

**Descriptor de bordas e
quantização espacial flexível
aplicados a categorização de objetos**

Arnaldo Câmara Lara

TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
DOUTOR EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Roberto Hirata Júnior

Durante o desenvolvimento deste trabalho, o autor recebeu auxílio financeiro do CNPq

São Paulo, março de 2013

Descriptor de bordas e quantização espacial flexível aplicados a categorização de objetos

Esta versão da tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 1/03/2013. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Roberto Hirata Júnior (orientador) - IME-USP
- Prof. Dr. Ronaldo Fumio Hashimoto - IME-USP
- Prof. Dr. Carlos Hitoshi Morimoto - IME-USP
- Prof. Dr. Hae Yong Kim - EP-USP
- Prof. Dr. Eduardo Alves do Valle Junior - UNICAMP

Agradecimentos

Gostaria de agradecer a todos aqueles com quem tive oportunidade de conhecer e conviver durante este período de doutorado e também a todos que me ajudaram direta ou indiretamente neste trabalho. Em especial, gostaria de agradecer:

ao Prof. Roberto Hirata Júnior, pela orientação neste trabalho, pela amizade, pelo incentivo, pelos ensinamentos e pelo entusiasmo que demonstra pelo seu trabalho;

aos membros da banca de qualificação, Prof. Dr. Ronaldo Fumio Hashimoto e Prof. Dr. Carlos Hitoshi Morimoto, e aos membros da banca de defesa, Prof. Dr. Ronaldo Fumio Hashimoto, Prof. Dr. Carlos Hitoshi Morimoto, Prof. Dr. Hae Yong Kim e Prof. Dr. Eduardo Alves do Valle Junior, pelas importantes contribuições e sugestões que deram a este trabalho;

aos meus pais, Marília e Francisco, que sempre incentivaram minha educação e apoiaram minhas decisões;

às minhas irmãs, Flávia e Lígia, pela amizade e pelo apoio;

aos demais familiares e amigos que torceram por mim nesta empreitada;

ao CNPQ que me apoio financeiramente durante parte do desenvolvimento deste trabalho;

aos professores do IME que contribuíram na minha formação científica e tornaram este trabalho possível;

aos funcionários do IME, em especial da CPG, pela boa vontade em ajudar sempre;

aos colegas do doutorado pela convivência e troca de experiências;

aos colegas do laboratório de Visão pela amizade, pelos bons momentos juntos e pelo incentivo e sugestões a este trabalho;

aos administradores da Rede Vision pelo empenho e competência em manter os recursos computacionais disponíveis.

Resumo

LARA, A. C. Descritor de bordas e quantização espacial flexível aplicados a categorização de objetos. 2013. 131 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

A área de reconhecimento de objetos tem assistido a um impressionante progresso na última década. O estudo de descritores, aliado à estratégias de amostragem usando quantizações espaciais e a combinação de classificadores têm permeado o estado da arte nos últimos anos. Neste trabalho é proposta uma nova quantização espacial com número arbitrário de níveis e subdivisões arbitrárias de regiões. Regiões adjacentes possuem sobreposição gerando redundância na representação destas regiões de fronteiras e, assim, evitando as quebras que acontecem nas pirâmides espaciais tradicionais que prejudicam a interpretação das formas. Apesar de melhorar o desempenho da abordagem do saco de palavras, as pirâmides espaciais não são robustas a variações na orientação dos objetos na imagem. Foi também proposto neste trabalho, uma divisão espacial utilizando regiões circulares concêntricas que aumentam a robustez a rotação dos objetos na imagem em aproximadamente 80% quando comparada às pirâmides espaciais.

Além das novas divisões espaciais, é proposto neste trabalho um novo descritor baseado na aplicação de granulometria morfológica no mapa de bordas da imagem original. Este descritor foi utilizado na criação de modelos de classes em aplicações de categorização de objetos utilizando uma base de dados pública com resultados superiores aos do melhor descritor baseado em bordas reportado pela literatura.

Todas estas novas técnicas propostas foram utilizadas em um problema desafiador de categorização de objetos de classes muito parecidas. Foi utilizado um subconjunto da base de pássaros Caltech-UCSD Birds-200 2011 com resultados comparáveis aos melhores resultados reportados pela literatura. A abordagem proposta cria uma classificação de dois níveis e utiliza modelos específicos por classe o que é intuitivo, pois cada espécie de pássaro possui características muito sutis que as diferenciam das demais espécies testadas. Vários descritores são utilizados na criação dos modelos de classes e uma combinação de classificadores gera a rotulação final para a amostra. O descritor proposto neste trabalho esteve presente no melhor modelo de 11 das 13 classes testadas e o resultado final obtido pela técnica de categorização proposta é o melhor resultado utilizando a abordagem do saco de palavras.

Palavras-chave: granulometria, quantização espacial, categorização de objetos.

Abstract

LARA, A. C. **Edge-based descriptor and flexible spatial quantization applied to object categorization** 2013. 131 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2013.

The object recognition area has experienced an impressive progress in the last decade. The study of descriptors, together with a sampling strategy using spatial quantization and the combination of classifiers have been presented in the state of art in recent years. This work proposes a new spatial quantizations with an arbitrary number of levels and divisions in each level. Adjacent regions have overlapping areas that generate redundant representation and avoid breakages in the structures that are in their border regions as it happens in the traditional spatial pyramids and impairs the correct interpretation of these structures. Despite spatial pyramids to improve the performance of the bag-of-words approach in object recognition, they are not robust to changes in object orientation in the image. It was also proposed, in this work, a spatial division using concentric circular regions that is almost 80% more robust to rotation of objects when compared to the spatial pyramids using rectangular divisions.

In addition to the new spatial division of the image, it is proposed a new granulometric-based descriptor that it is applied to the map of edges of the original image. This descriptor was used in the building of category's models for object categorization in a public database and showed a better performance than the most used edge-based descriptor reported in literature.

All these new proposed techniques were used in a challenge problem of object categorization of very similar classes. It was used a subset of the public database Caltech-UCSD Birds-200 2011 and the method obtained results compared to the best results reported in the literature. The proposed approach uses a 2-level classification and builds class-specific models that are an intuitive way to model the species of birds as very subtle characteristics differ in each tested class of birds. Many descriptors are used in the building of models of species and a combination of classifiers generates the final label for a tested sample. The descriptor proposed here were presented in 11 of 13 best models of birds' classes. The final result obtained by the proposed object categorization method is the best one using the bag-of-words approach.

Keywords: granulometry, spatial quantization, object categorization.

