

08 a 11 de Outubro de 2018  
Instituto Federal Fluminense  
Búzios - RJ

## AVALIANDO DIFERENTES IMPLEMENTAÇÕES DO DESCRITOR DE COR DOMINANTE

**João Victor de Oliveira Novaes**<sup>1</sup> - j.victoresp@gmail.com

**Caio César da Silva Benedito**<sup>1</sup> - caiobenedito@msn.com

**Lúcio Fernandes Dutra Santos**<sup>1</sup> - lucio.santos@ifnmg.edu.br

<sup>1</sup>Instituto Federal Norte de Minas Gerais - Campus Montes Claros

**Resumo.** Neste trabalho é proposta uma plataforma de teste para implementações de quatro versões do extrator de característica descritor de cor dominante definido pelo padrão MPEG-7. A plataforma simula a tarefa de recuperação de imagem baseada em conteúdo e avalia tanto a qualidade das respostas fornecidas quanto o tempo médio de execução dos algoritmos. Foram conduzidos experimentos utilizando três conjuntos de dados reais e os resultados experimentais demonstram que a escolha da implementação do extrator de característica impactam tanto na etapa de extração dos vetores de características quanto no tempo de consulta.

**Keywords:** Recuperação de imagens por conteúdo, Extratores de Característica, Descritor de Cor Dominante

### 1. INTRODUÇÃO

A utilização de dados no formato de imagens tem se tornado cada vez mais frequente no cotidiano e uma prova deste fenômeno é a quantidade de imagens geradas diariamente em redes sociais, centros médicos de diagnóstico (Pedrosa, 2016). O crescente aumento na geração de imagem está associado a facilidade de acesso à dispositivos que geram/capturam imagens digitais, por exemplo, *smartphones*, câmeras digitais, *notebooks* e outros *gadgets*, dentre outros. Logo, imagens se tornaram uma parte importante do mundo moderno, além de fonte de entretenimento, elas também são de suma importância para áreas como a medicina, publicidade e gestão de produção.

Dada a importância da utilização de imagens em vários domínios de aplicação, surgiu uma demanda por sistemas que sejam capazes de recuperar e/ou armazenar imagens de forma eficiente, culminando nos então chamados sistemas de recuperação de imagens por conteúdo (CBIR na sigla em inglês – *Content-Based Image Retrieval*) (Pedrosa, 2016; Silva et al., 2016). Um sistema CBIR possibilita a recuperação de imagens considerando apenas o conteúdo visual (cor, textura e forma) presente nas próprias imagens, em oposição as tradicionais técnicas de recuperação baseada em texto que utilizam atributos textuais associados externamente as imagens. Para que um sistema CBIR seja capaz de recuperar as imagens baseado no conteúdo visual

é necessário utilizar algoritmos automáticos ou semi-automáticos de visão computacional para extrair características visuais consideradas relevantes das imagens. Este tipo de algoritmo é denominado extrator de características cujo objetivo é representar o conteúdo visual das imagens por meio de medidas numéricas que capturam alguma propriedade visual da imagem em estruturas de dados chamadas de vetores de característica. A representação das imagens por características se faz necessária, pois a comparação *pixel à pixel* entre duas imagens possui pouca correspondência com a percepção humana (Pedrosa, 2016; Santos, 2017). Assim, um sistema CBIR utiliza a representação de vetores de característica em oposição à imagem propriamente dita para realizar uma consulta ou outras tarefas de mineração de dados tais como agrupamento e classificação (Kimura et al., 2011).

