

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO ENGENHARIA DE COMPUTAÇÃO (COENC-CP)

GUSTAVO VIAIS DE BRITO SILVA

**VETORES DE CARACTERÍSTICAS DE ALTO NÍVEL VISANDO A
PERCEPÇÃO DO USUÁRIO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2016

GUSTAVO VIAIS DE BRITO SILVA

**VETORES DE CARACTERÍSTICAS DE ALTO NÍVEL VISANDO A
PERCEPÇÃO DO USUÁRIO**

Proposta apresentada à disciplina de Trabalho de Conclusão de Curso, do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Pedro Henrique Bugatti

CORNÉLIO PROCÓPIO

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Nome da Diretoria
Nome da Coordenação
Nome do Curso



TERMO DE APROVAÇÃO

Vetores de Características de Alto Nível Visando a Percepção do Usuário

por

Gustavo Viais de Brito Silva

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de “Engenheiro em Computação” e aprovado em sua forma final pelo Programa de Graduação em Engenharia da Computação da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 23/06/2016

Prof. Dr. Pedro Henrique Bugatti

Prof. Titulação. Nome Professor Orientador

Profa. Dra. Priscila Tiemi Maeda Saito

Prof. Titulação. Nome Professor membro da banca

Prof. Dr. Claiton de Oliveira

Prof. Titulação. Nome professor membro da banca

AGRADECIMENTOS

Primeiramente a Deus pela saúde, possibilidades de realizar sonhos pessoais e inúmeras bênçãos derramadas sobre mim e toda minha família. Por seu amor e misericórdia.

Agradeço ao meu orientador, prof.^o Dr.^o Pedro Henrique Bugatti por me proporcionar um imenso aprendizado, pela dedicação e competência. Obrigado por ter me dado a oportunidade de ser seu aluno.

À Universidade Tecnológica Federal do Paraná (UTFPR), à todos os professores, funcionários e colegas que contribuíram de alguma forma para a realização deste trabalho.

À minha noiva Ana Flávia de Arruda Piovesani por sempre estar ao meu lado, sobretudo nos momentos mais difíceis e decisivos. Por viver comigo muitos dos meus sonhos e me ajudar a alcançá-los.

À minha mãe Márcia Aparecida Viais de Brito pelo amor incondicional, apoio e incentivo nos momentos de dificuldade. Pela intensidade com que sempre confiou em mim e pela garra que sempre superou as dificuldades. Você é meu exemplo!

Ao meu pai Gelson de Brito Silva pelo seu amor, cuidado e apoio fundamental. Por entender minhas decisões, me aconselhar e ajudar muito além das possibilidades.

À toda minha família, meu irmão Guilherme, minha avó Neide e, principalmente, a meu avô que nos deixou um grande exemplo de ser humano.

À todos amigos da UTFPR pelos anos de convivência, companheirismo e conversas sempre muito produtivas. Guardarei cada uma de vocês no meu coração como parte fundamental desta conquista.

RESUMO

SILVA, Gustavo Viais de Brito. VETORES DE CARACTERÍSTICAS DE ALTO NÍVEL VISANDO A PERCEPÇÃO DO USUÁRIO. 37 f. Trabalho de conclusão de curso – Coordenação ENGENHARIA DE COMPUTAÇÃO (COENC-CP), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

O aumento na quantidade de imagens médicas digitais armazenadas nos últimos tempos tem criado grande necessidade de algoritmos para recuperação de forma mais fácil, rápida e eficiente. Todavia, um dos problemas para o uso desta técnica trata-se de conseguir um nível de similaridade baseado na percepção do usuário, pois é necessário caracterizar e quantificar da melhor maneira essa similaridade. Para isso, é necessário um trabalho de integração entre três elementos (extratores de características, função de distância e parâmetro perceptual) que resultem na criação de operadores de similaridade, os quais podem ser utilizados para aproximar o sistema computacional ao usuário final. Portanto, esse trabalho visa criar um sistema inteligente de aprendizado que seja capaz de recuperar imagens de acordo com a percepção do usuário.

Palavras-chave: CBIR, Funções de Distância, Extratores de Características, Percepção Visual

ABSTRACT

SILVA, Gustavo Viais de Brito. HIGH-LEVEL FEATURES VECTORS ACCORDING TO THE USER PERCEPTION. 37 f. Trabalho de conclusão de curso – Coordenação ENGENHARIA DE COMPUTAÇÃO (COENC-CP), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

The increase in the amount of digital medical images stored recently has created the necessity of algorithms for recovering data in an easier, faster and efficient way. However, one of the problems to use this technique is finding a similarity level based on the user perception, because it is necessary to characterize and quantify this similarity. Therefore, an integration between three elements (characteristic extractor, distance functions and perceptual parameter) which results in the creation of similarity operators is needed. It could be used to approximate the computational system to the final user. The main goal of this research is developing an intelligent learning system able to recovery images according to the user perception.

Keywords: CBIR, Distance Functions, Feature Extraction, Visual Perception

LISTA DE FIGURAS

FIGURA 1	– Busca por abrangência	10
FIGURA 2	– Busca pelos vizinhos mais próximos	11
FIGURA 3	– Exemplo do histograma de cores de uma imagem	12
FIGURA 4	– Exemplo cálculo da Matriz de Co-ocorrência	13
FIGURA 5	– Formas de uma imagem. (a) Imagem Original. (b) Imagem achatada dentro de uma forma circular.	15
FIGURA 6	– Abrangência das funções de distância	17
FIGURA 7	– Camadas do MLP	19
FIGURA 8	– Fluxo do método proposto	24
FIGURA 9	– Exemplo de cálculo do vetor de distâncias	26
FIGURA 10	– Cálculo do APF	27
FIGURA 11	– Exemplos de imagens de cada classe pertencentes ao conjunto de teste da base de imagens de ROIs de pulmão oriundas de exames de tomografia computadorizada <i>CTROIsPulmão</i>	29

SUMÁRIO

1	INTRODUÇÃO	7
1.1	PROBLEMA	8
1.2	OBJETIVOS	8
1.2.1	Objetivo Geral	8
1.2.2	Objetivos Específicos	8
1.3	JUSTIFICATIVA	9
1.4	ORGANIZAÇÃO DO TEXTO	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	CONSULTA POR SIMILARIDADE	10
2.2	EXTRAÇÃO DE CARACTERÍSTICAS	11
2.2.1	Características baseadas em Histograma de Cores	11
2.2.2	Características baseadas em Textura	12
2.2.2.1	Matrizes de co-ocorrência	13
2.2.2.2	Descritores de Haralick	13
2.2.3	Características baseadas em Forma	14
2.2.3.1	Momentos de Zernike	15
2.3	FUNÇÕES DE DISTÂNCIA	16
2.4	CLASSIFICAÇÃO	17
2.4.1	WEKA	17
2.4.2	Classificadores	17
2.4.2.1	Naive Bayes	17
2.4.2.2	Multilayer Perceptron	18
2.4.2.3	J48	20
2.4.2.4	AdaBoost	21
2.5	FATOR DE PERCEPÇÃO	23
3	MÉTODO PROPOSTO	24
3.1	INTRODUÇÃO	24
3.2	MÉTODO PROPOSTO	24
3.3	EXPERIMENTOS	28
3.3.1	Materiais e Métodos	28
3.3.2	Descrição da base de imagens	29
3.3.3	Resultados	29
4	CONCLUSÕES	34
4.1	CONSIDERAÇÕES FINAIS	34
4.2	RESULTADOS OBTIDOS	34
	REFERÊNCIAS	35

1 INTRODUÇÃO

De acordo com Ponciano-Silva et al. (2009), o atual avanço dos equipamentos tecnológicos, o aumento do volume de imagens capturadas e o barateamento relativo das tecnologias de coleta de dados e imagens, faz com que a maioria dos hospitais, clínicas, laboratórios e até alguns consultórios médicos utilizem, cotidianamente, ferramentas computacionais que gerenciem, armazenem e organizem as informações clínicas dos pacientes.

Um modo de permitir que a interação entre o médico e as ferramentas computacionais ocorra é por meio da busca de imagens ou casos similares. Desta forma, esse profissional poderia obter tais informações, por exemplo, por meio de uma busca do tipo “recuperar as imagens geradas por meio do Raio-X de tórax, que são similares àquela imagem sendo analisada”. Nesse caso, o cruzamento das informações independe de chaves de busca tradicionais, baseadas em dados numéricos ou textuais associados a uma imagem (como nome ou data do exame). Esse novo tipo de busca leva em consideração o conteúdo intrínseco das imagens, fazendo com que as que possuam características visuais próximas sejam consideradas como semelhantes. Esse conceito é denominado como *CBIR (Content-Based Image Retrieval)* (DATTA et al., 2008).

De acordo com Zakariya et al. (2010), a área médica pode se beneficiar muito das Aplicações *CBIR* devido à enorme quantidade de imagens geradas por dia em hospitais e centros médicos. A Recuperação de Imagens Baseada em Conteúdo emprega algoritmos de processamento de imagens para extrair relevantes características das mesmas, organizando-as em vetores de características. Os vetores de características são empregados no lugar das imagens, permitindo uma rápida indexação e uma recuperação eficiente. Quantificar características visuais e intrínsecas das imagens, como cor, forma e textura, pode geralmente causar uma alta-dimensionalidade nos vetores. Embora várias técnicas como *clustering*, busca pelo vizinho mais próximo e indexação já tenham sido estudadas, em alto espaço dimensional essas técnicas algorítmicas podem ser insuficientes e falharem (do ponto de vista de desempenho).

O uso de um grande número de características para representar imagens é um problema, porque os métodos de indexação, recuperação e comparação entre imagens tornam-se

menos precisos e mais demorados. O uso de uma grande quantidade de características faz sistemas de busca por conteúdo enfrentarem o problema da *dimensionality curse* (maldição da dimensionalidade). Isto ocorre quando se utiliza um grande número de características (e consequentemente, o uso de uma base de dados com grande dimensão) e quando cada recurso é realizado separadamente para diminuir o custo do processo. Assim, um dos principais desafios em um sistema *CBIR* é determinar um subconjunto mínimo de características que melhor representam as imagens de acordo com cada intenção de um especialista (BUGATTI et al., 2008).

