UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DESCRITORES DE IMAGENS BASEADOS EM ASSINATURA TEXTUAL

Joyce Miranda dos Santos

Novembro de 2016

Manaus - AM

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DESCRITORES DE IMAGENS BASEADOS EM ASSINATURA TEXTUAL

Joyce Miranda dos Santos

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Computação - IComp, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Doutora em Informática.

Orientador: Prof. Edleno Silva de Moura, D.Sc.

Novembro de 2016

Manaus - AM

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Santos, Joyce Miranda dos
S237d Descritores de imagens baseados em assinatura textual / Joyce Miranda dos Santos. 2016
66 f.: il. color; 31 cm.

Orientador: Edleno Silva de Moura
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Descritor de imagem. 2. Palavra visual. 3. Assinatura textual. 4. Recuperação por conteúdo.. I. Moura, Edleno Silva de II. Universidade Federal do Amazonas III. Título

DESCRITORES DE IMAGENS BASEADOS EM ASSINATURA TEXTUAL

Joyce Miranda dos Santos

TESE SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO DO INSTITUTO DE COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DO AMAZONAS COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE DOUTORA EM INFORMÁTICA.

Aprovada por:

Prof. Edleno Silva de Moura, D.Sc.

Prof. Altigran Soares da Silva, D.Sc.

Prof. Marco Antônio Pinheiro de Cristo, D.Sc.

Prof. Ricardo da Silva Torres, D.Sc.

Prof. Thierson Couto Rosa, D.Sc.

NOVEMBRO DE 2016

MANAUS, AM – BRASIL

A minha mãe Cássia e a minha irmã Jéssica que me deram o suporte emocional necessário. Em especial, dedico este trabalho ao meu pai Jazon (in memoriam), meu amigo, apoiador, de quem sinto uma saudade imensurável.

Agradecimentos

A Deus, pelo dom da vida.

Aos meus pais, por todo apoio.

Ao professor Edleno Moura, por sua orientação.

A todos que, de alguma forma, estiveram comigo e me ajudaram a superar os desafios durante toda essa caminhada.

"O rio não corta a rocha por causa de sua força, mas sim por causa de sua persistência."

(Jim Watkins)

Resumo

A técnica de representar imagens por meio de um conjunto de palavras visuais, conhecida como *bag of visual words*, tem sido aplicada com sucesso em tarefas de recuperação de imagens baseada em conteúdo. Neste trabalho, é proposto o paradigma *Signature-based Bag of Visual Words* (S-BoVW), uma definição formal para métodos que descrevem imagens por meio de palavras visuais, sem que para isso seja necessária a construção prévia de um vocabulário visual. Métodos baseados nesse paradigma dispensam o uso de algoritmos de agrupamento, o que permite reduzir de forma significativa o custo associado à etapa de descrição das imagens. A codificação e a combinação de características, como cor e textura, foram investigadas neste trabalho com o intuito de definir novos descritores de imagens baseados no paradigma S-BoVW. Experimentos foram realizados com o objetivo de propor formas eficazes e eficientes de aplicar o conceito proposto pelo paradigma S-BoVW. Os resultados obtidos a partir deste trabalho demonstram que a escolha adequada da técnica de processamento de consulta e da função de cálculo de similaridade garante a obtenção de um desempenho otimizado por parte dos métodos S-BoVW, como também assegura a competitividade destes em relação aos *baselines* em diversos cenários.

PALAVRAS-CHAVE: descritor de imagem, palavra visual, assinatura textual, recuperação por conteúdo.

Abstract

The technique of representing images by a set of visual words, known as *bag of visual words*, has been successfully applied to content-based image retrieval. In this work, it is proposed the paradigm *Signature-based Bag of Visual Words* (S-BoVW), a formal definition for methods that describe images by visual words, without the previous construction of a visual vocabulary. Methods based on this paradigm not require the use of clustering algorithms, which allows to reduce the cost associated with the images description step. The coding and combination of features such as color and texture were explored in this work in order to define new descriptors of images based on the S-BoVW paradigm. Experiments were carried out in order to propose effective and efficient ways to apply the concept proposed by S-BoVW paradigm in the definition of new methods of content-based image retrieval. The results obtained demonstrate that proper choice of query processing technique and the similarity function ensures obtaining optimized performance by S-BoVW methods and also ensures their competitiveness compared to baselines in many scenarios.

KEY-WORDS: image descriptor, visual word, textual signature, content-based image retrieval.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O crescente avanço tecnológico tem possibilitado o acesso massivo da população a dispositivos como *smartphones* e modernas câmeras digitais, capazes de capturar e armazenar de forma simples e rápida um grande volume de imagens. Esse cenário requer o desenvolvimento de métodos que sejam cada vez mais eficazes e eficientes na tarefa de indexar e buscar imagens em grandes coleções. De uma forma geral, os métodos usados na recuperação de imagens podem ser classificados como *Text-Based Image Retrieval* (TBIR) ou *Content-Based Image Retrieval* (CBIR).

A abordagem TBIR permite a utilização de técnicas tradicionais de busca textual para recuperar imagens. Para aplicar essa abordagem, é necessário que exista alguma anotação textual associada às imagens da coleção. Normalmente, essa anotação é feita de forma manual, estando sujeita à interpretação e à subjetividade da pessoa que descreveu a imagem. Apesar da abordagem TBIR ser bastante utilizada, existem situações nas quais fazer uma anotação textual da coleção torna-se inviável. Um exemplo disso acontece quando o tamanho da coleção é muito grande, tornando a anotação da coleção impraticável devido ao tempo necessário para realizar essa tarefa.

Na abordagem CBIR, a busca é feita a partir de uma imagem de consulta, e não a partir de um texto como na abordagem TBIR. Sistemas CBIR usam como agente principal descritores que são responsáveis por representar o conteúdo visual das imagens e estabelecer um critério de similaridade entre elas. Esse conteúdo visual é representado por características de baixo nível como cor, forma e textura. Uma limitação encontrada na área de CBIR é a falta de semântica associada às imagens. Isso porque os usuários, ao buscarem por uma imagem, estão interessados não somente em características visuais

semelhantes, mas também no significado associado à imagem, que dificilmente é obtido por meio de características de baixo nível.

Uma estratégia, conhecida como *bag of visual words* [39, 28, 49], tem sido bastante explorada na literatura com o intuito de melhorar a eficiência obtida por métodos CBIR. Essa estratégia busca representar imagens por meio de um conjunto de palavras visuais, tornando possível o uso de técnicas eficientes de recuperação textual em tarefas como indexação e busca de imagens. Essas palavras são chamadas visuais pois não pertencem a uma linguagem natural, como por exemplo inglês ou português. Na verdade, essas palavras são termos gerados artificialmente por processos que buscam a representação de padrões que ocorrem repetidamente em uma coleção de imagens. A maioria dos métodos que aplica o conceito de *bag of visual words*, adota um paradigma baseado em algoritmos de agrupamento para a construção de um vocabulário visual cujas palavras serão utilizadas para descrever as imagens. Neste trabalho, esse paradigma é referenciado como *Cluster-based bag of visual words* (C-BoVW).

No paradigma C-BoVW, descritores locais das imagens de uma coleção de referência são agrupados de acordo com alguma função de similaridade. Cada grupo (*cluster*) gerado representa uma palavra visual. O conjunto de palavras visuais é chamado de vocabulário visual. O processo de descrição de imagens seguido pelo paradigma C-BoVW consiste em associar, de acordo com algum critério de similaridade, cada ponto de interesse presente em uma imagem a um grupo, que representa uma palavra visual no vocabulário gerado. Dessa forma, cada imagem será representada por um conjunto de palavras visuais.

No trabalho apresentado por Vidal et al. [46], é definido o método *Sorted Dominant Local Color* (SDLC), no qual a geração de palavras visuais é feita apenas com base na informação das cores mais frequentes de cada bloco da imagem. Uma vantagem muito importante do SDLC em relação aos métodos C-BoVW está relacionada ao baixo custo associado à etapa de descrição das imagens. Isso é devido, em grande parte, à eliminação da etapa de geração do vocabulário visual, indispensável aos métodos C-BoVW. Experimentos realizados com o SDLC demonstraram que esta abordagem de representação também apresenta resultados competitivos em termos de qualidade em relação a outros métodos CBIR.

A estratégia de geração de palavras visuais adotada pelo SDLC, foi o ponto de partida para a definição do paradigma *Signature-based bag of visual words* (S-BoVW), proposto

neste trabalho. O S-BoVW, é apresentado como uma nova categoria de métodos, uma vez que possibilita que outras funções de mapeamento entre blocos e assinaturas textuais possam ser adotadas para derivar descritores diferentes do SDLC. O paradigma de geração de palavras visuais baseada em assinaturas textuais é o objeto de estudo deste trabalho.

1.1 Motivação e Justificativa

A recuperação de imagens baseada em conteúdo por meio de palavras visuais ainda é um problema em aberto para o qual não existe uma solução ideal em termos de eficiência e eficácia. Aprofundar o estudo sobre o paradigma baseado em assinaturas textuais se apresenta como um caminho promissor, uma vez que este além de reduzir o custo associado à etapa de descrição das imagens, também alcança resultados competitivos em relação a outros métodos CBIR em diversos cenários.

1.2 Objetivos

O objetivo geral deste trabalho consiste em propor formas eficazes e eficientes de aplicar o paradigma S-BoVW em tarefas de recuperação de imagens baseada em conteúdo.

1.2.1 Objetivos Específicos

- Formalizar a definição do paradigma S-BoVW;
- Propor alternativas para a geração de assinaturas textuais, explorando a codificação e a combinação de características de baixo nível das imagens, como cor e textura;
- Analisar o impacto da configuração de parâmetros que podem ocasionar a variação da eficácia em métodos baseados no paradigma S-BoVW.
- Verificar o desempenho de métodos considerados eficientes para o processamento de busca textual no cenário de representação proposto pelo paradigma S-BoVW.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte forma. No Capítulo 2, são apresentados conceitos e fundamentos considerados relevantes para o entendimento deste trabalho. O Capítulo 3 apresenta soluções propostas na literatura que estão relacionadas aos objetivos deste trabalho. No Capítulo 4, uma generalização do paradigma de recuperação de imagens baseada em assinatura textual é apresentada como uma nova classe de métodos chamada S-BoVW (*Signature-based Bag of Visual Words)*. No Capítulo 5, são apresentados os resultados do estudo do impacto de diferentes parâmetros e funções de similaridade nos métodos S-BoVW em diferentes coleções. O Capítulo 6 apresenta uma análise comparativa de desempenho entre métodos S-BoVW e os *baselines* selecionados. No Capítulo 7 é apresentada uma análise de desempenho dos métodos de recuperação textual no cenário S-BoVW. Por fim, o Capítulo 8 apresenta as considerações finais do trabalho e algumas direções para a realização de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, são apresentados conceitos fundamentais considerados importantes para o entendimento deste trabalho.

2.1 Descritores de Imagem

O descritor de imagem é um dos componentes mais importantes em um sistema CBIR. Sua função é a de quantificar quão similar são duas imagens. No trabalho apresentado por Torres e Falcão [43], um descritor é definido como uma tupla $(\varepsilon_d, \delta_d)$, onde:

- ε_d: é a função responsável por caracterizar o conteúdo visual de uma imagem e codificá-lo-lo em um vetor de características.
- δ_d: é a função responsável por comparar dois vetores de características. Dados dois vetores, essa função calcula um score que define a similaridade ou a distância entre duas imagens.

A Figura 2.1 mostra o fluxo típico de uma sistema CBIR. Um processo de extração de características é aplicado sobre cada imagem de uma coleção, por meio da função ε_d. O resultado desse processo é a obtenção de vetores que codificam características visuais, tais como: cor, forma e textura. O tamanho do vetor depende da quantidade de características usada para representar as imagens.

Na submissão de uma consulta, o mesmo processo de extração é realizado sobre a imagem e um vetor de característica também é obtido. A partir desse momento, esse vetor é comparado com os vetores de características que foram gerados para as imagens

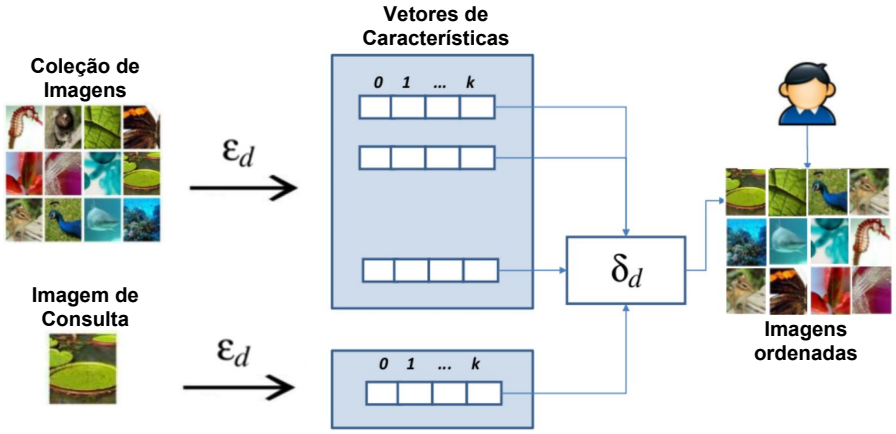

Figura 2.1: Fluxo de uma solução CBIR típica.

da coleção, por meio da função δ_d. Como resultado desse processo de comparação, cada imagem da coleção é colocada em uma lista ordenada (*ranking*) de acordo com seu grau de similaridade com a imagem de consulta.

Dentre as propriedades desejáveis para um descritor estão: insensibilidade a ruídos, invariância a algumas classes de transformação (rotação e translação), geração de vetores de características compactos que exijam pouco espaço de armazenamento e utilização de uma função de extração computacionalmente eficiente [44].

É importante ressaltar que a eficácia de um descritor não depende somente do algoritmo de extração de características, mas depende também da função de similaridade usada. O uso adequado de medidas de similaridade ajuda a melhorar o resultado das consultas.

Escolher o descritor mais adequado para uma determinada aplicação é crucial para o sucesso de um sistema CBIR. Por isso, é importante que sejam conduzidos experimentos comparativos utilizando diferentes descritores com o intuito de utilizar aquele que alcança o melhor desempenho. No trabalho feito por Penatti e Torres [32], uma ferramenta foi desenvolvida com o objetivo de facilitar a comparação experimental entre descritores de imagens. Essa ferramenta, conhecida como Eva, tem como vantagens: a integração entre as etapas mais importantes de um processo de recuperação de imagens baseada em conteúdo, a facilidade da execução de experimentos comparativos e a apresentação dos resultados obtidos pelos descritores em termos de eficiência e de eficácia.

Definir quais tipos de evidência serão codificadas nos vetores de características depende diretamente do contexto onde o descritor será aplicado. Cada evidência representa

aspectos específicos da imagem que devem ser considerados de acordo com a necessidade da aplicação. Em geral, as evidências mais exploradas por métodos CBIR são: cor, forma e textura.

2.1.1 Descritores de Cor

A cor é uma das principais evidências utilizadas em descritores CBIR. Isso é devido à simplicidade e ao baixo custo computacional associado à extração dessa evidência. Ao escolher codificar essa característica, o foco deve estar na representação da distribuição das cores de uma imagem de forma a recuperar imagens que possuam uma composição de cor similar, mesmo que elas pertençam a contextos diferentes.

De uma forma geral, os descritores de cor podem ser classificadas em: (i) globais: descrevem a distribuição de cores das imagens como um todo, desprezando a sua distribuição espacial, (ii) baseadas em particionamento: decompõem espacialmente as imagens utilizando uma estratégia de particionamento simples e comum a toda imagem, sem levar em consideração o seu conteúdo visual, e (iii) regionais: utilizam técnicas automáticas de segmentação para decompor as imagens de acordo com o seu conteúdo visual.

Uma imagem digital pode ser representada como uma matriz $n \; x \; m$, em que cada elemento da matriz (*pixel*) está associado a uma intensidade que representa uma cor. A escolha do espaço de cor pelo qual as imagens serão representadas, analisadas e comparadas é o primeiro passo na definição de um descritor baseado em cor [44]. O modelo de espaço de cor mais conhecido é o RGB (*red*, *green*, *blue*). No espaço RGB, a cor é representada por três valores, um para cada canal de cor.

Ao codificar cores é comum usar 8 bits para representar um canal de cor. Isso equivale a $2^8 = 256$ níveis de cor para cada canal. Considerando o RGB que possui três canais de cores, isso resultaria em aproximadamente 17 milhões ($256[R] \times 256[G] \times 256[B]$) de cores distintas. Considerando, uma imagem com resolução de 300 x 300, 90.000 *pixels* deveriam ser considerados em uma análise comparativa entre duas imagens, *pixel* a *pixel*. Esses valores, em termos de quantidade de cores e dimensão espacial, tornaria inviável o processamento de sistemas de recuperação de imagens.

Um sistema de recuperação de imagens necessita de uma representação compacta da distribuição de cores [40]. Para isso ser possível, é feito um processo de quantização que consiste em reduzir a quantidade de bits usada por *pixel*. Na prática, normalmente, são

utilizados no máximo 8 bits por *pixel* que equivalem a 256 cores diferentes. Neste caso, torna-se necessário definir um índice com as 256 cores mais significativas e determinar a equivalência entre as cores da imagem e as cores do índice.

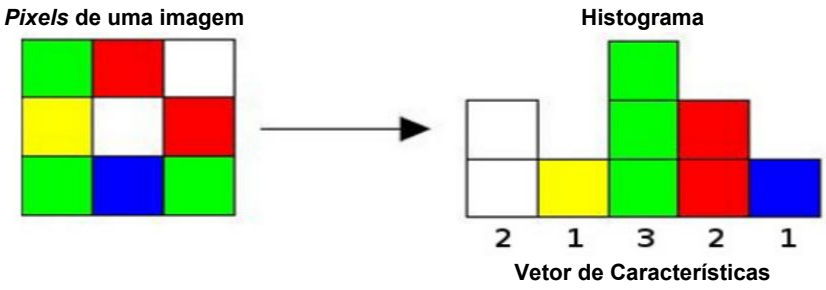

Figura 2.2: Exemplo da construção de um histograma.

A implementação mais comum de um descritor de cor consiste na construção de um histograma que representam a distribuição de cores de uma imagem. Na Figura 2.2 é ilustrada a construção de um histograma com base em um conjunto de nove *pixel*s. O processo se inicia com a formação de pilhas, uma para cada cor presente da imagem. Em seguida, é feito um somatório das ocorrências de uma cor, incrementando assim, a pilha correspondente.

Uma forma simples de montar o vetor de características a partir de um histograma é inserindo sequencialmente, em cada índice do vetor, a frequência de cada cor. Com os vetores de características gerados, é possível aplicar métricas para verificar a similaridade entre dois vetores.

2.1.2 Descritores de Textura

Existem contextos em que apenas as características de cor ou sua intensidade são insuficientes para realizar a descrição das imagens. Os descritores de textura são usados quando existe nas imagens um padrão visual com propriedades de homogeneidade que são indiferentes às variações de cor. Esses descritores buscam representar aspectos da superfície de um objeto, analisando para isso o relacionamento de vizinhança entre os *pixels* da imagem. Isso permite a representação de atributos como: rugosidade, contraste, aspereza e semelhança com linhas.

2.1.3 Descritores de Forma

A informação semântica de uma imagem normalmente está associada a elementos e a objetos que estão presentes nela. Por exemplo, se uma pessoa é requisitada para escolher imagens parecidas com uma imagem que possui forte semântica associada a objetos, esta pessoa provavelmente irá ignorar ou colocar em segundo plano características como cor e textura.

Descritores de forma usam técnicas que consideram a descrição total da borda de objetos (abordagem baseada em contorno) ou a descrição das características morfológicas das regiões presentes na imagem (abordagem baseada em regiões). Esses descritores são usados quando é necessário executar pesquisas com base no perfil e na estrutura física de um objeto. Um contexto bastante comum para esse tipo de aplicação é a busca por informações em bancos de dados de medicina, nos quais as imagens têm características de cor e textura muito semelhantes. Outras aplicações possíveis são: o reconhecimento de caracteres alfanuméricos em documentos, rastreamento de objetos em vídeos e reconhecimento de pessoas em sistemas de segurança.

Em coleções nas quais o conteúdo é conhecido e controlado é possível ajustar mais facilmente os parâmetros necessários para a deteção de formas. Em bases heterogêneas, como é o caso da Web, ajustar parâmetros que satisfaçam de maneira razoável as possíveis categorias de busca é uma tarefa impraticável. Outro problema associado aos descritores de forma é o alto custo computacional exigido para o seu processamento.

2.2 *Cluster-based Bag of Visual Words*

A fim de lidar com o desafio da busca eficiente em grandes coleções de imagens, uma série de métodos tem adotado a estratégia de modelar imagens como documentos constituídos por palavras visuais, conhecidas como *visual words* [39, 28, 49]. Comumente, esses métodos adotam algoritmos de agrupamento (*clustering*) para gerar as palavras visuais que serão usadas para descrever as imagens de uma coleção. Essa abordagem, conhecida como *Cluster-based Bag of Visual Words* (C-BoVW), permite que uma imagem seja indexada de forma similar a um documento textual, sendo possível assim aplicar técnicas usadas na recuperação textual com o objetivo de aumentar eficiência.

