al. [62] utilizam algoritmos genéticos para determinar os melhores pesos para avaliar descritores. Kernels e SVM têm sido usados para CBIR. Exemplos incluem [29, 79]. Torres et al. [17] exploram PG para combinar descritores e então encontrar o melhor peso para cada descritor. Estes algoritmos otimizam a recuperação de imagens e buscam diminuir o *gap* semântico.

Com o objetivo de incluir usuários no processo de CBIR, técnicas de Realimentação de Relevância têm sido propostas para melhorar a eficácia dos sistemas de recuperação. Em [26, 33, 48], técnicas de aprendizagem são usadas para caracterizar

as necessidades dos usuários. Nestas técnicas, o usuário indica para o sistema quais as imagens são mais relevantes e o sistema aprende com essas indicações buscando retornar as imagens mais similares na próxima iteração. Esses algoritmos de realimentação de relevância tentam caracterizar as necessidades ou percepções do usuário a partir de geração de funções de combinação de descritores.

Técnicas de *learning to rank* têm sido utilizadas na recuperaąão de documentos textuais, mas existem poucos trabalhos na literatura que o abordam para CBIR. Em [34] é proposto o uso de "multiplas instâncias" baseadas em um arcabouço de margem máxima (método adaptado do algoritmo RankSVM [32]), onde informação local é extraída das imagens. O trabalho descrito em [34] considera que o tratamento das imagens para recuperação é mais flexível se estas tiverem um grau de relevância (uma imagem é mais ou menos relevante que a outra) ao invés de usar o tradicional método de realimentação de relevância, no qual as imagens pertencem apenas a dois grupos (relevante e não relevante).

2.2.1 Programação Genética (PG)

Programação genética (PG) [41] é uma técnica da inteligência artificial para a solução de problemas baseados nos princípios da herança biológica e evolução. Nesse contexto, cada solução potencial é chamada de indivíduo em uma população. Sobre essa população são aplicadas transformações genéticas, como crossover e mutações, com o intuito de criar indivíduos mais aptos (melhores soluções) em gerações subsequentes. Uma função de adequação (fitness) é utilizada para atribuir valores para cada indivíduo com o intuito de definir o seu grau de evolução. A Tabela 2.1 mostra os principais componentes de um arcabouço de PG.

Em PG, são utilizadas estruturas de dados complexas, como árvores (a estrutura mais comum - ver Figura 2.2), listas encadeadas ou pilhas [43]. Além disso, o tamanho das estruturas de dados em PG não é fixo, embora seja possível restringir certos limites na implementação. Em virtude do paralelismo intrínseco no mecanismo de busca e poderosa capacidade de exploração global em espaços de dimensões mais elevadas, PG é utilizada para resolver uma ampla gama de problemas de otimização em que normalmente a melhor solução não é conhecida.

O Algoritmo 1 mostra os passos da evolução de indivíduos da PG. Dada uma população inicial de indivíduos, para cada uma das N gerações, calcula-se a aptidão de cada indivíduo, selecionam-se os indivíduos para sofrerem operações genéticas (*cros-*

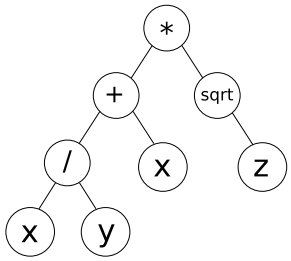


Figura 2.2: Representação de um indivíduo PG. Retirado de [60].

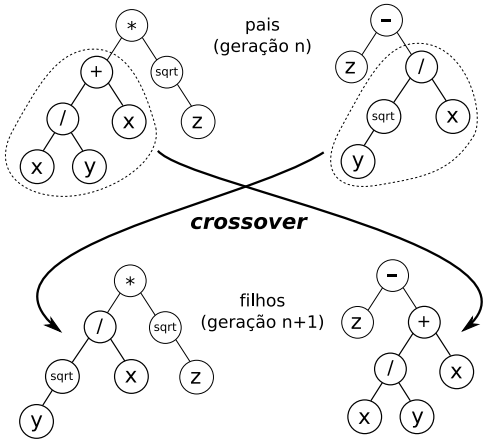


Figura 2.3: Operação de *crossover* entre indivíduos. Retirado de [60].

Componentes	Significado
Terminais	Nós folhas na estrutura da árvore.
Funções	Nós não-folhas utilizados para combinar os nodos folha. Operações numéricas comuns: +, -, *, /, log, *sqrt* .
Função de Adequação	A função que PG busca otimizar.
Reprodução	Um operador genético que copia os indivíduos com os melhores valores de adequação diretamente para a próxima geração, sem passar pela operação de *crossover*.
Crossover	Um operador genético que troca sub-árvores de dois pais para formar dois novos descendentes. A Figura 2.3 ilustra este operador.
Mutação	Um operador genético que troca uma sub-árvore de um determinado indivíduo, cuja raiz é um ponto de mutação escolhido, com uma sub-árvore gerada aleatóriamente. A Figura 2.4 ilustra este operador.

Tabela 2.1: Componentes essenciais de Programação Genética.

sover, mutação e reprodução) e ao final da N-ésima geração os melhores indivíduos são retornados.

Algoritmo 1 Algoritmo de evolução de indivíduos da Programação Genética.

```
1: Gere a população inicial de indivíduos
2:   Para N gerações faça
3:     Calcule a adequação de cada indivíduo
4:     Selecione os indivíduos para operações genéticas
5:       crossover, mutação e reprodução
6:   Fim Para
7: retorne os melhores indivíduos
```

2.2.2 *Support Vector Machine* (SVM)

Support Vector Machine é uma técnica de aprendizagem de máquina, introduzido por [9] e fundamentada em teorias estatísticas, em que são necessários exemplos previamente identificados para construir um modelo de classificação. O objetivo

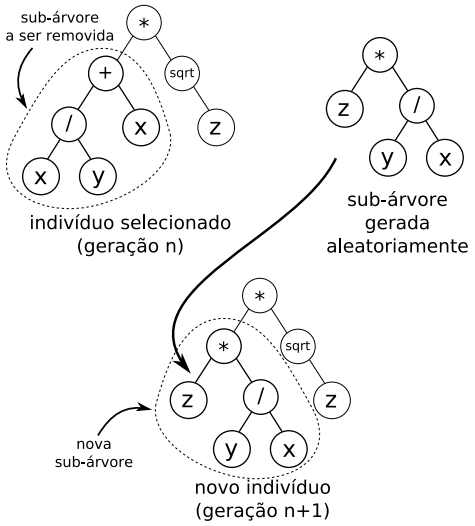


Figura 2.4: Operação de mutação entre indivíduos. Retirado de [60].

desta técnica é construir um hiperplano ótimo que separa o espaço n-dimensional (onde n é o número de característica da entidade que está sendo classificada, i.e., a dimensionalidade), tal que maximiza uma margem entre as duas classes. A margem pode ser interpretada como uma medida de separação entre duas classes e representa o grau de separabilidade entre elas (medida de qualidade da classificação). Os pontos sobre as fronteiras entre as classes são chamados de vetores de suporte (*support vectors*), e o meio da margem é o hiperplano de separação ótima. Quando não é possível achar um separador linear entre as classes, os dados (características) são mapeados em um espaço com alta dimensionalidade utilizando um mapeamento não linear. Segundo teorema de Cover [15], toda a amostra que não é separável em um espaço pode ser mapeada em um outro espaço de maior dimensionalidade através de transfomações não lineares e tornando-se linearmente separáveis. A Figura 2.5 ilustra o uso do SVM para "separar" duas classes.

Para maiores detalhes, sugere-se a consulta das referências [9, 11, 32, 33, 38]

2.2.3 Regras de Associação

Associar é uma tarefa descritiva que busca retornar itens de interesse em uma base de dados. Permite associar um conjunto de registros e suas ocorrências com outro

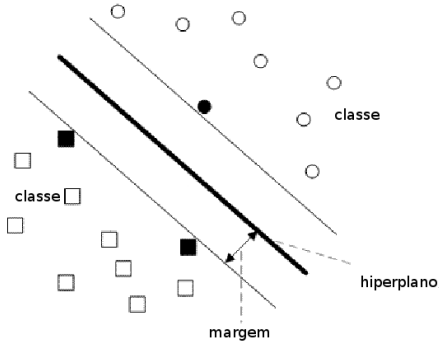


Figura 2.5: O classificador SVM encontra o hiperplano que separa as duas classes (quadrados e círculos) pela máxima margem possível.

conjunto. Uma regra é da forma $X \rightarrow Y$, onde $\mathcal{X}$ e $\mathcal{Y}$ são conjuntos de itens, ou seja, se $\mathcal{X}$ ocorre em uma transação da base de dados, $\mathcal{Y}$ tende a ocorrer também [1].

O uso de regras de associação é muito importante para diversas aplicações de mineração de dados como estratégia de negócios, processo de tomada de decisão, determinação de efeitos de alteração climáticas, entre outras. A análise de associação em uma base de dados pode gerar muitas regras de associação, algumas relevantes (mais frequentes), outras nem tanto. Desta forma, existem medidas de interesses (suporte e confiança mais usadas) que definem quais regras serão utilizadas.

As medidas de interesse são utilizadas para aumentar a utilidade, relevância ou mesmo criar uma ordenação dos padrões descobertos, resolvendo o problema de escolha de regras "ruins"ou não relevantes [57]. Os fatores mais usados ao definir medidas de interesse em regras de associação $\mathcal{X}$ implica $\mathcal{Y}$ ($\mathcal{X} \rightarrow \mathcal{Y}$) são:

- Cobertura: o número de tuplas que possui o antecedente da regra $\mathcal{X}$.
- Completude: a proporção de tuplas contendo $\mathcal{Y}$ que também contém $\mathcal{X}$.
- Confiança: proporção de tuplas contendo $\mathcal{X}$ que também contém $\mathcal{Y}$.

O problema de regras de associação foi introduzido em [1], como um problema no qual busca-se encontrar associações entre itens em bases de dados. A regra de associação é definida como, $I = \{i_1, ...i_n\}$ um conjunto de literais (itens), um conjunto $\mathcal{A}$ que pertence a I é chamado de *itemset*. Seja R uma relação com tuplas t que envolvem elementos que são subconjuntos de I. A tupla t suporta um itemset $\mathcal{A}$

se $\mathcal{A}$ está contido em t. O suporte de um *itemset* $\mathcal{A}$ é a razão entre o número de tuplas em R que suportam $\mathcal{A}$, e o número total de tuplas de R. Um *itemset* $\mathcal{A}$ é chamado *itemset* frequente se o suporte de $\mathcal{A}$ for maior ou igual ao suporte mínimo especificado pelo usuário. O suporte de regra é a razão entre o número de tuplas em R que contém $\mathcal{A}$ e $\mathcal{B}$, e o número total de tuplas de R. A confiança é a razão entre o número de tuplas que contém $\mathcal{A}$ e $\mathcal{B}$, e o número de tuplas que contém $\mathcal{A}$. As regras fortes são aquelas que satisfazem as restrições de ter suporte e confiança maiores que os seus mínimos especificados por susuários [57].

Algoritmos

A tarefa de encontrar regras de associação tradicional consiste em duas etapas: encontrar todos os *itemsets* frequentes e, gerar todas as regras fortes. A determinação dos *itemsets* frequentes é feita em duas etapas: gerar o conjunto de *itemsets* candidatos e a outra, percorrer a base de dados, determinando o suporte dos *itemsets* candidatos e encontrando o conjunto de *itemsets* frequentes [57].

O algoritmo mais conhecido nos dias atuais é o Apriori [1]. Este algoritmo percorre a base de dados inúmeras vezes, o que limita o seu desempenho. Como a fase que mais exige processamento é a determinação dos *itemsets* frequentes, novos algoritmos foram desenvolvidos para tentar resolver este problema. Dentre eles, incluem-se os algoritmos Partition [61], FP-Growth [31] e Eclat [72]. Para determinar as regras, uma vez encontrado os *itemsets* frequentes, basta gerar as combinações de cada *itemset* e calcular a confiança de cada combinação, descartando as que não satisfazem a confiança mínima estabelecida. O Algoritmo 2 apresenta os passos necessários para geração de regras de associação [2]:

Algoritmo 2 Algoritmo Básico para Identificação de Regras de Associação

1: Para cada *itemset* $\mathcal{A} \in I$ faça
2: Para todos os subconjuntos $\mathcal{B} \in \mathcal{A}$
3: Se $\mathcal{B} \rightarrow (\mathcal{A} - \mathcal{B})$ satisfaz a confiança mínima
4: Adicione $\mathcal{B} \rightarrow (\mathcal{A} - \mathcal{B})$ ao conjunto de regras C
5: Fim Se
6: Fim Para
7: Fim Para

2.3 Classificação de Imagens

Esta seção mostra o conceito de classificação de imagens e as técnicas utilizadas neste trabalho.