Portanto, esse trabalho visa adstringir elementos das percepções dos usuários (ordem de seleção) e suas respectivas funções de distância, com características de similaridade entre as imagens, de modo que os resultados possam ser satisfatórios para o usuário.

1.1 PROBLEMA

Um dos problemas para o uso das técnicas de recuperação de imagens atuais trata-se de conseguir um nível de similaridade baseado na percepção do usuário, pois é necessário caracterizar e quantificar da melhor maneira essa similaridade. O autor Zakariya et al. (2010) cita que as técnicas de comparação de imagens não visam distinguir características de acordo com a percepção do usuário, baseando-se apenas em extratores de características (cor, textura, forma) e utilizam funções de distâncias conhecidas, sem se preocuparem em escolher uma função mais adequada para cada caso. Esse fato culmina, certas vezes, em resultados não esperados.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O presente trabalho visa criar um sistema inteligente de aprendizado que seja capaz de recuperar imagens de acordo com a percepção do usuário. Para tanto, é necessária a criação de vetores de características de alto nível mais próximos da semântica de consulta desejada pelo usuário.

1.2.2 OBJETIVOS ESPECÍFICOS

1. Estudo de métricas e funções de distância
2. Desenvolvimento de consultas por similaridade tradicionais
3. Análise de integração entre par de distância e extrator de características

4. Desenvolvimento de técnicas de aproximação do usuário às consultas por similaridade
5. Análise de custo e desempenho dos métodos em questão

1.3 JUSTIFICATIVA

Com o intuito de diminuir ou sanar os presentes problemas, o foco desse trabalho é, além de estabelecer os extratores e suas respectivas funções de distâncias para cada classe de imagem, filtrar os resultados de acordo, também, com a percepção do usuário por meio da ordem de seleção de imagens realizada pelo mesmo. Para isso, será necessário o uso de alguns *classificadores* (os quais têm por objetivo criar um modelo que prevê a classe de uma determinada imagem baseando-se apenas em algumas de suas características (BANDA; ANGRYK, 2010)) para a base de dados em questão.

1.4 ORGANIZAÇÃO DO TEXTO

O texto deste trabalho está dividido em quatro seções. Na primeira seção são apresentados a introdução do trabalho, problema, objetivos (gerais e específicos) e a justificativa dos mesmos. A segunda seção diz respeito à fundamentação teórica do trabalho, na qual são apresentadas as técnicas da literatura necessárias para a realização do trabalho. Entre elas estão as Consultas por Similaridade, Extração de Características, Funções de Distância, Classificadores e Fator de Percepção. Na terceira seção é apresentado o método proposto, os experimentos realizados, resultados e a descrição da base de imagens utilizada. A quarta e última seção é composta pelas conclusões do presente trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CONSULTA POR SIMILARIDADE

Os Sistemas Gerenciadores de Banco de Dados (SGBDs) atuais manipulam dados que possuem relação de ordem entre seus elementos (números, textos, etc). Portanto, identificar e organizar esses dados de acordo com uma relação de ordem ou igualdade é, na maioria das vezes, uma tarefa simples. Contudo, muitas vezes surge a necessidade de manipular dados complexos, os quais não possuem uma relação de ordem (como por exemplo, a comparação entre duas imagens). Para satisfazer tal necessidade, surge o conceito de Consulta por Similaridade. As consultas por similaridade baseiam-se na ideia de comparar todos os objetos de um conjunto com um objeto escolhido, retornando os objetos mais parecidos ou mais diferentes do objeto selecionado. Isso cria um critério de similaridade entre esses elementos (ZAKARIYA et al., 2010).

São vários os tipos de consultas por similaridade, todavia os mais comuns são as buscas por abrangência e busca pelos vizinhos mais próximos. A primeira consiste na busca de objetos que tenham até um nível de similaridade com o objeto selecionado, de acordo com a Figura 1. A segunda fundamenta-se em retornar k objetos mais similares ao objeto de consulta (DATTA et al., 2008), de acordo com a Figura 2.

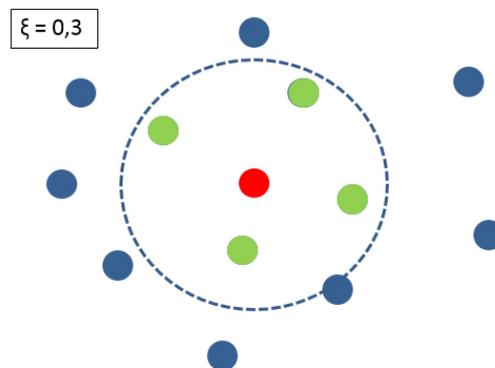


Figura 1 – Busca por abrangência

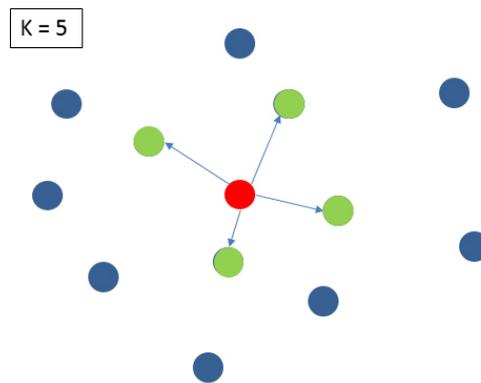


Figura 2 – Busca pelos vizinhos mais próximos

2.2 EXTRAÇÃO DE CARACTERÍSTICAS

A Recuperação de Imagens Baseada em Conteúdo envolve cor, textura, forma e outras características visuais de imagens de baixo nível. Os parâmetros da imagem que são baseados em características de baixo nível podem ser obtidos diretamente a partir da imagem (XIA-OHONG; JINHUA, 2008).

2.2.1 CARACTERÍSTICAS BASEADAS EM HISTOGRAMA DE CORES

As características de cor são as mais intuitivas e óbvias à imagem, e geralmente adotam histogramas para descrevê-la (FLICKNER et al., 1995).

De acordo com Zhang et al. (2009), o histograma de cores é utilizado como uma representação efetiva para o conteúdo de uma imagem se a cor padrão for única quando comparada com as demais imagens do conjunto de dados.

Segundo Kotoulas e Andreadis (2003), a extração de características baseadas em histograma de cores é provavelmente a mais popular por ser um método simples de ser utilizado.

A recuperação de imagens baseadas em histograma de cores possui diversas vantagens, porém há algumas limitações:

- É robusta, uma vez que o histograma de cores é invariante à rotações da imagem no eixo de rotação e a pequenas variações no grau quando rotacionadas em sentido contrário.
- É rápida. O cálculo do histograma possui complexidade $O(M^2)$ para uma imagem de $M \times M$ pixels, enquanto a comparação de histograma possui complexidade $O(n)$, na qual n é o número de barras do histograma para cada cor utilizada.
- Requer baixa capacidade de armazenamento, pois o tamanho do histograma de cores é muito

menor que o tamanho da imagem original.

- Nenhuma informação espacial é incluída, portanto duas imagens totalmente diferentes podem ter histograma similares.
- Luzes externas podem alterar o histograma de qualquer imagem (RASHEED et al., 2008).

O histograma de uma imagem representa a distribuição dos níveis de cinza pelo número de pixels da mesma (STRICKER; ORENGO, 1995). A Figura 3 mostra um exemplo do histograma de cores de uma imagem. No Algoritmo 1 é demonstrado o pseudocódigo do cálculo do histograma de cores.

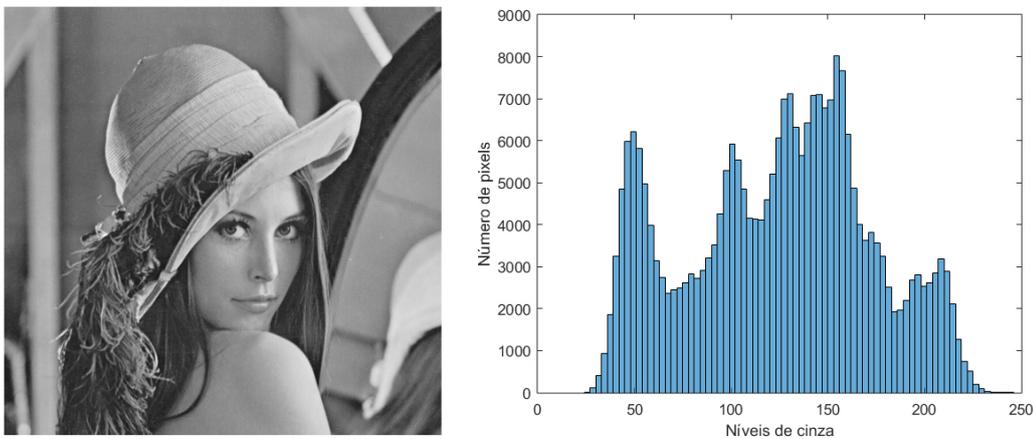


Figura 3 – Exemplo do histograma de cores de uma imagem

Algorithm 1 Cálculo do histograma (H) de uma imagem $f(x,y)$ de dimensão $M \times N$ pixels com L níveis de cinza no intervalo $[0, L_{max}]$

- setar todos os elementos do vetor com valor zero

for i de 0 até L_{max} **do**

 | $H[i] = 0;$

end

- efetuar o cálculo da distribuição dos níveis de cinza para cada pixel da imagem

for x de 0 até $M - 1$ **do**

 | **for** y de 0 até $N - 1$ **do**

 | $H[f(x,y)] = H[f(x,y)] + 1;$

 | **end**

end

2.2.2 CARACTERÍSTICAS BASEADAS EM TEXTURA

De acordo com Luo et al. (2009), a extração de textura é um processo que propõe quantificar os padrões de textura dentro de uma vizinhança específica de tamanho $N \times N$ pixels

em volta de um pixel de interesse. Basicamente, existem quatro categorias de análise de textura denominadas estrutural, estatística, baseada em modelo e baseada em transformação.