A origem dessa abordagem vem do modelo *Bag of Words* (BoW) da área de

recuperação textual. No modelo BoW, um documento é representado como um histograma formado pela quantidade de palavras existentes em seu conteúdo. Na Figura 2.3, é apresentada uma ilustração de como o método BoW representa os documentos. De uma forma geral, são contadas todas as palavras de um dicionário que aparecem no documento. Esse dicionário deve possuir um único termo que simboliza um conjunto de sinônimos. O vetor de termos que representa um documento é um vetor esparso em que cada elemento é um termo do dicionário e o valor desse elemento é o número de vezes que o termo aparece no documento. O vetor de termos é a representação BoW, que é chamada de *bag* (saco) pois toda a informação de ordem das palavras no documento é perdida.

Figura 2.3: Representação de imagens gerada pelo método BoW.

Na abordagem C-BoVW, o dicionário, conhecido também como vocabulário visual ou *codebook*, é construído a partir do agrupamento de pontos de interesse detectados em imagens presentes em uma coleção de referência. O processo de agrupamento permite gerar um vocabulário discreto a partir de milhões (ou bilhões) de pontos de interesse.

De uma forma geral, a geração da representação C-BoVW é feita a partir de três etapas principais: (i) deteção e representação de pontos de interesse, (ii) geração do vocabulário e (iii) descrição das imagens. Na Figura 2.4, é apresentada uma ilustração do processo de representação de imagens adotado por métodos C-BoVW.

2.2.1 Detecção e representação de pontos de interesse

A detecção de pontos de interesse é a etapa responsável por decidir quais áreas da imagem são relevantes para representá-la. A saída dessa etapa consiste em um conjunto de pontos de interesse que especificam coordenadas relativas a escalas e orientações. Basicamente, existem três estratégias utilizadas para detectar pontos de interesse [30], são

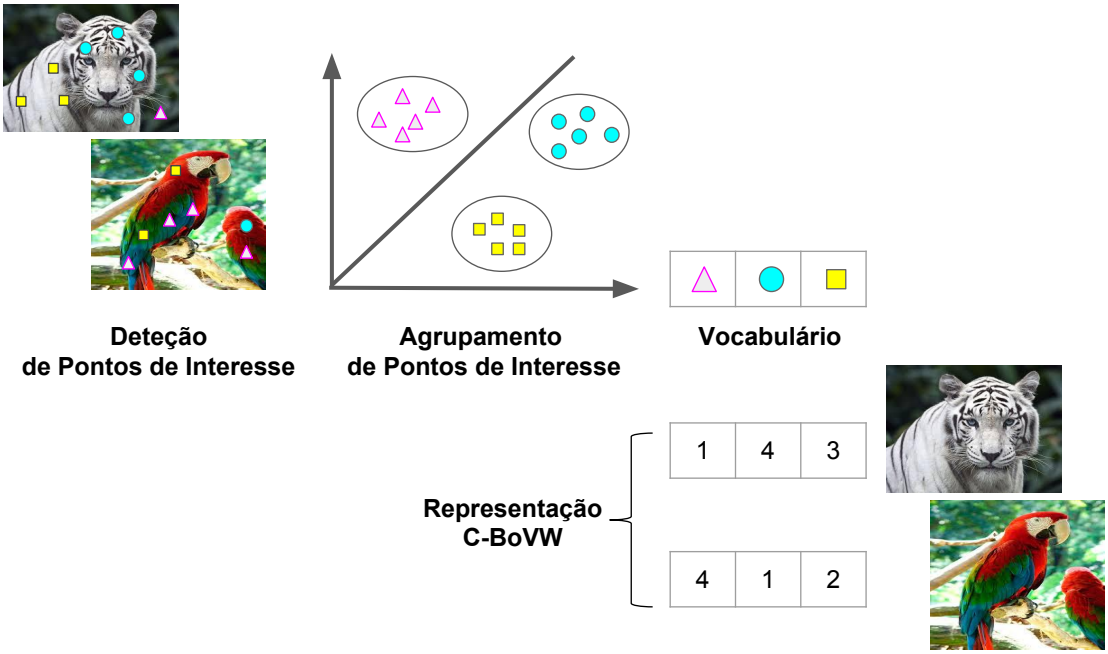

Figura 2.4: Representação de imagens gerada por métodos C-BoVW.

elas: (i) operadores de ponto de interesse, (ii) saliência visual e (iii) escolha randômica ou determinística.

A estratégia que usa operadores de ponto de interesse tem como objetivo identificar pontos estáveis na imagem, que sejam invariantes a transformações. Para isso, é feita uma busca por pontos que possam ser atribuídos repetidamente a diferentes visões de um mesmo objeto. Tipicamente esses pontos são detectados usando representações do espaço de escala. Um espaço de escala consiste em representar uma imagem sob múltiplas resoluções. As respostas para um filtro proposto para detectar esses pontos estão localizadas em um espaço tridimensional de coordenadas (x, y, s), em que (x, y) é a localização do *pixel* e (s) é a escala. Cada detector enfatiza diferentes aspectos de invariância, podendo resultar em pontos de interesse bem distintos. Dentre os detectores mais populares estão: Laplacian of Gaussian (LoG), Difference of Gaussian (DoG), Harris Laplace, Hessian Laplace, Harris Affine, Hessian Affine e Maximally Stable Extremal Regions (MSER) [20, 27].

Alguns detectores são baseados em modelos desenvolvidos a partir do sistema de atenção visual humana. Esses métodos se concentram em encontrar locais na imagem que sejam visualmente salientes. Nesse caso, os métodos são avaliados por sua capacidade de prever a fixação do olho humano registrado por meio de equipamentos que conseguem rastrear a visão. Um estudo feito por Frintrop et al. [15] apresenta vários

métodos baseados na atenção visual humana.

Por fim, existe uma linha de pesquisa que defende que mais importante do que escolher qual detector usar é se deve usar ou não um detector. Resultados obtidos por operadores de pontos de interesse mostram que a detecção dos pontos deixa cerca de metade da imagem sem ser representada, deixando de lado regiões que podem ser discriminativas para os descritores de imagem. Por isso, alguns trabalhos buscam caracterizar a extração de pontos de interesse como um problema de amostragem, sugerindo usar estruturas de grid ou pirâmide, e até mesmo amostragem randômica.

Uma vez detectados os pontos de interesse, torna-se necessário definir como esses pontos serão representados. A forma mais comum é codificando informações sobre a vizinhança dos *pixels* relativos aos pontos de interesse detectados. De uma forma geral, descritores locais são usados com esse propósito. Espera-se que a descrição obtida possua propriedades de invariância. O descritor deve ser robusto com relação a variações da imagem, tais como distorções, mudanças de escala e mudanças de iluminação. Dentre os descritores locais mais populares estão: *Scale Invariant Feature Transform* (SIFT) [25], *Speeded Up Robust Features* (SURF) [4] e *Local Binary Patter* (LBP) [31]. Esses descritores consideram apenas a composição dos níveis de cinza da imagem.

No descritor SIFT, os pontos de interesse são obtidos pelo detector *Difference of Gaussians* (DoG) [27]. Uma região circular em torno de cada ponto de interesse é dividida em um grid de 16 células (4x4). Um histograma de orientações de gradientes é calculado para cada célula. Uma suavização do histograma é feita com o objetivo de evitar mundanças bruscas de orientação. O tamanho do histograma é reduzido para 8 *bins* de forma a limitar o tamanho do descritor. O resultado é um vetor de dimensão 128 (4x4x8) para cada ponto de interesse detectado.

O descritor SURF produz pontos de interesse semelhantes aos produzidos pelo SIFT. Os pontos de interesse são obtidos pelo detector Hessian-Laplace [27], usando aproximações eficientes. Sua versão original é considerada muito mais rápida e robusta com relação a diferentes transformações da imagem quando comparada ao SIFT.

O LBP é um descritor de textura baseado em uma codificação binária de valores de intensidade limiarizado por meio da comparação de cada pixel com seus vizinhos. Esse descritor é invariante a transformações de valores de cinza monotônicos, mas não é invariante a tranformações de rotação.

2.2.2 Geração do vocabulário

Métodos C-BoVW geram o vocabulário visual por meio de algoritmos de agrupamento que recebem como entrada descritores locais referentes a pontos de interesse detectados. Cada centróide dos grupos gerados durante o processo de agrupamento é tratado como uma palavra visual do vocabulário. O conjunto de centróides representa o vocabulário visual. A geração do vocabulário é um processo *offline* e pode se tornar muito lento devido à quantidade de descritors locais que podem chegar a casa dos bilhões dependendo da coleção. O *K-means* [16] é o algoritmo de agrupamento mais comumente usado para a geração do vocabulário.

Na recuperação textual o tamanho do vocabulário é pré-definido pela coleção. No paradigma C-BoVW, o tamanho do vocabulário é definido de acordo com parâmetros usados na execução do algoritmo de agrupamento. Ainda não existe consenso relacionado ao tamanho ideal do vocabulário. Um vocabulário pequeno pode perder seu poder discriminativo uma vez que dois descritores locais distintos podem ser atribuídos ao mesmo grupo, mesmo que não sejam semelhantes entre si. Por outro lado, um vocabulário grande é menos genérico, menos tolerante a ruídos e gera sobrecarga de processamento [20].

2.2.3 Descrição das imagens

Na etapa de descrião das imagens é feita a quantização de seus descritores locais em relação às palavras do vocabulário. Isso é feito por meio da verificação de proximidade entre os descritores locais e os centróides do vocabulário. A forma mais simples de se fazer isso, é aplicando estratégias de busca por vizinhos mais próximos (ex. Knn). Nesse contexto, duas abordagens podem ser aplicadas: *hard assignment* e *soft assignment*. Na abordagem *hard assignment*, apenas o vizinho ligeiramente mais próximo é selecionado para representar um descritor local, enquanto que na abordagem *soft assignment*, mais de um vizinho pode ser levado em consideração para representar o descritor local [18, 20].

Dessa forma, cada imagem da coleção terá seu conteúdo representado por um conjunto de palavras visuais. Com essa representação, as palavras visuais podem ser vistas como termos de um índice, tornando possível a aplicação de qualquer modelo de recuperação textual para buscar imagens.

O Modelo de Espaço Vetorial [37] é o modelo de recuperação textual mais popular. Isso se estende também à recuperação de imagens baseada em palavras visuais. Esse

modelo representa os documentos como vetores cujas dimensões são os termos do índice. A similaridade entre dois documentos é computada com a medida do ângulo entre os vetores (distância do coseno) ou com a distância entre os vetores (tipicamente L1 ou L2). Nesse modelo, o conteúdo dos índices do vetor representa a importância do termo na descrição do documento. Isso pode ser representado pela presença ou não presença do termo no documento, pela frequência do termo ou por outro peso, com a seguinte suposição: quanto maior for o peso de um termo, melhor esse termo estará descrevendo o documento.

A atribuição de pesos aos termos do vetor é uma estratégia usada para penalizar termos considerados muito comuns e enfatizar termos que são mais exclusivos na coleção. O peso w_{i_j} de um termo t_i em um documento d_j está normalmente dividido em três partes, sendo elas: (i) um peso local (l_{i_j}), (ii) um peso global (g_i) e um fator de normalização (n_j), de forma que $w_{i_j} = l_{i_j} \times g_i \times n_j$.

- Peso local (l_{i_j}): reflete o peso do termo dentro do documento. Ele pode ser usado com o objetivo de enfatizar termos com alta frequência, limitar a influência da frequência do termo ou normalizar a frequência do termo de acordo com o tamanho do documento.

- Peso global (g_i): enfatiza a importância do termo na coleção. A suposição é que quanto maior a quantidade de documentos onde o termo ocorre, menos discriminativo esse termo será. Por outro lado, se um termo ocorre em poucos documentos, ele deverá ser considerado um descritor para o conteúdo desses documentos. *Inverse Document Frequency* (IDF) é o peso global clássico.

- Fator de normalização (n_j): depende apenas do documento e tem como objetivo manter as distâncias entre consulta e documentos em uma escala similar para que seja possível compará-las e gerar um *ranking*. O fator de normalização utilizado deve ser consistente com a medida de distância utilizada.

2.3 Processamento de consulta textual

Máquinas de busca atuais realizam a recuperação textual a partir de uma estrutura denominada índice invertido [3]. O índice invertido é formado por listas geradas para cada

termo presente na coleção de documentos. Essas listas são estruturas responsáveis por identificar os documentos nos quais um termo ocorre, assim como o impacto do termo em cada documento. De uma forma geral, o processamento de uma consulta usando essa estrutura consiste em recuperar no índice invertido as listas referentes aos termos da consulta e, em seguida, percorrer cada lista de forma a calcular o valor de similaridade (*score*) de cada documento da coleção em relação à consulta. Ao final, os documentos da coleção são retornados, ordenados de acordo com seus respectivos valores de similaridade. Esse processo é conhecido como *ranking*.

Em grandes coleções de dados, as listas invertidas tendem a ser longas. Por isso, algumas técnicas são utilizadas para limitar o uso da memória e permitir o acesso rápido ao conteúdo das listas durante o processamento da consulta. Buscando atender esses objetivos, algumas das estratégias utilizadas são: (i) armazenar os documentos na lista de forma comprimida e (ii) dividir cada lista em blocos de documentos de forma a agilizar a busca por um documento específico. A organização de um índice invertido é determinante para a escolha de qual algoritmo será melhor aplicado no processamento de consultas. Existem duas formas principais de organização de um índice invertido: (i) índices ordenados por impacto e (ii) índices ordenados por documento.

Em índices ordenados por impacto, os documentos de cada lista invertida são ordenados de acordo com seu peso, ou seja, sua contribuição para o *score* final calculado para o documento. A forma mais simples de verificar esse peso é utilizando a frequência do termo da consulta no documento. Anh e Moffat [1] propõem que ao invés de utilizar a frequência para determinar esse impacto, é possível utilizar diretamente o valor pré-computado do *score* que define a similaridade entre o termo da consulta e o documento dentro da lista invertida. Isso evita que os cálculos de similaridade sejam feitos durante o processamento da consulta.

Em índices ordenados por documento, as listas possuem seus documentos ordenados de acordo com um identificador único (docId). Para acelerar o acesso às entradas do índice são utilizadas estruturas conhecidas como *skip-lists*. Como os documentos da lista estão ordenados por um docId, as *skip-lists* podem armazenar para cada bloco, a quantidade de documentos representados por ele e a identificação do maior (ou menor) docID presente nele. Assim, o algoritmo de processamento utiliza as *skip-lists* para saltar entradas desnecessárias, tornando o processamento da consulta mais rápido.

Segundo as formas de organização de um índice invertido, existem duas estratégias principais que os algoritmos de processamento de consulta utilizam: (i) processamento termo-a-termo (TAT), que utiliza índices ordenados por impacto e (ii) processamento documento-a-documento (DAD), que utiliza índices ordenados por documento.

Algoritmos de processamento termo-a-termo percorrem as listas invertidas sequencialmente. Como um documento pode ocorrer em qualquer posição das listas invertidas, durante o processamento, é criado um acumulador para cada novo documento avaliado, a fim de armazenar o valor de similaridade entre o documento e a consulta. O valor de similaridade é calculado parcialmente ao longo do processo e apenas no final é possível obter o valor completo de similaridade entre um documento e a consulta.

Algoritmos de processamento documento-a-documento percorrem as listas invertidas em paralelo. Cada lista tem um ponteiro que aponta para o documento atual que está sendo processado. Essa estratégia permite que o documento que estiver sendo processado tenha seu valor de similaridade completamente calculado. Isso evita que seja necessário manter em memória uma lista muito grande de acumuladores, sendo suficiente manter em memória apenas um conjunto com um número limitado de documentos com os maiores valores de similaridade. Esse conjunto é atualizado à medida que o processamento avança. Esse tipo de processamento permite reduzir consideravelmente o consumo de memória em relação ao processamento termo-a-termo.

No Capítulo 7 são apresentados experimentos tanto com algoritmos de processamento termo-a-termo quanto com algoritmos de processamento documento-a-documento a fim de verificar qual estratégia se adequa melhor no cenário provido pela representação S-BoVW.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, são apresentadas soluções propostas na literatura consideradas relevantes e que estão relacionadas aos objetivos deste trabalho.

3.1 Métodos Baseados em Blocos

Alguns descritores CBIR adotam estratégias que consideram apenas características globais da imagem. Em geral, adotar essa prática permite que o tamanho da coleção de características se mantenha razoavelmente baixo. Por outro lado, o uso exclusivo de descritores globais pode fazer com que áreas de interesse de cunho local deixem de ser devidamente representadas.

Nesse sentido, algoritmos de segmentação podem ser utilizados para analisar propriedades locais das imagens. Segmentação é o processo de agrupamento dos pixels pertencentes a um mesmo objeto ou região presente em uma imagem. Algumas das categorias nas quais técnicas de segmentação podem ser classificadas, são [29]: (i) técnicas baseadas em região, (ii) técnicas baseadas em limiar e (iii) técnicas baseadas em arestas.

Uma outra alternativa que pode ser adotada para analisar propriedades locais é a extração de subáreas da imagem independente da detecção de objetos ou de regiões. Essa estratégia, conhecida como particionamento, consiste em dividir as imagens em um conjunto de blocos que podem ser de tamanho fixo ou não.

Um exemplo interessante de descritor local que divide a imagem em blocos é o Dense SIFT [5, 45], uma variação do SIFT que dispensa a etapa de detecção de pontos de interesse. Ele divide a imagem em uma grade densa (*dense grid*) e utiliza o *Histogram of*

Oriented Gradients (HOG) para descrever cada célula dessa grade. Apesar do Dense SIFT eliminar a etapa de detecção dos pontos de interesse, ele ainda necessita da etapa de agrupamento, na qual os descritores SIFT individuais são reduzidos a um vocabulário menor de palavras visuais. Além disso, o Dense SIFT foi proposto para tarefas de detecção de objetos e reconhecimento de cenas, e normalmente não é aplicado em tarefas de recuperação em grande escala [19].

Existe uma outra corrente de métodos baseados em blocos que utiliza uma estratégia de divisão de espaço piramidal [23, 6] para representar características locais das imagens. Um exemplo que segue essa estratégia é o método *Pyramidal Histogram of Oriented Gradients* (PHOG) [6], que também utiliza descritores SIFT extraídos de células de uma grade densa, mas no lugar de formar diretamente o vetor de características para a imagem, os termos são reunidos em um pirâmide de histogramas, na qual a base é equivalente à representação BoVW padrão para a imagem toda. Em cada nível subsequente da pirâmide, a imagem é dividida em sub-regiões, de uma forma recursiva, com cada região em cada nível da pirâmide sendo representada pelo seu próprio histograma. Esse método também precisa usar algoritmo de agrupamento para a construção do vocabulário visual. Assim como o Dense SIFT, o PHOG foi proposto para o contexto de classificação. A maior diferença entre esses descritores e os descritores que são estudados aqui, é que esses utilizam o paradigma S-BoVW para (i) descrever as imagens por meio de palavras visuais sem a necessidade de gerar um vocabulário visual prévio; e para (ii) aplicar técnicas tradicionais de indexação, busca e *ranking* para calcular similaridade entre imagens.

Dagli e Huang [9] apresentaram uma proposta de particionamento fixo que é realizada em duas etapas. Primeiro cada imagem é particionada em 16 blocos (4×4). Em seguida, é feita uma segmentação para distinguir o *background* e o *foreground* em cada bloco da imagem. Para representar as características de baixo nível foram utilizados os descritores HSV e o MPEG-7 HS [26]. A imagem de consulta passa por esse mesmo processo. Características semelhantes aos blocos da imagem de consulta são buscadas em todos os blocos da coleção, considerando inclusive diferentes escalas.

A atribuição de diferentes pesos para blocos de acordo com sua localidade na imagem é apresentada por Zhu e Yang [50]. Os pesos são distribuídos de forma não uniforme de tal forma que um peso maior é atribuído à região central, um peso menor é atribuído

às regiões de borda e um peso menor ainda é atribuído às regiões de extremidade. A similaridade entre as imagens é calculada considerando os pesos para fatores globais e fatores locais.

No trabalho apresentado por Petrina et al. [21], foi proposto um descritor que extrai evidências de cor baseando-se em uma análise local considerando partições fixas. Foram experimentados esquemas de particionamento que incluíam 2, 3 e 5 partições, considerando sempre que o objeto de interesse estaria localizado na região central da imagem. Os experimentos realizados demonstraram que o esquema com 5 partições apresentou eficácia melhor do que os outros esquemas.

Dois métodos baseados em blocos para representar textura são propostos por Takala e Pietikäinen [42]. A abordagem faz uso do descritor de textura LBP [31] para representar as imagens. O primeiro método divide a imagem em blocos de tamanhos iguais de onde são extraídos os histograms LBP. Esses histograma são comparados utilizando a função de distância $L1$. A segunda abordagem usa a representação em blocos apenas para as imagens da coleção, enquanto que para a imagem de consulta é gerado um histograma simples. Assim, os histogramas da coleção são concatenados de acordo com o tamanho da imagem de consulta de forma a encontrar a melhor combinação a partir da exploração da técnica de deslizamento de janela.

