



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro

Lucas Pascotti Valem

Combinação Seletiva Não Supervisionada de Listas Ranqueadas Aplicada à Busca de Imagens pelo Conteúdo

Rio Claro - SP

2019

Lucas Pascotti Valem

**Combinação Seletiva Não Supervisionada de Listas
Ranqueadas Aplicada à Busca de Imagens pelo
Conteúdo**

Orientador: Prof. Dr. Daniel Carlos Guimarães Pedronette

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Rio Claro.

Financiadora: FAPESP – Proc. 2017/02091-4.

Rio Claro - SP

2019

V151c

Valem, Lucas Pascotti

Combinação Seletiva Não Supervisionada de Listas Ranqueadas Aplicada à Busca de Imagens pelo Conteúdo / Lucas Pascotti Valem. -- Rio Claro, 2019
106 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro

Orientador: Daniel Carlos Guimarães Pedronette

1. Ciência da Computação. 2. Recuperação de Informações. 3. Recuperação de Imagens por Conteúdo. 4. Seleção e Combinação de Listas Ranqueadas. 5. Aprendizado Não Supervisionado. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

ERRATA

VALEM, Lucas Pascotti. **Combinação Seletiva Não Supervisionada de Listas Ranqueadas Aplicada à Busca de Imagens pelo Conteúdo**. 2019. 106 f. Dissertação (Mestrado em Ciência da Computação) - Instituto de Biociências Letras e Ciências Exatas, Universidade Estadual Paulista (Unesp), São José do Rio Preto, 2019.

Folha	Linha	Onde se lê	Leia-se
1	14	Financiadora: FAPESP – Proc. 2017/02091-4.	Financiadores: FAPESP e CAPES – Proc. FAPESP nº 2017/02091-4.
3	13	Financiadora: FAPESP – Proc. 2017/02091-4.	Financiadores: FAPESP e CAPES – Proc. FAPESP nº 2017/02091-4.
4	7	Agradeço à FAPESP pela concessão da bolsa de pesquisa, sob o processo nº 2017/02091-4, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).	Agradeço à FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo e a CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pela concessão da bolsa de pesquisa, sob o processo FAPESP nº 2017/02091-4.

IBILCE/UNESP

SEÇÃO DE PÓS-GRADUAÇÃO

Aprovado pelo Conselho do Programa em
reunião de 1/1 Encaminhado-se a
STPG para providências

São José do Rio Preto, 19 de 07 de 2019

x 

Prof. Dr. Adriano Mauro Cansia
Vice - Coordenador do Programa de
Pós-Graduação em Ciências da
Computação

Lucas Pascotti Valem

Combinação Seletiva Não Supervisionada de Listas Ranqueadas Aplicada à Busca de Imagens pelo Conteúdo

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de Rio Claro.

Financiadora: FAPESP – Proc. 2017/02091-4.

Comissão Examinadora

- Prof. Dr. Daniel Carlos Guimarães Pedronette (Orientador)
Departamento de Estatística, Matemática Aplicada e Computação
Universidade Estadual Paulista - UNESP
- Prof. Dr. Fabio Augusto Faria
Instituto de Ciência e Tecnologia (ICT)
Universidade Federal de São Paulo (UNIFESP)
- Prof. Dr. Fabricio Aparecido Breve
Departamento de Estatística, Matemática Aplicada e Computação
Universidade Estadual Paulista - UNESP

Rio Claro - SP

01 de Março de 2019

Agradecimentos

Agradeço primeiramente a Deus pela vida, toda saúde e disposição.

Aos meus pais e familiares, pelo amor e apoio incondicional.

Ao meu orientador, pela orientação, apoio e confiança.

Agradeço à universidade e a todo o corpo docente do programa de pós-graduação.

Aos amigos e colegas, pelo incentivo e apoio.

Agradeço à FAPESP pela concessão da bolsa de pesquisa, sob o processo nº 2017/02091-4, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Resumo

Em virtude do grande crescimento das coleções multimídia e da evolução das tecnologias no decorrer dos últimos anos, buscar por imagens considerando o conteúdo visual das mesmas se tornou uma tarefa de extrema importância. Os sistemas de Recuperação de Imagens pelo Conteúdo (CBIR) são a principal solução dentro desse cenário e representam um campo de pesquisa cada vez mais ativo. Atualmente, esses sistemas empregam diferentes abordagens para execução de suas tarefas, dentre elas estão diferentes descritores e métodos de aprendizado. Contudo, mensurar a similaridade de maneira eficaz ainda é um desafio, principalmente devido ao conhecido problema da lacuna semântica. Como diferentes listas ranqueadas possuem resultados distintos e muitas vezes complementares, uma estratégia que tem ganhado significativa atenção é a de combinar essas listas. Este trabalho propõe três diferentes métodos completamente não supervisionados para seleção e combinação de listas ranqueadas. As abordagens foram avaliadas em cinco coleções de imagens e diversos descritores. Foram obtidos resultados comparáveis ou superiores ao estado-da-arte na maioria dos cenários analisados.

Palavras-chave: Recuperação de Informações, Recuperação de Imagens por Conteúdo, Seleção e Combinação de Características, Seleção e Combinação de Listas Ranqueadas, Reconhecimento de Padrões, Aprendizado Não Supervisionado, Rerankeamento.

Abstract

Due to the evolution of technologies to store and share images, finding effective methods to index and retrieve this type of information is indispensable. The CBIR (Content-Based Image Retrieval) systems are the main solution for image retrieval tasks. These systems consist in the use of different descriptors, supervised learning methods, and more recently unsupervised learning methods. However, accurately estimating the similarity between images is still a challenging task, mainly due to the well-known semantic gap problem. As different ranked lists present different effectiveness results, an interesting methodology is to combine these ranked lists. This work proposes three different unsupervised methods for selecting and combining ranked lists. The approaches were evaluated in five different image collections and several descriptors. The methods achieved results comparable or superior to the state-of-the-art in most of the evaluated scenarios.

Keywords: Information Retrieval, Content-Based Image Retrieval, Feature Selection and Fusion, Pattern Recognition, Unsupervised Learning, Rank-aggregation, Re-ranking.

Lista de Figuras

Figura 1 – Arquitetura típica de um sistema CBIR.	18
Figura 2 – Ilustração do processo de extração de características e da lacuna semântica.	20
Figura 3 – Esquema de um sistema CBIR considerando a combinação de características.	27
Figura 4 – Exemplo de etapas com presença de <i>early fusion</i>	28
Figura 5 – Exemplo de etapas com presença de <i>late fusion</i>	28
Figura 6 – Exemplo de gene na programação genética.	37
Figura 7 – Ilustração da coocorrência de vizinhos recíprocos no grafo de vizinhança.	43
Figura 8 – Etapas do algoritmo CPRR.	46
Figura 9 – Visão geral do método USRF.	47
Figura 10 – Ilustração das etapas que compõem o USRF.	49
Figura 11 – Ilustração do algoritmo de seleção proposto.	51
Figura 12 – Impacto dos parâmetros k e β na média aritmética ponderada considerando o MAP dos cinco primeiros resultados selecionados para cada coleção.	54
Figura 13 – Avaliação do parâmetro β considerando cenários aleatórios com diferentes números de descritores.	56
Figura 14 – Avaliação do parâmetro L_R nos resultados de seleção.	57
Figura 15 – Distribuição dos pares considerando a métrica de seleção comparada ao MAP nos cenários personalizados.	58
Figura 16 – Avaliação da seleção proposta (USRF) para pares de ranqueadores nos cenários personalizados.	59
Figura 17 – Avaliação da seleção proposta (USRF) para pares de ranqueadores nos cenários completos.	60
Figura 18 – Avaliação da seleção proposta (USRF) para pares de ranqueadores em diferentes cenários.	61
Figura 19 – Análise do tamanho das combinações em relação aos resultados de seleção para cada uma das coleções considerando todos os ranqueadores (cenários completos).	63
Figura 20 – Principais etapas que compõem o método RGSF.	64
Figura 21 – Ilustração das iterações realizadas para a seleção dos ranqueadores pelas componentes conexas.	66
Figura 22 – Impacto do parâmetro k no MAP da combinação selecionada para as coleções com maior número de imagens por classe.	68
Figura 23 – Impacto do parâmetro k no MAP da combinação selecionada para as coleções com menor número de imagens por classe.	69

Figura 24 – Avaliação do método RGSF nos cenários personalizados.	70
Figura 25 – Visão geral do <i>framework</i> UGAF-RSF.	72
Figura 26 – Exemplo de cromossomo binário.	73
Figura 27 – Comparação entre as diferentes funções de aptidão.	75
Figura 28 – Análise do tamanho da população.	75
Figura 29 – Avaliação da taxa de cruzamento.	76
Figura 30 – Avaliação da taxa de mutação.	76
Figura 31 – Comparação das estratégias evolutivas.	76
Figura 32 – Análise da taxa de elitismo máxima.	76
Figura 33 – Análise do tamanho da vizinhança k	77
Figura 34 – Dois exemplos de resultados visuais que ilustram o impacto da seleção e combinação na coleção UKBench.	83

Lista de Tabelas

Tabela 1 – Exemplos de métricas de distância.	19
Tabela 2 – Exemplos de operações aritméticas simples para obtenção de F'	29
Tabela 3 – Algoritmos de combinação da família Comb*.	32
Tabela 4 – Coleções de imagens utilizadas na avaliação experimental.	40
Tabela 5 – Descritores de imagens utilizados na avaliação experimental.	40
Tabela 6 – Descritores considerados nos cenários personalizados.	41
Tabela 7 – Métodos considerados como <i>baselines</i> para <i>late fusion</i>	42
Tabela 8 – Análise do tamanho dos conjuntos de combinações.	48
Tabela 9 – Média ponderada do MAP para os cinco primeiros pares selecionados considerando diferentes combinações de métricas e $\beta = 1$	55
Tabela 10 – MAP das três primeiras combinações selecionadas considerando diferentes tamanhos nos cenários personalizados e completos.	62
Tabela 11 – Especificação das combinações selecionadas para cada uma das coleções nos cenários personalizados e completos.	63
Tabela 12 – MAP da combinação selecionada para as coleções em diferentes cenários.	70
Tabela 13 – Combinações mais frequentemente selecionadas e a média do MAP para cada uma das coleções.	77
Tabela 14 – Comparação com métodos de <i>late fusion</i> nos cenários personalizados.	79
Tabela 15 – Comparação com métodos de <i>early fusion</i>	80
Tabela 16 – Comparação com diversos métodos de pós-processamento na coleção MPEG-7 (<i>Bull's eye score - Recall@40</i>).	81
Tabela 17 – Comparação com o estado-da-arte na coleção Holidays (MAP).	82
Tabela 18 – Comparação com o estado-da-arte na coleção UKBench (<i>N-S score</i>).	83
Tabela 19 – Métodos de aprendizado não supervisionado utilizados para o experimento.	84
Tabela 20 – Resultados de MAP para combinação de métodos de aprendizado não supervisionado.	85
Tabela 21 – Tempo de execução dos métodos propostos nos cenários completos.	85
Tabela 22 – Análise de eficácia e eficiência dos métodos propostos.	86

Lista de Abreviaturas

ACC	<i>Color Autocorrelogram Descriptor</i>
AIR	<i>Articulation-Invariant Representation Descriptor</i>
AP	<i>Average Precision</i>
ASC	<i>Aspect Shape Context Descriptor</i>
BAS	<i>Beam Angle Statistics Descriptor</i>
BIC	<i>Border/Interior Pixel Classification Descriptor</i>
BoW	<i>Bag of Words</i>
BoVW	<i>Bag of Visual Words</i>
BRIEF	<i>Binary Robust Independent Elementary Features</i>
CBIR	<i>Content-Based Image Retrieval</i>
CEDD	<i>Color and Edge Directivity Descriptor</i>
CLD	<i>Color Layout Descriptor</i>
COMO	<i>Compact Composite Moment-Based Descriptor</i>
CPRR	<i>Cartesian Product of Ranking References</i>
CCA	<i>Canonical Correlation Analysis</i>
CNN	<i>Convolutional Neural Networks</i>
COMO	<i>Compact Composite Moment-Based Descriptor</i>
DCA	<i>Discriminant Correlation Analysis</i>
DCCA	<i>Deep Canonical Correlation Analysis</i>
DCNN	<i>Deep Convolutional Neural Networks</i>
EHD	<i>Edge Histogram Descriptor</i>
FCTH	<i>Fuzzy Color and Texture Histogram</i>
FOH	<i>Fuzzy Opponent Histogram</i>
GIST	<i>Global Image Descriptor for low-dimensional features</i>
HoG	<i>Histogram of Oriented Gradients</i>
HSV	Sistema de cores formado pelas componentes <i>hue</i> (matiz), <i>saturation</i> (saturação) e <i>value</i> (brilho).
HSVD3	<i>Three-dimensional HSV Histogram</i>
IDSC	<i>Inner Distance Shape Context Descriptor</i>
JCD	<i>Joint Composite Descriptor</i>
LASSO	<i>Least Absolute Shrinkage and Selection Operator</i>
LBP	<i>Local Binary Patterns</i>
LDA	<i>Linear Discriminant Analysis</i>
MAP	<i>Mean Average Precision</i>
MCFS	<i>Muti-cluster Feature Selection</i>
MKL	<i>Multiple Kernel Learning</i>
MRA	<i>Median Rank Aggregation</i>
NDFS	<i>Nonnegative Discriminative Feature Selection</i>

OPF	<i>Optimum-Path Forest</i>
ORB	<i>Oriented Fast and Rotated BRIEF</i>
PCA	<i>Principal Component Analysis</i>
PCC	<i>Particle Competition and Cooperation</i>
PG	Programação Genética
PHOG	<i>Pyramidal Histogram of oriented gradients</i>
RBO	<i>Rank-biased Overlap</i>
RGB	<i>Red, Green, Blue</i>
RI	Recuperação de Informações
SCD	<i>Scalable Color Descriptor</i>
SCH	<i>Simple Color Histogram</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SPACC	<i>Spatial Pyramid Color Autocorrelogram Descriptor</i>
SPCEDD	<i>Spatial Pyramid CEDD</i>
SPEC	<i>Spectral Regression</i>
SPFCTH	<i>Spatial Pyramid FCTH</i>
SPJCD	<i>Spatial Pyramid JCD</i>
SPLBP	<i>Spatial Pyramid LBP</i>
SS	<i>Segment Saliences Descriptor</i>
SURF	<i>Speeded-Up Robust Features</i>
SVM	<i>Support Vector Machines</i>
UDFS	<i>Unsupervised Discriminative Feature Selection</i>
UDLF	<i>Unsupervised Distance Learning Framework</i>
USRF	<i>Unsupervised Selective Rank Fusion</i>
RGSF	<i>Ranker Graph for Selection and Fusion</i>
UGAF-RSF	<i>Unsupervised Genetic Algorithm Framework for Rank Selection and Fusion</i>
VOC	<i>Vocabulary Tree Descriptor</i>

Lista de Símbolos

\mathcal{C}	Coleção de imagens.
\mathcal{N}	Tamanho da coleção de imagens.
\mathcal{A}	Matriz quadrada de distância ou similaridade.
\mathcal{T}	Conjunto das listas ranqueadas de todas as imagens de uma coleção.
L	Tamanho das listas ranqueadas das imagens de consulta.
τ_i	Lista ranqueada da imagem de índice i .
$\tau_q(i)$	Posição da imagem i na lista ranqueada da imagem de consulta q .
f_s	Função de seleção.
R_i	Ranqueador de índice i .
$\tau_{i,q}$	Lista ranqueada da imagem de índice q calculada pelo ranqueador i .
\mathcal{T}_i	Conjunto de listas ranqueadas produzidas pelo ranqueador R_i .
\mathfrak{R}	Conjunto de ranqueadores fornecidos como entrada do método.
m	Tamanho do conjunto \mathfrak{R} .
γ	Função que calcula a estimativa de eficácia.
λ	Função que calcula a correlação.
th_γ	Limiar (<i>threshold</i>) para estimativa de eficácia.
th_λ	Limiar (<i>threshold</i>) para correlação.
n	Número de ranqueadores que compõem uma combinação.
\mathfrak{X}_n	Combinação de n ranqueadores.
\mathfrak{X}_{cc_i}	Combinação fornecida pela componente conexa de índice i .
\mathfrak{X}'	Combinação de ranqueadores após a etapa de agregação.
\mathfrak{X}_n^*	Combinação de tamanho n selecionada pelo método.
\mathfrak{X}^*	Combinação selecionada pelo método.
C_n	Conjunto de combinações em que cada combinação é de tamanho n .
τ_n^R	Lista ordenada das combinações de tamanho n . Também referenciada como lista de seleção.
$\tau_n^R(\mathfrak{X}_n^i)$	Posição da combinação \mathfrak{X}_n^i na lista de seleção τ_n^R .
L_R	Tamanho das listas de seleção τ_n^R .
\mathcal{S}	Conjunto de seleção de todas as possíveis combinações de ranqueadores.
\mathcal{S}_p	Conjunto de seleção de pares de ranqueadores.
S	Lista de combinações selecionadas pelo RGSF.
S'	Lista de combinações selecionadas pelo RGSF após a etapa de agregação.
w	Métrica de seleção de combinações de ranqueadores.
w_p	Métrica de seleção de pares de ranqueadores.
β	Peso ou relevância da correlação na métrica de seleção.
k	Tamanho do conjunto de vizinhança.
T	Número de iterações.

Sumário

1	INTRODUÇÃO	13
2	ASPECTOS TEÓRICOS	17
2.1	Recuperação de Informações	17
2.1.1	Sistemas de Recuperação de Imagens pelo Conteúdo	17
2.1.2	Descritores de Imagens	19
2.1.3	Definição Formal	22
2.1.4	Métricas de Eficácia	23
2.2	Aprendizado de Máquina	24
2.2.1	Aprendizado Supervisionado	25
2.2.2	Aprendizado Não Supervisionado	25
2.2.3	Aprendizado Semi-Supervisionado	26
3	TRABALHOS RELACIONADOS	27
3.1	Categorização das Abordagens de Combinação	27
3.1.1	<i>Early Fusion</i>	27
3.1.2	<i>Late Fusion</i>	28
3.2	Abordagens de Combinação	29
3.2.1	Combinação Aritmética Não Treinável	29
3.2.2	Combinação Conservativa	29
3.2.3	Agregação de Listas Ranqueadas	31
3.2.4	Combinação de Núcleos	33
3.2.5	Abordagens Estatísticas	35
3.2.6	Combinação baseada em Programação Genética	36
3.3	Abordagens Recentes	37
4	PROTOCOLO EXPERIMENTAL	39
4.1	Coleções e Descritores de Imagens	39
4.2	<i>Baselines</i>	41
4.3	Métricas para Seleção	42
4.3.1	Estimativas de Eficácia	42
4.3.2	Métricas de Correlação	44
4.4	Método de Agregação	46
5	SELEÇÃO POR MÉTRICAS PAR-A-PAR	47
5.1	Descrição do Método	48

5.1.1	Estratégia de Seleção	48
5.1.1.1	Seleção de Pares de Ranqueadores	49
5.1.1.2	Seleção de Conjuntos de Ranqueadores	51
5.2	Avaliação Experimental	53
5.2.1	Análise dos Parâmetros	53
5.2.2	Resultados de Seleção	57
5.2.2.1	Seleção de Pares	57
5.2.2.2	Seleção de Conjuntos de Ranqueadores	61
6	SELEÇÃO POR GRAFOS E COMPONENTES CONEXAS	64
6.1	Combinação de Ranqueadores pelas Componentes Conexas	65
6.2	Avaliação Experimental	68
6.2.1	Análise dos Parâmetros	68
6.2.2	Resultados de Seleção	69
7	SELEÇÃO POR ALGORITMOS GENÉTICOS	72
7.1	Descrição do Método	72
7.1.1	Representação dos Cromossomos	73
7.1.2	Agregação Não Supervisionada	73
7.1.3	Função de Aptidão Não Supervisionada	73
7.1.4	Seleção de Cromossomos	73
7.1.5	Operações de Cruzamento	74
7.1.6	Operações de Mutação	74
7.2	Avaliação Experimental	74
7.2.1	Análise dos Parâmetros	74
8	DISCUSSÕES, COMPARAÇÕES E ANÁLISES	79
8.1	Avaliação Comparativa	79
8.1.1	Comparação com <i>Baselines</i> de <i>Late Fusion</i>	79
8.1.2	Comparação com <i>Baselines</i> de <i>Early Fusion</i>	80
8.1.3	Comparação com Estado-da-arte	80
8.2	Análise Visual	82
8.3	Combinação de Métodos de Aprendizado Não Supervisionado	83
8.4	Discussões	85
8.5	Contribuições e Publicações	86
9	CONCLUSÕES	88
9.1	Trabalhos Futuros	88
	REFERÊNCIAS	90

1 Introdução

O grande crescimento na quantidade de dispositivos móveis disponíveis aos usuários e suas capacidades de não somente consumir como também produzir conteúdo geram um aumento crescente na quantidade de imagens e das coleções multimídia em geral [1]. Os sistemas de recuperação de informações são baseados na ideia de recuperar as informações mais similares a partir de um objeto de entrada, fornecendo uma lista ranqueada (*ranked list*) com os elementos mais relevantes no topo da lista. A recuperação de informações se iniciou considerando busca textual e tem evoluído de maneira considerável nos últimos anos, fornecendo hoje resultados já bastante satisfatórios em grande parte dos cenários existentes. Apesar da eficácia e desempenho dos sistemas de recuperação textual terem atingido um bom grau de maturidade, ainda há desafios na área, especialmente em aplicações de recuperação de imagens e conteúdos multimídia, devido a riqueza de conteúdo e subjetividade na interpretação dos mesmos [2].

Devido ao surgimento de novas tecnologias, a busca de imagens se tornou uma necessidade imprescindível. Inicialmente, a metodologia para recuperação de conteúdo multimídia consistia no uso de metadados (*metadata*) associados aos arquivos como por exemplo, geolocalização, tempo (*timestamp*), palavras-chave, marcações, ou até mesmo descrições. A partir desses metadados, uma busca textual era aplicada [3]. Catalogar manualmente uma grande coleção de imagens, além de ser um trabalho extremamente custoso, muitas vezes não é eficaz devido a grande quantidade de informações e até mesmo ao grau de subjetividade do conteúdo envolvido. Além disso, é importante considerar que os metadados associados a uma imagem muitas vezes não descrevem de maneira completa o que está sendo representado, o que pode ocasionar em ambiguidades.

Dessa forma, a fim de explorar o conteúdo visual das coleções de imagens e amenizar o impacto dos problemas citados, os Sistemas de Recuperação de Imagem pelo Conteúdo (CBIR) são propostos e possuem as mais diferentes aplicações (diagnóstico de doenças [4], reconhecimento facial [5], sensoriamento remoto [6], identificação de objetos [7], entre outras). Originalmente, esses sistemas são unicamente baseados no uso de descritores. Um descritor tem como objetivo definir uma função, capaz de extrair um vetor numérico a partir de uma imagem de entrada. Tal vetor atua como uma forma de representação ou identificação (“*fingerprint*”) para aquela imagem. Os descritores visuais consideram características visuais como cor, textura, contorno e outros. Esse tem sido um campo de pesquisa muito ativo nos últimos 20 anos [8, 9, 1].

Uma das maiores limitações do uso de descritores se deve a lacuna semântica (“*semantic gap*”) [10] presente entre as características de baixo e alto nível. Com a

representação do vetor de características, conceitos e significados originalmente contidos na imagem podem ser perdidos ou apenas parcialmente caracterizados. Atualmente, há sistemas de recuperação de imagens eficazes para finalidades específicas. No entanto, um sistema de busca de imagens para propósito geral que seja maduro constitui uma tarefa desafiadora, mesmo considerando as técnicas de estado-da-arte [1]. Na maioria dos casos, pode-se dizer que quanto mais específico o domínio do sistema, mais fácil de ser desenvolvido [8]. Dessa forma, diferentes métodos de aprendizado têm sido propostos com a finalidade de minimizar o problema da lacuna semântica.

Técnicas supervisionadas, como a realimentação de relevância (*relevance feedback*) [11], surgiram como uma forma de aprimorar os resultados. Tais técnicas são baseadas na interação com o usuário, o qual submete ao sistema uma imagem de interesse e o sistema retorna as primeiras k imagens consideradas mais relevantes. A partir disso, o usuário informa quais das imagens retornadas são relevantes e quais não são. Dessa forma, o sistema CBIR pode atribuir (“aprender”) novos pesos de similaridade ou distância entre as imagens para uma próxima busca.

Recentemente, uma nova categoria de descritores foi proposta e vem sendo continuamente utilizada e desenvolvida através do uso de *Deep Convolutional Neural Networks* (CNN) [12]. São utilizadas diversas camadas de informação interconectadas para tarefas de reconhecimento de padrões e aprendizado de características (*feature learning tasks*) [13]. O aspecto mais interessante desse paradigma de *deep learning* é que as características não precisam necessariamente ser extraídas a partir dos dados originais (*raw data*) de antemão, mas os dados originais são processados pela rede produzindo uma representação interna dos dados que pode ser utilizada como uma representação de características. Um dos trabalhos pioneiros em CNN para recuperação e classificação de imagens [12], mostra que as camadas superiores da rede podem ser utilizadas como descritores visuais em diferentes cenários. Alguns autores [14] revelam, inclusive, que dados de treinamento para uma determinada coleção de imagens apresentam resultados consideravelmente satisfatórios em outras bases, mesmo sendo relativamente diferentes. Apesar disso, os resultados podem ser geralmente melhorados retreinando para a base em questão. No entanto, apesar de promissores, os resultados nem sempre atingem o estado-da-arte quando comparados a outras estratégias tradicionais, como pode ser visto em [15].

Além da diversidade de métodos para extração de características, outro fenômeno notável é o crescimento da quantidade e tamanho das coleções de imagens, geralmente em cenários em que dados rotulados são inexistentes. Dessa forma, com grandes coleções de imagens, nem sempre uma quantidade considerável de dados rotulados está disponível. Sendo assim, recentemente, métodos de aprendizado não supervisionado vêm sendo propostos e têm se mostrado bastante promissores, principalmente no que se refere aos ganhos de eficácia obtidos. Esses métodos não necessitam de nenhuma intervenção do

usuário, dados rotulados ou treinamento. A estratégia desses métodos reside em substituir medidas de distância par a par por medidas de afinidade globais, realizando uma análise da informação contextual entre as imagens [16]. Na literatura, são encontrados métodos de aprendizado não supervisionado baseados em diferentes abordagens (processos de difusão [17], grafos [18], análises das listas ranqueadas [19], etc.). No decorrer dos últimos anos, os métodos de aprendizado não supervisionado baseados na análise das listas ranqueadas ganharam um grande destaque. Isso se deve ao fato desses métodos serem eficientes e de baixo custo computacional [20]. A fim de melhorar os resultados de eficácia desses métodos, foram propostas abordagens com o objetivo de realizar a agregação de resultados complementares providos pelos descritores [21].

Embora os métodos de aprendizado não supervisionado operem de forma independente à intervenção do usuário, a seleção dos descritores utilizados consiste em uma tarefa desafiadora. Geralmente, costuma ser difícil selecionar o melhor tipo de descritor (cor, forma, textura, etc.) para um determinado cenário e as características extraídas por ele não definem ou representam a imagem como um todo. Se compararmos, por exemplo, a imagem de um limão com a de uma laranja [22] considerando apenas o contorno, diremos que as imagens são muito semelhantes. No entanto, se fizermos o mesmo utilizando um descritor de cor, teremos que as imagens são consideravelmente diferentes. Considerando esses desafios, pesquisas vêm sendo realizadas visando a combinação de diferentes descritores [23]. Em alguns cenários, pode ser necessário ainda considerar graus de relevância para diferentes informações, de tal forma que possam ser atribuídos pesos diferentes para cada tipo de característica.

Como diferentes métodos apresentam resultados distintos e, muitas vezes complementares, para uma mesma coleção de imagens, uma estratégia de grande importância consiste em realizar a combinação desses métodos. No entanto, a escolha da melhor combinação depende muito da coleção de imagens e do descritor a ser utilizado. Trabalhos recentes apresentam propostas de seleção de combinação de métodos usando técnicas supervisionadas e algoritmos genéticos [24].

Este trabalho propõe três diferentes métodos para seleção e combinação de listas ranqueadas completamente não supervisionados. Cada um deles consiste de diferentes estratégias, cuja aplicação pode ser mais ou menos indicada a depender do cenário de uso.

O USRF (*Unsupervised Selective Rank Fusion*) faz a seleção através da atribuição de valores para cada um dos possíveis pares de ranqueadores (nesse caso descritores) pela utilização de métricas de correlação e estimativas de eficácia. O método aplica a seleção para combinações de qualquer tamanho através da união dos pares mais relevantes, o que é imprescindível para viabilizar a seleção para grandes conjuntos de ranqueadores.

Enquanto isso, o RGSF (*Ranker Graph for Selection and Fusion*) explora um grafo de ranqueadores em que valores de estimativa de eficácia são atribuídos aos vértices e

métricas de correlação às arestas. As componentes conexas são utilizadas para selecionar as combinações a serem agregadas.

Por outro lado, o UGAF-RSF consiste em um *framework* de algoritmo genético em que os cromossomos, representados por *bit-strings*, descrevem diferentes combinações. Estas combinações são selecionadas a partir de estimativas de eficácia não supervisionadas que são utilizadas como funções de aptidão. Tal característica possibilita uma evolução completamente não supervisionada.

As abordagens foram avaliadas em cinco coleções de imagens com tamanhos variando de 1.360 a 10.200 imagens. Mais de trinta descritores de categorias distintas (global, local e *deep learning*) e tipos foram considerados para cada uma das coleções. Diferentes métricas de estimativa de eficácia e de correlação foram analisadas juntamente aos demais parâmetros. Além disso, vários métodos foram utilizados como *baselines*, incluindo métodos do estado-da-arte. Os métodos propostos se revelaram comparáveis ou superiores aos demais métodos nos cenários avaliados. A maior parte das abordagens da literatura operam a nível de características [25, 26, 27, 28, 29], enquanto as abordagens propostas atuam a nível de informações de ranqueamento.

O restante do documento é organizado como se segue:

- o Capítulo 2 define os aspectos teóricos fundamentais dentro do contexto da obra;
- o Capítulo 3 descreve a categorização e os principais métodos de seleção e combinação aplicados à recuperação de imagens;
- o Capítulo 4 relata o protocolo experimental utilizado nos experimentos;
- o Capítulo 5 apresenta o USRF, método baseado em métricas par-a-par, proposto para seleção e combinação de listas ranqueadas;
- o Capítulo 6 apresenta o RGSF, método baseado em grafos e componentes conexas, proposto para seleção e combinação de listas ranqueadas;
- o Capítulo 7 apresenta o UGAF-RSF, método que consiste em um algoritmo genético, proposto para seleção e combinação de listas ranqueadas;
- o Capítulo 8 compara os três métodos propostos com diferentes *baselines*, o estado-da-arte, discute os resultados e as principais contribuições e publicações.
- o Capítulo 9 discute as conclusões obtidas e os possíveis trabalhos futuros.

2 Aspectos Teóricos

Neste capítulo são definidos os principais aspectos teóricos relevantes utilizados como base deste trabalho. Enquanto a Seção 2.1 descreve os principais conceitos de um sistema de recuperação de informações, a Seção 2.2 discute as principais categorias de métodos de aprendizado de máquina.

2.1 Recuperação de Informações

A Recuperação de Informações (RI) é uma área muito abrangente da Ciência da Computação e pode ser definida como:

“A Recuperação de Informação trata da representação, armazenamento, organização e acesso a itens de informação, como documentos, páginas Web, registros estruturados e semi-estruturados, objetos multimídia, etc. A representação e a organização dos itens de informação devem fornecer aos usuários facilidade de acesso às informações de seu interesse.” [30].

Em outras palavras, de forma sucinta, um problema de recuperação de informações pode ser entendido como uma tarefa de busca de um determinado item a partir de um critério fornecido pelo usuário.

Esta seção é subdividida em quatro partes: a apresentação de sistemas de recuperação de imagens pelo conteúdo (Seção 2.1.1), uma discussão sobre os principais descritores visuais e suas respectivas abordagens (Seção 2.1.2), a definição formal do problema de recuperação multimídia (Seção 2.1.3) e, por último, são definidas as métricas de eficácia desses sistemas (Seção 2.1.4).

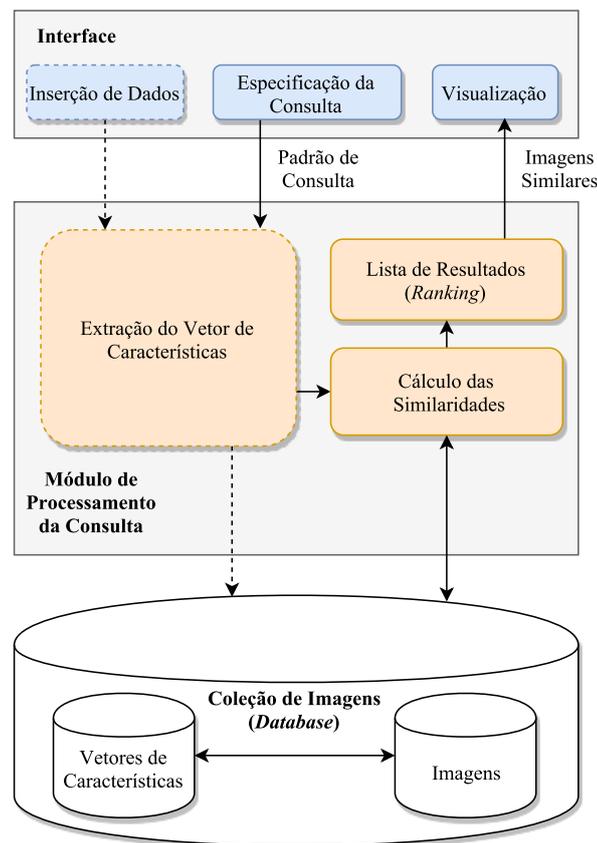
2.1.1 Sistemas de Recuperação de Imagens pelo Conteúdo

Devido ao grande aumento das coleções de imagens, torna-se cada vez mais necessário encontrar formas eficazes e eficientes para realizar a indexação e a busca desse tipo de informação [1]. Os primeiros sistemas de recuperação de imagens realizavam as buscas considerando metadados e palavras-chave [31]. No entanto, essa metodologia é bastante limitada, uma vez que ela ignora todo o conteúdo visual disponível. Sendo assim, surgiram os Sistemas de Recuperação de Imagens pelo Conteúdo (CBIR) que possuem como papel analisar e comparar as imagens considerando o conteúdo das mesmas [32].

Uma estratégia comum aplicada por sistemas CBIR é a utilização de descritores visuais [33] que tem como finalidade representar imagens como vetores de características,

utilizando diferentes abordagens (cor, forma, textura, etc.) [34, 35]. Sendo assim, esses sistemas inicialmente realizam a indexação e a extração de vetores de características para uma coleção de imagens fornecida a fim de construir uma base de dados (*database*) que possibilite comparações com futuras imagens de busca. Quando uma imagem de busca é fornecida, seu vetor de características é extraído e comparado com os existentes na base de dados, de tal forma que os resultados mais relevantes são retornados em uma lista ranqueada em ordem crescente de distância.