Um dos principais métodos estatísticos baseia-se nas Matrizes de Co-ocorrência, propostas por Haralick (1979), as quais caracterizam a inter-relação espacial dos tons de cinza em uma imagem. Os valores dos elementos da matriz representam as frequências relativas com que dois *pixels* vizinhos aparecem nesta imagem (SUGANYA; RAJARAM, 2013).

2.2.2.1 MATRIZES DE CO-OCORRÊNCIA

As matrizes de co-ocorrência podem ser definidas como a co-ocorrência de valores de intensidade em um ângulo específico. Portanto, para uma imagem com N níveis de intensidade, a matriz de co-ocorrência deve possuir tamanho N x N para um deslocamento específico em uma direção específica. Isto explica como os valores de intensidade são distribuídos em uma direção e um ângulo particular. Matrizes de co-ocorrência podem ser utilizadas para calcular diversos parâmetros como correlação, contraste e outros (KHAN et al., 2011).

A Figura 4 ilustra a matriz de uma imagem de tamanho 6x6 e três níveis de intensidade (0, 1 e 2). A matriz de co-ocorrência formada, considerando ângulo zero, será de tamanho 3x3.

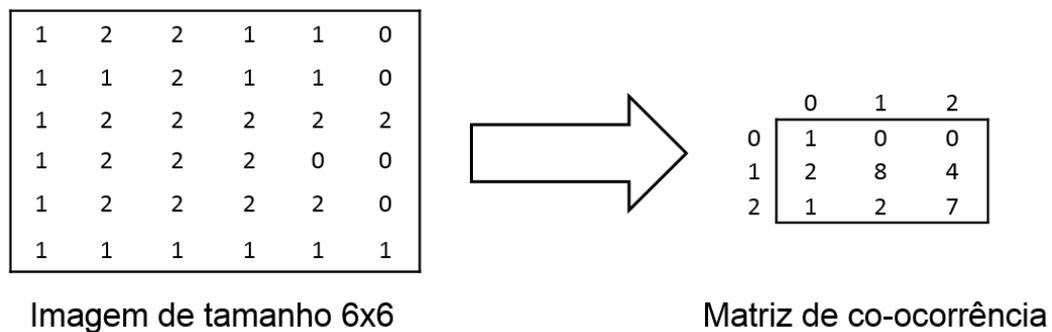


Figura 4 – Exemplo cálculo da Matriz de Co-ocorrência

2.2.2.2 DESCRITORES DE HARALICK

De acordo com Ruberto e Fodde (2014), os descritores de Haralick são baseados em estatísticas de segunda ordem, representadas por pares de pixels, que são calculadas por meio das Matrizes de co-ocorrência. Estes descritores tem como objetivo descrever algumas propriedades contidas na textura de uma imagem. Em Haralick et al. (1973), são propostas quatorze medidas estatísticas, entre elas energia (segundo momento angular), entropia, contraste e homogeneidade.

A energia, definida pela equação 8, representa a uniformidade da textura. O momento angular recebe valor zero quando a imagem é constante (texturas ásperas).

$$energia = \sum_{i=0}^{H_g} \sum_{j=0}^{H_g} c(i, j)^2 \quad (8)$$

A entropia, definida pela equação 9, expressa a desordem de valores na textura (imagem não uniforme).

$$entropia = - \sum_{i=0}^{H_g} \sum_{j=0}^{H_g} c(i, j) \log(c(i, j)) \quad (9)$$

O contraste é definido pela média ponderada de todas as diagonais paralelas à diagonal principal. O valor do contraste é zero se a imagem é constante. A equação 10 mostra o cálculo do contraste.

$$contraste = \sum_{i=0}^{H_g} \sum_{j=0}^{H_g} (i, j)^2 c(i, j) \quad (10)$$

O valor da homogeneidade é alto quando a imagem possui uma textura com pequenas variações de níveis de cinza entre os pares de pixels. A homogeneidade é calculada por meio da equação 11.

$$homogeneidade = \sum_{i=0}^{H_g} \sum_{j=0}^{H_g} \frac{1}{1 + (i, j)^2} c(i, j) \quad (11)$$

2.2.3 CARACTERÍSTICAS BASEADAS EM FORMA

Em relação as características fundamentadas na forma da imagem, inúmeros métodos foram propostos. Entre eles, os descritores de Fourier propõem duas representações: representação borda base e representação baseada região. A transformada de Fourier pode realizar uma representação entre ambos os métodos. Portanto, utilizando-se uma transformada, é possível adotar os limites, descrevendo a imagem em forma. Resultados de experiências já realizadas demonstraram que o método de combinação usando as duas representações em conjunto, pode levar ao melhor caminho (XIAOHONG; JINHUA, 2008). Outro método proposto são os mo-

mentos de Zernike. Este método é derivado de um conjunto complexo de funções ortogonais conhecido como Polinômios de Zernike e podem ser usados para reconstrução da forma da imagem (BELKASIM et al., 2004).

2.2.3.1 MOMENTOS DE ZERNIKE

De acordo com Osman et al. (2009), os momentos de Zernike (ZM) tem sido extensivamente utilizados como características globais invariantes para reconhecimento de imagens. Os ZM são construídos por meio de um conjunto de complexos polinomiais, os quais formam um conjunto de bases ortogonais definidas por $x^2 + y^2 \leq 1$.

Os descritores de Zernike são obtidos aplicando a transformada do momento de Zernike em uma forma de uma imagem (AMAYEH et al., 2007). Quando há achatamento ou alongamento severos na imagem, a forma de região distorcida dentro da unidade de um círculo é alterada drasticamente. Como exemplo, na Figura 5, perceptualmente, as formas (a) e (b) são homogêneas. Entretanto, devido ao achatamento, a forma (b) ocupa somente metade do espaço contido no círculo. Portanto, em (b) há menos informações de forma disponível do que em (a).

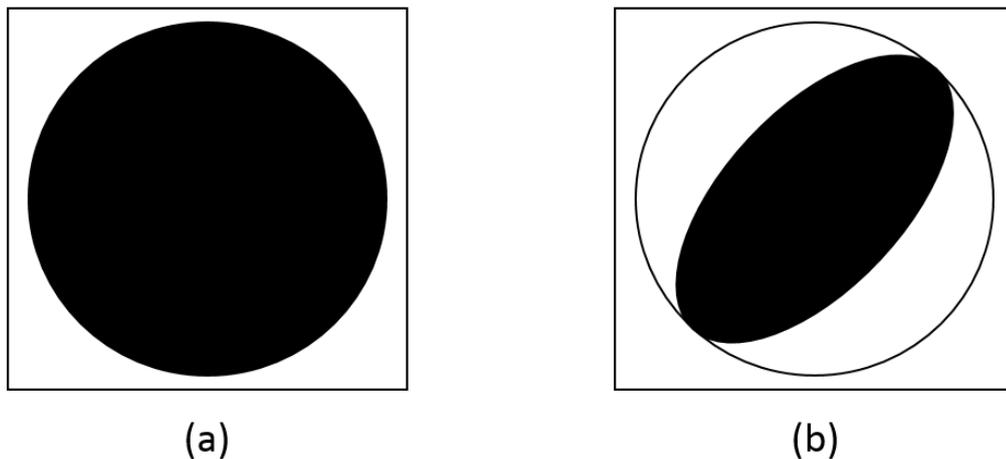


Figura 5 – Formas de uma imagem. (a) Imagem Original. (b) Imagem achatada dentro de uma forma circular.

Segundo Dhir (2010), os momentos de Zernike, com propriedades de achatamento ou alongamento, são eficientes descritores de forma e podem ser utilizados como descritores de forma global em uma imagem, especialmente para grandes bases de dados.

Para uma imagem digital, os momentos de Zernike, de ordem n e repetição m , são dados por:

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) V_{nm}^*(x,y) \quad (12)$$

onde $V_{nm}^*(x,y)$ é o conjugado de um complexo de $V_{nm}(x,y)$, enquanto n é um inteiro (que pode ser +ve ou ve) que descreve a dependência angular ou rotação.

Os valores absolutos dos momentos de Zernike são somente invariantes à rotação. Para a escala de ganho e invariância de translação, a imagem precisa ser normalizada através dos momentos regulares de Zernike. Uma das mais conhecidas propriedades dos momentos de Zernike é que a uma imagem pode ser reconstruída por meio da transformação inversa, dada por:

$$f(x,y) = \sum_{n=0}^{n_{max}} \sum_{m=-n}^n A_{nm} V_{nm}(x,y) \quad (13)$$

na qual n_{max} é a ordem máxima dos momentos de Zernike (considerando uma aplicação particular). As magnitudes dos ZM são invariantes à rotações. Além disso, elas podem também ser invariantes à translações e escalas.

2.3 FUNÇÕES DE DISTÂNCIA

A eficiência e a eficácia da recuperação de dados multimídias podem ser significante-mente afetadas pelo modo que as funções de distância organizam esses dados (BUGATTI et al., 2008). As funções utilizadas nesse artigo são baseadas nas distâncias da família Minkowski, de acordo com a equação 1. Para essa equação, é necessário definir dois vetores de características $X = (x_1, \dots, x_n)$ e $Y = (y_1, \dots, y_n)$. Nesse caso, o vetor X representa a imagem de consulta e o vetor Y representa as imagens da base que serão comparadas com a imagem X . Os parâmetros x_n e y_n contém cada coordenada, num espaço dimensional, dessas imagens.

$$L_p((x_1, \dots, x_n), (y_1, \dots, y_n)) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p} \quad (1)$$

Para derivar funções de distância da equação anterior, basta substituir o valor dese-jado no lugar de p . Para esse trabalho foram utilizadas distâncias $L1$, $L2$ e $L\infty$. A distância $L1$, também conhecida como *Manhattan*, é geralmente utilizada para calcular a similaridade

entre imagens coloridas. Já a distância $L2$ (Euclidiana) é comumente utilizada para calcular a distância entre vetores de características. Para calcular $L\infty$, conhecida como *Infinity*, aplica-se um limite a p fazendo-o tender ao infinito. Isso faz com que o resultado obtido seja o valor máximo entre suas coordenadas. A Figura 6 demonstra a abrangência de cada uma das funções de distâncias.