2.3.1 Visão Geral

A tarefa de classificação de imagens consiste em categorizar novas imagens em classes previamente definidas. O fator determinante para que imagens pertençam a uma mesma classe é que tenham as mesmas propriedades visuais entre si. O problema central está em treinar um classificador de forma que indique a classe correta para um conjunto de imagens não conhecidas.

Na classificação de imagem existe a necessidade de uma coleção de imagens pré-classificadas e rotuladas, chamado de conjunto de treinamento. Esta coleção é utilizada no processo de aprendizagem das características que definem cada classe. Uma vez realizado esse aprendizado, pode-se classificar ou rotular imagens de interesse. A tarefa de classificação de imagem pode ser dividida nas seguintes etapas:

1. Considere um conjunto I de imagens, onde $I = \{img_1, img_2, ..., img_m\}$;
2. Considere um conjunto $C = \{c_1, c_2, ..., c_n\}$ contendo n classes;
3. Cada imagem img_i de I pertence a uma das classes do conjunto C;
4. As imagens classificadas e representadas pelo conjunto $c_j = \{img_l\}$, onde $j = 1, 2, 3, ..., n$ e $l = 1, 2, 3, ..., m$, serão utilizadas na etapa de aprendizagem do classificador (módulo que faz a classificação);
5. Uma vez o classificador "ensinado", ele pode ser usado para classificar um novo conjunto de imagens $X = \{x_1, x_2, ..., x_w\}$ definido pelo usuário.

Etapas do processo de classificação de imagens

- **Aquisição de imagem:** inicialmente deve-se construir uma base de dados de imagens de interesse;
- **pré-processamento:** nesta etapa, utilizam-se técnicas de processamento de imagem para preparar todas as imagens da base de dados. Essas técnicas

incluem, por exemplo, cropping (remover as partes que não interessam), realce e equalização do histograma (acentuar as características).

- **extração de características:** esta fase utiliza as imagens pré-processadas, ou seja, já tratadas na etapa anterior, e as organizam na forma de transações. As transações são registros de dados na forma $\{Id_{Image}, Class_{label}, F_1, F_2, ...F_n\}$ onde $F_1, F_2, ...F_n$, são n vetores de características extraídos da imagem. As transações servirão de entrada para alguma técnica de classificação. Ao final deste processo, um modelo de classificação é gerado.

- **classificação:** esta fase é a classificação propriamente dita, realizando a rotulação das novas imagens nas suas respectivas classes.

A Figura 2.6 ilustra as etapas do processo de classificação de imagens.

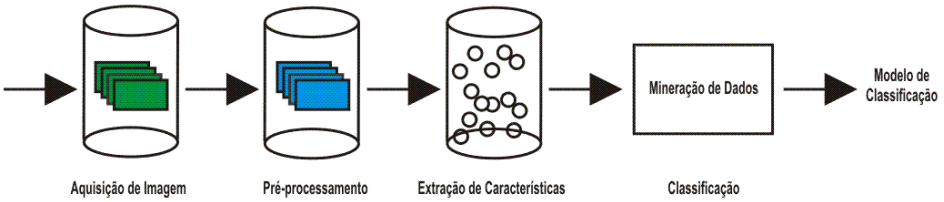


Figura 2.6: Processo de classificação de imagens. Adaptado de [5].

2.3.2 Técnicas de Aprendizagem para Classificação

As técnicas descritas nesta seção serão utilizadas como métodos de comparação nos experimentos realizados, reportados no Capítulo 3.

Bagging (BAGG)

Esta abordagem conhecida como *Bootstrap aggregation* ou *Bagging* consiste em avaliar as predições sobre uma coleção de amostras (*bootstrap samples*) que dado um conjunto inicial de treinamento T, este é dividido em B partes ou amostras iguais denotada por $Z^i, i = 1, 2, ..., B$ [27]. Cada uma dessas amostras é usada no treinamento de B classificadores que, geralmente, são os mesmos. Após o treinamento,

cada classificador obtém um coeficiente (α) que será utilizado na etapa de classificação. Dado um novo objeto a ser testado, os coeficientes referentes a cada um dos classificadores são usados e a classe atribuída ao objeto será o resultado da votação majoritária entre os B classificadores [58]. Para cada elemento x, a predição $\vec{f^i}(x)$ de cada classificador é armazenada e o resultado calculado:

$$\vec{f_{bag}}(x) = \frac{1}{B}\sum_{i=1}^{B}\vec{f^i}(x) \tag{2.1}$$

A Figura 2.7 ilustra o treinamento e a classificação da abordagem *Bagging*.

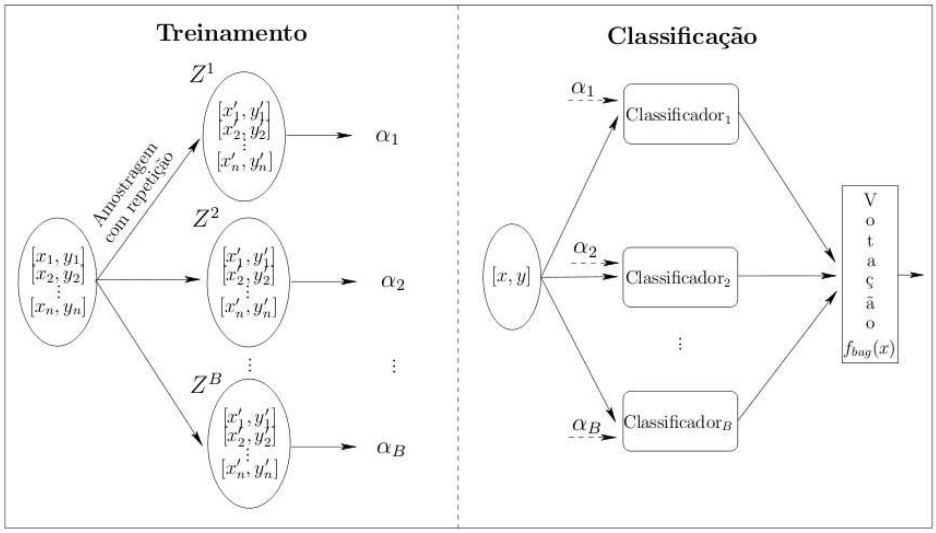


Figura 2.7: Treinamento e Classificação utilizando a abordagem *Bagging*. Retirado de [58].

K-Nearest Neighbor (KNN)

É a técnica mais simples de classificação de objetos que se baseia nos exemplos de treinamento mais próximos no espaço de característica [27]. A Equação 2.2 mostra o ajuste do kNN definido para x.

$$kNN(x) = \sum_{x_i \in \mathcal{N}_k(x)} y_i \tag{2.2}$$

onde $\mathcal{N}_k(x)$ são os k vizinhos mais próximos de x no conjunto de treinamento, y_i é o valor de distância entre x e o vizinho x_i atual (a função de distância pode ser uma simples Euclideana. A tarefa de classificação é decidida pela votação majoritária dos vizinhos mais próximos. A Figura 2.8 ilustra um exemplo de classificação usando kNN.

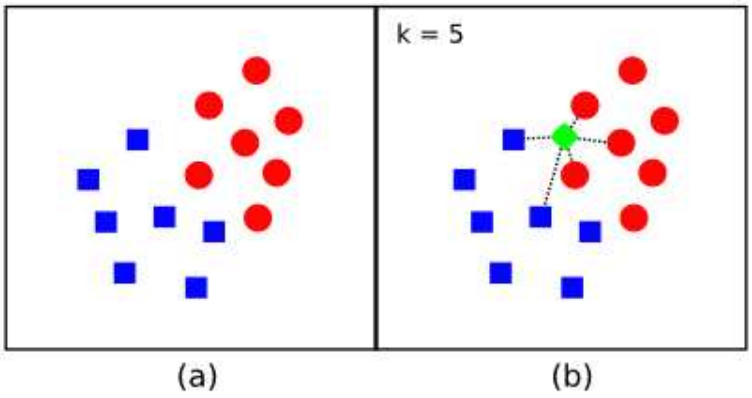


Figura 2.8: (a) Amostras de duas classes (quadrado e círculo) no espaço de características. (b) Dado um novo objeto desconhecido, os $k = 5$ vizinhos mais próximos a ele definirão sua classe, neste caso será a classe vermelha.

Linear Discriminant Analysis (LDA)

O classificador LDA [27, 58], também conhecido como Discriminante de Fischer, é uma técnica usada em estatística e aprendizagem de máquina para encontrar uma combinação linear de características que separa duas ou mais classes de objetos.

Considere duas classes C_1 e C_2 e seus conjuntos de amostras X_1 e X_2 respectivamente. Suponha que ambas as classes seguem uma distribuição Gaussiana. A

Equação 2.3 mostra o cálculo da média intra-classe de C_1 (idem para C_2).

$$\mu_1 = \frac{1}{|X_1|} \sum_{x_i \in X_1} x_i \tag{2.3}$$

A Equação 2.4 mostra o cáculo da média entre as classes.

$$\mu = \frac{1}{|X_1| + |X_2|} \sum_{x \in X_1 \cup X_2} x \tag{2.4}$$

O cálculo da matriz S_w de dispersão intra-classe e a matriz S_{bet} entre classe são mostrados nas Equações 2.5 e 2.6 respectivamente.

$$S_w = M_1 M_1^T + M_2 M_2^T \tag{2.5}$$

$$S_{bet} = |X_1|(\mu - \mu_1)(\mu - \mu_1)^T + |X_2|(\mu - \mu_2)(\mu - \mu_2)^T \tag{2.6}$$

Para maximizar a diferença entre as duas classes é necessário calcular o autovalor-autovetor generalizado $\vec{e}$ maximal de S_{bet} e S_w ($S_{bet}\vec{e} = \lambda S w\vec{e}$) para projetar as amostras em um espaço linear e aplicar um limiar para realização da classificação [58].

2.4 Recuperação de Imagens por Conteúdo (CBIR)

Esta seção mostra o conceito de recuperação de imagens por conteúdo e as técnicas utilizadas neste trabalho.

2.4.1 Visão Geral

Recuperação de imagens por conteúdo está relacionado com a noção de similaridade entre imagens. Dada uma grande base de imagens, o usuário deseja recuperar as imagens que são mais similares a uma imagem de consulta (padrão desejado). O sistema compara a consulta com as demais imagens da base e retorna para o usuário as mais similares.

2.4.2 Técnicas de Aprendizagem para Ordenação

Muitas técnicas de aprendizagem têm sido usadas para ordenação de diferentes tipos de objetos (documentos textuais e imagens) e bons resultados têm sido obtidos para documentos textuais. Os trabalhos descritos em [3, 20], por exemplo, usam Programação Genética (GP) para otimizar as funções de ordenação e obter melhores desempenhos na busca por documentos. Zobel e Mofat apresentam diversas possibilidades para o cálculo de funções de ordenação [81]. Outras abordagens baseadas em *Support Vector Machine* (SVM) têm sido propostas [11, 32, 38] para descoberta das melhores funções de busca. Já no trabalho de [72] padrões (ou regras) são encontrados associando características de documentos textuais usando Regras de Associação. Depois estas regras são usadas para ordenar documentos.

RankSVM

Recentemente [32,38], SVM foi aplicada para aprendizagem de funções de ordenação no contexto de recuperação de informação. Ela tem sido empregada especialmente para CBIR [30,34]. O processo de aprendizagem SVM usado na tarefa de recuperação segue abaixo:

Dado um espaço de entrada $X \in R^n$, onde n é o número de características e uma espaço de saída do *rank* representado pelos rótulos $Y = \{r_1, r_2, ..., r_q\}$, onde q denota número de posições do *rank*. Neste *rank* existe uma ordem, $r_q \succ r_{q-1} \succ ... \succ r_1$, onde $\succ$ representa uma relação de preferência [11].

Cada instância $\vec{x_i} \in X$ denota um par (α,β), onde α denota imagem da base e β a imagem consulta, e eles estão rotulados com um *rank* cada. Uma relação de preferência entre instâncias existentes, $\vec{x_i}$ é preterida a $\vec{x_j}$ e é denotado por $\vec{x_i} \succ \vec{x_j}$. Uma função de ordenação $f \in F$ pode ser usada para atribuir um valor de *score* para cada instância $\vec{x_i} \in X$. Essa função possibilita determinar a relação de preferência entre as instâncias do conjunto de funções F, como pode ser visto a seguir:

$$x_i \succ x_j \Leftrightarrow f(x_i) > f(x_i) \tag{2.7}$$

O problema de ordenação pode ser interpretado como um problema de aprendizagem para classificação de pares de instâncias $(\vec{x_i}, \vec{x_j})$. A classificação pode ser boa ou ruim. A relação de preferência $\vec{x_i} \succ \vec{x_j}$ é representada por um novo vetor $\vec{x_i} - \vec{x_j}$ como na Eq. 2.8.

$$\left(\vec{x}_i - \vec{x}_j, z = \begin{cases} +1 & y_i \succ y_j \\ -1 & y_j \succ y_i \end{cases}\right) \tag{2.8}$$

Para classificar cada par de imagens $(\vec{x_i}, \vec{x_j})$, duas classes são consideradas: pares corretamente (+1) e incorretamente ordenados (-1). A tarefa é selecionar a melhor função $f^* \in F$ que minimiza uma função de perda, dadas as instâncias ordenadas, resultando no modelo de ordenação SVM. Mais detalhes da descrição de aprendizagem das funções de ordenação usada em SVMs são apresentados em [11, 34, 37].