3.2 Métodos C-BoVW

A ideia de representação de imagens por meio de palavras visuais no contexto de recuperação visual foi primeiramente apresentada por Sivic e Zisserman [39]. Os autores adotaram os detectores MSER e Harris-Affine para identificar os pontos de interesse das imagens da coleção. O descritor SIFT foi utilizado para descrever os pontos de interesse detectados. O vocabulário foi construído usando o algoritmo de agrupamento *K-means*. A função de distâcia utilizada para o agrupamento foi a Mahalanobis. Foi utilizada uma heurística com o intuito de eliminar as palavras visuais mais frequentes e as mais raras. Assim, foram removidas 5% das palavras visuais mais frequentes e 10% das menos frequentes para o processo de indexação e processamento da consulta. Esses limiares foram adotados de forma empírica com o objetivo de reduzir o tamanho do índice invertido e manter ainda assim uma quantidade discriminativa de palavras visuais. Além disso, foi

explorada a ideia de consistência espacial que é baseada em uma estratégia utilizada na recuperação textual que alcança melhores resultados quando os termos da consulta aparecem juntos nos documentos recuperados. Assim, em um primeiro momento a recuperação é realizada utilizando os vetores de termos ponderados e em um segundo momento é feit uma reordenação com base em uma medida de consistência espacial. Uma forma simples de garantir a consistência espacial é exigir que vizinhos associados a regiões da consulta estejam presentes em uma área circundante das respostas recuperadas. O vocabulário usado foi formado com 10.000 palavras visuais. Essa estratégia é referenciada neste trabalho como método BoVW.

No trabalho realizado por Jégou et al. [19], o problema de recuperação de imagens em larga escala é tratado considerando três fatores: acurácia, eficiência e uso de memória. O método proposto é chamado *Vector of Locally Aggregated Descriptor* (VLAD). O VLAD agrega descritores SIFT, buscando uma representação compacta das imagens da coleção. Em comparação com o BoVW, menos palavras visuais são necessárias. Primeiramente, o vocabulário é aprendido com o algoritmo *K-means*. Posteriormente, cada descritor local da imagem é associado à palavras visual mais próxima. A representação VLAD consiste em uma dimensão $D(k \times d)$, em que k representa a quantidade de palavras visuais (*clusters*) e d é a dimensão dos descritores locais. A ideia usada no VLAD consiste em acumular, para cada palavra visual, a soma das diferenças entre os vetores dos descritores associados a cada palavra visual e o vetor que representa cada palavra visual (centróide). Os experimentos realizados mostraram que bons resultados podem ser obtidos com um vocabulário relativamente menor, resultando assim em índices mais compactos. Entretanto, como o vocabulário é pequeno, a chance de associação entre as palavras do vocabulário e as palavras que representam uma imagem é maior, o que pode tornar o processamento de consulta mais lento.

3.3 SDLC: *Sorted Dominant Local Color*

Vidal et al. [46] propuseram o *Sorted Dominant Local Color* (SDLC), um descritor que consiste em dividir as imagens em blocos e gerar uma assinatura textual para cada bloco, apenas com base em suas cores mais frequentes, dispensando qualquer técnica de agru-

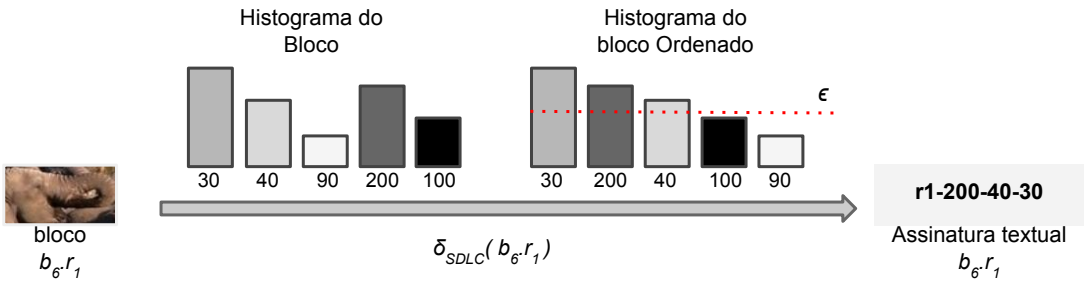

Figura 3.1: Função de mapeamento do SDLC.

pamento nesse processo. A assinatura textual gerada pelo SDLC é definida como

$$\delta(b_i.r_j) = \langle \hat{r_j} - \hat{c}_1 - \hat{c}_2 - \cdots - \hat{c}_n \rangle,$$

em que $n \geq 0$, $\hat{r_j}$ identifica a região r_j onde o bloco b_i ocorre, '-' é o símbolo separador. Cada valor $\hat{c}_k$ representa a cor que ocorre no bloco, de tal forma que a frequência da cor no bloco seja maior que um limiar ε e $(\hat{c}_i) > (\hat{c}_j)$, para $i < j$.

Em outras palavras, a função de mapeamento do SDLC recebe como entrada o histograma de cor de um bloco da imagem e produz uma assinatura composta pela identificação da região onde o bloco ocorreu, concatenada com os códigos das cores cujas frequências estão acima de um limiar ε. Tais códigos são ordenados de forma decrescente. Por exemplo, se um bloco de uma região 1 de uma imagem contiver apenas as cores 200, 30 e 40 com frequência superior a um limiar ε escolhido, esse bloco será representado pela *string* "r1-200-40-30". Um exemplo da assinatura textual gerada pelo SDLC para um bloco é apresentado na Figura 3.1.

$$\text{sim}(d, q) = \frac{\sum_{i=1}^{t} wid * wiq}{\sqrt{\sum_{i=1}^{t} wid^2} * \sqrt{\sum_{i=1}^{t} wiq^2}}$$

Figura 3.2: Função de similaridade do cosseno.

A representação gerada pelo SDLC para uma imagem é semelhante a um documento composto por um conjunto de termos, e por isso é possível aplicar técnicas de recuperação textual para indexar e processar consultas. No SDLC original foi utlizado o Modelo de Espaço Vetorial e a função de similaridade do cosseno, cuja fórmula é apresentada na Figura 3.2, em que d é o documento, q é a consulta, t é a quantidade de termos da consulta, w_{id} é o peso do termo i no documento d e w_{iq} é o peso do termo i na consulta q.

Para computar esses pesos foi usada a fórmula $w_{id} = tf \times idf$, onde tf é a frequência do termo no documento e idf é frequência invertida do documento. Sendo o idf, computado pela fórmula $idf_i = log(N/n)$, onde N é a frequência total do termo i na coleção e n é a quantidade de documentos da coleção que contém o termo i.

3.4 Técnicas de *Deep Learning*

Métodos de aprendizagem de máquina têm sido bastante explorados com o intuito de derivar soluções eficazes para tarefas de classificação e recuperação de imagens [47, 2, 38]. Mais especificamente, técnicas de apredizagem profunda, conhecidas como técnicas de *deep learning*, têm apresentado resultados promissores para esses tipos de tarefa. Wan et al. [47] e Babenko et al. [2] exploraram o uso de técnicas de *deep learning* em métodos CBIR, com foco no uso de redes neurais convolutivas. Em particular, Babenko et al. [2] investigaram possíveis formas de agregar características profundas locais para produzir descritores globais compactos para representar o conteúdo de imagens. A conclusão desses estudos é que os descritores de imagens produzidos pelas redes neurais convolutivas fornecem o que consideram o estado-da-arte para problemas de classificação e de recuperação de imagens.

Apesar de apresentarem resultados promissores, o uso de técnicas de aprendizagem também apresenta suas limitações. No caso do *deep learning*, o processo de aprendizagem não só exige o uso de grandes coleções para treinamento, como também requer o uso de *hardware* especializado para comportar a execução dos algoritmos que exigem grande capacidade de processamento. A investigação desses tipos de métodos não foi considerada durante o desenvolvimento deste trabalho.

3.5 Métodos de processamento de consulta textual

Persin et al. [33] propuseram uma estratégia de poda na qual o processamento termo-a-termo é interrompido quando se tem certeza que o conjunto dos *top-k* documentos de resposta já foi obtido. Como as listas invertidas estão ordenadas por impacto, é possível determinar a contribuição máxima que um documento teria em determinada lista invertida. Isso torna possível calcular um limiar de poda que seja capaz de terminar o processo

sem alterar o conjunto de resposta.

Strohman e Croft [41] propuseram um método considerado eficiente para o processamento de consultas termo-a-termo. Nele, o processamento é feito sobre um índice invertido ordenado por *score* que é mantido em memória. Uma poda dinâmica é aplicada em cada fase do processamento com o objetivo de reduzir a quantidade de acumuladores necessários para obter o conjunto final de respostas sem ser necessário avaliar todos os candidatos a compor o topo final do *ranking*.

Broder e Carmel [7] propuseram um dos trabalhos mais importantes sobre o processamento de consultas documento-a-documento. O método proposto conhecido como *Weak AND* ou *Weighted AND* (WAND), armazena o maior *score* de cada lista invertida durante a indexação da coleção de documentos. Essa informação é chamada de *MaxScore*. Ter conhecimento do maior *score* que os documentos de uma lista podem atingir ajuda a evitar o processamento de documentos que não possuem a chance de alterar o conjunto de respostas. Durante o processamento de uma consulta, os documentos que estão sendo avaliados, chamados de pivôs, são analisados em duas etapas. Na primeira etapa, é verificado o *MaxScore* de cada lista onde o pivô tem chance de ocorrer. Quando a soma dos *MaxScores* de cada lista analisada for maior do que o limiar de poda atual, a lista invertida de cada termo será acessada para que o documento pivô tenha seu *score* real calculado. Quando o *MaxScore* não superar o limiar de poda, o documento pivô é descartado e um novo pivô é selecionado.

Ding e Suel [11] propuseram um método de processamento de consultas documento-a-documento baseado no WAND, conhecido como *Block Max Wand* (BMW). No BMW, além do *MaxScore*, também é utilizado o maior *score* dos documentos presentes nos blocos das *skip-lists*. A estratégia de poda é similar à utilizada no WAND, com a vantagem de possuir um *score* máximo mais próximo do *score* real de cada documento. Com essa abordagem, o BMW consegue descartar ainda mais documentos que não apresentam peso suficiente para serem inseridos no topo do *ranking* de respostas. A poda no BMW acontece em dois momentos: (i) quando o *MaxScore* não supera o limiar de descarte e (ii) quando o *BlockMaxScore* não supera o limiar de descarte. O limiar de descarte é dinamicamente atualizado sempre que um documento possui um *score* superior ao limiar atual. Esse documento atualiza o topo de respostas e o limiar de poda é atualizado com o peso do documento com menor *score* do topo de respostas.

Rossi et al. [36] propuseram um método baseado no BMW, chamado *Block Max Wand Candidate Selector* (BMW-CS), que aplica uma estratégia de poda utilizando as informações de *MaxScore* e *BlockMaxScore* no processamento do índice em duas camadas. Na primeira fase do método a primeira camada, que contém os documentos de maior impacto para o termo, é processada para a seleção de documentos candidatos a resposta. Os documentos que não forem encontrados em todas as listas serão processados em uma segunda etapa. Na segunda etapa, são buscados na segunda camada do índice apenas os documentos que não tiveram o *score* total calculado. Com essa estratégia o método consegue processar consultas de forma mais rápida do que o método BMW. A desvantagem deste método está em não garantir o *ranking* de respostas correto, pois não seleciona para a lista de candidatos documentos que não ocorrem na primeira camada. Dessa forma, se um documento com score suficiente para estar no *ranking* de respostas não aparecer na primeira camada de pelo menos um dos termos da consulta, então tal documento será eliminado do *ranking* pelo BMW-CS.

Daoud et al. [10] propuseram uma modificação para o método BMW-CS com o objetivo de garantir o ranking correto. O método proposto chama-se *Block Max WAND with Candidate Selection and Preserving Top-K Results* (BMW-CSP) e inclui uma terceira fase de processamento que é ativada no caso de não haver garantia dos *top-k* documentos do *ranking*. Ou seja, a terceira fase é executada quando existir a chance de uma entrada que ocorre apenas na segunda camada dos termos da consulta ter peso suficiente para alterar o ranking de documentos da resposta. Tal situação ocorre quando a soma das pontuações máximas dos termos da consulta na segunda camada do índice é maior do que o limiar de descarte após o processamento da segunda fase. A terceira fase realiza uma segunda passagem na segunda camada para procurar novos documentos a serem incluídos entre os *top-k* documentos do *ranking*. Os melhores resultados computados nas duas primeiras fases são utilizados para acelerar a terceira fase, uma vez que aumentam o limiar de descarte. Embora a execução de uma terceira fase pareça ser dispendiosa, os resultados apresentados demonstraram que ela não causa alterações significativas nos tempos finais de processamento de consulta quando se consideram os melhores cenários de configuração, uma vez que é raramente ativada. Por outro lado, com este passo final, o algoritmo de processamento de consultas preserva os resultados, ao mesmo tempo que ainda se mantém mais rápido do que o algoritmo BMW.

Capítulo 4

S-BoVW: *Signature-based Bag of Visual Words*

Neste capítulo, é apresentado o paradigma *Signature-based Bag of Visual Words* (S-BoVW), uma definição formal e genérica para a estratégia de representação adotada pelo SDLC. Na definição do SDLC, apenas uma função de mapeamento entre blocos e palavras visuais foi considerada. A ideia por trás do S-BoVW é deixar claro que esse paradigma define uma nova categoria de métodos, uma vez que sua estratégia de representação permite que novas funções de mapeamento sejam propostas, derivando assim descritores diferentes do SDLC.

4.1 Visão Geral

No paradigma S-BoVW, imagens são divididas em blocos e assinaturas textuais são criadas de forma a representar o conteúdo de cada bloco como uma *string*. Em seguida, as assinaturas textuais geradas são usadas para calcular a similaridade a partir do uso de funções de similaridade comumente adotadas em sistemas de recuperação textual. Esses passos são formalmente descritos a seguir.

Seja $\hat{I}$ uma imagem, definida como um par $(D_I, \vec{I})$, em que [43]:

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

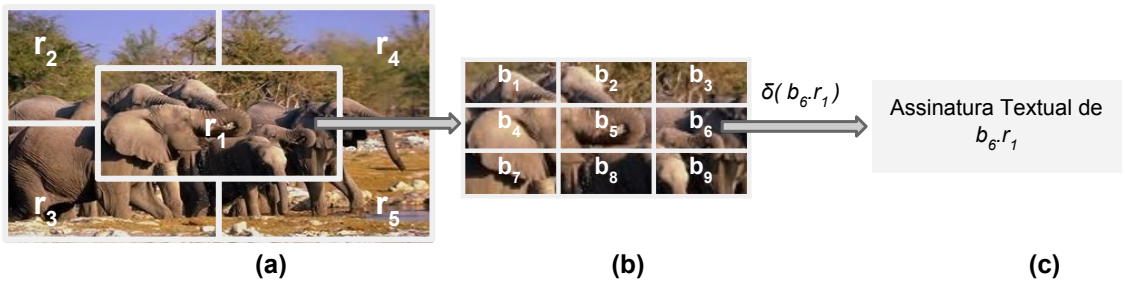

Figura 4.1: Processo de extração de assinatura textual pelo S-BoVW.

sistema RGB for atribuída a cada *pixel*).

Seja $\mathcal{R} = \{r_1, r_2, \ldots, r_\eta\}$ um conjunto de η regiões de $\hat{I}$, tal que cada *pixel* p em D_I pertence a uma única região $r_j \in \mathcal{R}$, como apresentado na Figura 4.1(a). Seja $\mathcal{B}_{r_j} = \{b_1.r_j, b_2.r_j, \ldots, b_\beta.r_j\}$ uma partição da região r_j consistindo de blocos de pixels consecutivos não sobrepostos, como apresentado na Figura 4.1(b).

Cada bloco $b_i.r_j \in \mathcal{B}_{r_j}$ tem seu conteúdo representado em termos de uma assinatura textual que é gerada por meio de uma função de mapeamento $\delta(b_i.r_j)$, como apresentado na Figura 4.1(c). Essa função é responsável por mapear um bloco $b_i.r_j$ em uma *string*. A função de mapeamento é o elemento central que torna os métodos S-BoVW diferentes entre si.

A partir da representação produzida pelo paradigam S-BoVW, um modelo de recuperação textual pode ser adotado a fim de calcular a similaridade entre as imagens. Esse passo requer a escolha de funções de similaridade apropriadas como também a escolha de esquemas de pesos para produzir o *ranking* final. Para cada função de mapeamento, torna-se necessário o estudo da melhor combinação entre funções de similaridade, esquemas de pesos e também tamanhos de bloco. Felizmente, existe uma vasta literatura sobre modelos de recuperação de informação e algoritmos de recuperação textual que podem ser adotados nessa etapa.

O fato do paradigma S-BoVW não precisar da etapa de agrupamento para gerar as palavras visuais permite que sejam criadas representações visuais para todos os blocos da imagem, enquanto que nos métodos C-BoVW isso não é um processo viável devido ao custo computacional requerido. Como no S-BoVW é possível indexar todos os blocos da imagem a um baixo custo, o uso de algoritmos de detecção de pontos de interesse torna-se dispensável. Como consequência, a abordagem S-BoVW tende a produzir métodos de processamento de consultas mais rápidos quando comparados com a abordagem C-

BoVW.

4.2 Funções de Mapeamento S-BoVW

Métodos que se baseiam no paradigma S-BoVW podem usar diferentes funções de mapeamento para descrever os blocos das imagens com palavras visuais. A seguir são apresentadas funções de mapeamento que seguem o paradigma S-BoVW. As funções *Sorted Dominant Local Texture* (SDLT) e *Sorted Dominant Local Color and Texture* (SDLCT) são funções novas que foram propostas por nós durante o desenvolvimento deste trabalho e foram publicadas no trabalho apresentado em [13].

4.2.1 SDLC

O método SDLC pode ser considerado a primeira função de mapeamento S-BoVW. O SDLC utiliza a composição de cores dos blocos para gerar as palavras visuais, conforme foi explicado na Seção 3.3.

4.2.2 SDLT

A primeira função de mapeamento proposta por nós durante o desenvolvimento deste trabalho foi chamada de *Sorted Dominant Local Texture* (SDLT). Essa função foi definida como uma alternativa para representar o conteúdo de textura presente nas imagens. A função SDLT consiste em extrair características de textura das imagens e transformá-las em uma representação textual correspondente. Para cada bloco extraído da imagem, é gerado um histograma *Local Binary Pattern* (LBP) [42]. Um código LBP é calculado para cada *pixel* do bloco por meio de uma análise de vizinhança $3x3$, na qual o valor do *pixel* central é usado como um limiar na comparação com seus 8 pixels vizinhos. Para pixels com valores iguais ou acima do limiar é atribuído o valor 1, enquanto que para pixels com valores abaixo do limiar, é atribuído o valor 0 (*zero*). O código final gerado pelo LBP é produzido pelo somatório da multiplicação dos valores dos pixels limiarizados por potências de dois de acordo com a posição de cada pixel na vizinhança. O código gerado para cada pixel é usado para montar um histograma de 256 *bins*, onde cada *bin* corresponde a um padrão de textura. A Figura 4.2 apresenta um exemplo do código LBP gerado para um pixel cujo código de cor equivale à $p_x = 50$.

A assinatura textual gerada pela função SDLT é definida como

$$\delta(b_i.r_j) = \langle \hat{r_j} - \hat{t}_1 - \hat{t}_2 - \cdots - \hat{t}_n \rangle,$$

em que $n \geq 0$, $\hat{r_j}$ identifica a região r_j onde o bloco b_i ocorre, '-' é o símbolo separador. Cada valor $\hat{t}_k$ representa um padrão de textura que ocorre no bloco, de tal forma que a frequência desse padrão de textura no bloco seja maior que um limiar ε e $(\hat{t}_i) > (\hat{t}_j)$, para $i < j$.

Em outras palavras, a função de mapeamento SDLT recebe como entrada o histograma de código LBP de um bloco da imagem e produz uma assinatura composta pela identificação da região onde o bloco ocorreu, concatenada com os códigos dos padrões de textura cuja frequência esteja acima de um limiar ε. Tais códigos são ordenados de forma decrescente. Por exemplo, se um bloco de uma região 1 de uma imagem contiver os padrões de textura 129, 64 e 256 com frequência superior a um limiar ε, esse bloco será representado pela *string* 'r1-256-129-64'. Na Figura 4.2 é ilustrada a geração da assinatura textual realizada pela função SDLT para o bloco de uma imagem.