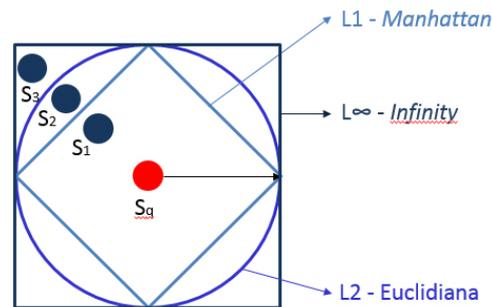


Figura 6 – Abrangência das funções de distância

2.4 CLASSIFICAÇÃO

2.4.1 WEKA

O Weka é um software com uma coleção de algoritmos de aprendizado de máquina para mineração de dados. Os algoritmos podem ser aplicados em um conjunto de dados a partir do próprio código em Java. O Weka também possui ferramentas para o pré-processamento de dados, classificação, regressão, clusterização, associação de regras e visualização. Ele é um software open source (código aberto) e disponibilizado sob a GNU General Public License.

2.4.2 CLASSIFICADORES

Os *classificadores* são algoritmos supervisionados, os quais têm por objetivo descobrir a classe de uma imagem somente através de algumas características da mesma. No presente trabalho, os classificadores utilizados foram *Naive Bayes*, *Multilayer Perceptron*, *J48* e *AdaBoost*.

2.4.2.1 NAIVE BAYES

O algoritmo Naive Bayes é um método de aprendizado supervisionado baseado em classificação por probabilidade condicional, que tem como fundamento assumir que seus atributos são condicionalmente independentes dado um rótulo de classificação (DOMINGOS; PAZZANI, 1997). Ele usa o teorema de *Bayes*, o qual tem como princípio encontrar a probabilidade

de um evento ocorrer dado que a probabilidade de um outro caso em que já tenha ocorrido. Embora o Naive Bayes tenha muitas vantagens, tanto conceitualmente como computacionalmente, sua irrealista independência de atributos gera uma estimação de probabilidade pobre. Para superar a limitação do Naive Bayes, uma maneira natural é representar a dependência dos atributos (JIANG; ZHANG, 2005).

Dado uma instância de teste, o algoritmo de Bayes é questionado à prever sua classe de acordo com a evidência dada pelo conjunto de treinamento (KLAWONN; ANGELOV, 2006). Em Qin (2006) foi definido c como uma variável randômica para uma classe da instância: $x = (x_1, \dots, x_n)$ como um vetor de variáveis denotando os valores dos atributos observados. O erro de classificação esperado pode ser minimizado escolhendo $\operatorname{argmax}_c(P(c = C|x))$, de acordo com o teorema de Bayes:

$$P(H|D) = \frac{(D|H)}{H} \quad (2)$$

Dado uma classe particular $C_k \in C$, na qual C representa o conjunto de classes. Pode então obter-se:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)} \quad (3)$$

Desde que o denominador seja invariante em todas as classes da equação 3, o mesmo pode ser considerado como um parâmetro de normalização. Portanto:

$$P(C_k|x) = P(x|C_k)P(C_k) \quad (4)$$

Supondo que para cada variável x_j em que o resultado é independente do valor de todas as outras variáveis da classe C_k . Neste caso pode ser obtido o classificador Naive Bayes:

$$P(C_k|x) = \prod_{j=1}^n P(x_j|C_k)P(C_k) \quad (5)$$

2.4.2.2 MULTILAYER PERCEPTRON

Nos últimos anos, o Multilayer Perceptron (MLP) tem se tornado um dos algoritmos de rede neurais mais utilizados em diversas áreas de aplicação. Embora consiga obter excelentes resultados na classificação de conjuntos, um dos pontos fracos do MLP é sua lenta velocidade

de classificação (CHIANG; FU, 1992).

De acordo com Guha e Patra (2010), o MLP é composto por um conjunto de unidades sensoriais organizadas em três ou mais camadas. A primeira camada é denominada camada de entrada, a qual não executa nenhuma tarefa computacional. Então há uma ou mais camadas escondidas (intermediárias) e uma camada de saída, todas formadas por nós computacionais. Em um MLP típico, todos os nós da rede de uma camada são conectados com cada nó da camada anterior e da camada seguinte, como mostra a Figura 7. Não há conexões entre os nós na mesma camada, da mesma forma que não há conexões entre nós em camadas não adjacentes. Os nós não computacionais na camada de entrada usam uma função identidade, enquanto os nós computacionais nas camadas intermediárias e de saída usam uma função sigmoide (BERNARD, 1993).

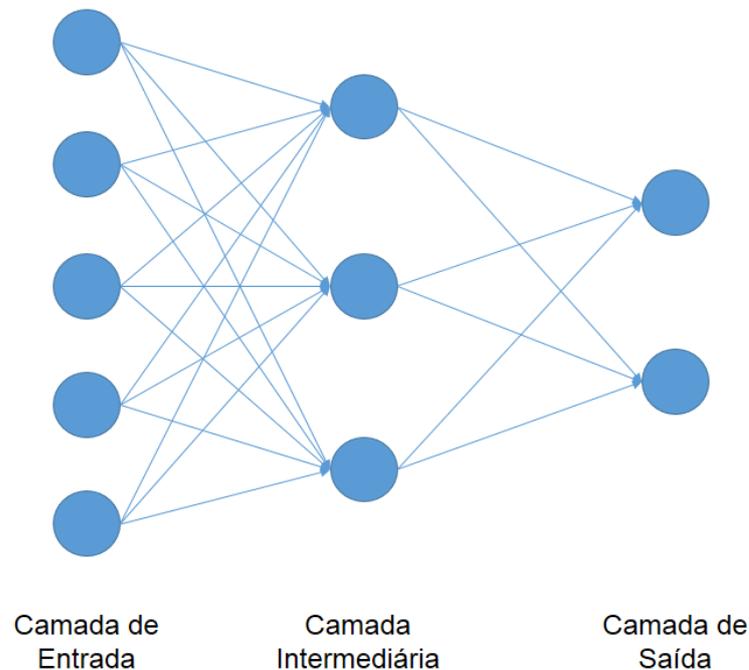


Figura 7 – Camadas do MLP

Essa rede neural tem sido utilizada para resolver problemas em treinamento de conjuntos através do algoritmo de retropropagação de erro, o qual consiste basicamente em duas partes: a primeira em que o sinal se propaga através de unidades computacionais até obter a camada de saída e a segunda em que todos os pesos sinápticos são ajustados de acordo com uma regra de correção de erros (BREVE et al., 2007).

2.4.2.3 J48

O *J48*, proposto por Quinlan (1993), implementa o algoritmo *C4.5* para gerar árvores de decisão, as quais são utilizadas para classificar os conjuntos de dados (WU et al., 2007). *J48* constrói essas árvores a partir de um conjunto de treinamento de dados rotulado usando o conceito de *information entropy* (entropia da informação). Isto significa que cada atributo dos dados podem ser usados para criar uma decisão dividindo esses dados em subconjuntos menores (SAHU; MEHTRE, 2015). O *J48* examina o ganho de informação normalizada (diferença na entropia), que resulta em escolher um atributo para dividir os dados, de acordo com a equação 6.

$$ganho = info(T) - \sum_{i=1}^s \frac{|Ti|}{|T|} \times info(Ti) \quad (6)$$

onde T é um conjunto de casos e Ti são subconjuntos de T compostos por valores distintos de um atributo. A equação 7 é conhecida como *entropy function* (função entropia).

$$info(T) = - \sum_{j=1}^{Nclass} \frac{freq(C_j, T)}{|T|} \times \log_2 \left(\frac{freq(C_j, T)}{|T|} \right) \quad (7)$$

A decisão é feita de acordo com o atributo de maior ganho de informação normalizada. Então o algoritmo retorna aos subconjuntos menores. O procedimento de divisão para se todas instâncias do subconjunto pertencem à mesma classe. Neste caso, um nó folha é criado na árvore de decisão no intuito de escolher a classe. Entretanto, pode também ocorrer que nenhuma das características representem o ganho de informação. Neste caso, o *J48* cria um nó de decisão na árvore usando o valor esperado da classe. Uma das vantagens do *J48* é poder gerenciar atributos contínuos e discreto, treinando os dados de atributos sem valores e atributos com diferentes custos (SHARMA et al., 2015). O Algoritmo 2 mostra o pseudocódigo do *J48*.

Algorithm 2 Pseudocódigo do algoritmo *J48*

- Criar um nó raiz N;

if (*T* pertence a mesma categoria de *C*) **then**

 Nó Folha = N;

 Setar N como classe *C*;

 return N;

end

for *i* de 1 até *n* **do**

 Calcular o ganho de informação (*G*);

end

- *t* = atributo de teste;

- *N.t* = atributo que contém o maior ganho de informação;

if (*N.t* == *contínuo*) **then**

 Encontrar threshold;

end

for (cada *T'* na divisão de *T*) **do**

if (*T'* é vazio) **then**

 Filho de N é um nó folha;

else

 Filho de N = *dtree(T')*;

end

end

- Calcular o erro de classificação do nó N;

- return N;

2.4.2.4 ADABOOST

O Adaboost é um dos mais famosos algoritmos para criar um classificador de conjuntos forte combinando vários classificadores com membros fracos (HUANG; WANG, 2012).

A ideia principal é construir uma sucessão de conjuntos de aprendizado fracos, usando diferentes conjuntos de treinamento que são derivados da reamostragem dos dados originais (CAO et al., 2012).

A cada ciclo de execução, o peso dos membros desses conjuntos é ajustado; o peso dos membros das amostras de treinamento que são ordenadas incorretamente pelo classificador fraco utilizado são aumentados, enquanto o peso das amostras de treinamento que são classificadas corretamente são diminuídos (JI et al., 2014).