Programação Genética (PG)

A programação genética é uma técnica de inteligência artificial que vem sendo recentemente utilizada para a tarefa de recuperação de imagens por conteúdo [17, 26]. O Algoritmo 3 ilustra o seu funcionamento.

O uso de PG para combinação de descritores pode ser descrito da seguinte forma. Para um dado banco de imagens e um padrão de consulta fornecido pelo usuário, como uma imagem, o sistema retorna uma lista das imagens que são mais similares às características da consulta, de acordo com um conjunto de propriedades visuais da imagem como cor, textura ou forma. Essas propriedades são caracterizadas por descritores simples. Esses descritores são combinados usando um descritor composto D_{PG}, onde δD_{PG} é uma expressão matemática representada como uma árvore de expressão, em que os nós não-folhas são operadores numéricos (veja Tabela 2.1) e os nós folhas são um conjunto composto de valores de similaridade d_i, i=1, 2, ..., k. A Figura 2.9 mostra uma possível combinação (obtida através do framework de PG) dos valores de similaridade d_1,d_2 e d_3 de três descritores simples e a Equação 2.9 mostra a expressão correspondente.

Algoritmo 3 Algoritmo Básico de Recuperação de Imagens utilizando PG

1: Gere a população inicial de árvores "aleatórias".
2: Faça os seguintes sub-passos no treinamento de imagens para N_{gen} gerações:
3: Calcule a adequação de cada árvore de similaridade.
4: Armazene as melhores $N_{melhores}$ árvores de similaridade.
5: Crie uma nova população por: Reprodução, *Crossover* e Mutação.
6: Aplique a "melhor árvore de similaridade" (isto é., a primeira árvore da última geração) em um conjunto de imagens de teste.

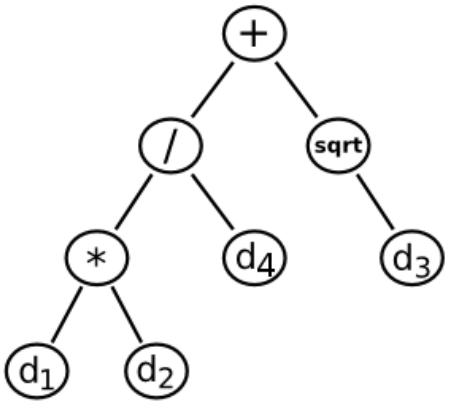


Figura 2.9: Exemplo de uma função de similaridade baseada em PG representada em uma árvore.

$$\delta D_{PG} = \frac{d_1 * d_2}{d_4} + \sqrt{d_3} \tag{2.9}$$

***Lazy Associative Classification* (LAC)**

Regras de Associação são padrões que descrevem implicações da forma $\mathcal{X} \rightarrow \mathcal{Y}$, onde $\mathcal{X}$ é o antecedente de uma regra, e $\mathcal{Y}$ o consequente. A regra não expressa uma implicação lógica clássica onde $\mathcal{X}$ necessariamente envolve $\mathcal{Y}$, mas denota a tendência de observar $\mathcal{Y}$ quando $\mathcal{X}$ é observado. Regras de Associação têm sido empregada em mineração de dados [1] e recuperação de informações textuais [72].

No contexto de ordenação de imagens, foi usado um conjunto de treinamento para associar valores de similaridade ($\mathcal{X}$) calculados do uso de descritores de imagens, a níveis de relevância (r_i). Os valores de similaridade são primeiramente discretizados usando um procedimento proposto em [24]. As regras são da forma $\mathcal{X} \rightarrow r_i$, onde $\mathcal{X}$ (o antecedente da regra) é um conjunto de valores de similaridade (potenciamente fornecido de diferentes descritores) e r_i (o consequente) é o nível de relevância.

Duas medidas são usadas para estimar a qualidade das regras:

- O suporte de $\mathcal{X} \rightarrow r_i$, representado por $\sigma(\mathcal{X} \rightarrow r_i)$, é a fração de exemplos no conjunto de treinamento que contém a característica $\mathcal{X}$ e a relevância r_i.

- A confiança de $\mathcal{X} \rightarrow r_i$, representada por $\theta(\mathcal{X} \rightarrow r_i)$, é a probabilidade condicional de r_i dado $\mathcal{X}$. Quanto mais alta confiança, mais forte é a associação entre $\mathcal{X}$ e r_i.

Para evitar uma explosão combinatorial de regras extraídas, um valor de suporte mínimo é empregado.

Para estimar a relevância de uma imagem, é necessário combinar as predições de diferentes regras [72]. A estratégia é interpretar cada uma das regras $\mathcal{X} \rightarrow r_i$ dando voto a elas ($\mathcal{X}$ para nível de relevância r_i). Os votos têm pesos diferentes, dependendo da confiança das regras correspondentes. O pesos dos votos para a relevância r_i são somados e então são calculadas suas médias considerando o total de regras associadas à r_i, como é mostrado na Equação 2.10, onde $\mathcal{R}$ é o conjunto de regras usadas no processo de votação:

$$s(r_i) = \frac{\sum\limits_{\mathcal{X} \rightarrow r_i \in \mathcal{R}} \theta(\mathcal{X} \rightarrow r_i)}{|\mathcal{R}|} \tag{2.10}$$

O valor de *score* associado à relevância r_i, $s(r_i)$, é essencialmente a média da confiança associada com as regras que prediz r_i. Finalmente, a relevância de uma imagem é estimada pela combinação linear dos *scores* normalizados associados a cada nível de relevância ($r_i \in \{0, 1\}$), conforme mostrado na Equação 2.11:

$$rank = \sum_{i \in \{0,1\}} \left(r_i \times \frac{s(r_i)}{\sum\limits_{j \in \{0,1\}} s(r_j)} \right) \tag{2.11}$$

O valor de $rank$ é uma estimativa da relevância da imagem i usando regras em $\mathcal{R}_i$, e intervalos de r_0 a r_k, onde r_0 é a menor relevância e r_k a maior.

Considere um exemplo fictício mostrado na Tabela 2.2. Existem três consultas no conjunto de treinamento e uma consulta no conjunto de teste. Para cada consulta existem três imagens recuperadas, e cada imagem é representada por três características fornecidas pelos descritores de imagem − Desc 1, Desc 2 e Desc 3. Estas medidas foram normalizadas e discretizadas. Existem nove exemplos no conjunto de treinamento, e três imagens no conjunto de teste. Considerando $\sigma_{min} = 0,2$ e $\theta_{min} = 0,67$. Neste caso, o seguinte conjunto de regras (ou modelo), $\mathcal{R}$, é gerado:

1. Desc 1=[0,23-0,44]→ r=1 (θ=1,00)

Conjunto	Imagens Consulta	Imagens Recuperadas				Relevância
		id	Desc 1	Desc 2	Desc 3	
Treinamento	carro	1	[0,23-0,44]	[0,31-0,41]	[0,57-0,64]	1
		2	[0,11-0,22]	[0,15-0,30]	[0,36-0,38]	0
		3	[0,61-0,71]	[0,01-0,07]	[0,57-0,64]	0
	frutas	4	[0,45-0,60]	[0,15-0,30]	[0,39-0,56]	0
		5	[0,82-0,89]	[0,01-0,07]	[0,58-0,65]	1
		6	[0,23-0,44]	[0,08-0,14]	[0,36-0,38]	1
	insetos	7	[0,45-0,60]	[0,15-0,30]	[0,10-0,15]	1
		8	[0,82-0,89]	[0,31-0,41]	[0,16-0,26]	1
		9	[0,72-0,81]	[0,15-0,30]	[0,36-0,38]	0
Teste	diversos	10	[0,23-0,44]	[0,15-0,30]	[0,36-0,38]	0
		11	[0,82-0,89]	[0,31-0,41]	[0,39-0,56]	0
		12	[0,11-0,22]	[0,08-0,14]	[0,16-0,26]	1

Tabela 2.2: Exemplo fictício de Consultas, Imagens Recuperadas e Relevância. Adaptado de [72].

2. Desc 1=[0,82-0,89]$\rightarrow$ r=1 (θ=1,00)

3. Desc 2=[0,15-0,30]$\rightarrow$ r=0 (θ=0,75)

4. Desc 2=[0,31-0,41]$\rightarrow$ r=1 (θ=1,00)

5. Desc 3=[0,36-0,38]$\rightarrow$ r=0 (θ=0,67)

Para calcular o *rank* da imagem 10 no conjunto de teste usando $\mathcal{R}$, as regras 2 e 4 não são aplicadas, pois a característica dela não antecedente (Desc 1=[0,82-0,89] e Desc 2=[0,31-0,41]) não estão presentes na imagem 10. Uma estratégia seria selecionar a regra com valor de θ mais alto em $\mathcal{R}_i$, e aplicar o consequente da regra selecionada como nível de relevância predito. O problema desta estratégia reside no fato de que características raras vindas de outras regras, porém importantes, podem ser ocultadas. Uma alternativa é usar todas as regras presentes em $\mathcal{R}_i$ para estimar a relevância da imagem i. Dado $\mathcal{R}_i$, o conjunto de todas as regras em $\mathcal{R}$ que são aplicadas a imagem i. Esta estratégia tem a vantagem de usar todas evidências disponíveis em $\mathcal{R}_i$, fornecendo uma melhor estimativa de *rank*.

O Algoritmo 4 mostra os passos para o cálculo de estimativa de relevância usando regras de associação.

Algoritmo 4 *Lazy Associative Classification*

Require: Exemplos em $\mathcal{D}$ e $\mathcal{T}$, *thresholds* σ_{min} e θ_{min}
Ensure: Um valor de *rank* para cada imagem $d \in \mathcal{T}$

1: $\mathcal{R} \Leftarrow$ regras extraídas de $\mathcal{D} \mid \sigma \geq \sigma_{min}, \theta \geq \theta_{min}$
2: **for all** par $(d, q) \in \mathcal{T}$ **do**
3: $\quad \mathcal{R}_i \Leftarrow$ regras $\mathcal{X} \rightarrow r_i$ em $\mathcal{R} \mid \mathcal{X} \subseteq d$
4: $\quad$ **for all** $i \mid 0 \leq i \leq k$ **do**
5: $\quad\quad s(r_i) \Leftarrow \frac{\sum_{\mathcal{X} \rightarrow r_i \in \mathcal{R}_i} \theta(\mathcal{X} \rightarrow r_i)}{|\mathcal{R}_i|}$ (i.e., Eq. 2.10)
6: $\quad$ **end for**
7: $\quad rank \Leftarrow 0$
8: $\quad$ **for all** $i \mid 0 \leq i \leq k$ **do**
9: $\quad\quad rank \Leftarrow rank + r_i \times \frac{s(r_i)}{\sum_{j=0}^{k} s(r_j)}$ (i.e., Eq. 2.11)
10: $\quad$ **end for**
11: **end for**

Para mostrar como o método funciona, suponha que se determina o valor de *rank* da imagem 10 usando o conjunto de regras (ou modelo) mostrado anteriormente. As regras aplicáveis a imagem 10 são a 1 com relevância 1 e as regras 3 e 5 com relevância 0. De acordo com a Eq. 2.10, os *scores* associados com relevâncias 0 e 1 são respectivamente $s(0)=\frac{0{,}75+0{,}67}{2}=0{,}71$ e $s(1)=\frac{1{,}00}{1}=1{,}00$. Aplicando a Eq. 2.11, o valor de *rank* da imagem 10 é dada pela $0 \times \frac{0{,}71}{1{,}00+0{,}71} + 1 \times \frac{1{,}00}{1{,}00+0{,}71}=0{,}58$.

A próxima imagem a ter seu *rank* calculado é a 11. As regras aplicáveis a imagem 11 são a 2 e 4 com relevância 1 e nenhuma regra com relevância 0. De acordo com a Eq. 2.10, os *scores* associados com relevâncias 0 e 1 são respectivamente $s(0)=0$ e $s(1)=\frac{1{,}00+1{,}00}{2}=1{,}00$. Aplicando a Eq. 2.11, o valor de *rank* da imagem 11 é dada pela $0 \times \frac{0}{1{,}00} + 1 \times \frac{1{,}00}{1{,}00}=1{,}00$.

Já na imagem 12 não existem regras aplicáveis. Para gerar regras aplicáveis à imagem 11, σ_{min} deve ser diminuído para 0,10, porém neste caso uma "explosão" de regras seriam geradas (boas e ruins). Soluções para este caso e maiores detalhes são mostrados no trabalho de Veloso et al. [72].

O uso de regras de associação na ordenação de imagens é uma das contribuições

deste trabalho.