Dentre as informações visuais presentes em uma imagem, as informações de cor são as mais expressivas e simples, pois a cor é uma das características que se ajusta perfeitamente à visão humana e, geralmente, é uma das primeiras características em que um observador considera ao observar uma imagem ou paisagem (Dewi et al., 2017). Por este e outros fatores, a cor é uma das características mais utilizadas em sistemas CBIR (Delmondes et al., 2015), por conseguinte, sendo motivo de pesquisa há vários anos, resultando em diversos algoritmos de extração de características que tem como objetivo extrair informações de cor relevantes das imagens. Uma das mais antigas e utilizadas técnicas para extrair informação de cor de imagens é o histograma de cor (M.J. Swain, 1991). De maneira geral, um histograma de cor consiste em representar a frequência de cada cor presente na imagem em um determinado espaço de cor. Há diversas maneiras de se empregar os histogramas de cores, por exemplo, considerando um único histograma para cada imagem (histograma global de cor) ou particionando a imagem em um número fixo de células e computando um histograma de cada célula (histograma local de cor). No entanto, apesar de ser uma maneira eficiente e muito simples de se capturar informação de cor, desvantagens como não capturar a localização de objetos e contar com estruturas de tamanho fixo tornaram o uso menos frequente nas atuais aplicações.

Visando superar tais desvantagens foi definido o padrão MPEG-7 (Martinez et al., 2002), em que diversos métodos de extração de características de cor, cada um com uma especialidade, são listados para auxiliar na tarefa de escolha de um extrator de acordo com o conjunto de imagem. Dentre os extratores listados temos o descritor de cor dominante (DCD na sigla em inglês - *Dominant Color Descriptor*) que gera uma representação da imagem, tal que, um pequeno conjunto de cores, bem como as propriedades estatísticas (distribuição ou variância) de cada uma dessas cores, sobre toda a imagem ou sobre regiões dela (Dewi et al., 2017).

Contudo, há várias implementações de DCD, como as apresentadas em Pavão (2016), Oliveira et al. (2018) e Yang et al. (2012), porém no melhor do nosso conhecimento, ainda não há relatos comparativos sobre as diferentes formas de implementar um DCD. Assim, o objetivo deste trabalho consiste em definir uma plataforma de teste para comparar as maneiras de se codificar um descritor de cor dominante em relação ao tempo de execução e a qualidade quando submetidos em uma tarefa de recuperação por conteúdo de imagem. Foram considerados três conjuntos de dados de imagens disponíveis publicamente para os testes e os resultados indicam que não houve um método superior em todos os conjuntos de dados quando se trata da qualidade, porém em relação ao desempenho, um dos métodos se mostrou superior em todos os testes.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 apresenta os principais conceitos relacionados. A seção 3 descreve a metodologia adotada. A Seção 4 detalha os experimentos realizados em três conjuntos de dados reais de imagem e analisa os resultados alcançados. Finalmente, a Seção 5 resume as conclusões deste trabalho.

## 2. CONCEITOS RELACIONADOS

### 2.1 MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICA DA MPEG-7

O padrão MPEG-7 foi proposto pela ISO/IEC JTC1 (Martinez et al., 2002). De maneira geral, o padrão define a representação esperada para extração de características em relação as propriedades visuais de cor, textura e forma. Para alcançar este objetivo, um conjunto de extratores foi definido para extrair as representações das imagens, propriamente ditas, de forma eficiente sem considerar qualquer especificidade do conjunto de dados (Bedo et al., 2015).

O objetivo da MPEG-7 é padronizar a representação das imagens armazenadas ou em fluxo contínuo de dados (*data stream*) por meio de um conjunto de extratores de características de baixo valor semântico baseados nas propriedades visuais listadas a seguir com os respectivos extratores:

- Cor: Plano de Cor (*Color Layout*), Estrutura de Cor (*Color Structure*), Cor Escalável (*Scalable Color*), Temperatura de Cor (*Color Temperature*), Descritor de Cor Dominante (*Dominant Color Descriptor*), Correlograma de Cor (*Color Correlogram*) e Grupo de Moldura (*Group-of-Frames*);
- Textura: Histograma de Borda (*Edge Histogram*), Navegação de Textura (*Texture Browsing*) e Textura Homogênea (*Homogeneous Texture*);
- Forma: Contorno da Forma (*Contour Shape*), Espectro da Forma (*Shape Spectrum*), Região da Forma (*Region Shape*).