Através de uma votação ponderada, esses conjuntos são combinados para prever a classe de uma nova instância de teste (CAO et al., 2012).

Uma das razões para o Adaboost possuir bom desempenho é devido a sua habilidade de expansão e diversidade (ser utilizado com diversos classificadores fracos) (AN; KIM, 2010).

O algoritmo AdaBoost O algoritmo Adaboost, proposto por Freund e Schapire (1997), foi o primeiro algoritmo de boosting e ainda permanece como um dos mais utilizados e estudados, com aplicações em diversos campos. Ao longo do tempo, uma grande variedade de tentativas foram feitas para explicar o Adaboost como um algoritmo de aprendizado, no intuito de entender o porquê ele funciona, como ele funciona, e quando ele funciona (ou falha).

No Algoritmo 3, são dados m exemplos de treinamentos, no quais x_i pertence ao domínio de \mathcal{X} , e os rótulos y_i pertence à $\{-1, +1\}$. Em cada execução $t=1, \dots, T$, uma distribuição D_t é calculada e um dado conjunto de treinamento fraco é aplicado no intuito de encontrar uma hipótese fraca H_t , a qual tem por objetivo encontrar o erro ϵ_t relativo à D_t . A hipótese final H calcula o sinal de uma combinação ponderada de hipóteses fracas.

Algorithm 3 Algoritmo *Adaboost*

Dado: $(x_1, y_1), \dots, (x_m, y_m)$ onde $x_i \in \mathcal{X}$, $y_i \in$

Inicializa: $D_1(i) = 1/m$ para $i=1, \dots, m$

Para $t=1, \dots, T$:

- Treinar um conjunto de aprendizado usando a distribuição D_t ;
- Obter uma hipótese $h_t: \mathcal{X} \rightarrow \{-1, +1\}$
- O objetivo é selecionar h_t com baixo erro ponderado:

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$

- Escolher $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$

- Atualizar, para $i = 1, \dots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

onde Z_t é um fator de normalização (escolhido para que D_{t+1} seja uma distribuição)

Portanto, a hipótese de saída final será dada por:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

2.5 FATOR DE PERCEPÇÃO

A partir das imagens selecionadas como mais semelhantes de acordo com a seção 2.1, o usuário irá posicioná-las utilizando um critério de similaridade de acordo com sua opinião. Para isso, pensou-se na possibilidade de atribuir valores à percepção do usuário. Em Bugatti et al. (2011), definiu-se o Fator de Percepção, o qual é fundamentado em uma comparação aritmética entre a ordem de retorno da consulta (gerada pelo algoritmo) e a ordem de seleção (gerada pelo usuário). O Fator de Percepção é representado por:

$$PerceptionFactor(m, \varphi) = \frac{1}{\varepsilon_m} |s_m - r_m| \quad (8)$$

onde s_m é a localização da imagem m no vetor ordenado pela percepção do usuário e r_m é a posição da imagem m na lista gerada através da ordem de retorno da consulta para uma distância φ . ε_m é um fator de penalidade que tem como prioridade estabelecer um peso maior às primeiras imagens, uma vez que, de acordo com o usuário, elas possuem maior similaridade à imagem de consulta. Deste modo:

$$\varepsilon_m = \frac{|M|}{(|M| + 1) - s_m} \quad (9)$$

onde $|M|$ é a quantidade de imagens que foram selecionadas.

Com a intenção de reunir dados do Fator de Percepção para obter um nível de similaridade em relação à consulta, foi proposto o Fator de Percepção Médio:

$$APF(\varphi) = \frac{1}{|M|} \sum_{m \in M} PerceptionFactor(m, \varphi) \quad (10)$$

Desta maneira, pode-se notar que, quanto mais o valor do APF convergir para 0, melhor o resultado. Portanto, conclui-se que a seleção de imagens que retornar o menor APF é a que possui a melhor função de distância.

3 MÉTODO PROPOSTO

3.1 INTRODUÇÃO

Um dos desafios da recuperação de imagens baseada em conteúdo é conseguir um nível de similaridade que caracterize e quantifique, da melhor forma possível, a percepção do usuário. Desta forma, foi proposto neste capítulo uma metodologia que visa criar um sistema inteligente de aprendizado, baseado em vetores de característica de alto nível, que seja capaz de recuperar imagens por meio desta percepção do usuário.

3.2 MÉTODO PROPOSTO

A metodologia utilizada, de um modo geral, seguiu o fluxograma apresentado na Figura 8.

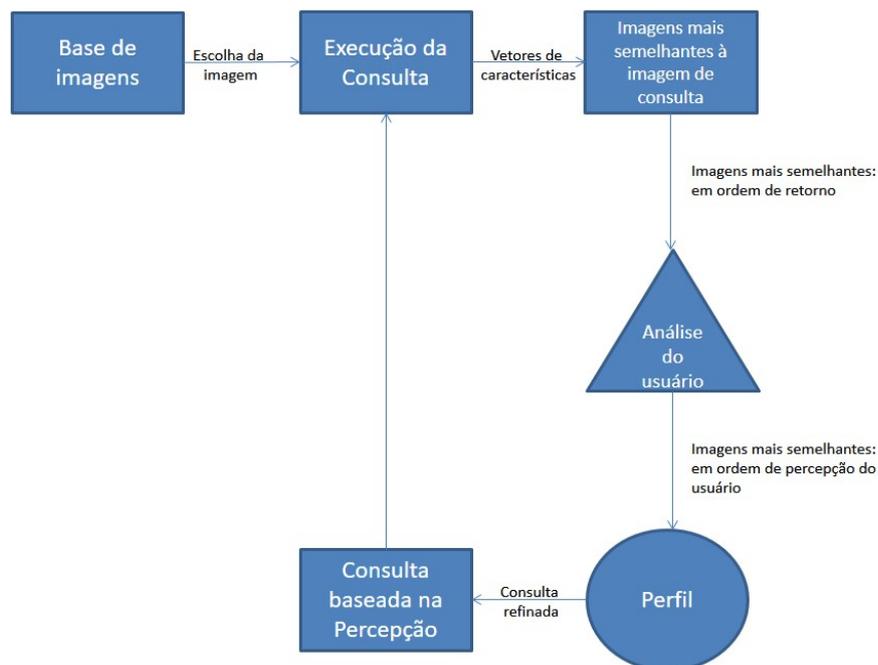


Figura 8 – Fluxo do método proposto

Portanto, o primeiro passo foi, por meio da *base de imagens*, realizar a extração de características de cada uma destas imagens. As características de cor foram obtidas por meio do histograma. Para a textura, foram utilizadas as Matrizes de Co-ocorrência com descritores de Haralick e para as características de forma os momentos de Zernike, de acordo com a Seção 2.2.

Em seguida, o usuário *escolhe uma imagem* de cada tipo de lesão para que fosse realizada a *execução da consulta*, na qual são gerados os vetores de distâncias e *vetores de características*.

No intuito de encontrar o melhor par extrator de características/distância, são definidas nove classes com as seguintes combinações:

- Histograma/L1 → *Classe 1*
- Histograma/L2 → *Classe 2*
- Histograma/L ∞ → *Classe 3*
- Textura/L1 → *Classe 4*
- Textura/L2 → *Classe 5*
- Textura/L ∞ → *Classe 6*
- Forma/L1 → *Classe 7*
- Forma/L2 → *Classe 8*
- Forma/L ∞ → *Classe 9*

A fim de encontrar as *Imagens mais semelhantes à imagem de consulta*, deve-se descobrir o par extrator/distância que obtém melhores resultados para cada tipo de imagem. Inicialmente este par é definido através do cálculo da precisão (proporção de imagens relevantes que foram recuperadas por todas as imagens recuperadas), por meio da Equação 5.

$$precision = \frac{|R_A|}{|R|} \quad (5)$$

A imagem escolhida pelo usuário são comparadas, portanto, com todas as outras imagens do banco. A classe que obtém melhor precisão, é definida como o “melhor” par inicial. Por exemplo, se a melhor precisão é obtida pela classe 1, a função de distância *L1* será aplicada aos vetores de característica do *Histograma*.

A Figura 9 demonstra um exemplo de cálculo do vetor de distâncias para uma base

com 5 imagens.