Capítulo 3

Classificação de Imagens utilizando Programação Genética

Este capítulo apresenta o sistema de classificação de documentos textuais proposto em [76] e sua adaptação para classificação de imagens. São descritos também experimentos visando validá-lo.

3.1 Classificação baseada em Programação Genética

O sistema de classificação de documentos textuais proposto por Zhang et al. [76] consiste em 5 etapas.

1. Para cada classe, gere uma população de indivíduos (funções de similaridade);
2. Para cada classe, execute as sub-etapas seguintes no conjunto de documentos de treinamento para N_{gen} gerações
   (a) Calcule a aptidão de cada indivíduo usando o algoritmo de adequação;
   (b) Guarde os indivíduos de maiores similaridades N_{top};
   (c) Crie uma nova população por: reprodução, *crossover*, e mutação;
3. Aplique os $(N_{gen} \times N_{top})$ indivíduos (candidatos), anteriormente gravados, a um conjunto de documentos válidos e selecione o melhor indivíduo b_c que será único para cada classe C;

4. Para cada classe C, use o indivíduo b_c em um classificador kNN e aplique o resultado a um conjunto de documentos de teste;

5. Combinar a saída de cada classe por meio de votação.

As etapas 1, 2 e 3 consistem no processo de aprendizagem com o objetivo de descobrir a melhor função de similaridade para cada classe. Cada função pode ser usada somente para calcular similaridade entre pares de documentos. As etapas 4 e 5 fazem a classificação das documentos. A Figura 3.1 mostra o arcabouço de classificação. No algoritmo kNN, para um dado documento de teste d é calculado um *score*, $s_{c_i,d}$, associado a d para cada classe candidata c_i. Este *score* está definido na Eq. 3.1.

$$s_{c_i,d} = \sum_{d' \in \mathcal{N}_k(d)} similaridade(d, d') f(c_i, d') \tag{3.1}$$

onde $\mathcal{N}_k(d)$ são as k documentos mais similares a d no conjunto de treinamento e $f(c_i, d')$ é uma função que retorna 1 se a documento d' pertence a classe c_i e 0 caso contrário. No passo 4 a função "genérica" de similaridade do kNN é substituída pela função descoberta para cada classe.

No problema de classificação multi-classes com n classes o sistema utiliza n classificadores kNN. Para produzir o resultado final da classificação, a saída de todos os n classificadores são combinadas usando um esquema de votação. A classe de um documento d_i é decidida pela classe mais votada pelos n classificadores.

O sistema de classificação tem um custo elevado para encontrar uma boa solução. A complexidade de um experimento é da $O(N_{gen} \times N_{ind} \times T_{eval})$, onde N_{gen} é o número de gerações, N_{ind} é o número de indivíduos na população e T_{eval} é o tempo de avaliação de um indivíduo. O T_{eval} é determinado pela complexidade de um indivíduo ($size$) e o número de amostras de treinamento ($N_{samples}$), resultando na complexidade final $O(N_{gen} \times N_{ind} \times size \times N_{samples})$.

3.1.1 Adaptação para Classificação de Imagens

A adaptação da técnica apresentada acima para classificação de imagens consiste basicamente na troca das evidências usadas dentro do arcabouço PG. No caso de imagens, são usados descritores que caracterizam as propriedades visuais. Um in-

divíduo PG representa agora uma função de combinação de valores de similaridaes obtidos ao se usar descritores de imagens.

3.1.2 Algoritmo para Cálculo de Adequação de Indivíduos

Em [76], foram analisadas diferentes funções de adequação como a Macro F1, FFP4, Recall Class, entre outras. A partir dos resultados obtidos nestes experimentos, a função de adequação que obteve melhor desempenho, FFP4, foi selecionada para este trabalho. O Algoritmo 5 mostra os passos da função de adequação $FFP4$.

Algoritmo 5 FFP4 Algoritmo da Função de Adequaccão

1: F = 0
2: Para cada imagem D na classe C
3: $Fitness_D = FFP4$ calculado com base no conjunto de $|C|$ documentos similares a D
4: $F+ = Fitness_D$
5: $F = F/C$

Quanto maior o valor de F, melhor será a função de similaridade. A escolha da função mais apta tem influência direta no desempenho da classificação.

A seguir é definida a equação $FFP4$:

$$FFP4 = \sum_{i=1}^{|C|} r(d_i) * k_8 * (k_9)^i \tag{3.2}$$

onde $r(d)$ é a relevância de um documento, 1 se o documento é relevante e 0 caso contrário. $|C|$ é o número total de documentos similares ao documento D. Para cada documento na classe C, o valor da função de adequação é calculada baseada na Eq. 3.2, e o valor final da função é obtida da média do valor de $FFP4$ de cada documento. Os valores de k_8=7,0 e k_9=0,982 foram escolhidos por análise exaustiva no trabalho de [76].

Esta função de adequação também é usada no arcabouço para classificação de imagens.

3.2 Validação do Sistema de Classificação

Esta seção mostra os experimentos relacionados ao uso do arcabouço de classificação de imagens baseado em programação genética.

3.2.1 Configurações básicas

Bases de Imagem

Foram usadas quatro bases de imagen na avaliação. A primeira base, FreeFoto, foi extraída de uma base de imagem contendo 129.559 imagens com 171 seções organizadas em 3.542 classes da base *FreeFoto.com*. O subconjunto é formado por 3.462 imagens. Existem 9 classes de imagens e o número de imagens por classe varia de 70 a 854 imagens.

A segunda base, Borboletas, é uma base constituída de 165 imagens de asas de borboletas e 7 classes. Essa base foi sedida por colegas do Instituto de Biologia da Unicamp [39].

A terceira base, Café, é uma imagem (Figura 3.4(a)) capturada pelo satélite SPOT do município de Monte Santos de Minas (MG), uma tradicional região montanhosa produtora de café. Nesta região as plantações de café são normalmente localizadas em uma pequena parte da fazenda e foi definido que 75×75 metros (corresponde a uma subimagem de 30×30 pixels) é um bom valor para o tamanho de cada subimagem. Foram geradas 6400 subimagens para classificar e cada subimagem pode pertencer a uma das 2 classes: plantação de café (com mais de 50% dos seus pixels de café) e não café (com menos de 50% dos seus pixels de café).

Descritores de Imagem

A Tabela 3.1 lista o conjunto de descritores usados em todos os experimentos.

Medidas de Avaliação

A medida de avaliação adotada nesses experimentos foi a média da diagonal principal da matriz de confusão (Acc). Considerando que existam duas classes de imagens, $c1$ e $c2$, o cálculo de Acc será:

$$Acc = \frac{A_{c1} + A_{c2}}{N} \tag{3.3}$$

Descritores	Tipo de Evidência	Bases		
		FreeFoto	**Borboletas**	**Café**
GCH [68]	Cor	X	X	X
BIC [65]	Cor	X	X	X
CCV [56]	Cor	X	X	
JAC [73]	Cor			X
LAS [69]	Textura	X	X	
HTD [51, 74]	Textura	X	X	
QCCH [35]	Textura			X
SID [75]	Textura			X

Tabela 3.1: Os descritores de imagem usados em cada uma das bases.

onde A_{c1} e A_{c2} são as quantidades de imagens corretamente classificadas para as classes $c1$ e $c2$, respectivamente e N é o total de imagens do conjunto de teste. Mais detalhes, em [18].

Nos experimentos com imagens de sensoriamento remoto a medida de avaliação adotada foi o índice kappa. Kappa é a medida utilizada para comparar a eficácia da classificação em imagens de ISR [14].

$$kappa = \frac{total * (truePositive + trueNegative) - (falsePositive + falseNegative)}{total^2 - (falsePositive + falseNegative)} \tag{3.4}$$

Metodologia e Parâmetros

Nos experimentos realizados os conjuntos de treinamento, validação e teste foram criados utilizando a abordagem chamada "5-fold cross-validation" [58] para a base FreeFoto. A base foi dividida em 5 conjuntos (*folds*) de treinamento, validação e teste totalizando 5 *folds*.

No caso da base borboletas, não foi usado"cross-validation" devido ao pequeno número de imagens presentes na base. Foi selecionado o mesmo número de elementos por classe para o conjunto de treinamento.

Nos experimentos com imagens de sensoriamento remoto, devido ao pouco tempo para a realização dos experimentos, as imagens foram separadas em 2 folds, cada um composto por conjuntos de treinamento, validação e teste (todos conjuntos escolhi-

dos aleatoriamente proporcional ao tamanho de cada classe) sendo cada um deles contendo 15%, 5% e 80% da base, respectivamente.

A Tabela 3.2 mostra os parâmetros PG usados no sistema. Os valores de cada parâmetro foram determinados empiricamente, após vários experimentos.

	Bases		
Parâmetro	**FreeFoto**	**Borboletas**	**Café**
kNN	1 e 3	7	13
População	5	5	30 e 50
Gerações	10	20	15 e 30
Reprodução	0,30	0,30	0,30
Crossover	0,65	0,65	0,80
Mutação	0,05	0,05	0,05
Funções	+, -, *, /, sqrt		
Função de Adequação	FFP4		

Tabela 3.2: Os parâmetros do sistema de classificação para cada uma das bases.

3.2.2 Resultados

Esta seção mostra os resultados obtidos na comparação de diferentes técnicas de classificação da literatura.

Efeito da Programação Genética na Combinação de Evidências Visuais

Muitos estudos mostraram bons resultados no uso da programação genética para combinar diferentes evidências visuais (cor e textura) [17, 26]. Esta seção comprova o bom desempenho da PG, por meio do classificador PG+KNN, comparados às outras técnicas. O sufixo "KNN-'i" significa que foram considerados i-vizinhos mais próximos do classificador kNN. O SVM-LINEAR é o classificador SVM que está restrito a um hiperplano linear. Já o SVM-RBF utiliza funções radiais de base para resolver problemas que não são resolvidos com hiperplano linear. A diferença entre o BAGG-7 e BAGG-13 está na quantidade de repetições (iterações) realizadas em cada um dos classificadores. A Tabela 3.3 mostra a eficácia das técnicas. Nota-se que nem todas as técnicas obtêm êxito quando tentam combinar diferentes evidências visuais, ou seja, a eficácia das técnicas quando utilizam menos descritores é maior (compare as Tabelas 3.3 e 3.4).

Classificadores	Eficácia (%)	Desvio
PG+KNN-3	92,17	1,06
SVM-RBF	90,81	1,10
BAGG-13	90,18	0,55
SVM-LINEAR	88,53	1,08
BAGG-7	87,93	0,70
LDA	85,04	1,68
KNN-1	84,43	0,75
KNN-3	80,39	2,44
KNN-7	77,76	1,95
KNN-13	75,13	2,83

Tabela 3.3: Comparação entre os classificadores utilizando a base FreeFoto e cinco descritores (BIC, CCV, GCH, HTD e LAS).

Imagens Heterogêneas da Base FreeFoto

Nos experimentos realizados foi possível verificar que o classificador PG+KNN apresenta bons resultados quando comparado com outras técnicas de classificação estudadas. A Tabela 3.4 mostra a comparação dos classificadores utilizados nos experimentos. Note, no entanto, o bom desempenho do dos classificadores KNN e SVM-RBF quando apenas um descritor é usado.

Classificador	**Eficácia (%)**	**Desvio**	**Descritores**
PG + KNN-3	92,17	1,06	5 (BIC, CCV, GCH, HTD e LAS)
KNN-1	91,59	0,72	1 (BIC)
SVM-RBF	91,10	1,70	1 (BIC)
BAGG-13	90,21	1,17	3 (BIC, CCV, GCH)
BAGG-7	88,88	0,68	3 (BIC, CCV, GCH)
KNN-3	88,73	0,77	1 (BIC)
SVM-LINEAR	88,53	1,08	5 (BIC, CCV, GCH, HTD e LAS)
KNN-7	85,50	1,16	1 (BIC)
LDA	85,04	1,68	5 (BIC, CCV, GCH, HTD e LAS)
KNN-13	82,96	1,94	1 (BIC)

Tabela 3.4: Os melhores valores de eficácia de cada um dos classificadores para a base FreeFoto.

Imagens de Borboletas

Nos experimentos com imagens de borboletas foi possível comprovar a dificuldade que as diferentes técnicas de classificação encontram quando utilizadas em pequenos conjuntos de treinamento. A Tabela 3.5 mostra que o PG+KNN obteve resultados muito bons quando comparados ao desempenho de outras técnicas.

Classificador	**Eficácia (%)**	**Desvio**	**Descritores**
PG + KNN-7	60,80	2,06	5 (BIC, CCV, GCH, HTD e LAS)
KNN-7	56,80	13,68	1 (BIC)
SVM-RBF	54,40	2,19	1 (BIC)
BAGG-7	54,40	16,15	1 (BIC)
BAGG-13	52,00	11,66	1 (BIC)
SVM-LINEAR	49,60	4,56	1 (BIC)
KNN-3	46,40	9,21	1 (BIC)
KNN-13	45,60	6,07	1 (BIC)
KNN-1	44,00	6,32	1 (BIC)

Tabela 3.5: Os valores de eficácia de cada um dos classificadores para a base Borboletas.