Neste artigo, estamos interessados nas implementações para o descritor de Cor Dominante (DCD), sendo que sua implementação é descrita pelo padrão MPEG-7 da seguinte maneira (Royo, 2010; Dewi et al., 2017):

- Inicialmente é criado um grupo (*cluster*) no espaço de cores da imagem, representado pelo centroíde de todos os *pixels* pertencentes a imagem;
- O algoritmo segue um conjunto de divisões de *cluster*, gerando assim novos *cluster*, até que algum critério de parada seja alcançado (número máximo de centroídes ou distorção mínima do *cluster*);
- Na etapa de divisão de *clusters*, o *cluster* com a maior distorção é dividido em dois novos *clusters*, tal que, os centroídes dos novos *clusters* são obtidos a partir de um vetor de perturbação ( $\epsilon$ );
- O vetor de perturbação  $\epsilon$  pode ser obtido a partir da direção, de variação máxima do espaço de cores, que é dada pelo autovetor correspondente ao maior valor da matriz de covariância da imagem;
- A distorção  $d_i$  do centroíde  $c_i$  pode ser encontrada usando a equação (1). Nesta equação,  $n$  representa o número de *pixels* no *cluster* de centroíde  $c_i$ ,  $\mathbf{x}(n)$  a cor do *pixel*  $n$  do *cluster* e  $h(n)$  o peso percentual do *pixel*  $n$ .

$$d_i = \sum_n h(n) |\mathbf{x}(n) - c_i|^2, \quad \mathbf{x}(n) \in c_i \quad (1)$$

A primeira implementação considerada neste trabalho é a apresentada em Oliveira et al. (2018), que consiste em uma modificação do algoritmo de agrupamento de dados *k-means*, para atender os passos definidos no padrão MPEG-7 descrito anteriormente. Desta forma, o resultado obtido por esta implementação segue o padrão definido na equação (2).

$$F = \{(c_i, p_i, v_i), s\} \quad (i = 1, 2, 3, \dots, N) \quad (2)$$

Onde,  $N$  é o número de cores dominantes,  $c_i$  o valor da cor dominante  $i$ ,  $p_i$  a porcentagem normalizada  $[0, 1]$  de *pixels* da imagem que correspondem a cor  $c_i$ ,  $v_i$  um valor opcional que representa a variância das cores correspondentes a  $c_i$  e  $s$  a coerência espacial que representa a homogeneidade espacial das cores dominantes da imagem.

A segunda implementação discutida neste trabalho é a definida em Pavão (2016) que propõe um algoritmo denominado Base, que consiste em contar todos os *pixels* de uma imagem:

- Inicialmente é feita uma contagem simples das cores *pixel* por *pixel* na imagem;
- Ordena-se a frequência das cores em forma decrescente;
- Elege as três cores mais dominantes, ou seja as que tem maior frequência na imagem;
- Por fim calcula a porcentagem das cores dominantes em relação as demais e retorna os valores.

A terceira implementação do DCD também foi apresentada em Pavão (2016) e consiste no algoritmo denominado Base com votos, na qual se faz a divisão da imagem em blocos, nos seguintes passos:

- Dividi-se a imagem em blocos de (4X4) pixels
- Para cada bloco chama-se o algoritmo Base;
- Cada bloco contribui com um voto para cada uma das suas três cores dominantes (exceto em casos de empates ou não haver três cores ou mais);
- Se o número de *pixels* de uma cor num dado bloco corresponder a um valor menor que 10% dos *pixels* do bloco, então essa cor é descartada;
- No final avalia-se e soma-se os votos dos vários blocos, elegendo-se como as três cores dominantes na imagem as que obtiverem mais votos.

A última implementação de DCD discutida é a apresentada em Yang et al. (2012) que pode ser definida da seguinte forma:

- Converter a imagem para o espaço de cor HSV (Matiz, Saturação, Valor em inglês - *Hue, Saturation, Value*), pois este espaço de cor representa as cores de forma mais próxima da intuição humana (Yang et al., 2012).