Sumário

Lista de Abreviaturas	xii
Lista de Símbolos	xiii
Lista de Figuras	xv
Lista de Tabelas	xix
1 Introdução	1
1.1 Objetivos	4
1.2 Contribuições	5
1.3 Publicações	6
1.4 Organização do Trabalho	6
2 Revisão Bibliográfica	9
2.1 Definições Preliminares	9
2.2 Reconhecimento de Objetos	10
2.3 Saco de Palavras	12
2.3.1 Detecção de Pontos	14
2.3.2 Extração dos Descritores Locais	14
2.3.3 Codificação	14
2.3.4 Geração do Dicionário Visual	15
2.3.5 Agrupamento	16
2.3.6 Modelos de Aprendizado	16
2.4 Pirâmides Espaciais	17
2.5 Categorização Refinada	18
2.6 Descritores	22
2.6.1 SIFT	22
2.6.2 PHOW	23
2.6.3 SURF	23
2.6.4 PHOG	23
2.6.5 LBP	24
2.6.6 Histograma RGB	24
2.7 Classificadores	25
2.7.1 Validação Cruzada	25
2.7.2 Combinação de Classificadores	25

2.7.3	Seleção de Características	27
2.8	Bases de Dados de Reconhecimento de Objetos	27
2.9	Medidas	28
2.10	Morfologia Matemática	30
2.10.1	Definições Preliminares	31
2.10.2	Definição de Granulometria Morfológica	33
3	Desenvolvimento	35
3.1	Quantização Espacial Flexível	35
3.2	Descriptor GRABED	38
3.2.1	Descriptor PPE	41
3.2.2	PPD	43
3.2.3	Mapas de Bordas: Canny e a Abordagem Morfológica	44
3.3	Pirâmide de Regiões Circulares Concêntricas	45
4	Categorização Refinada	49
4.1	Etapas	49
4.1.1	Inicialização	49
4.1.2	Geração dos descritores	50
4.1.3	Criação de Modelos	52
4.1.4	Testes	52
4.1.5	Medidas de Desempenho	53
5	Experimentos e Resultados	55
5.1	Ambiente dos Experimentos	55
5.2	Experimentos com a Quantização Espacial Flexível	55
5.2.1	Discussão	57
5.3	Experimentos com o Descriptor GRABED	58
5.3.1	Resultados	59
5.3.2	Discussão	60
5.4	Reconhecimento de Instâncias de Monumentos	62
5.4.1	A Base de Dados de Instâncias de Monumentos	62
5.4.2	Parametrização do GRABED	62
5.4.3	Parametrização do descritor PHOW	64
5.4.4	Experimentos com Descritor Único	65
5.4.5	Combinação de Descritores	66
5.4.6	Discussão	66
5.5	Experimentos das Pirâmides de Regiões Circulares Concêntricas	69
5.5.1	Discussão dos Resultados	71
5.6	Experimentos de Categorização Refinada	72
5.6.1	Base de Dados Caltech-UCSD Birds-200 2011	72
5.6.2	Experimentos	74
5.6.3	Discussão	78

6 Conclusões	83
6.1 Sugestões para Pesquisas Futuras	85
A Detalhes de Codificação e o Arcabouço Orientado a Objetos	87
A.1 Codificação em Matlab	87
A.2 Arcabouço Orientado a Objetos	89
A.2.1 Escolha do Java	89
A.2.2 Comunicação Java-Matlab	90
A.2.3 Implementação do Arcabouço Orientado a Objetos	91
A.2.4 Direções Futuras	93
B Participação no TRECVID 2011	97
B.1 Bases de Dados	97
B.1.1 Sound & Vision	97
B.1.2 BBC Rushes	97
B.2 Revisão Bibliográfica	98
B.3 Submissão	99
B.4 Conclusão	99
Referências Bibliográficas	101
Índice Remissivo	109

Listas de Abreviaturas

Caltech	Instituto de Tecnologia da Califórnia (<i>California Institute of Technology</i>)
UCSD	Universidade da Califórnia em San Diego (<i>University of California, San Diego</i>)
VC	Visão Computacional
VOC	Classes de Objetos Visuais (<i>Visual Object Classes</i>)
SUN Database	Base de Dados de Compreensão de Cena (<i>Scene Understanding Database</i>)
GRABED	Descriptor Baseado em Granulometria Aplicado ao Mapa de Bordas (<i>Granulometry Based Descriptor Applied in Map of Edges</i>)
PHOW	Histograma Piramidal de Palavras Visuais (<i>Pyramidal Histogram of Visual Words</i>)
PPE	Padrões Piramidais de Bordas (<i>Pyramidal Patterns of Edges</i>)
PPD	Padrões Piramidais de Discos (<i>Pyramidal Patterns of Disks</i>)
PRCC	Pirâmides de Regiões Circulares Concêntricas
TREC	Conferência de Recuperação de Textos (<i>Text Retrieval Conference</i>)
TRECVID	Avaliação de Recuperação de Vídeos TREC (<i>TREC Video Retrieval Evaluation</i>)
VISAPP	Conferência Internacional de Visão Computacional Teoria e Aplicações (<i>International Conference on Computer Vision Theory and Applications</i>)
ISMM	Simpósio Internacional de Morfologia Matemática (<i>International Symposium on Mathematical Morphology</i>)
SIBGRAPI	significado antigo: “Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens”; atualmente: “Conferência de Gráficos, Padrões e Imagens” (<i>Conference on Graphics, Patterns and Images</i>)
RGB	Vermelho, Verde, Azul (<i>Red, Green, Blue</i>)
BoW	Saco de Palavras (<i>Bag of Words</i>)
SIFT	Transformada Característica Invariante a Escala (<i>Scale Invariant Feature Transform</i>)
SVM	Máquina de Vetores de Suporte (<i>Support Vector Machine</i>)
DoG	Diferença de Gaussianas (<i>Difference of Gaussians</i>)
SURF	Características Robustas Aceleradas (<i>Speeded Up Robust Features</i>)
PHOG	Histograma Piramidal de Gradientes Orientados (<i>Pyramidal Histogram of Oriented Gradients</i>)
HOG	Histograma de Gradientes Orientados (<i>Histogram of Oriented Gradients</i>)
LBP	Padrões Binários Locais (<i>Local Binary Patterns</i>)

ECOC	Codificação de Saída com Correção de Erro (<i>Error-Correcting Output Coding</i>)
SFS	Seleção Sequencial para Frente (<i>Sequential Forward Selection</i>)
TP	Positivo Verdadeiro (<i>True Positive</i>)
TN	Negativo Verdadeiro (<i>True Negative</i>)
FP	Positivo Falso (<i>False Positive</i>)
FN	Negativo Falso (<i>False Negative</i>)
acc	Acurácia (<i>Accuracy</i>)
prec	Precisão (<i>Precision</i>)
rec	Recuperação (<i>Recall</i>)
AP	Precisão Média (<i>Average Precision</i>)
MAP	Média das Precisões Médias (<i>Mean of Average Precisions</i>)
MM	Morfologia Matemática (<i>Mathematical Morphology</i>)
EE	Elemento Estruturante
PS	Padrão de Espectro (<i>Pattern Spectrum</i>)
SPM	Casamento de Pirâmides Espaciais (<i>Spatial Pyramid Matching</i>)
QEF	Quantização Espacial Flexível
GNU	acrônimo recursivo para “GNU Não é Unix” (<i>GNU’s Not Unix</i>)
GPL	Licença Pública Geral GNU (<i>GNU General Public License</i>)
GME	Gradiente Morfológico Externo
API	Interface de Programação de Aplicativos (<i>Application Programming Interface</i>)
JVM	Máquina Virtual Java (<i>Java Virtual Machine</i>)
JNI	Interface Nativa Java (<i>Java Native Interface</i>)
JAMAL	Ligaçao Java Matlab (<i>Java Matlab Linking</i>)
RMI	Chamada de Método Remoto (<i>Remote Method Invocation</i>)
UML	Linguagem de Modelagem Unificada (<i>Unified Modeling Language</i>)
GB	Gigabytes
MPEG	Grupo de Especialistas de Imagens em Movimento (<i>Moving Picture Experts Group</i>)
BBC	Corporação de Radiodifusão Britânica (<i>British Broadcasting Corporation</i>)
HSV	Matiz, Saturação, Valor (<i>Hue, Saturation, Value</i>)