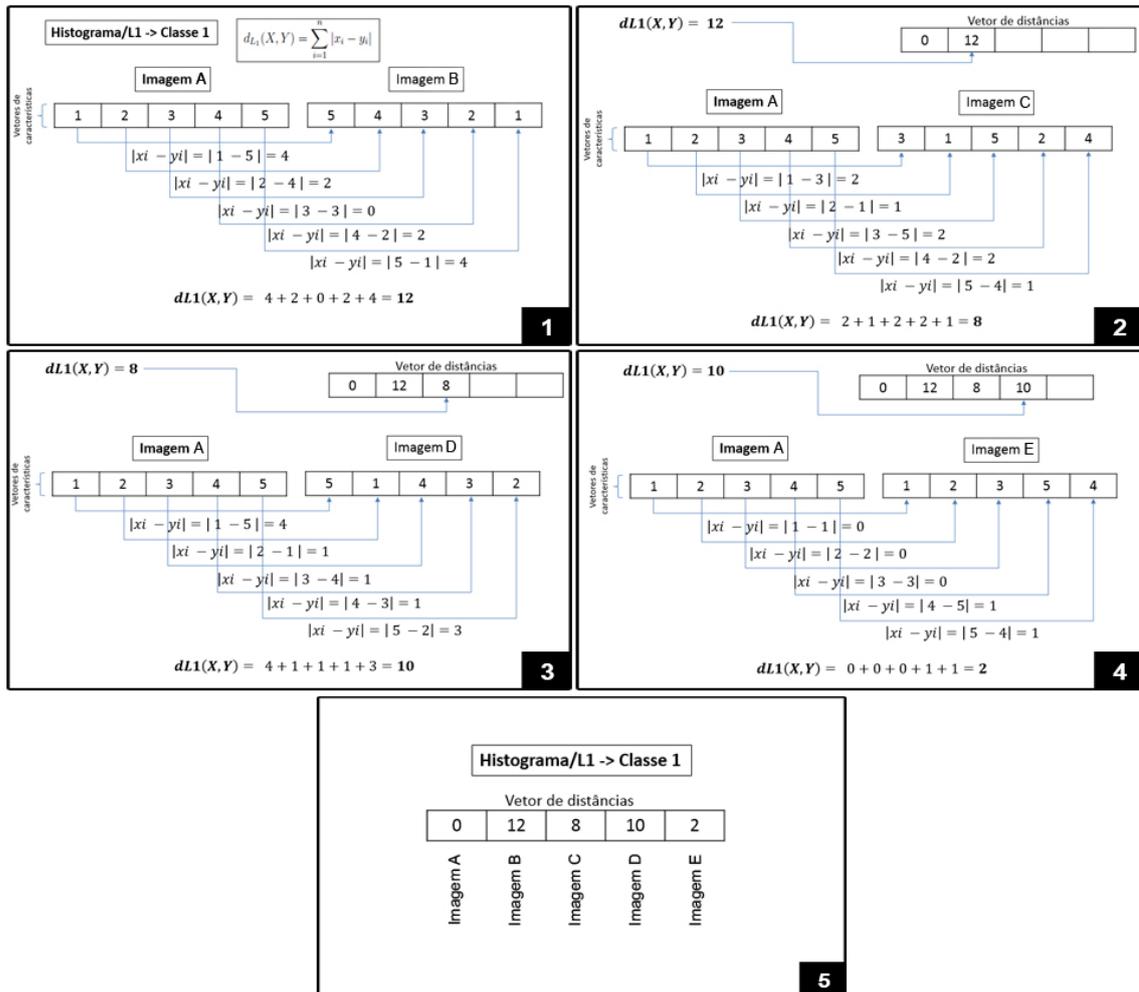


Figura 9 – Exemplo de cálculo do vetor de distâncias

Em seguida, os vetores de distância são ordenados de forma crescente, afim de relacionar as imagens que possuem menor distância (e, portanto, mais semelhantes) da imagem de consulta, gerando assim novos vetores denominados *ordem de retorno* (r_m).

Portanto, de acordo com o conceito de Consultas por Similaridade abordado na Seção 2.1, são retornadas “k” (busca pelos vizinhos mais próximos) *imagens mais semelhantes à imagem de consulta em ordem de retorno*.

A partir dos vetores r_m , é realizada a *análise do usuário* por meio do Algoritmo 4, de acordo com o método exposto na Seção 2.5. Nessa etapa o usuário pode reordenar essas imagens de acordo com seu critério de similaridade, gerando assim um vetor *ordem de percepção do usuário* (s_m) com o *perfil* deste usuário.

Algorithm 4 Simulação da Percepção de Usuários

Data: Considere “vet1” um vetor de “k” posições, ordenado de acordo com uma função de distância. “Vet2” foi preenchido de acordo com uma simulação randômica a partir do “vet1”.

```

for i de 0 até m do
  vet[i] ← aleatorio() * (k + 1);
  for k de -1 até i do
    if vet[k] = vet[i] then
      vet[i] ← aleatorio() * (k + 1);
      k ← -1;
    end
  end
  vet2[i] ← vet1[vet[i]];
end

```

Logo após, com o auxílio dos vetores s_m e r_m , é realizado o cálculo do APF. O par extrator/distância que obtiver *menor* APF, é definido como *melhor*. A Figura 10 demonstra um exemplo deste processo, o qual é executado pelo Algoritmo 5.

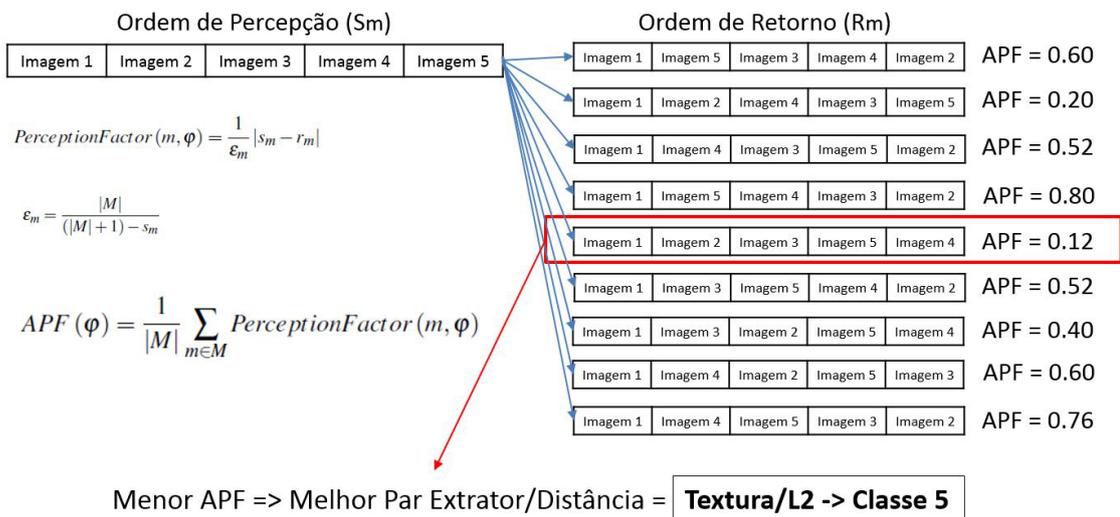


Figura 10 – Cálculo do APF

Algorithm 5 Definição de classes

```

for  $j$  de 0 até 2 do
  for  $i$  de 0 até  $m$  do
     $vet[i] \leftarrow (((MOD(m) + 1) - (vetor_{r,m}[i]))/MOD(m)) * MOD((vetor_{r,m}[i] -$ 
     $(vetor_{s,m}[i]));$ 
     $APF \leftarrow APF + vet[i];$ 
  end
   $APF \leftarrow APF / m;$ 
  if  $APF < MenorAPF$  then
     $MenorAPF \leftarrow APF;$ 
     $classe \leftarrow j;$ 
  end
end

```

Portanto, o sistema pode ser realimentado utilizando este par extrator/distância como inicial (ao invés do par definido por meio do cálculo da precisão), gerando assim uma *consulta baseada na percepção do usuário*.

Após essa simulação, é aplicado ao conjunto de dados alguns classificadores (Seção 2.4), no intuito de descobrir qual deles obtém melhor acurácia para a base de dados em questão. A Equação 6, a qual indica o cálculo da porcentagem de acurácia, representa a quantidade de imagens previstas corretamente de cada classe em relação ao número total de imagens.

$$acurácia = 100 \times \frac{(VP + VN)}{(VP + VN + FP + FN)} \quad (6)$$

em que VP é o número de verdadeiros positivos, VN o número de verdadeiros negativos, FP o número de falsos positivos e FN o número de falsos negativos.

Uma das vantagens de utilizar este processo é que, além de captar a semântica de consulta desejada pelo usuário, vetores de características são gerados com dimensões bem menores quando comparadas às dos extratores de baixo nível aplicados às imagens.

3.3 EXPERIMENTOS

3.3.1 MATERIAS E MÉTODOS

Para cada classe de imagens da base de dados, foi gerado um arquivo de texto, o qual possui 3000 registros (usuários), cada um com suas respectivas classes (1 à 9), ordem de

percepção (Sm) e fator de percepção médio (APF). Em seguida, estes arquivos foram convertidos em arquivos *.arff* para que fosse possível aplicar o processo de classificação na aplicação WEKA.

Foi utilizado como opção de teste, para todos os classificadores (Adaboost, J48, MLP e Naive Bayes), a validação cruzada, com o número de folds igual a 10.

Este processo foi realizado em um computador com processador Intel Core i7, 1.8 GHz 3.0 GHz com Turbo Boost (Geração 4) e memória ram de 8 GB, 1600 MHz, DDR3.

3.3.2 DESCRIÇÃO DA BASE DE IMAGENS

Para gerar os vetores de características com base na percepção do usuário, foi necessário primeiramente extrair as características de cor, textura e forma de um banco de imagens. A base utilizada nesse trabalho, denominada *CTROIsPulmão*, possui 3248 imagens de regiões extraídas de imagens de Tomografia Computadorizada do Tórax. Tais imagens têm dimensões de 64 x 64 e *pixels* de 8 bits, o que resulta em 256 tons de cinza. A base foi fragmentada em 6 classes. Por isso, é de extrema importância captar a melhor percepção do usuário especialista. A Figura 11 ilustra exemplos das imagens pertencentes a cada uma das classes da base de imagens: (a) Normal (590 imagens), (b) Consolidação (451 imagens), (c) Enfisema (502 imagens), (d) Espessamento (590 imagens), (e) Favo de Mel (520 imagens) e (f) Vidro Fosco (595 imagens).

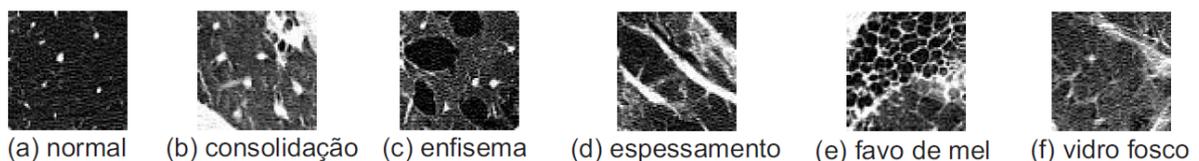


Figura 11 – Exemplos de imagens de cada classe pertencentes ao conjunto de teste da base de imagens de ROIs de pulmão oriundas de exames de tomografia computadorizada *CTROIs-Pulmão*

3.3.3 RESULTADOS

As Tabelas de 1 a 6 mostram os resultados das análises realizadas para cada classificador sendo que, individualmente, são apresentadas as médias de acurácia (em %) com seu respectivo desvio padrão e o tempo de classificação (em segundos) dos mesmos.