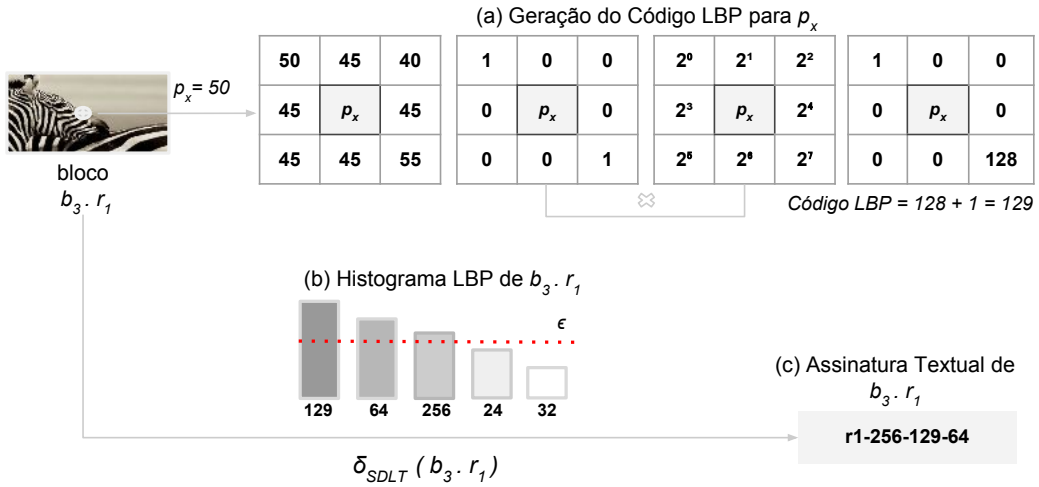

Figura 4.2: Função de mapeamento do SDLT.

4.2.3 SDLCT

A segunda função de mapeamento proposta por nós durante o desenvolvimento deste trabalho foi chamada de *Sorted Dominant Local Color and Texture* (SDLCT). Essa função consiste em uma combinação das representações geradas pelas funções SDLC e SDLT em nível de indexação. A representação textual gerada por essas funções são unificadas,

formando um único conjunto de palavras visuais que passará pelo processo de indexação. Desta forma, uma imagem será descrita por palavras visuais geradas com base tanto na composição de cor quanto na composição de textura dos blocos.

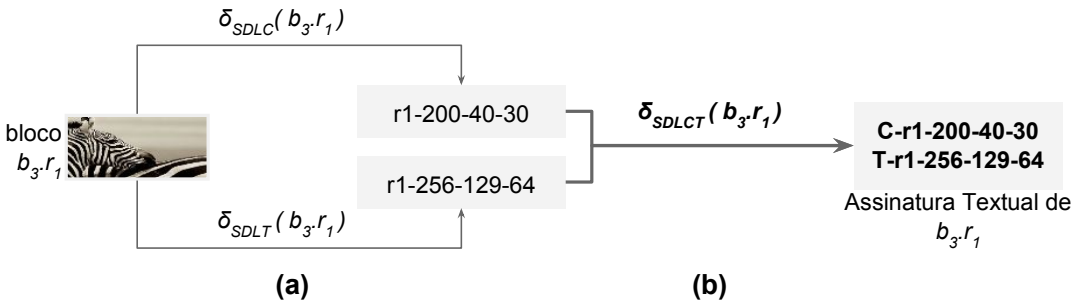

Figura 4.3: Função de mapeamento do SDLCT.

A proposta da função SDLCT é que as características de cor e textura se complementem, buscando uma representação mais completa das imagens. A Figura 4.3 apresenta o esquema de representação SDLCT para um bloco. Cada bloco da imagem será passado como entrada tanto para a função de mapeamento do SDLC quanto para a função de mapeamento do SDLT, como pode ser visualizado na Figura 4.3(a). Assim, a assinatura textual gerada pela função SDLCT para um bloco será composta por duas palavras, sinalizando apenas qual foi o método que deu origem à palavras, por exemplo *C* para o SDLC e *T* para o SDLT, como pode ser visualizado na Figura 4.3(b).

Capítulo 5

Estudo de parâmetros e funções de similaridade no S-BoVW

Neste capítulo, são apresentados os resultados do estudo do impacto de diferentes parâmetros e funções de similaridade em métodos baseados no paradigma S-BoVW.

5.1 Coleções e Protocolo de Avaliação

Os experimentos foram realizados com o total de sete coleções de imagens com diferentes tamanhos e características. A Tabela 5.1 apresenta a quantidade de imagens (#imagens) e o tamanho do conjunto de consultas (#consultas) de cada coleção. A coleção YAHOO-INRIA é uma expansão da coleção INRIA que adiciona um conjunto de imagens coloridas de tamanhos variados coletadas do diretório do Yahoo!.

Tabela 5.1: Coleções de Imagens.

	#imagens	#consultas
CCD [8]	5.457	50
INRIA [17]	1.491	500
OXFORD [34]	5.062	55
TEXTURE [22]	1.000	50
UKBENCH [28]	10.200	2.550
WANG [24]	1.000	50
YAHOO-INRIA	103.935	50

As métricas de avaliação utilizadas foram o MAP (*Mean Average Precision*) e P@10 [3], exceto na coleção UKB onde foi utilizada a métrica KS proposta pelos autores da coleção. O teste estatístico Wilcoxon [48] foi utilizado para validar diferenças

estatísticas, considerando significativas apenas as diferenças com confiança igual ou superior a 95%.

Para a avaliação de eficiência, foi utilizado o tempo médio de processamento de resposta a uma única consulta, o que significa que as consultas não foram processadas em *batch*. Não foi considerado o tempo de carregamento do índice em memória. Os experimentos foram executados em uma máquina Intel Xeon 3.33GHz X5680 24-core, com processador de 64GB de memória.

5.2 Parâmetros do S-BoVW

Os métodos S-BoVW atualmente propostos, exigem a configuração dos seguintes parâmetros: tipo de particionamento, número de blocos, limiar ε e função de similaridade. Neste trabalho, a coleção WANG foi adotada para estudar os parâmetros nos diferentes métodos. As configurações que obtiveram os melhores resultados foram utilizadas para avaliar o desempenho dos métodos em outras coleções.

Primeiro, foram estudadas estratégias de particionamento com o intuito de verificar seu impacto na qualidade dos resultados nas tarefas de recuperação. Ao utilizar um particionamento é possível definir a localização de um bloco dentro da imagem. As estratégias de particionamento foram definidas com base no trabalho apresentado em [21]. Na abordagem S-BoVW, a quantidade de partições afeta a formação das palavras e a composição do índice. Foram feitos experimentos sem o uso de partições ($0R$), com duas partições ($2R$), três partições ($3R$), e cinco partições ($5R$). A forma como as imagens foram particionadas é apresentada na Figura 5.1.

O número de blocos está diretamente relacionado à quantidade de palavras visuais que irá representar cada imagem. Foram experimentados valores de 64 (8×8) blocos, 625 (25×25) blocos, 1.296 (36×36) blocos e 4.096 (64×64) blocos.

O limiar ε afeta a composição de uma palavra e sua capacidade discriminativa. A escolha desse limiar deve ser feita de forma estratégica com o objetivo de garantir que uma mesma palavra seja capaz de representar não somente os blocos iguais, mas também blocos visualmente semelhantes. O valor desse limiar não pode ser muito alto, nem muito baixo. Um limiar muito baixo pode fazer com que as palavras geradas sejam muito específicas para um determinado bloco, perdendo sua capacidade de generalização. Por

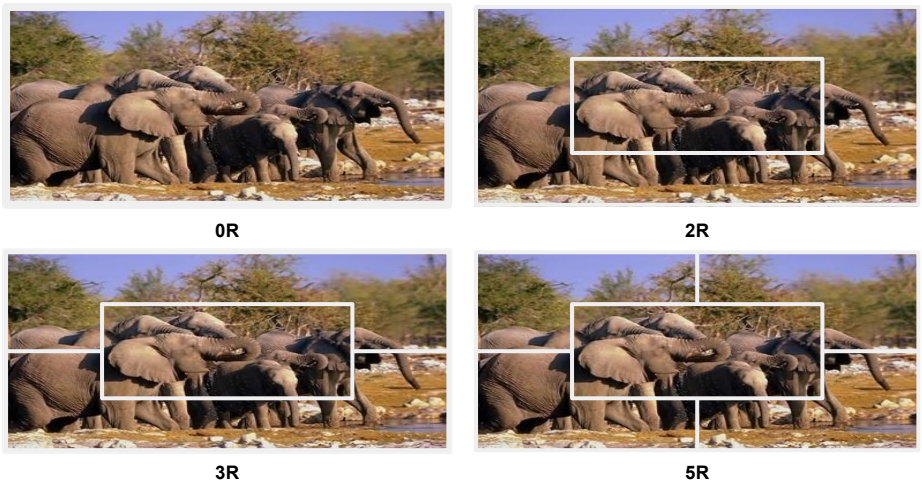

Figura 5.1: Estratégia de particionamento: sem partição (0R), duas partições (2R), três partições (3R) e cinco partições (5R).

outro lado, as palavras geradas a partir de um limiar muito alto podem se tornar genéricas em excesso, sendo ineficazes para a representação de especificidades dos blocos.

A Figura 5.2 mostra como o tipo de particionamento (0R, 2R, 3R e 5R), o número de blocos (64, 625, 1296 e 4096) e o limiar ε (1%, 3%, 5%, 7%, 10% e 20%) impactam na composição do índice, considerando a indexação da representação do SDLC gerada para as imagens da coleção WANG. O impacto pode ser analisado considerando o número total de termos únicos no índice e a quantidade total de entradas das listas invertidas do índice. Analisando os gráficos, é possível perceber que o tamanho do índice é fortemente afetado pelo número de blocos e pelo limiar ε, enquanto que o tipo de particionamento causa um impacto mais modesto. Esse comportamento foi o mesmo observado para os índices gerados para o SDLT.

Fica claro concluir que quanto maior o tamanho do índice, maior será o tempo de processamento de uma consulta sobre esse índice. No cenário de processamento de consultas, é necessário que haja um equilíbrio entre a qualidade da resposta e o tempo de processamento. Por esse motivo, a escolha da configuração a ser adotada para o método não pode estar baseada apenas na análise do tamanho do índice. Para alcançar o equilíbrio desejado, é preciso avaliar a qualidade dos resultados obtidos com a variação dos parâmetros adotados. Tanto a função de similaridade qanto o esquema de pesos aplicados no processamento da consulta, podem impactar na qualidade dos resultados obtidos. Na Seção 5.3, serão apresentados os experimentos que foram realizados com o intuito de verificar qual configuração consegue alcançar o melhor equilíbrio entre a eficácia e a eficiência nos

Figura 5.2: SDLC: Impacto do tipo de particionamento, do número de blocos, e do limiar ε na composição do índice da coleção WANG, considerando a quantidade de termos e o número total de entradas no índice.

métodos S-BoVW.

5.3 Impacto de Funções de Similaridade e Esquemas de Pesos na Seleção de Parâmetros do S-BoVW

Foram realizados experimentos considerando três funções de similaridade para verificar o impacto de cada uma nos métodos S-BoVW. Dentre as funções utilizadas estão a função probabilística BM25 [35], a função do Cosseno adotada no Modelo Vetorial (VSM - *Vector Space Model*) [37], e a função Yaelnn [14] adotada na implementação disponibilizada para os métodos VLAD [19] e BoVW [39].

No Modelo Vetorial, dado um vocabulário com n *visual words* distintas, cada imagem i da coleção é representada por um vetor $\hat{v}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,n})$. Cada dimensão desse vetor refere-se a um esquema de pesos que define a importância de uma palavra visual

presente na imagem i. A Tabela 5.2 apresenta os sete esquemas de pesos experimentados durante a aplicação do Modelo Vetorial. Essas alternativas combinam as seguintes informações básicas para calcular o peso de uma palavra visual t_j em uma imagem i:

- $tf_{i,j}$ representa o número de ocorrências de t_j em i.
- idf_j representa a importância de t_j na coleção, que é calculada como $idf_j = log(\frac{N}{df_j})$, onde N é o número de imagens da coleção e df_j é o número de imagens onde t_j aparece.
- $match_{q,i}$ representa um fator de *score* que calcula quantos termos da consulta q foram encontrados na imagem i. É uma informação dependente da consulta e não do termo.

Tabela 5.2: Esquemas de pesos.

w_1	$w_{i,j} = tf_{i,j}$
w_2	$w_{i,j} = idf_i$
w_3	$w_{i,j} = tf_{i,j} \times idf_i$
w_4	$w_{i,j} = tf_{i,j} \times idf_i \times match_{q,j}$
w_5	$w_{i,j} = idf_i \times match_{q,j}$
w_6	$w_{i,j} = match_{q,j}$
w_7	$w_{i,j} = tf_i \times match_{q,j}$

Durante os experimentos, foi possível concluir que o uso da norma dos documentos no VSM não apresentou um impacto relevante na qualidade dos resultados. Isso porque a quantidade dos termos gerados para representar as imagens na abordagem S-BoVW em geral é constante, tornando o valor da norma relativamente homogêneo. Experimentos realizados nas outras coleções demonstraram que remover a norma dos documentos também não causa mudanças significativas no *ranking* dos resultados. Assim, para calcular a similaridade entre duas imagens, foi adotada a medida de similaridade do produto interno. Essa medida alcança resultados bem similares à medida do Cosseno e apresenta menor custo computacional, uma vez que não requer a normalização dos vetores, como é requerido na medida do Cosseno.

As Figuras 5.3 e 5.4 apresentam, respectivamente, as variações de P@10 e MAP para o SDLC ao aplicar diferentes tipos de particionamento (0R, 2R, 3R e 5R), número de blocos (64, 625, 1296 e 4096), funções de similaridade e valores para o limiar ε (1%, 3%, 5%, 7%, 10% e 20%) na coleção WANG. Analisando os resultados apresentados, é

possível perceber que ao usar a função de similaridade VSM com o esquema de pesos w_4 o desempenho geral do SDLC se mantém competitivo mesmo ao usar parâmetros que produzem índices reduzidos.

A Tabela 5.3 apresenta uma comparação entre a configuração sugerida originalmente para o SDLC [46] e a configuração sugerida a partir dos experimentos realizados neste trabalho. Para essa análise foram consideradas as seguintes informações: número total de termos únicos no índice (#termos), quantidade total de entradas das listas invertidas do índice (#entradas), P@10, MAP e tempo médio do processamento das consultas. Conforme pode ser observado, a configuração composta por 625 blocos, sem particionamento ($0R$), limiar $\varepsilon = 10\%$ e a função de similaridade VSM com o esquema de pesos w_4 quando comparada à configuração adotada originalmente pelo SDLC (1296 blocos, particionamento $5R$, limiar $\varepsilon = 5\%$ e função de similaridade VSM com esquema de pesos w_3) alcança uma qualidade elevada dos resultados, ao mesmo tempo que produz um índice reduzido e diminui o tempo de processamento das consultas.

Tabela 5.3: SDLC: Comparação entre a configuração original (1296 blocos, particionamento $5R$, limiar $\varepsilon = 5\%$, VSM com w_3) e a configuração sugerida (625 blocos, particionamento $0R$, limiar $\varepsilon = 10\%$, VSM com w_4).

SDLC	#termos	#entradas	P@10	MAP	Tempo (em ms)
Configuração Original	130577	353733	0,82	0,57	1,10
Configuração Sugerida	17472	178507	0,86	0,59	0,80

As Figuras 5.5 e 5.6 apresentam, respectivamente, as variações de P@10 e MAP para o SDLT ao aplicar diferentes tipos de particionamento ($0R$, $2R$, $3R$ e $5R$), número de blocos (64, 625, 1296 e 4096), funções de similaridade e valores para o limiar ε (1%, 3%, 5% e 7%) na coleção WANG. Ao analisar os resultados, foi verificado que a configuração sem particionamento ($0R$), 625 blocos, limiar $\varepsilon = 3\%$ e função de similaridade VSM com o esquema de pesos w_3 forneceu um bom equilíbrio entre a qualidade dos resultados e o tamanho final do índice gerado.

Para realizar os experimentos com o SDLCT, foram utilizadas as melhores configurações encontradas individualmente para o SDLC ($0R$, 625 blocos, limiar $\varepsilon = 10\%$) e para o SDLT ($0R$, 625 blocos, limiar $\varepsilon = 3\%$). Uma função de combinação linear foi utilizada para variar o peso atribuído às palavras de acordo com as suas funções de origem (SDLC ou SDLT). Sendo α o fator de importância atribuído às palavras, S_{sdlc} e S_{sdlt} os *scores* calculados respectivamente para as palavras geradas pelo SDLC e SDLT, o

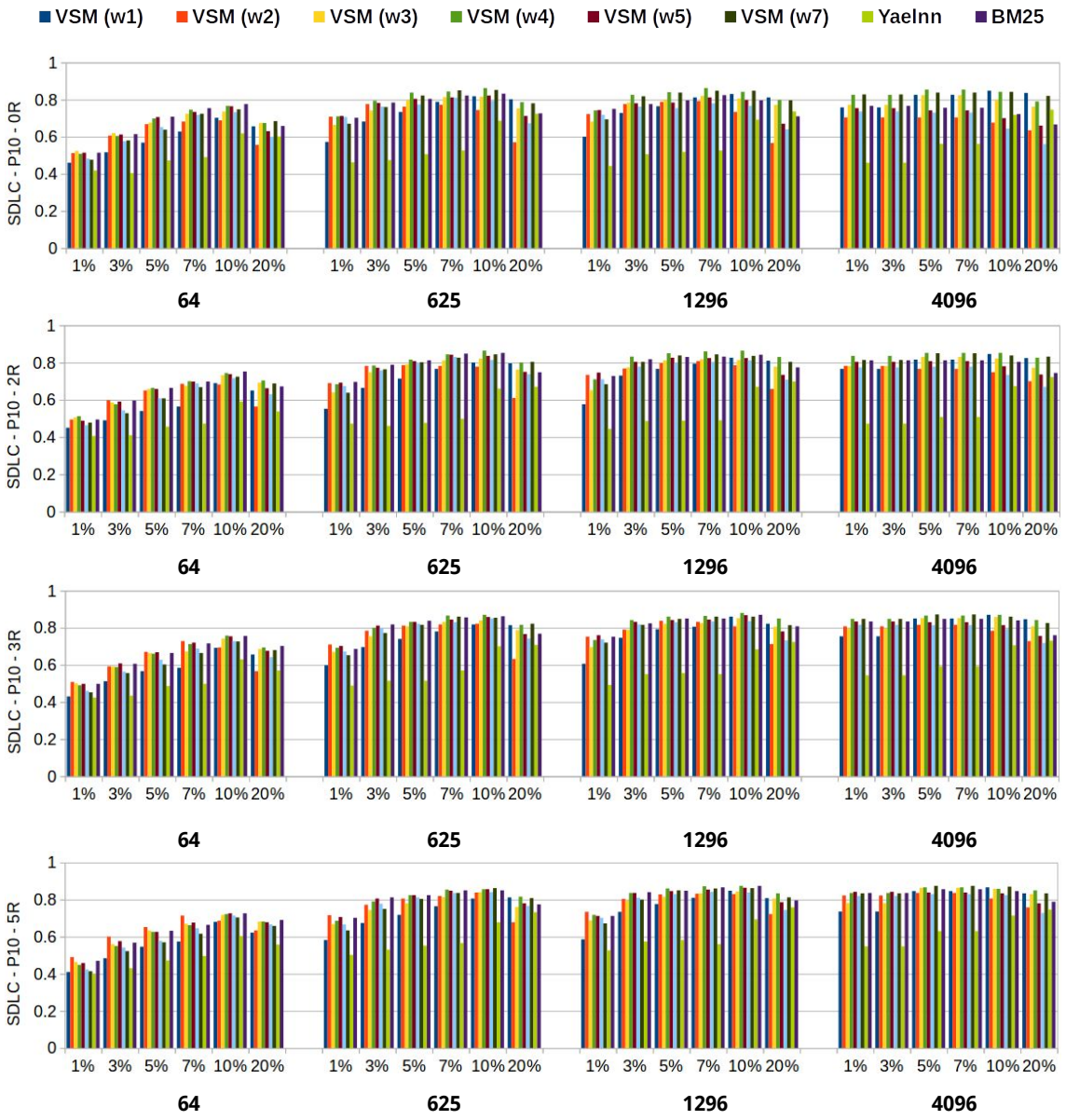

Figura 5.3: SDLC: Análise de P@10 sobre a variação de partições, número de blocos, funções de similaridade e valores do limiar ε na coleção WANG.

cálculo do *score* final do SDLCT para uma imagem i é apresentado como:

$$S_{sdlct}(i) = \alpha\, S_{sdlc}(i) + (1-\alpha)\, S_{sdlt}(i) \tag{5.1}$$

Também foram realizados experimentos para analisar o impacto de diferentes funções de similaridade ao processar consultas utilizando o SDLCT. Os resultados desses experimentos são apresentados na Figura 5.7. Analisando os resultados, foi possível perceber que a função de similaridade VSM com o esquema de pesos w_4 apresentou um com-