Listas de Símbolos

- $\dot{+}$ Adição em um intervalo
- $\dot{-}$ Subtração em um intervalo
- \check{b} Reflexão de b
- \oplus Dilatação
- \ominus Erosão
- \circ Abertura
- \bullet Fechamento
- \wedge Ínfimo
- \vee Supremo
- \triangle Reconstrução Inf-Geodésica
- ∇ Reconstrução Sup-Geodésica
- ψ Granulometria
- Ω Distribuição de Tamanhos
- Φ Distribuição Normalizada de Tamanhos

Listas de Figuras

1.1	A figura mostra três diferentes cenários. A primeira linha mostra imagens de praias, a segunda linha mostra imagens de montanhas e a última linha mostra imagens de cidades. Todas as imagens são da base de dados SUN [XHE ⁺ 10].	3
1.2	Exemplo de uma aplicação de detecção de instância, amostra de imagens da base de dados VOC Pascal 2011 com objetos de 20 categorias diferentes e uma amostra de imagem de cada uma das 7 classes de pássaros do gênero <i>Vireo</i> da base de pássaros Caltech-UCSD Birds-200 2011.	8
2.1	Exemplo da abordagem casamento de modelos. O modelo é procurado na cena e o resultado final aparece no ponto vermelho.	11
2.2	Resultado da detecção de objetos utilizando descritores globais, neste exemplo utilizou-se um histograma de níveis de cinza com 15 intervalos.	12
2.3	Imagen representando a abordagem de modelagem baseada em partes. Imagen do trabalho de [FE73].	13
2.4	Imagen do hotel Burj Al Arab da base de instâncias de monumentos utilizada nesta tese com pontos de interesse localizados pelo detector SIFT e ilustração das etapas da abordagem do saco de palavras.	19
2.5	Imagen do Cristo Redentor da base de instâncias de monumentos utilizando uma pirâmide espacial de 2 níveis para composição do descritor final.	20
2.6	Gráficos mostrando os principais parâmetros levados em consideração para calcular o ponto de interesse SIFT e o descritor SIFT. O histograma das orientações com 8 intervalos mostrado na primeira região da Figura 2.6(b) é calculado em cada uma das 16 regiões utilizadas no cômputo do descritor.	24
2.7	Gráfico precisão × recuperação.	30
3.1	A figura mostra imagens de duas classes da base de dados Caltech-101 com divisões espaciais que exploram a geometria dos objetos.	36
3.2	A figura mostra a imagem de um guarda-chuva (classe <i>umbrella</i> da base de dados Caltech-101), o mapa de bordas e duas divisões espaciais. Uma sem utilizar sobreposição de regiões e outras utilizando-a.	38
3.3	Passos para o cálculo do padrão de espectro para uma imagem da classe <i>Faces_easy</i> . A família de EEs usada para calcular o padrão de espectro tem 0° de orientação. . . .	40
3.4	A figura mostra um exemplo da imagem da classe <i>Leopards</i> em 3 escalas diferentes. .	41

3.5 Imagem da base de dados Caltech-UCSD Birds-200 2011 em 3 escalas diferentes, fator de redução 0,7 e seus respectivos mapas de bordas calculados utilizando o detector de bordas Canny.	42
3.6 Aplicação do detector de bordas Canny e do gradiente morfológico externo filtrado por operadores conexos em duas imagens da base de dados Caltech-UCSD Birds-200 2011.	45
3.7 Imagem original da classe <i>BACKGROUND_Google</i> da base de dados Caltech-101 e a imagem rotacionada (196° de rotação). A segunda linha mostra o descritor de cada imagem na abordagem do saco de palavras com 600 palavras visuais. A terceira linha mostra o descritor de ambas as imagens na abordagem do saco de palavras junto com uma quantização espacial de 2 níveis que totaliza 3000 dimensões e a última linha mostra os descritores das imagens usando a PRCC com 2396 dimensões.	47
3.8 A figura mostra a mesma imagem da Figura 3.7. A imagem original da classe <i>BACKGROUND_Google</i> da base de dados Caltech-101 e a imagem rotacionada em 196°. São mostradas 4 regiões da PRCC para cada imagem. A primeira coluna mostra a primeira região circular com raio de 15% da altura da imagem, a próxima coluna mostra a segunda região e assim sucessivamente, até a última coluna que mostra a imagem inteira que é a quarta região utilizada para calcular o descritor final.	48
4.1 A imagem mostra, no canto superior esquerdo, a foto original da base de dados de um pássaro da classe 157 e as fotos seguintes mostram 8 exemplos de imagens geradas pela rotina de geração de imagens de treinamento expandidas.	50
5.1 A imagem mostra as 5 espécies de pássaros utilizadas neste experimento. Começando no canto superior esquerdo, em sentido horário, a ordem é a mesma da Tabela 5.1.	56
5.2 Duas imagens para cada classe da base de dados Caltech-101 utilizadas neste experimento. Da esquerda para a direta, as classes são: <i>BACKGROUND_Google</i> , <i>Faces</i> , <i>Faces_Easy</i> , <i>Leopards</i> e <i>Motorbikes</i>	58
5.3 Um exemplo de cada instância da base de dados de instâncias de monumentos. Da esquerda para a direita, de cima para baixo, as imagens apresentam a mesma sequência apresentada na Tabela 5.10.	63
5.4 O mapa de bordas de uma imagem do Coliseu foi computada utilizando os três métodos testados diferentes: Canny, Prewitt e GME.	64
5.5 Gráficos mostrando a acurácia × número de palavras do dicionário visual, a acurácia × tamanho do conjunto de treinamento, a acurácia × altura das imagens e a acurácia × número de descritores	65
5.6 São mostradas imagens que ilustram a diferença entre a base de dados SUN e a base de dados de instâncias de monumentos. A primeira figura mostra imagens de diferentes igrejas ao redor do mundo, pertencentes à classe de igrejas da base SUN. A segunda figura mostra diferentes imagens da mesma igreja, a Catedral de Colônia, que é rotulada como uma classe única na base de monumentos e a terceira figura mostra diferentes imagens do Templo da Sagrada Família, uma outra igreja que é rotulada como uma classe diferente na base de dados de instâncias de monumentos.	68

5.7 Esta figura apresenta uma amostra de cada classe do conjunto de teste da base de dados Caltech-101. A primeira linha mostra a imagem original e a segunda linha mostra a mesma imagem rotacionada. Da esquerda para direita, as classes são: <i>BACKGROUND_Google, Faces, Faces_Easy, Leopards</i> e <i>Motorbikes</i>	71
5.8 Esta figura mostra a imagem de um albatroz e de um rouxinol da base de dados Caltech-UCSD Birds-200 2011	73
5.9 Esta figura mostra imagens de duas espécies diferentes de pardais da base de dados Caltech-UCSD Birds-200 2011	73
5.10 Esta figura mostra imagens da mesma espécie de albatroz em situações diferentes: em vôo, na água e na terra.	73
5.11 Esta figura mostra as 6 espécies de pica-paus utilizados neste experimento. Juntamente com as 7 espécies de pássaros do gênero <i>Vireo</i> , perfazem o total de 13 espécies utilizadas nos experimentos de categorização refinada.	75
5.12 Pássaros da classe 151 da base de pássaros com o bico marcado com um quadrado vermelho.	75
A.1 Diagrama de classes mostrando a classe abstrata <i>AplicRecObjetos</i> e suas classes derivadas.	91
A.2 Diagrama de classes mostrando as classes de bases de dados.	92
A.3 Diagrama de classes mostrando as classes de relacionadas com as imagens.	93
A.4 Diagrama de classes mostrando as classes de descritores.	94
A.5 Diagrama de classes mostrando as classes de classificadores.	95
A.6 Diagrama de classes mostrando as classes de utilitários.	95
A.7 Diagrama de classes mostrando as classes de resultados.	96
B.1 Exemplos de quadros das duas bases de dados utilizadas na tarefa busca por instâncias da edição 2011 do desafio TRECVID.	98