Tabela 1 – Classe 1

Classificador	Acurácia Média	Tempo Médio (s)	Desvio Padrão
NaiveBayes	86.50	0.27	0.609
MLP	87.07	111.60	0.325
J48	88.67	1.42	0.356
AdaBoost/NaiveBayes	93.20	8.83	0.351
AdaBoost/MLP	87.67	600.40	0.328
AdaBoost/J48	97.07	14.40	0.447

Conforme demonstrado na Tabela 1, o classificador *Naive Bayes*, obteve uma precisão de 86.50% e levou 0.27 segundos para classificar o conjunto de dados. Portanto, ele teve boa acurácia levando em conta o tempo de classificação. Seu desvio padrão ficou em uma faixa razoável, o que mostra que esse classificador pode ser relativamente confiável. O classificador *Multilayer Perceptron* obteve uma acurácia de 87.07% e gastou 111.60 segundos para realizar a classificação. Pode-se concluir então, que esse classificador possui uma boa acurácia, todavia o seu tempo de classificação é bem alto comparado com o primeiro concorrente. Seu desvio padrão foi bem pequeno, e tudo indica que o *Multilayer* é bastante confiável por apresentar pequenas variações. Já o classificador *J48* apresentou uma acurácia de 88.67% e classificou a amostra em 1.42 segundos. Esse classificador possuiu, então, um bom desempenho. Isso se deve ao fato do mesmo ter obtido uma acurácia alta em curto tempo. Contudo, seu desvio foi maior que os outros classificadores já analisados, o que mostra que o *J48* pode apresentar maiores variações de acurácia. Com o uso do *AdaBoost* foi possível notar, com exceção do *Multilayer Perceptron*, que as acurácias foram ainda melhores e os desvios ainda menores. Apenas o tempo de classificação (fato que já era esperado) foi maior.

Tabela 2 – Classe 2

Classificador	Acurácia Média	Tempo Médio	Desvio Padrão
NaiveBayes	87.37	0.27	0.650
MLP	86.73	110.33	0.293
J48	87.87	1.68	0.261
AdaBoost/NaiveBayes	94.30	9.01	0.451
AdaBoost/MLP	86.73	2141.69	0.293
AdaBoost/J48	95.90	16.47	0.206

Seguindo a Tabela 2, pode-se observar que o classificador *Naive Bayes* obteve uma precisão de 87.37% e realizou a classificação em 0.27 segundos. Da mesma forma que na Tabela 1, ele atingiu boa importância de acurácia levando em conta o tempo de classificação. Contudo, seu desvio padrão foi maior do que na primeira análise e apresentou um valor de 0.650. Já o

classificador *Multilayer Perceptron* atingiu uma acurácia de 86.73% e gastou 110.33 segundos para classificar o conjunto de dados. Desta forma, pode-se concluir que esse classificador obteve uma boa acurácia, todavia o seu tempo de classificação foi, novamente, alto. Seu desvio padrão de 0.261 foi bem pequeno, e mais uma vez observa-se que esse classificador é bastante confiável. Utilizando o classificador *J48*, obteve-se uma acurácia de 87.87% e um tempo de 1.68 segundos. Esse classificador apresentou, mais uma vez, um bom desempenho. Porém, desta vez seu desvio foi de 0.261, que é menos da metade em relação ao obtido anteriormente. Mais uma vez, ao usar *AdaBoost* notou-se melhorias, com exceção do *Multilayer Perceptron*, no desempenho dos classificadores.

Tabela 3 – Classe 3

Classificador	Acurácia Média	Tempo Médio	Desvio Padrão
NaiveBayes	87.80	0.34	0.657
MLP	86.77	165.31	0.427
J48	87.77	1.68	0.427
AdaBoost/NaiveBayes	94.67	11.89	0.347
AdaBoost/MLP	87.10	494.35	0.427
AdaBoost/J48	94.70	17.35	0.516

Analisando a Tabela 3, foi possível notar que o classificador *Naive Bayes* atingiu uma precisão de 87.80% e levou um 0.34 segundos para realizar a classificação. Portanto, mais uma vez observa-se que esse classificador possui uma boa acurácia levando em conta o tempo de classificação. Contudo, seu desvio padrão de 0.657 foi ainda maior do que os apresentados anteriormente, o que pode induzir que talvez este não seja um classificador confiável. Para o classificador *Multilayer Perceptron* a acurácia foi de 86.77%, com um tempo de classificação de 165.31 segundos. Com isso, conclui-se que esse classificador obteve uma boa acurácia, todavia o seu tempo de classificação foi, de modo semelhante às análises anteriores, alto. Seu desvio padrão foi de 0.427, o qual está em uma faixa razoável. Com o uso classificador *J48*, a acurácia média convergiu para 87.77% e um tempo de 11.89 segundos. Para a terceira Classe de imagens, esse classificador obteve um desvio padrão alto se comparado com os desvios das Classes 1 e 2. Utilizando o *AdaBoost*, com exceção do *Multilayer Perceptron*, os classificadores aumentaram as acurácias.

Observando a Tabela 4, o classificador *Naive Bayes*, convergiu para uma precisão de 86.23% e levou um tempo de 0.27 segundos para realizar a classificação. Da mesma forma que nas Tabelas de 1 a 3, ele obteve boa acurácia levando em conta o tempo de classificação. Seu desvio padrão não foi tão alto se comparado aos casos anteriores, e apresentou um valor de

Tabela 4 – Classe 4

Classificador	Acurácia Média	Tempo Médio	Desvio Padrão
NaiveBayes	86.23	0.27	0.573
MLP	75.73	188.10	0.345
J48	87.70	1.65	0.378
AdaBoost/NaiveBayes	94.00	9.91	0.397
AdaBoost/MLP	76.63	376.16	0.544
AdaBoost/J48	95.50	33.56	0.503

0.573. O classificador *Multilayer Perceptron* atingiu uma acurácia de 75.73% e gastou 188.10 segundos para classificar o conjunto de dados. Seu desvio padrão foi novamente bem pequeno, aumentando o conceito de que este é um classificador bastante confiável. Para o classificador *J48*, obteve-se uma acurácia de 87.70% e um tempo de 1.65 segundos. Como nos casos anteriores, esse classificador apresentou um bom desempenho. Porém, desta vez seu desvio de 0.378 foi bem baixo. Como era de se esperar, o uso do AdaBoost trouxe melhorias, com exceção do *Multilayer Perceptron*, no desempenho dos classificadores.

Tabela 5 – Classe 5

Classificador	Acurácia Média	Tempo Médio	Desvio Padrão
NaiveBayes	86.90	0.40	0.790
MLP	86.50	192.76	0.349
J48	87.00	1.67	0.395
AdaBoost/NaiveBayes	94.27	10.87	0.442
AdaBoost/MLP	87.70	1964.54	0.349
AdaBoost/J48	94.73	34.48	0.331

De acordo com a Tabela 5, o classificador *Naive Bayes*, obteve uma precisão de 86.90% e levou um 0.40 segundos para realizar a classificação. Igualmente aos casos anteriores, ele conseguiu boa acurácia levando em conta o tempo de classificação. Contudo, seu desvio padrão foi ainda maior do que os apresentados anteriormente, o que pode induzir que este não seja um classificador confiável. O classificador *Multilayer Perceptron* culminou em uma acurácia de 86.50% e gastou 192.76 segundos para realizar a classificação. Com isso, conclui-se que esse classificador obteve uma boa acurácia, todavia o seu tempo de classificação foi, de modo semelhante aos casos anteriores, alto. Seu desvio padrão foi de 0.349. Com o classificador *J48*, a acurácia média convergiu para 87.00% e um tempo de 1.67 segundos. O desvio padrão também foi baixo, e apresentou um valor de 0.395. Fazendo uma junção com o AdaBoost, com exceção do *Multilayer Perceptron*, os classificadores aumentaram as acurácias.

Tabela 6 – Classe 6

Classificador	Acurácia Média	Tempo Médio	Desvio Padrão
NaiveBayes	87.87	0.26	0.571
MLP	86.47	107.11	0.560
J48	86.23	1.55	0.177
AdaBoost/NaiveBayes	94.43	8.76	0.324
AdaBoost/MLP	86.47	424.43	0.260
AdaBoost/J48	95.20	14.76	0.353

Conforme Tabela 6, o classificador *Naive Bayes* obteve uma precisão de 87.87% e realizou a classificação em 0.26 segundos. Apesar de ter a acurácia decrementada, em razão decimal em relação aos outros casos, ele mostrou um bom desempenho levando em conta o tempo de classificação. Seu desvio padrão apresentou um valor de 0.571. Já o classificador *Multilayer Perceptron* atingiu uma acurácia de 86.47% e levou 107.11 segundos para classificar o conjunto de dados. Esse classificador obtém, portanto, uma boa acurácia e um tempo de classificação alto, se comparado ao de seus concorrentes. Seu desvio padrão de 0.760 foi o maior entre os já analisados. Esse fato, porém, não o tira o conceito do mesmo ser um classificador confiável. Utilizando o classificador *J48*, a acurácia obtida foi de 86.23% e um tempo de 1.55 segundos. Portanto, o que se pode concluir deste classificador é que ele possui boa relação entre acurácia e tempo. Contudo, seu desvio padrão é bastante variável. Para a Classe 6, ele apresentou um desvio de 0.177. Em relação ao *AdaBoost* pode-se observar que, com exceção do *Multilayer Perceptron*, ele aumenta a acurácia e diminui o desvio padrão.

Portanto, na média o classificador que obteve melhor resultado para a base de dados em questão, foi o *J48*. Contudo, se a necessidade do tempo de execução for maior que a necessidade de acurácia, o *Naive Bayes* pode ser o melhor. Já o classificador *Multilayer Perceptron* possuiu um desempenho diferente de seus concorrentes, pois apesar de obter uma boa acurácia, demorou para classificar o mesmo conjunto de dados. Também é possível observar que, embora tenha aumentado o tempo de classificação, utilizando o *AdaBoost* foi possível melhorar a acurácia. O desvio padrão para as amostras analisadas foi relativamente pequeno, o que significa que, de modo geral, os classificadores podem ser confiáveis. Pode-se então concluir que o melhor classificador dependerá muito da necessidade do usuário. Todavia, em termos de acurácia, o classificador que conseguiu atingir melhores resultados foi o *J48* com o auxílio do *AdaBoost*.