A Figura 1 ilustra a implementação clássica de um sistema CBIR [33], que pode ser dividida em três partes: a interface, responsável pela interação com o usuário (inserção e saída dos dados); o módulo de processamento da consulta que tem como tarefa extrair as características das imagens e realizar as operações necessárias para compará-las e, por último, a base de dados que armazena as imagens e seus vetores de características para evitar a necessidade de recalculá-los.



Fonte: Adaptado de [33].

Figura 1 – Arquitetura típica de um sistema CBIR.

Um aspecto importante que deve influenciar na performance de busca é o cálculo das distâncias ou similaridades. A distância entre os vetores de características pode ser calculada de diferentes maneiras, como apresentado pela Tabela 1. De maneira geral, a maior parte dos sistemas CBIR costumam utilizar a distância Euclidiana [36]. Apesar do

cálculo das distâncias ser fundamental para uma comparação eficaz, os descritores e as extrações de características são os que mais têm impacto na eficácia do sistema como um todo.

Tabela 1 – Exemplos de métricas de distância.

Distância	Equação
Euclidiana	$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Chebyshev	$D(x, y) = \max_{i=1}^n x_i - y_i $
Manhattan	$D(x, y) = \sum_{i=1}^n x_i - y_i $
Minkowsky	$D(x, y) = [\sum_{i=1}^n x_i - y_i ^p]^{1/p}$

Diferentemente de outros sistemas de busca, como os textuais, que possuem resultados mais maduros e robustos para cenários de propósito geral; a busca de imagens pelo conteúdo ainda possui diversos desafios de pesquisa [31]. O principal deles reside na dificuldade dos descritores de relacionar características de alto nível às de baixo nível. Tal problema é conhecido como lacuna semântica (“*semantic gap*”) [10] e é abordado em detalhes na subseção a seguir.

2.1.2 Descritores de Imagens

Para fins de representação, considerar os *pixels* de uma imagem sem nenhuma forma de pré-processamento geralmente produz resultados pouco eficazes. Dessa forma, os descritores visuais são propostos com o objeto de, a partir de uma imagem, extrair características consideradas relevantes e armazená-las em um vetor numérico, conhecido como vetor de características [1].

O desafio de descrever imagens é muito discutido na literatura. Tal fenômeno é evidente quando observamos a questão da lacuna semântica [10], que é a distância entre o conceito de alto nível (a semântica ou significado associado à imagem) e o conceito de baixo nível (o vetor de características). A Figura 2 ilustra o processo de extração de características e evidencia a lacuna semântica, pois o vetor de características mostra a perda do conceito de alto nível. Uma vez que, a partir dos valores numéricos não se pode mais assegurar que a imagem contém um peixe. Além disso, as diferenças entre elementos do mesmo conceito dificultam o problema. Visualiza-se que, apesar de representarem o mesmo conceito, os peixes possuem diferentes aspectos de cor, contorno e textura. Dessa forma, várias técnicas são apresentadas com o objetivo de tentar resolver ou minimizar o impacto da lacuna semântica, as quais são discutidas em mais detalhes na Seção 2.2.

Diversos tipos de descritores visuais foram criados com a finalidade de representar imagens, utilizando das mais diversas abordagens. Eles são geralmente divididos em duas categorias principais [1]: globais e locais. Essa categorização é feita a partir de como esses algoritmos analisam as imagens e buscam semelhanças entre elas. Enquanto os descritores

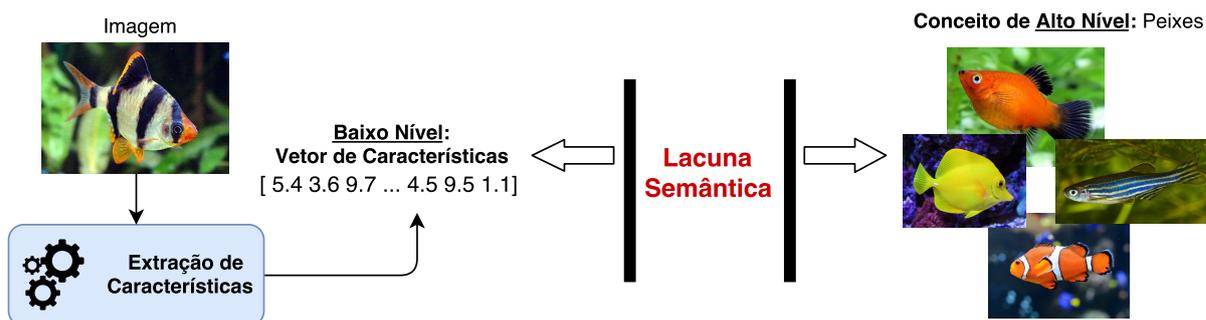


Figura 2 – Ilustração do processo de extração de características e da lacuna semântica.

globais descrevem a imagem como um todo, os descritores locais descrevem apenas algumas partes consideradas mais relevantes por eles. Recentemente, devido ao crescimento do poder computacional disponível, surgiu uma nova categoria de descritores que empregam de redes neurais convolucionais [31], os quais adquiriram uma notória popularidade pelos resultados muito significativos. As estratégias são descritas a seguir.

- **Globais**

Os descritores globais foram os primeiros a serem propostos e consistem em uma abordagem holística para analisar as imagens. Dada uma imagem, aplicam um algoritmo para descrever a imagem como um todo, isso pode ser feito explorando diferentes modalidades como cor, contorno e textura.

Os descritores de cores calculam um histograma de cores para as imagens de entrada, visando identificar as cores mais e menos frequentes, geralmente considerando o espaço de cores RGB (vermelho, verde e azul). Como exemplos de descritores dessa categoria têm-se *Color Autocorrelogram* (ACC) [35], *Border/Interior Classification* (BIC) [37] e *Global Color Histogram* (GCH) [38].

No caso dos descritores de contorno, é necessário aplicar um pré-processamento na imagem a fim de extrair os contornos (algoritmo *Canny edge detection*, por exemplo). Sendo assim, a imagem pode ser dividida em grupos de *pixels* para a contagem das linhas de contorno e construção de um histograma. Existem descritores de contorno, com diferentes variações, dentre os quais se encontram *Segment Saliences* (SS) [39], *Contour Features Descriptor* (CFD) [40] e *Aspect Shape Context* (ASC) [41].

Os descritores de textura geralmente são treinados considerando diferentes tipos de superfície (gramados, tijolos, entre outras) e posteriormente são aplicados às imagens a fim de classificar novas superfícies. Como exemplos desse tipo de descritor estão *Local Binary Patterns* (LBP) [42] *Color Co-Occurrence Matrix* [43] e *Local Activity Spectrum* (LAS) [44].

Um descritor global que tem uma abordagem que diverge dos demais é o *Histogram of Oriented Gradients* (HoG) [45], que divide a imagem em subconjuntos de *pixels* e a

descreve em função de vetores orientados, a partir dos vetores computa-se um histograma. Esse descritor é muito eficaz para reconhecimento de faces.

Um dos principais problemas dos descritores globais é que, muitas vezes, possuem dificuldades ao tentar identificar a mesma imagem quando o objeto está rotacionado ou transladado. Tal questão é conhecida como o problema da variância [31].

- **Locais**

A fim de solucionar o problema da variância e de aumentar ainda mais a eficácia das buscas de imagens pelo conteúdo, os descritores locais foram propostos. Esses descritores geralmente aplicam abordagens de BoVW (*bag of visual words*), se inspirando no BoW (*bag of words*) utilizado na recuperação textual. A principal ideia dessa abordagem consiste em analisar diferentes imagens procurando por “pontos chave” (*keypoints*) que sejam de alta relevância para descrever as imagens e também sejam invariantes, esses pontos nada mais são do que partes das imagens que podem possuir tamanhos distintos [46]. Essas partes são utilizadas a fim de construir um dicionário de palavras visuais (*bag of visual words*). Um vetor de características pode ser construído, por exemplo, considerando a ocorrência de cada um dos elementos do dicionário. Como principais exemplos de descritores dessa modalidade, temos: *Scale-Invariant Feature Transform* (SIFT) [46], *Speeded Up Robust Features* (SURF) [47], *Binary Robust Independent Elementary Features* (BRIEF) [48] e ORB (*Oriented Fast and Rotated BRIEF*) [49].

Um aspecto importante é que abordagens globais e locais podem ser combinadas. É possível, por exemplo, executar uma primeira etapa para descrever as imagens considerando uma abordagem global (cor, textura, contorno, etc.) e, em seguida, executar um descritor local [47, 50]. Uma vez que, uma execução prévia de descritores globais pode, de certa forma, remover conteúdos pouco relevantes para a descrição final da imagem.

- **Redes Neurais**

Com a grande evolução da capacidade de processamento, houve uma grande popularização dos algoritmos de *deep learning* que necessitam de grande custo computacional [31]. Uma rede de *deep learning* é uma rede neural de múltiplas camadas, possuindo camadas ocultas (posicionadas entre a primeira e a última camada). Sua grande popularização se deve aos resultados bastante satisfatórios no ramo de visão computacional e reconhecimento de padrões.

Geralmente, redes de *deep learning* são treinadas de maneira supervisionada e seus vetores de pesos são utilizados como vetores de características para uma determinada imagem. Um tipo muito comum de redes neurais são as CNN (*Convolutional Neural Networks*). Diferentes descritores podem ser combinados a fim de aprimorar a eficácia das redes neurais, uma possibilidade é a combinação com descritores locais como SIFT+CNN [51], por exemplo.

2.1.3 Definição Formal

A seguir é apresentada uma definição formal e os principais conceitos para o problema de recuperação multimídia tratado neste trabalho.

Seja $\mathcal{C} = \{obj_1, obj_2, \dots, obj_n\}$ uma coleção de objetos em que N é o tamanho da coleção \mathcal{C} . O ranqueador R pode ser definido com base em diferentes abordagens, incluindo descritores e métodos de pós-processamento supervisionados e não supervisionados. Na maior parte deste trabalho, o ranqueador R é usado como sinônimo de descritor definido em [33] como uma tupla (ϵ, ρ) em que $\epsilon : \mathcal{C} \rightarrow \mathbb{R}^N$ é uma função que extrai um vetor de características v_x de um objeto $obj_x \in \mathcal{C}$; e $\rho : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ é uma função que calcula a distância entre dois objetos de acordo com seus vetores de característica correspondentes. Formalmente, a distância entre dois objetos obj_i, obj_j é definida por $\rho(\epsilon(obj_i), \epsilon(obj_j))$. A notação $\rho(i, j)$ é utilizada para denotar a distância entre dois objetos.

A distância $\rho(i, j)$ entre todos os objetos $obj_i, obj_j \in \mathcal{C}$ pode ser calculada para se obter a matriz quadrada de distâncias A , tal que $A_{ij} = \rho(i, j)$. A matriz de distâncias A é utilizada como entrada de vários métodos de aprendizado não supervisionado.

Uma representação alternativa dos resultados de recuperação são as listas ordenadas de resultados ou listas ranqueadas (*ranked lists*). Baseada na função de distância ρ , uma lista ranqueada τ_q pode ser calculada para um dado objeto de consulta obj_q . As listas ranqueadas podem conter informações de uma coleção inteira e, especialmente em suas primeiras posições espera-se conter os objetos mais relevantes relacionados ao objeto de consulta. Portanto, uma estratégia adequada, realizada por alguns métodos a fim de acelerar o processo de busca, consiste em considerar um subconjunto dos L objetos mais similares, em que $L \ll N$ é o número de objetos nas primeiras posições das listas ranqueadas. Essa é uma estratégia útil especialmente para grandes coleções de objetos, em que N é muito grande, e portanto τ_q é muito custoso de se calcular.

A lista ranqueada $\tau_q = (obj_1, obj_2, \dots, obj_L)$ pode ser definida como a permutação de uma coleção de objetos $\mathcal{C}_L \subset \mathcal{C}$, que contêm os objetos mais similares a um determinado objeto de busca obj_q , tal que $|\mathcal{C}_L| = L$. A permutação τ_q é uma bijeção do conjunto \mathcal{C}_L sobre o conjunto $[L] = \{1, 2, \dots, L\}$. Para uma permutação τ_q , interpreta-se $\tau_q(i)$ como a posição (classificação ou *rank*) do objeto obj_i na lista ranqueada τ_q . Pode-se dizer que, se o objeto obj_i é classificado antes do objeto obj_j na lista ranqueada do objeto obj_q , ou seja, $\tau_q(i) < \tau_q(j)$, então $\rho(q, i) \leq \rho(q, j)$. Tomando cada objeto $obj_i \in \mathcal{C}$ como um objeto de busca obj_q , obtêm-se o conjunto $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ de listas ranqueadas para cada objeto da coleção \mathcal{C} .

Diferentes descritores e funções de distância originam diferentes ranqueadores R que, por sua vez, produzem diferentes conjuntos de listas ranqueadas \mathcal{T} . Seja $\mathfrak{R} = \{R_1, R_2, \dots, R_m\}$ um conjunto de ranqueadores e $R_i \in \mathfrak{R}$, denota-se por \mathcal{T}_i o conjunto de

listas ranqueadas produzidas por R_i . Uma lista ranqueada calculada pelo ranqueador R_j em resposta a um objeto de consulta obj_q é denotada por $\tau_{j,q}$.

Seja \mathfrak{X}_n uma combinação formada por n ranqueadores em que $n \geq 2$. O objetivo dos métodos propostos é de selecionar a combinação que produza o melhor resultado de eficácia a partir dos ranqueadores presentes em \mathfrak{R} . Formalmente, uma proposta de seleção pode ser definida como uma função f_s como se segue:

$$\mathfrak{X}^* = f_s(\mathfrak{R}), \quad (2.1)$$

em que \mathfrak{X}^* denota a combinação de ranqueadores selecionada.

2.1.4 Métricas de Eficácia

Métricas de eficácia são fundamentais para avaliar a qualidade dos resultados de busca [30]. Nesta subseção são apresentadas as métricas abordadas neste trabalho. Todas são calculadas a partir das listas ranqueadas. Os resultados são obtidos no intervalo $[0, 1]$ e quanto maior o valor, melhor o resultado.

- **Precisão**

A precisão pode ser entendida como a fração das instâncias relevantes entre as instâncias recuperadas, a qual é calculada como:

$$P_n = \frac{c}{n}, \quad (2.2)$$

em que n é o número de itens recuperados e c é o número de itens corretos dentre os recuperados. Observe que se n for igual ao tamanho da coleção, o resultado é sempre 1.

- **Revocação**

Diferente da precisão, a revocação é a fração de instâncias relevantes recuperadas sobre o número total de instâncias relevantes. Ressalta-se que a notação R utilizada a seguir não se refere a um ranqueador, mas sim a métrica de revocação a qual é definida como:

$$R_n = \frac{c}{n_r}, \quad (2.3)$$

em que n é o número de itens recuperados, c é o número de itens corretos dentre os itens recuperados e n_r é o número de itens relevantes. O número de itens relevantes é o número máximo de itens que se pode acertar em uma determinada circunstância.

- **MAP**

O MAP (*Mean Average Precision*) é a métrica mais comum para medir a eficácia de listas ranqueadas em tarefas de recuperação. Para cada uma das listas ranqueadas do conjunto \mathcal{T} , pode-se obter a precisão média AP (*Average Precision*). A precisão e a revocação são calculadas em cada uma das posições da lista ranqueada, dando origem a uma curva que descreve a função $P(R)$, na qual a precisão é fornecida em função da revocação. A precisão média considera o valor médio de $P(R)$ no intervalo $[0, 1]$ e é definida como:

$$AP = \int_0^1 P(R)dr, \quad (2.4)$$

que é a área abaixo da curva.

Ao se realizar a média ponderada da precisão média de cada uma das listas ranqueadas do conjunto \mathcal{T} , obtêm-se o MAP.

2.2 Aprendizado de Máquina

De maneira geral, algoritmos de aprendizado de máquina são aqueles que tem como tarefa analisar um determinado volume de dados e automaticamente extrair padrões e informações sobre eles, de tal forma que seja possível tomar decisões baseadas nos dados ao invés de serem explicitamente programados para uma determinada tarefa em específico [52]. Devido ao grande volume dados cada vez maior, esses métodos tem ganhado bastante atenção pela possibilidade de automatizar diversas tarefas.

Atualmente, programas de aprendizado de máquina são amplamente utilizados em diversas áreas, mas principalmente em reconhecimento de padrões e visão computacional. Diferentes métodos de aprendizado vêm sendo propostos no cenário de recuperação de imagens [18, 53] com a finalidade de solucionar o problema da lacuna semântica.

Nesta seção são discutidas as principais categorias de métodos de aprendizado de máquina que estão relacionadas à recuperação de imagens pelo conteúdo, que é o tema deste trabalho. Os métodos de aprendizado podem ser divididos em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado semi-supervisionado. Um conceito importante que será utilizado no decorrer do texto é o de dados rotulados e não rotulados. Para os itens de dados rotulados, além dos atributos são também fornecidas informações de categoria ou classe para o método. No caso de itens de dados não rotulados, nada é fornecido além dos atributos.

2.2.1 Aprendizado Supervisionado

Os métodos de aprendizado supervisionado são aqueles que aprendem a partir de dados rotulados e tomam decisões considerando dados não rotulados [52]. As principais vantagens desses métodos é que eles podem ser treinados para tarefas mais específicas, fornecendo um conjunto de classes bem definido, e tempo de resposta rápido para inferência de um novo dado.

As principais abordagens baseadas em aprendizado supervisionado no contexto de recuperação de informações utilizam do conceito de *relevance feedback*, o que é um tópico ativo de pesquisa na área [54, 11]. Um típico cenário de *relevance feedback* consiste em buscas iterativas, em que a cada iteração o usuário realiza uma determinada busca, fornecendo uma imagem de entrada. A partir da busca realizada, um lista ranqueada é obtida e cabe ao usuário julgar a qualidade da busca, atribuindo um nível de interesse em cada uma das imagens retornadas. Após isso a máquina é novamente treinada e esse processo se repete iterativamente com o objetivo de aprimorar a eficácia do sistema.

Outra abordagem muito comum é a utilização de classificadores supervisionados, como as de Máquinas de Vetores de Suporte (SVM) [55]. Basicamente, o problema é modelado como um cenário de classificação binária, no qual o objetivo é encontrar o hiperplano que melhor separa os dados relevantes dos não relevantes. Há também outras abordagens baseadas em programação genética [56, 57] e grafos [58]. Uma lista ranqueada pode ser gerada considerando a distância dessas imagens em relação ao hiperplano. No cenário atual, métodos de *deep learning* também têm atraído muita atenção, sendo utilizados como classificadores [59].

Apesar da popularidade, métodos de aprendizado supervisionado nem sempre são adequados, por serem muito sensíveis aos dados utilizados para treinamento. Se o volume de dados rotulados for muito limitado, pode ser que não seja possível fazer um treinamento que forneça um nível adequado de generalização para a tarefa desejada. Existe também o problema de sobreajuste (*overfitting*), quando o modelo se ajusta demasiadamente ao conjunto de treinamento, sendo menos eficaz para classificar novos dados [52]. Embora o tempo de inferência para novos dados seja em geral muito eficiente; quando a quantidade de dados é muito grande, o tempo de treinamento pode ser muito elevado.

2.2.2 Aprendizado Não Supervisionado

Por serem não supervisionados, esses métodos não necessitam de nenhuma intervenção do usuário, dados rotulados ou treinamento [52]. No contexto de recuperação de imagens, a ideia desses métodos consiste em substituir medidas de distância por a par por medidas de afinidade global, ou seja, realizar uma análise da informação contextual entre as imagens [16]. Na literatura, são encontrados métodos de aprendizado não supervisi-

onados baseados em diferentes abordagens: processos de difusão [17, 60], grafos [18, 16], clusterização [61], frequência de padrões [62] e análises das listas ranqueadas [19].

Ao longo dos anos, os métodos de aprendizado não supervisionado baseados na análise das listas ranqueadas ganharam um grande destaque [63, 64]. Isso se deve ao fato desses métodos terem baixo custo computacional, não necessitando das listas ranqueadas por completo. Os métodos são baseados em análises das listas ranqueadas, mas utilizam diferentes abordagens para tal [20, 65, 66, 67] (produto cartesiano, grafos, recomendações, entre outras).

2.2.3 Aprendizado Semi-Supervisionado

Esses métodos são menos comuns, mas tem ganhado cada vez mais popularidade. Os métodos de aprendizado semi-supervisionado aprendem utilizando tanto dados rotulados como não rotulados (*transductive learning*). Um dos principais objetivos desse tipo de aprendizado é solucionar cenários em que existem poucos dados rotulados, nos quais geralmente os métodos supervisionados não são eficazes [52]. Como desvantagem, é importante que os poucos dados rotulados fornecidos forneçam uma boa referência para que o algoritmo se direcione para um aprendizado com um bom nível de generalização.

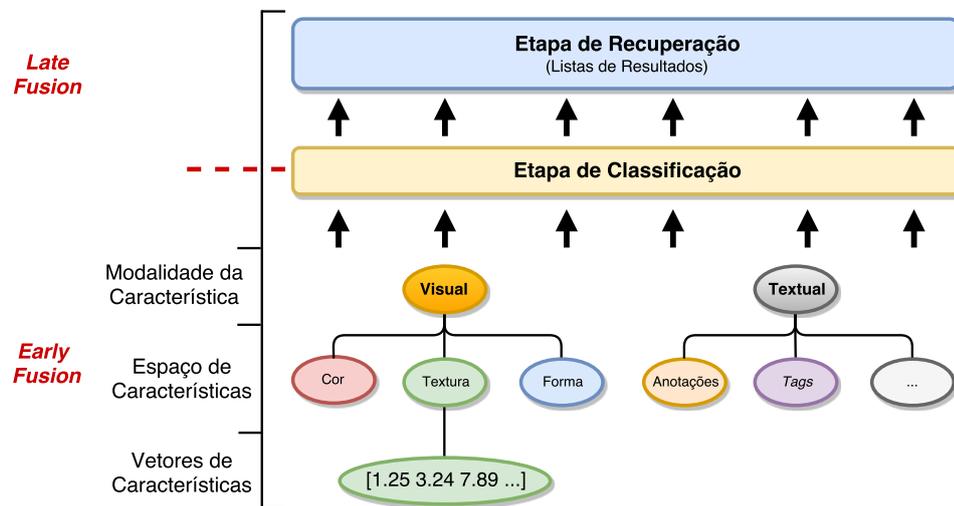
Podem ser citados diferentes classificadores semi-supervisionados, dentre eles o PCC (*Particle competition and cooperation*) [68], variações do *Optimum Path Forest* (OPF) [69] e adaptações das Máquinas de Vetores de Suporte (SVM) que realizam uma etapa de pós-processamento não supervisionada [70]. Os classificadores podem realizar uma tarefa de classificação binária a partir de uma imagem qualquer e uma imagem de busca, informando se aquela imagem é relevante quando comparada à imagem de busca.

3 Trabalhos Relacionados

Neste capítulo é discutida uma revisão bibliográfica abrangente considerando o tema de seleção e combinação. São apresentadas maneiras para se selecionar as características a serem combinadas e também formas e modelos para combinar esses dados.

3.1 Categorização das Abordagens de Combinação

Ao longo dos últimos anos, diversas técnicas para realizar a combinação de diferentes descritores de imagens foram apresentadas na literatura [71], sendo geralmente adaptadas para os descritores selecionados, a estratégia de combinação, e o cenário de aplicação. Técnicas de combinação, também conhecidas como técnicas de fusão, são frequentemente categorizadas em duas classes, chamadas de *early fusion* e *late fusion* [72, 73], as quais têm seus nomes relativos a qual parte do processo é realizada a combinação. A Figura 3 ilustra as principais etapas em um cenário de recuperação de imagens e em qual parte cada uma das combinações se enquadra. As seções seguintes discutem em detalhes cada uma das abordagens. É importante observar que, um sistema CBIR pode realizar tanto *early fusion* quanto *late fusion* em etapas distintas.



Fonte: Adaptado de [31]

Figura 3 – Esquema de um sistema CBIR considerando a combinação de características.

3.1.1 *Early Fusion*

O termo *early fusion* se refere a combinação de diferentes características em uma única representação, antes do cálculo das distâncias entre os elementos da coleção [72].

É uma técnica muito comum em aplicações de recuperações de imagens, a forma mais conhecida de *early fusion* é a concatenação de vetores de características em único vetor [74]. Outras operações podem ser realizadas também, como a soma ou a multiplicação dos vetores. No entanto, há diversas outras possibilidades a serem exploradas. Na literatura, encontram-se propostas em que são combinados vetores de características de descritores baseados em diferentes características [75], como cor, forma e textura, por exemplo. Na Figura 4 é apresentada uma sequência de etapas de recuperação de imagens em que ocorre um exemplo de *early fusion*.

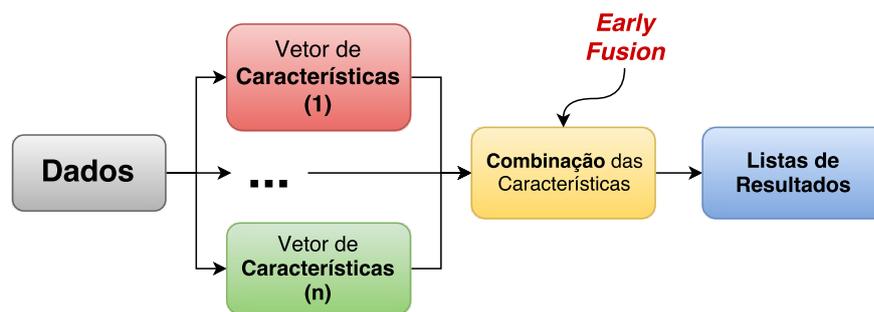


Figura 4 – Exemplo de etapas com presença de *early fusion*.

3.1.2 Late Fusion

No caso do *late fusion*, a combinação das características só ocorre após o cálculo das distâncias. Dessa forma, a fusão pode ser realizada tanto com matrizes de distância ou similaridade, listas ranqueadas (*ranked lists*), ou qualquer outra representação [76]. O que caracteriza o *late fusion* é o fato da combinação não ser diretamente realizada sob os vetores puros. Métodos de reranqueamento [20] são bastante conhecidos por implementar diferentes formas de *late fusion*. Outra possibilidade é atribuir pesos distintos a cada um dos dados a serem combinados [77, 78], aplicando maior relevância a certas informações do que outras, visando uma maior eficácia. Na Figura 5 é apresentada uma sequência de etapas de recuperação de imagens em que ocorre um exemplo de *late fusion*.

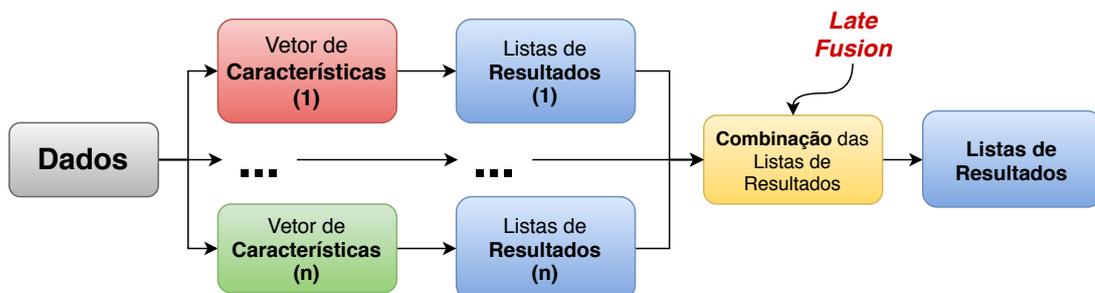


Figura 5 – Exemplo de etapas com presença de *late fusion*.

3.2 Abordagens de Combinação

Devido ao fato da seleção e combinação de características ser extremamente útil e abrangente, existem diferentes abordagens aplicadas em diversas áreas (bioinformática [79], biometria [5], recuperação de informações [31], entre outras). A partir de um levantamento bibliográfico, esta seção apresenta abordagens encontradas na literatura que contribuem direta ou indiretamente com a recuperação de imagens pelo conteúdo. Em virtude da grande quantidade de métodos disponíveis, para uma maior facilidade do leitor o texto é dividido: a Seção 3.2.1 mostra as estratégias baseadas em operações aritméticas; a Seção 3.2.2 apresenta as abordagens conservativas; a Seção 3.2.3 descreve os métodos baseados em combinação de listas ranqueadas que são de fundamental importância em recuperação de informações; a Seção 3.2.4 aborda estratégias de combinação e aprendizado de núcleos voltadas à classificadores, as quais têm recentemente ganhado atenção na área de visão computacional e reconhecimento de padrões; a Seção 3.2.5 apresenta os algoritmos que são baseados em metodologias estatísticas e; por último, a Seção 3.2.6 aborda a estratégia de programação genética.

3.2.1 Combinação Aritmética Não Treinável

Uma combinação aritmética não treinável visa combinar diferentes características a partir de operações algébricas aplicadas em todos os vetores e que, não requer nenhum treinamento [80]. Tais combinações são classificadas como prematuras (*early fusion*).

A partir de um conjunto de vetores de características $\mathcal{F} = \{F_1, \dots, F_t\}$ e sendo t o tamanho do conjunto, é desejado obter um vetor de combinação resultante F' . A Tabela 2 apresenta alguns exemplos de combinação simples que podem ser realizadas, as operações são aplicadas a todas as posições (valores) dos vetores de características.

Tabela 2 – Exemplos de operações aritméticas simples para obtenção de F' .

Tipo	Equação
Somatório	$F' = \sum_{i=1}^t F_i$
Produtório	$F' = \prod_{i=1}^t F_i$
Media Aritmética	$F' = \frac{1}{t} \sum_{i=1}^t F_i$
Regra do Máximo	$F' = \max_{i=1}^t F_i$

Observe que a dimensão do vetor F' é igual a dimensão do maior vetor pertencente a \mathcal{F} e que como os vetores a serem combinados geralmente possuem valores em escalas diferentes, uma normalização prévia é quase sempre necessária e recomendada.

3.2.2 Combinação Conservativa

Esse trabalho considera como conservativa qualquer combinação que busque, de certa forma, preservar as características que foram combinadas. Tais métodos são classifi-

cados como de combinação prematura (*early fusion*).

Novamente, dado um conjunto de vetores de características $\mathcal{F} = \{F_1, \dots, F_t\}$ e sendo t o tamanho do conjunto, é desejado obter um vetor de combinação resultante F' .

- **Combinação Sequencial**

A combinação sequencial consiste em concatenar os vetores de características em um vetor resultante [81]. Sendo \parallel o operador de concatenação, a combinação é definida pela Equação 3.1.

$$F' = \text{combSeq}(\mathcal{F}) = F_1 \parallel F_2 \parallel \dots \parallel F_t \quad (3.1)$$

Essa estratégia tem como principal vantagem a preservação das características originais, diferente da soma ou do produto dos valores que, muitas vezes, acabam atenuando determinados valores e requerem de estratégias adequadas de normalização. Com a concatenação, o algoritmo que recebe F' como entrada pode explorar as características das diferentes fontes individualmente se necessário.

Apesar de sua simplicidade, essa abordagem ainda é usada na literatura em casos que o tempo de treinamento não é relevante e os vetores não são muito grandes. Uma vez que, a dimensão do vetor F' é dada pela soma da dimensão de todos os vetores que foram combinados, ou seja, $\dim(F') = \sum_{i=1}^t \dim(F_i)$, isso faz que F' tenha uma alta dimensionalidade e produza resultados menos eficientes computacionalmente.

- **Combinação Paralela**

Tendo como objetivo solucionar o principal problema da combinação sequencial (a questão da dimensionalidade), a combinação paralela [81] foi proposta. Nessa abordagem, vetores de características podem ser combinados utilizando de números complexos. Considerando $i = \sqrt{-1}$ como a unidade imaginária, $F'(x)$ a posição de índice x no vetor de características F' , a partir dos vetores de características F_1 e F_2 pode-se calcular uma nova posição do vetor resultante:

$$F'(x)_{0 \leq x \leq (\dim(F_1) = \dim(F_2))} = \text{combPar}(F_1(x), F_2(x)) = F_1(x) + F_2(x) \times i. \quad (3.2)$$

O cálculo acima é feito para todas as posições, caso F_1 e F_2 tenham dimensões diferentes, o vetor menor é completado com zeros antes do cálculo, para que assim haja correspondência entre todas as posições. Enquanto os valores do primeiro vetor assumem a parte real, os valores do segundo vetor assumem o valor imaginário.

A Equação 3.2 possui o problema de considerar apenas pares de vetores e não o conjunto (\mathcal{F}) como um todo. Para que todo o conjunto seja considerado, é necessário criar

um conjunto de unidades virtuais $\mathcal{I} = \{i_1, \dots, i_t\}$, uma para cada $F \in \mathcal{F}$, assim a equação anterior é reformulada para calcular as posições de F' para todo o conjunto:

$$F'(x)_{0 \leq x \leq \dim(F)} = \text{combPar}(\mathcal{F}(x)) = F_1(x) \times i_1 + \dots + F_t(x) \times i_t. \quad (3.3)$$

Essa abordagem possui algumas variações, mas sua principal ideia é construir uma dimensão para cada uma das características e deixá-las implícitas em um único espaço que permita a recuperação dos valores originais. No geral, os resultados são fiéis aos da combinação serial, mas esse método quase não é mais usado por não ser prático (todas as etapas e algoritmos precisam ser adaptados para os espaços virtuais criados) e ser mais útil apenas para uma pequena quantidade de vetores.

3.2.3 Agregação de Listas Ranqueadas

Esta subseção apresenta as principais técnicas que tem como intuito combinar diversas listas ranqueadas em uma lista consensual e que são de fundamental importância na área de recuperação pelo conteúdo. Todas as abordagens nesta subseção são classificadas como combinação tardia (*late fusion*).

Seja $\mathcal{T} = \{\tau_1, \dots, \tau_t\}$ o conjunto de listas ranqueadas (*ranked lists*) a serem combinadas, em que t é o tamanho desse conjunto. A partir das listas fornecidas, busca-se obter uma lista ordenada ranqueada ótima τ' . O tamanho máximo de cada uma das listas τ é n , que é o tamanho da coleção (*dataset*). Nesta definição estamos considerando as listas ranqueadas para um único elemento de busca (*query*), mas o modelo pode ser generalizado para aplicar a mesma estratégia para múltiplas listas.

- **Contagem de Borda**

O método de Contagem de Borda [82] (*Borda Count*) calcula uma pontuação para cada um dos elementos presentes em uma coleção, a qual é definida pela Equação 3.4.

$$S_B(i) = \sum_{j=0}^t \tau_j(i) \quad (3.4)$$

Uma nova lista resultante τ' é obtida através da ordenação crescente de cada um dos elementos considerando a pontuação fornecida na equação.