Listas de Tabelas

2.1	Exemplo de uma tabela ECOC	27
5.1	Espécies de pássaros utilizadas no experimento de comparação da QEF e SPM.	56
5.2	Resultados da comparação entre a SPM e a QEF em 5 classes da base de dados Caltech-UCSD Birds-200 2011.	57
5.3	Classes da base de dados Caltech-101 utilizadas neste experimento e o respectivo número de imagens por classe	58
5.4	Médias das medidas para as 5 classes utilizando o descritor GRABED básico (sem o uso de múltiplas escalas e sem o uso de quantização espacial)	59
5.5	Média das medidas das 5 classes para o descritor GRABED utilizando múltiplas escalas	60
5.6	Média das medidas para as 5 classes utilizando múltiplas escalas e as QEFs propostas neste trabalho.	60
5.7	Resultados do descritor GRABED para cada uma das classes testadas utilizando a parametrização do melhor resultado médio apresentado na Tabela 5.6.	61
5.8	Resultados do descritor GRABED com a melhor parametrização para cada uma das 5 classes.	61
5.9	Resultados comparativos entre o descritor GRABED e os outros descritores testados: PHOG, PHOW, SURF and SIFT. A tabela apresenta a média das 5 classes testadas.	61
5.10	Base de dados de instâncias de monumentos.	62
5.11	Acurácia do descritor GRABED usando diferentes métodos para computar as bordas usando canais RGB ou imagens em níveis de cinza.	64
5.12	Experimentos utilizando o descritor PHOW em categorização de objetos na base de dados Caltech-101.	66
5.13	Acurácia dos descritores no problema de categorização na base de instâncias de monumentos.	66
5.14	Acurácia da classificação utilizando combinação dos descritores PHOW, GRABED e histograma de cores AB.	67
5.15	Matriz de confusão para a classificação utilizando a combinação de descritores.	67
5.16	Matriz de confusão da abordagem saco de palavras.	69
5.17	Matriz de confusão do experimento usando saco de palavras e testado em imagens aleatoriamente rotacionadas.	70
5.18	Matriz de confusão usando a abordagem do saco de palavras adicionada à quantização espacial.	70

5.19 Matriz de confusão do experimento utilizando o saco de palavras acrescido da quantização espacial com imagens de teste rotacionadas.	70
5.20 Matriz de confusão da abordagem PRCC com imagens de teste não rotacionadas. . .	71
5.21 Matriz de confusão da abordagem PRCC com imagens de teste rotacionadas.	71
5.22 Espécies de pássaros utilizadas nos experimentos de categorização refinada deste trabalho.	76
5.23 A precisão média (AP) para cada classe testada e a configuração utilizada.	76
5.24 Parâmetros da Configuração 1.	77
5.25 Parâmetros da Configuração 2.	77
5.26 Parâmetros da Configuração 3.	78
5.27 Parâmetros da Configuração 4.	78
5.28 Parâmetros da Configuração 5.	79
5.29 Parâmetros da Configuração 6.	79
5.30 Parâmetros da Configuração 7.	79
5.31 Resultados de abordagens de categorização refinada utilizando o mesmo subconjunto da base de pássaros Caltech-UCSD Birds-200 2011 reportados na literatura.	80
5.32 Tempo de processamento em cada etapa do processo da aplicação de categorização refinada de pássaros	80
5.33 Resultados da validação cruzada de modelos criados a partir da localização das partes utilizando a PRCC, junto com dicionário visual de 600 palavras construídos com descritores PHOW colorido.	80

Capítulo 1

Introdução

Reconhecimento de objetos é uma área da Visão Computacional (VC) que estuda problemas como a detecção de objetos de uma determinada classe em uma imagem ou sequência de vídeo ou, ainda, a rotulação da imagem de um objeto como pertencente a uma classe de objetos dentre um conjunto de classes conhecidas. É uma área bastante ativa de pesquisa em VC [Tre10]. De todos os problemas em VC, é também um dos mais complexos. Mesmo os algoritmos no estado da arte não conseguem reconhecer todos os objetos presentes, em todos os tipos de cena e em todas as condições possíveis. Os objetos podem estar presentes nas cenas em diferentes poses, escalas, tamanhos, em diferentes posições na imagem, em rotações diversas ou oclusos parcialmente. Alterações na iluminação afetam a aparência dos objetos. Enfim, são inúmeras as situações que influenciam o aspecto dos objetos produzindo uma enorme variabilidade em suas aparências e dificultando a tarefa de identificá-los [Sze11].

Existe uma grande variedade de aplicações reais que utilizam técnicas de reconhecimento de objetos. Pode-se citar como exemplos [Tre10]:

- **Medição:** algumas vezes, num ambiente industrial, é necessário medir a posição precisa de um objeto em relação a outros com diversos intuiros como o de transportar, processar, agarrar através de um braço mecânico, entre outras atividades.
- **Inspeção:** algoritmos de reconhecimento de objetos podem ser utilizados para inspeção visual com o intuito de controle de qualidade identificando anomalias de peças em uma esteira de montagem, por exemplo.
- **Ordenação:** peças recém-produzidas podem ser classificadas de acordo com seu tamanho através de um algoritmo de reconhecimento de objetos.
- **Contagem:** a contagem de peças, pessoas, animais, objetos pode ser efetuada automaticamente através de uma aplicação de VC.
- **Detecção:** a identificação da presença, em uma cena, de um objeto conhecido.
- **Categorização:** a classificação de um objeto presente em uma imagem como pertencente a uma categoria de objetos conhecidos.
- **Recuperação de imagens:** uma imagem é dada como exemplo e um conjunto de imagens com características semelhantes é retornado.

Diferentes tipos de algoritmos são utilizados para resolver diferentes tipos de aplicações de reconhecimento de objetos. Os principais tipos de algoritmos da área de reconhecimento de objetos são [Sze11]:

- **Detecção de objetos:** o algoritmo possui uma fase de treinamento quando um conjunto de imagens da categoria do objeto em questão é utilizado para criar um modelo desta categoria. Durante a fase de teste, a cada imagem, o algoritmo retorna a probabilidade de um objeto desta

categoria estar ou não presente. Exemplo deste tipo de algoritmo é a detecção de faces, comum, ultimamente, em câmeras digitais pessoais [YAK00]. Este é um problema de classificação binária.