- Quantizar todos os pixels da imagem seguindo as equações (3) e (4), em que  $h, s$  e  $v$  nas referidas equações são os valores do *pixel* no espaço de cores HSV.

$$H = \begin{cases} 0 & \text{se } h \in [0, 20) \cup [315, 360] & S = 0 \text{ se } s < 0.2 \\ 1 & \text{se } h \in [20, 50) & S = 1 \text{ se } 0.2 < s \leq 0.7 \\ 2 & \text{se } h \in [50, 75) & S = 2 \text{ se } 0.7 \leq s \leq 1 \\ 3 & \text{se } h \in [75, 155) & V = 0 \text{ se } v < 0.2 \\ 4 & \text{se } h \in [155, 195) & V = 1 \text{ se } 0.2 < v \leq 0.7 \\ 5 & \text{se } h \in [195, 275) & V = 2 \text{ se } 0.7 \leq v \leq 1 \\ 6 & \text{se } h \in [275, 315) \end{cases} \quad (3)$$

$$C = \begin{cases} 0 & \text{se } v < 0.2 \\ 1 & \text{se } s \leq 0.1, v \in (0.2, 0.5] \\ 2 & \text{se } s \leq 0.1, v \in (0.5, 0.8] \\ 3 & \text{se } s \leq 0.1, v \in (0.8, 0.1] \\ 4 + 9H + 3S + V & \text{caso contrario} \end{cases} \quad (4)$$

- Dividir a imagem em 16 sub-blocos e, para cada sub-bloco encontrar a cor de maior frequência.
- Tomar as cores de maior frequência de cada sub-bloco e suas frequências como as características da imagem.

## 2.2 RECUPERAÇÃO POR CONTEÚDO - SISTEMA CBIR

Para que um sistema CBIR possa realizar uma busca baseada apenas no conteúdo de uma imagem é necessário considerar mais outros dois conceitos além da extração de característica das imagens, sendo esses chamados de funções de distância e operadores de consultas por similaridade. Assim, um sistema CBIR de posse dos vetores de características das imagens utiliza um operador de consulta por similaridade, para recuperar os elementos do conjunto de dados que atendem a um dado critério de similaridade (Pedrosa, 2016; Carvalho et al., 2016), sendo que os elementos recuperados dependem do elemento de busca ( $s_q$ ), chamado de “elemento central de busca”. Existem vários critérios de similaridade (Santos, 2017), sendo que dois são considerados fundamentais: os que expressam as consultas por abrangência e aos  $k$ -vizinhos mais próximos. A partir destes dois critérios é possível determinar qualquer outra variação dos critérios de similaridade. O critério de abrangência é utilizado, tipicamente, em aplicações em que os resultados são enviados para outro algoritmo de mineração de dados ou sumarização (Carvalho et al., 2016).

Neste trabalho, estamos interessados no critério que expressam as consultas aos  $k$ -vizinhos mais próximos (*k-nearest neighbor query- k-NNq*). De forma intuitiva, essa consulta retornar os  $k$  elementos mais próximos ao elemento central de busca (Santos, 2017). De maneira mais formal, seja  $S$  um conjunto de elementos do banco de dados,  $\delta()$  uma função que permita calcular a similaridade entre dois elementos (por exemplo, uma função de distância Euclidiana),  $s_q$  o elemento central de busca e  $k$  o número de elementos que devem ser retornados, uma consulta aos  $k$ -vizinhos mais próximos pode ser descrita pela equação(5)(Santos, 2017).

$$k - NNq(S, k, s_q, \delta) = A = (s_i \in S, \forall s_j \in S - A : \delta(s_q, s_i) \leq \delta(s_q, s_j), |A| \leq k) \quad (5)$$

## 2.3 AVALIAÇÃO DE QUALIDADE EM SISTEMAS CBIR

Os resultados de uma consulta por similaridade são avaliados de acordo com a relevância das imagens na resposta para o usuário. Assim, dado um extrator de característica e uma função de distância, os resultados são ordenados de acordo com a similaridade ao elemento central de consulta. Uma estratégia utilizada para medir a relevância é o gráfico de Precisão X Revocação (P X R). A precisão (equação 6) indica a proporção de imagens relevantes (de mesma classe) que foram recuperadas na resposta, enquanto a revocação (equação 7) mostra a proporção entre as imagens recuperadas em relação ao total de imagens relevantes presentes no conjunto de imagens.