- **Combinação Recíproca de Listas Ranqueadas**

A Combinação Recíproca de Listas Ranqueadas (*Reciprocal Rank Fusion*) [83] também calcula uma pontuação para cada elemento, como apresentado na Equação 3.5.

$$S_R(i) = \sum_{j=0}^t \frac{1}{k + \tau_j(i)} \quad (3.5)$$

O método considera apenas as k primeiras posições da lista ranqueada, caso algum elemento esteja acima das k posições (seja um *outlier*), a função retorna $k + 1$. Novamente, uma nova lista resultante τ' é obtida através da ordenação decrescente de cada um dos elementos considerando a pontuação fornecida na equação.

• **Métodos Comb***

A família Comb* de métodos [23] aplica uma função para atribuir uma pontuação a todo item i da coleção (*dataset*) a partir de sua posição nas listas ranqueadas τ_j , como apresentado pela Equação 3.6.

$$S_C(i, \tau_j) = 1 - \frac{\tau_j(i) - 1}{|\tau_j|} \tag{3.6}$$

Após o cálculo das medidas, elas são combinadas usando uma variedade de métodos, apresentados na Tabela 3. Uma nova lista resultante τ' é obtida através da ordenação crescente de cada um dos elementos considerando a pontuação calculada em cada caso.

Tabela 3 – Algoritmos de combinação da família Comb*.

CombMIN	$\min_{\tau_j \in \mathcal{T}} S_C(i, \tau_j)$
CombMAX	$\max_{\tau_j \in \mathcal{T}} S_C(i, \tau_j)$
CombSUM	$\sum_{\tau_j \in \mathcal{T}} S_C(i, \tau_j)$
CombMED	$\frac{1}{ \mathcal{T} } \times CombSUM(i, \mathcal{T})$
CombANZ	$\frac{1}{ \{\tau_i \tau_i \in \mathcal{T} : i \in \tau_i\} } \times CombSUM(i, \mathcal{T})$
CombMNZ	$ \{\tau_i \tau_i \in \mathcal{T} : i \in \tau_i\} \times CombSUM(i, \mathcal{T})$

Observe que tanto CombANZ quanto CombMNZ fazem uma análise dos elementos que não aparecem nas k primeiras posições, ao calcularem o valor de $|\{\tau_i | \tau_i \in \mathcal{T} : i \in \tau_i\}|$.

• **Concordância e Discordância de Pares (*Outrank*)**

O método de *Outrank* [84] compara os itens nas listas ranqueadas. Para cada par ordenado de itens x e y (sendo $x, y \in \tau_i \mid \forall \tau_i \in \mathcal{T}$), é calculada uma condição de concordância ou discordância considerando a ordem $x <_{\tau} y$ (x aparece antes de y em uma lista ranqueada τ). A condição de concordância é utilizada para verificar se a maioria das listas ranqueadas concordam que $x < y$. A condição de discordância verifica se alguma lista fortemente discorda que $x < y$, ou seja, x aparece muito a frente de y , o limiar para isso deve ser definido. Após o cálculo das condições, os itens são agregados em uma lista ranqueada consensual, em que os itens com grande grau de concordância e baixo grau de discordância aparecem nas primeiras posições.

• **Agregação de Listas pela Mediana (MRA)**

O algoritmo conhecido como *Median Rank Aggregation* (MRA) [85] ordena todos os itens pela mediana das posições que eles receberam nas listas ranqueadas $\tau \in \mathcal{T}$.

- **Aprendizado de Listas Ranqueadas (*Learning-to-Rank*)**

Várias abordagens supervisionadas foram propostas para o aprendizado de listas ranqueadas [86], conhecidas como estratégias de *Learning-to-Rank*. Tais métodos podem ser classificados em três categorias distintas:

- **Análise atômica (*Pointwise*):** um classificador é treinado para, dado um item de entrada, atribuir a ele uma pontuação de significância ou relevância em relação a um objeto de busca (*query*). A classificação é realizada para todos os elementos e eles são posteriormente ranqueados considerando a pontuação atribuída. Atualmente, essa estratégia é a menos utilizada.
- **Análise de pares (*Pairwise*):** um classificador é treinado para, dado um par de elementos, retornar qual deles é mais relevante em relação a um objeto de busca (*query*). A análise dos pares é usada para construir uma lista ranqueada consensual. Exemplos: RankingSVM [87] e RankNET [88].
- **Análise em listas (*Listwise*):** técnicas baseadas em listas operam diretamente sobre listas de elementos e visam gerar uma lista ordenada ótima, isso pode ser feito visando otimizar uma métrica de recuperação como o MAP ou otimizando uma função de perda que foi modelada para o tipo o problema em questão. Exemplos: ListNet [89] e AdaRank [90].

3.2.4 Combinação de Núcleos

As abordagens baseadas em núcleos (*kernels*) foram inicialmente propostas com a finalidade de obter núcleos híbridos para classificadores conhecidos como Máquinas de Vetores de Suporte (SVM) [91]. O SVM define suas relações no espaço utilizando do produto interno. Um núcleo substitui a relação de produto interno padrão e permite trabalhar a separabilidade de classes em espaços de maior dimensionalidade sem a necessidade de qualquer conversão dos dados, trazendo uma grande vantagem em termos de eficiência. Diferentes núcleos podem ser usados como, por exemplo, linear, polinomial, sigmoidal e gaussiano (conhecido como RBF - *Radial Basis Function*).

No entanto, essas abordagens se revelaram muito mais úteis quando aplicadas a combinação de diferentes características. A ideia consiste em associar uma característica distinta para cada um dos núcleos. Dessa forma, abordagens baseadas na combinação de núcleos também são vistas como uma combinação de características, o que tem atraído bastante a atenção da comunidade científica.

Nesta subseção é apresentado o MKL original e suas principais variações. Formalmente, considera-se um conjunto de núcleos $\mathcal{K} = \{K_1, \dots, K_t\}$ em que t é o número de

núcleos a serem combinados. A partir deles, deseja-se obter um núcleo combinado ou híbrido K' .

- **Soma dos Núcleos**

Um método bastante simples consiste em realizar a soma dos núcleos [92], como apresentado na Equação 3.7. Uma estratégia muito parecida que oferece uma eficácia similar é realizar a média dos núcleos.

$$K' = \sum_{i=1}^t K_i \quad (3.7)$$

- **Produto dos Núcleos**

Outra técnica bastante simples consiste no produto dos núcleos [92], como apresentado na Equação 3.8. Caso um dos núcleos tenha fornecido uma performance muito menor em um determinado caso, isso pode impactar bastante na combinação final. Essa estratégia possui resultados muito similares a selecionar o valor máximo dos núcleos de entrada.

$$K' = \prod_{i=1}^t K_i \quad (3.8)$$

- **Aprendizado de Múltiplos Núcleos (MKL)**

Combinar os núcleos sem aplicar nenhum treinamento prévio pode não fornecer resultados satisfatórios dependendo do caso. Dessa forma, foi proposto o *Multiple Kernel Learning* (MKL) uma abordagem que originalmente procura encontrar a melhor combinação linear de diferentes núcleos fornecidos [93]. Sendo $\mathcal{B} = \{\beta_1, \dots, \beta_t\}$ o conjunto de pesos a serem aplicados nessa combinação linear e $\sum_{i=1}^t \beta_i = 1$, a combinação linear é descrita pela Equação 3.9.

$$K' = \sum_{i=1}^t \beta_i K_i \quad (3.9)$$

Encontrar os pesos que otimizam a equação é um problema de maximização de uma função convexa, ou seja, uma função que nesse caso possui um único ponto de máximo. O problema também pode ser resolvido pela sua forma dual, ou seja, como um problema de minimização e obter resultados equivalentes.

Vários modelos supervisionados foram adotados para o aprendizado dos pesos, mas também existem versões semi-supervisionadas e não supervisionadas. Uma variação muito conhecida, considera o uso de normalizações do tipo $l_{p,q}$ antes da execução do algoritmo, conhecido como $L_{p,q}$ -MKL [94, 95]. Além disso, também existem variações que

adotam diferentes heurísticas e, inclusive, lógica nebulosa (*fuzzy*) [96]. O maior problema relacionado ao MKL é o seu alto custo computacional na maior parte dos cenários [91].

3.2.5 Abordagens Estatísticas

Ao abordar sobre os métodos de combinação de listas ranqueadas, citamos alguns algoritmos como a contagem de borda que, de certa forma, também podem ser considerados estatísticos. No entanto, esta subseção tem por objetivo apresentar métodos que possuem um embasamento estatístico mais forte e não fazem uso explícito de listas ranqueadas. Como existe uma grande variedade de aplicações, o texto se limita em apresentar apenas as abordagens principais e mais relevantes dentro do contexto do trabalho.

- **Análise de Probabilidades**

A análise de probabilidades é muito útil no contexto de combinação de características, o conceito abrangente diversas possibilidades. Uma abordagem comum consiste na combinação de classificadores bayesianos [52], os quais fornecem um conjunto de probabilidades de um elemento x pertencer a cada uma das classes existente no problema.

Seja x um elemento de uma coleção $\Omega = \{\omega_1, \dots, \omega_c\}$ o conjunto de classes existentes para solucionar um determinado problema, L classificadores organizados em conjunto $D = \{D_1, \dots, D_L\}$. Considera-se que $d_{i,j}(x)$ denota a probabilidade de x pertencer a classe j inferida pelo classificador D_i . Calculando todas as probabilidades possíveis nesse cenário, obtemos uma matriz de perfil de decisão (*decision profile*) conhecida como $DP(x)$:

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \dots & d_{1,j}(x) & \dots & d_{1,c}(x) \\ d_{L,1}(x) & \dots & d_{L,j}(x) & \dots & d_{L,c}(x) \end{bmatrix}.$$

Cada linha da matriz representa a saída de um classificador diferente e cada coluna representa as inferências para uma determinada classe fornecida por diferentes classificadores.

Diferentes estratégias não treináveis podem ser aplicadas sobre as matrizes, dentre elas o produto, a soma, regra de minimização, regra de máximo, por exemplo. No contexto de recuperação por conteúdo, pode-se considerar uma matriz de similaridades ou distâncias que é calculada a partir dos vetores de características fornecidos por diferentes descritores e aplicar essas operações sobre elas. Como essas matrizes não são propriamente de probabilidades, são consideradas apenas como uma inspiração do modelo bayesiano. Apesar disso, são de extrema utilidade em diversas aplicações de recuperação.

Existem também as estratégias treináveis que podem ser aplicadas sobre as matrizes de perfil de decisão como, por exemplo, os algoritmos baseados em modelos de decisão (*decision templates*) e combinação de Dempster-Shafer [52].

- **Análise de Correlação Canônica (CCA)**

O *Canonical Correlation Analysis* (CCA) [97] é um método que, a partir de dois vetores de características F_1 e F_2 , busca encontrar uma combinação linear dos dois de tal forma que a correlação entre as características combinadas seja máxima. O algoritmo não visa maximizar a complementaridade, mas sim detectar características de alta relevância a partir das amostras, o que pode ser inadequado dependendo do tipo de seleção desejada. Esse algoritmo é utilizado principalmente para redução de dimensionalidade.

- **Análise de Correlação Discriminante (DCA)**

Muito similar ao CCA, o *Discriminant Correlation Analysis* (DCA) [98] foi recentemente proposto e possui a mesma proposta do CCA. No entanto, o DCA considera dados rotulados para que seja aplicado. Enquanto o CCA, aumenta a correlação entre todas as amostras fornecidas, o DCA aumenta a correlação entre indivíduos pertencentes a mesma classe, mas minimiza a correlação de elementos de classes distintas.

- **Métodos de Regressão**

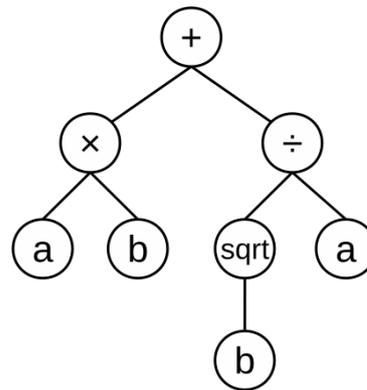
Uma regressão logística procura associar um conjunto de variáveis independentes a um conjunto de variáveis dependentes, a fim de realizar previsões sobre um novo dado. No cenário deste trabalho, pode-se por exemplo tentar prever o conteúdo de uma imagem a partir de seu vetor de características. Essas abordagens são aplicadas em cenários de combinação a fim de encontrar uma combinação linear ótima. As principais abordagens incluem LASSO (*Least Absolute Shrinkage and Selection Operator*) e a sua versão que utiliza empacotamento (*wrapper*), o *Grouped-LASSO* [99]. Ambos os métodos fazem a seleção e regularização dos valores para previsão.

3.2.6 Combinação baseada em Programação Genética

Programação genética (PG) é um caso específico de algoritmo genético no qual os indivíduos da população não são sequências de *bits*, mas sim programas de computador armazenados na forma de árvores sintáticas. Há um trabalho que apresenta o modelo de programação genética para recuperação multimídia (imagens e textos) proposto por [24]. A ideia consiste em modelar os genes de tal forma que os nós internos sejam métodos ou funções que produzam listas ranqueadas a partir dos vetores de características, representados pelos nós externos/folha (Figura 6).

Um pseudocódigo para a programação genética nesse cenário é apresentado no Algoritmo 1. Os cruzamentos e mutações são fundamentais para manter a variabilidade de combinações. Um cruzamento é feito a partir de dois pais, os nós são escolhidos aleatoriamente e troca-se a sub-árvore correspondente, originando dois filhos. Uma mutação é feita sorteando um nó aleatoriamente e uma nova sub-árvore é gerada aleatoriamente para substituir a anterior. Pelo uso de valores aleatórios, é importante ressaltar que esse

método é estocástico, tendo variabilidade de resultados entre as execuções.



Fonte: [24]

Figura 6 – Exemplo de gene na programação genética.

Algoritmo 1. Algoritmo de Programação Genética para \mathcal{N} gerações.

- 1 Gerar uma população inicial de indivíduos
 - 2 **para** \mathcal{N} gerações **fazer**
 - 3 Calcular a **aptidão** de cada indivíduo
 - 4 **Selecionar** os indivíduos para as operações genéticas
 - 5 Aplicar **cruzamentos**
 - 6 Aplicar **mutações**
 - 7 **fim para**
-

Um dos pontos cruciais para o sucesso desses métodos no cenário de recuperação de imagens é o uso de uma boa função de adaptação, a qual é responsável por inferir se os indivíduos estão “evoluindo” para formas mais eficazes de combinação.

Dentre os principais trabalhos que aplicam programação genética a cenários de CBIR, pode-se citar [24, 56, 57]. Pelo que se tem conhecimento, todas as abordagens apresentadas na literatura usam de funções de adaptação supervisionadas ou que requerem dados rotulados.

3.3 Abordagens Recentes

Dentre as mais recentes abordagens para seleção e combinação de características, nota-se que métodos baseados em grafos tem ganhado bastante atenção. O método *Graph Fusion* [100], por exemplo, constrói um grafo ponderado considerando a correlação de Jaccard como o peso das arestas e aplica o algoritmo de *PageRank* com a finalidade de atribuir incrementos entre os nós. A proposta é não supervisionada e agrega diferentes listas ranqueadas considerando a vizinhança dos nós e as referências entre si. Pode-se dizer

que o *Graph Fusion* é um algoritmo que faz combinações do tipo *late fusion* que possui resultados comparáveis ao estado-da-arte.

Uma outra abordagem é proposta em [101], na qual é construído um grafo de difusão ponderado que considera uma métrica não supervisionada de similaridade. Os valores do peso das arestas são propagados no decorrer do grafo para cada um dos nós considerando uma função (núcleo) gaussiano e os elementos são combinados ao término do processo dependendo da pontuação atribuída a cada vértice. Por ser uma estratégia que agrega listas ranqueadas, é do tipo *late fusion*. Foram alcançados resultados de estado-da-arte em várias bases e o método é não supervisionado.

Mais recentemente, os métodos tem aplicado hipergrafos para selecionar e combinar características. Em grafos comuns, as arestas podem relacionar no máximo dois vértices, nos hipergrafos elas podem combinar um conjunto de vértices, o que favorece a análise local dos dados. Em [102] os autores utilizam *Deep Canonical Correlation Analysis* (DCCA) para propor um modelo de recuperação *Cross-Media* que combina a similaridade de documentos de texto e imagens de maneira não supervisionada. O DCCA é uma versão modificada do CCA que utiliza de conceitos de *deep learning*.

Um dos grandes problemas para seleção e combinação de características é, muitas vezes, a alta dimensionalidade dos dados. A partir disso, pesquisadores utilizaram de hipergrafos para selecionar características e representá-las em estruturas de baixa dimensionalidade como o *low-rank* [103], sem supervisão. O conceito de *low-rank* consiste em dada uma matriz de alta dimensionalidade, gerar estruturas de dimensionalidade menor e, a partir delas (por operações de produto interno, por exemplo) ser possível recuperar os dados da matriz original. A vantagem desses métodos em relação a outros de redução de dimensionalidade mais comuns (PCA, LDA, CCA, etc.) é que os métodos de seleção são capazes de remover os dados irrelevantes e redundantes a partir da análise contextual dos dados. Em [104] o modelo de hipergrafo é expandido de trabalhos anteriores para utilizar também de normalizações $l_{p,q}$ e Máquinas de Vetores de Suporte (SVM).

Por enquanto, nota-se que as propostas de hipergrafos são, quase sempre, voltadas para *early fusion*. Há exemplos de seleção de hipergrafos utilizados para cenários de recuperação multimodal [105, 106], os hipergrafos consideram diferentes métricas de correlação como peso de suas arestas a fim de realizar recomendações e incrementos entre os vértices.

4 Protocolo Experimental

Este capítulo apresenta o protocolo experimental utilizado durante a avaliação experimental dos métodos apresentados nesta dissertação. O protocolo experimental é descrito precocemente com o intuito de facilitar ao leitor o entendimento das análises realizadas nos capítulos seguintes.

A descrição do protocolo experimental é organizada conforme se segue: a Seção 4.1 apresenta as coleções de imagens, os descritores e os seus respectivos resultados; a Seção 4.2 descreve os métodos considerados como *baselines*; a Seção 4.3 apresenta as métricas de estimativa de eficácia e correlação utilizadas para compor as estratégias de seleção; a Seção 4.4 descreve o método de agregação utilizado para realizar a fusão dos resultados a partir de uma combinação de ranqueadores.

4.1 Coleções e Descritores de Imagens

A análise experimental considerou cinco coleções de imagens distintas com tamanhos variando de 1.360 a 10.200 imagens, as quais são apresentadas na Tabela 4. Em geral, todas são frequentemente utilizadas em tarefas de recuperação de imagens. O MAP foi calculado considerando uma consulta para cada imagem da coleção, com exceção da Holidays na qual se considerou um conjunto específico de consultas conforme o protocolo proposto pelos autores da coleção em [107]. Além do MAP, utilizou-se *Recall@40* para a MPEG-7 e *N-S Score* para a UKBench com o objetivo de possibilitar a comparação dos resultados obtidos com o estado-da-arte.

A fim de conduzir uma análise experimental abrangente, uma grande diversidade de descritores foi utilizada, englobando diferentes categorias: globais, locais e *deep learning*. A Tabela 5 apresenta cada um deles acompanhados dos seus tipos, referências na literatura, descrição geral, número total de características e o MAP obtido em cada uma das coleções. Cada descritor forneceu um conjunto de vetores de características e as listas ranqueadas foram obtidas a partir da distância euclidiana entre os vetores. O número total de características é omitido para alguns descritores, uma vez que as listas ranqueadas foram cedidas por outros pesquisadores. Note que, para todas as coleções de imagens, o conjunto de descritores utilizados é similar, com exceção da MPEG-7 que apresenta um cenário de recuperação mais específico e, sendo assim, optou-se por utilizar apenas descritores de contorno. Os descritores de contorno (exceto PHOG) foram aplicados apenas para a MPEG-7, uma vez que eles foram propostos para serem aplicados sobre imagens de contorno já segmentadas, como ocorre na coleção MPEG-7.

Tabela 4 – Coleções de imagens utilizadas na avaliação experimental.

Coleção	Tamanho	Tipo	Descrição Geral	Métrica
Flowers [108]	1.360	Flores	Composta de 17 espécies de flores com 80 imagens cada, apresentado variações de posição e iluminação. A coleção é distribuída pela Universidade de Oxford.	MAP
MPEG-7 [109]	1.400	Contorno	Composta de 1.400 imagens de contorno divididas em 70 classes. Frequentemente utilizada para avaliação de métodos de pós-processamento.	MAP, <i>Recall@40</i>
Holidays [107]	1.491	Cenas	Frequentemente utilizada como referência em tarefas de recuperação de imagens pelo conteúdo, a coleção é composta de 1.491 fotos de férias/viagens compondo 500 consultas distintas.	MAP
Corel5k [110]	5.000	Objetos/ Cenas	Composta de 50 categorias com 100 imagens cada, incluindo cenas diversas como fogos de artifício, barcos, imagens de microscópio, azulejos, árvores, etc.	MAP
UKBench [111]	10.200	Objetos/ Cenas	Composta de 2.550 objetos e cenas. Cada objeto/cena possui 4 fotos capturadas a partir de diferentes ângulos, distâncias e condições de iluminação.	MAP, <i>N-S Score</i>

Tabela 5 – Descritores de imagens utilizados na avaliação experimental.

Categoria	Tipo	Descritor	Descrição Geral (Total de Características)	MAP (%)				
				MPEG-7	Flowers	Corel5k	UKBench	Holidays
Global	Cor	ACC [35]	<i>Auto Color Correlogram</i> (256 caract.)	—	18.99	23.44	87.72	64.29
		SPACC [35, 112]	<i>Spatial Pyramid ACC</i> (5376 caract.)	—	19.20	23.86	85.30	62.37
		CLD [113]	<i>Color Layout Descriptor</i> (33 caract.)	—	18.54	17.86	59.58	37.59
		SCD [113]	<i>Scalable Color Descriptor</i> (64 caract.)	—	10.25	14.56	83.04	54.26
		SCH [113]	<i>Simple Color Histogram</i> (64 caract.)	—	13.43	17.56	48.98	24.19
		FOH [114, 112]	<i>Fuzzy Opponent Histogram</i> (576 caract.)	—	11.42	15.87	57.05	25.77
		BIC [37]	<i>Border/Interior Pixel Classification</i> (128 caract.)	—	25.56	—	80.46	—
		PHOG [45, 112]	<i>Pyramidal Histogram of oriented gradients</i> (630 caract.)	—	—	14.74	15.80	41.60
	Contorno	AIR [115]	<i>Articulation-Invariant Representation</i>	89.39	—	—	—	—
		ASC [41]	<i>Aspect Shape Context</i>	85.28	—	—	—	—
		IDSC [116]	<i>Inner Distance Shape Context</i>	81.70	—	—	—	—
		CFD [40]	<i>Contour Features Descriptor</i>	80.71	—	—	—	—
		BAS [34]	<i>Beam Angle Statistics</i>	71.42	—	—	—	—
Textura	SS [39]	<i>Segment Saliences</i>	37.82	—	—	—	—	
	LBP [42]	<i>Local Binary Patterns</i> (256 caract.)	—	10.34	14.83	47.19	28.82	
	SPLBP [42, 112]	<i>Spatial Pyramid LBP</i> (756 caract.)	—	10.92	15.41	52.14	33.09	
Cor e Textura	EHD [117]	<i>Edge Histogram Descriptor</i> (80 caract.)	—	12.46	16.80	44.10	25.83	
	CEDD [118]	<i>Color and Edge Directivity Descriptor</i> (144 caract.)	—	20.48	23.00	70.45	51.59	
	SPCEDD [118, 112]	<i>Spatial Pyramid CEDD</i> (3024 caract.)	—	21.94	28.70	74.98	56.09	
	FCTH [119]	<i>Fuzzy Color and Texture Histogram</i> (192 caract.)	—	20.56	23.93	73.70	48.44	
	SPFCTH [119, 112]	<i>Spatial Pyramid FCTH</i> (4032 caract.)	—	21.73	26.43	77.78	55.43	
	JCD [120]	<i>Joint Composite Descriptor</i> (168 caract.)	—	20.89	24.73	74.85	52.84	
	SPJCD [120, 112]	<i>Spatial Pyramid JCD</i> (3528 caract.)	—	22.56	28.02	76.67	56.58	
	COMO [50]	<i>Compact Composite Moment-Based Descriptor</i> (144 caract.)	—	21.83	21.05	79.77	49.66	
Holístico	GIST [121]	<i>Global Image Descriptor for low-dim. features</i> (960 caract.)	—	9.82	15.98	45.44	21.59	
Local	Bag of Words	SIFT [46]	<i>Scale-Invariant Feature Transform with VLAD</i> (8192 caract.)	—	28.47	12.60	74.52	54.63
		VOC [122]	<i>Vocabulary Tree</i>	—	—	—	91.14	—
Deep Learning	Rede Neural Convol.	CNN-SENet [123]	<i>154-layers Squeeze-and-Excitation Neural Network</i> (2048 caract.)	—	43.16	56.92	92.15	71.60
		CNN-ResNet [124]	<i>152-layers Residual Neural Network</i> (2048 caract.)	—	51.83	64.81	94.54	74.88
		CNN-FBResNet [124]	<i>152-layers ResNet trained by Facebook AI Research</i> (2048 caract.)	—	52.56	64.21	93.88	72.65
		CNN-ResNeXt [125]	<i>101-layers "Next Generation" ResNet</i> (2048 caract.)	—	51.91	62.39	93.67	74.16
		CNN-DPNet [126]	<i>92-layers Dual Path Neural Network</i> (1000 caract.)	—	50.93	65.15	90.47	70.59
		CNN-VGGNet [127]	<i>19-layers VGG Neural Network</i> (4096 caract.)	—	39.05	47.85	87.99	67.96
		CNN-BnVGGNet [127]	<i>19-layers Binaural VGG Neural Network</i> (4096 caract.)	—	41.87	52.72	89.24	67.60
		CNN-InceptionV4 [128]	<i>Fourth version of the Inception Neural Network</i> (1536 caract.)	—	42.35	58.66	86.82	63.84
		CNN-InceptionResNet [128]	<i>Inception architecture with residual connections</i> (1536 caract.)	—	42.20	61.17	87.23	62.87
		CNN-BnInception [129]	<i>Binaural Inception Neural Network</i> (1024 caract.)	—	46.58	46.60	91.84	70.06
		CNN-NASnet-Large [130]	<i>Convolutional Neural Architecture Search Network</i> (4032 caract.)	—	40.74	53.55	86.90	64.48
		CNN-AlexNet [131]	<i>Alex Krizhevsky Convolutional Neural Network</i> (4096 caract.)	—	46.04	37.67	85.57	65.25
		CNN-Xception [132]	<i>Depthwise Separable Convolutions Neural Network</i> (2048 caract.)	—	47.31	54.44	90.83	64.94
	Rede Híbrida	CNN-OLDFP [133]	<i>Object Level Deep Feature Pooling</i> (4096 caract.)	—	—	—	97.74	88.46

Os resultados de *deep learning* foram obtidos através do *PyTorch* [134], uma das mais populares plataformas de aprendizado de máquina e completamente *open-source*. Todas as redes foram treinadas na coleção ImageNet [135], a qual é popularmente utilizada

para treinamento de classificadores e redes neurais de propósito geral. Foram considerados os valores fornecidos pela penúltima camada de cada uma das redes, uma vez que esta apresentou resultados superiores aos da camada de classificação. Para a *CNN-OLDFP*, os resultados utilizados são fornecidos por [133], a rede mescla conceitos de *deep learning* à técnicas de *bag of words* a fim de otimizar a eficácia dos resultados.

Durante a análise experimental, diferentes cenários de seleção foram avaliados: (i) os **completos** que consideram todos os descritores; (ii) somente descritores **globais e locais**; (iii) apenas **deep learning**; e (iv) os cenários **personalizados**, os quais consideram um conjunto de seis descritores diversificados para cada coleção. O cenário personalizado é formado pelos dois descritores de *deep learning* mais eficazes, o descritor local mais eficaz e os três globais mais eficazes de tipos diferentes. Esse critério não se aplica à MPEG-7, visto que há poucos descritores. A Tabela 6 apresenta os cenários personalizados para cada uma das coleções, os descritores são apresentados em ordem decrescente de MAP.

Tabela 6 – Descritores considerados nos cenários personalizados.

Coleção	Descritores
MPEG-7	AIR, ASC, IDSC, CFD, BAS, SS
Flowers	CNN-FBResNet, CNN-ResNeXt, SIFT, BIC, SPJCD, PHOG
Corel5k	CNN-DPNet, CNN-ResNet, SIFT, SPACC, SPCEDD, EHD
UKBench	CNN-OLDFP, CNN-ResNet, VOC, ACC, COMO, SPLBP
Holidays	CNN-OLDFP, CNN-ResNet, SIFT, ACC, SPJCD, SPLBP

4.2 Baselines

Com o propósito de apresentar uma análise experimental o mais completa possível, tanto abordagens de *late fusion* quanto de *early fusion* foram utilizadas como *baselines*. Todos os métodos apresentados nesta seção são completamente não supervisionados.

A Tabela 7 descreve os métodos considerados como *baselines* para *late fusion*. Tais métodos foram escolhidos por serem recentes, apresentarem resultados comparáveis ao estado-da-arte e possuírem código fonte publicamente disponível. Dessa forma, é possível fazer uma comparação justa com os métodos propostos, utilizando os mesmos dados como entrada para todos os métodos.

Cinco métodos de *early fusion* diferentes foram considerados como *baseline* neste trabalho: *Laplacian Score* [25], *Spectral Regression* (SPEC) [26], *Muti-cluster Feature Selection* (MCFS) [27], *Unsupervised Discriminative Feature Selection* (UDFS) [28], *Nonnegative Discriminative Feature Selection* (NDFS) [29]. Sendo os dois primeiros baseados em métricas de similaridade e os demais no processamento de matrizes esparsas. Todos estão disponíveis em código aberto na biblioteca de seleção de características *scikit-feature* [139]¹.

¹ featureselection.asu.edu

Tabela 7 – Métodos considerados como *baselines* para *late fusion*.

Método	Descrição Geral
<i>Correlation Graph</i> [136]	Considera a geometria intrínseca da coleção a fim de definir uma distância mais efetiva entre as imagens. Dentre as várias estratégias empregadas, o método constrói um grafo e analisa as suas componentes fortemente conexas.
<i>Query-Adaptive Fusion</i> [137]	A partir de uma matriz que fornece a similaridade entre os elementos de uma coleção, a eficácia de um descritor é estimada como inversamente proporcional à área abaixo da curva de similaridade para cada elemento da coleção. As estimativas são utilizadas para calcular valores de similaridade mais eficazes, os quais são fornecidos como saída do método.
<i>Graph Fusion</i> [100]	Para cada um dos arquivos de entrada, um grafo ponderado e não direcionado é construído considerando cada uma das imagens de consulta. Os grafos são combinados considerando diferentes técnicas, incluindo o algoritmo de <i>PageRank</i> [138].

No modelo de recuperação considerado neste trabalho, os métodos de *early fusion* apresentaram diversas desvantagens em relação aos de *late fusion*, dentre as quais podem ser citadas: maior custo computacional, se demonstraram sensíveis a vetores esparsos e redução considerável de eficácia caso os tamanhos dos vetores de características fossem diferentes entre os descritores. Dessa forma, antes de executar a seleção com qualquer um desses métodos, os vetores de características tiveram suas dimensionalidades reduzidas através de PCA e foram posteriormente normalizados considerando uma regularização L2-norm [140]. Tal abordagem aumentou a eficácia e a eficiência dos *baselines* de *early fusion* em todos os casos.

4.3 Métricas para Seleção

Esta seção apresenta as métricas utilizadas neste trabalho para compor as estratégias de seleção de combinações, sendo que nenhuma delas requer de dados rotulados. A Seção 4.3.1 descreve as estimativas de eficácia e a Seção 4.3.2 descreve as métricas de correlação.

4.3.1 Estimativas de Eficácia

As estimativas de eficácia são fundamentais, tendo como objetivo fornecer um valor que deduza a eficácia de uma lista ranqueada sem o uso de dados rotulados. Elas podem ser utilizadas visando uma seleção não supervisionada das combinações.

A seguir são apresentadas as estimativas de eficácia utilizadas neste trabalho. O princípio de tais métricas consiste em explorar a hipótese de agrupamento (*cluster hypothesis*) [141] que baseia-se na ideia de que itens similares tendem a ser relevantes entre si e, portanto, possuem muitos elementos em comum nas primeiras k posições. Os valores

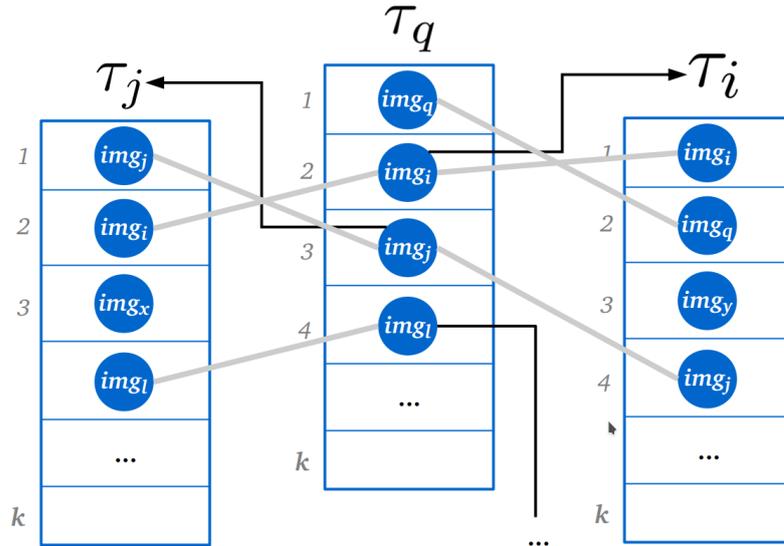
são calculados no intervalo de $[0, 1]$, sendo quanto maior o valor melhor o resultado.

- **Authority Score**

O *Authority Score* (A_s) [142], ou métrica de autoridade, fornece uma estimativa de eficácia não supervisionada que explora a densidade do grafo de vizinhança em uma lista ranqueada de uma imagem q . Seja $\mathcal{N}(q, k)$ o conjunto natural dos primeiros k elementos de τ_q e $f_{in}(j, q)$ uma função que retorna 1 se $img_j \in \mathcal{N}(q, k)$. A pontuação é definida pela Equação 4.1 e obtida no intervalo $[0, 1]$ (quanto maior o valor, maior a autoridade).

$$Authority(q, k) = \frac{\sum_{i \in \mathcal{N}(q, k)} \sum_{j \in \mathcal{N}(i, k)} f_{in}(j, q)}{k^2} \quad (4.1)$$

A Figura 7 ilustra a coocorrência de vizinhos recíprocos no grafo de vizinhança da lista ranqueada da imagem de consulta (τ_q). No exemplo apresentado, nota-se que a imagem img_q possui três vizinhos recíprocos com as imagens img_i e img_j .



Fonte: Figura retirada de [142].