- **Detecção de instâncias:** consiste em localizar em uma imagem a instância de um objeto previamente conhecido. As imagens geralmente apresentam um fundo confuso ou os objetos apresentam-se parcialmente oclusos. Os algoritmos retornam, em sua maioria, a localização mais provável da instância do objeto buscado e também a estimativa de pose do objeto. [LF06] mostra um exemplo de algoritmo deste tipo. Algoritmos de reconhecimento de faces podem ser vistos como um caso especial de detecção de instâncias [LJ11]. Este também é um problema de classificação binária visto que é criado um modelo da instância do objeto buscado e este modelo é utilizado nas imagens de teste para verificar a presença ou não da instância.
- **Localização de objetos:** este é um passo além da detecção de objetos. O algoritmo retorna, além da probabilidade da presença do objeto na imagem de teste, a possível localização deste objeto naquela imagem. O desafio VOC Pascal possui como um dos problemas da competição, a localização de objetos [EGW⁺10].
- **Categorização de objetos:** é uma simplificação do problema mais geral de identificar todos os objetos presentes em uma imagem. É um problema em um mundo fechado. A categorização trata de objetos pertencentes a apenas um conjunto conhecido de categorias e, geralmente, as imagens de treinamento e teste possuem apenas um único objeto. O algoritmo recebe como entrada um conjunto de imagens para treinamento com um número finito de rótulos e um modelo é criado para cada categoria. O algoritmo retorna para uma imagem de teste uma lista de categorias e as probabilidades da imagem conter um objeto das categorias listadas. [CDF⁺04] propõe uma abordagem para solucionar este tipo de problema, que é um problema de classificação multiclass. São criados modelos para cada uma das categorias testados no algoritmo. A categorização pode ser implementada combinando N algoritmos de detecção de objetos, onde N é o tamanho do conjunto de categorias.
- **Identificação de cenas:** o contexto de um objeto tem um papel fundamental na sua identificação. Algoritmos de identificação de cena focam em identificar em qual cena o objeto está e isto poderá ser utilizado posteriormente como mais um elemento na identificação do objeto unindo-se às características como formas, texturas ou cores. [TMFR03] propõe o uso do contexto para o reconhecimento de objetos. Pode ser visto como um caso particular da categorização de objetos, onde cada objeto seria um cenário diferente. Figura 1.1 mostra imagens de exemplos de 3 cenários diferentes: praia, montanha e cidade. As imagens são da base de dados SUN [XHE⁺10], especializada em cenários.
- **Detecção de partes:** existem uma gama de algoritmos que tenta reconhecer um objeto a partir da junção de vários algoritmos treinados para detectar partes de um todo. Um exemplo é a detecção de faces a partir de algoritmos de detecção de olhos, nariz e boca e utilizando a relação espacial esperada entre estes componentes determina-se a probabilidade de existir uma face nas imagens inspecionadas. O trabalho de [FGM10] utiliza esta abordagem de detecção de partes para obter a detecção do objeto.

O reconhecimento de objetos é uma tarefa extremamente difícil e até hoje, algoritmos não atingiram o nível de acerto de uma criança de 2 anos de idade [Sze11]. Os algoritmos de categorização de objetos que mapeiam as possibilidades de presença de categorias de objetos em uma coleção fechada conseguem obter resultados mais satisfatórios. Mesmo assim, estão longe de obter um desempenho semelhante ao de um ser humano. Um adulto consegue distinguir cerca de 30000 categorias diferentes de objetos [ZBMM06]. [DBLFF10] mostrou que para 10000 categorias de imagens a melhor acurácia até hoje é menor que 5%.



Figura 1.1: A figura mostra três diferentes cenários. A primeira linha mostra imagens de praias, a segunda linha mostra imagens de montanhas e a última linha mostra imagens de cidades. Todas as imagens são da base de dados SUN [XHE⁺ 10].

Nas últimas décadas, tem havido um esforço grande da comunidade de VC no desenvolvimento de novos algoritmos de reconhecimento de objetos cada vez mais eficientes. Porém o esforço tem se concentrado quase que exclusivamente no reconhecimento de categorias de objetos ou, na outra ponta, no reconhecimento de instâncias [FOZ⁺ 11]. Muito recentemente, a comunidade tem começado a tratar o problema de categorização de objetos de classes muito parecidas, a chamada *categorização refinada* ou *de categorias subordinadas*.

A Figura 1.2 mostra um exemplo das aplicações de detecção de instância, categorização de objetos e da categorização refinada. A Figura 1.2(a) mostra um exemplo de aplicação de detecção de instância. O modelo do objeto a ser encontrado na sequência de vídeo é conhecido e aparece na parte direita da imagem. Na parte esquerda, é mostrado um quadro de uma sequência de vídeo onde o objeto em questão é buscado. Um círculo mostra a provável localização do objeto buscado no quadro. Segundo [Sze11], o problema de detecção de instâncias é mais fácil que o problema de detecção ou de categorização de objetos, visto que se conhece exatamente a instância do objeto a ser procurado. Não é necessário nenhum grau de generalização. Este problema foi, então, tratado mais cedo e existe uma série de abordagens bem sucedidas. Uma delas utiliza descritores invariantes a transformações, que tratam este problema com um alto grau de acurácia. A Figura 1.2(b) mostra uma imagem para cada uma das 20 categorias de objetos existentes na base de dados de reconhecimento de objetos do desafio *Visual Object Classes* (VOC) Pascal [EGW⁺ 10]. Da esquerda para a direita, de cima para baixo as classes mostradas na figura são: avião, bicicleta, gato, sofá, cadeira, pessoa, vaca, cachorro, trem, pássaro, TV/monitor, ônibus, bote, ovelha, cavalo, carro, mesa de jantar, garrafa, motocicleta e vaso de planta. O desafio VOC Pascal é uma competição anual com uma base de dados diferente a cada ano. A versão 2011 apresentou aproximadamente 11500 imagens rotuladas em 20 classes. Ela consiste de imagens difíceis e vem se tornando a base de dados padrão para testar técnicas de reconhecimento e categorização de objetos. O mais desafiador nesta base de

dados é que são imagens reais recolhidas do sítio de compartilhamento de imagens Flickr¹ existindo então grande variabilidade da aparência dos objetos dentro de uma mesma categoria. Existem ainda oclusões parciais, variabilidade de iluminação, de poses, probabilidade de ocorrência de objetos de mais de uma categoria em uma mesma imagem e de mais de um objeto da mesma categoria em uma imagem. A Figura 1.2(c) mostra uma imagem de cada uma das 7 classes das espécies de pássaros do gênero *Vireo*. São categorias diferentes da base de pássaros Caltech-UCSD Birds-200 2011 [WBW⁺11]. A base de pássaros Caltech-UCSD Birds-200 2011 é uma das principais bases de dados que focam a categorização refinada e possui 200 espécies diferentes de pássaros. Como se pode notar, a categorização refinada é um problema bem mais difícil que a categorização de objetos tradicional.

Para compararmos o nível de dificuldade de um problema de categorização de objetos (Figura 1.2(b)) e um problema de categorização refinada (Figura 1.2(c)) pode-se dizer que praticamente qualquer pessoa consegue classificar os objetos presentes nas imagens em uma base de dados de categorização de objetos. Uma criança consegue identificar com sucesso um cachorro, uma mesa, um carro ou um trem. Entretanto apenas um especialista ou alguém que passou por um treinamento específico conseguiria classificar corretamente as espécies de pássaros da base de dados de categorização refinada mostradas na imagem.

A categorização refinada é um dos problemas mais difíceis na área de reconhecimento de objetos e só passou a ser tratada sistematicamente muito recentemente, de 2010 para cá, devido, principalmente, à introdução da base de pássaros Caltech-UCSD Birds-200 [WBM⁺10] que é a primeira versão da base de dados que foi utilizada neste trabalho, a base Caltech-UCSD Birds-200 2011.