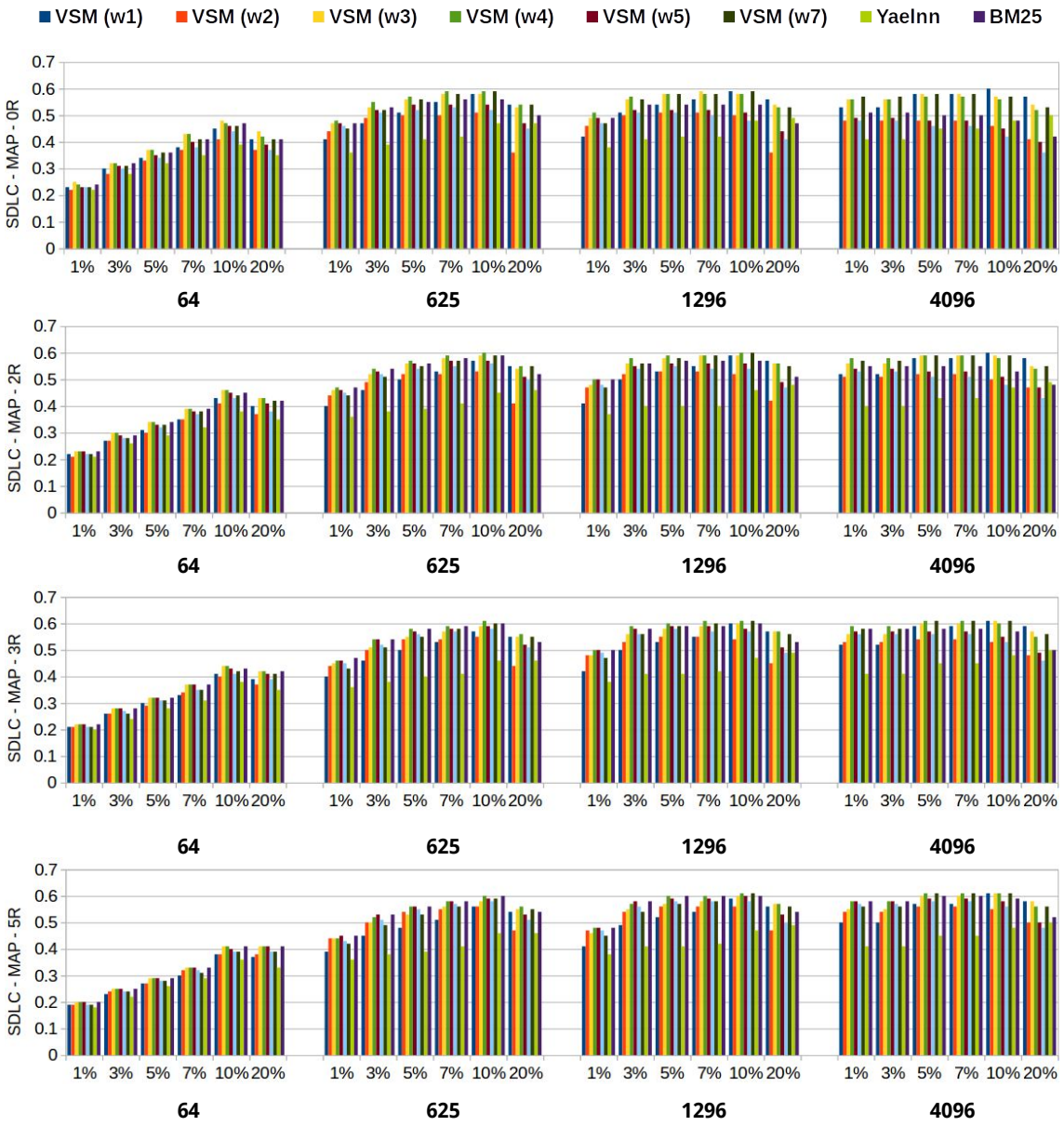

Figura 5.4: SDLC: Análise de MAP sobre a variação de partições, número de blocos, funções de similaridade e valores do limiar ε na coleção WANG.

portamento estável e competitivo quando comparada às outras funções de similaridade. Também é possível verificar que a estratégia de combinação adotada pelo SDLCT foi capaz de melhorar os resultados obtidos pelo SDLC e pelo SDLT individualmente. Para demonstrar a eficácia do método SDLCT, a Figura 5.8 apresenta um conjunto de consultas da coleção WANG com as respectivas respostas retornadas para o SDLCT. Analisando as imagens, é possível observar que o método consegue retornar bons resultados em diferentes contextos.

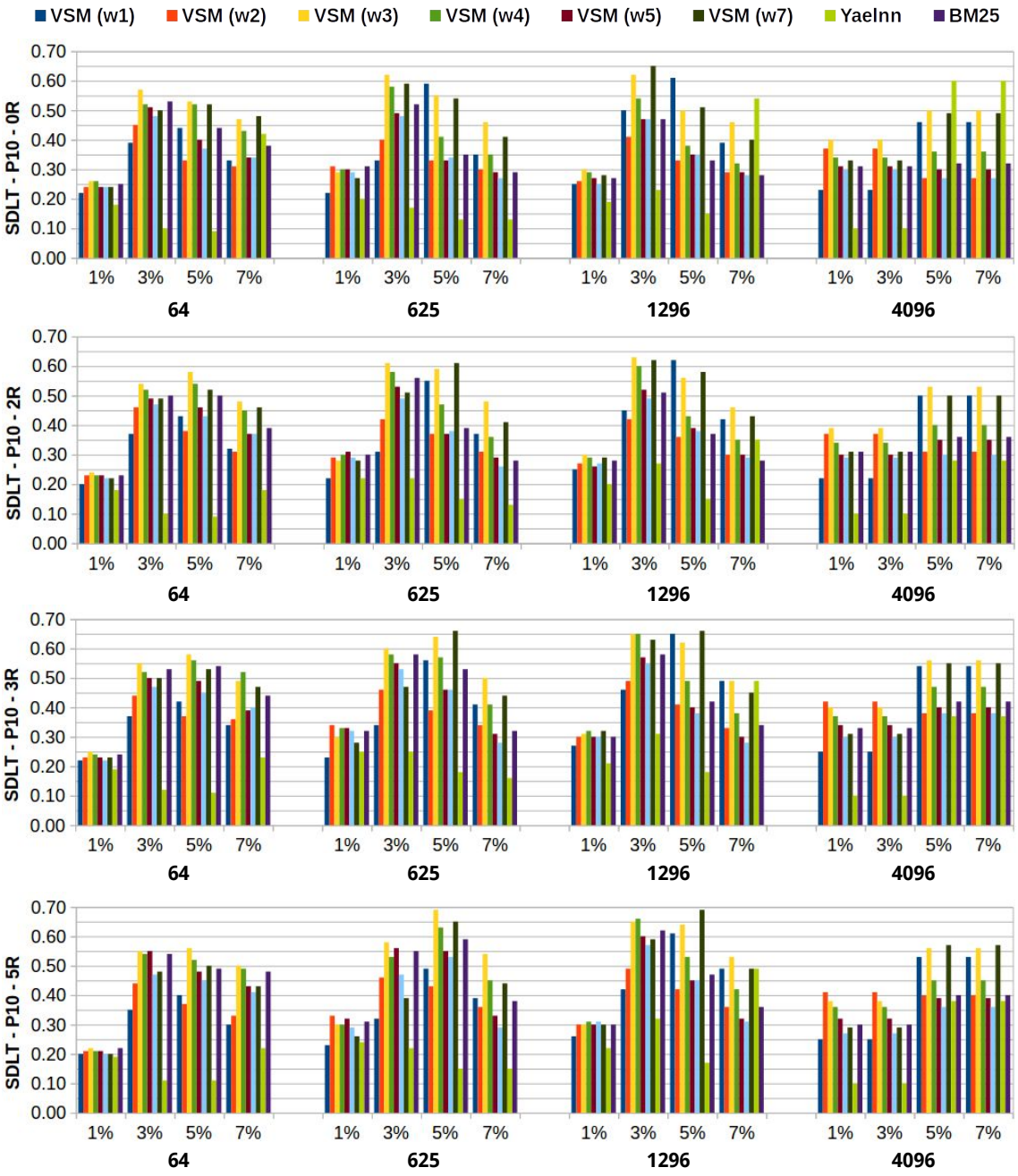

Figura 5.5: SDLT: Análise de P@10 sobre a variação de partições, número de blocos, funções de similaridade e valores do limiar ε na coleção WANG.

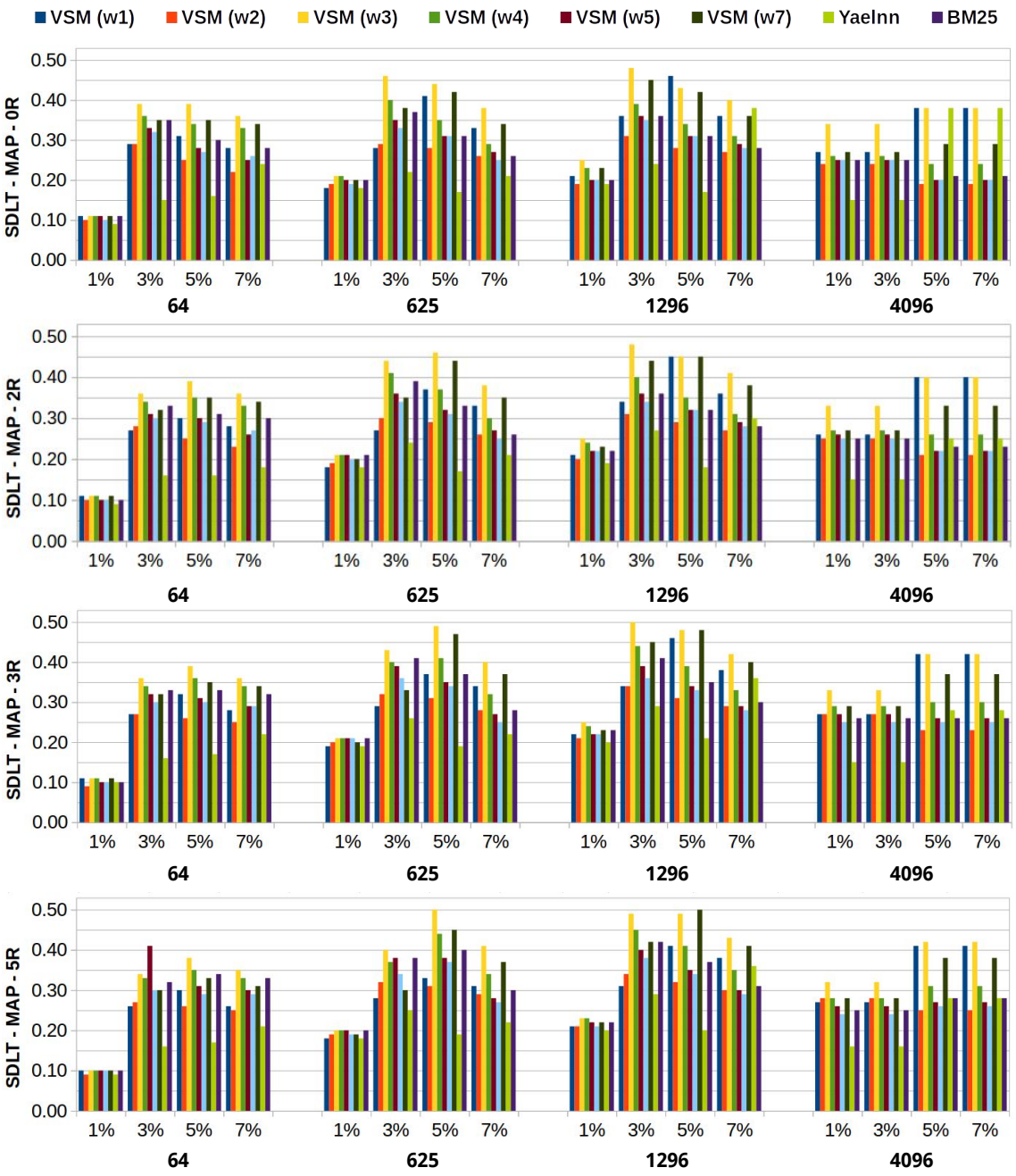

Figura 5.6: SDLT: Análise de MAP sobre a variação de partições, número de blocos, funções de similaridade e valores do limiar ε na coleção WANG.

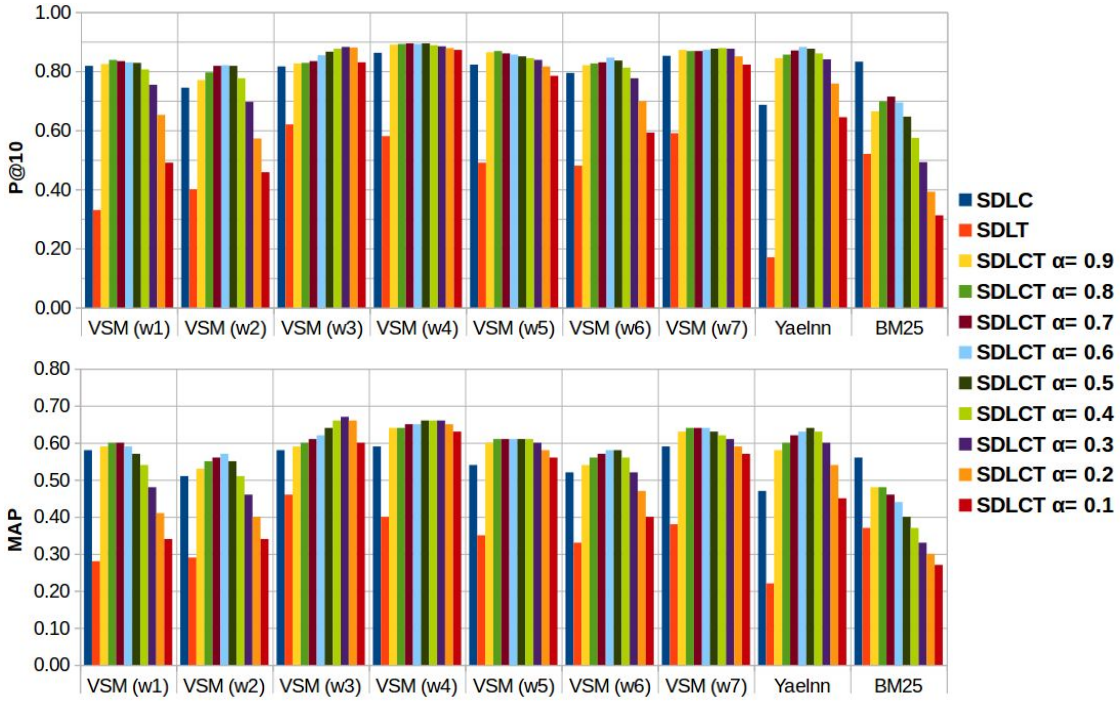

Figura 5.7: SDLCT: Análise de P@10 e MAP na coleção WANG ao variar pesos atribuídos aos termos do SDLC e do SDLT.

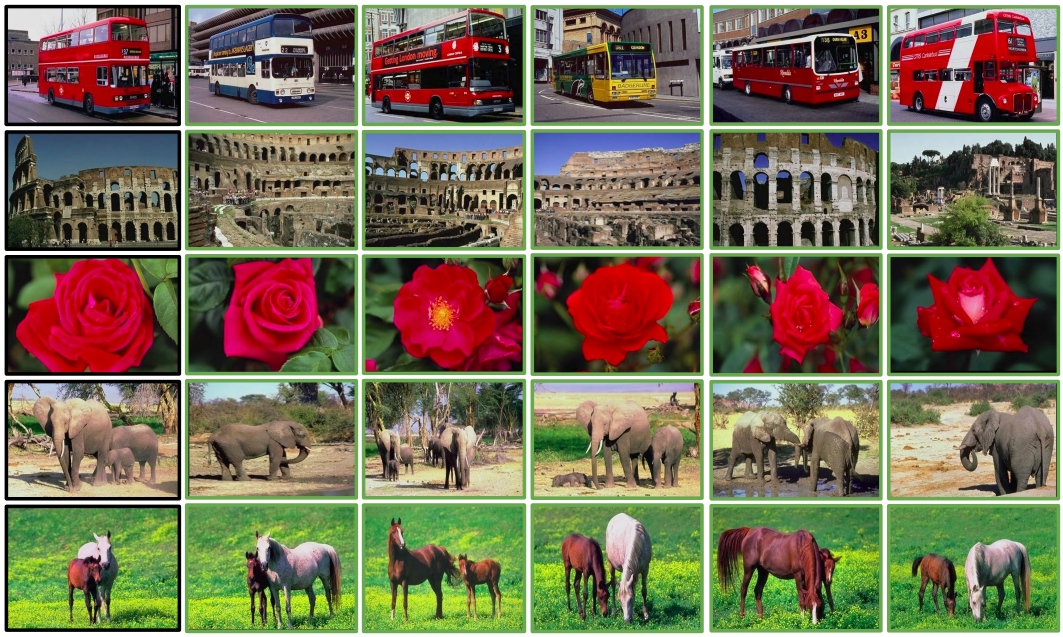

Figura 5.8: SDLCT: Resultado de processamento de consultas na coleção WANG. Cada linha apresenta a primeira imagem como sendo a consulta, seguida pelas cinco primeiras respostas retornadas.

Capítulo 6

Análise comparativa entre métodos S-BoVW e *baselines*

Neste capítulo, os experimentos com os métodos S-BoVW foram realizados utilizando as configurações que obtiveram melhor desempenho na coleção WANG. Para o SDLC, foi utilizada a configuração sem particionamento ($0R$), 625 blocos, limiar $\varepsilon = 10\%$ e aplicação da função de similaridade VSM com esquema de pesos w_4. Para o SDLT, foi utilizada a configuração sem particionamento ($0R$), 625 blocos, limiar $\varepsilon = 3\%$ e aplicação da função de similaridade VSM com esquema de pesos w_3. Para o SDLCT, foram utilizadas as mesmas configurações consideradas para o SDLC e SDLCT em termos de particionamento, número de blocos e limiar ε. A função de similaridade utilizada para o SDLCT foi VSM com esquema de pesos w_4 e o fator de combinação $\alpha = 0,5$.

A Figura 6.1 apresenta uma comparação entre os métodos S-BoVW, em termos de P@10 e MAP em diferentes coleções. Os resultados da aplicação do teste estatístico Wilcoxon [48] para validação das diferenças entre os métodos em termos de P@10, MAP e KS nas diferentes coleções são apresentados na Figura 6.4. Ao comparar o SDLT com o SDLC, é possível verificar que o desempenho do SDLT não foi expressivo em relação ao SDLC na maioria dos cenários. Ao buscar possíveis razões para o baixo desempenho do SDLT em relação ao SDLC, foi verificada a influência do descritor LBP [31] nos resultados obtidos pelo SDLT. O LBP foi o descritor utilizado pelo SDLT para codificar as informações de textura dos blocos. Conforme pode ser observado na Figura 6.2, o LBP também obteve desempenho inferior ao do SDLC na maioria dos cenários. É possível verificar uma tendência na qual quando o LBP obtém um desempenho inferior ao do SDLC,

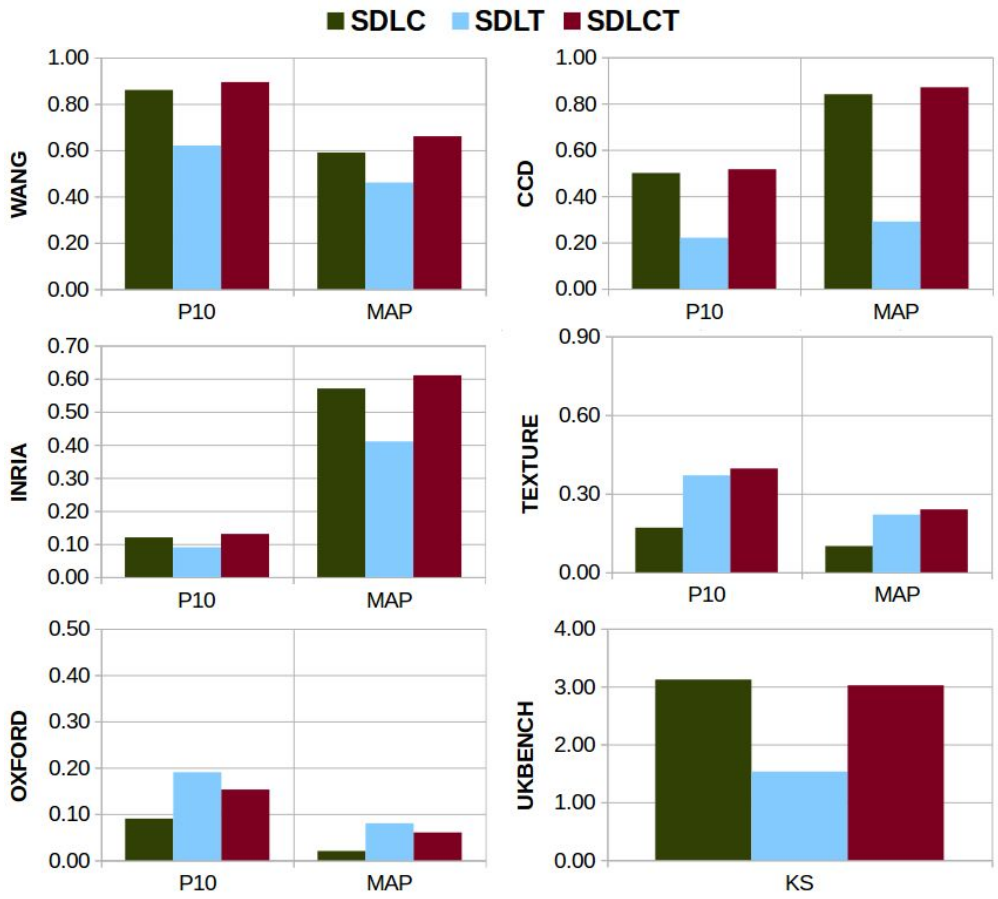

Figura 6.1: Comparação de desempenho entre os métodos S-BoVW.

o SDLT obtém um desempenho similar. Como conclusão, pode-se dizer que a estratégia S-BoVW adotada no SDLT foi capaz de capturar a informação provida pelo descritor de bloco LBP, mantendo seu desempenho qualitativo e adiconando as vantagens de usar uma representação textual das imagens. Os resultados interessantes relacionados ao SDLT ocorreram nas coleções TEXTURE e OXFORD. Especificamente na coleção TEXTURE isso foi esperado devido ao fato de o SDLT considerar aspectos de textura para representar as imagens, enquanto que o SDLC só considera aspectos de cor. Outra constatação interessante, é a que na coleção OXFORD, o SDLT obteve desempenho superior ao do SDLC, mesmo o LBP tendo apresentado desempenho inferior em relação ao SDLC.