Imagens de Sensoriamento Remoto

O uso de Imagens de Sensoriamento Remoto (ISR) como fonte de informação em aplicações de na Agricultura é muito comum. Nestas aplicações, é fundamental conhecer como é o espaço de ocupação. Entretanto, o reconhecimento de alguns tipos de regiões de plantação não é fácil. O local ou a idade da plantação pode atrapalhar o processo de reconhecimento. Nestes casos, a resposta espectral e o padrão de textura do mesmo tipo de plantação pode ser diferentes. Uma plantação pode ser plantada de modo diferentes e aliado a este fator, os diferentes estágios da planta criam distinção nas ISR entre regiões da mesma classe.

A abordagem proposta, extensão do trabalho de [60], pode ser dividida em duas fases (a Figura 3.3 ilustra estas fases): (i) a descrição da imagem e (ii) classificação da imagem. A descrição da imagem consiste na caracterização do conteúdo da imagem que é realizada off-line. Primeiro, as imagens são selecionadas e inseridas no sistema (passo 1 na Figura 3.3). Esta imagem é dividida em várias subimagens retangulares

(passo 2). Finalmente, descritores são usados para extrair propriedades espectral e de textura (passo 3).

O processo de classificação inclui os passos 4, 5, 6 e 7 na Figura 3.3. O processo de identificação das subimagens relevantes é realizada usando PG para combinar as similaridades fornecidas pelos descritores. Cada subimagem é considerada uma imagem independente e este processo se inicia pela indicação de amostras relevantes (passo 4). Assume-se que estas amostras contenham as mesmas propriedades espectral e de textura das regiões de interesse. Um reconhecimento de padrão é realizado usando PG e todas as subimagens são rotuladas (passo 5). Após a classificação das subimagens, a segmentação das regiões relevantes é feita (passo 7). Este processo de segmentação é realizado usando um algoritmo "watershed" que segmenta imagens a partir de sementes. Estas sementes são escolhidas das áreas de interesse identificadas no passo anterior.

Para avaliar a eficácia do sistema de classificação, foi usado uma máscara (Figura 3.4(b)) que indica todas as plantações de café na imagem da Figura 3.4(a). As plantações de café mostradas na máscara foram identificadas manualmente por especialistas. A imagem tem vários rótulos que podem ser facilmente confundidos com café, mas que são matas nativas, cana-de-açúcar, entre outros. A Figura 3.4(c) ilustra exemplos de plantações de café e não café. Note a similaridade das amostras de café e não café.

O sistema proposto é comparado com a classificação *Maximum Likelihood* [63] (MaxVer) com probabilidade "threshold" 0,98 e usando 43.630 pontos de amostra de café. Esta classificação é o método mais usado em dados de ISR. O resultado de eficácia Kappa para o MaxVer foi 66,00. A Tabela 3.6 mostra os resultados de 4 experimentos (Exp), em cada um com 2 *folds* e a média deles.

De acordo com a Tabela 3.6, é possível notar que a melhor eficácia observada foi de 66,33 que é ligeiramente maior que o melhor resultado alcançado pelo MaxVer (66,00). Este resultado é encarado como promissor, considerando que foram realizados poucos testes e a avaliação de outros valores para diversos parâmetros da PG precisa ser estendida.

Exp	População	Geração	*Fold* 1	*Fold* 2	Média
1	30	15	65,92	66,04	65,98
2	30	30	64,88	66,51	65,70
3	50	15	65,90	66,46	66,18
4	50	30	66,37	66,29	**66,33**

Tabela 3.6: Resultados *Kappa* para reconhecimento de café aplicando o sistema proposto.

3.3 Problemas Identificados

Um dos problemas identificados no sistema de classificação diz respeito à sua instabilidade para diferentes composições dos *folds*.

3.3.1 Instabilidade do Classificador

A Tabela 3.7 ilustra um exemplo dessa instabilidade considerando experimentos envolvendo as bases FreeFoto e Borboletas. Note, por exemplo, como a eficácia do classificador é baixa para os *folds* 1 e 2 quando comparada com os *folds* 3, 4 e 5 considerando-se a base FreeFoto. Na base Borboleta, o mesmo fenômeno foi observado para o *fold* 2. Para este *fold*, o classificador apresentou eficácia bem menor que os outros *folds*.

	Eficácia (%)	
fold	**FreeFoto**	**Borboletas**
1	79,05	56,00
2	73,70	36,00
3	90,17	64,00
4	88,87	52,00
5	87,18	52,00
Média	83,79	50,40
Desvio	7,11	10,81

Tabela 3.7: Exemplo de instabilidade do sistema de classificação nas bases FreeFoto e Borboletas.

3.3.2 Questões de Eficiência

Durante os experimentos foi testada uma modificação no sistema de classificação visando ganho de tempo de execução. Essa modificação foi o chamado "balanceamento" que consiste em adicionar à etapa 2 do algoritmo, mostrado na seção 3.1, uma mudança na quantidade de imagens pertencentes ao conjunto de treinamento. Ou seja, a quantidade de imagens da classe relevante é igual à das classes não relevantes. Essa modificação levou à redução do tempo gasto na execução.

A Tabela 3.8 mostra uma comparação entre o sistema original e o sistema com balanceamento, na base FreeFoto.

folds	Eficácia (%)		Tempo (minutos)	
	PG+KNN-3 (original)	**PG+KNN-3 (balanceado)**	**PG+KNN-3 (original)**	**PG+KNN-3 (balanceado)**
1	89,88	91,47	39	7
2	91,04	90,46	39	7
3	91,76	90,61	39	6
4	91,76	90,90	40	7
5	88,18	88,90	37	7
Média	90,52	90,47	38,80	6,80
Desvio	1,52	0,96	1,10	0,45

Tabela 3.8: Comparação entre o sistema original e o balanceado.

A redução do tempo nos experimentos com a base Borboletas não foi muito significativa (ganho de 1s) devido à quantidade de imagens, porém na base FreeFoto o tempo melhorou 83%.

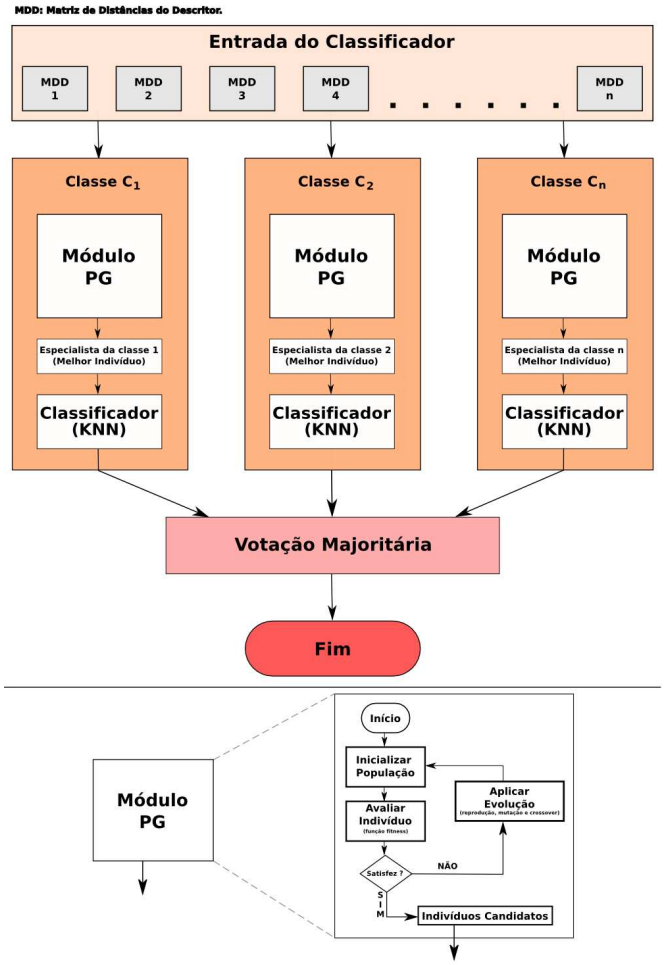


Figura 3.1: Arcabouço de classificação de documentos.

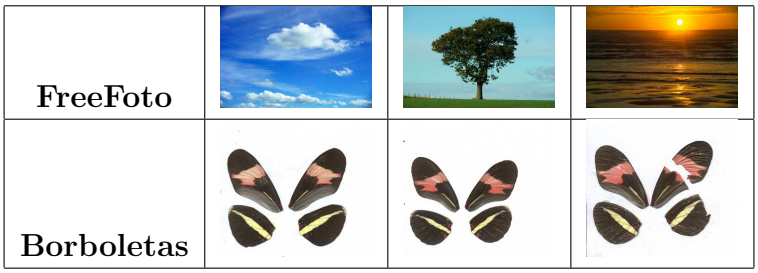


Figura 3.2: Exemplo de imagens das Bases de Imagens (FreeFoto, Fruits e Borboletas) usadas.

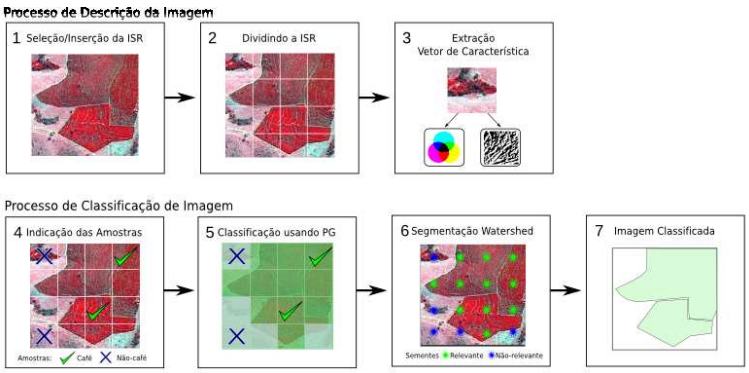
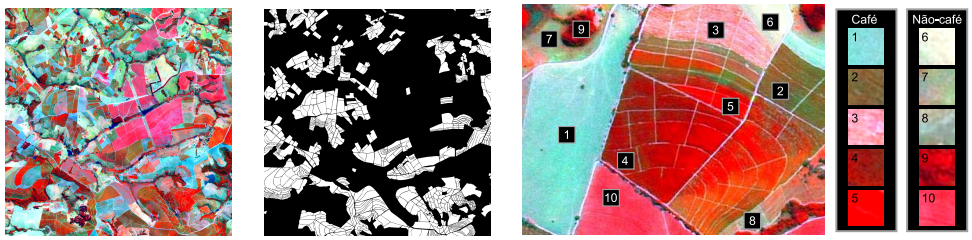


Figura 3.3: Etapas da classificação. Adaptado de [60]



(a) (b) (c)

Figura 3.4: (a) Imagem de plantações de café capturada do satélite SPOT correspondente ao município de Monte Santo de Minas (MG), (b) a máscara correspondente e (c) são diferentes tipos de evidências visuais para plantações de café. Adaptado de [60]

Capítulo 4

Técnicas de Aprendizagem para Recuperação de Imagem

Este capítulo descreve os experimentos realizados que têm como objetivo comparar técnicas de aprendizagem (em especial, Programação Genética) em tarefas de recuperação de imagens por conteúdo. Os resultados aqui reportados foram publicados em [23].

4.1 Definição do problema

No domínio de Recuperação de Imagens por conteúdo, o uso de descritores de imagens é a base para seu funcionamento e diferentes descritores podem caracterizar diferentes propriedades visuais (cor, textura ou forma). Além disso, descritores diferentes podem caracterizar diferentemente uma mesma propriedade visual. Certamente um descritor pode ser mais ou menos eficaz que outro em uma dada coleção, mas não existe um descritor "perfeito" para qualquer base de imagens.

O objetivo desses experimentos é avaliar o desempenho da técnica de PG quando comparada com outras técnicas de aprendizagem na tarefa de recuperação de imagens combinando diferentes evidências visuais (cor, textura e forma) fornecidas por descritores de imagem. Para realizar a combinação desses descritores foram usadas três técnicas de aprendizagem Programação Genética, *Support Vector Machine*, e Regras de Associação que foram chamados de CBIR-PG, CBIR-SVM e CBIR-RA, respectivamente. A adaptação da técnica baseada em Regras de Associação para o

problema de recuperação de imagens por conteúdo é uma contribuição original do trabalho. Este capítulo apresenta os resultados experimentais para a avaliação das técnicas de aprendizagem para a tarefa de recuperação imagens. Inicialmente serão mostradas informações gerais sobre como foram conduzidos os experimentos (base de imagens, medidas de avaliação, parâmetros, entre outros), e em seguida serão mostrados os resultados obtidos.