Este trabalho propôs uma forma mais flexível de quantização espacial com ganho de desempenho em relação às pirâmides espaciais tradicionais. As pirâmides espaciais possuem uma grande desvantagem: são invariantes a rotação. Uma outra divisão espacial que centrada em um ponto qualquer da imagem, regiões circulares concêntricas são utilizadas para dividir espacialmente a imagem mantendo o ganho de desempenho da quantização espacial tradicional e, ainda, aumentando a robustez a rotação dos objetos na imagem. Três novos descritores são propostos: um deles utiliza granulometria morfológica aplicada ao mapa de bordas da imagem; o outro cria um histograma de padrões 3×3 que aparecem no mapa de bordas e o último transforma uma imagem em níveis de cinza em várias imagens binárias, aplicam-se aberturas morfológicas utilizando elementos estruturante circulares com raios crescentes. Todas estas propostas foram utilizadas em um novo método que trata o problema desafiador de categorização refinada de objetos. A técnica foi testada na principal base de dados pública que foca o problema da categorização refinada [WBW⁺11]. Esta base é utilizada como referência na comparação de técnicas de categorização refinada. O método de categorização refinada proposto utiliza uma classificação em 2 níveis e modelos específicos por classe. O subconjunto da base de dados utilizada neste experimento possui 13 espécies diferentes de pássaros que podem ser claramente divididos em 2 grupos: espécies do gênero *Vireo* e espécies de pica-paus. Portanto a classificação em 2 níveis explora este agrupamento das espécies. A ideia de criar modelos com descritores e parametrizações diferentes para cada espécie é intuitiva pelo fato de cada espécie de pássaros possuir características muito sutis que as diferenciam das demais espécies testadas. O objetivo deste trabalho é utilizar as técnicas propostas aqui em um problema realmente desafiador e melhorar o desempenho da abordagem do saco de palavras em aplicações de categorização refinada.

1.1 Objetivos

Este trabalho tem como objetivo principal melhorar o desempenho da abordagem do saco de palavras no problema de categorização refinada de espécies de pássaros. Utilizamos um subconjunto da base de pássaros Caltech-UCSD Birds-200 2011 [WBW⁺11]. Várias trabalhos reportaram seus resultados utilizando este mesmo subconjunto. Para atingir este objetivo, foram utilizados uma nova proposta de quantização espacial e um novo descritor de bordas.

¹ www.flickr.com

Como objetivo específico, este trabalho se propõe a utilizar uma nova quantização espacial que possua desempenho melhor que as tradicionais pirâmides espaciais propostas em [LSP06].

Este trabalho apresenta, ainda como objetivo específico, um novo descritor de bordas que possua desempenho superior ao descritor de bordas mais utilizado atualmente [BZM07b] em tarefas de categorização de objetos.

Outro objetivo específico é a proposta de uma nova quantização espacial que seja mais robusta a rotação de objetos quando comparada às quantizações espaciais tradicionais.

1.2 Contribuições

As principais contribuições deste trabalho são:

- Foi proposta a eliminação de algumas restrições na divisão espacial das tradicionais pirâmides espaciais propostas no artigo original de [LSP06] gerando uma nova quantização espacial. Foi adicionado também um parâmetro que controla a sobreposição de regiões adjacentes em um mesmo nível. Esta sobreposição ajuda na caracterização de estruturas discriminantes, importantes nas fronteiras de regiões adjacentes de um nível das quantizações espaciais criando um modelo de categoria de objetos mais efetivo. Foi feita uma comparação entre a proposta original e as modificações que este trabalho propôs utilizando para testes um subconjunto de uma base de dados de categorização de objetos. A modificação proposta, neste trabalho, teve uma acurácia 6% melhor que a proposta original nos testes efetuados.
- Foi proposto um descritor baseado na aplicação de granulometria morfológica no mapa de bordas da imagem, o GRABED. Este descritor é utilizado para caracterizar as extensões e orientações das bordas e construir um modelo genérico de objetos para ser utilizado em aplicações de reconhecimento de objetos. A granulometria morfológica é utilizada de uma maneira diferente quando comparada ao uso clássico da granulometria. A granulometria morfológica é geralmente aplicada em imagens em níveis de cinza ou binárias e é utilizada para caracterizar a distribuição de formas e tamanhos dos componentes conexos destas imagens. Foi proposto também um par de extensões: o uso de diferentes escalas da imagem e o uso em conjunto com a quantização espacial flexível. Melhorias no desempenho foram obtidas utilizando as duas extensões. O descritor foi testado em uma base de dados de reconhecimento de objetos e foram obtidos resultados similares aos dois melhores descritores utilizados nestas aplicações. O GRABED, por caracterizar as bordas das imagens de uma maneira diferente, mostrou-se então um forte candidato a entrar num modelo de objetos e contribuir para sua robustez. No geral, o descritor GRABED mostrou melhores resultados que o principal descritor baseado em bordas, segundo a literatura, nas bases de dados testadas. O descritor GRABED foi também o descritor que apareceu em 11 dos 13 melhores modelos de classes da aplicação de categorização refinada que foi desenvolvida neste trabalho. Foi o descritor que mais apareceu entre os melhores modelos de classes juntamente com o PHOW que é descrito como o melhor descritor para aplicações de reconhecimento de objetos [WYY⁺10].
- Foi proposto um outro descritor, chamado de PPE, que é aplicado também ao mapa de bordas de uma imagem e caracteriza todas as combinações de padrões 3×3 pixels presentes neste mapa de bordas. Um histograma destes padrões caracteriza o mapa de bordas e a imagem.
- O último descritor proposto é o PPD que converte imagens em níveis de cinza em imagens binárias e utiliza uma sequência de aberturas morfológicas utilizando elementos estruturantes circulares com raios crescentes.
- A Pirâmide de Regiões Circulares Concêntricas é uma nova maneira de particionar uma imagem formando uma pirâmide espacial a partir de regiões circulares concêntricas. A abordagem saco de palavras com um dicionário visual é utilizada e um histograma normalizado de frequência das palavras visuais é calculado em cada uma das regiões circulares. O descritor final é formado

pela concatenação dos histogramas de cada uma das regiões circulares mais o histograma da imagem inteira. Esta nova maneira de particionar a imagem e calcular o histograma das palavras visuais se mostrou tão eficiente quanto as pirâmides espaciais tradicionais, quando aplicada em imagens não rotacionadas e apresentou uma acurácia quase 80% maior nos testes realizados, quando o conjunto de imagens de teste apresenta rotação dos objetos presentes.

- Todas as técnicas anteriores são utilizadas em um novo método de categorização refinada que utiliza uma classificação em 2 níveis e modelos específicos por classe para categorizar espécies de pássaros muito similares. Foi utilizado um subconjunto de uma base pública de espécies de pássaros. Os resultados deste método são comparáveis aos melhores resultados reportados pela literatura e mostrou o melhor resultado usando a abordagem do saco de palavras.
- Um arcabouço orientado a objetos para aplicações de reconhecimento de objetos com o intuito de aumentar a reutilização de código e facilitar o desenvolvimento de novas funcionalidades e o teste de novas configurações.