Figura 7 – Ilustração da coocorrência de vizinhos recíprocos no grafo de vizinhança.

- **Reciprocal Density**

O *Reciprocal Density* [143] também explora o grafo de vizinhança, mas atribuindo pesos para cada ocorrência de vizinho recíproco. A definição formal é dada pela Equação 4.2.

$$Reciprocal(q, k) = \frac{\sum_{i \in \mathcal{N}(q, k)} \sum_{j \in \mathcal{N}(i, k)} f_{in}(j, q) \times w_r(q, i) \times w_r(i, j)}{k^4} \quad (4.2)$$

A função $f_{in}(j, q) \rightarrow \{0, 1\}$ determina se $img_j \in \mathcal{N}(q, k)$. Um peso é calculado de acordo com a função $w_r(q, i) = k + 1 - \tau_q(i)$. Quanto maior o peso, maior tende a ser a ocorrência de vizinhos recíprocos nas primeiras posições da lista ranqueada.

4.3.2 Métricas de Correlação

Métricas de correlação são utilizadas com o objetivo de calcular um valor de similaridade entre listas ranqueadas. Elas são úteis no contexto de seleção por fornecerem um valor que estima a complementaridade de dados de origem distintas.

- **Pearson**

Seja $\mathcal{N}_k(q)$ o conjunto contendo os k primeiros vizinhos de um dada imagem da coleção img_q e $\mathcal{N}_k(q, j) = \mathcal{N}_k(q) \cup \mathcal{N}_k(j)$. Sejam X e Y os vetores que contêm as distâncias das imagens img_q , img_j e $img_i \in \mathcal{N}_k(q, j)$: $X_i = \rho(q, i)$ e $Y_i = \rho(j, i)$, em que \bar{X} e \bar{Y} são a média dos vetores. A correlação de Pearson é definida pela Equação 4.3.

$$Pearson(q, j) = \frac{\sum_{i=1}^{k_u} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{k_u} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{k_u} (Y_i - \bar{Y})^2}}. \quad (4.3)$$

A correlação de Pearson fornece um resultado no intervalo $[-1, 1]$. Se a correlação é positiva, indica que a disposição dos valores tende a uma formação linear crescente. Caso seja negativa, a disposição dos valores tende a uma formação linear decrescente. Se for igual a zero, não há correlação entre os dados fornecidos.

- **Jaccard**

O índice de Jaccard é uma métrica estatística que calcula a correlação entre duas listas ranqueadas e é definida como demonstrado na Equação 4.4.

$$Jaccard(\tau_i, \tau_j) = \frac{|\tau_i \cap \tau_j|}{|\tau_i \cup \tau_j|}. \quad (4.4)$$

No cenário de recuperação de imagens, as listas ranqueadas não possuem elementos repetidos, podendo ser tratadas como conjuntos. Dados dois conjuntos, a métrica conta o número de elementos em comum e divide pelo total de elementos distintos das duas listas.

A métrica retorna um valor no intervalo de $[0, 1]$. Quanto maior a correlação das listas, maior o valor. Dessa maneira, o valor obtido pode ser tratado como uma medida de similaridade entre as listas de entrada.

- **Jaccard_k**

Tradicionalmente, o Jaccard realiza sua análise em uma única profundidade k de tal forma que todos os elementos são tratados igualmente. Em [144], uma modificação da métrica é proposta a fim de considerar diferentes profundidades. A ideia consiste em atribuir pesos maiores às primeiras posições. A métrica é definida na Equação 4.5.

$$Jaccard_k(\tau_i, \tau_j, k) = \frac{\sum_{d=1}^k Jaccard(\tau_i, \tau_j, d)}{k}. \quad (4.5)$$

- **RBO**

O *Rank-Biased Overlap* (RBO) [145] também considera a sobreposição entre os primeiros k elementos em diferentes profundidades. No entanto, diferente das métricas de intersecção, o peso atribuído para cada sobreposição é calculado baseado nas probabilidades definidas em cada profundidade. Sendo assim, a métrica pode ser definida como:

$$RBO(\tau_i, \tau_j, k, p) = (1 - p) \sum_{d=1}^k p^{d-1} \times \frac{|\mathcal{N}(i, k) \cap \mathcal{N}(j, k)|}{d}, \quad (4.6)$$

em que $p = 0.9$ foi utilizado como o valor padrão para todos os experimentos, conforme adotado em outros trabalhos [146].

- **Spearman**

A métrica de Spearman é não-paramétrica, a qual avalia a relação entre pares de variáveis. Pode ser entendida como a distância L1 entre duas permutações em que se considera a diferença das posições entre os elementos [147]. Formalmente é definida como:

$$Spearman(\tau_i, \tau_j, k) = 1 - \frac{\sum_{x, y \in \mathcal{N}(i, k) \cup \mathcal{N}(j, k)} s(x, y)}{2 \times k^2}, \quad (4.7)$$

em que $s(x, y)$ é a diferença entre as posições:

$$s(x, y) = |\max(k + 1, \tau_x(y)) - \max(k + 1, \tau_y(x))| \quad (4.8)$$

O cálculo da distância foi restrito as k posições em virtude do tamanho do conjunto de vizinhança.

- **Kendall τ**

O Kendall τ é uma métrica de correlação que analisa permutações de pares de elementos. Em outras palavras, pode ser entendida como o número de trocas necessárias em uma ordenação bolha (*bubble sort*) para converter uma permutação a outra [147]. Pode-se defini-la como uma métrica de correlação de listas ranqueadas, como se segue:

$$Kendall_\tau(\tau_i, \tau_j, k) = 1 - \frac{\sum_{x, y \in \mathcal{N}(i, k) \cup \mathcal{N}(j, k)} \bar{K}_{x, y}(\tau_i, \tau_j)}{k \times (k - 1)}, \quad (4.9)$$

em que $\bar{K}_{x, y}(\tau_i, \tau_j)$ é a função que determina se os objetos o_x e o_y estão na mesma ordem nas primeiras k posições de τ_i e τ_j . Formalmente, a função é definida como:

$$\bar{K}_{x, y}(\tau_i, \tau_j) = \begin{cases} 0 & \text{se } (\tau_i(x) \leq \tau_i(y) \wedge \tau_j(x) \leq \tau_j(y)), \\ 0 & \text{se } (\tau_i(x) \geq \tau_i(y) \wedge \tau_j(x) \geq \tau_j(y)), \\ 1 & \text{caso contrário.} \end{cases} \quad (4.10)$$

Para realizar o cálculo da função \bar{K} , considera-se a posição máxima como k (adota-se $k + 1$ para posições acima de k).

4.4 Método de Agregação

Os métodos propostos neste trabalho requerem de um método agregador para realizar a fusão dos ranqueadores que compõem a combinação selecionada. Cada ranqueador fornece um conjunto de listas ranqueadas distinto e o método de agregação possui o objetivo de processá-las e fornecer as listas de resultados de recuperação para cada uma das imagens de consulta. As abordagens propostas neste trabalho são flexíveis, podendo utilizar de diferentes métodos para realizar a agregação dos ranqueadores. Neste trabalho, é utilizado o CPRR (*Cartesian Product of Ranking References*) [20], um método muito eficaz e eficiente que apresenta resultados comparáveis ao estado-da-arte. Dentre as propostas de trabalhos futuros, está avaliar outros métodos de agregação.

O CPRR tem como ideia central o uso do produto cartesiano com o objetivo de maximizar a informação contextual codificada nas listas ranqueadas. Considera-se, por razões de eficiência e escalabilidade, apenas um subconjunto das L primeiras imagens das listas ranqueadas. O algoritmo trabalha com medidas de similaridade. A Figura 8 apresenta um diagrama com as etapas que compõem o método.

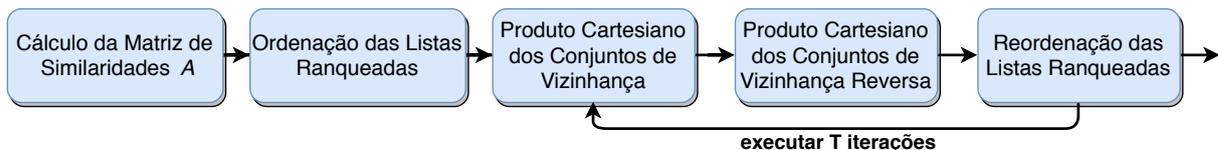


Figura 8 – Etapas do algoritmo CPRR.

Sendo assim, o algoritmo pode ser resumido em duas etapas principais:

- **Preenchimento da Matriz Esparsa de Similaridades:** a partir das listas ranqueadas, pode-se preencher uma matriz de similaridades esparsa na qual a similaridade é computada como a soma das posições das imagens em suas listas ranqueadas correspondentes.
- **Produto Cartesiano das Referências:** é realizado o produto cartesiano para os conjuntos de vizinhança natural e conjunto de vizinhança reversa, realizando a multiplicação das posições. O conjunto de vizinhança natural corresponde às k primeiras imagens de uma lista ranqueada. No caso do conjunto de vizinhança reversa, são consideradas as referências de uma imagem nas listas ranqueadas de todas as imagens do conjunto.

No cenário de *rank aggregation*, o método executa uma iteração separada para cada um dos ranqueadores de entrada e soma esses valores em uma matriz de similaridades A que é utilizada para as T iterações restantes.

5 Seleção por Métricas Par-a-par

Este capítulo apresenta o USRF (*Unsupervised Selective Rank Fusion*), abordagem proposta que possui como objetivo realizar a seleção e combinação de ranqueadores de maneira completamente não supervisionada. Tal tarefa é realizada através da aplicação de métricas para pares de ranqueadores. Considerando cenários em que vários descritores visuais são extraídos e não há dados de treinamento ou *feedback* do usuário, é uma tarefa extremamente desafiadora selecionar e combinar os descritores de forma a produzir resultados de recuperação mais eficazes. Tal dificuldade dá-se em função de dois principais fatores: (i) não há como estabelecer qualquer métrica de seleção em função de dados rotulados e (ii) o custo computacional para a avaliação de todas as combinações é, muitas vezes, proibitivo. Um diagrama que ilustra os dados de entrada e de saída é apresentado pela Figura 9. Nota-se que ranqueadores de diferentes categorias podem ser utilizados (descritores de cor, redes neurais convolucionais, abordagens locais, entre outros) com o intuito de explorar a complementaridade dos dados.

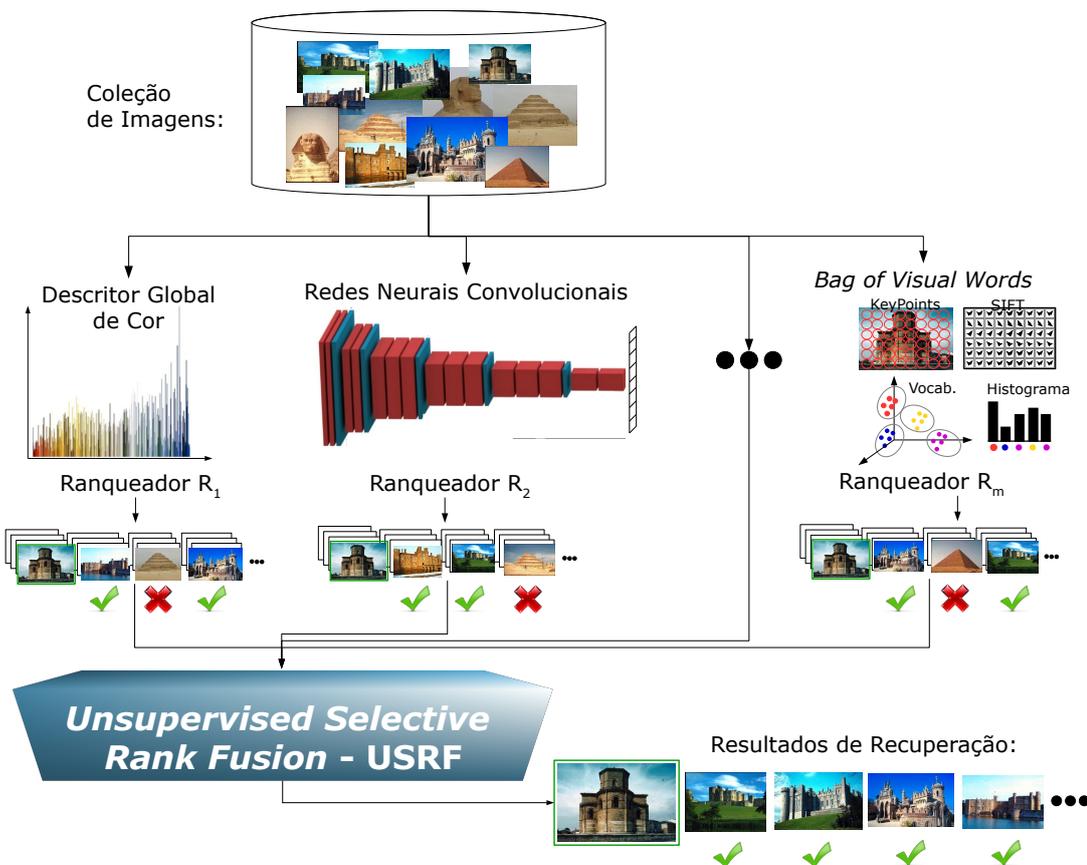


Figura 9 – Visão geral do método USRF.

Dado um conjunto de m ranqueadores $\mathfrak{R} = \{R_1, R_2, \dots, R_m\}$ como entrada do método, define-se $\mathfrak{R}^2 = \mathfrak{R} \times \mathfrak{R}$ como o conjunto de todos os possíveis pares não ordenados

Tabela 8 – Análise do tamanho dos conjuntos de combinações.

Conjunto	Tamanho do Conjunto				
	5	10	15	20	25
Ranqueadores (\mathfrak{R})	5	10	15	20	25
Pares Não Ordenados (\mathcal{S}_p)	10	45	105	190	300
Todas as Combinações (\mathcal{S})	26	1.013	32.752	1.048.555	33.554.406

$\{R_i, R_j\}$ de ranqueadores. Considera-se combinação um conjunto de dois ou mais ranqueadores. Sendo assim, pode-se generalizar que $\mathfrak{R}^n = \prod_1^n \mathfrak{R}$ contém todas as combinações de tamanho n . O conjunto de seleção disponível ao USRF é definido como $\mathcal{S} = \cup_{i=1}^m \mathfrak{R}^i$. A tarefa de seleção é muito complexa, uma vez que o conjunto de todas as combinações \mathcal{S} pode ser extremamente grande, mesmo para um pequeno \mathfrak{R} de entrada. Consequentemente, a estratégia baseia-se na seleção dos pares, a qual é realizada em $\mathcal{S}_p = \mathfrak{R}^2$. A Tabela 8 exhibe o tamanho dos conjuntos \mathcal{S}_p e \mathcal{S} conforme \mathfrak{R} aumenta. Nota-se que o tamanho de \mathcal{S} cresce exponencialmente em relação ao \mathfrak{R} e que \mathcal{S}_p é muito menor do que \mathcal{S} .

5.1 Descrição do Método

A Figura 10 apresenta cada uma das etapas do USRF e a ordem em que estas acontecem. A entrada do método consiste em conjuntos de listas ranqueadas (conjunto \mathfrak{R}) que podem ter diferentes origens. Neste trabalho são considerados conjuntos de listas ranqueadas gerados a partir de descritores visuais distintos. A partir disso, é calculada a estimativa de eficácia (1) e a correlação para cada uma das listas (2). Em seguida, é aplicada uma métrica cujo objetivo consiste em selecionar a melhor combinação (3) em função dos cálculos realizados em (1) e (2). Note que a combinação selecionada é denotada por \mathfrak{X}^* . Em (4) os ranqueadores selecionados são utilizados como entrada de um método de agregação de listas ranqueadas (*rank-aggregation*), o qual faz o reranqueamento e fornece o resultado de recuperação final. As próximas subseções detalham a estratégia de seleção.

5.1.1 Estratégia de Seleção

Qualquer conjunto de dois ou mais ranqueadores define uma combinação, a qual é denotada por \mathfrak{X}_n^i em que i e n indicam o índice e o tamanho da combinação, respectivamente. Conforme descrito no início do capítulo, dado um conjunto \mathfrak{R} que contém m ranqueadores, o USRF busca as melhores combinações no conjunto $\mathcal{S} = \cup_{i=1}^m \mathfrak{R}^i$. O conjunto a ser explorado na seleção de pares é denotado por $\mathcal{S}_p = \mathfrak{R}^2$. Sempre que o texto se referir a pares está se referindo a pares não ordenados. Como o conjunto \mathcal{S} geralmente é muito grande e custoso de se calcular, a seleção em \mathcal{S} é baseada na seleção dos pares.

A seguir o texto é organizado como se segue: a Seção 5.1.1.1 descreve a seleção de pares e define a principal métrica que atribui uma pontuação para os pares; a Seção 5.1.1.2

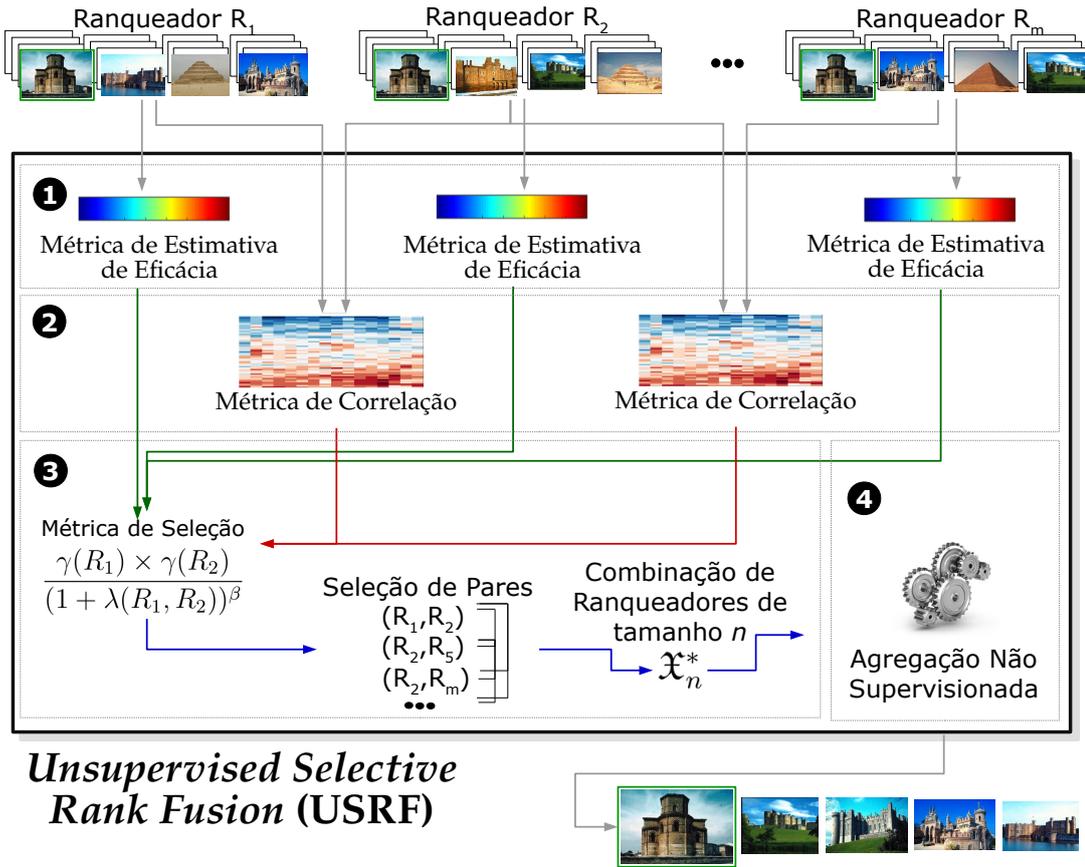


Figura 10 – Ilustração das etapas que compõem o USRF.

redige sobre a seleção considerando combinações de diferentes tamanhos (no conjunto \mathcal{S}) e como esta deriva da seleção de pares.

5.1.1.1 Seleção de Pares de Ranqueadores

A proposta tem como base o cálculo da métrica de seleção não supervisionada, a qual atribui uma pontuação para pares de ranqueadores. Esta se baseia em duas hipóteses distintas:

1. Quanto maior a estimativa de eficácia de um ranqueador, maiores as chances de este fornecer um resultado efetivo se combinado aos demais;
2. Quanto menor a correlação entre dois ranqueadores, maiores as chances de informações complementares, as quais podem ser combinadas para obter um resultado eficaz.

Tais hipóteses foram apresentadas e analisadas em [148]. No entanto, a métrica em [148] precisa de dados rotulados devido à utilização de métricas de eficácia que requerem dados rotulados (MAP e Precisão), enquanto o método discutido neste capítulo utiliza métricas não supervisionadas de estimativa de eficácia. A seguir cada uma das conjecturas é descrita em detalhes para a formulação da métrica de seleção final.

- **Seleção por Estimativa de Eficácia**

A primeira hipótese parte do pressuposto de que uma combinação relevante é realizada por listas ranqueadas de alta eficácia. Seja Γ a métrica de seleção por eficácia aplicada a um par $\{R_1, R_2\}$ e $\gamma(R_i)$ uma função que retorna a estimativa de eficácia das listas ranqueadas fornecidas pelo ranqueador R_i . A métrica é definida como:

$$\Gamma(R_1, R_2) = \gamma(R_1) \times \gamma(R_2). \quad (5.1)$$

Note que $\gamma(R_i)$ pode se referir a qualquer uma das métricas apresentadas na Seção 4.3.1.

- **Seleção por Correlação**

Outra hipótese é de que listas com baixa correlação tem maior potencial de combinação por fornecerem dados complementares. Seja Δ a métrica de seleção por correlação e $\rho(R_1, R_2)$ uma função que retorna a correlação (similaridade no intervalo $[0, 1]$) entre as listas ranqueadas fornecidas pelos ranqueadores R_1 e R_2 . A métrica é definida pela Equação 5.2.

$$\Lambda(R_1, R_2) = \frac{1}{1 + \lambda(R_1, R_2)}. \quad (5.2)$$

A função $\lambda(R_1, R_2)$ pode considerar como métrica de correlação qualquer uma das apresentadas na Seção 4.3.2 (Jaccard, RBO, Spearman ou Kendall τ , por exemplo).

- **Métrica de Seleção**

A métrica de seleção para pares de ranqueadores é formulada como a junção das duas equações anteriores, com o objetivo de selecionar pares de alta eficácia e baixa correção (alta complementaridade). A métrica w_p é definida como:

$$w_p(\{R_1, R_2\}) = \Gamma(R_1, R_2) \times \Lambda(R_1, R_2)^\beta = \frac{\gamma(R_1) \times \gamma(R_2)}{(1 + \lambda(R_1, R_2))^\beta}, \quad (5.3)$$

em que o expoente β é utilizado com o propósito de aplicar um peso à seleção por correlação, o qual possibilita o uso da métrica em diferentes cenários. Ao longo dos experimentos, constatou-se que para os casos em que há um grande número de ranqueadores, é mais vantajoso utilizar a correlação a fim de combinar listas ranqueadas similares. Sendo assim, pode-se utilizar $\beta = 1$ para cenários com menor diversidade e $\beta = -1$ para cenários em que há grande variedade de descritores, conforme discutido em mais detalhes na avaliação experimental.

• Seleção dos Pares

Por fim, os pares podem ser ordenados em ordem decrescente a partir de w_p , a fim de se obter uma lista ranqueada de pares, a qual é denotada por τ_2^R . Formalmente, a lista de seleção $\tau_2^R = (\mathfrak{X}_2^1, \mathfrak{X}_2^2, \dots, \mathfrak{X}_2^{L_R})$ pode ser definida como a permutação de \mathfrak{R}^2 . A permutação τ_2^R é uma bijeção do conjunto \mathfrak{R}^2 em $[L_R] = \{1, 2, \dots, L_R\}$. Para uma permutação τ_2^R , interpreta-se $\tau_2^R(\mathfrak{X}_2^i)$ como a posição de \mathfrak{X}_2^i em τ_2^R . Pode-se dizer que se $\tau_2^R(\mathfrak{X}_2^i) \leq \tau_2^R(\mathfrak{X}_2^j)$ então $w_p(\mathfrak{X}_2^i) \geq w_p(\mathfrak{X}_2^j)$. Dessa forma, o elemento na primeira posição é o mais eficaz e assim sucessivamente. De maneira geral, o par selecionado pode ser definido pela equação:

$$\mathfrak{X}_2^* = \arg \max_{\mathfrak{X}_2^i \in \tau_2^R} w_p(\mathfrak{X}_2^i). \tag{5.4}$$

5.1.1.2 Seleção de Conjuntos de Ranqueadores

Tendo como base a seleção de pares, esta seção estende a abordagem para combinações de qualquer número de ranqueadores. O método aplica a seleção para combinações de qualquer tamanho através da união dos pares mais relevantes, o que é imprescindível para viabilizar a seleção para grandes conjuntos de ranqueadores.

A Figura 11 ilustra um exemplo hipotético de aplicação do algoritmo de seleção de combinações de ranqueadores. Pode-se observar, por exemplo, que a combinação $\{R_1, R_2, R_3\}$ é originada a partir dos pares $\{R_2, R_3\}$, $\{R_1, R_2\}$, $\{R_1, R_3\}$ que aparecem em τ_2^R e que o valor de w corresponde à soma do w_p dos pares originários. Além disso, nota-se que o par $\{R_4, R_5\}$ não origina nenhuma combinação em C_3 , uma vez que este par não possui intersecção com nenhum dos pares disponíveis em τ_2^R . O mesmo processo se repete para a obtenção de combinações de tamanho quatro e assim sucessivamente.

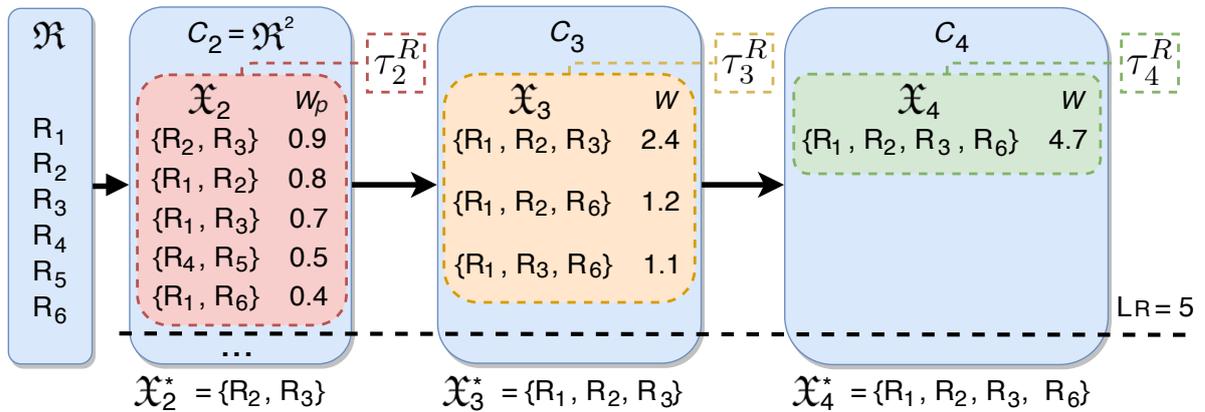


Figura 11 – Ilustração do algoritmo de seleção proposto.

O conjunto C_n pode ser entendido como aquele que contém todas as combinações de tamanho n formadas a partir da união das combinações de tamanho $n - 1$ pertencentes

à τ_{n-1}^R . Formalmente, este conjunto C_n é definido pela Equação 5.5. Ressalta-se que, para um determinado tamanho, pode acontecer do conjunto C vir a ser vazio.

$$C_n = \begin{cases} \mathfrak{R}^2 & \text{se } n = 2 \\ \{\mathfrak{X}_{n-1}^i \cup \mathfrak{X}_{n-1}^j \mid \mathfrak{X}_{n-1}^i, \mathfrak{X}_{n-1}^j \in \tau_{n-1}^R \wedge |\mathfrak{X}_{n-1}^i \cup \mathfrak{X}_{n-1}^j| = n\} & \text{se } n \geq 3 \end{cases} \quad (5.5)$$

Como a função w_p define uma pontuação apenas para pares de ranqueadores, esta pode ser generalizada para combinações de qualquer tamanho, conforme apresentado pela Equação 5.6. Aplica-se um somatório dos valores de w das combinações de tamanho $n - 1$ recursivamente, o qual tem como caso base o valor obtido em w_p .

$$w(\mathfrak{X}_n) = \begin{cases} w_p(\mathfrak{X}_n) & \text{se } n = 2 \\ \sum_{\mathfrak{X}_{n-1}^i \in \tau_{n-1}^R} w(\mathfrak{X}_{n-1}^i) & \text{se } n \geq 3 \end{cases} \quad (5.6)$$

Pode-se calcular o valor de w para todas combinações da coleção C_n e ordená-las de maneira decrescente a fim de se obter uma lista ordenada τ_n^R , a qual denominamos de lista de seleção. Formalmente, a lista de seleção $\tau_n^R = (\mathfrak{X}_n^1, \mathfrak{X}_n^2, \dots, \mathfrak{X}_n^{L_R})$ pode ser definida como a permutação de $C_n \subset \mathfrak{R}^n$. A permutação τ_n^R é uma bijeção do conjunto C_n em $[L_R] = \{1, 2, \dots, L_R\}$. Para uma permutação τ_n^R , interpreta-se $\tau_n^R(\mathfrak{X}_n^i)$ como a posição de \mathfrak{X}_n^i em τ_n^R . Pode-se dizer que se $\tau_n^R(\mathfrak{X}_n^i) \leq \tau_n^R(\mathfrak{X}_n^j)$ então $w(\mathfrak{X}_n^i) \geq w(\mathfrak{X}_n^j)$.

Sendo assim, generalizando o algoritmo de seleção, a obtenção da combinação selecionada (\mathfrak{X}_n^*) para qualquer tamanho (n) é definida pela Equação 5.7. Alternativamente, a combinação selecionada \mathfrak{X}_n^* também pode ser entendida como a primeira colocada na lista de seleção τ_n^R , ou seja, $\tau_n^R(1)$.

$$\mathfrak{X}_n^* = \arg \max_{\mathfrak{X}_n^i \in \tau_n^R} w(\mathfrak{X}_n^i). \quad (5.7)$$

O Algoritmo 2 apresenta o pseudocódigo para o método de seleção proposto. Como entrada é fornecido um conjunto de ranqueadores \mathfrak{R} e o tamanho t da combinação a ser selecionada. A abordagem não oferece uma estratégia para definir automaticamente o tamanho da combinação a ser selecionada, o que é um possível trabalho futuro.

Todo o processo de seleção tem como base a seleção de pares, a qual ocorre nas linhas (1) e (2). Enquanto em (1) o conjunto C_2 é inicializado com todos os pares disponíveis a partir do \mathfrak{R} de entrada, em (2) obtêm-se a lista de seleção dos pares. A função $ordenar(C_n, w, L_R)$ retorna uma lista que contém as primeiras L_R combinações de C_n em ordem decrescente a partir da função w que é calculada internamente à função $ordenar$. Entre (3) e (12) o pseudocódigo descreve o processo iterativo para obter os conjuntos C_n e τ_n^R para diferentes valores de n até atingir o valor de t . Pode-se enunciar

que a Equação 5.5 é equivalente ao processo descrito pelas linhas (1), para pares, e (4) à (10) para combinações de tamanho arbitrário. Por fim, (13) descreve a obtenção da combinação selecionada \mathfrak{X}_t^* referente à Equação 5.7.

Algoritmo 2 Seleção de Combinações de Ranqueadores

Entrada: Conjunto de ranqueadores \mathfrak{R} e o tamanho t da combinação a ser selecionada.

Saída: Combinação selecionada \mathfrak{X}_t^* .

```

1:  $C_2 \leftarrow \mathfrak{R}^2$ 
2:  $\tau_2^R \leftarrow \text{ordenar}(C_2, w, L_R)$ 
3: para  $n \leftarrow 3$  até  $t$  faça
4:    $C_n \leftarrow \emptyset$ 
5:   para todo  $\mathfrak{X}^i, \mathfrak{X}^j \in \tau_{n-1}^R$  faça
6:      $\mathfrak{X}^u \leftarrow \mathfrak{X}^i \cup \mathfrak{X}^j$ 
7:     se  $|\mathfrak{X}^u| = n$  então
8:        $C_n \leftarrow C_n \cup \{\mathfrak{X}^u\}$ 
9:     fim se
10:  fim para
11:   $\tau_n^R \leftarrow \text{ordenar}(C_n, w, L_R)$ 
12: fim para
13:  $\mathfrak{X}_t^* \leftarrow \tau_t^R(1)$ 
14: retornar  $\mathfrak{X}_t^*$ 

```

5.2 Avaliação Experimental

Primeiramente, a Seção 5.2.1 avalia os parâmetros do método proposto neste capítulo. Em seguida, a Seção 5.2.2 exhibe os resultados de seleção de pares e de conjuntos de ranqueadores em diferentes cenários.

5.2.1 Análise dos Parâmetros

Esta seção tem o intuito de mensurar o impacto dos principais parâmetros e métricas no desempenho do método proposto. Sendo assim, é possível definir os parâmetros padrões para os demais experimentos. Todos os resultados apresentados nesta seção consideram apenas a seleção de pares. Além disso, para fins comparativos, calcula-se a média aritmética ponderada do MAP dos cinco primeiros pares selecionados pelo USRF (peso cinco para o primeiro colocado, quatro para o segundo colocado e assim sucessivamente).

- **Tamanho da Vizinhança**

O tamanho da vizinhança, denotado por k , é utilizado em três diferentes etapas do USRF: no cálculo da métrica de estimativa de eficácia; no cálculo da métrica de correlação; e pelo método de combinação/fusão, no caso o CPRR [20]. Apesar de ser possível o uso de diferentes valores de k para cada uma das etapas, a avaliação foi realizada considerando um k unificado.

Uma avaliação conjunta dos parâmetros k e β foi conduzida para cada uma das coleções de imagens considerando os cenários personalizados. Os resultados são apresentados na Figura 12. Para a realização deste experimento, foi utilizado *Authority Score* como estimativa de eficácia e RBO como métrica de correlação, uma vez que estas apresentam resultados superiores ou comparáveis às demais em [142] e [146], respectivamente.