$$\text{Precisão} = \frac{|ImagensRelevantes| \cap |ImagensRecuperadas|}{|ImagensRecuperadas|} \quad (6)$$

$$\text{Revocação} = \frac{|ImagensRelevantes| \cap |ImagensRecuperadas|}{|ImagensRelevantes|} \quad (7)$$

De modo geral, quanto maior for a revocação menor será precisão, isso acontece, porque a quantidade de imagens irrelevantes de um conjunto de dados tende a ser muito maior do que as imagens relevantes de uma mesma classe, reduzindo assim a precisão (Pedrosa, 2016). Em um gráfico de P X R, quanto mais próximo do topo um extrator de característica se mantém, melhor o resultado da consulta com este extrator de característica.

## 3. METODOLOGIA

Nesta seção é apresentada a plataforma de teste definida neste trabalho para comparação das implementações para o descritor de cor dominante em um sistema de recuperação baseado em conteúdo (CBIR). A figura 1 apresenta a arquitetura da plataforma proposta neste trabalho. O módulo de extração de característica foi definido utilizando as implementações da MPEG-7 para o descritor de cor dominante discutidos na Seção 2.1, agora referenciados como: k-means (Oliveira et al., 2018); base sem votos (Pavão, 2016); base com votos (Pavão, 2016) e Yang (Yang et al., 2012). Ao receber um conjunto de dados de imagens e/ou um conjunto de testes, o módulo de recuperação extraí as características e define um conjunto de vetor para cada um dos extratores. O módulo de recuperação recebe os vetores de característica do conjunto de dados, os vetores do conjunto de teste e um parâmetro  $k$  referente a quantidade de imagens que devem ser recuperadas, ou seja, apenas a consulta aos  $k$ -vizinhos mais próximos foi implementada (Seção 2.2). Como saída, o módulo recuperação retorna um conjunto de resultados, o tempo de execução médio para cada consulta no conjunto de testes. O módulo de avaliação recebe os conjuntos de resultados gerados e aplica a estratégia de precisão e revocação (Seção 2.3) para gerar os gráficos de P X R. Por fim, a plataforma retorna o gráfico de P X R e o tempo médio de execução de cada uma das implementações definidas.

## 4. EXPERIMENTOS

Nesta seção apresentamos os resultados experimentais da comparação entre as implementações (descritas na seção 2.1) do descritor de cor dominante para a tarefa de recuperação por conteúdo

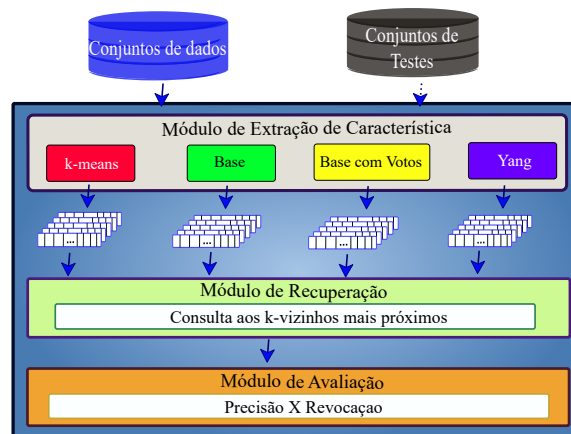


Figura 1- Arquitetura da plataforma de testes proposta.

de imagens utilizando a plataforma de testes apresenta na seção 3. Os experimentos visam avaliar tanto o desempenho das consultas e a qualidade de cada uma das implementações definidas.