1.3 Publicações

Esta seção lista os artigos publicados e aqueles que planejamos publicar, fruto desta tese:

- Em [LH11b], foi publicado um artigo completo que descreve a utilização da combinação de descritores na categorização de instâncias de monumentos. A parte teórica do artigo está descrita na Seção 3.2 e os experimentos citados neste artigo estão descritos na Seção 5.4.
- Em [LH11a], foi publicado um resumo descrevendo a nossa participação no TRECVID 2011. Detalhes sobre esta participação estão no Apêndice B.
- O artigo [LH13b] descrevendo as Pirâmides de Regiões Circulares Concêntricas (Seção 3.3) e seus experimentos descritos na Seção 5.5 foram apresentados no VISAPP 2013², *International Conference on Computer Vision Theory and Applications*, que ocorreu em Barcelona, Espanha, entre os dias 21 e 24 de fevereiro de 2013.
- O artigo [LH13a] descrevendo o descritor GRABED (Seção 3.2), a quantização espacial flexível (Seção 3.1) e a aplicação de ambos em um problema de categorização refinada de objetos (Capítulo 4) foi apresentado no ISMM 2013³, *11th. International Symposium on Mathematical Morphology*, que ocorreu entre 27 e 29 de maio de 2013 em Uppsala, Suécia.
- Planejamos submeter a uma revista da área, em 2013, uma versão extendida do artigo [LH13a] descrevendo o método de categorização refinada proposto, os descritores morfológicos utilizados e a estratégia de quantização espacial utilizada.
- Planejamos submeter um artigo descrevendo o arcabouço orientado a objetos para aplicações de reconhecimento de objetos (Apêndice A) no próximo ano.

1.4 Organização do Trabalho

Este capítulo introduz o problema da categorização refinada, situando o problema dentro da área mais abrangente de reconhecimento de objetos. Os objetivos e as contribuições decorrentes deste trabalho também estão listados neste capítulo. O próximo capítulo introduz conceitos importantes para o entendimento das técnicas propostas, lista artigos importantes relacionados aos conceitos discutidos relacionado-os com a presente tese. São discutidos o reconhecimento de objetos, o saco de palavras, as pirâmides espaciais, a categorização refinada, introduz-se a Morfologia Matemática,

²<http://www.visapp.visigrapp.org/>

³<http://ismm.cb.uu.se>

descreve os principais descritores utilizados na área de reconhecimento de objetos, classificador utilizado, lista as principais bases de dados de reconhecimento de objetos e discute as medidas utilizadas na mensuração de desempenho dos algoritmos de reconhecimento de objetos. O Capítulo 3 descreve os novos descritores, a nova quantização espacial proposta e a nova partição das pirâmides espaciais e sua robustez a presença de rotação dos objetos na imagem. O Capítulo 4 descreve a técnica de categorização refinada proposta neste trabalho. No Capítulo 5, são descritos alguns dos experimentos realizados e seus resultados e apresenta-se uma análise desses resultados. O Capítulo 6 discute as contribuições e dá direções para trabalhos futuros. O Apêndice A descreve alguns detalhes de implementação relativos à preocupação em reutilização de código e o arcabouço orientado a objetos construído com esta mesma preocupação. O Apêndice B descreve a nossa participação no TRECVID 2011.



(a) Detecção de instância



(b) Categorização de objetos



(c) Categorização refinada

Figura 1.2: Exemplo de uma aplicação de detecção de instância, amostra de imagens da base de dados VOC Pascal 2011 com objetos de 20 categorias diferentes e uma amostra de imagem de cada uma das 7 classes de pássaros do gênero Vireo da base de pássaros Caltech-UCSD Birds-200 2011.

Capítulo 2

Revisão Bibliográfica

Este capítulo introduz conceitos fundamentais e descreve técnicas usadas nos capítulos posteriores. A Seção 2.1 faz algumas definições preliminares. A Seção 2.2 descreve a área de reconhecimento de objetos e as principais abordagens utilizadas para tratar o problema. A Seção 2.3 introduz a abordagem do saco de palavras e o conceito do dicionário visual. A Seção 2.4 define as pirâmides espaciais. A Seção 2.5 descreve o recente problema da categorização refinada. Os principais descritores utilizados em aplicações de reconhecimento de objetos são descritos na Seção 2.6. A Seção 2.7 descreve o algoritmo de classificação utilizado e os conceitos de combinação de classificadores, seleção de características e validação cruzada. A Seção 2.8 descreve as principais bases de dados de reconhecimento de objetos. A Seção 2.9 descreve as principais medidas utilizadas para avaliar as aplicações de reconhecimento de objetos e a Seção 2.10 introduz a Morfologia Matemática e conceitua a granulometria morfológica utilizada em descritores propostos no Capítulo 3.

2.1 Definições Preliminares

Imagen digital: pode ser definida como uma função, $f(x, y)$, onde x e y são coordenadas espaciais e a amplitude de f em um ponto (x, y) é chamada de *nível de cinza*. Quando os valores de x , y e o nível de cinza são todos finitos e discretos, dá-se o nome de imagem digital à função f [GW02].

De uma maneira mais formal, seja \mathbb{Z} o conjunto dos números inteiros, seja \mathbb{E} um subconjunto de \mathbb{Z}^2 , seja K o intervalo $[0, k - 1] \in \mathbb{Z}$, com $k > 1$, o produto cartesiano de K é denotado por K^d , onde d é a quantidade de vezes que K aparece no produto. Assim, podemos definir uma imagem digital como sendo a função de \mathbb{E} em K e o conjunto de todas as imagens digitais possíveis é denotado por $K^{\mathbb{E}}$ [Hei95]. O valor de k é o número de níveis de cinza que a imagem digital possui. Geralmente, as imagens digitais são denotadas por letras do alfabeto romano como f ou I .

Pixel: é cada um dos elementos espaciais finitos que compõem uma imagem digital [GW02].

Imagen em níveis de cinza: a definição de *imagen digital* se encaixa no conceito de imagem digital em níveis de cinza quando o valor de $k > 2$. Geralmente, $k = 256$.

Imagen binária: uma imagem digital é dita binária se sua amplitude (ou níveis de cinza) apresentar apenas dois valores possíveis, geralmente 0 e 1. Ou seja, o tamanho do contradomínio na definição de imagem digital vale 2.

Imagen multibandas: é uma imagem digital que possui mais de um valor para cada pixel. Cada valor do pixel está associado a uma *banda* da imagem.

Imagen colorida: é uma imagem multibanda cujas bandas estão associadas a um modelo de cor.

Modelo de cor: um sistema de coordenadas tridimensional onde cada cor é representada por um único ponto [GW02].

RGB: o modelo de cor mais popular, baseado num sistema de coordenadas de três eixos definindo um subespaço em forma de cubo. Os eixos representam a intensidade de vermelho, verde e azul [GW02].

2.2 Reconhecimento de Objetos

Reconhecimento de objetos é uma das áreas mais ativas de pesquisa em VC [Tre10]. Em linhas gerais, uma aplicação de reconhecimento de objetos recebe um conjunto de imagens contendo objetos da categoria de interesse durante uma etapa inicial. Ele é chamado conjunto positivo. Um outro conjunto, chamado de negativo, contém imagens de objetos de outras categorias diferentes da categoria de interesse. Um modelo visual desta categoria de interesse é criado na etapa de treinamento. Este modelo é utilizado para tratar a possibilidade de existir um objeto desta mesma categoria em uma imagem durante a etapa de teste. Várias abordagens diferentes foram utilizadas na tentativa de solucionar este problema. O *casamento de modelos* foi a primeira abordagem a tratar este problema ainda na década de 70. Um modelo do objeto é construído usando, geralmente, uma imagem contendo apenas o objeto de interesse. Um casamento pixel a pixel entre a cena e um modelo do objeto é feita tentando detectar o objeto na imagem de teste. Esta abordagem se mostrou inadequada para uma análise tridimensional da cena, principalmente devido a alterações na visada da cena, oclusões e articulações ou deformações das partes do objeto de interesse [NB77]. Uma alternativa a esta abordagem utilizava descritores globais, como um histograma de cor, para representar o objeto e a cena e uma medida de similaridade era aplicada na imagem de teste, na tentativa de localizar o objeto na cena [NBE⁺93]. Novamente, a abordagem não era robusta o suficiente para tratar cenários reais.