O problema é apresentado como um problema clássico de aprendizagem, onde o conjunto de de exemplares anotados é composto por imagens consulta e o seus *ground-truth* correspondentes (i.e., o graus de relevância da imagem na base para a imagem consulta). O grau de relevância pode ter valor 1 ou 0, ou seja, imagem relevante e não relevante para a consulta, respectivamente. Cada umas das técnicas de aprendizagem utilizam matrizes de distâncias, pré-calculadas, entre as imagens consulta e todas as imagens da base para cada um dos descritores relacionados para os experimentos.

4.2 Projeto Experimental

4.2.1 Configurações básicas

Bases de Imagem

Foi usado um subconjunto de duas grandes bases de imagens na avaliação. A primeira base, Corel, foi extraída de uma base de imagem contendo 20.000 imagens da *Corel GALLERY Magic - Stock Photo Library 2*. O subconjunto é formado por 3.906 imagens e 123 imagens consultas. Existem 85 classes de imagens e o número de imagens por classe varia de 7 a 98 imagens.

A segunda base, Caltech, contém 8.677 imagens coloridas extraídas da base de imagens Caltech101 [45]. Estas imagens são agrupadas em 101 classes (imagens de aviões, câmeras, formigas, cérebros, entre outras.) e o número de imagens por classes variam de 40 a 800. Foram usadas 122 imagens consultas escolhidas aleatoriamente dentre as classes, garantindo a existência de pelo menos 1 imagem de cada classe.

Em ambas as bases, as classes são mutuamente exclusivas, ou seja, uma imagem pertence a apenas uma classe. Uma imagem é considerada relevante para uma imagem de consulta se ambas pertencem a uma mesma classe.

Para cada base de imagens, uma matriz de valores de similaridade entre o par

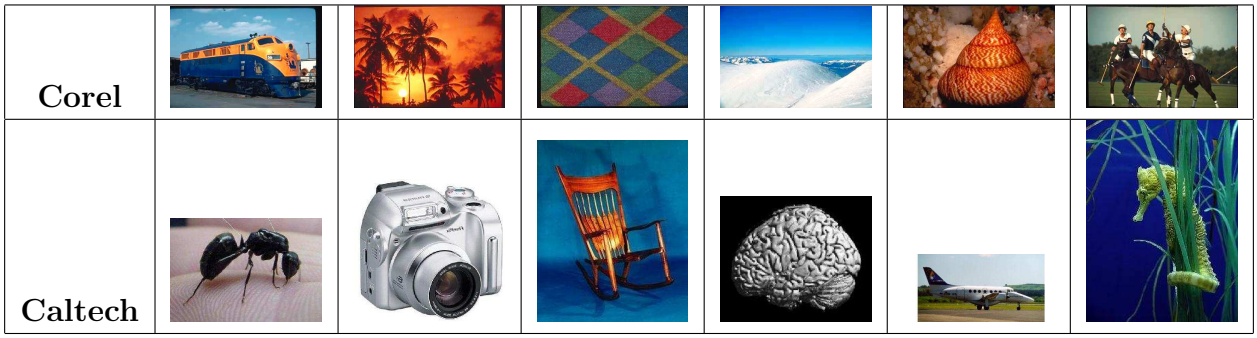


Figura 4.1: Exemplo de imagens das Bases de Imagens (Corel e Caltech) usadas.

de imagens foi calculada, usando cada um dos dezoito descritores mostrados na Tabela 4.1.

A Figura 4.2 mostra a porcentagem de imagens relevantes para cada imagem consulta usada. Pode ser notada nesta figura a dificuldade de recuperação das imagens em cada uma das bases e assim, o desafio para o sistema de CBIR obter bons resultados.

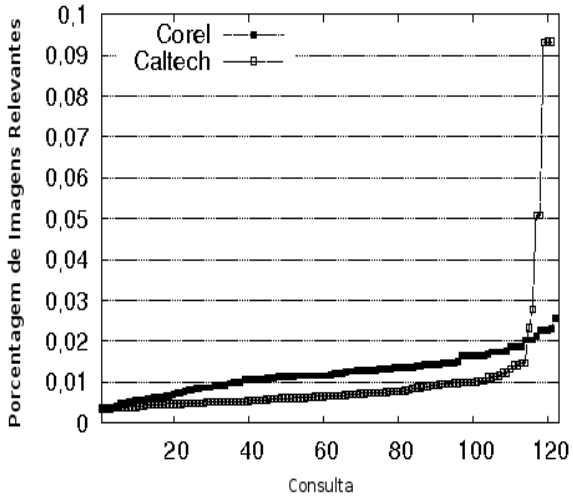


Figura 4.2: Distribuição das imagens relevantes nas bases de imagens.

A distribuição indica a abundância/escassez de respostas corretas para cada consulta. Isto tem impacto no grau de dificuldade na tarefa de recuperação (mostra que a base Caltech é mais desafiadora), pois o número de imagens relevantes para cada

consulta é muito inferior ao tamanho da base.

Descritores de Imagem

A Tabela 4.1 lista o conjunto de descritores usados nos experimentos. A coluna Tipos de Evidência indica a qual propriedade visual o descritor está associado.

Descritor	Tipo de Evidência
GCH [68]	Cor
BIC [65]	Cor
COLORBITMAP [47]	Cor
ACC [36]	Cor
CCV [56]	Cor
CGCH [66]	Cor
CSD [50]	Cor
JAC [73]	Cor
LCH [68]	Cor
CCOM [40]	Textura
LAS [69]	Textura
LBP [54]	Textura
QCCH [35]	Textura
SASI [12]	Textura
SID [75]	Textura
UNSER [71]	Textura
EOAC [49]	Forma
SPYTEC [44]	Forma

Tabela 4.1: Os dezoito descritores de imagem usados nos experimentos.

Medidas de Avaliação

Para avaliar a eficácia de cada uma das técnicas, duas métricas foram usadas: precisão (tomados em algumas posições do topo da lista ordenada), e MAP (*Mean Average Precision*). São essas medidas de avaliação muito utilizadas em sistemas de Recuperação da Informação (RI) [11].

- **Precisão**: é a razão entre as imagens relevantes ($r(j)$) pelo total de imagens

retornadas (j).

$$P(j) = \frac{r(j)}{j} \tag{4.1}$$

- **MAP**: calcula a precisão média de um conjunto de consultas. Dada uma consulta q_i, a precisão média é dada pela média das precisões de cada instância relevante (n_r) recuperada (medida sumarizada da curva precision x recall).

$$AvgP_i = \sum_{j=1}^{M} \frac{P(j) \times pos(j)}{|n_r|} \tag{4.2}$$

onde j é o $rank$, M é o número de instâcias retornadas, $pos(j)$ indica 1 se instância relevante e 0 caso contrário e $P(j)$ é o valor de precisão no $rank$ com corte j.

Ambas as medidas enfatizam a qualidade no topo da lista ordenada, porém o efeito é menos observado na MAP por considerar a lista inteira imagens. A medida precisão é mais focada no topo da lista ordenada. Para mais detalhes sobre as medidas usadas nos experimentos, recomenda-se a consulta das referências [11, 46].

Metodologia e Parâmetros

Nos experimentos realizados os conjuntos de treinamento, validação e teste foram criados utilizando a abordagem "5-fold cross-validation" [58] para todas as bases (Corel e Caltech). As bases foram divididas em 5 conjuntos de treinamento, validação e teste, totalizando 5 *folds*. As Tabelas 4.2 e 4.3 mostram os parâmetros utilizados por cada uma das técnicas nos experimentos realizados.

Avaliação dos Descritores de Imagem

A Tabela 4.4 mostra os valores de precisão para a base de imagem Corel, considerando os dezoitos descritores (estes apresentados na Tabela 4.1). Pode-se observar que o descritor BIC apresenta os melhores resultados em termos de valores de precisão para diferentes números de imagens retornadas. Para a base Caltech, foram observados resultados similares.

Considerando esses resultados, o descritor BIC foi usado como referência na comparação das técnicas de aprendizagem. O objetivo ao se incluir o descritor BIC é

Run	**Corel**		**Caltech**	
	CBIR-RA (σ_{min})	CBIR-SVM (C)	CBIR-RA (σ_{min})	CBIR-SVM (C)
1	0,0025	0,2000	0,0005	0,9000
2	0,0050	0,2000	0,0001	0,2000
3	0,0050	0,0900	0,0002	0,0900
4	0,0025	0,1000	0,0001	30.000
5	0,0075	10,000	0,0001	0,0900

Tabela 4.2: Os parâmetros usados nas técnicas CBIR-RA e CBIR-SVM, para cada base de imagem.

Crossover	0,85
Gerações	30
Mutação	0,10
População	600
Reprodução	0,05
Tree Depth	8

Tabela 4.3: Os parâmetros usados na técnica CBIR-PG, para cada base de imagem.

confirmar que os resultados da combinação de diferentes descritores na recuperação de imagens são melhores que a recuperação utilizando um único descritor.

4.2.2 Resultados

Primeiramente, serão mostrados os resultados obtidos da avaliação de todos os descritores usados nos experimentos (Seção 4.2.1), em seguida as análises de alta (média por consulta e por conjunto) e mais granularidade (correlação entre as técnicas de aprendizagem).

Imagens Heterogêneas

A Tabela 4.5 mostra os valores de MAP para as bases de imagem Corel e Caltech. O resultado para cada *run* é calculado pela média dos resultados parciais considerando cada consulta na *run*. O resultado final é obtido pela média aritmética dos cinco *runs*. Uma *run* é o experimento realizado utilizando a tupla (*fold*, parâmetro) que a corresponde e cada *fold* é composto por 1 conjunto de treinamento, validação e

teste. Por exemplo, na Tabela 4.5 o valor MAP da *run* 1 para a técnica CBIR-RA é referente ao resultado do experimento utilizando o *fold* 1 e o parâmetro da Tabela 4.2.

Para a base Corel, CBIR-RA e CBIR-PG obtiveram os melhores resultados de eficácia. Na média, CBIR-RA e CBIR-PG apresentam desempenhos similares e superiores a CBIR-SVM e BIC. O CBIR-RA mostrou-se melhor 4,2% e 21,0% quando comparado com CBIR-SVM e BIC, respectivamente. Já o CBIR-PG, mostrou-se melhor 4,7% e 21,6%, quando comparado com CBIR-SVM e BIC, respectivamente.

Resultados similares ocorreram na base Caltech. Novamente, CBIR-RA e CBIR-PG apresentaram as melhores médias. Especificamente, para a *run* 3, CBIR-RA e CBIR-SVM estão empatados. O CBIR-RA mostrou-se melhor 14,3% e 24,0% quando comparado com CBIR-SVM e BIC, respectivamente. Já o CBIR-PG, mostrou-se melhor 9,7% e 19,8%, quando comparado com CBIR-SVM e BIC, respectivamente.

O próximo conjunto de experimentos avalia a eficácia do CBIR-RA, CBIR-PG, CBIR-SVM, e BIC em termos de precisão. Tabela 4.6 mostra os valores de precisão obtido por cada uma das técnicas.

Na base Corel, CBIR-RA apresentou melhor desempenho nas primeiras quatro posições do *ranking*. Entretanto, CBIR-PG mostrou-se melhor nas últimas posições. Para a base Caltech, CBIR-RA apresentou melhor desempenho em todas as posições de precisão considerada.

Em relação à medida MAP, verificou-se um empate estatístico entre CBIR-RA e CBIR-PG, ambas melhores que CBIR-SVM.

Uma análise mais detalhada foi realizada visando comparar as três técnicas de aprendizagem. A Figura 4.3 mostra os valores MAP de cada uma das técnicas para cada consulta. As coordenadas associadas com cada ponto em um dos gráficos são dados pelos valores MAP obtidos por duas técnicas indicadas nos eixos. Por exemplo, o ponto p_1 (indicado no topo do gráfico) representa uma consulta na base Corel para o qual CBIR-RA consegue um valor MAP de 1,0 e CBIR-PG consegue um valor MAP de 0,10 (eixo x). Do mesmo jeito, o ponto indicado p_2 representa outra consulta, na qual CBIR-RA consegue um MAP de 0,01 e CBIR-PG consegue um MAP de 1,0.