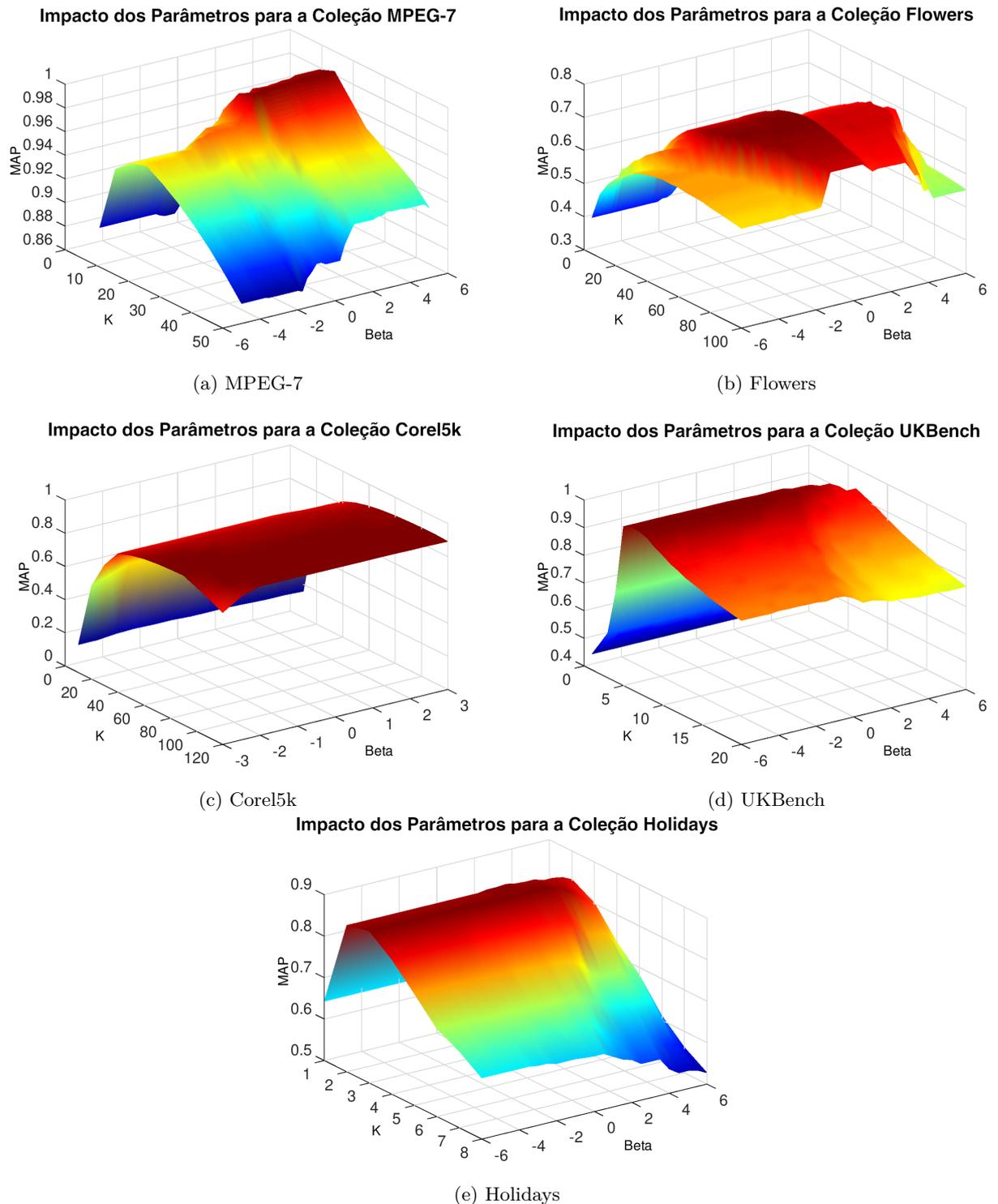


Figura 12 – Impacto dos parâmetros k e β na média aritmética ponderada considerando o MAP dos cinco primeiros resultados selecionados para cada coleção.

A partir dos resultados obtidos, adotou-se 20, 50, 50, 5 e 3 como o valor de k padrão para as coleções MPEG-7, Flowers, Corel5k, UKBench e Holidays, respectivamente. O valor de β é avaliado e discutido em mais detalhes nos próximos experimentos, uma vez que este pode variar de acordo com o cenário de seleção. No entanto, pelo perfil das superfícies, pode-se perceber que $\beta = 1$ aparenta ser satisfatório.

- **Métricas para Seleção**

Como o método é flexível, diferentes métricas de estimativa de eficácia e de correlação podem ser empregadas. A Tabela 9 mostra os resultados de seleção para cada uma das coleções de imagens em seus cenários personalizados considerando diferentes combinações de métricas. Percebe-se que os resultados não variam muito, o que evidencia que a nossa estratégia é robusta e não é demasiadamente sensível às métricas escolhidas. De qualquer forma, para os demais experimentos, adotou-se o *Reciprocal Density* como estimativa de eficácia e a *RBO* como métrica de correlação, uma vez que estas atingiram os melhores resultados na maior parte dos casos.

Tabela 9 – Média ponderada do MAP para os cinco primeiros pares selecionados considerando diferentes combinações de métricas e $\beta = 1$.

Métricas		Média Ponderada do MAP (%)					
Estim.	Eficácia	Correlação	MPEG-7	Flowers	Corel5k	Ukbench	Holidays
<i>Authority</i>		Jaccard	98.95	70.15	85.93	96.32	80.14
		Jaccard _k	99.33	72.82	85.93	97.07	86.64
		RBO	99.33	73.39	85.93	96.97	86.53
		Spearman	99.09	70.15	85.60	97.02	86.64
		Kendall τ	99.33	73.37	85.93	97.07	86.64
<i>Reciprocal</i>		Jaccard	99.50	70.44	85.93	97.06	84.77
		Jaccard _k	99.62	70.44	85.93	97.15	86.64
		RBO	99.62	72.52	85.93	97.43	86.53
		Spearman	99.60	70.44	85.60	97.34	86.64
		Kendall τ	99.62	72.52	85.93	97.34	86.64

- **Relevância da Correlação**

Tendo em vista a escolha dos valores de k e das métricas, é necessário avaliar mais detalhadamente o valor de β . Anteriormente, foi possível verificar que $\beta = 1$ é satisfatório nos cenários personalizados. No entanto, no decorrer da análise experimental, percebeu-se que o β é relativamente sensível ao número de descritores disponíveis para seleção. Dessa forma, conduzimos um experimento que avalia o desempenho do algoritmo considerando cenários aleatórios de diferentes tamanhos, os resultados são apresentados na Figura 13 para duas coleções de imagens distintas. Cada um dos pontos exibidos no gráfico corresponde à média do valor de 20 execuções. Novamente, para cada execução, considera-se a média ponderada do MAP dos cinco primeiros pares selecionados.

A partir dos resultados obtidos é possível perceber que, de maneira geral, para cenários com poucos descritores (em torno de 6 ou menos) $\beta = 1$ é mais satisfatório. No entanto, conforme o número aumenta, $\beta = -1$ revela resultados superiores. Isso se deve provavelmente ao fato de que, com um pequeno conjunto de descritores, o USRF é capaz de utilizar a correlação como uma forma de explorar a complementaridade dos dados. Enquanto que, quando o número de descritores é muito elevado, a correlação se torna uma forma de identificar os elementos mais similares, uma estratégia que evita os *outliers*.

Portanto, para os experimentos restantes, adotou-se $\beta = 1$ nos cenários personalizados, os quais são compostos por seis descritores, e $\beta = -1$ nos demais.

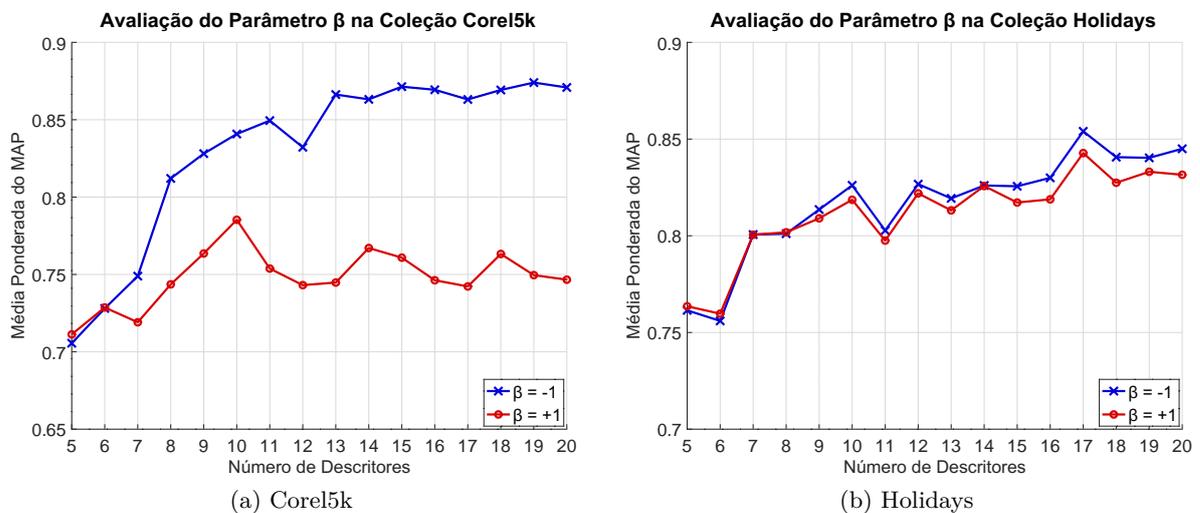


Figura 13 – Avaliação do parâmetro β considerando cenários aleatórios com diferentes números de descritores.

• Tamanho das Listas de Seleção

As listas de seleção τ_n^R armazenam as combinações ordenadas de acordo com o valor da métrica de seleção w . Conforme o tamanho das combinações (n) aumenta, o número de combinações possíveis cresce exponencialmente. Portanto, torna-se fundamental não considerar as listas de seleção por completo, mas apenas as L_R primeiras posições.

Um experimento foi realizado a fim de avaliar o impacto do parâmetro L_R nos resultados do USRF. Nesta análise considerou-se a média ponderada do MAP das combinações ranqueadas nas cinco primeiras posições de τ_n^R para n (tamanho da combinação) no intervalo $[3, 5]$. Em todas as coleções, foram utilizados todos os descritores disponíveis. Os resultados são exibidos pela Figura 14. Para uma melhor visualização dos dados, os gráficos são exibidos separadamente para cada uma das coleções devido a escala dos valores. Pode-se observar que há pouca variação de eficácia conforme L_R aumenta. Dessa forma, constata-se que o valor escolhido não pode ser muito pequeno ($L_R < 30$) para que a seleção não seja comprometida. Adotou-se $L_R = 100$ para os experimentos restantes, uma vez que altos tempos de execução só ocorrem com valores muito superiores a esse.

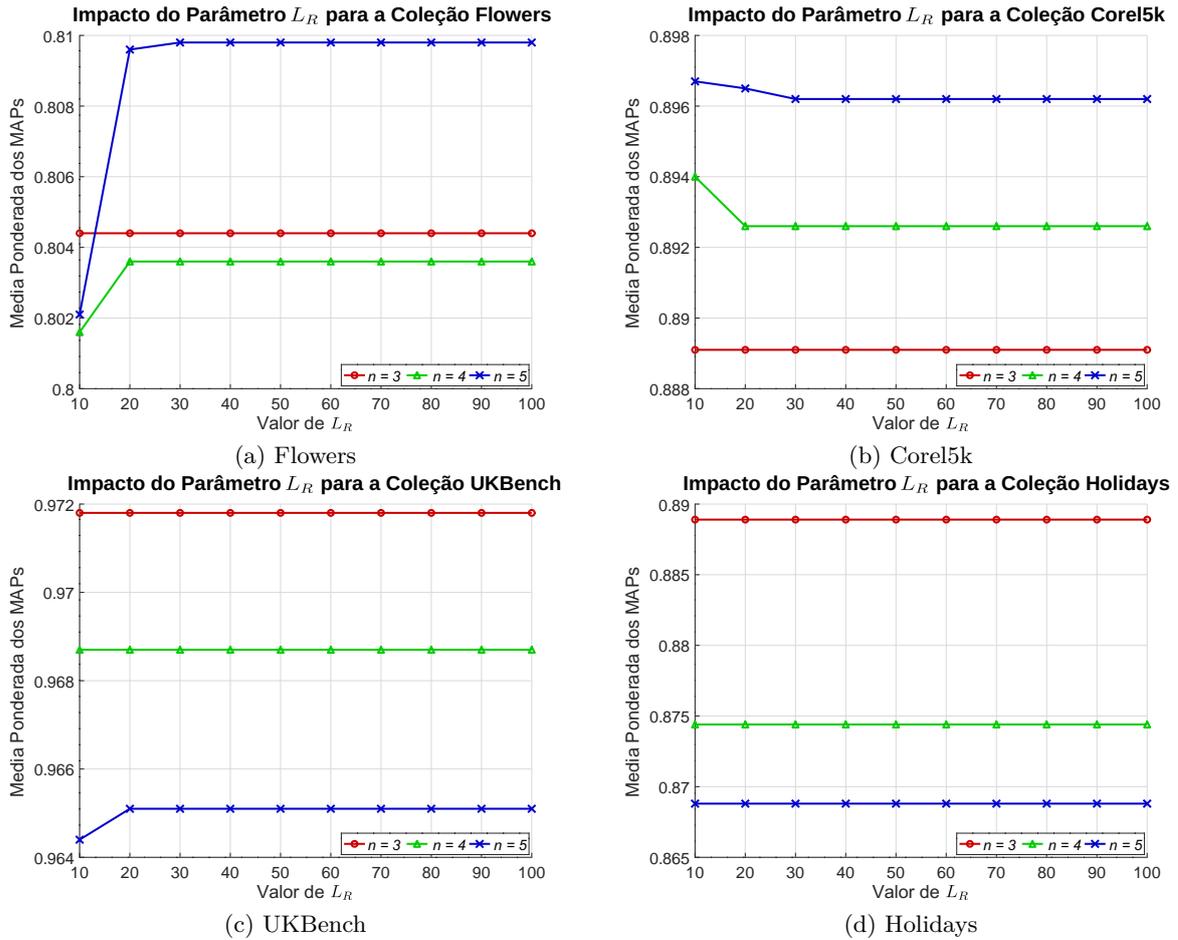


Figura 14 – Avaliação do parâmetro L_R nos resultados de seleção.

5.2.2 Resultados de Seleção

Os resultados do USRF são apresentados em duas partes. Enquanto a Seção 5.2.2.1 mostra as análises considerando apenas a seleção de pares, a Seção 5.2.2.2 exibe os experimentos para conjuntos de ranqueadores de qualquer cardinalidade. Em ambos os casos, diferentes cenários de seleção são apontados.

5.2.2.1 Seleção de Pares

Uma vez que a seleção de combinações de tamanhos arbitrários é baseada na seleção por pares, esta seção tem o objetivo de avaliar primeiramente esse tipo de seleção, considerando apenas pares.

Inicialmente são apresentados os experimentos realizados nos cenários personalizados. A Figura 15 apresenta uma visualização da seleção de pares para duas coleções de imagens. Cada um dos pontos representa um par de ranqueadores que é posicionado em relação ao valor fornecido pela métrica de seleção do USRF e do MAP obtido através da combinação/fusão do par. A linha pontilhada indica o resultado do melhor descritor isolado. O par selecionado como a melhor combinação está destacado em verde. Nota-se

que, em ambos os casos, o par selecionado como candidato a melhor combinação possui também o melhor MAP, o que evidencia a alta eficácia da seleção.

Além disso, espera-se que, quanto maior o valor fornecido pela métrica de seleção, maior o MAP do resultado. A correlação de Pearson da distribuição é de 0.88 e 0.78 para os gráficos (a) e (b), respectivamente. Tais valores indicam uma forte correlação linear crescente, os quais revelam a acurácia do método proposto.

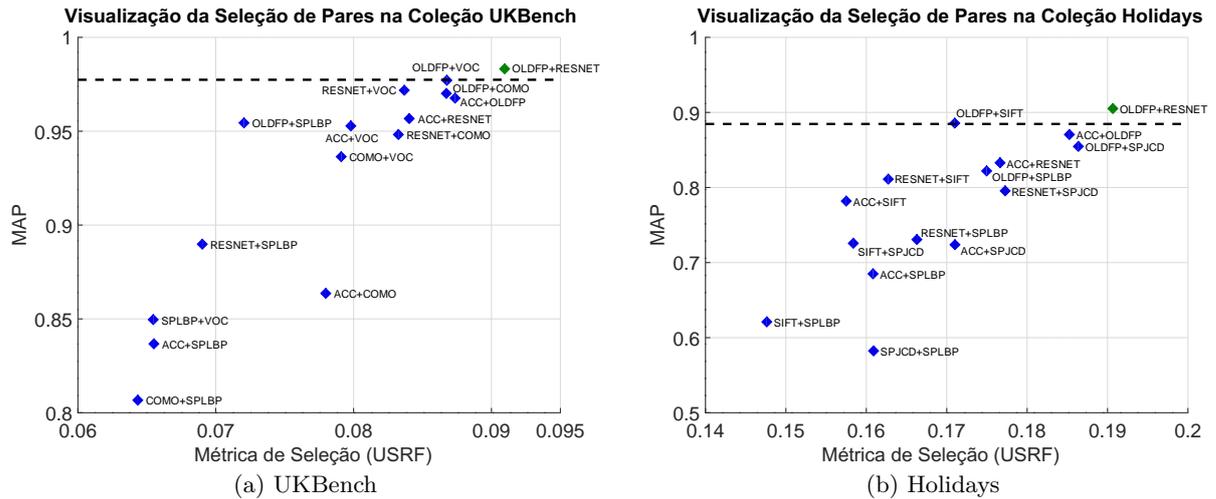


Figura 15 – Distribuição dos pares considerando a métrica de seleção comparada ao MAP nos cenários personalizados.

Nos próximos experimentos, para propósito comparativo, são considerados três referenciais hipotéticos, que atuam como *baselines* fictícios. A ideia é visualizar como a estratégia proposta se situa em relação a cada uma das modelagens a seguir:

- **Melhor Caso:** o caso ideal de seleção é aquele em que sempre é possível selecionar a melhor combinação dentre as combinações possíveis, em que são avaliados todos os pares. Considera-se, dessa forma, uma lista dos ranqueadores baseados no MAP.
- **Caso Médio:** a partir das combinações disponíveis, seleciona-se a intermediária, ou seja, aquela que se encontra na mediana caso todas as combinações estivessem ordenadas pela eficácia (MAP).
- **Pior Caso:** o pior caso de seleção é aquele em que sempre é selecionada a pior combinação dentre as combinações possíveis, em que são avaliados todos os pares.

Com o objetivo de facilitar a visualização dos resultados, foram construídos gráficos de linhas: são apresentadas três linhas tracejadas considerando os referenciais hipotéticos (melhor caso, caso médio e pior caso), uma linha tracejada considerando o melhor descritor isolado e, por último, a linha do USRF. No eixo das abcissas tem-se o número de pares selecionados e no eixo vertical a média aritmética dos MAPs dos pares selecionados. Essa representação avalia a abordagem proposta se situa em relação a cada caso.

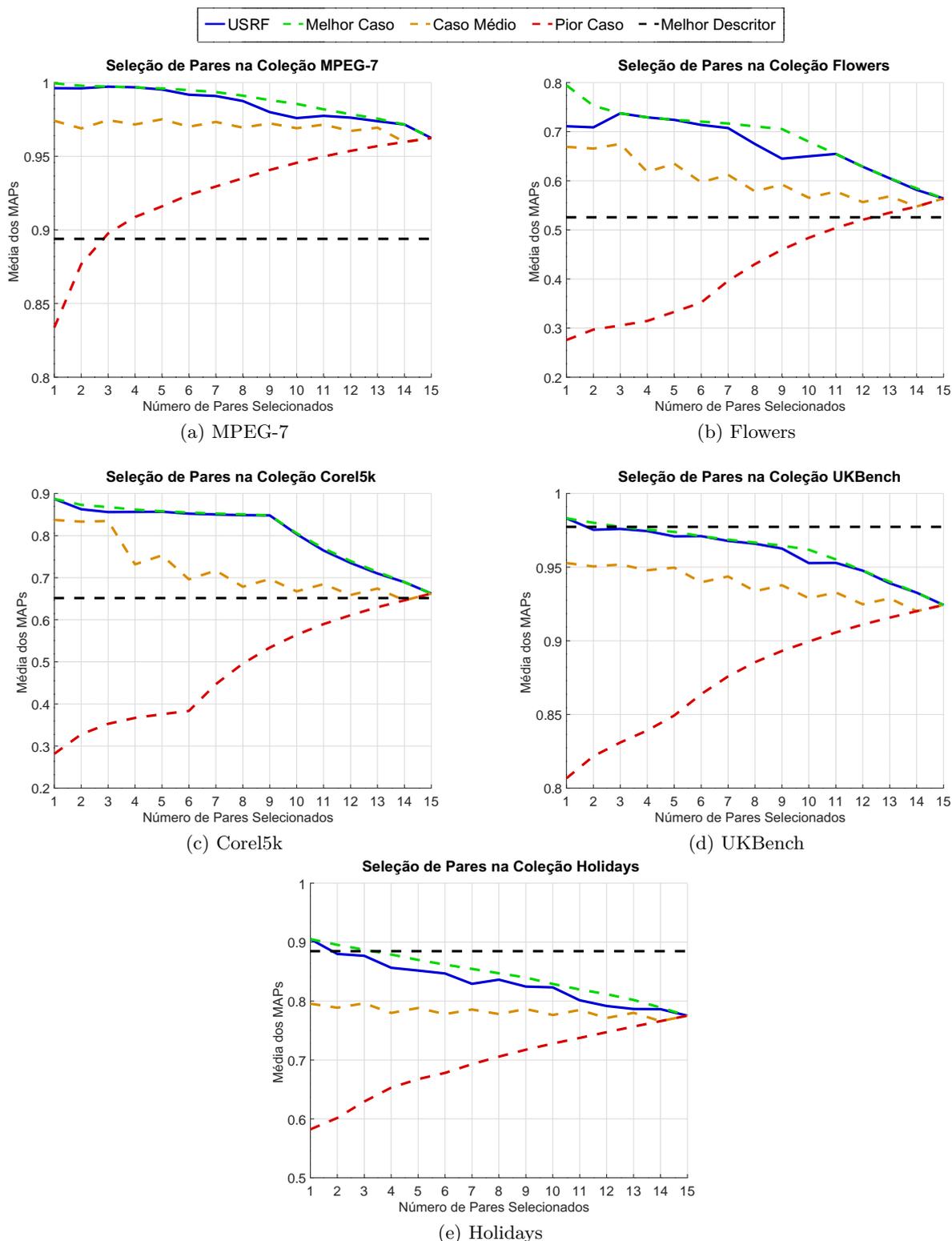


Figura 16 – Avaliação da seleção proposta (USRF) para pares de ranqueadores nos cenários personalizados.

A Figura 16 exibe os resultados de seleção nos cenários personalizados para todas as coleções de imagens. Nota-se que o USRF atingiu valores comparáveis ao melhor caso em todas as coleções. Observe que utilizar o melhor caso como referência é um critério

extremamente rigoroso, uma vez que este sempre se baseia na eficácia (MAP) obtida a partir dos dados rotulados, enquanto o USRF é totalmente não supervisionado. Dessa forma, ressalta-se que os resultados estarem nesse patamar é realmente muito significativo.

De maneira análoga, o experimento anterior foi realizado considerando todos os ranqueadores, ou seja, os cenários completos. Os resultados são apresentados pela Figura 17 para cada uma das coleções de imagens, com exceção da MPEG-7 na qual o cenário personalizado é igual ao completo (são apenas seis descritores). Em geral, os cenários que utilizam todos os descritores são mais desafiadores, uma vez que a variação entre os resultados dos ranqueadores pode vir a ser muito maior. Novamente, os resultados de seleção se situam muito próximos ao melhor caso na maioria das coleções. Observa-se que, para as coleções UKBench e Holidays, são poucas as combinações que se situam acima do resultado do melhor descritor e, apesar disso, nosso método foi capaz de selecionar os melhores pares.

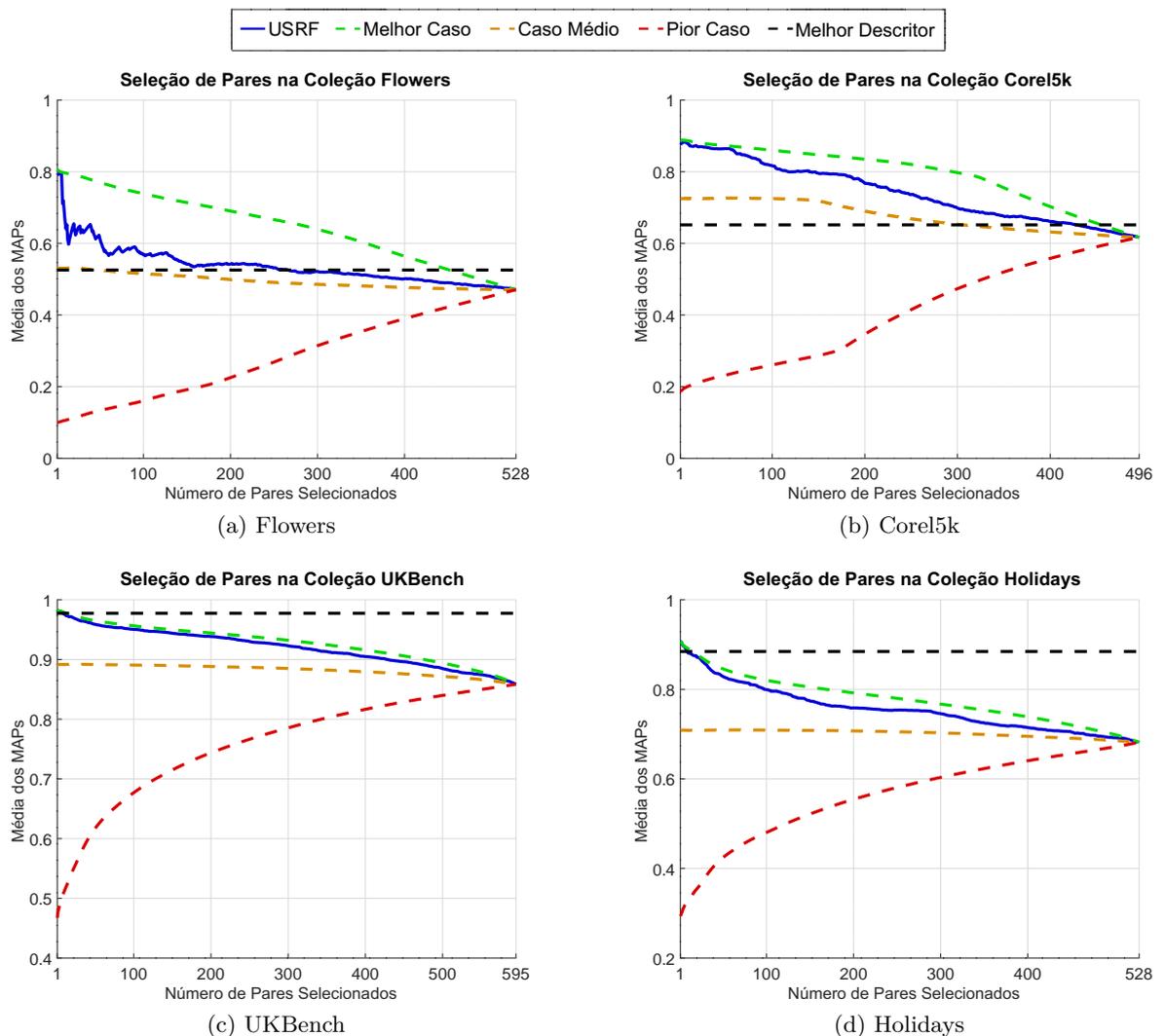


Figura 17 – Avaliação da seleção proposta (USRF) para pares de ranqueadores nos cenários completos.

Outros contextos de seleção mais específicos também foram analisados. A Figura 18 apresenta a avaliação para as coleções Corel5k e UKBench em dois cenários: utilizando apenas descritores globais e locais; e somente em descritores de *deep learning*. Os resultados de seleção do gráfico (a) se situam próximo ao caso médio, provavelmente devido à maior variabilidade entre o MAP dos descritores. Apesar disso, combinações superiores ao melhor descritor isolado ainda foram selecionadas dentre as primeiras posições.

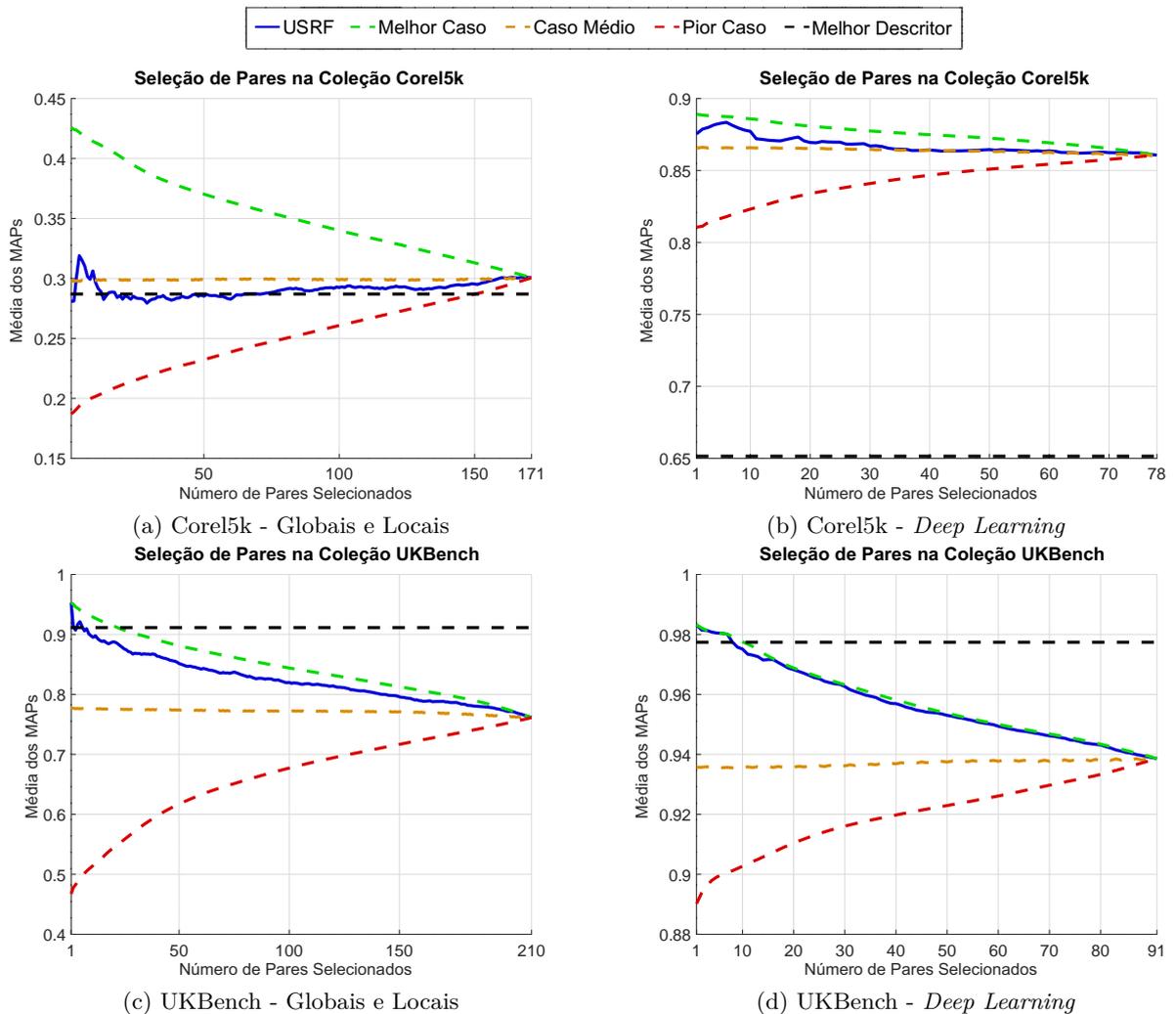


Figura 18 – Avaliação da seleção proposta (USRF) para pares de ranqueadores em diferentes cenários.

5.2.2.2 Seleção de Conjuntos de Ranqueadores

A Tabela 10 apresenta o MAP das combinações ranqueadas nas três primeiras posições pelo USRF (os três primeiros elementos de τ_n^R) considerando diferentes tamanhos (n no intervalo $[2, 10]$). Note que a primeira posição de τ_n^R é também o \mathfrak{X}_n^* . Os valores são reportados para todas as coleções nos cenários personalizados e completos. Para cada coleção e cenário, alguns resultados foram destacados: a melhor combinação entre as três primeiras colocações está em **negrito e sublinhada**; a melhor combinação entre as primeiras

colocadas (\mathfrak{X}^*) está em negrito e é utilizada no restante do texto para comparação com os demais métodos.

Tabela 10 – MAP das três primeiras combinações selecionadas considerando diferentes tamanhos nos cenários personalizados e completos.

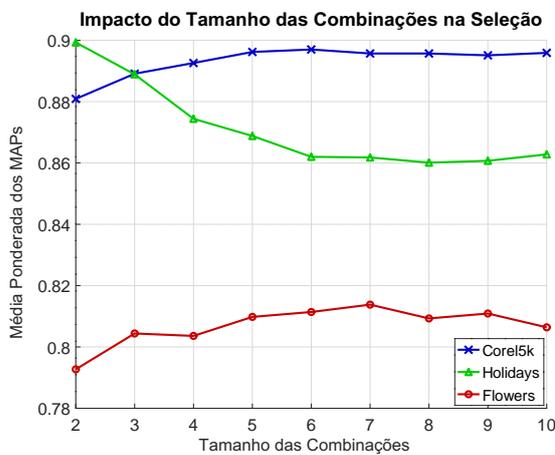
		Negrito: \mathfrak{X}^* selecionado Negrito e sublinhado: Melhor \mathfrak{X}									
Cenário	Coleção	MAP (%)									
		\mathfrak{X}_2	\mathfrak{X}_3	\mathfrak{X}_4	\mathfrak{X}_5	\mathfrak{X}_6	\mathfrak{X}_7	\mathfrak{X}_8	\mathfrak{X}_9	\mathfrak{X}_{10}	
Personalizado	MPEG-7	\mathfrak{X}^*	99.62	98.25	99.57	99.92	99.89	—	—	—	—
		2°	99.60	100.0	99.38	99.74	—	—	—	—	—
		3°	99.95	99.99	99.91	99.58	—	—	—	—	—
	Flowers	\mathfrak{X}^*	71.11	78.67	79.10	74.06	74.04	—	—	—	—
		2°	70.67	72.51	76.71	78.53	—	—	—	—	—
		3°	79.45	68.33	77.54	76.00	—	—	—	—	—
	Corel5k	\mathfrak{X}^*	88.73	90.32	88.99	89.25	89.55	—	—	—	—
		2°	83.80	90.82	90.31	89.34	—	—	—	—	—
		3°	84.26	88.44	90.80	90.66	—	—	—	—	—
	UKBench	\mathfrak{X}^*	98.32	98.40	99.02	98.81	98.74	—	—	—	—
		2°	96.76	98.84	97.89	98.91	—	—	—	—	—
		3°	97.71	98.45	98.90	97.83	—	—	—	—	—
	Holidays	\mathfrak{X}^*	90.51	88.83	88.91	88.23	89.24	—	—	—	—
		2°	85.47	90.49	87.07	89.73	—	—	—	—	—
		3°	87.05	84.33	89.34	88.78	—	—	—	—	—
Completo	Flowers	\mathfrak{X}^*	79.03	80.19	81.14	81.44	80.47	81.13	81.71	80.83	80.45
		2°	79.45	80.47	80.74	79.89	81.83	81.28	80.27	81.39	80.45
		3°	79.51	80.79	77.88	81.47	80.77	82.28	80.52	81.21	80.77
	Corel5k	\mathfrak{X}^*	87.54	88.67	89.64	89.67	89.37	89.88	89.60	89.49	89.49
		2°	88.21	89.22	89.01	89.51	90.27	89.06	89.77	89.54	89.59
		3°	88.24	88.24	89.39	90.41	89.33	89.28	89.03	89.73	89.74
	UKBench	\mathfrak{X}^*	98.32	96.88	96.16	96.79	96.67	96.75	96.76	97.82	97.60
		2°	97.97	97.19	97.41	96.12	96.90	96.61	97.97	96.54	98.33
		3°	98.11	96.82	97.46	96.48	96.70	96.83	97.84	97.70	97.37
	Holidays	\mathfrak{X}^*	90.51	89.56	87.35	86.30	86.16	85.41	86.69	85.87	85.60
		2°	90.72	88.42	87.23	87.32	85.95	87.54	84.79	86.66	87.04
		3°	88.75	88.72	86.97	86.36	87.35	85.34	85.25	84.84	85.63

Os ranqueadores que compõem as combinações selecionadas para cada coleção são exibidos na Tabela 11. O ganho relativo é calculado em relação ao resultado do melhor descritor isolado em cada um dos casos. Nota-se que o USRF atingiu ganhos positivos em todas as circunstâncias apresentadas.