#### 4.1 CONJUNTO DE DADOS

Os experimentos realizados foram realizados utilizando três conjuntos de imagens reais e disponíveis publicamente: *Aloi*, *Corel* e *Pasadena-Buildings*. O conjunto de dados *Aloi* é composto por 72.000 imagens de objetos rotacionados a cada cinco graus obtidos do sítio *Amsterdam Library of Object Images*<sup>1</sup>. O conjunto de dados *Corel* (Wang et al., 2001) contém 10 classes e possui 1.000 imagens de diversos conteúdos, como pessoas, praias, flores, construções, carros, cavalos, montanhas e comida. E o conjunto *Pasadena Buildings*<sup>2</sup> contém 125 classes e possui 750 imagens de diversas casas da cidade de Pasadena - Califórnia.

Tanto a plataforma quanto todos os algoritmos foram implementados em linguagem de programação Python 3 com auxílio da biblioteca *Opencv*<sup>3</sup> e executados em um notebook com processador Intel Core i7 7700HQ, 16 GB de RAM, 1TB de HD, utilizando sistema operacional Linux Deepin 15.

#### 4.2 EXPERIMENTO DE DESEMPENHO

Para cada um dos conjuntos de dados testados foram selecionadas aleatoriamente (sem repetições) 100 imagens para compor o conjunto de testes. Assim, foram executadas 100 consultas aos  $k$ -vizinhos mais próximos (cada imagem do conjunto de teste atua como um elemento central de consulta -  $s_q$ ) e o valor do parâmetro  $k$  foi definido de forma empírica à garantir que todas as buscas sempre atinjam revocação igual à 1 (recuperem todas as imagens relevantes para cada conjunto de dados).

A tabela 1 mostra o tempo para a extração dos vetores de características em horas e o tempo médio de busca em segundos. Como pode ser observado na tabela, na extração dos vetores de característica, o algoritmo *k-means* precisou de menos tempo, ao contrário do *Yang*, que foi o método mais demorado, possivelmente pela sua complexidade executando mais comparações

<sup>1</sup>Amsterdam Library of Object Images Homepage: <http://aloi.science.uva.nl/> Acessado em: 30/07/2018

<sup>2</sup>Computational Vision at CALTECH: <http://www.vision.caltech.edu/archive.html> Acessado em: 30/07/2018

<sup>3</sup><https://opencv.org>

e operações aritméticas sobre os dados. Os algoritmos Base sem votos e Base com Votos extraíram os vetores quase com o mesmo custo, indicando que a estratégia de votação não acrescenta um custo ao processo como um todo. Além disso, ambas técnicas tiveram melhores desempenho que o algoritmo do Yang. Em relação ao tempo médio de consulta aos  $k$ -vizinhos mais próximos, as implementações base sem votos e base com votos foram 50 % mais eficientes que o Yang e o k-means. No entanto, para os conjuntos de dados do *Corel* e *Pasadena-Buildings* mesmo com a diferença também em 50 %, os conjuntos apresentam baixa cardinalidade para apresentar uma diferença significativa. Deste modo, pode-se concluir que durante o processo de consulta que dependendo do algoritmo de extração de característica, o processo de recuperação pode ser mais custoso.

Tabela 1- Comparação entre os tempos de extração de características e dos tempos médios de consulta para cada um dos conjuntos de dados testados.

Extrator	Tempo de extração (conjunto de dados) (h)			Tempo Médio de consulta (s)		
	Aloi	Corel	Pasadena-Buildings	Aloi	Corel	Pasadena-Buildings
<b>k-means</b>	<b>0,62</b>	<b>0,03</b>	<b>0,13</b>	0,670	0,008	0,006
<b>Yang</b>	9,30	2,60	3,43	0,660	0,008	0,006
<b>Base Com Votos</b>	2,75	0,13	0,54	0,341	<b>0,004</b>	<b>0,003</b>
<b>Base sem Votos</b>	2,27	0,10	0,46	<b>0,340</b>	<b>0,004</b>	<b>0,003</b>

### 4.3 EXPERIMENTO DE QUALIDADE

A partir dos resultados das consultas aos  $k$ -vizinhos mais próximos da secção 4.2, foram gerados os gráficos de precisão e revocação (P X R) de cada um dos algoritmos sobre os conjuntos de dados.