Na Figura 6.1, ao comparar o SDLCT com o SDLC, é possível verificar que o SDLCT apresenta desempenho superior ao SDLC com diferença estatística significativa na maioria das coleções. As exceções ocorreram na coleção CCD, onde não houve diferença estatística significativa, e na coleção UKBENCH onde o SDLCT obteve desempenho inferior ao SDLC com diferença estatística significativa. Ao comparar o SDLCT com o SDLT, o SDLCT também apresenta desempenho superior ao SDLT com diferença es-

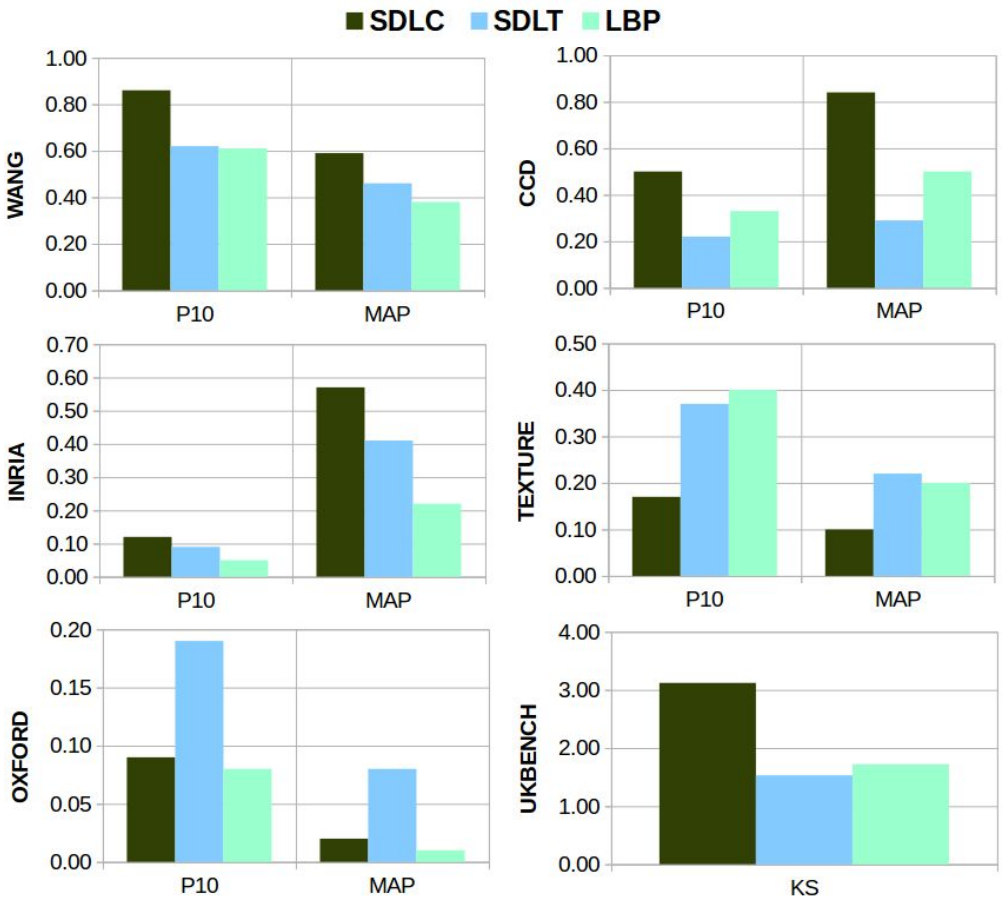

Figura 6.2: Análise de desempenho do SDLT a partir da comparação com os métodos SDLC e LBP.

tatística significativa na maioria das coleções. A exceções ocorreram na coleção TEXTURE, onde não houve diferença estatística significativa, e na coleção OXFORD, onde o SDLCT obteve desempenho inferior ao SDLT com diferença estatística significativa.

6.1 Comparação com *baselines*

Para representar o paradigma C-BoVW, os métodos BoVW [39] e VLAD [19] foram usados como *baselines*. Para o método BoVW, foram utilizados dois vocabulários distintos: um composto por $1k$ palavras visuais ($BoVW_{1k}$) e outro composto por $20k$ palavras visuais ($BoVW_{20K}$). Para o VLAD, foi utilizado um vocabulário composto por 64 palavras visuais. Para os dois métodos, foi utilizada uma implementação que aplica o extrator Hessian-Affine e o descritor SIFT respectivamente para as etapas de extração e de descrição dos pontos de interesse. Para ambos os métodos foram utilizados vocabulários aprendidos na coleção Flickr60K [17].

A Figura 6.3 apresenta a comparação de desempenho qualitativo entre os métodos

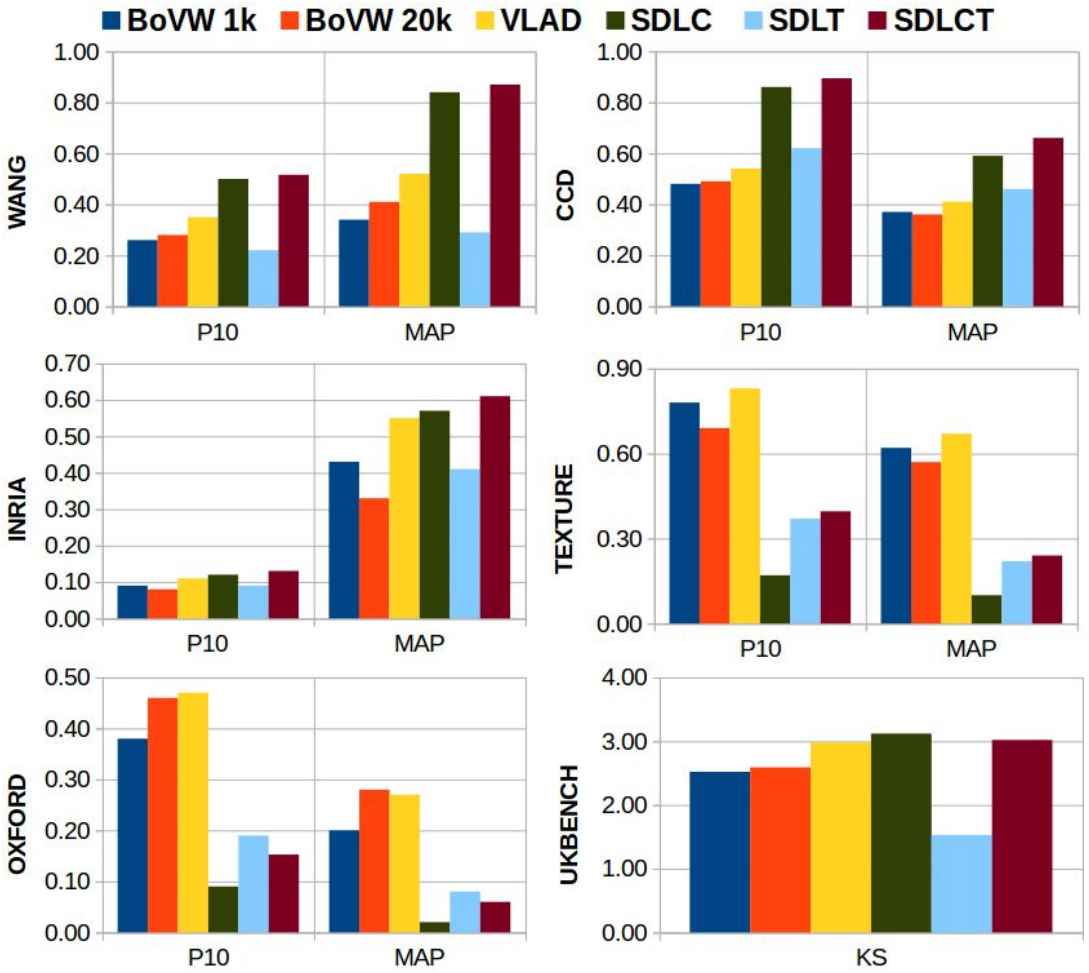

Figura 6.3: Comparação de desempenho entre os métodos S-BoVW e os *baselines*.

C-BoVW e S-BoVW. Os resultados da aplicação do teste estatístico Wilcoxon [48] para validação das diferenças entre os métodos em termos de P@10, MAP e KS nas diferentes coleções são apresentados na Figura 6.4.

Ao comparar o método SDLCT com os *baselines* C-BoVW, o SDLCT alcançou ganhos estatísticos significativos nas coleções WANG, CCD e INRIA. Nas coleções onde o desempenho dos métodos SDLC e SDLT foi muito inferior ao desempenho dos métodos C-BoVW, como nas coleções OXFORD e TEXTURE, o SDLCT também foi superado pelos *baselines*. Na coleção UKBENCH, mesmo o SDLC tendo o desempenho superior aos *baselines*, o SDLCT não conseguiu superar os *baselines*. O baixo desempenho da representação do SDLT na UKBENCH teve uma forte influência no desempenho final do SDLCT nessa coleção.

O baixo desempenho obtido pelos métodos S-BoVW em relação aos métodos C-BoVW nas coleções OXFORD e TEXTURE indica tarefas para as quais a estratégia S-BoVW não é competitiva. A Figura 6.5 apresenta um exemplo de consulta na coleção

WANG – P10

	1	2	3	4	5	6	7
bow1k 1	0	68	100	100	100	100	100
bow20k 2	68	0	100	100	100	100	99
lbp 3	100	100	0	100	100	17	95
sdlc 4	100	100	100	0	95	100	100
sdlct 5	100	100	100	95	0	100	100
sdlt 6	100	100	17	100	100	0	96
vlad 7	100	99	95	100	100	96	0

CCD – P10

	1	2	3	4	5	6	7
bow1k 1	0	67	100	100	100	17	100
bow20k 2	67	0	99	100	100	71	100
lbp 3	100	99	0	100	100	100	2
sdlc 4	100	100	100	0	8	100	99
sdlct 5	100	100	100	8	0	100	99
sdlt 6	17	71	100	100	100	0	100
vlad 7	100	100	2	99	99	100	0

INRIA – P10

	1	2	3	4	5	6	7
bow1k 1	0	100	100	100	100	4	100
bow20k 2	100	0	100	100	100	100	100
lbp 3	100	100	0	100	100	100	100
sdlc 4	100	100	100	0	100	100	97
sdlct 5	100	100	100	100	0	100	100
sdlt 6	4	100	100	100	100	0	100
vlad 7	100	100	100	97	100	100	0

TEXTURE – P10

	1	2	3	4	5	6	7
bow1k 1	0	99	100	100	100	100	100
bow20k 2	99	0	100	100	100	100	100
lbp 3	100	100	0	100	7	46	100
sdlc 4	100	100	100	0	100	100	100
sdlct 5	100	100	7	100	0	74	100
sdlt 6	100	100	46	100	74	0	100
vlad 7	100	100	100	100	100	100	0

OXFORD – P10

	1	2	3	4	5	6	7
bow1k 1	0	100	100	100	100	100	100
bow20k 2	100	0	100	100	100	100	81
lbp 3	100	100	0	99	99	84	100
sdlc 4	100	100	99	0	100	100	100
sdlct 5	100	100	99	100	0	97	100
sdlt 6	100	100	84	100	97	0	100
vlad 7	100	81	100	100	100	100	0

UKBENCH – KS

	1	2	3	4	5	6	7
bow1k 1	0	100	100	100	100	100	100
bow20k 2	100	0	100	100	100	100	100
lbp 3	100	100	0	100	100	100	100
sdlc 4	100	100	100	0	100	100	100
sdlct 5	100	100	100	100	0	100	100
sdlt 6	100	100	100	100	100	0	100
vlad 7	100	100	100	100	100	100	0

WANG – MAP

	1	2	3	4	5	6	7
bow1k 1	0	98	54	100	100	100	100
bow20k 2	98	0	81	100	100	100	100
lbp 3	54	81	0	100	100	100	64
sdlc 4	100	100	100	0	100	98	100
sdlct 5	100	100	100	100	0	100	100
sdlt 6	100	100	100	98	100	0	99
vlad 7	100	100	64	100	100	99	0

CCD – MAP

	1	2	3	4	5	6	7
bow1k 1	0	99	100	100	100	54	100
bow20k 2	99	0	86	100	100	96	100
lbp 3	100	86	0	100	100	100	44
sdlc 4	100	100	100	0	88	100	100
sdlct 5	100	100	100	88	0	100	100
sdlt 6	54	96	100	100	100	0	100
vlad 7	100	100	44	100	100	100	0

INRIA – MAP

	1	2	3	4	5	6	7
bow1k 1	0	100	100	100	100	100	100
bow20k 2	100	0	100	100	100	100	100
lbp 3	100	100	0	100	100	100	100
sdlc 4	100	100	100	0	100	100	67
sdlct 5	100	100	100	100	0	100	100
sdlt 6	100	100	100	100	100	0	100
vlad 7	100	100	100	67	100	100	0

TEXTURE – MAP

	1	2	3	4	5	6	7
bow1k 1	0	99	100	100	100	100	100
bow20k 2	99	0	100	100	100	100	100
lbp 3	100	100	0	100	42	52	100
sdlc 4	100	100	100	0	100	100	100
sdlct 5	100	100	42	100	0	20	100
sdlt 6	100	100	52	100	20	0	100
vlad 7	100	100	100	100	100	100	0

OXFORD – MAP

	1	2	3	4	5	6	7
bow1k 1	0	63	100	100	100	91	73
bow20k 2	63	0	100	100	100	98	38
lbp 3	100	100	0	97	100	100	100
sdlc 4	100	100	97	0	100	100	100
sdlct 5	100	100	100	100	0	100	100
sdlt 6	91	98	100	100	100	0	99
vlad 7	73	38	100	100	100	99	0

Figura 6.4: Resultados dos testes estatísticos realizados para validação das diferenças entre os métodos, em termos de P@10, MAP e KS em diferentes coleções.

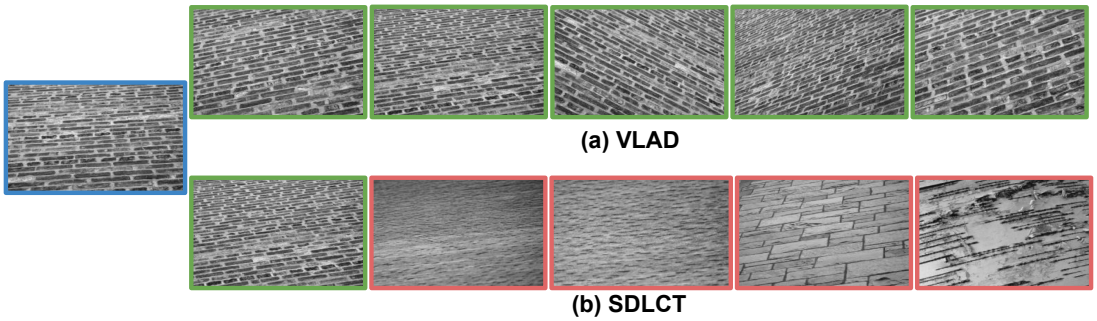

Figura 6.5: Exemplo de consulta da coleção TEXTURE. A consulta está sendo apresentada à esquerda. A primeira linha apresenta as cinco primeiras respostas retornadas para o VLAD (a) e a segunda linha apresenta as cinco primeiras respostas retornadas para o SDLCT. Em verde resultados considerados relevantes. Em vermelho resultados não considerados relevantes.

TEXTURE seguida pelas cinco primeiras respostas retornadas para o método VLAD e para o método SDLCT. A aplicação dos algoritmos de detecção e de representação dos

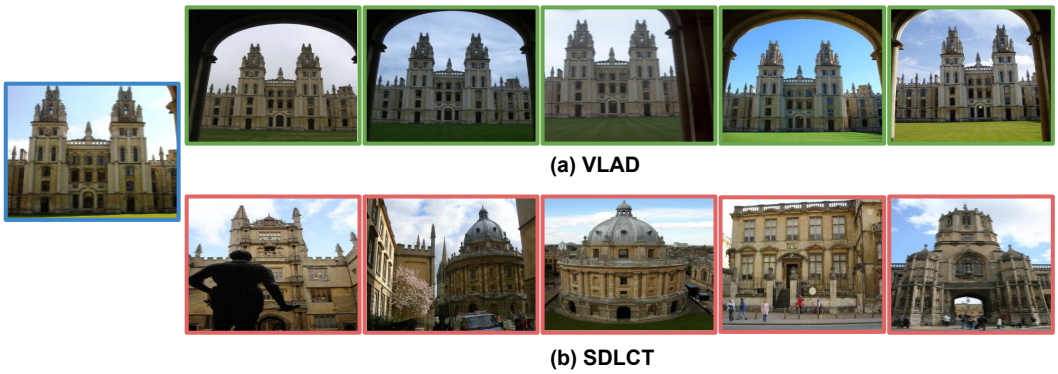

Figura 6.6: Exemplo de consulta da coleção OXFORD. A consulta está sendo apresentada à esquerda. A primeira linha apresenta as cinco primeiras respostas retornadas para o VLAD (a) e a segunda linha apresenta as cinco primeiras respostas retornadas para o SDLCT. Em verde resultados considerados relevantes. Em vermelho resultados não considerados relevantes.

pontos de interesse foi fundamental para o alto desempenho dos métodos C-BoVW na coleção TEXTURE. A tarefa principal nessa coleção é exclusivamente a detecção de padrões de textura em imagens disponibilizadas em uma escala de cinza. Esses algoritmos foram capazes de reconhecer os padrões de textura de forma isolada e de fazer o *matching* (associação) entre esses padrões. Os métodos S-BoVW adotam uma abordagem de representação baseada em blocos fixos e não conseguem detectar nem representar pontos de interesse de forma isolada.

Ao analisar os resultados na coleção OXFORD, foi possível verificar que nessa coleção existem muitas imagens semelhantes às imagens da consulta em termos de cor e de textura, mas que não são consideradas relevantes. Isso porque na OXFORD a tarefa específica é de detecção de objetos, mais especificamente o reconhecimento de prédios históricos. A Figura 6.6 apresenta um exemplo no qual a consulta é um castelo, e na coleção OXFORD existem muitas imagens de castelos e de construções antigas com cores, texturas e até arquiteturas similares. Os resultados providos pelos métodos S-BoVW, como o SDLCT retornam imagens de construções que não são relevantes, mas que não são tão diferentes das consultas. Os métodos C-BoVW apresentam resultados melhores do que os S-BoVW nesse aspecto. Observando novamente o exemplo, é possível verificar que o método VLAD, retorna somente imagens relevantes nas cinco primeiras respostas retornadas. Algumas das imagens relevantes apresentam grandes porções de céu e de grama, que não ocorrem na imagem de consulta. É possível verificar também uma região escura em todas as imagens relevantes, enquanto que na imagem de consulta isso não se

apresenta. Ao combinar esses detalhes com o fato de existirem imagens não relevantes com cores e texturas muito próximas à consulta, é possível entender o baixo desempenho do método nessa tarefa. Assim, é possível concluir que as estratégias S-BoVW adotadas até o momento não são adequadas para tarefas onde a consula mostra uma cena, mas o foco da consulta está em um objeto específico a ser reconhecido. As mudanças nas cenas podem mudar drasticamente a representação dos métodos S-BoVW, resultando em um baixo desempenho.