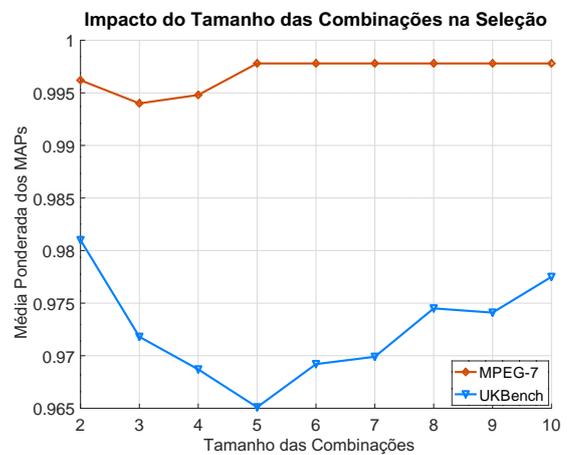
Um experimento também foi conduzido com o propósito de analisar o impacto do tamanho das combinações nos resultados do USRF. Para cada uma das coleções, foi calculada a média aritmética ponderada das cinco primeiras combinações selecionadas. Os resultados são contemplados na Figura 19. A fim de facilitar a visualização dos dados, os gráficos foram divididos devido a diferença na escala dos valores. Nota-se que não é possível estabelecer uma cardinalidade fixa para \mathfrak{X}^* , uma vez que o tamanho que fornece o melhor valor difere consideravelmente entre as coleções. Enquanto nas coleções Holidays e UKBench os melhores valores são encontrados para combinações de tamanho dois, as demais coleções necessitam de tamanhos maiores. Dessa forma, a necessidade de escolher manualmente o melhor tamanho está entre uma das limitações desse método.

Tabela 11 – Especificação das combinações selecionadas para cada uma das coleções nos cenários personalizados e completos.

Cenário	Coleção	Combinação Selecionada	MAP (%)	Ganho Relativo
Personalizado	MPEG-7	AIR + ASC + BAS + CFD + IDSC	99.92	+11.78%
	Flowers	BIC + FBResNet + ResNeXt + SIFT	79.10	+50.50%
	Corel5k	DPNet + ResNet + SPACC	90.32	+38.63%
	UKBench	ACC + OLDFP + ResNet + VOC	99.02	+1.31%
	Holidays	OLDFP + ResNet	90.51	+2.32%
Completo	Flowers	AlexNet + BnInception + DPNet + FBResNet + ResNeXt + ResNet + Xception + LBP	81.71	+55.46%
	Corel5k	DPNet + FBResNet + InceptionResNet + InceptionV4 + ResNeXt + ResNet + SENet	89.88	+37.96%
	UKBench	OLDFP + ResNet	98.32	+0.59%
	Holidays	OLDFP + ResNet	90.51	+2.32%



(a) Corel5k, Holidays, Flowers



(b) MPEG-7, UKBench

Figura 19 – Análise do tamanho das combinações em relação aos resultados de seleção para cada uma das coleções considerando todos os ranqueadores (cenários completos).

6 Seleção por Grafos e Componentes Conexas

Este capítulo apresenta o método *Ranker Graph for Selection and Fusion* (RGSF), o qual se baseia na utilização de grafos e componentes conexas. Diferentemente do método anterior, o RGSF trata combinações de tamanhos diferentes de maneira conjunta, sem realizar distinção de tamanho. Tendo como entrada um conjunto de ranqueadores, o algoritmo executa uma etapa inicial de seleção através de um grafo, em que se tem como objetivo encontrar as componentes fortemente conexas. Cada componente corresponde a uma combinação diferente que é submetida às etapas de agregação e seleção posteriores.

A Figura 20 ilustra as principais etapas que compõem o método. Em (1), um grafo não direcionado é construído a partir de métricas que não requerem de dados rotulados ou treinamento. Cada vértice corresponde a um ranqueador, o qual está associado a um valor de estimativa de eficácia, e as arestas correspondem à correlação entre eles. São considerados limiares (*thresholds*) tanto para a estimativa de eficácia quanto para a correlação, os quais variam ao longo das iterações. Dessa forma, o processo é realizado iterativamente com a finalidade de detectar componentes conexas, as quais correspondem às combinações que são submetidas ao método de fusão (agregação). A notação \mathfrak{X}_{cc_i} é utilizada para se referir a combinação da componente conexa i , em que i é apenas um índice e não possui relação com o tamanho da combinação. Essa etapa é descrita em mais detalhes na Seção 6.1.

A agregação de cada uma das combinações selecionadas em (1) é realizada em

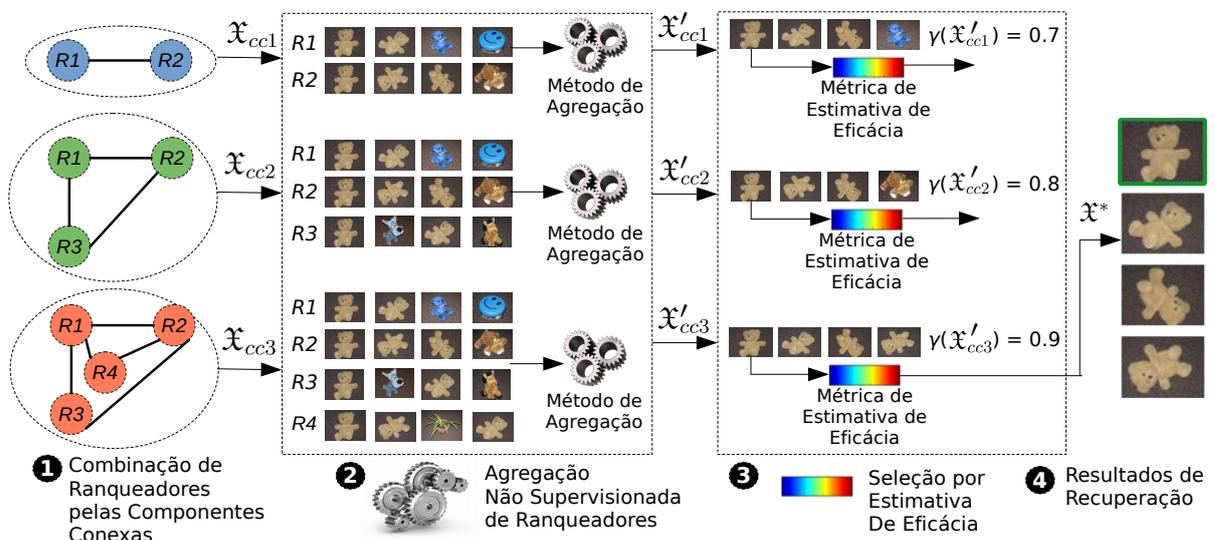


Figura 20 – Principais etapas que compõem o método RGSF.

(2) por um método de aprendizado não supervisionado, gerando diferentes resultados de recuperação. Para evitar ambiguidades, a notação \mathfrak{X}' foi adotada apenas neste capítulo com o propósito de diferenciar as combinações antes e após a execução da agregação. Como relatado no protocolo experimental (Capítulo 4), o método *Cartesian Product of Ranking References* (CPRR) [20] foi utilizado para as fusões. Ressalta-se, no entanto, que o RGSF é flexível e permite a utilização de outros métodos de agregação de ranqueadores.

Em (3), a métrica de estimativa de eficácia é executada para cada um dos resultados de agregação obtidos anteriormente. O objetivo desta etapa consiste em fazer uma seleção final, selecionando a combinação que apresentou o maior valor da estimativa de eficácia como o resultado de recuperação de saída (4). Formalmente, seja $\gamma(\mathfrak{X}')$ uma função que calcula a estimativa de eficácia das listas ranqueadas fornecidas pela combinação \mathfrak{X}' e S' a lista que contém todas as \mathfrak{X}' calculadas pelo método agregador. Dessa forma, a obtenção de \mathfrak{X}^* é definida pela Equação 6.1.

$$\mathfrak{X}^* = \arg \max_{\mathfrak{X}'_{cc_i} \in S'} \gamma(\mathfrak{X}'_{cc_i}). \quad (6.1)$$

6.1 Combinação de Ranqueadores pelas Componentes Conexas

Esta seção é dedicada à etapa inicial do RGSF, uma vez que se trata do principal e mais importante estágio de seleção do método. Devido ao grande número de combinações possíveis, podendo ser até mesmo bilhões, uma seleção eficaz que ocorra previamente a etapa de agregação é fundamental, uma vez que realizar a fusão de todas as combinações existentes seria completamente impraticável. Dessa forma, tal etapa possui como objetivo fornecer uma lista de combinações selecionadas para serem submetidas à etapa de agregação e, posteriormente, à seleção pela estimativa de eficácia a fim de obter a combinação final (de saída), denotada por \mathfrak{X}^* .

Para essa tarefa, um grafo de ranqueadores é proposto a fim de representar cada um dos conjuntos de listas ranqueadas e as relações de complementaridade entre elas. Uma métrica de estimativa de eficácia é associada a cada um dos vértices (ranqueadores), enquanto a métrica de correlação é associada à cada uma das arestas. O propósito é de selecionar um conjunto de nós conectados (combinação de ranqueadores) com altos valores de estimativa de eficácia. Tal processo é realizado através do aumento gradual dos valores baseados em limiares (*thresholds*). Inicialmente, os limiares são inicializados em valores altos, de tal forma que apenas os ranqueadores de alta estimativa de eficácia são incluídos no grafo. Para cada um dos nós, o limiar de correlação é variado a cada iteração. Para cada atualização do grafo baseado nos limiares, as componentes conexas são identificadas como combinações selecionadas.

Formalmente, o procedimento é realizado por um grafo não direcionado $G = (V, E)$, em que o conjunto de vértices V é definido por um conjunto de ranqueadores \mathcal{R} , tal que

cada um dos ranqueadores é representado por um nó e $V = \mathcal{R}$. Os vértices disponíveis em cada uma das iterações do algoritmo varia de acordo com o limiar th_γ , tal que V é definido a seguir:

$$V = \{R_i \mid R_i \in \mathcal{R} \wedge \gamma(R_i) \geq th_\gamma\} \tag{6.2}$$

O conjunto de arestas E é definido considerando a correlação entre os ranqueadores:

$$E = \{(R_i, R_j) \mid R_i, R_j \in V \wedge \lambda(R_i, R_j) \geq th_\lambda\}, \tag{6.3}$$

em que th_λ define o limiar para a métrica de correlação.

A Figura 21 ilustra as iterações da combinação de ranqueadores pelas componentes conexas, na qual os eixos x e y representam a variação dos limiares de estimativa de eficácia (th_γ) e de correlação (th_λ), respectivamente. Inicialmente, ambos os limiares possuem o valor 1 (canto superior direito). Em seguida, o limiar de correlação é decrementado em intervalos de 0.25 até atingir o valor 0. Ao atingir este valor, o limiar de estimativa de eficácia é decrementado em 0.25 e o limiar de correlação é reinicializado em 1. As iterações se repetem, conforme a sequência indicada pelas setas, até que ambos os limiares atinjam o valor 0 (canto inferior esquerdo).

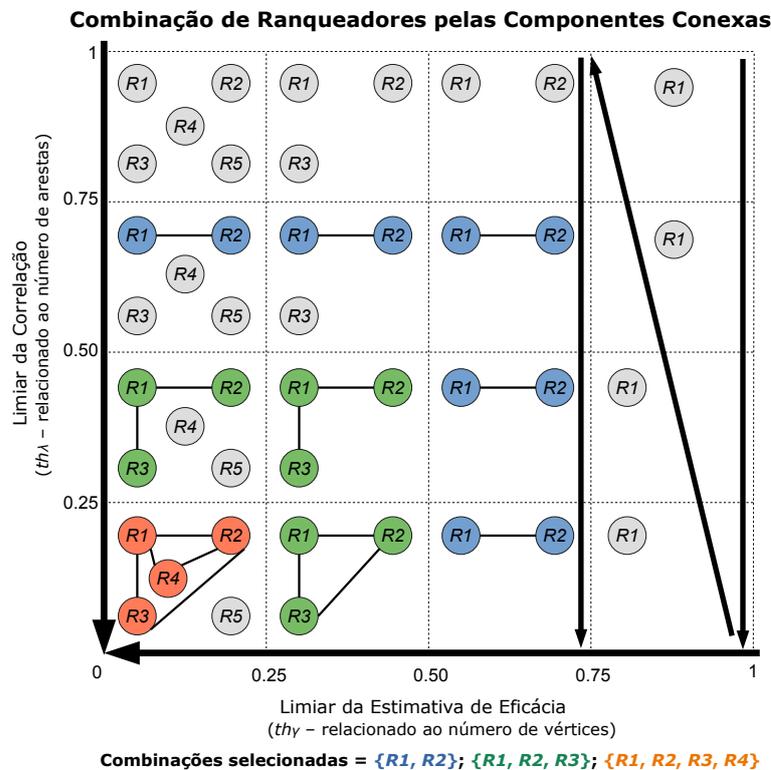


Figura 21 – Ilustração das iterações realizadas para a seleção dos ranqueadores pelas componentes conexas.

Pela ilustração, observa-se que o valor do limiar da estimativa de eficácia (th_γ) é inversamente proporcional ao número de nós presentes no grafo e vice-versa, ou seja,

quanto menor for th_γ , maior tende a ser o número de ranqueadores incluídos no grafo. Enquanto isso, o limiar de correlação (th_λ) ao ser decrementado, provoca o aumento no número de arestas. Cada uma das componentes conexas representam combinações diferentes. No exemplo ilustrado, há três diferentes combinações que foram selecionadas, as quais são identificadas por diferentes cores.

O Algoritmo 3 apresenta os procedimentos correspondentes a combinação de ranqueadores pelas componentes conexas. É fornecido como entrada um conjunto de ranqueadores com a finalidade de se obter uma lista de combinações S . As linhas de (1) a (3) correspondem a inicialização das principais variáveis, em que $th_{\gamma end}$ e $th_{\lambda end}$ representam os valores mínimos atingíveis pelos limiares de eficácia e correlação após os decrementos, respectivamente. O $th_{\lambda end}$ não atinge o valor 0 porque, caso atingisse, o método calcularia um grafo completo o que poderia ser muito custoso. As linhas de (6) a (10) são responsáveis pela criação dos vértices do grafo, sendo adicionados apenas os que possuem estimativa de eficácia acima do limiar da iteração (th_λ). A criação das arestas

Algoritmo 3 Combinação de Ranqueadores pelas Componentes Conexas

Entrada: Conjunto de ranqueadores \mathfrak{R} .

Saída: Lista S com as combinações selecionadas pelo grafo.

```

1:  $S \leftarrow listaVazia()$ ,  $maxComb = 10$ 
2:  $V \leftarrow \{\}$ ,  $E \leftarrow \{\}$ 
3:  $th_{\gamma end} \leftarrow 0$ ,  $th_{\lambda end} \leftarrow 0.2$ ,  $th_{dec} \leftarrow 0.01$ 
4:  $th_\gamma \leftarrow 1$ 
5: enquanto  $th_\gamma \geq th_{\gamma end}$  faça
6:   para todo  $R_i \in \mathcal{R}$  faça
7:     se  $\gamma(R_i) \geq th_\gamma$  então
8:        $V \leftarrow V \cup \{R_i\}$ 
9:     fim se
10:  fim para
11:   $th_\lambda \leftarrow 1$ 
12:  enquanto  $th_\lambda \geq th_{\lambda end}$  faça
13:    para todo  $R_i \in \mathcal{R}$  faça
14:      para todo  $R_j \in \mathcal{R}$  faça
15:        se  $\lambda(R_i, R_j) \geq th_\lambda$  então
16:           $E \leftarrow E \cup \{R_i, R_j\}$ 
17:        fim se
18:      fim para
19:    fim para
20:     $S.inserir(calcularNovasCC(S, V, E))$ 
21:    se  $S.tamanho() == maxComb$  então
22:      retornar  $S$ 
23:    fim se
24:     $th_\lambda \leftarrow th_\lambda - th_{dec}$ 
25:  fim enquanto
26:   $th_\gamma \leftarrow th_\gamma - th_{dec}$ 
27: fim enquanto
28: retornar  $S$ 

```

ocorre das linhas (13) a (19). Em (20) é referenciado o procedimento para a identificação das componentes conexas, cujos ranqueadores que as compõem são adicionados à lista S .

Dependendo do número de ranqueadores e do cenário de seleção, S pode vir a ser muito grande. No entanto, sabe-se que as combinações obtidas para os limiares maiores (nas primeiras iterações) possuem maior potencial para serem candidatas à \mathcal{X}^* . Sendo assim, as iterações são executadas até as $maxComb$ primeiras combinações serem identificadas pelo método (linhas (21) a (23)). O valor $maxComb = 10$ foi escolhido por ter sido observado que a partir desse valor o método não apresentou variações significativas nos resultados.

6.2 Avaliação Experimental

Para todos os experimentos, o *Reciprocal Score* e RBO foram considerados como estimativa de eficácia e métrica de correlação, respectivamente, uma vez que estes apresentaram resultados comparáveis ou superiores às outras métricas em [142] e [144], respectivamente. Os intervalos de limiar para a estimativa de eficácia e métrica de correlação são $[1, 0]$ e $[1, 0.2]$, respectivamente; em que o primeiro número do intervalo representa o valor inicial do limiar e o segundo o valor mínimo que o limiar pode atingir. Os limiares são incrementados em 0.01 a cada iteração a fim de obter resultados precisos. Após o término das iterações, as primeiras $maxComb$ combinações selecionadas são submetidas para agregação. Como abordado anteriormente, considerou-se $maxComb = 10$ para todos os experimentos, uma vez que valores maiores não apresentaram melhorias nos resultados na maioria dos casos.

6.2.1 Análise dos Parâmetros

Experimentos foram conduzidos com o propósito de avaliar o valor do parâmetro k para as métricas de correlação e estimativa de eficácia quando comparadas ao valor de k

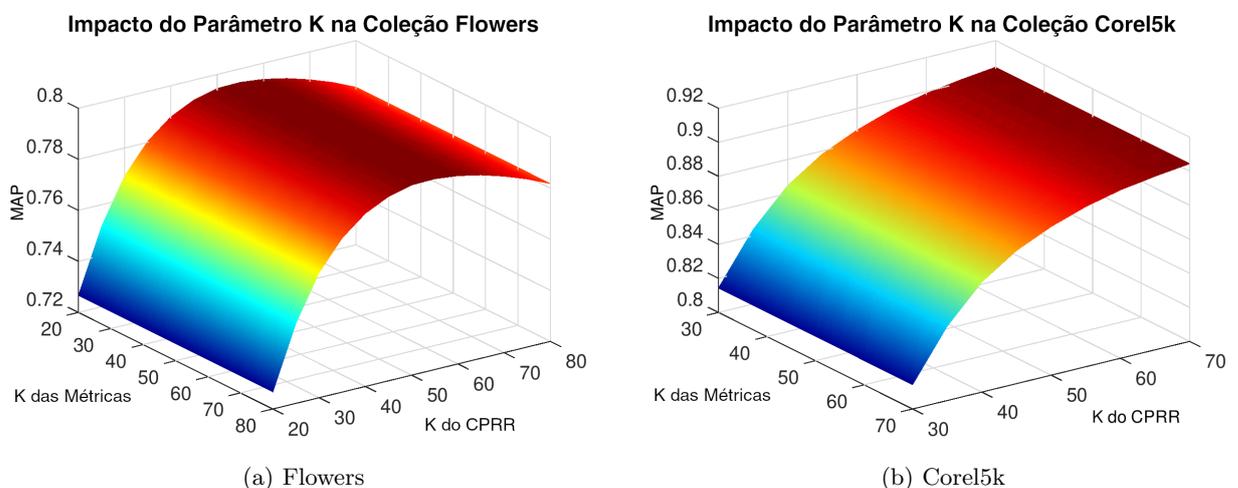


Figura 22 – Impacto do parâmetro k no MAP da combinação selecionada para as coleções com maior número de imagens por classe.

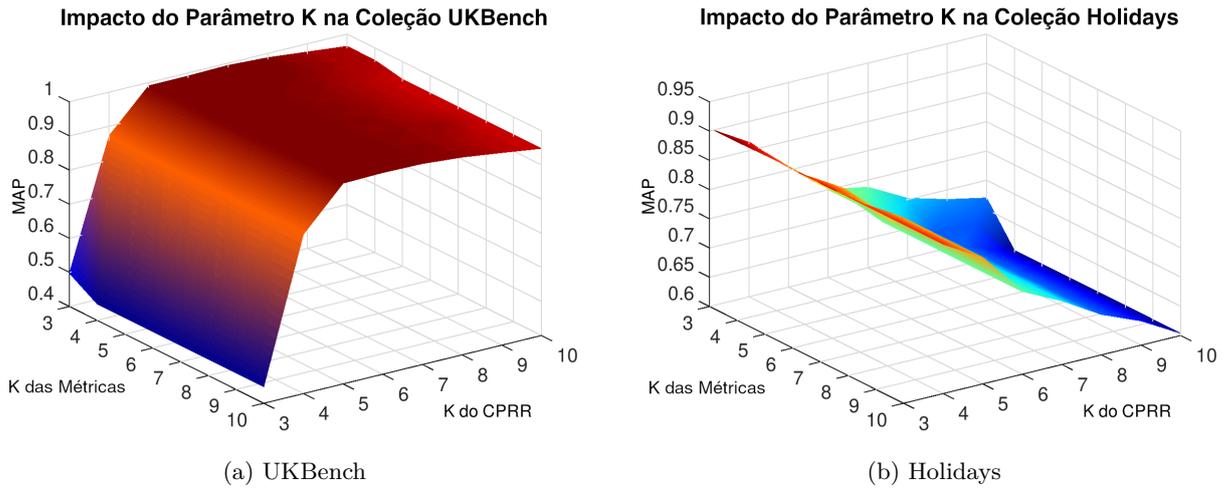


Figura 23 – Impacto do parâmetro k no MAP da combinação selecionada para as coleções com menor número de imagens por classe.

utilizado no método de agregação (CPRR). A Figura 22 apresenta as superfícies para as coleções com maior número de imagens por classe (Flowers e Corel5k) e a Figura 23 para as coleções com menor número de imagens por classe (UKBench e Holidays). As superfícies revelam que o método de agregação é muito sensível ao k do CPRR, mas não ao k das métricas. Dessa forma, optou-se por utilizar o mesmo k tanto para as métricas quanto para o agregador. Com base nos resultados obtidos, adotou-se os valores de k (20, 50, 50, 5, 3) para as coleções MPEG-7, Flowers, Corel5k, UKBench e Holidays, respectivamente.

6.2.2 Resultados de Seleção

Os resultados obtidos pelo RGSF são apresentados na Tabela 12, em que (\mathfrak{x}^*) indica a combinação selecionada. O ganho relativo é calculado em relação ao melhor descritor isolado para cada coleção. Verifica-se que o método proposto atingiu ganhos positivos na maior parte dos casos. Para a coleção UKBench, observa-se perda no cenário completo, isso se deve ao fato do método não ter sido capaz de selecionar uma combinação que incluísse o ranqueador OLDFP, o qual é o descritor de maior MAP.

De maneira análoga ao capítulo anterior, para propósitos comparativos, os experimentos a seguir consideram os mesmos três *baselines* hipotéticos. O objetivo é visualizar como o método proposto se compara em relação a cada um dos casos: (i) o **melhor caso** seleciona o par que possui o maior MAP dentre os pares disponíveis; (ii) o **caso médio** seleciona o par que possui o MAP que se encontra na mediana considerando todos os pares ordenados pelo MAP; (iii) o **pior caso** seleciona o par que possui o menor MAP dentre os pares disponíveis.

A Figura 24 contém os resultados de seleção para o cenário personalizado em quatro coleções. Para facilitar a visualização dos resultados, as seguintes linhas são utilizadas:

Tabela 12 – MAP da combinação selecionada para as coleções em diferentes cenários.

Cenário	Coleção	Combinação Selecionada (\mathcal{X}^*)	MAP (%)	Ganho Relativo
Personalizado	MPEG-7	AIR + ASC + BAS + CFD + IDSC	99.92	+11.78%
	Flowers	FBResNet + ResNeXt	79.45	+51.16%
	Corel5k	DPNet + ResNet	88.73	+36.19%
	UKBench	OLDFP + ResNet + VOC	98.84	+1.13%
	Holidays	OLDFP + ResNet	90.51	+2.32%
Completo	Flowers	BnInception + FBResNet + ResNeXt + ResNet	81.33	+54.73%
	Corel5k	FBResNet + ResNet	87.54	+34.37%
	UKBench	FBResNet + ResNet	95.93	-1.85%
	Holidays	OLDFP + ResNet + ResNeXt	89.56	+1.24%

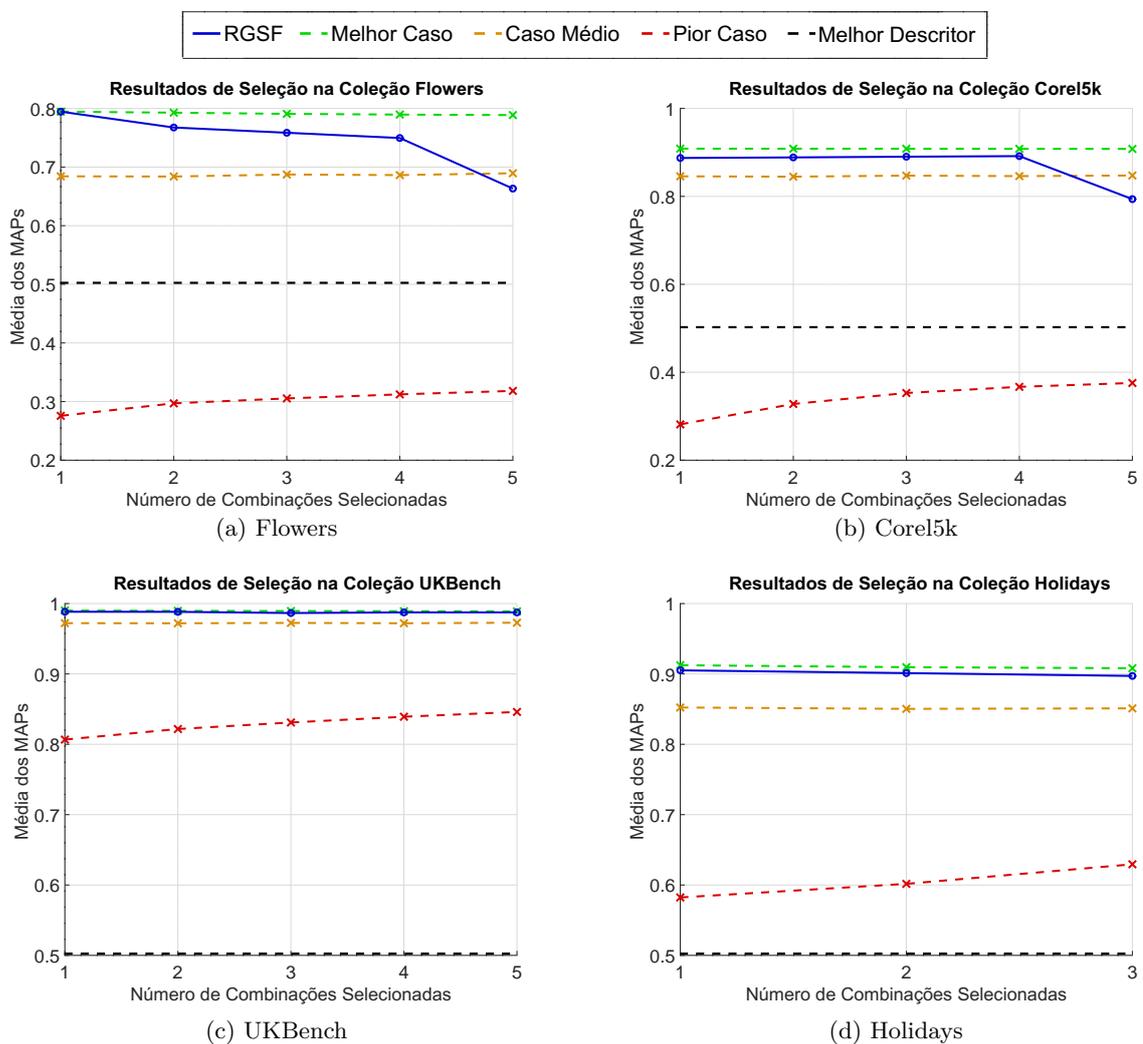


Figura 24 – Avaliação do método RGSF nos cenários personalizados.

três linhas considerando cada um dos casos hipotéticos (melhor, médio e pior caso), uma linha pontilhada considerando o melhor ranqueador (descritor) isolado e, por fim, um linha para o RGSF. O eixo horizontal corresponde ao número de pares selecionados e o eixo vertical indica a média aritmética do MAP para cada um dos pares selecionados. Esta representação avalia como o RGSF se compara aos demais em cada caso.

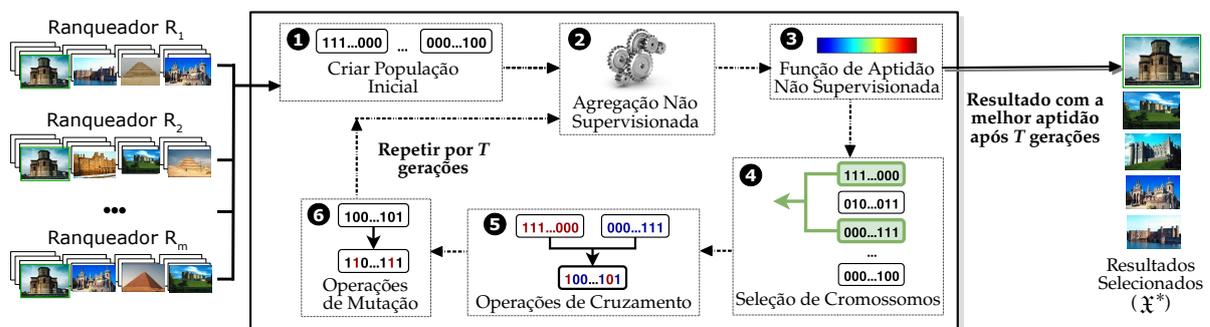
Note que, diferente dos *baselines* hipotéticos, os resultados do RGSF não consideram apenas pares, mas sim combinações de qualquer tamanho. É perceptível que o RGSF atingiu resultados comparáveis ao melhor caso em todas as coleções. Ressalta-se que o melhor caso é um critério de comparação muito rigoroso, uma vez que ele sempre se baseia no MAP enquanto o RGSF é completamente não supervisionado. Além disso, os resultados obtidos pelo RGSF são superiores ao melhor ranqueador isolado em todas as coleções avaliadas.

7 Seleção por Algoritmos Genéticos

Este capítulo apresenta o *Genetic Algorithm Framework for Rank Selection and Fusion* (UGAF-RSF). Algoritmos genéticos são uma classe particular de algoritmos evolutivos, os quais têm como propósito encontrar soluções aproximadas para problemas de otimização e busca. O processo evolutivo depende, em grande parte, de uma função de aptidão eficaz que direcione a evolução da população para a otimalidade. Este trabalho apresenta um *framework* que emprega um algoritmo genético que utiliza de uma função de aptidão completamente não supervisionada para seleção de ranqueadores. Assim como o RGSF, o UGAF-RSF também não segrega combinações de diferentes tamanhos ao longo de suas iterações. Diferente dos métodos desenvolvidos até então, o UGAF-RSF é estocástico, ou seja, os resultados variam para cada execução.

7.1 Descrição do Método

A Figura 25 ilustra as principais etapas que definem o *framework*. Um conjunto de listas ranqueadas são fornecidas como entrada do método. As combinações possíveis entre os m ranqueadores são modeladas como cromossomos binários, os quais são representados por uma sequência de genomas que são, por sua vez, representados por *bits* que podem assumir os valores 0 ou 1. A combinação definida por cada um dos cromossomos é calculada através de um método não supervisionado de *late fusion* baseado em agregação de listas ranqueadas. Os resultados de recuperação são avaliados por métricas de estimativa de eficácia, as quais são utilizadas para realizar a etapa de seleção. A partir dos indivíduos selecionados, a próxima geração é criada através de operações de cruzamento e mutação. Após T gerações, os resultados de agregação definidos pelos ranqueadores selecionados (\mathcal{X}^*) são tomados como saída. Cada um dos passos é detalhado nas subseções seguintes.



Unsupervised Genetic Algorithm Framework for Rank Selection and Fusion (UGAF-RSF)

Figura 25 – Visão geral do *framework* UGAF-RSF.

7.1.1 Representação dos Cromossomos

Cada indivíduo é descrito como um cromossomo, o qual é representado por uma *bitstring*. A representação do cromossomo é ilustrada por um exemplo na Figura 26. O tamanho da *string* é igual ao número de ranqueadores disponíveis (m). Para cada ranqueador, há um *bit* (genoma) que indica se ele faz parte da combinação ("1") ou não ("0"). O método requer que todo cromossomo possua no mínimo um *bit* em "1", ou seja, uma combinação precisa ser composta por no mínimo um ranqueador. Antes do início das iterações, uma população inicial é gerada aleatoriamente.



Figura 26 – Exemplo de cromossomo binário.

7.1.2 Agregação Não Supervisionada

Para cada iteração, a combinação definida por cada um dos cromossomos é calculada usando um método de agregação não supervisionado. O modelo de *framework* proposto é flexível e possibilita o uso de diferentes métodos de agregação. Assim como os outros métodos apresentados, utiliza-se o CPRR (*Cartesian Product of Ranking References*) [20] para realizar as agregações.