4 CONCLUSÕES

4.1 CONSIDERAÇÕES FINAIS

No presente trabalho foi proposta uma nova técnica para geração de vetores de características de alto nível capazes de descrever a semântica da consulta por similaridade desejada pelo usuário. Dessa forma, contribuindo de maneira importante para o tratamento do problema do *gap* semântico em sistemas CBIR. O enfoque do trabalho foi em imagens biomédicas devido à grande aplicação e utilidade de tais técnicas no contexto abordado. No entanto, vale ressaltar que a técnica proposta pode ser aplicada a qualquer contexto de imagens, gerando assim possibilidades de extensões a diversas aplicações. Com o auxílio dos resultados obtidos, foi possível demonstrar a viabilidade e aplicabilidade da técnica a ser proposta, bem como a possibilidade de extensão da mesma para outros contextos de imagens, tipos de características e funções de distância.

4.2 RESULTADOS OBTIDOS

O presente trabalho obteve os seguintes resultados:

- Desenvolvimento de uma nova técnica para geração de vetores de características de alto nível;
- Estudo e análise do comportamento de diversos algoritmos de classificação supervisionada para o contexto em questão, levando em consideração não só a acurácia dos mesmos, bem como o tempo de processamento, dada a necessidade de escalabilidade dos mesmos ao tratar de repositórios de imagens de crescimento contínuo;
- Acoplamento da técnica desenvolvida ao processo tradicional de consultas por similaridade;

REFERÊNCIAS

- AMAYEH, G. et al. Improvement of zernike moment descriptors on affine transformed shapes. In: **Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on**. [S.l.: s.n.], 2007. p. 1–4.
- AN, T. K.; KIM, M. H. A new diverse adaboost classifier. In: **Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on**. [S.l.: s.n.], 2010. v. 1, p. 359–363.
- BANDA, J.; ANGRYK, R. Selection of image parameters as the first step towards creating a cbir system for the solar dynamics observatory. In: **Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on**. [S.l.: s.n.], 2010. p. 528–534.
- BELKASIM, S.; HASSAN, E.; OBEIDI, T. Radial zernike moment invariants. In: **Computer and Information Technology, 2004. CIT '04. The Fourth International Conference on**. [S.l.: s.n.], 2004. p. 790–795.
- BERNARD, I. Multilayer perceptron and uppercase handwritten characters recognition. In: **Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on**. [S.l.: s.n.], 1993. p. 935–938.
- BREVE, F. A.; PONTI-JUNIOR, M. P.; MASCARENHAS, N. D. A. Multilayer perceptron classifier combination for identification of materials on noisy soil science multispectral images. In: **Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on**. [S.l.: s.n.], 2007. p. 239–244. ISSN 1530-1834.
- BUGATTI, P. et al. Content-based retrieval of medical images by continuous feature selection. In: **Computer-Based Medical Systems, 2008. CBMS '08. 21st IEEE International Symposium on**. [S.l.: s.n.], 2008. p. 272–277. ISSN 1063-7125.
- BUGATTI, P. H. et al. Integrating user profile in medical cbir systems to answer perceptual similarity queries. In: . [S.l.: s.n.], 2011. v. 7963, p. 79630Q–79630Q–12.
- BUGATTI, P. H.; TRAINA, A. J. M.; TRAINA JR., C. Assessing the best integration between distance-function and image-feature to answer similarity queries. In: **Proceedings of the 2008 ACM symposium on Applied computing**. New York, NY, USA: ACM, 2008. (SAC '08), p. 1225–1230. ISBN 978-1-59593-753-7. Disponível em: <<http://doi.acm.org/10.1145/1363686.1363969>>.
- CAO, J.; KWONG, S.; WANG, R. A noise-detection based adaboost algorithm for mislabeled data. **Pattern Recogn.**, Elsevier Science Inc., New York, NY, USA, v. 45, n. 12, p. 4451–4465, dez. 2012. ISSN 0031-3203. Disponível em: <<http://dx.doi.org/10.1016/j.patcog.2012.05.002>>.
- CHIANG, C.-C.; FU, H.-C. A variant of second-order multilayer perceptron and its application to function approximations. In: **Neural Networks, 1992. IJCNN., International Joint Conference on**. [S.l.: s.n.], 1992. v. 3, p. 887–892 vol.3.

DATTA, R. et al. Image retrieval: Ideas, influences, and trends of the new age. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 40, n. 2, p. 5:1–5:60, maio 2008. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1348246.1348248>>.

DHIR, R. Moment based invariant feature extraction techniques for bilingual character recognition. In: **Education Technology and Computer (ICETC), 2010 2nd International Conference on**. [S.l.: s.n.], 2010. v. 4, p. V4–80–V4–84.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, n. 2-3, p. 103–130, nov. 1997. ISSN 0885-6125. Disponível em: <<http://dx.doi.org/10.1023/A:1007413511361>>.

FLICKNER, M. et al. Query by image and video content: the qbic system. **Computer**, v. 28, n. 9, p. 23–32, 1995. ISSN 0018-9162.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **J. Comput. Syst. Sci.**, Academic Press, Inc., Orlando, FL, USA, v. 55, n. 1, p. 119–139, ago. 1997. ISSN 0022-0000. Disponível em: <<http://dx.doi.org/10.1006/jcss.1997.1504>>.

GUHA, D.; PATRA, S. Cochannel interference minimization using wilcoxon multilayer perceptron neural network. In: **Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on**. [S.l.: s.n.], 2010. p. 145–149.

HUANG, T.; WANG, Z. Face detection by improved adaboost. In: **Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on**. [S.l.: s.n.], 2012. p. 981–985.

JI, S.; LU, X.; XU, Q. A fast face detection method combining skin color feature and adaboost. In: **Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on**. [S.l.: s.n.], 2014. p. 1–5.

JIANG, L.; ZHANG, H. Learning instance greedily cloning naive bayes for ranking. In: **Data Mining, Fifth IEEE International Conference on**. [S.l.: s.n.], 2005. p. 8 pp.–. ISSN 1550-4786.

KHAN, M. S. A.; ANSARI, M. A.; MIYA, J. Enhancement of color image retrieval capabilities: Fusion of color with texture optimization. In: **Communication Systems and Network Technologies (CSNT), 2011 International Conference on**. [S.l.: s.n.], 2011. p. 541–544.

KLAWONN, F.; ANGELOV, P. Evolving extended naive bayes classifiers. In: **Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on**. [S.l.: s.n.], 2006. p. 643–647.

KOTOULAS, L.; ANDREADIS, I. Colour histogram content-based image retrieval and hardware implementation. **IEE Proceedings - Circuits, Devices and Systems**, v. 150, n. 5, p. 387–93–, Oct 2003. ISSN 1350-2409.

LUO, S. et al. A liver segmentation algorithm based on wavelets and machine learning. In: **Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on**. [S.l.: s.n.], 2009. v. 2, p. 122–125.

- OSMAN, M. K. et al. 3d object recognition using manfis network with orthogonal and non-orthogonal moments. In: **Signal Processing Its Applications, 2009. CSPA 2009. 5th International Colloquium on.** [S.l.: s.n.], 2009. p. 302–306.
- PONCIANO-SILVA, M. et al. Including the perceptual parameter to tune the retrieval ability of pulmonary cbir systems. In: **Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on.** [S.l.: s.n.], 2009. p. 1–8. ISSN 1063-7125.
- QIN, Z. Naive bayes classification given probability estimation trees. In: **Machine Learning and Applications, 2006. ICMLA '06. 5th International Conference on.** [S.l.: s.n.], 2006. p. 34–42.
- RASHEED, W. et al. Image retrieval using maximum frequency of local histogram based color correlogram. In: **Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on.** [S.l.: s.n.], 2008. p. 62–66.
- RUBERTO, C. D.; FODDE, G. Statistical features for image retrieval a quantitative comparison. In: **Computer Vision Theory and Applications (VISAPP), 2014 International Conference on.** [S.l.: s.n.], 2014. v. 1, p. 610–617.
- SAHU, S.; MEHTRE, B. M. Network intrusion detection system using j48 decision tree. In: **Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on.** [S.l.: s.n.], 2015. p. 2023–2026.
- SHARMA, P.; SINGH, D.; SINGH, A. Classification algorithms on a large continuous random dataset using rapid miner tool. In: **Electronics and Communication Systems (ICECS), 2015 2nd International Conference on.** [S.l.: s.n.], 2015. p. 704–709.
- STRICKER, M. A.; ORENKO, M. Similarity of color images. In: **San Jose, CA:s.n.** [S.l.: s.n.], 1995. p. 381–392.
- SUGANYA, R.; RAJARAM, S. Feature extraction and classification of ultrasound liver images using haralick texture-primitive features: Application of svm classifier. In: **Recent Trends in Information Technology (ICRTIT), 2013 International Conference on.** [S.l.: s.n.], 2013. p. 596–602.
- WU, X. et al. Top 10 algorithms in data mining. **Knowl. Inf. Syst.**, Springer-Verlag New York, Inc., New York, NY, USA, v. 14, n. 1, p. 1–37, dez. 2007. ISSN 0219-1377. Disponível em: <<http://dx.doi.org/10.1007/s10115-007-0114-2>>.
- XIAOHONG, Y.; JINHUA, X. The related techniques of content-based image retrieval. In: **Computer Science and Computational Technology, 2008. ISCST '08. International Symposium on.** [S.l.: s.n.], 2008. v. 1, p. 154–158.
- ZAKARIYA, S. M.; ALI, R.; AHMAD, N. Combining visual features of an image at different precision value of unsupervised content based image retrieval. In: **Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on.** [S.l.: s.n.], 2010. p. 1–4.
- ZHANG, Z.; LI, W.; LI, B. An improving technique of color histogram in segmentation-based image retrieval. In: **Information Assurance and Security, 2009. IAS '09. Fifth International Conference on.** [S.l.: s.n.], 2009. v. 2, p. 381–384.