6.2 Análise de Tempo de Processamento

A Figura 6.7 apresenta uma comparação considerando o tempo médio requerido pelos métodos para realizarem o pré-processamneto da imagem de consulta. O pré-processamento consiste na etapa de descrição da imagem da consulta com palavras visuais. No caso dos métodos C-BoVW, essa etapa consiste no processo de detecção dos pontos de interesse na imagem de consulta e na quantização desses pontos em relação às palavras existentes no vocabulário visual utilizado. Já nos métodos S-BoVW, essa etapa consiste na divisão da imagem em blocos e na geração das asinaturas textuais para esses blocos de acordo com a função de mapeamento utilizada.

Ao analisar a Figura 6.7, o primeiro ponto a ser considerado é que no SDLC o tempo de pré-processamento da imagem de consulta é relativamente baixo. Uma das razões para isso é que o SDLC não requer a execução de um processo de detecção de pontos de interesse nem de um processo de quantização, ambos presentes nos métodos C-BoVW. Apesar dos métodos SDLT e SDLCT também dispensarem esses processos, foi possível perceber que os tempos de pré-processamento deles são consideralvelmente superiores aos do SDLC, e com exceção do método $BoVW_{1K}$, também são superiores aos métodos C-BoVW. Quanto a isso, foi possível verificar que o processo de geração dos descritores LBP para cada bloco da imagem foi determinante para o aumento do tempo de pré-processamento. Ao comparar com o SDLC, é fácil verificar que a geração dos histogramas de cor realizada pelo SDLC é muito mais simples e rápida do que a geração dos histogramas de textura do LBP. Nesse sentido, como trabalhos futuros podem ser pesquisadas formas mais eficientes que a do descritor LBP para representar textura. Com relação aos tempos de pré-processamento, pode-se dizer que tais tempos são menos importantes,

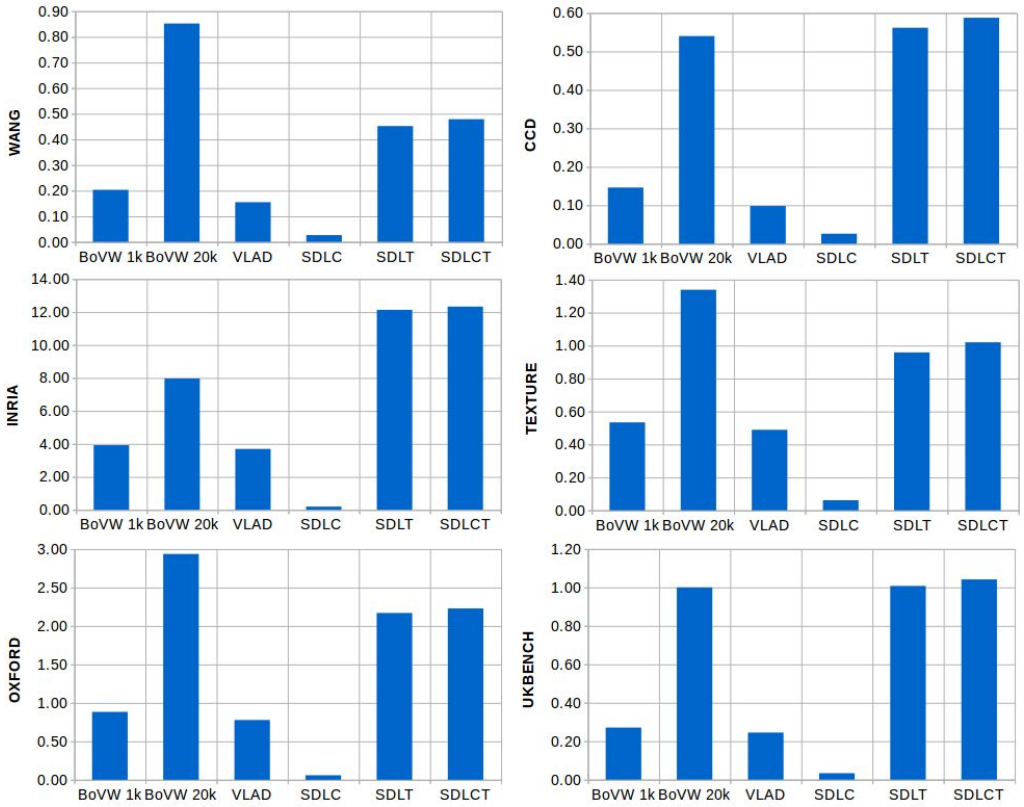

Figura 6.7: Análise de tempo médio (em segundos) para pré-processamento da consulta.

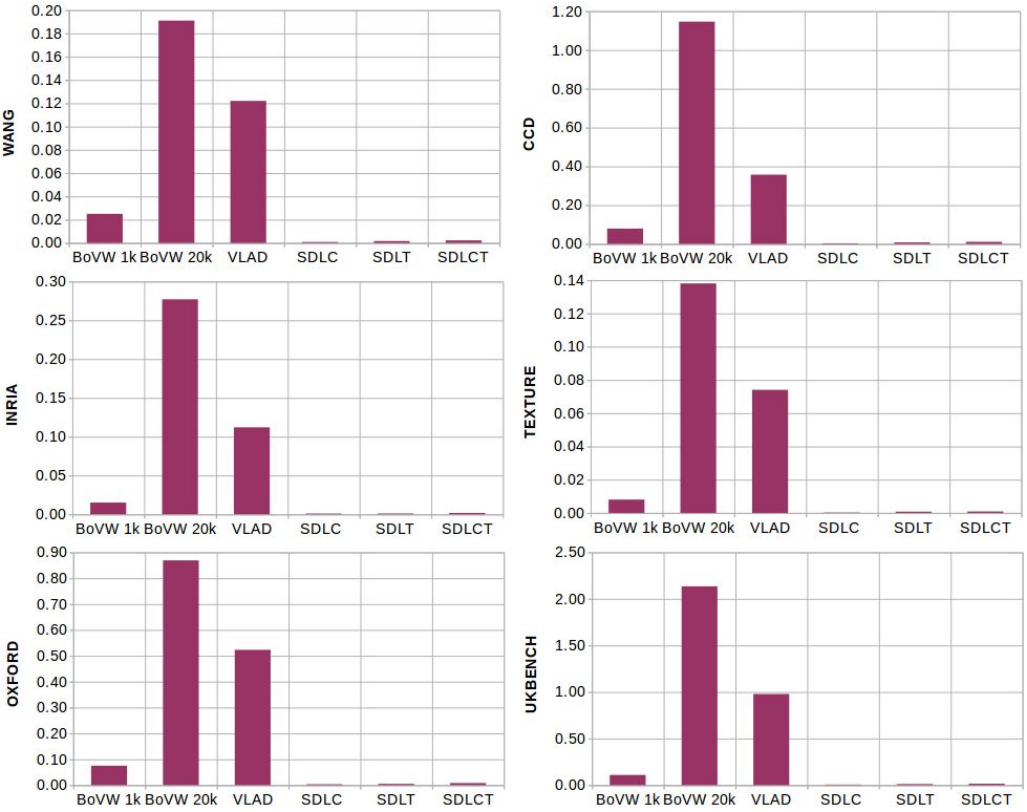

Figura 6.8: Análise de tempo médio (em segundos) para processamento da consulta sem etapa de pré-processamento.

uma vez que não dependem do tamanho da coleção, sendo dependentes unicamente das propriedades da imagem de consulta, como por exemplo a resolução da imagem.

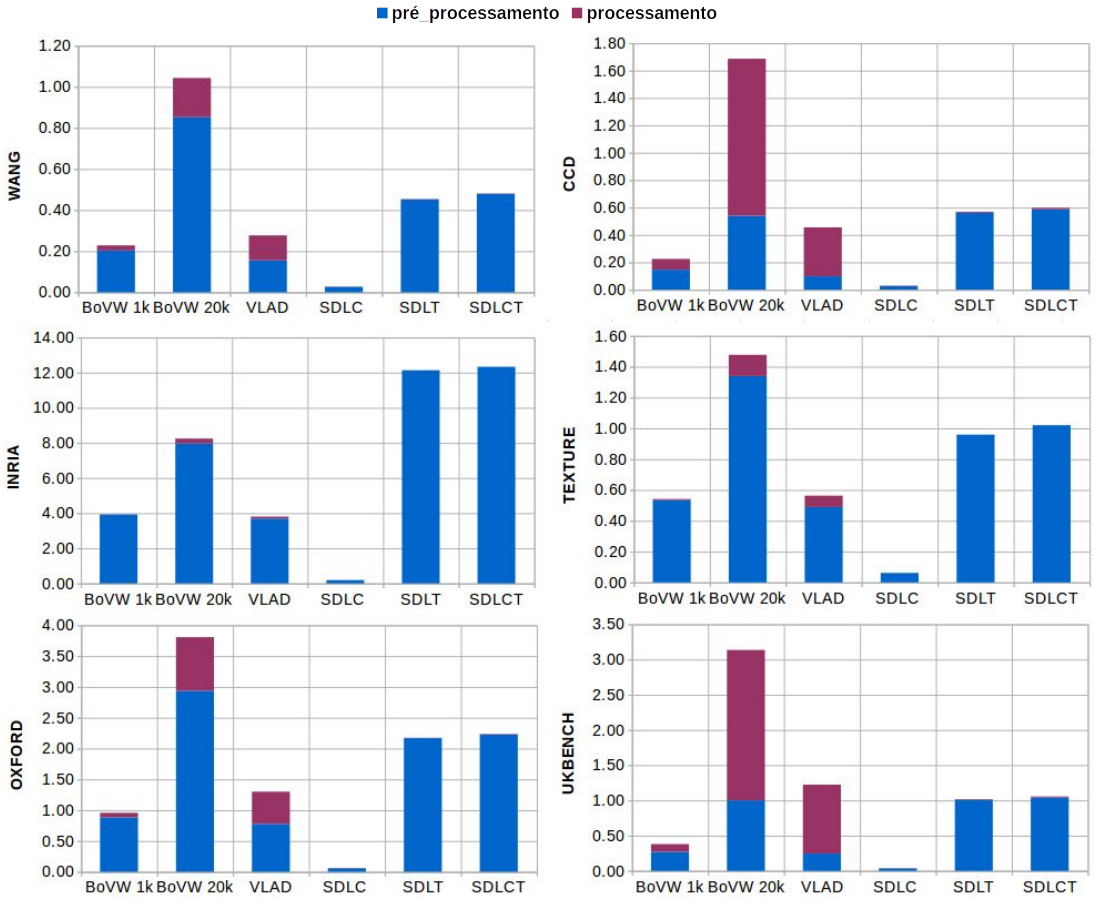

Figura 6.9: Análise de tempo médio total (em segundos) para processamento da consulta.

A Figura 6.8 apresenta uma comparação considerando o tempo médio requerido pelos métodos para realizar o processamento de consultas sem considerar o tempo de pré-processamento. Nesse aspecto, os métodos S-BoVW apresentam tempos bem inferiores em relação aos métodos C-BoVW. Os métodos S-BoVW tem a vantagem de utilizar estratégias de indexação e processamento de consulta consideradas mais eficientes para o processamento de busca textual. Os métodos S-BoVW são consideravelmente mais rápidos que o VLAD e o $BoVW_{20K}$, e apresentam tempos próximos aos alcançados pelo $BoVW_{1K}$. Ao desconsiderar o tempo de pré-processamento, é possível perceber que, embora o SDLT e o SDLCT tenham tempos relativamente superiores quando comparado ao SDLC, os tempos se apresentam competitivos quando comparados aos *baselines* $BoVW_{20K}$ e VLAD.

A Figura 6.9 apresenta uma comparação considerando o tempo médio requerido

pelos métodos para realizar o processamento da consulta incluindo o tempo de pré-processamento. No caso do SDLC, o tempo de pré-processamento é baixo. Nos métodos BoVW, VLAD, SDLT e SDLCT esse tempo é bastante alto, o que afeta negativamente o desempenho geral desses métodos em termos de tempo total de processamento de consultas.

Foi verificado que os tempos de processamento reportados por Jégou e Douze [19] podem ser reproduzidos se as consultas forem submetidas em *batch*, quando o sistema processa a similaridade para todas as consultas de forma simultânea. Entretanto, os experimentos usados para comparar métodos de processamento de consulta não seguem essa estratégia. Esse não é um cenário realístico para processar consultas, uma vez que máquinas de busca reais não processam consultas em *batch*. No entanto, o SDLC e os outros métodos experimentados aqui podem ser beneficiados pelo processamento de consulta em *batches*. Neste trabalho, os tempos reportados para todos os métodos são aqueles alcançados a partir da média de tempo obtida para o processamento de uma consulta por vez.

Capítulo 7

Análise de desempenho de métodos de recuperação textual no cenário S-BoVW

Uma das vantagens de usar o paradigma S-BoVW é o fato de ser possível aplicar em tarefas de busca visual métodos de recuperação considerados eficientes para a busca textual. A ideia é que as palavras visuais que representam uma imagem sejam equiparadas aos termos de um documento textual, sendo possível aplicar os mesmos processos de indexação e processamento de consulta adotados nesses métodos. Várias estratégias têm sido propostas no contexto de recuperação textual para aumentar a eficência dos métodos em termos de tempo de processamento. Neste capítulo, pretende-se verificar como os métodos considerados estado-da-arte para a recuperação textual se comportam no cenário S-BoVW.

7.1 Particularidades da representação textual S-BoVW

As palavras visuais geradas por modelos S-BoVW apresentam propriedades que se diferenciam da realidade das propriedades observadas em modelos de recuperação textual. Diferenças quanto à composição do vocabulário, ao significado das palavras e ao tamanho das consultas podem afetar o desempenho esperado por métodos de recuperação textual no contexto S-BoVW.

Em modelos de recuperação textual, o vocabulário é composto pelas palavras presentes nos documentos que serão indexados. Sendo o vocabulário formado por palavras de uma linguagem natural, como por exemplo português ou inglês, é possível usar al-

gum tipo de conhecimento *a priori* sobre esta linguagem para melhorar os sistemas de recuperação. Um exemplo disso é uso do recurso de *stop-lists*, que são listas que contêm palavras como artigos e preposições que não afetam muito o desempenho dos sistemas de recuperação. Essas listas podem ser facilmente criadas por humanos com o objetivo de otimizar os resultados. Nos modelos S-BoVW, o processo de composição do vocabulário inclui diversas variáveis que podem resultar em vocabulários totalmente diferentes para uma mesma coleção. Além disso, não existe nenhum conhecimento prévio sobre o vocabulário, tornando mais complexa a aplicação de estratégias que se baseiam em conhecimentos *a priori*. Parâmetros como a função de mapeamento, o tipo de particionamento e a quantidade de blocos utilizados para representar as imagens podem alterar consideravelmente o vocabulário final obtido.

Uma palavra textual está associada a conceitos e a objetos que podem ser facilmente reconhecidos por um ser humano. Uma palavra visual não apresenta um sentido claro para um ser humano, pois esta trata-se de um código gerado para a representação de uma imagem. Essa característica evita a aplicação direta de técnicas que tiram vantagens das propriedades semânticas das palavras, como por exemplo o uso de sinônimos, a utilização de *stop-lists* e a aplicação de *stemming*.

O tamanho da consulta é uma diferença importante entre a recuperação textual e a recuperação de imagens baseada em palavras visuais. No caso da recuperação textual, as consultas são geralmente pequenas (cerca de meia dúzia de palavras). No caso da recuperação visual, as consultas são documentos inteiros ou parte de um documento (no caso de uma consulta por uma região específica da imagem), e contém muito mais termos. Na Figura 7.1, é apresentado o tamanho médio das consultas em diferentes coleções a partir da utilização da função de mapeamento SDLC com a configuração sem particionamento ($0R$), 625 blocos e limiar $\varepsilon = 10\%$. É possível verificar que em média as consultas são formadas por centenas de termos, um número bem superior à realidade das buscas textuais. Essa diferença traz muitas consequências que impactam o desempenho dos esquemas de recuperação textual: (i) índices invertidos se tornam menos eficientes do que na recuperação textual, devido ao tamanho das consultas e à presença de ruídos no vocabulário, e (ii) esquemas de pesos resultantes da recuperação probabilística devem ser adaptados uma vez que não provêem os mesmos pesos para termos dos documentos e para os termos da consulta.

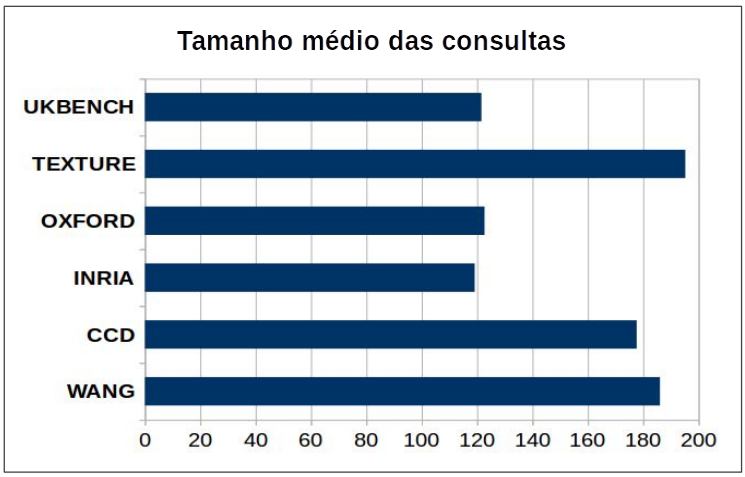

Figura 7.1: SDLC: Tamanho médio das consultas em diferentes coleções.

Outro desafio relacionado à adaptação dos métodos textuais para a realidade S-BoVW é a distribuição do índice. A Figura 7.2 demonstra o tamanho das listas invertidas nas coleções WANG e YAHOO-INRIA com a representação SDLC. Como pode ser observado, a maioria das listas possui até dez documentos, sendo uma boa parcela desse montante composta por listas com apenas uma entrada, nas quais o termo ocorre apenas no documento da consulta. Essa característica também foi confirmada nas outras coleções adotadas neste trabalho. Muitos métodos considerados eficientes na busca textual, aplicam estratégias de poda para desconsiderar documentos da lista com o objetivo de acelerar o processamento sem perder qualidade. A distribuição das listas no cenário S-BoVW pode tornar questionável a adoção de um processamento documento a documento com estratégias de poda.

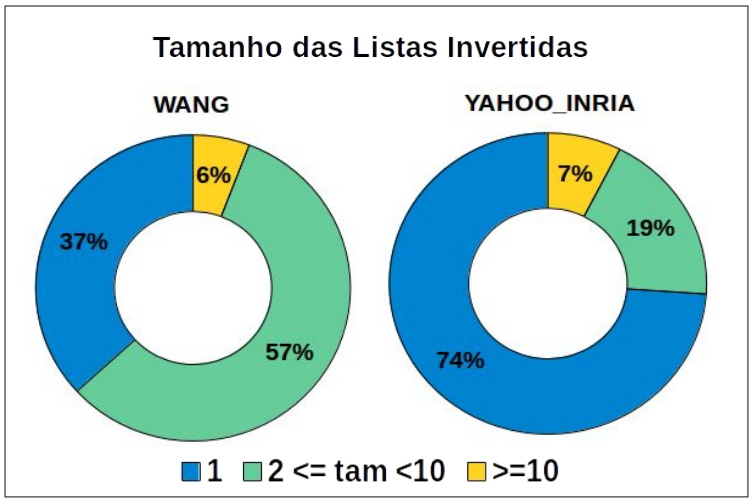

Figura 7.2: SDLC: Tamanho das listas invertidas.

7.2 Experimentos

Para verificar o desempenho dos métodos de recuperação textual no cenário S-BoVW, foram selecionados métodos que são amplamente reconhecidos como eficientes no cenário do processamento de consultas textuais. Os métodos escolhidos foram: WAND [7], BMW [11] e BMW-CSP [10]. Além desses métodos, são apresentados os resultados referentes ao desempenho do processamento termo-a-termo (TAT) e do processamento documento-a-documento (DAT), ambos sem a aplicação de técnicas de poda.

Os experimentos foram realizados com a representação S-BoVW gerada pela função de mapeamento SDLC utilizando uma configuração sem particionamento ($0R$), 625 blocos e limiar $\varepsilon = 10\%$, gerada para as coleções WANG e YAHOO-INRIA.

A Tabela 7.1 apresenta, para as coleções WANG e YAHOO-INRIA, um resumo contendo informações sobre o tamanho da coleção, o tamanho do vocabulário, o número de entradas no índice, o tamanho médio das listas invertidas, quantidade média de termos nas consultas e quantidade de consultas.

Tabela 7.1: WANG e YAHOO-INRIA: Quadro informativo.

	WANG	**YAHOO-INRIA**
Tamanho da Coleção	1000	105307
Tamanho do Vocabulário	17472	685345
Entradas no Índice	178507	17091762
Tamanho Listas Invertidas (média)	5,52	24,94
Termos na Consulta (média)	186,04	119,28
Quantidade de Consultas	50	500

Nos métodos testados, foram utilizadas a função probabilística BM25 [35] e a função do Cosseno adotada no Modelo Vetorial (VSM - *Vector Space Model*) [37] com os pesos $w_3 = (tf \times idf)$, referenciado como Vetorial, e $w_4 = (tf \times idf \times match)$, referenciado como Vetorial-Match.