7.1.3 Função de Aptidão Não Supervisionada

O UGAF-RSF utiliza de estimativas de eficácia como funções de aptidão, o que permite que o algoritmo evolua a população de maneira completamente não supervisionada ao longo das gerações. Tais estimativas exploram a hipótese de agrupamento (*cluster hypothesis*) [141] que se baseia na ideia de que itens similares tendem a ser relevantes entre si se estes possuem elementos em comum nas primeiras k posições. Cada cromossomo é executado pelo método de agregação (CPRR) e é posteriormente submetido a estimativa de eficácia. A partir das métricas de eficácia descritas no protocolo experimental (Seção 4.3.1), este trabalho propõe e avalia a estratégia de uma métrica híbrida, que combina as métricas *Authority Score* e *Reciprocal Density* (Equação 7.1).

$$Hyb(q, k) = (Auth(q, k) + 1) \times (Rec(q, k) + 1) \quad (7.1)$$

7.1.4 Seleção de Cromossomos

O estágio de seleção filtra um conjunto de indivíduos para as próximas etapas, os quais são submetidos a operações de cruzamento e mutação com a finalidade de criar uma nova população para a próxima geração. Neste trabalho, a seleção dos indivíduos é realizada

pela seleção por torneio (*tournament selection*). Tal método sorteia aleatoriamente três diferentes cromossomos e seleciona aquele que possui a maior aptidão entre eles. O elitismo é aplicado em seguida, fazendo que uma certa porcentagem dos indivíduos com a maior aptidão de toda a população sejam obrigatoriamente mantidos na próxima iteração.

7.1.5 Operações de Cruzamento

A reprodução entre os cromossomos é um elemento crucial para manter a diversidade da população e, simultaneamente, aumentar a convergência e estabilidade do algoritmo. Existem diferentes tipos de operação de cruzamento disponíveis. Neste trabalho, o cruzamento uniforme é aplicado para gerar novos indivíduos. Nesta forma de cruzamento, cada genoma (*bit*) é independentemente escolhido dentre qualquer um de seus pais. Cada *bit* é escolhido de qualquer um dos pais com a mesma probabilidade.

7.1.6 Operações de Mutação

A mutação também é de significativa importância para manter a diversidade da população e pode prevenir a convergência do algoritmo para máximos locais. Para a mutação, cada genoma possui uma probabilidade de mutação que define a chance de cada *bit* ser invertido (por exemplo, se o *bit* está em "0", ele é alterado para "1" e vice-versa).

7.2 Avaliação Experimental

Esta seção apresenta e discute os resultados obtidos para o UGAF-RSF. Todos os resultados foram obtidos através de um **processo evolucionário completamente não supervisionado**. Para propósitos de avaliação, os resultados dos gráficos **reportam o maior valor de MAP de toda a população** ao longo das iterações. Os **resultados contidos nas tabelas são da combinação com a maior aptidão na última população (\mathcal{X}^*)**. Por essa razão, os resultados das tabelas podem diferir dos exibidos pelos gráficos.

Todos os experimentos apresentados nesse capítulo consideram apenas os cenários completos, uma vez que os cenários personalizados possuem apenas 6 descritores e, em consequência, apenas 36 combinações possíveis. Dessa forma, o espaço de busca seria percorrido por completo pelo algoritmo genético em poucas gerações.

7.2.1 Análise dos Parâmetros

A coleção Flowers foi escolhida para avaliar os parâmetros (exceto o valor de k , em que mais coleções foram consideradas), uma vez que é a menor coleção presente no

protocolo experimental. Observa-se que os resultados são muito superiores aos obtidos pelo melhor descritor (CNN-FBResNet com MAP de **52.56%** para a coleção Flowers).

- **Função de aptidão:** um experimento foi conduzido a fim de comparar os resultados de cada uma das três estimativas de eficácia quando consideradas como função de aptidão para o *framework* do algoritmo genético proposto. A Figura 27 apresenta os resultados dos indivíduos que possuem o maior MAP de toda a população ao longo de 30 gerações. Os resultados revelaram que a métrica híbrida apresenta os melhores resultados e se posiciona sutilmente acima do *Authority*. Portanto, a métrica híbrida foi considerada como a estimativa de eficácia para todos os experimentos apresentados para o UGAF-RSF;

- **Tamanho da população e número de gerações:** A Figura 28 apresenta o algoritmo genético ao longo de 60 gerações para cinco diferentes tamanhos de população (25, 50, 100, 150 e 200). Como pode ser visto, há pouco ganho de eficácia após as 30 primeiras gerações. Sendo assim, 30 gerações foi adotado como valor padrão para os demais experimentos. Observa-se também que quanto maior o tamanho da população, maior os resultados obtidos. No entanto, há um *trade-off* entre eficiência e eficácia neste contexto. Portanto, para todos os experimentos restantes, optou-se por utilizar o valor 100 como padrão para o tamanho da população, uma vez que este demonstrou fornecer resultados próximos aos apresentados por 150 e 200;

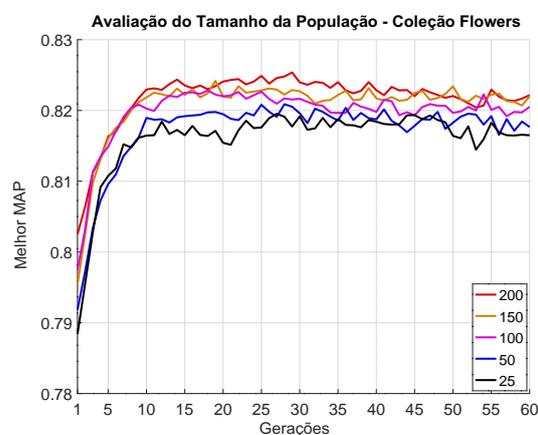
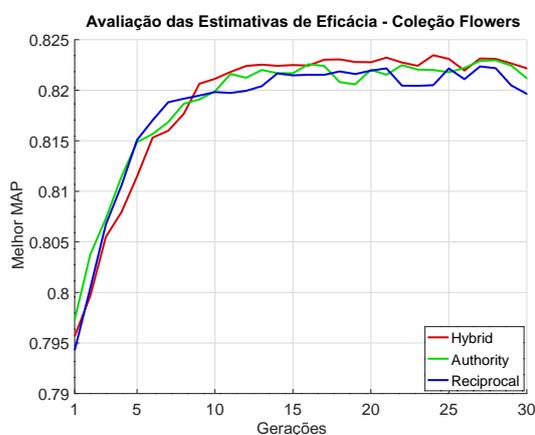


Figura 27 – Comparação entre as diferentes funções de aptidão.

Figura 28 – Análise do tamanho da população.

- **Taxa de Cruzamento:** A Figura 29 apresenta uma análise da taxa de cruzamento. Constatou-se que a variação da taxa de cruzamento não gera um impacto muito significativo nos resultados. Para os demais experimentos, adotou-se uma taxa de cruzamento de 70%, sendo que esse valor ofereceu os melhores resultados.

- **Taxa de Mutação:** A Figura 30 apresenta uma análise da taxa de mutação. Novamente, observa-se que não há grande variação entre os resultados. Apesar disso, o valor de 8% se revelou levemente superior aos demais e, por essa razão, é definido como valor padrão para os experimentos restantes.

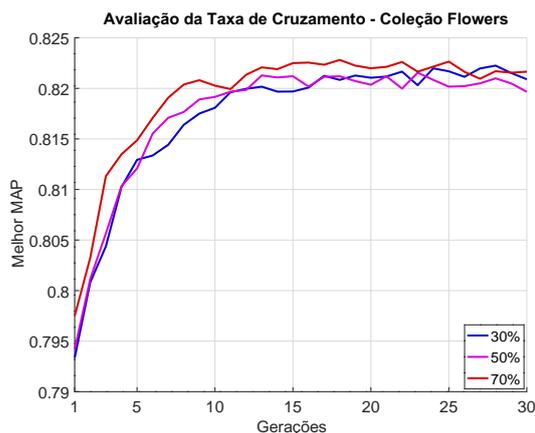


Figura 29 – Avaliação da taxa de cruzamento.

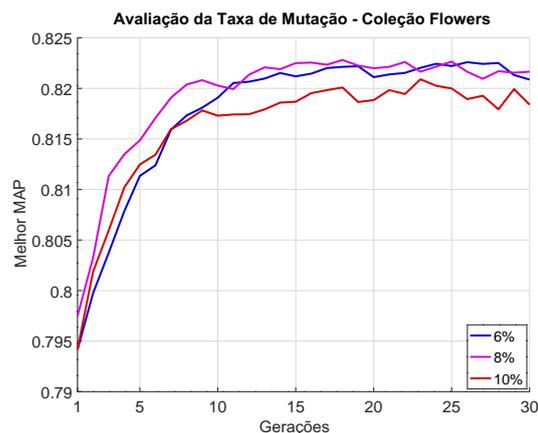


Figura 30 – Avaliação da taxa de mutação.

- **Estratégia Evolutiva:** três diferentes estratégias são consideradas com o propósito de melhorar os resultados do algoritmo genético. **(1) Aumento do elitismo:** nesta estratégia, a taxa de elitismo é elevada gradualmente ao longo das gerações; **(2) decremento do cruzamento e mutação:** nesta estratégia, as taxas de cruzamento e mutação são gradualmente decrementadas ao longo das gerações; **(3) ambas 1 e 2 em conjunto:** aplica ambas as estratégias anteriores simultaneamente.

A Figura 31 avalia o impacto das estratégias propostas nos resultados. Os melhores resultados foram revelados pela estratégia (1), a qual é aplicada em todos os experimentos reportados nas tabelas, ou seja, os valores de \mathfrak{X}^* ;

- **Taxa de Elitismo Máxima:** a taxa determina o valor máximo de elitismo que pode ser atingido ao utilizar a estratégia (1), ou seja, o limite do incremento. O parâmetro é avaliado pela Figura 32. Considerou-se 20% para todos os experimentos restantes;

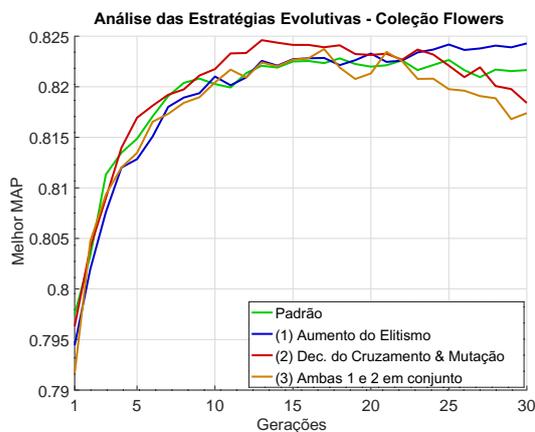


Figura 31 – Comparação das estratégias evolutivas.

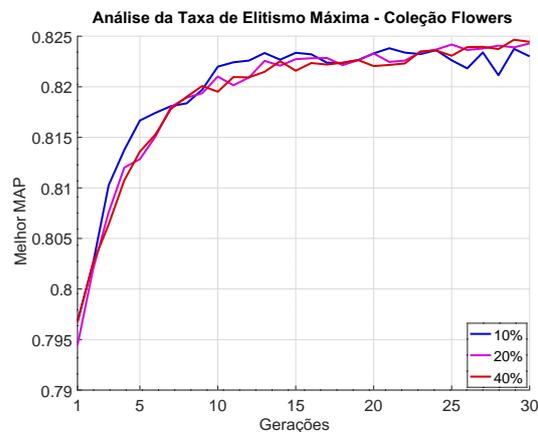


Figura 32 – Análise da taxa de elitismo máxima.

- **Conjunto de Vizinhança:** o conjunto de vizinhança k é considerado para calcular a função de aptidão e como parâmetro para o método de agregação (nesse caso, o CPRR).

Tal parâmetro é avaliado nas coleções Flowers (Figura 33 (a)), Holidays (Figura 33 (b)) e UKBench (Figura 33 (c)). A partir dos resultados obtidos, adotou-se $k = 50$ para as coleções Flowers e Corel5k, $k = 3$ para a Holidays e $k = 6$ para a UKBench. Em geral, coleções com poucas imagens por classe requerem valores de k menores e vice-versa.

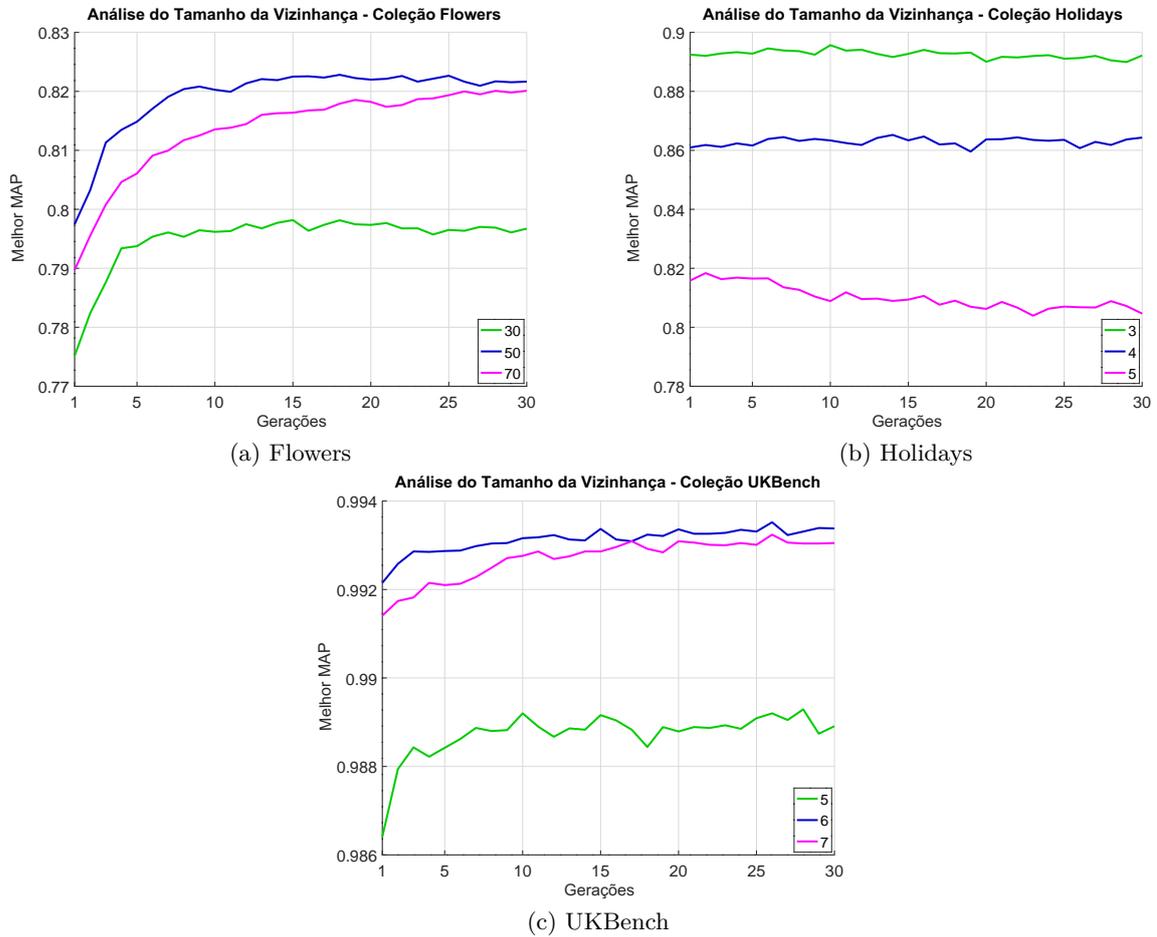


Figura 33 – Análise do tamanho da vizinhança k .

Tabela 13 – Combinações mais frequentemente selecionadas e a média do MAP para cada uma das coleções.

Coleção	Combinação Frequentemente Selecionada (\mathcal{X}^*)	MAP (%)	Ganho Relativo
Flowers	ACC + AlexNet + BnInception + ResNeXt + ResNet + SIFT	80.92 ± 0.6005	+53.96%
Corel5k	CEDD + BnInception + FBResNet + InceptionV4 + NASnet-Large + ResNeXt + GIST + LBP + SCD	91.45 ± 0.3738	+40.37%
UKBench	AlexNet + BnInception + InceptionV4 + OLDFP + ResNeXt + SENet + VGGNet + COMO + FOH + GIST + SCD + SIFT + SPACC + VOC	99.25 ± 0.0573	+1.54%
Holidays	ACC + CLD + AlexNet + BnInception + BnVGGNet + FBResNet + ResNeXt + ResNet + SENet + VGGNet + FOH + PHOG + SIFT + SPACC + SPCEDD + SPFCTH	86.99 ± 1.1297	-1.66%

Os resultados obtidos pelo *framework* proposto são apresentados na Tabela 13, em que (\mathcal{X}^*) indica a combinação mais frequentemente selecionada (uma vez que o método é estocástico) e o MAP médio de 10 execuções. O ganho relativo é calculado em relação ao melhor descritor isolado em cada caso. Observa-se que o UGAF-RSF atingiu ganhos positivos na maior parte das circunstâncias apresentadas.

8 Discussões, Comparações e Análises

Este capítulo tem como intuito analisar os resultados de maneira conjunta e descrever as distinções das técnicas apresentadas, destacando as contribuições. A Seção 8.1 compara os três métodos entre si e também com os principais métodos da literatura. Enquanto a Seção 8.2 apresenta resultados visuais (qualitativos) evidenciando o impacto da seleção e combinação, a Seção 8.3 conduz uma análise experimental considerando métodos de aprendizado não supervisionado como ranqueadores de entrada dos métodos propostos. A Seção 8.4 discute as principais vantagens e desvantagens de cada uma das abordagens propostas e as diferenças entre elas. Por fim, a Seção 8.5 apresenta as contribuições do trabalho e lista todas as publicações elaboradas em virtude do mestrado.

8.1 Avaliação Comparativa

Esta seção possui como objetivo apresentar uma comparação entre os três métodos propostos com diferentes *baselines* da literatura (*late* e *early fusion*) incluindo abordagens do estado-da-arte.

8.1.1 Comparação com *Baselines* de *Late Fusion*

Para todos os métodos da literatura considerados como *baselines* foram empregados os parâmetros padrões reportados pelos autores em seus respectivos trabalhos. Em relação a casos nos quais a coleção ainda não havia sido avaliada no artigo original, utilizou-se o mesmo valor de k adotado pelos nossos métodos propostos, de forma que a avaliação possa ser a mais justa possível. Devido ao custo computacional dos métodos considerados como *baselines*, a fim de viabilizar as execuções em todas as coleções, os resultados são apresentados apenas para os cenários personalizados. A comparação das abordagens propostas com diferentes *baselines* de *late fusion* é exibida na Tabela 14.

Tabela 14 – Comparação com métodos de *late fusion* nos cenários personalizados.

Método	MAP (%)				
	MPEG-7	Flowers	Corel5k	UKBench	Holidays
<i>Correlation Graph</i> [136]	95.79	46.22	58.25	96.06	65.11
<i>Query-Adaptive Fusion</i> [137]	92.03	49.56	69.60	96.61	82.59
<i>Graph Fusion</i> [100]	99.05	53.60	68.21	99.24	85.71
USRF	99.92	79.10	90.32	99.02	90.51
RGSF	—	79.45	88.73	98.84	90.51

O UGAF-RSF não foi incluído na comparação, uma vez que tal método possui resultados apenas para os cenários completos. Constata-se que as abordagens propostas

são superiores às demais em quase todas as coleções. No caso da UKBench, os nossos MAPs são pouco inferiores ao obtido pelo *Graph Fusion* [100].

8.1.2 Comparação com *Baselines* de *Early Fusion*

Apesar dos métodos propostos na dissertação serem de *late fusion*, esta seção apresenta uma comparação com métodos de seleção de características *early fusion*, os quais também são não supervisionados. Para que a comparação seja possível, um conjunto de etapas foram realizadas.

Após aplicar PCA e regularização L2-norm para todos os vetores, os métodos de *early fusion* foram empregados a fim de selecionar as 100 características mais relevantes dentre as disponíveis. A partir da distância euclidiana dos novos vetores, foram calculadas as listas ranqueadas que foram submetidas ao método de agregação CPRR que forneceu os resultados finais para comparação. A fim de viabilizar as execuções, os resultados são apresentados apenas para as coleções Flowers e Corel5K. Um comparativo da abordagem proposta com métodos de seleção de características, *early fusion*, são apresentados na Tabela 15.

Tabela 15 – Comparação com métodos de *early fusion*.

Método	MAP (%)			
	Flowers		Corel5k	
	Perso.	Compl.	Perso.	Compl.
<i>Laplace</i> [25]	70.84	61.28	86.09	78.40
SPEC [26]	71.46	49.67	72.82	63.99
MCFS [27]	75.81	55.95	86.38	84.74
UDFS [28]	69.70	63.97	80.77	78.20
NDFS [29]	71.68	65.41	86.09	87.17
USRF	79.10	81.71	90.32	89.88
RGSF	79.45	81.33	88.73	87.54
UGAF-RSF	—	80.92 ±	—	91.45 ±
	—	0.6005	—	0.3738

8.1.3 Comparação com Estado-da-arte

Por fim, nossos métodos foram avaliados em contraste às principais abordagens de estado-da-arte, dentre elas técnicas de pós-processamento e métodos de recuperação de imagens. As comparações são apresentadas para as coleções: Holidays, UKBench e MPEG-7. Tais coleções foram utilizadas por serem usualmente utilizadas como *benchmark* em recuperação de imagens pelo conteúdo. Ressalta-se que a comparação com o estado-da-arte é um critério muito rigoroso, uma vez que os métodos propostos têm como entrada diversos ranqueadores sem nenhuma pré-seleção (cenário completo); **enquanto os métodos da literatura em grande parte consideram ranqueadores pré-selecionados pelos próprios autores.**

Ressalta-se que os resultados reportados nesta subseção consideram as mesmas combinações (\mathfrak{X}^*) apresentados nos capítulos anteriores para cada um dos métodos. No entanto, para as coleções UKBench e MPEG-7 os resultados são reportados utilizando *N-S score* e *Recall@40*, respectivamente. A mudança de métrica para avaliação se deve ao fato da maior parte dos métodos da literatura reportarem resultados considerando esse protocolo.

A Tabela 16 apresenta os resultados para a coleção MPEG-7 utilizando o *Recall@40*, também conhecido como *bull's eye score*. Constata-se que ambos o USRF e RGSF foram capazes de selecionar uma combinação com o resultado máximo de **100%**.

Tabela 16 – Comparação com diversos métodos de pós-processamento na coleção MPEG-7 (*Bull's eye score* - *Recall@40*).

Descritores de Contorno		<i>Recall@40</i>
<i>Beam Angle Statistics</i> (BAS) [34]		75.21%
<i>Contour Feat. Descriptor</i> (CFD) [40]		84.43%
<i>Inner Dist. Shape Context</i> (IDSC) [116]	-	85.40%
<i>Aspect Shape Context</i> (ASC) [41]		88.39%
<i>Articulation-Invariant Rep.</i> (AIR) [115]		93.67%
Métodos de pós-processamento	Descritores	<i>Recall@40</i>
<i>Contextual Dissimilarity Measure</i> [149]		88.30%
<i>Graph Transduction</i> [60]		91.00%
<i>Self-Smoothing Operator</i> [150]		92.77%
<i>Local Constr. Diff. Process</i> [17]		93.32%
<i>Mutual kNN Graph</i> [18]	IDSC [116]	93.40%
<i>SCA</i> [151]		93.44%
<i>Smooth Neighborhood</i> [152]		93.52%
<i>Reciprocal kNN Graph CCs</i> [153]		93.62%
<i>Graph Fusion</i> [100]		89.76%
<i>Index-Based Re-Ranking</i> [154]		92.85%
<i>RL-Sim</i> [155]		94.27%
<i>Correlation Graph</i> [136]	CFD [40]	94.84%
<i>Reciprocal kNN Graph CCs</i> [153]		96.51%
<i>Generic Diffusion Process</i> [156]		93.95%
<i>Index-Based Re-Ranking</i> [154]		94.09%
<i>Correlation Graph</i> [136]		95.50%
<i>Local Constr. Diff. Process</i> [17]		95.96%
<i>Smooth Neighborhood</i> [152]	ASC [41]	95.98%
<i>Reciprocal kNN Graph CCs</i> [153]		96.04%
<i>Tensor Product Graph</i> [157]		96.47%
<i>Graph Fusion</i> [100]		98.76%
<i>Index-Based Re-Ranking</i> [154]		99.93%
<i>RL-Sim</i> [155]		99.94%
<i>Tensor Product Graph</i> [157]	AIR [115]	99.99%
<i>Generic Diffusion Process</i> [156]		100%
<i>Neighbor Set Similarity</i> [158]		100%
<i>Reciprocal kNN Graph CCs</i> [153]		100%
USRF	Comb. Seleccionada	100%
RGSF	Comb. Seleccionada	100%

Os resultados na coleção Holidays são exibidos na Tabela 17. Vários métodos de estado-da-arte são incluídos na comparação. Observa-se que, entre as abordagens propostas,

o maior MAP foi de **90.51%**, situando-se pouco atrás de [159] que obteve 90.89%. É possível perceber que o grande número de ranqueadores disponíveis no cenário completo não comprometeram a seleção do USRF, obtendo resultados equivalentes ou comparáveis ao estado-da-arte em ambos os casos.

Tabela 17 – Comparação com o estado-da-arte na coleção Holidays (MAP).

MAP para métodos do estado-da-arte				
Jégou <i>et al.</i> [107]	Tolias <i>et al.</i> [160]	Paulin <i>et al.</i> [161]	Qin <i>et al.</i> [162]	Zheng <i>et al.</i> [163]
75.07%	82.20%	82.90%	84.40%	85.20%
Sun <i>et al.</i> [164]	Zheng <i>et al.</i> [165]	Pedronette <i>et al.</i> [153]	Li <i>et al.</i> [166]	Liu <i>et al.</i> [159]
85.50%	85.80%	86.19%	89.20%	90.89%

Método Proposto	Cenário	
	Perso.	Compl.
USRF	90.51%	90.51%
RGSF	90.51%	89.56%
UGAF-RSF	—	86.99% ±
	—	1.1297

Analogamente, a Tabela 18 apresenta uma comparação dos resultados na UKBench com os principais resultados de *benchmark* para a coleção. O maior *N-S score*, o qual é similar ao P@4, atingido no cenário personalizado é de 3.93 (RGSF) e em torno de **3.95** (UGAF-RSF) no cenário completo. O cenário completo é mais desafiador por requerer a seleção em um grande conjunto de ranqueadores. Apesar disso, os métodos propostos, em geral, atingiram resultados ainda mais significativos do que no cenário personalizado, evidenciando a eficácia dos algoritmos de seleção propostos. Nota-se que o UGAF-RSF fornece o melhor resultado, superior ao apresentado por [167].

Tabela 18 – Comparação com o estado-da-arte na coleção UKBench (*N-S score*).

<i>N-S scores</i> para métodos do estado-da-arte					
Zheng <i>et al.</i> [168]	Wang <i>et al.</i> [101]	Sun <i>et al.</i> [164]	Paulin <i>et al.</i> [161]	Zhang <i>et al.</i> [100]	Zheng <i>et al.</i> [137]
3.57	3.68	3.76	3.76	3.83	3.84
Bai <i>et al.</i> [151]	Xie <i>et al.</i> [169]	Liu <i>et al.</i> [159]	Pedronette <i>et al.</i> [153]	Bai <i>et al.</i> [167]	
3.86	3.89	3.92	3.93	3.94	

Método Proposto	Cenário	
	Perso.	Compl.
USRF	3.90	3.94
RGSF	3.93	3.79
UGAF-RSF	—	3.95 ±
	—	0.3206

8.2 Análise Visual

Com o propósito de fornecer uma visualização qualitativa dos resultados, a Figura 34 ilustra dois exemplos de listas ranqueadas que comparam os resultados individuais de cada um dos ranqueadores (descritores) e o obtido pela combinação selecionada pelo USRF na coleção UKBench. As imagens de consulta são apresentadas em bordas verdes e os resultados incorretos em bordas vermelhas.

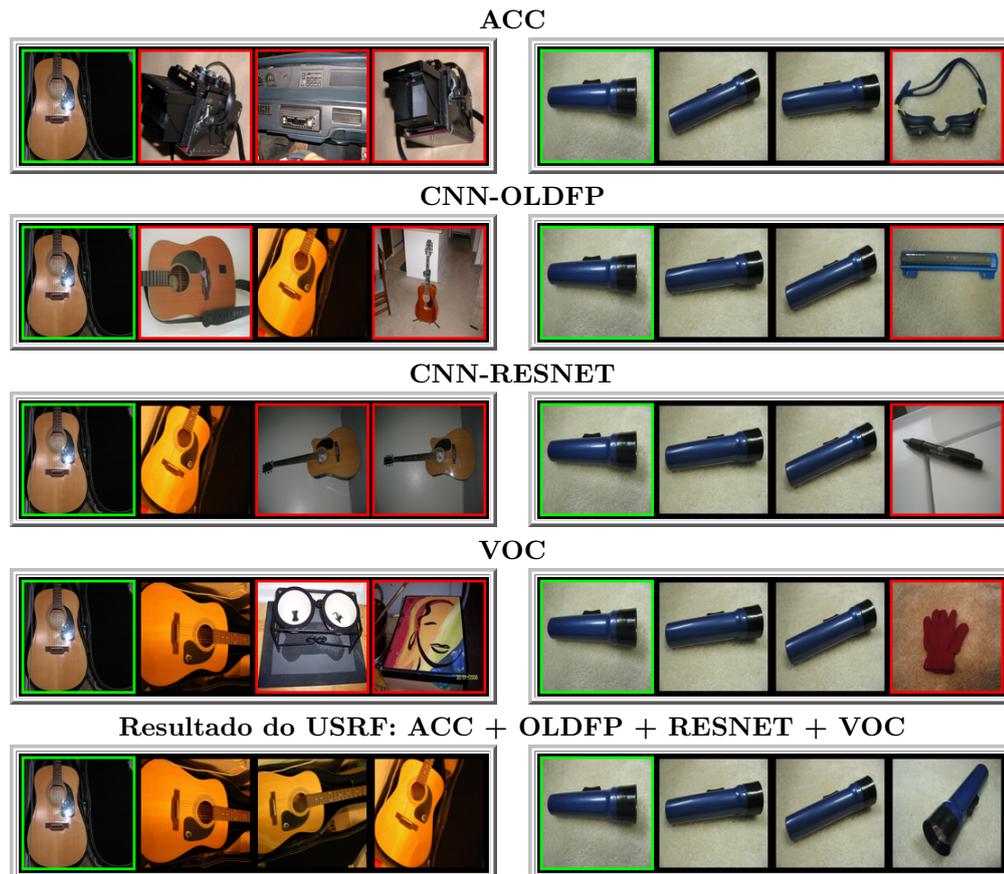


Figura 34 – Dois exemplos de resultados visuais que ilustram o impacto da seleção e combinação na coleção UKBench.

8.3 Combinação de Métodos de Aprendizado Não Supervisionado

Todos os experimentos anteriores consideraram como entrada dos algoritmos ranqueadores originados a partir da distância euclidiana de descritores de imagens. Esta seção apresenta resultados onde foram fornecidos como entrada dos métodos ranqueadores que tiveram como origem o resultado de saída de diferentes métodos de aprendizado não supervisionado. Para todas as coleções, foram tomados os seis ranqueadores dos cenários personalizados e estes foram submetidos como entrada de sete diferentes métodos de aprendizado não supervisionado, originando 42 ranqueadores para cada coleção. Os métodos utilizados são descritos na Tabela 19.

Tabela 19 – Métodos de aprendizado não supervisionado utilizados para o experimento.

Método	Descrição
<i>ContextRR</i>	A partir da informação contextual das listas ranqueadas, uma distância mais efetiva é calculada.
<i>Correlation Graph</i>	É construído um grafo de correlações, em que as componentes fortemente conexas são utilizadas para realizar incrementos de similaridade entre as imagens.
CPRR	Realiza operações de produto cartesiano nos conjuntos de vizinhança k NN e k NN reverso obtidos a partir das listas ranqueadas.
<i>Ranked List Graph Distance</i>	Cada lista ranqueada é representada como um grafo ponderado considerando as suas primeiras k posições. O peso das arestas é utilizado como incremento das similaridades.
<i>ReckNNGraph</i>	Um grafo considerando as referências recíprocas nas primeiras k posições das listas ranqueadas é construído. A estrutura do grafo é explorada a fim de analisar a estrutura intrínseca da base de imagens.
<i>RL-Recommendation</i>	A informação de similaridade das listas ranqueadas é utilizada a fim de recomendar as imagens entre si.
<i>RL-Sim*</i>	Se duas imagens são similares, suas listas ranqueadas também são. Dessa forma, as distâncias são decrementadas.

A Tabela 20 apresenta o MAP para a combinação de métodos de aprendizado não supervisionado em cada uma das coleções. Os resultados foram incluídos parcialmente para o UGAF-RSF devido ao alto custo computacional do método. Observa-se que os resultados são, em geral, inferiores aos obtidos na combinação de descritores, mas ainda assim superiores aos melhores descritores de cada uma das coleções. Com exceção da UKBench e Holidays, em que percebe-se resultados inferiores ao melhor descritor no caso do RGSF. A dificuldade em se obter resultados acima do melhor descritor nas coleções UKBench e Holidays se deve ao fato de o melhor descritor possuir um MAP bastante acima dos demais, o que torna o cenário de seleção mais sensível nesses casos. Além disso, ambas as coleções possuem um número pequeno de imagens por classe.

Tabela 20 – Resultados de MAP para combinação de métodos de aprendizado não supervisionado.

Coleção de Imagens	MAP (%)			
	Melhor Desc.	USRF	RGSF	UGAF-RSF
MPEG-7	89.39	98.12	97.21	100.0 ± 0.0000
Flowers	52.56	79.10	74.39	78.89 ± 0.5838
Corel5k	65.15	86.96	80.11	—
UKBench	97.74	99.08	82.48	—
Holidays	88.46	90.36	76.92	87.20 ± 1.4534

8.4 Discussões

Esta seção tem como objetivo comparar os três métodos propostos neste trabalho (USRF, RGSF e UGAF-RSF) de maneira conjunta a partir de seus resultados de eficácia, eficiência e também discutir suas principais vantagens e desvantagens em cada caso.

Os resultados apresentados até o momento consideram apenas os valores de eficácia. No entanto, para realizar uma comparação completa entre os métodos, uma análise de eficiência foi conduzida. A Tabela 21 apresenta os tempos de execução dos métodos propostos em quatro das principais coleções considerando os cenários completos. Os resultados foram obtidos a partir da média e desvio padrão de 5 execuções em uma máquina com *hardware* composto pela *CPU Intel Xeon E5-2620 2.40GHz*, 64 GB de RAM e ambiente de *software* Ubuntu 14.04. As execuções foram feitas com paralelismo em 24 *threads*, o que possibilitou a execução em paralelo de 24 combinações distintas por iteração de cada um dos métodos. Pode ser observado que o RGSF é o método mais eficiente em todos os casos. O USRF possui tempos de execução, de certa forma, próximos aos do RGSF; enquanto o UGAF-RSF é muito mais custoso e pode requerer diversas horas para uma única execução.