Nesse gráfico pode ser visto que as técnicas CBIR-PG e CBIR-SVM são fortemente correlacionadas (coeficiente de correlação acima de 0,90), indicando que essas técnicas conseguem desempenhos similares considerando as mesmas consultas. No entanto, o desempenho de CBIR-RA é pouco correlacionado com as outras duas técnicas. Ou seja, CBIR-RA apresenta eficácia melhor quando as outras obtêm eficácia ruim e vice-versa. Este fenômeno é observado nas duas bases de imagens uti-

Descritores	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10	Média
ACC	1,000	0,514	0,405	0,325	0,271	0,234	0,196	0,162	0,129	0,088	0,332
BIC	**1,000**	**0,536**	**0,425**	**0,349**	**0,293**	**0,249**	**0,218**	**0,179**	**0,139**	**0,096**	**0,348**
CCOM	1,000	0,344	0,259	0,223	0,181	0,154	0,128	0,106	0,088	0,064	0,255
CCV	1,000	0,380	0,255	0,203	0,175	0,153	0,118	0,094	0,076	0,052	0,251
CGCH	1,000	0,222	0,130	0,095	0,079	0,063	0,055	0,048	0,039	0,034	0,176
CSD	1,000	0,355	0,233	0,168	0,130	0,107	0,091	0,075	0,066	0,055	0,228
EOAC	1,000	0,234	0,147	0,107	0,091	0,077	0,065	0,057	0,048	0,039	0,186
GCH	1,000	0,404	0,263	0,213	0,183	0,156	0,129	0,104	0,074	0,044	0,257
JAC	1,000	0,467	0,344	0,276	0,231	0,182	0,147	0,119	0,090	0,052	0,291
LAS	1,000	0,207	0,134	0,103	0,084	0,066	0,055	0,046	0,041	0,034	0,177
LBP	1,000	0,139	0,083	0,064	0,056	0,046	0,040	0,038	0,035	0,032	0,153
LCH	1,000	0,351	0,243	0,190	0,162	0,135	0,114	0,093	0,075	0,058	0,242
LUCOLOR	1,000	0,316	0,199	0,155	0,132	0,110	0,093	0,078	0,064	0,052	0,220
QCCH	1,000	0,154	0,093	0,069	0,061	0,054	0,049	0,044	0,039	0,035	0,160
SASI	1,000	0,289	0,205	0,154	0,125	0,102	0,081	0,068	0,055	0,043	0,212
SID	1,000	0,150	0,087	0,069	0,059	0,051	0,044	0,040	0,036	0,031	0,157
SPYTEC	1,000	0,036	0,029	0,026	0,026	0,025	0,025	0,024	0,024	0,023	0,124
UNSER	1,000	0,086	0,043	0,034	0,032	0,030	0,029	0,028	0,027	0,026	0,133

Tabela 4.4: Valores de precisão para a base Corel. Os melhores resultados estão em negrito.

Run	Corel				Caltech			
	CBIR-RA	**CBIR-PG**	**CBIR-SVM**	**BIC**	**CBIR-RA**	**CBIR-PG**	**CBIR-SVM**	**BIC**
1	**0,405**	0,399	0,374	0,279	0,051	**0,059**	0,051	0,058
2	0,312	**0,328**	0,308	0,298	**0,106**	0,057	0,038	0,018
3	0,366	**0,376**	0,351	0,341	**0,098**	0,089	**0,098**	0,093
4	**0,362**	0,354	0,344	0,290	0,046	0,049	0,039	**0,061**
5	**0,250**	0,246	0,251	0,193	0,064	**0,098**	0,092	0,063
Final(Avg)	**0,339**	**0,341**	0,326	0,280	**0,073**	**0,070**	0,064	0,058
CBIR-RA	-	-0,5%	4.2%	21,0%	-	4.1%	14.3%	24.0%
CBIR-PG	0,5%	-	4.7%	21,6%	-4.1%	-	9.7%	19.8%

Tabela 4.5: Valores de MAP para as bases Corel e Caltech. Os melhores resultados estão em negrito. Os valores em porcentagem representam os ganhos relativos entre as técnicas.

@	Corel				Caltech			
	CBIR-RA	**CBIR-PG**	**CBIR-SVM**	**BIC**	**CBIR-RA**	**CBIR-PG**	**CBIR-SVM**	**BIC**
1	**0,772**	0,711	0,750	0,633	**0,273**	0,196	0,206	0,237
2	**0,699**	0,660	0,668	0,606	**0,247**	0,196	0,184	0,196
3	**0,651**	0,641	0,622	0,569	**0,230**	0,197	0,182	0,178
4	**0,625**	0,620	0,597	0,545	**0,211**	0,185	0,175	0,169
5	0,602	**0,604**	0,581	0,537	**0,197**	0,174	0,166	0,155
6	0,593	**0,593**	0,564	0,515	**0,193**	0,165	0,155	0,149
7	0,582	**0,584**	0,556	0,495	**0,182**	0,160	0,151	0,140
8	0,561	**0,573**	0,545	0,481	**0,175**	0,163	0,147	0,134
9	0,549	**0,560**	0,531	0,476	**0,166**	0,163	0,146	0,132
10	0,540	**0,550**	0,521	0,464	**0,161**	0,156	0,141	0,127

Tabela 4.6: Valores precisão para as bases Corel e Caltech. Os melhores resultados estão em negrito.

lizadas em nossos experimentos (coeficiente de corelação mostrado na Tabela 4.7).

Uma inspeção manual das consultas revelou que para CBIR-PG e CBIR-SVM, a propriedade chave que leva um bom desempenho é o número de imagens relevantes para cada consulta. Especificamente, CBIR-PG e CBIR-SVM obtêm os mais altos valores de MAP em consultas com muitas imagens relevantes, e mais baixo em consultas com poucas imagens relevantes. Já o CBIR-RA obtém melhor desempenho sempre que as consultas têm poucas imagens relevantes.

Técnicas	**Corel**	**Caltech**
CBIR-RA×CBIR-PG	-0,064	0,318
CBIR-RA×CBIR-SVM	-0,091	0,257
CBIR-PG×CBIR-SVM	0,978	0,981

Tabela 4.7: Coeficientes de correlação entre números MAP para cada consulta.

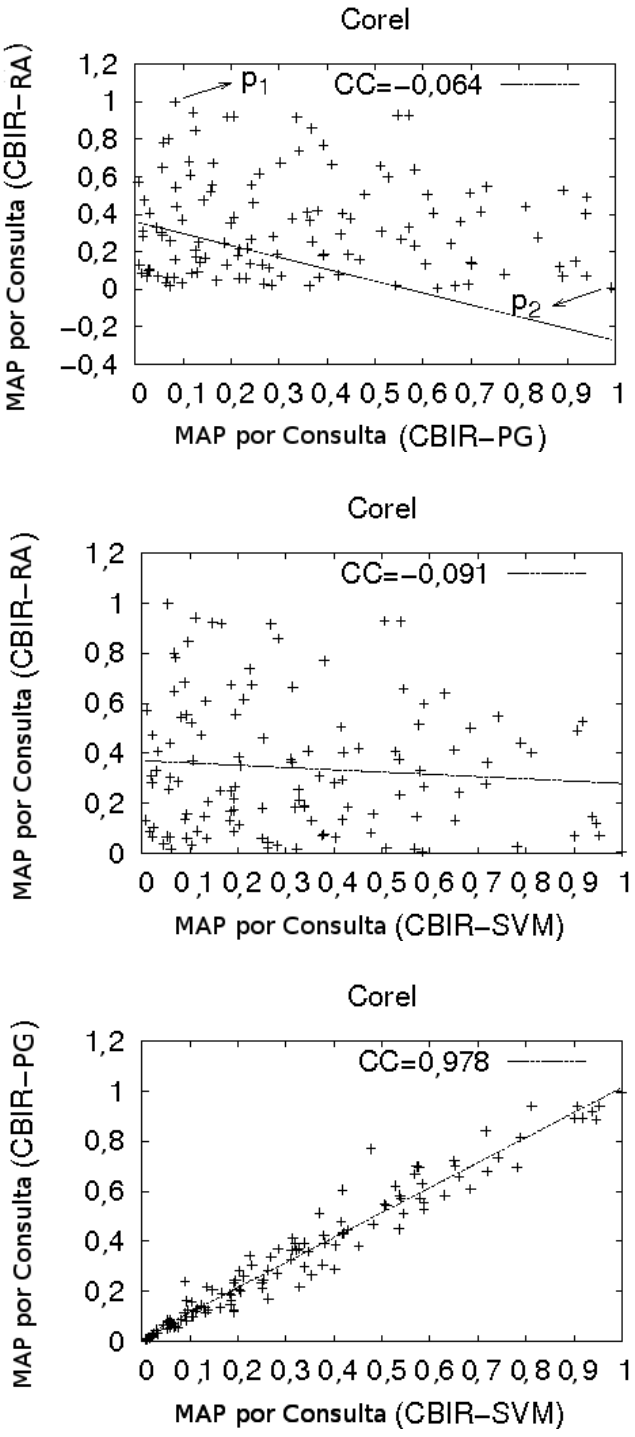


Figura 4.3: Cada gráfico mostra os valores MAP obtidos para cada consulta entre duas técnicas diferentes. No topo: CBIR-RA x CBIR-PG; no meio CBIR-RA x CBIR-SVM; em baixo CBIR-PG x CBIR-SVM. O coeficiente de correlação (CC) é mostrado na parte superior de cada gráfico.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Nos dias atuais, o avanço da tecnologia de aquisição e armazenamento de imagem, bem como a utilização da internet fez com que bases de dados de imagens tivessem um enorme crescimento em seu volume. Estas imagens digitais são geradas todos os dias e, se analisadas, podem revelar informações úteis aos seus usuários.

A análise de grande quantidade de imagens torna-se impraticável de ser realizada por seres humanos e, por esse motivo, há necessidade de se criar ferramentas eficazes para procurar e encontrar padrões presentes nessas imagens armazenadas. Neste trabalho de mestrado foram investigadas técnicas de aprendizagem em problemas de recuperação e classificação de imagens.

Uma técnica muito utilizada em diversas áreas, a programação genética, foi alvo dos estudos desenvolvidos neste trabalho. Esta técnica, que está inserida no contexto de inteligência artificial, busca soluções ótimas se baseando na seleção natural de espécies, nas quais os indivíduos mais aptos tendem a se reproduzir e evoluir nas gerações futuras. A seleção natural é realizada por uma função de adequação que verifica o quão apto o indivíduo está para resolver um determinado problema. As operações de mutação e *crossover* são usadas para evoluir os indivíduos e a reprodução faz com que os melhores indivíduos passem para próxima geração.

Nos experimentos de classificação de imagens, uma técnica de classificação baseada em PG (PG+KNN) foi comparada com técnicas baseadas em *Support Vector Machines* (SVM), *Bagging* (BAGG), *k-nearest neighbor* (kNN), entre outras. Nos

experimentos realizados pode-se notar uma instabilidade do sistema de classificação de imagens PG+KNN. Apesar da instabilidade, o sistema proposto obteve bom desempenho, comparável a outros classificadores da literatura nas bases de imagens (FreeFotos e Borboletas) utilizadas nos experimentos.

Já nos experimentos de recuperação de imagens foram avaliados três diferentes técnicas de aprendizagem: CBIR-PG (baseado em Programação Genética), CBIR-SVM (baseado em *Support Vector Machines*) e CBIR-RA (baseado em Regras de Associação). Os experimentos mostraram que as técnicas de aprendizagem usadas para combinar múltiplos descritores obtiveram melhores resultados quando comparados ao melhor descritor (BIC). O desempenho das técnicas CBIR-PG e CBIR-RA foram similares, mas o desempenho do CBIR-RA obteve melhores resultados nas primeiras posições da lista ordenada que o CBIR-PG. Ambos foram melhores que CBIR-SVM considerando todas as métricas. Uma análise mais detalhada mostrou que há uma correlação entre a qualidade do resultado do CBIR-RA e as outras técnicas, indicando uma possível combinação entre as técnicas para obter melhores desempenhos na recuperação.

As principais contribuições desse trabalho de mestrado foram:

- Adaptação do modelo de classificação de documentos textuais, baseado em Programação Genética, proposto por Zhang et al. [77] visando à classificação de imagens;
- Modelagem da técnica de ordenação de documentos textuais, baseado em regras de associação, para imagens proposto por Veloso et al. [72];
- Comparação da Programação Genética em relação a outras técnicas de aprendizagem em problemas de classificação e recuperação de imagens;
- Implementação parcial de um sistema de classificação de imagens usando programação genética.

5.2 Extensões

Existem diversos tópicos que podem ser aprimorados para melhorar os resultados de eficácia das técnicas de aprendizagem para classificação e recuperação de imagens. As principais extensões desse trabalho podem ser divididas em três tópicos principais:

Comum, pois ambas as tarefas (Classificação e Recuperação) podem se beneficiar; Classificação de Imagens e; Recuperação de Imagens;

5.2.1 Comum

Descritores de Imagens

Realizar a substituição/adição de descritores nas tarefas de classificação e recuperação de imagens com o objetivo de melhorar a eficácia dos métodos.

Avaliação do Espaço Paramétrico da Programação Genética

O sucesso no uso de programação genética em uma dada aplicação depende da escolha de valores apropriados para diversos parâmetros, como número de indivíduos de uma população, quantidade de gerações, porcentagem de crossover, mutação e reprodução. Isto abre espaço para investigação de metodologia para avaliação de parâmetros da Programação Genética em tarefas de classificação e recuperação de imagens.

5.2.2 Classificação de Imagens

Votação Ponderada

No sistema de classificação implementado existe uma votação marjoritária, usando o resultado de classificadores kNN para decidir a classe predita de cada uma das imagens. Uma alternativa seria substituir o esquema de votação atual por uma votação ponderada segundo a qual a densidade de probabilidade (distribuição *a priori*) das classes no conjunto de treinamento fosse considerada na classificação final do sistema.