A Tabela 7.2 apresenta, para as coleções WANG e YAHOO-INRIA, os resultados de P@10 e MAP obtidos a partir do processamento de consultas com diferentes funções de similaridade.

Tabela 7.2: WANG e YAHOO-INRIA: Análise de P@10 e MAP.

	WANG		**YAHOO-INRIA**	
	P@10	**MAP**	**P@10**	**MAP**
BM25	0,83	0,56	0,08	0,37
Vetorial	0,82	0,58	0,09	0,41
Vetorial-Match	0,86	0,59	0,09	0,45

A Tabela 7.3 apresenta, para as coleções WANG e YAHOO-INRIA, os números referentes ao processamento das consultas com diferentes métodos quando aplicada diferentes funções de similaridade, em termos de: número médio de pivôs (#pivôs), número médio de documentos computados (#docs), número médio de blocos acessados (#blocos) e tempo médio de processamento das consultas (em ms).

Tabela 7.3: WANG e YAHOO-INRIA: Resultados do processamento das consultas.

WANG - VSM - Vetorial				
	#pivots	**#docs**	**#blocos**	**tempo (ms)**
WAND	5968	446	115	2,70
BMW	6568	396	115	4,52
BMW-CSP	5931	739	166	7,79
DAT	992	992	115	1,85
TAT	-	992	115	0,79
YAHOO-INRIA - VSM - Vetorial				
	#pivôs	**#docs**	**#blocos**	**tempo (ms)**
WAND	281420	55686	12239	99,33
BMW	532409	29989	12232	226,41
BMW-CSP	566038	37382	22220	377,54
DAT	96481	96481	12239	99,42
TAT	-	96481	12239	50,81
WANG - VSM - Vetorial-Match				
	#pivôs	**#docs**	**#blocos**	**tempo (ms)**
WAND	8589	260	115	3,06
BMW	8754	234	115	6,85
BMW-CSP	6153	735	169	8,61
DAT	992	992	115	1,85
TAT	-	992	115	0,80
YAHOO-INRIA - VSM - Vetorial-Match				
	#pivôs	**#docs**	**#blocos**	**tempo (ms)**
WAND	554197	28587	12237	146,76
BMW	698430	12203	12219	349,26
BMW-CSP	613819	30006	22724	484,22
DAT	96481	96481	12239	96,75
TAT	-	96481	12239	50,51
WANG - BM25				
	#pivôs	**#docs**	**#blocos**	**tempo (ms)**
WAND	6445	425	115	2,33
BMW	6675	406	115	3,19
BMW-CSP	6496	778	163	6,13
DAT	992	992	115	1,89
TAT	-	992	115	0,81
YAHOO-INRIA - BM25				
	#pivôs	**#docs**	**#blocos**	**tempo (ms)**
WAND	519689	28696	12238	124,50
BMW	577968	22267	12235	203,63
BMW-CSP	510579	56683	20334	342,82
DAT	96481	96481	12239	96,44
TAT	-	96481	12239	53,47

Como apresentado na Tabela 7.2, o processamento de consulta usando VSM com Vetorial-Match alcançou os melhores resultados em termos de P@10 e MAP. Por isso, serão apresentados de forma mais detalhada os resultados do processamento de consulta aplicando o VSM com Vetorial-Match. A Figura 7.3 apresenta o tempo de processamento

(em ms) para cada consulta processada e a Figura 7.4 apresenta a média de tempo de processamento (em ms), ambos para as coleções WANG e YAHOO-INRIA.

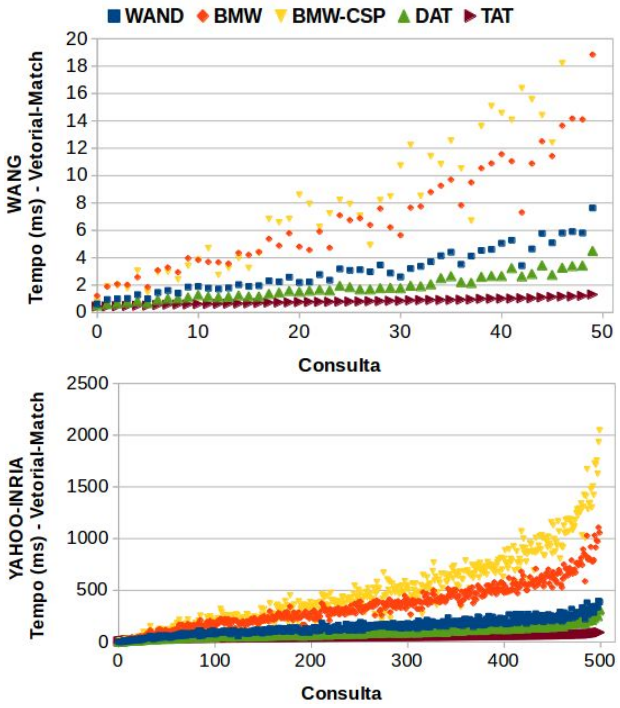

Figura 7.3: Tempo de processamento com Vetorial-Match por consulta.

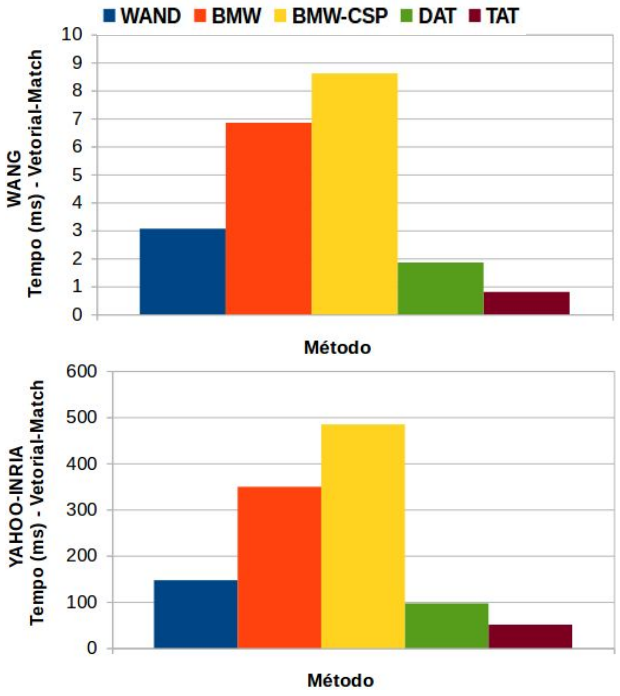

Figura 7.4: Tempo médio de processamento com Vetorial-Match.

7.2.1 Análise dos Resultados

De acordo com as pesquisas existentes na literatura [7, 11, 10], no contexto do processamento de busca textual, o método BMW-CSP é mais rápido que o BMW, enquanto que o BMW é mais rápido que o WAND. Além disso, sabe-se que esses métodos foram propostos como soluções mais eficientes do que o processamento termo-a-termo (TAT) e o processamento documento-a-documento (DAT) sem poda. Entretanto, o que foi observado no cenário S-BoVW foi um resultado inverso.

Nas coleções WANG e YAHOO-INRIA, com as diferentes funções de similaridade, foi verificado que as estratégias de processamento sem poda apresentaram maior agilidade em comparação com as estratégias de processamento com poda. Sendo que, a estratégia de processamento termo-a-termo foi mais eficiente que a estratégia de processamento documento-a-documento. Além disso, o método BMW-CSP apresentou desempenho inferior ao método BMW, que por sua vez obteve desempenho inferior ao WAND.

Uma das razões para esses resultados está relacionada às características das coleções sobre as quais esses métodos foram aplicados. As propriedades da representação textual S-BoVW teve uma forte influência sobre o desempenho obtido por esses métodos. A quantidade de termos da consulta muito maior e as listas invertidas do índice muito mais curtas, fazem com que as estratégias de poda adotadas em métodos como WAND, BMW e BMW-CSP não obtenham ganhos de desempenho em relação às técnicas sem poda. Um número muito grande de termos nas consultas torna as verificações de descarte e de processamento realizadas por esses métodos muito mais caras.

Em um processamento documento-a-documento, as listas invertidas referentes aos termos da consulta devem estar sempre ordenadas entre si de acordo com os seus respectivos documentos correntes. No contexto S-BoVW são muitas listas a serem ordenadas, o que torna o processamento muito mais lento. No caso dos métodos que adotam uma estratégia de poda no processamento documento-a-documento, como o WAND, BMW e BMW-CSP, um documento é selecionado como pivô somente quando a soma dos *MaxScores* das listas anteriores até sua própria lista supera o limiar de poda. Uma vez o documento sendo selecionado como pivô, esse documento deverá ser buscado nas listas anteriores. A cada movimento na lista em busca do pivô, o conjunto das listas deve ser reordenado de acordo com seus documentos correntes. A cada reordenação, a busca pelo pivô recomeça. Todo esse processo de reordenação de listas e busca pelo pivô torna-se

muito caro no cenário S-BoVW, tendo em vista a quantidade expressiva de listas a serem consideradas.

Outro fato a ser considerado é que listas muito curtas provêem um cenário no qual existem poucos documentos a serem saltados. Nesse caso, as verificações realizadas para saltar documentos nas listas tornam-se ineficientes e de certa forma desnecessárias. Como foi apresentado na Tabela 7.2, no contexto S-BoVW a maioria das listas é curta, o que faz com que essas verificações resultem em um desperdício de processamento, uma vez que não produzem ganhos significativos em termos de tempo de processamento. Assim, essas verificações tornam o processamento mais caro do que simplesmente processar todos os documentos das listas.

Com base nessas considerações fica claro concluir a razão que levou o processamento termo-a-termo ser mais eficiente do que todas as outras estratégias analisadas. Em comparação com as outras técnicas, o processamento termo-a-termo é um processamento mais simples que não depende da reordenação frequente das listas e que não realiza verificações que resultam em desperdício de processamento.

Capítulo 8

Conclusão

Este trabalho foi desenvolvido com o objetivo de aprofundar o estudo sobre descritores de imagens baseados em assinatura textual, de forma a identificar formas eficazes e eficientes de aplicá-los em tarefas de recuperação de imagens baseada em conteúdo. Podem ser citadas como contribuições obtidas a partir deste trabalho:

- Definição do paradigma *Signature-based bag of visual words* (S-BoVW), como uma nova categoria de métodos que possiblita a criação de novas funções de mapeamento entre blocos e assinaturas textuais.

- Apresentação de um estudo sobre o impacto dos parâmetros de configuração no desempenho dos métodos S-BoVW, incluindo a realização de experimentos com diferentes funções de similaridade e esquemas de peso. Por meio desse estudo, foi constatado que ao utilizar uma configuração otimizada dos métodos S-BoVW é possível reduzir o tamanho do índice, diminuir o tempo de processamento e não perder qualidade [12].

- Definição de duas novas funções de mapeamento *Sorted Dominant Local Texture* (SDLT) e *Sorted Dominant Local Color and Texture* (SDLCT) que consideram a codificação do conteúdo de textura das imagens. O SDLCT apresentou ganhos significativos em termos de P@10 e MAP em relação aos métodos *Cluster-based bag of visual words* (C-BoVW), na maioria dos cenários experimentados [13].

- Apresentação de um estudo sobre o desempenho de métodos considerados eficientes no processamento de consulta textual no cenário S-BoVW. Como conclusão,

observou-se que as particularidades da coleção gerada pela representação S-BoVW, como o tamanho das consultas e das listas invertidas, fazem com que esses métodos não apresentem ganhos em relação às técnicas tradicionais de processamento termo-a-termo e documento-a-documento.

8.1 Trabalhos Futuros

Como trabalhos futuros, existe a possibilidade de estudar formas mais eficientes para gerar assinaturas textuais com o objetivo de codificar informações de textura das imagens, uma vez que o descritor LBP, além de apresentar o processo de extração de características muito lento, também não apresentou bons resultados em termos de qualidade de resposta, mesmo em coleções específicas de textura.

Outra possibilidade de trabalho futuro é a investigação de soluções que permitam a escolha do método S-BoVW mais adequado a ser utilizado de acordo com características específicas da imagem de consulta. Além disso, esse trabalho poderia se estender no sentido de definir, de acordo com a imagem de consulta, qual seria a melhor configuração a ser utilizada, em termos de tamanho dos blocos, particionamento e limiares em geral.

Um caminho que poderia ser seguido seria a definição de uma função de mapeamento que fosse capaz de codificar conteúdos de forma presentes nas imagens com o intuito de melhorar o desempenho dos métodos S-BoVW em tarefas de detecção de objetos e reconhecimento de cenas.

Por fim, como sugestão para o prosseguimento deste trabalho, poderia ser realizado um estudo visando a proposta de algoritmos específicos para o processamento de consultas que considerem as particularidades existentes nas coleções representadas pelo paradigma S-BoVW, como o tamanho das consultas e das listas invertidas.

Bibliografia

[1] ANH, V. N., AND MOFFAT, A. Pruned query evaluation using pre-computed impacts. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), ACM, pp. 372–379.

[2] BABENKO, A., AND LEMPITSKY, V. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1269–1277.

[3] BAEZA-YATES, R., AND RIBEIRO-NETO, B. *Modern Information Retrieval: The concepts and technology behind search.* 2011.

[4] BAY, H., TUYTELAARS, T., AND VAN GOOL, L. Surf: Speeded up robust features. *Computer Vision* (2006), 404–417.

[5] BOSCH, A., ZISSERMAN, A., AND MUÑOZ, X. Image classification using random forests and ferns. In *2007 IEEE 11th International Conference on Computer Vision* (2007), IEEE, pp. 1–8.

[6] BOSCH, A., ZISSERMAN, A., AND MUNOZ, X. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* (New York, NY, USA, 2007), CIVR '07, ACM, pp. 401–408.

[7] BRODER, A. Z., CARMEL, D., HERSCOVICI, M., SOFFER, A., AND ZIEN, J. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the twelfth international conference on Information and knowledge management* (2003), ACM, pp. 426–434.

[8] CHATZICHRISTOFIS, S. A., ZAGORIS, K., BOUTALIS, Y. S., AND PAPAMARKOS, N. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *International Journal of Pattern Recognition and Artificial Intelligence 24* (2010), 207–244.

[9] DAGLI, C., AND HUANG, T. S. A framework for grid-based image retrieval. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (2004), vol. 2, IEEE, pp. 1021–1024.

[10] DAOUD, C. M., DE MOURA, E. S., CARVALHO, A., DA SILVA, A. S., FERNANDES, D., AND ROSSI, C. Fast top-k preserving query processing using two-tier indexes. *Information Processing & Management* (2016).

[11] DING, S., AND SUEL, T. Faster top-k document retrieval using block-max indexes. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (2011), ACM, pp. 993–1002.

[12] DOS SANTOS, J. M., DE MOURA, E. S., DA SILVA, A. S., CAVALCANTI, J. M. B., DA SILVA TORRES, R., AND VIDAL, M. L. A. A signature-based bag of visual words method for image indexing and search. *Pattern Recognition Letters 65* (2015), 1–7.

[13] DOS SANTOS, J. M., DE MOURA, E. S., DA SILVA, A. S., AND DA SILVA TORRES, R. Color and texture applied to a signature-based bag of visual words method for image retrieval. *Multimedia Tools and Applications* (2016), 1–18.

[14] DOUZE, M., AND JÉGOU, H. The yael library. In *International Conference on Multimedia* (2014), ACM, pp. 687–690.

[15] FRINTROP, S., ROME, E., AND CHRISTENSEN, H. I. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP) 7*, 1 (2010), 6.

[16] HARTIGAN, J. A., AND WONG, M. A. Algorithm as 136: A k-means clustering algorithm. *Applied statistics* (1979), 100–108.

[17] JÉGOU, H., DOUZE, M., AND SCHMID, C. Hamming embedding and weak geometric consistency for large scale image search. In *International Conference on Computer Vision*. Springer, 2008, pp. 304–317.

[18] JÉGOU, H., HARZALLAH, H., AND SCHMID, C. A contextual dissimilarity measure for accurate and efficient image search. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8.

[19] JÉGOU, H., PERRONNIN, F., DOUZE, M., SÁNCHEZ, J., PÉREZ, P., AND SCHMID, C. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34* (2012), 1704–1716.

[20] JIANG, Y.-G., NGO, C.-W., AND YANG, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval* (2007), ACM, pp. 494–501.

[21] KIMURA, P., CAVALCANTI, J., SARAIVA, P., TORRES, R., AND GONÇALVES, M. Evaluating retrieval effectiveness of descriptors for searching in large image databases. *Journal of Information and Data Management 2* (2011), 305–321.

[22] LAZEBNIK, S., SCHMID, C., AND PONCE, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27* (2005), 1265–1278.

[23] LAZEBNIK, S., SCHMID, C., AND PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), vol. 2, IEEE, pp. 2169–2178.

[24] LI, J., AND WANG, J. Z. Automatic linguistic indexing of pictures by a statistical modeling approach. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2003), vol. 25, pp. 1075–1088.

[25] LOWE, D. Object recognition from local scale-invariant features. In *International Conference on Computer Vision* (1999), vol. 2, pp. 1150–1157.

[26] MANJUNATH, B., OHM, J., VASUDEVAN, V., AND YAMADA, A. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology 11*, 6 (2001), 703–715.

[27] MIKOLAJCZYK, K., AND SCHMID, C. Scale & affine invariant interest point detectors. *International journal of computer vision 60*, 1 (2004), 63–86.

[28] NISTER, D., AND STEWENIUS, H. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, IEEE, pp. 2161–2168.

[29] NYMA, A., KANG, M., KWON, Y.-K., KIM, C.-H., AND KIM, J.-M. A hybrid technique for medical image segmentation. *BioMed Research International 2012* (2012).

[30] O'HARA, S., AND DRAPER, B. A. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354* (2011).

[31] OJALA, T., PIETIKÄINEN, M., AND HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition 29*, 1 (1996), 51–59.

[32] PENATTI, O., AND TORRES, R. Eva: an evaluation tool for comparing descriptors in content-based image retrieval tasks. In *Proceedings of the international conference on Multimedia information retrieval* (2010), ACM, pp. 413–416.

[33] PERSIN, M., ZOBEL, J., AND SACKS-DAVIS, R. Filtered document retrieval with frequency-sorted indexes. *JASIS 47*, 10 (1996), 749–764.

[34] PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. Object retrieval with large vocabularies and fast spatial matching. In *Conference on Computer Vision and Pattern Recognition* (2007), IEEE, pp. 1–8.

[35] ROBERTSON, S. E., AND WALKER, S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval* (1994), pp. 232–241.

[36] ROSSI, C., DE MOURA, E. S., CARVALHO, A. L., AND DA SILVA, A. S. Fast document-at-a-time query processing using two-tier indexes. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (2013), ACM, pp. 183–192.

[37] SALTON, G., AND MCGILL, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[38] SARAIVA, P. C., CAVALCANTI, J. M. B., S. DE MOURA, E., GONCALVES, M. A., AND DA S. TORRES, R. A multimodal query expansion based on genetic programming for visually-oriented e-commerce applications. *Information Processin & Management* (2016).

[39] SIVIC, J., AND ZISSERMAN, A. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision* (2003), IEEE, pp. 1470–1477.

[40] STEHLING, R., NASCIMENTO, M., AND FALCAO, A. Techniques for color-based image retrieval. *Multimedia Mining* (2002), 61–82.

[41] STROHMAN, T., AND CROFT, W. B. Efficient document retrieval in main memory. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), ACM, pp. 175–182.

[42] TAKALA, V., AHONEN, T., AND PIETIKÄINEN, M. Block-based methods for image retrieval using local binary patterns. *Image Analysis* (2005), 13–181.

[43] TORRES, R., AND FALCÃO, A. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada 2*, 13 (2006), 161–185.

[44] TORRES, R., ZEGARRA, J., SANTOS, J., FERREIRA, C., PENATTI, O., ANDALÓ, F., AND ALMEIDA JR, J. Recuperaçao de imagens: Desafios e novos rumos. In *XXXV Seminário Integrado de Software e Hardware (SEMISH)* (2008), SBC, pp. 223–237.

[45] VEDALDI, A., AND FULKERSON, B. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia* (New York, NY, USA, 2010), MM '10, ACM, pp. 1469–1472.

[46] VIDAL, M. L., CAVALCANTI, J. M., DE MOURA, E. S., DA SILVA, A. S., AND DA SILVA TORRES, R. Sorted dominant local color for searching large and heterogeneous image databases. In *International Conference on Pattern Recognition* (2012), IEEE, pp. 1960–1963.

[47] WAN, J., WANG, D., HOI, S. C. H., WU, P., ZHU, J., ZHANG, Y., AND LI, J. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), ACM, pp. 157–166.

[48] WILCOXON, F. Individual comparisons by ranking methods. *Biometrics 1* (1945), 80–83.

[49] ZHANG, S., TIAN, Q., HUA, G., HUANG, Q., AND LI, S. Descriptive visual words and visual phrases for image applications. In *ACM Multimedia* (2009), pp. 75–84.

[50] ZHU, S., AND YANG, J. A novel image retrieval approach based on integer and block color distribution. *Journal of Computational Information Systems 7*, 2 (2011), 593–598.