A Figura 2 mostra os gráficos de precisão e revocação gerados. Como pode ser visto, o algoritmo k-means mostrou-se o menos eficaz em todos os conjuntos de dados, enquanto que o Yang foi o mais eficaz para os conjuntos *Aloi* e *Pasadena-Buildings* (Figura 2 (a) e (c)) e, o algoritmo Base com Votos o mais eficaz para o conjunto *Corel* (Figura 2 (b)). Os algoritmos Base com votos e Base sem votos obtiveram resultados muito próximos, sendo que para o conjunto *Aloi* o algoritmo Base sem votos foi mais eficaz que o Base com votos, o que mostra que a estratégia de votos não garante melhores resultados.

## 5. CONCLUSÕES

Imagens digitais *se tornaram uma parte importante do mundo moderno*, sendo utilizadas tanto em tarefas voltadas ao entretenimento quanto em tarefas do setor produtivo. Contudo, para se utilizar este tipo de dado em sistema de recuperação de imagens por conteúdo, é necessário um processo de representação do conteúdo das imagens por meio de algoritmos chamados de extratores de características. Neste trabalho avaliamos tanto a qualidade quanto a eficiência de quatro implementações do descritor de cor dominante definido no padrão MPEG-7. No âmbito de precisão e revocação (qualidade), nenhum algoritmo se mostrou superior em todos os conjuntos de imagens testados, entretanto, para os conjuntos *Aloi* e *Pasadena-Buildings* o algoritmo Yang mostrou-se superior e, para o conjunto *Corel* foi o algoritmo Base com Votos. Contudo, os testes de desempenho mostram que a escolha de uma das implementações impactam tanto no tempo de representação das imagens quanto no tempo de consulta em um sistema



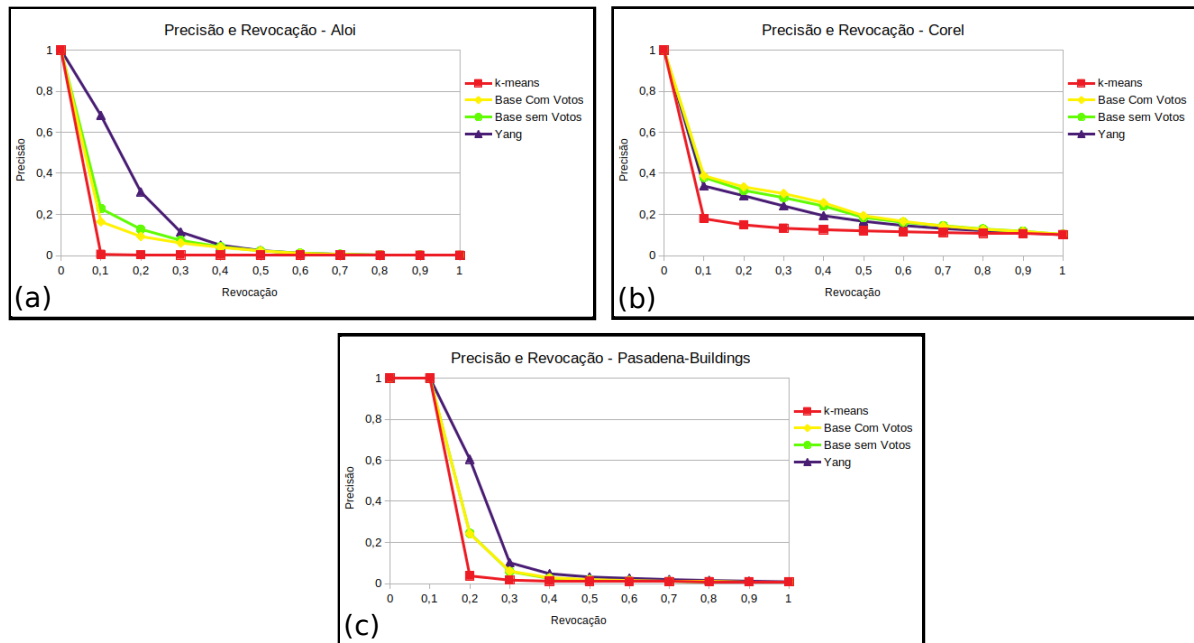


Figura 2- Gráfico de precisão e revocação

de recuperação baseado em conteúdo de imagens. Logo, a escolha de uma implementação deve se basear no conjunto de imagens.