Tabela 21 – Tempo de execução dos métodos propostos nos cenários completos.

Coleção de Imagens	Média do Tempo Total de Execução (segundos)		
	USRF	RGSF	UGAF-RSF
Flowers	1094.564 ± 6.8096	752.158 ± 4.8114	1871.241 ± 62.3845
Corel5k	8116.090 ± 29.7900	3295.157 ± 6.0478	46539.312 ± 502.2175
UKBench	882.952 ± 10.9725	247.672 ± 1.7095	6685.592 ± 336.6056
Holidays	417.486 ± 0.8452	104.478 ± 0.3991	6559.075 ± 230.1128

A Tabela 22 ranqueia cada um dos métodos considerando os resultados obtidos tanto para eficácia quanto eficiência em cada uma das coleções nos cenários completos. Observa-se que, em termos de eficácia, o USRF e o UGAF-RSF possuem os melhores resultados. No entanto, são mais custosos que o RGSF em todos os casos avaliados.

Tabela 22 – Análise de eficácia e eficiência dos métodos propostos.

Critério Avaliado	Classificação (<i>Rank</i>)		
	1°	2°	3°
Avaliação de Eficácia			
Melhor MAP na Flowers	USRF	RGSF	UGAF-RSF
Melhor MAP na Corel5k	UGAF-RSF	USRF	RGSF
Melhor MAP na UKBench	UGAF-RSF	USRF	RGSF
Melhor MAP na Holidays	USRF	RGSF	UGAF-RSF
Avaliação de Eficiência			
Tempo de Exec. na Flowers	RGSF	USRF	UGAF-RSF
Tempo de Exec. na Corel5k	RGSF	USRF	UGAF-RSF
Tempo de Exec. na UKBench	RGSF	USRF	UGAF-RSF
Tempo de Exec. na Holidays	RGSF	USRF	UGAF-RSF

8.5 Contribuições e Publicações

Esta dissertação apresentou como contribuição três novos métodos de *late fusion* completamente não supervisionados que podem ser aplicados à seleção e combinação de ranqueadores. Cada um deles emprega abordagens distintas para explorar o espaço de combinações e, em geral, apresentaram ganhos consideráveis de eficácia nos diferentes cenários avaliados.

No decorrer do mestrado, foram elaborados oito artigos científicos, listados a seguir. Dentre eles há quatro artigos publicados (1 a 4) e quatro submetidos (5 a 8) em importantes revistas e congressos da área. Com exceção dos artigos (2) e (5), que correspondem a colaborações, todos possuem o aluno como primeiro autor. Os conteúdos dos artigos (6), (7) e (8) são apresentados nos Capítulos 5, 6 e 7 do texto, respectivamente. Os demais artigos não estão contidos na dissertação, mas podem ser consultados nas publicações.

O artigo (1) consiste em uma continuação de trabalho iniciado na graduação e aborda o problema de seleção por meio de métricas supervisionadas, em que algumas das premissas propostas são exploradas ao longo desta dissertação. Os artigos (3), (4) e (5) consistem em trabalhos desenvolvidos pelo aluno durante o mestrado, mas cujos temas, embora relacionados, diferem do tema central dessa dissertação. O artigo (3) foi publicado como resultado do trabalho de conclusão de uma das disciplinas do mestrado.

1. **VALEM, L. P.** ; PEDRONETTE, D. C. G. . Selection and Combination of Unsupervised Learning Methods for Image Retrieval. In: 15th International Workshop on Content-Based Multimedia Indexing (CBMI' 17), 2017, Florence - Italy.
(Qualis B1 - Publicado [148])
2. ALMEIDA, J. ; **VALEM, L. P.** ; PEDRONETTE, D. C. G. . A Rank Aggregation Framework for Video Interestingness Prediction. In: 9th International Conference on Image Analysis and Processing (ICIAP), 2017, Catania - Italy.
(Qualis B1 - Publicado [170])
3. **VALEM, L. P.** ; PEDRONETTE, D. C. G. ; BREVE, F. ; GUILHERME, I. R. . Manifold Correlation Graph for Semi-Supervised Learning. In: IJCNN IEEE WCCI, 2018, Rio de Janeiro - Brazil.
(Qualis A1 - Publicado [171])
4. **VALEM, L. P.** ; DE OLIVEIRA C. R. ; PEDRONETTE, D. C. G. ; ALMEIDA, J. Unsupervised Similarity Learning through Rank Correlation and kNN Sets. In: The ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM); 2018.
(Qualis B1 - Publicado [172])

5. PEDRONETTE, D. C. G. ; **VALEM, L. P.** ; ALMEIDA, J. ; TORRES, R. da S. .
Multimedia Retrieval through Unsupervised Hypergraph-based Manifold Ranking.
In: IEEE Transactions on Image Processing (TIP).
(Qualis A1 - Submetido)
6. **VALEM, L. P.** ; PEDRONETTE, D. C. G. . Unsupervised Selective Rank Fusion
for Image Retrieval Tasks. In: Image and Vision Computing.
(Qualis A1 - Em Submissão)
7. **VALEM, L. P.** ; PEDRONETTE, D. C. G. . Graph-based Selective Rank Fusion
for Unsupervised Image Retrieval. In: Pattern Recognition Letters.
(Qualis A1 - Submetido)
8. **VALEM, L. P.** ; PEDRONETTE, D. C. G. . An Unsupervised Genetic Algorithm
Framework for Rank Selection and Fusion on Image Retrieval. In: ACM International
Conference on Multimedia Retrieval (ICMR).
(Qualis A1 - Submetido)

9 Conclusões

A dissertação apresentou como contribuição três novos métodos para seleção e combinação de ranqueadores completamente não supervisionados que consideram técnicas distintas. Dentre os principais conceitos empregados, fez-se o uso de métricas de estimativa de eficácia, métricas de correlação, grafos, abordagens de reranqueamento e também algoritmos genéticos. Todos os métodos propostos foram avaliados em diferentes cenários demonstrando resultados muito promissores, os quais são próximos ou superiores ao estado-da-arte.

Diversos desafios foram encontrados durante a execução deste trabalho. Apesar do grande número de métodos desenvolvidos na área de aprendizado de máquina e CBIR em geral, métodos robustos capazes de selecionar um grande conjunto de ranqueadores de maneira completamente não supervisionada são, pelo que se tem conhecimento, escassos na literatura. Como a maior parte dos métodos utilizam de conjuntos pequenos e pré-selecionados de descritores, encontrar *baselines* adequados foi uma das tarefas mais difíceis durante a pesquisa.

A análise experimental foi conduzida em diferentes contextos, considerando cinco coleções de imagens e uma grande diversidade de descritores (mais de trinta) de diversas categorias e tipos. O que demonstrou que nossos métodos não são só capazes de explorar informações complementares entre diversos ranqueadores, mas também de filtrar os ranqueadores menos relevantes tanto em pequenos como em grandes conjuntos.

Os métodos propostos foram comparados considerando o mesmo conjunto de descritores tanto em *baselines* de *early* quanto *late fusion*, se revelando superiores ou equiparáveis na maioria dos casos. A execução de métodos de terceiros é uma tarefa complexa pelo fato da grande parte dos métodos não possuírem código aberto e serem supervisionados ou semi-supervisionados. Manteve-se a maior fidelidade possível aos parâmetros utilizados pelos autores originais.

As comparações com o estado-da-arte foram possíveis em virtude da grande diversidade de coleções de imagens avaliadas e pelo fato destas serem frequentemente utilizadas como *benchmarks* em CBIR. Nota-se que os métodos propostos são comparáveis ou superiores aos demais métodos nos casos apresentados.

9.1 Trabalhos Futuros

Dentre os diversos trabalhos futuros possíveis, é apresentada a seguir uma lista de alguns tópicos que podem ser considerados para uma continuação desse trabalho:

- **Utilização de outros métodos de agregação:** Como as abordagens propostas para seleção e combinação são muito flexíveis, novos métodos de agregação podem ser utilizados para agregar os ranqueadores. Pode-se, por exemplo, avaliar os disponíveis no UDLF [173], uma plataforma de código aberto com diversos métodos de reranqueamento não supervisionados;
- **Proposta de novas estimativas de eficácia:** As estimativa de eficácia utilizadas neste trabalho se baseiam na vizinhança recíproca de elementos nas listas ranqueadas. Dessa forma, é possível que sejam atribuídos valores altos de estimativa de eficácia caso os ranqueadores errem de maneira consistente. Portanto, dentre os trabalhos futuros está propor e avaliar novas abordagens para estimar a eficácia das listas ranqueadas;
- **Alterações no grafo do RGSF:** O método RGSF considera apenas a correlação para formular o peso das arestas. No entanto, é possível incorporar a estimativa de eficácia e outras métricas de diversidade;
- **Programação genética:** Para a seleção e combinação de ranqueadores, o UGAF-RSF utiliza de um algoritmo genético. No entanto, é possível fazer uma modelagem de programação genética. Em tal abordagem, cada indivíduo da população pode ser representado como uma árvore na qual os nós folhas são ranqueadores e os nós externos são métodos de pós-processamento não supervisionados;
- **Outras modalidades de recuperação:** Há também a possibilidade de aplicar e avaliar os métodos propostos em outros cenários de recuperação como som, vídeo e texto, por exemplo.

Referências

- [1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Computing Surveys*, vol. 40, no. 2, pp. 5:1–5:60, 2008.
- [2] C. Kofler, M. Larson, and A. Hanjalic, “User intent in multimedia search: A survey of the state of the art and future challenges,” *ACM Comput. Surv.*, vol. 49, pp. 36:1–36:37, Aug. 2016.
- [3] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, “Faceted metadata for image search and browsing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’03, pp. 401–408, 2003.
- [4] M. Agarwal and J. Mostafa, “Content-based image retrieval for alzheimer’s disease detection,” in *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 13–18, June 2011.
- [5] M. Sultana and M. Gavrilova, *A Content Based Feature Combination Method for Face Recognition*, pp. 197–206. Heidelberg: Springer International Publishing, 2013.
- [6] M. Wang and T. Song, “Remote sensing image retrieval by scene semantic matching,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 2874–2886, May 2013.
- [7] J. pierre Schober, T. Hermes, and O. Herzog, “Content-based image retrieval by ontology-based object recognition,” in *in KI-2004 Workshop on Applications of Description Logics*, pp. 61–67, 2004.
- [8] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1349–1380, Dec 2000.
- [9] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: State of the art and challenges,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, pp. 1–19, Feb. 2006.
- [10] S. C. Hoi, W. Liu, and S.-F. Chang, “Semi-supervised distance metric learning for collaborative image retrieval and clustering,” *ACM Transactions on Multimedia Computing and Communication Applications*, vol. 6, pp. 18:1–18:26, August 2010.
- [11] B. Thomee and M. S. Lew, “Interactive search in image retrieval: a survey,” *International Journal of Multimedia Information Retrieval*, vol. 1, pp. 71–86, Jul 2012.

-
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’12, pp. 1097–1105, 2012.
- [13] L. Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning,” *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.
- [14] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, *Neural Codes for Image Retrieval*, pp. 584–599. Cham: Springer International Publishing, 2014.
- [15] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.
- [16] X. Yang and L. J. Latecki, “Affinity learning on a tensor product graph with applications to shape and image retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2011)*, pp. 2369–2376, 2011.
- [17] X. Yang, S. Koknar-Tezel, and L. J. Latecki, “Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval.,” in *CVPR*, pp. 357–364, 2009.
- [18] P. Kontschieder, M. Donoser, and H. Bischof, “Beyond pairwise shape similarity analysis,” in *ACCV*, pp. 655–666, 2009.
- [19] D. C. G. Pedronette and R. da S. Torres, “Image re-ranking and rank aggregation based on similarity of ranked lists,” *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.
- [20] L. P. Valem and D. C. G. Pedronette, “Unsupervised similarity learning through cartesian product of ranking references for image retrieval tasks,” *SIBGRABI*, 2016.
- [21] F. F. Faria, A. Veloso, H. M. Almeida, E. Valle, R. d. S. Torres, M. A. Gonçalves, and W. Meira, Jr., “Learning to rank for content-based image retrieval,” in *Proceedings of the International Conference on Multimedia Information Retrieval, MIR ’10*, pp. 285–294, 2010.
- [22] M. L. Kherfi and D. Ziou, “Relevance feedback for cbir: a new approach based on probabilistic feature weighting with positive and negative examples,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 1017–1030, 2006.
- [23] J. A. Shaw, E. A. Fox, J. A. Shaw, and E. A. Fox, “Combination of multiple searches,” in *The Second Text REtrieval Conference (TREC-2)*, pp. 243–252, 1994.

- [24] J. A. Vargas Muñoz, R. da Silva Torres, and M. A. Gonçalves, “A soft computing approach for learning to aggregate rankings,” in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15*, pp. 83–92, 2015.
- [25] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS'05*, pp. 507–514, 2005.
- [26] Z. Zhao and H. Liu, “Spectral feature selection for supervised and unsupervised learning,” in *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pp. 1151–1157, 2007.
- [27] D. Cai, C. Zhang, and X. He, “Unsupervised feature selection for multi-cluster data,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10*, pp. 333–342, 2010.
- [28] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, “L_{2,1}-norm regularized discriminative feature selection for unsupervised learning,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two, IJCAI'11*, pp. 1589–1594, 2011.
- [29] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu, “Unsupervised feature selection using nonnegative spectral analysis,” in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [30] B. R.-N. Ricardo Baeza-Yates, *Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca*. Editora Bookman, 2013.
- [31] L. Piras and G. Giacinto, “Information fusion in content based image retrieval: A comprehensive overview,” vol. 37, pp. 50 – 60, 2017.
- [32] S. V. Shriram K. Vasudevan, P.L.K. Priyadarsini, *Content Based Image Retrieval (CBIR): A deeper insight*. LAP LAMBERT Academic Publishing, 2012.
- [33] R. D. S. Torres and A. X. Falcão, “Content-based image retrieval: Theory and applications,” *Revista de Informática Teórica e Aplicada*, vol. 13, pp. 161–185, 2006.
- [34] N. Arica and F. T. Y. Vural, “BAS: a perceptual shape descriptor based on the beam angle statistics,” *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1627–1639, 2003.
- [35] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, “Image indexing using color correlograms,” in *CVPR*, pp. 762–768, 1997.

-
- [36] S. Patil and S. Talbar, *Content Based Image Retrieval Using Various Distance Metrics*, pp. 154–161. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
- [37] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, “A compact and efficient image retrieval approach based on border/interior pixel classification,” in *CIKM*, pp. 102–109, 2002.
- [38] M. J. Swain and D. H. Ballard, “Color indexing,” *International Journal on Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [39] R. da S. Torres and A. X. Falcão, “Contour Saliency Descriptors for Effective Image Retrieval and Analysis,” *Image and Vision Computing*, vol. 25, no. 1, pp. 3–13, 2007.
- [40] D. C. G. Pedronette and R. da S. Torres, “Shape retrieval using contour features and distance optimization,” in *VISAPP*, vol. 1, pp. 197 – 202, 2010.
- [41] H. Ling, X. Yang, and L. J. Latecki, “Balancing deformability and discriminability for shape matching,” in *ECCV*, vol. 3, pp. 411–424, 2010.
- [42] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *PAMI*, vol. 24, no. 7, pp. 971–987, 2002.
- [43] V. Kovalev and S. Volmer, “Color co-occurrence descriptors for querying-by-example,” in *ICMM*, p. 32, 1998.
- [44] B. Tao and B. W. Dickinson, “Texture recognition and image retrieval using gradient indexing,” *JVCIR*, vol. 11, no. 3, pp. 327–342, 2000.
- [45] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [46] D. Lowe, “Object recognition from local scale-invariant features,” in *ICCV*, pp. 1150–1157, 1999.
- [47] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [48] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, *BRIEF: Binary Robust Independent Elementary Features*, pp. 778–792. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [49] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International Conference on Computer Vision*, pp. 2564–2571, Nov 2011.

- [50] S. A. Vassou, N. Anagnostopoulos, A. Amanatiadis, K. Christodoulou, and S. A. Chatzichristofis, “Como: A compact composite moment-based descriptor for image retrieval,” in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pp. 30:1–30:5, 2017.
- [51] L. Zheng, Y. Yang, and Q. Tian, “SIFT meets CNN: A decade survey of instance retrieval,” *CoRR*, vol. abs/1608.01807, 2016.
- [52] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
- [53] D. C. G. Pedronette and R. da S. Torres, “Exploiting clustering approaches for image re-ranking,” *Journal of Visual Languages and Computing*, vol. 22, no. 6, pp. 453–466, 2011.
- [54] T. S. Huang and X. S. Zhou, “Image retrieval with relevance feedback: from heuristic weight adjustment to optimal learning methods,” in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 3, pp. 2–5 vol.3, 2001.
- [55] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pp. 107–118, 2001.
- [56] C. D. Ferreira, J. A. dos Santos, R. da S. Torres, M. A. Gonçalves, R. C. Rezende, and W. Fan, “Relevance feedback based on genetic programming for image retrieval,” *Pattern Recognition Letters*, vol. 32, no. 1, pp. 27–37, 2011.
- [57] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox, “A genetic programming framework for content-based image retrieval,” vol. 42, pp. 283 – 292, 2009. Learning Semantics from Multimedia Content.
- [58] J. Papa, A. Falcão, and C. Suzuki, “Supervised pattern classification based on optimum-path forest,” *Int. J. Imaging Syst. Technol.*, vol. 19, pp. 120–131, June 2009.
- [59] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, “Deep learning for content-based image retrieval: A comprehensive study,” in *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pp. 157–166, 2014.
- [60] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, “Improving shape retrieval by learning graph transduction,” in *ECCV*, vol. 4, pp. 788–801, 2008.
- [61] G. Park, Y. Baek, and H.-K. Lee, “Re-ranking algorithm using post-retrieval clustering for content-based image retrieval,” *Information Processing and Management*, vol. 41, no. 2, pp. 177–194, 2005.

- [62] W. Voravuthikunchai, B. Crémilleux, and F. Jurie, “Image re-ranking based on statistics of frequent patterns,” in *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pp. 129:129–129:136, 2014.
- [63] Y. Chen, X. Li, A. Dick, and R. Hill, “Ranking consistency for image matching and object retrieval,” *Pattern Recognition*, vol. 47, no. 3, pp. 1349 – 1360, 2014.
- [64] X. Bai, S. Bai, and X. Wang, “Beyond diffusion process: Neighbor set similarity for fast re-ranking,” *Information Sciences*, vol. 325, pp. 342 – 354, 2015.
- [65] D. C. G. Pedronette and R. d. S. Torres, “A correlation graph approach for unsupervised manifold learning in image retrieval tasks,” *Neurocomputing*, vol. 208, no. Supplement C, pp. 66 – 79, 2016.
- [66] D. C. G. Pedronette, J. Almeida, and R. d. S. Torres, “A graph-based ranked-list model for unsupervised distance learning on shape retrieval,” *Pattern Recognition Letters*, vol. 83, Part 3, pp. 357 – 367, 2016. Efficient Shape Representation, Matching, Ranking, and its Applications.
- [67] L. P. Valem, D. C. G. Pedronette, R. d. S. Torres, E. Borin, and J. Almeida, “Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks,” *ICMR*, 2015.
- [68] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, “Particle competition and cooperation in networks for semi-supervised learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 1686–1698, Sept 2012.
- [69] W. P. Amorim, A. X. Falcão, and M. H. d. Carvalho, “Semi-supervised pattern classification using optimum-path forest,” in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 111–118, Aug 2014.
- [70] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, “Semi-supervised svm batch mode active learning for image retrieval,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, June 2008.
- [71] N. Bhowmik, V. R. González, V. Gouet-Brunet, H. Pedrini, and G. Bloch, “Efficient fusion of multidimensional descriptors for image retrieval,” in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 5766–5770, Oct 2014.
- [72] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, “Early versus late fusion in semantic video analysis,” in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pp. 399–402, 2005.

- [73] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, “Multimodal fusion for multimedia analysis: a survey,” *Multimedia Systems*, vol. 16, pp. 345–379, Nov 2010.
- [74] J. Yu, Z. Qin, T. Wan, and X. Zhang, “Feature integration analysis of bag-of-features model for image retrieval,” *Neurocomputing*, vol. 120, no. Supplement C, pp. 355 – 364, 2013. Image Feature Detection and Description.
- [75] J. Yue, Z. Li, L. Liu, and Z. Fu, “Content-based image retrieval using color and texture fused features,” *Mathematical and Computer Modelling*.
- [76] H. J. Escalante, C. A. Hérnandez, L. E. Sucar, and M. Montes, “Late fusion of heterogeneous methods for multimedia image retrieval,” in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, MIR '08*, pp. 172–179, 2008.
- [77] W. Zhang, Z. Qin, and T. Wan, “Image scene categorization using multi-bag-of-features,” in *2011 International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 1804–1808, July 2011.
- [78] L. Piras, R. Tronci, and G. Giacinto, *Diversity in Ensembles of Codebooks for Visual Concept Detection*, pp. 399–408. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [79] L. Wang, Y. Wang, and Q. Chang, “Feature selection methods for big data bioinformatics: A survey from the search perspective,” *Methods*, vol. 111, no. Supplement C, pp. 21 – 31, 2016. Big Data Bioinformatics.
- [80] B. Ionescu, J. Benois-Pineau, T. Piatrik, and G. Qunot, *Fusion in Computer Vision: Understanding Complex Visual Content*. Springer Publishing Company, Incorporated, 2014.
- [81] J. Yang, J.-y. Yang, D. Zhang, and J.-f. Lu, “Feature fusion: parallel strategy vs. serial strategy,” *Pattern Recognition*, vol. 36, no. 6, pp. 1369 – 1381, 2003.
- [82] H. Young, “An axiomatization of borda’s rule,” *Journal of Economic Theory*.
- [83] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms condorcet and individual rank learning methods,” in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, pp. 758–759, 2009.
- [84] M. Farah and D. Vanderpooten, “An outranking approach for rank aggregation in information retrieval,” in *Proceedings of the 30th Annual International ACM*

- SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 591–598, 2007.
- [85] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top k lists,” in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pp. 28–36, 2003.
- [86] T.-Y. Liu, “Learning to rank for information retrieval,” *Found. Trends Inf. Retr.*, vol. 3, pp. 225–331, Mar. 2009.
- [87] T. Joachims, “Training linear svms in linear time,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 217–226, 2006.
- [88] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pp. 89–96, 2005.
- [89] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, “Learning to rank: From pairwise approach to listwise approach,” in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 129–136, 2007.
- [90] J. Xu and H. Li, “Adarank: A boosting algorithm for information retrieval,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 391–398, 2007.
- [91] S. S. Bucak, R. Jin, and A. K. Jain, “Multiple kernel learning for visual object recognition: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2014.
- [92] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
- [93] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, “Multiple kernel learning, conic duality, and the smo algorithm,” in *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pp. 6–, 2004.
- [94] S. Sonnenburg, G. Rätsch, and C. Schäfer, “A general and efficient multiple kernel learning algorithm,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pp. 1273–1280, 2005.
- [95] M. Kowalski, M. Szafranski, and L. Ralaivola, “Multiple indefinite kernel learning with mixed norm regularization,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 545–552, 2009.

- [96] A. Pinar, T. C. Havens, D. T. Anderson, and L. Hu, “Feature and decision level fusion using multiple kernel learning and fuzzy integrals,” in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, Aug 2015.
- [97] M. Haghigat, M. Abdel-Mottaleb, and W. Alhalabi, “Fully automatic face normalization and single sample face recognition in unconstrained environments,” *Expert Systems with Applications*, vol. 47, no. Supplement C, pp. 23 – 34, 2016.
- [98] M. Haghigat, M. Abdel-Mottaleb, and W. Alhalabi, “Discriminant correlation analysis: Real-time feature level fusion for multimodal biometric recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 1984–1996, Sept 2016.
- [99] F. R. Bach, “Consistency of the group lasso and multiple kernel learning,” *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, June 2008.
- [100] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas, “Query specific rank fusion for image retrieval,” *IEEE TPAMI*, vol. 37, pp. 803–815, April 2015.
- [101] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, “Unsupervised metric fusion by cross diffusion,” in *CVPR*, pp. 3013 –3020, 2012.
- [102] J. Shao, L. Wang, Z. Zhao, F. Su, and A. Cai, “Deep canonical correlation analysis with progressive and hypergraph learning for cross-modal retrieval,” *Neurocomputing*, vol. 214, no. Supplement C, pp. 618 – 628, 2016.
- [103] W. He, X. Cheng, R. Hu, Y. Zhu, and G. Wen, “Feature self-representation based hypergraph unsupervised feature selection via low-rank representation,” *Neurocomput.*, vol. 253, pp. 127–134, Aug. 2017.
- [104] X. Cheng, Y. Zhu, J. Song, G. Wen, and W. He, “A novel low-rank hypergraph feature selection for multi-view classification,” *Neurocomputing*, vol. 253, no. Supplement C, pp. 115 – 121, 2017. Learning Multimodal Data.
- [105] L. Zhang, Y. Gao, C. Hong, Y. Feng, J. Zhu, and D. Cai, “Feature correlation hypergraph: Exploiting high-order potentials for multimodal recognition,” *IEEE Transactions on Cybernetics*, vol. 44, pp. 1408–1419, Aug 2014.
- [106] M. Akbari, L. Nie, and T.-S. Chua, “amm: Towards adaptive ranking of multi-modal documents,” *International Journal of Multimedia Information Retrieval*, vol. 4, pp. 233–245, Dec 2015.
- [107] H. Jegou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *European Conference on Computer Vision, ECCV ’08*, pp. 304–317, 2008.

- [108] M.-E. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1447–1454, 2006.
- [109] L. J. Latecki, R. Lakmper, and U. Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *CVPR*, pp. 424–429, 2000.
- [110] G.-H. Liu and J.-Y. Yang, “Content-based image retrieval using color difference histogram,” *Pattern Recognition*, vol. 46, no. 1, pp. 188 – 198, 2013.
- [111] D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2006)*, vol. 2, pp. 2161–2168, 2006.
- [112] M. Lux, “Content based image retrieval with lire,” in *Proceedings of the 19th ACM International Conference on Multimedia*, MM ’11, pp. 735–738, 2011.
- [113] L. Cieplinski, “Mpeg-7 color descriptors and their applications,” in *Computer Analysis of Images and Patterns* (W. Skarbek, ed.), (Berlin, Heidelberg), pp. 11–20, Springer Berlin Heidelberg, 2001.
- [114] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating color descriptors for object and scene recognition,” *PAMI*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [115] R. Gopalan, P. Turaga, and R. Chellappa, “Articulation-invariant representation of non-planar shapes,” in *ECCV*, vol. 3, pp. 286–299, 2010.
- [116] H. Ling and D. W. Jacobs, “Shape classification using the inner-distance,” *IEEE TPAMI*, vol. 29, no. 2, pp. 286–299, 2007.
- [117] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, “Color and texture descriptors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.
- [118] S. A. Chatzichristofis and Y. S. Boutalis, “Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval,” in *ICVS*, pp. 312–322, 2008.
- [119] S. A. Chatzichristofis and Y. S. Boutalis, “Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval,” in *WIAMIS*, pp. 191–196, 2008.
- [120] K. Zagoris, S. Chatzichristofis, N. Papamarkos, and Y. Boutalis, “Automatic image annotation and retrieval using the joint composite descriptor,” in *PCI*, pp. 143–147, 2010.
- [121] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *IJCV*, vol. 42, pp. 145–175, May 2001.

-
- [122] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. Han, “Contextual weighting for vocabulary tree based image retrieval,” in *ICCV’2011*, pp. 209–216, Nov 2011.
- [123] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [124] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [125] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.
- [126] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, “Dual path networks,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4467–4475, Curran Associates, Inc., 2017.
- [127] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 730–734, 2015.
- [128] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2017.
- [129] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML’15*, pp. 448–456, 2015.
- [130] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” *CoRR*, vol. abs/1707.07012, 2017.
- [131] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS’12*, pp. 1097–1105, 2012.
- [132] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017.

- [133] K. Reddy Mopuri and R. Venkatesh Babu, “Object level deep feature pooling for compact image representation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2015.
- [134] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017.
- [135] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [136] D. C. G. Pedronette and R. da S. Torres, “A correlation graph approach for unsupervised manifold learning in image retrieval tasks,” *Neurocomputing*, vol. 208, no. Sup C, pp. 66 – 79, 2016.
- [137] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, “Query-adaptive late fusion for image search and person re-identification,” in *CVPR*, 2015.
- [138] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” 1999.
- [139] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective,” *arXiv preprint arXiv:1601.07996*, 2016.
- [140] A. Y. Ng, “Feature selection, l1 vs. l2 regularization, and rotational invariance,” in *ICML '04*, 2004.
- [141] N. Jardine and C. van Rijsbergen, “The use of hierarchic clustering in information retrieval,” *Information Storage and Retrieval*, vol. 7, no. 5, pp. 217 – 240, 1971.
- [142] D. C. G. Pedronette, O. A. Penatti, and R. d. S. Torres, “Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks,”
- [143] D. C. G. Pedronette and R. d. S. Torres, “Unsupervised effectiveness estimation for image retrieval using reciprocal rank information,” in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 321–328, 2015.
- [144] C. Y. Okada, D. C. G. Pedronette, and R. da S. Torres, “Unsupervised distance learning by rank correlation measures for image retrieval,” in *ACM International Conference on Multimedia Retrieval (ICMR'2015)*, 2015.
- [145] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Transactions on Information Systems*, vol. 28, no. 4, pp. 20:1–20:38, 2010.

- [146] C. Y. Okada, D. C. G. a. Pedronette, and R. da S. Torres, “Unsupervised distance learning by rank correlation measures for image retrieval,” in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ICMR '15*, (New York, NY, USA), pp. 331–338, ACM, 2015.
- [147] R. Fagin, R. Kumar, and D. Sivakumar, “Comparing top k lists,” in *ACM-SIAM Symposium on Discrete algorithms (SODA'03)*, pp. 28–36, 2003.
- [148] L. P. Valem and D. C. G. a. Pedronette, “Selection and combination of unsupervised learning methods for image retrieval,” in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, CBMI '17*, pp. 27:1–27:6, 2017.
- [149] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, “Accurate image search using the contextual dissimilarity measure,” *PAMI*, vol. 32, no. 1, pp. 2–11, 2010.
- [150] J. Jiang, B. Wang, and Z. Tu, “Unsupervised metric learning by self-smoothing operator,” in *ICCV*, pp. 794–801, 2011.
- [151] S. Bai and X. Bai, “Sparse contextual activation for efficient visual re-ranking,” *IEEE Trans. on Image Processing (TIP)*, vol. 25, no. 3, pp. 1056–1069, 2016.
- [152] S. Bai, S. Sun, X. Bai, Z. Zhang, and Q. Tian, “Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity,” in *European Conference on Computer Vision (ECCV)*, pp. 592–608, 2016.
- [153] D. C. G. Pedronette, F. M. F. Gonçalves, and I. R. Guilherme, “Unsupervised manifold learning through reciprocal kNN graph and Connected Components for image retrieval tasks,” *Pattern Recognition*, 2017. On-line, To appear.
- [154] D. C. G. Pedronette, J. Almeida, and R. da S. Torres, “A scalable re-ranking method for content-based image retrieval,” *Information Sciences*, vol. 265, no. 1, pp. 91–104, 2014.
- [155] D. C. G. Pedronette and R. da S. Torres, “Image re-ranking and rank aggregation based on similarity of ranked lists,” *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.
- [156] M. Donoser and H. Bischof, “Diffusion processes for retrieval revisited,” in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1320–1327, 2013.
- [157] X. Yang, L. Prasad, and L. Latecki, “Affinity learning with diffusion on tensor product graph,” *IEEE TPAMI*, vol. 35, no. 1, pp. 28–38, 2013.
- [158] X. Bai, S. Bai, and X. Wang, “Beyond diffusion process: Neighbor set similarity for fast re-ranking,” *Information Sciences*, vol. 325, pp. 342 – 354, 2015.

- [159] Z. Liu, S. Wang, L. Zheng, and Q. Tian, “Robust imagegraph: Rank-level feature fusion for image search,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3128–3141, 2017.
- [160] G. Tolias, Y. Avrithis, and H. Jégou, “To aggregate or not to aggregate: Selective match kernels for image search,” in *IEEE International Conference on Computer Vision (ICCV’2013)*, pp. 1401–1408, Dec 2013.
- [161] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, “Convolutional patch representations for image retrieval: An unsupervised approach,” *Int. Journal of Computer Vision*, 2017.
- [162] D. Qin, C. Wengert, and L. V. Gool, “Query adaptive similarity for large scale object retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2013)*, pp. 1610–1617, June 2013.
- [163] L. Zheng, S. Wang, and Q. Tian, “Coupled binary embedding for large-scale image retrieval,” *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 8, pp. 3368–3380, 2014.
- [164] S. Sun, Y. Li, W. Zhou, Q. Tian, and H. Li, “Local residual similarity for image re-ranking,” *Information Sciences*, vol. 417, no. Sup. C, pp. 143 – 153, 2017.
- [165] L. Zheng, S. Wang, Z. Liu, and Q. Tian, “Packing and padding: Coupled multi-index for accurate image retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2014)*, pp. 1947–1954, June 2014.
- [166] X. Li, M. Larson, and A. Hanjalic, “Pairwise geometric matching for large-scale object retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2015)*, pp. 5153–5161, June 2015.
- [167] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, “Regularized diffusion process for visual retrieval,” in *Conf. on Artificial Intelligence (AAAI)*, pp. 3967–3973, 2017.
- [168] L. Zheng, S. Wang, and Q. Tian, “Lp-norm idf for scalable image retrieval,” *IEEE TIP*, vol. 23, pp. 3604–3617, Aug 2014.
- [169] L. Xie, R. Hong, B. Zhang, and Q. Tian, “Image classification and retrieval are one,” in *ACM ICMR’2015*, pp. 3–10, 2015.
- [170] J. Almeida, L. P. Valem, and D. C. G. a. Pedronette, “A rank aggregation framework for video interestingness prediction,” in *Proceedings of the 9th International Conference on Image Analysis and Processing, ICIAP ’17*, 2017.

-
- [171] L. P. Valem, D. C. G. Pedronette, F. Breve, and I. R. Guilherme, “Manifold correlation graph for semi-supervised learning,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2018.
- [172] L. P. Valem, C. R. D. Oliveira, D. C. G. a. Pedronette, and J. Almeida, “Unsupervised similarity learning through rank correlation and knn sets,” vol. 14, (New York, NY, USA), pp. 80:1–80:23, ACM, Oct. 2018.
- [173] L. P. Valem and D. C. G. a. Pedronette, “An unsupervised distance learning framework for multimedia retrieval,” in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR '17*, (New York, NY, USA), pp. 107–111, ACM, 2017.

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 15/03/2019

Lucas Rossetti Valem

Assinatura do autor