Biblioteca de Programação Genética

A biblioteca de programação genética utilizada no sistema de classificação foi a Java Genetic Algorithms and Genetic Programming Package (JGAP) [52], mas existem outras alternativas que poderiam ser verificada como a lil-gp [82]. A mudança da biblioteca poderá confirmar os resultados obtidos nos experimentos realizados neste trabalho ou até mesmo melhorá-los.

Substituição do Classificador

O arcabouço de classificação de Zhang et al. [76] utiliza classificadores kNN combinados com indivíduos da Programação Genética para se determinar a classe de uma imagem no conjunto de teste. Outras alternativas que podem ser investigadas estão relacionadas à substituição dos classificadores kNN por outros, tais como *Optimum-path Forest* (OPF) [55] e *Support Vector Machine* (SVM) [9].

Combinação de Classificadores

Investigar a combinação de diferentes classificadores pode ser uma outra alternativa para melhorar a eficácia do sistema de classificação.

5.2.3 Recuperação de Imagens

Combinação de Técnicas de Aprendizagem

Nos resultados obtidos nos experimentos de recuperação de imagens, verificou-se a possibilidade de obter melhores resultados de eficácia se as técnicas de aprendizagem fossem combinadas. O objetivo é explorar o fato de que os resultados obtidos entre as técnicas de aprendizagem que deram bons resultados (PG e RA) não estão correlacionados.

Extensão para diferentes aplicações

Nos experimentos realizados foram utilizadas bases de imagens heterogêneas, não associadas a alguma aplicação específica. Uma possível extensão dessa pesquisa seria aplicar as técnicas de aprendizagem em outras áreas do conhecimento como Agricultura e Medicina.

Referências Bibliográficas

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[3] H. M. Almeida, M. Gonçalves, M. Cristo, and P. Calado. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 399–406, 2007.

[4] F. A. Andaló and R. da S. Torres. Descritores de forma baseados em tensor scale. Master's thesis, Instituto de Computação, Unicamp, Campinas, SP, Brazil, 2007.

[5] M. L. Antonie, O. R. Zaiane, and A. Coman. Application of Data Mining Tecniques for Medical Image Classification. In *2nd Int. Workshop on Multimedia Data Mining MDM/KDD2001*, 2001.

[6] A. V. et al. Araújo. Aplicando Mineração de Imagem para Auxiliar na Determinação da Idade Gestacional em Recém-Nascidos. *XI Congresso Brasileiro de Informática em Saúde, Florianópolis, Brazil*, 2006.

[7] X. Bai and X. Qian. Medical image classification based on fuzzy support vector machines. In *ICICTA '08: Proceedings of the 2008 International Conference on*

Intelligent Computation Technology and Automation, pages 145–149, Washington, DC, USA, 2008. IEEE Computer Society.

[8] B. Bhanu and Y. Lin. Object Detection in Multi-Modal Images Using Genetic Programming. *Applied Soft Computing*, 2004.

[9] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152. Springer, 1992.

[10] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. Image understanding for iris biometrics: A survey. *Computer Vision and Image Understanding*, 110(2):281–307, 2008.

[11] Y. Cao, X. Jun, T. Liu, H. Li, Y. Huang, and H. Hon. Adapting ranking svm to document retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, 2006.

[12] A. Çarkacioglu and F. Yarman-Vural. Sasi: a generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615–2633, 2003.

[13] P. Cheng, B. Chien, and W. Yang. Medical image classification by supervised machine learning. In *AIC'06: Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications*, pages 106–110, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[14] R. G. Congalton and K. Green. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices.* Lewis Publishersr, Washington, DC, 1977.

[15] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, EC-14(3):326–334, 1965.

[16] R. da S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[17] R. da S. Torres, A. X. Falcão, M. A. Goncalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009.

[18] Department of Computer Science in University of Regina. Confusion matrix. Disponível em `http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/%confusion_matrix.html`. Acessado em 28/01/2010.

[19] W. Fan, E. A. Fox, P. Pathak, and H. Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search: Research articles. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004.

[20] W. Fan, M. D. Gordon, and P. Pathak. Genetic programming-based discovery of ranking functions for effective web search. *Journal of Management Information Systems*, 21(4):37–56, 2005.

[21] Weiguo Fan, Edward A. Fox, Praveen Pathak, and Harris Wu. The effects of fitness functions on genetic programming-based ranking discovery for web search: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 55(7):628–636, 2004.

[22] F. A. Faria, J. P. Papa, R. da S. Torres, and A. X. Falcão. Multimodal pattern recognition through particle swarm optimization. In *17th International Conference on Systems, Signals and Image Processing*, 2010.

[23] F. A. Faria, A. Veloso, E. Valle, R. Torres, Gonçalves, and W. Meira. Learning to rank for content-based image retrieval. In *ACM International Conference on Multimedia Information Retrieval (MIR 2010)*, 2010.

[24] U. Fayyad and K. Irani. Multi interval discretization of continuous-valued attributes for classification learning. In *13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[25] C. E. Fernandes and M. T. P. Vieira. Classificação de imagens de sensoriamento remoto com área desmatada. Master's thesis, Manaus, AM, Brazil, 2008.

[26] C. D. Ferreira, R. da S. Torres, M. A. Goncalves, and W. Fan. Image Retrieval with Relevance Feedback based on Genetic Programming. In *XXIII Simposio Brasileiro de Banco de Dados*, pages 120–134, 2008.

[27] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 1 edition, 2001.

[28] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using distance functions. In *Neural Information Processing Systems*, 2006.

[29] P.H. Gosselin and M. Cord. Color and texture descriptors. *Active learning methods for Interactive Image Retrieval.*, 17(7):1200–1211, 2008.

[30] J. Han, S. J. McKenna, and R. Wang. Learning query-dependent distance metrics for interactive image retrieval. *7th International Conference on Computer Vision Systems (ICVS)*, pages 374–383, 2009.

[31] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA, 2000. ACM.

[32] R. Herbrich, T. Graepel, and K. Obermayer. *Large Margin Rank Boundaries for Ordinal Regression*, pages 115–132. MIT Press, Cambridge, MA, 2000.

[33] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *In Proc. IEEE International Conference on Image Processing (ICIP)*, pages 750–753, 2000.

[34] Y. Hu, M. Li, and N. Yu. Multiple-instance ranking: Learning to rank images for image retrieval. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 1–8, 2008.

[35] C. Huang and Q. Liu. An orientation independent texture descriptor for image retrieval. In *International Conference on Computational Science*, pages 772–776, 2007.

[36] J. Huang, R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[37] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

[38] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[39] L. R. Jorge and A. V. L. Freitas. Variação morfologica dependente da planta hospedeira na borboleta heliconius erato phyllis (lepdoptera:nymphalidae). Master's thesis, Campinas, SP, Brazil, 2009.

[40] V. Kovalev and S. Volmer. Color co-occurence descriptors for querying-by-example. In *Proceedings of the 1998 Conference on MultiMedia Modeling*, pages 32–38, 1998.

[41] J. Koza. *Genetic Programming: On the programming of computers by natural selection*. MIT Press, 1992.

[42] A. Kumar and A. Passi. Comparison and combination of iris matchers for reliable personal authentication. *Pattern Recognition*, 43(3):1016–1026, 2010.

[43] W. B. Langdon. *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, volume 1 of *Genetic Programming*. Kluwer, Boston, 24 apr 1998.

[44] D. Lee and H. Kim. A fast content-based indexing and retrieval technique by the shape information in large image database. *Journal of Systems and Software*, 56(2):165–182, 2001.

[45] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[46] Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *Learning to Rank Workshop in conjuntion with SIGIR Conference on Research and Development on Information Retrieval*, 2007.

[47] T. Lu and C. Chang. Color image retrieval technique based on color features and image bitmap. *Inf. Processing and Management*, 43(2):461–472, 2007.

[48] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive content-based image retrieval using relevance feedback. *Computer Vision and Image Understanding*, 88(2):55–75, 2002.

[49] F. Mahmoudi, J. Shanbehzadeh, A. Eftekhari-Moghadam, and H. Soltanian-Zadeh. Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern Recognition*, 36(8):1725–1736, 2003.

[50] B. Manjunath, J. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[51] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[52] K. Meffert. Jgap - java genetic algorithms and genetic programming package. Disponível em `http://jgap.sf.net`. Acessado em 05/02/2010.

[53] Laboratory of Information Systems of the Institute of Computing at the University of Campinas. Bio-core - biodiversity and computing research. Disponível em `http://www.lis.ic.unicamp.br/projects/biocore/`. Acessado em 12/02/2010.

[54] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[55] J. P. Papa, A. A. Spadotto, A. X. Falcao, and J. C. Pereira. Optimum path forest classifier applied to laryngeal pathology detection. In *Systems, Signals*

and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, pages 249–252, 2008.

[56] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[57] M. X. Ribeiro and M. T. P. Vieira. Mineração de dados em múltiplas tabelas fato de um data warehouse. Master's thesis, São Carlos, SP, Brazil, 2004.

[58] A. Rocha and S. Goldenstein. Randomização progressiva para esteganálise. Master's thesis, Campinas, SP, Brazil, 2006.

[59] J. A. Santos, F. A. Faria, R. T. Calumby, and R. da S. Torres. A genetic programming aprroach for coffee crop recognition. In *International Geoscience and Remote Sensing Symposium*, 2010.

[60] J. A. Santos and R. da S. Torres. Reconhecimento semi-automaático e vetorização de regiões em imagens de sensoriamento remoto. Master's thesis, Campinas, SP, Brazil, 2009.

[61] A. Savasere, E. Omiecinski, and S. B. Navathe. An efficient algorithm for mining association rules in large databases. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pages 432–444, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[62] H. Shao, J. W. Zhang, W. C. Cui, and H. Zhao. Automatic Feature Weight Assignment based on Genetic Algorithm for Image Retrieval. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, pages 731–735, 2003.

[63] R. Showengerdt. *Techniques for Image Processing and Classification in Remote Sensing*. Academic Press, New York, 1983.

[64] Thiago V. Spina, Javier A. Montoya-Zegarra, Fábio Andrijauskas, Fábio A. Faria, Carlos E. A. Zampieri, Sheila M. Pinto-Cáceres, Tiago J. de Carvalho, and Alexandre X. Falc ao. A comparative study among pattern classifiers in interactive image segmentation. In *XXII SIBGRAPI*, pages 268–275, October 2009.

[65] R. Stehling, M. Nascimento, and A. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109, 2002.

[66] M. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[67] C. T. N. Suzuki. Segmentação de Imagem de Parasitos Intestinais do Homem. Master's thesis, Campinas, SP, Brazil, 2007.

[68] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[69] B. Tao and B. Dickinson. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327–342, 2000.

[70] A. Tofilski and D. Wing. A program for numerical description of insect wings. *Journal of Insect Science*, 4(17):1–5, 2004.

[71] M. Unser. Sum and difference histograms for texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):118–125, 1986.

[72] A. Veloso, H. M. Almeida, M. Gonçalves, and W. Meira. Learning to rank at query-time using association rules. In *31th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 267–274, 2008.

[73] A. Williams and P. Yoon. Content-based image retrieval using joint correlograms. *Multimedia Tools and Applications*, 34(2):239–248, 2007.

[74] P. Wu, B. S. Manjunanth, S. D. Newsam, and H. D. Shin. A texture descriptor for image retrieval and browsing. In *CBAIVL '99: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, page 3, Washington, DC, USA, 1999. IEEE Computer Society.

[75] J. Zegarra, N. Leite, and R. da S. Torres. Wavelet-based feature extraction for fingerprint image retrieval. *Journal of Computational and Applied Mathematics*, 2008.

[76] B. Zhang, Y. Chen, W. Fan, E. A. Fox, M. Gonçalves, M. Cristo, and P. Calado. Intelligent gp fusion from multiple sources for text classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 477–484, New York, NY, USA, 2005. ACM.

[77] B. Zhang, M. A. Gonçalves, W. Fan, Y. Chen, E. A. Fox, P. Calado, and M. Cristo. Combining structural and citation-based evidence for text classification. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 162–163, New York, NY, USA, 2004. ACM.

[78] J. Zhang, W. Hsu, and M. Lee. Image mining: Issues, frameworks and techniques, 2001.

[79] L. Zhang, Lin F., and Zhang B. Support vector machine learning for image retrieval. In *International Conference on Image Processing, 2001.*, pages 721–724 vol.2, 2001.

[80] Q. Zhang, K. Huang, and H. Yan. Fingerprint classification based on extraction and analysis of singularities and pseudoridges. In *VIP '01: Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 83–87, Darlinghurst, Australia, Australia, 2001. Australian Computer Society, Inc.

[81] J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 1998.

[82] D. Zongker, B. Puch, and B. Rand. lil-gp genetic programming system. Disponível em `http://garage.cse.msu.edu/software/lil-gp`. Acessado em 05/02/2010.