Como trabalhos futuros, consideramos realizar testes em outros conjuntos de imagens comparando a qualidade dos resultados com outros descritores de cor definidos na MPEG-7.

## 6. Agradecimentos

Os autores agradecem ao Instituto Federal de Educação Ciência e Tecnologia do Norte de Minas Gerais - *Campus Montes Claros* e a FAPEMIG.

## REFERÊNCIAS

- Bedo, M. V. N., Blanco, G., Oliveira, W. D., Cazzolato, M. T., Costa, A. F., Jr., J. F. R., Traina, A. J. M., and Jr., C. T. (2015). Techniques for effective and efficient fire detection from social media images. In *Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 34–45. INSTICC, SciTePress.
- Carvalho, L. O., Santos, L. F. D., Oliveira, W. D., Traina, A. J. M., and Traina, C. (2016). Efficient self-similarity range wide-joins fostering near-duplicate image detection in emergency scenarios. In *ICEIS*, pages 81–45.
- Delmondes, P. H. M., Bergamasco, L. C. C., and dos Santos Nunes, F. d. L. (2015). Recuperação de modelos médicos por conteúdo usando extrator local de área e distância. *Revista de Informática Teórica e Aplicada*, 22(2):10–30.
- Dewi, A. F., Arnia, F., and Muharar, R. (2017). Effectiveness of mpeg-7 color features in clothing retrieval. *Bulletin of Electrical Engineering and Informatics*, 6(2):166–173.
- Kimura, P. d. A. d. S. et al. (2011). Um descritor baseado em análise local de cor para busca de imagens em grandes coleções.

- Martinez, J. M., Koenen, R., and Pereira, F. (2002). Mpeg-7: the generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87.
- M.J. Swain, D. B. (1991). Color indexing. *International journal of computer vision*, 7(1):11–32.
- Oliveira, A. S., Carvalho, M. L. S., Neto, J. W. V. M., and Castro, A. F. (2018). Comparação entre os algoritmos k-means e dynamic cluster em imagens digitais. *Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574)*, 2(1).
- Pavão, R. N. (2016). *Eyecolors: identificação da cor dominante em fotografias*. PhD thesis.
- Pedrosa, G. V. (2016). *Caracterização e recuperação de imagens usando dicionários visuais semanticamente enriquecidos*. PhD thesis, Universidade de São Paulo.
- Royo, C. V. (2010). Image-based query by example using mpeg-7 visual descriptors.
- Santos, L. F. D. (2017). *Similaridade em big data*. PhD thesis, Universidade de São Paulo.
- Silva, G. O. d. et al. (2016). Realimentação de relevância em buscas de imagem usando programação genética.
- Wang, J. Z., Li, J., and Wiederhold, G. (2001). Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE TPAMI*, 23(9):947–963.
- Yang, L. F., Huang, X. L., Lv, R., and Lv, H. (2012). An effective similarity measurement algorithm for dominant color feature matching in image retrieval. In *Applied Mechanics and Materials*, volume 182, pages 1169–1173. Trans Tech Publ.

## EVALUATING DIFFERENT IMPLEMENTATIONS OF THE DOMINANT COLOR DESCRIPTOR

**Abstract.** *This paper proposes a framework to test implementations of four versions of the dominant color descriptor feature extractor defined by the MPEG-7 standard. The proposed framework simulates a content-based image retrieval task and evaluates both the answer quality and the average time to execute the queries. Experimental evaluations were conducted using three real datasets and the results pinpoints that the choice of the right implementation of the feature extractor impacts both in the extraction of the feature vectors and the querying time.*

**Keywords:** *Content-Based Image Retrieval, Feature Extractors, Dominant color descriptor*