A Figura 2.1 mostra uma aplicação utilizando o casamento de modelos em um exemplo real. A Figura 2.1(a) mostra o modelo do objeto a ser procurado na cena. O modelo é uma imagem contendo apenas o objeto de referência. A Figura 2.1(b) mostra a cena contendo uma instância do modelo. A Figura 2.1(c) mostra o mapa de casamentos entre o modelo e a cena e a Figura 2.1(d), a localização do ponto de maior coeficiente de casamento entre modelo e a cena indicando o centro do objeto. Nota-se que, em um exemplo bem comportado como este, o resultado final é muito bom. Vários tamanhos do modelo são utilizados nos casamentos entre a cena e o modelo. Quanto mais próximo o modelo estiver do tamanho do objeto procurado na cena real, maior será o coeficiente de casamento. A detecção do objeto na cena será positiva se, no mapa de casamento entre o objeto e a cena, houver pontos de máximo maior que um limiar pré-estabelecido.

A Figura 2.2 mostra a detecção e localização do objeto na cena utilizando um histograma de níveis de cinza com 15 intervalos. Foram utilizados o mesmo objeto e cena da Figura 2.1. Novamente, o resultado apresentado mostrou-se muito bom.

Um modo intuitivo de modelar um objeto é criar um modelo baseado em suas partes. A modelagem baseada em partes combina descritores locais e uma representação das relações espaciais entre elas [FE73]. Esta abordagem apresentava grande complexidade computacional e dificuldade para modelar objetos articulados. Outra dificuldade é o custo do processo de treinamento, pois as partes precisam estar rotuladas na base de dados [ZBMM06]. A Figura 2.3 mostra uma representação desta abordagem no trabalho de [FE73]. A primeira abordagem com bom desempenho e computacionalmente eficiente é a abordagem do saco de palavras. Esta abordagem era utilizada originalmente para categorização de textos e foi adaptada para a utilização em categorização de objetos por [CDF⁺04] e [SZ03].

[Tre10] lista algumas restrições e requisições que podem ser impostas a uma aplicação de reconhecimento de objetos:

- *Tempo de avaliação*: aplicações de reconhecimento de objetos, principalmente no ambiente industrial, podem ter que processar os dados de entrada em tempo-real. Aplicações de visão para colocação de circuitos integrados em placas têm que determinar a posição exata do componente em menos 50 ms, para garantir uma alta taxa de produtividade. Também espera-se de aplicações de vigilância eletrônica a detecção de um evento em tempo-real.
- *Acurácia na determinação da posição do objeto*: aplicações de visão para colocação de circuitos integrados requerem um altíssimo grau de acurácia na determinação da posição dos objetos que eles detectam na câmera. A acurácia neste posicionamento pode chegar à ordem de um décimo de pixel.

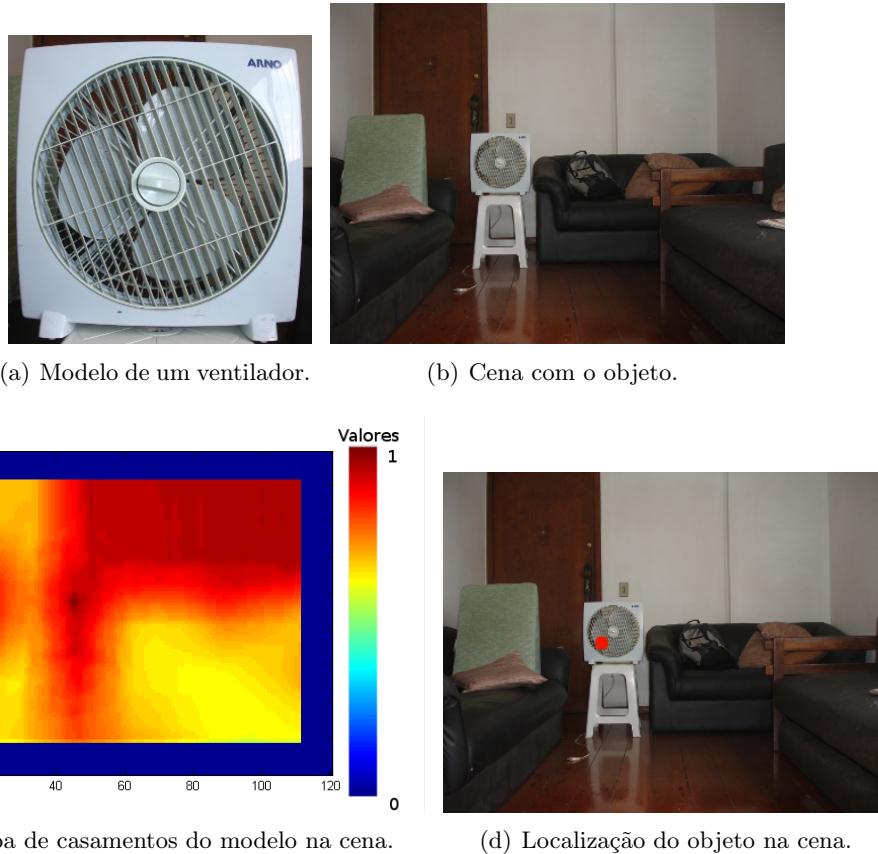


Figura 2.1: Exemplo da abordagem casamento de modelos. O modelo é procurado na cena e o resultado final aparece no ponto vermelho.

- *Confiança no reconhecimento*: todas as aplicações de reconhecimento de objetos tentam reduzir as taxas de falsos positivos, objetos com defeitos classificados como perfeitos, e falsos negativos, objetos perfeitos classificados como defeito.
- *Propriedade de invariância*: praticamente todas as aplicações de reconhecimento de objetos precisam possuir algum tipo de invariância. Algoritmos de reconhecimento de objetos são projetados para maximizar as discrepâncias entre objetos de categorias diferentes e minimizar as discrepâncias entre diferentes objetos da mesma categoria.

Ainda de acordo com [Tre10], um algoritmo de reconhecimento de objetos pode ter a invariância em relação às seguintes propriedades:

- *Iluminação*: a aparência dos objetos é influenciada pela iluminação presente na cena onde o objeto está. A iluminação pode variar em intensidade, ângulo de incidência ou cor.
- *Escala*: objetos aparecem maiores ou menores em uma imagem dependendo da distância em que foram fotografados. Os algoritmos devem estar preparados para reconhecer o objeto independente do tamanho em pixel que ele aparece em uma imagem.
- *Rotação*: alguns objetos são frequentemente mostrados nas imagens em diferentes rotações.
- *Fundo de cena confuso*: imagens reais dificilmente mostram apenas o objeto de interesse. Ele está inserido em um contexto onde vários outros objetos também podem aparecer. Existe uma grande variabilidade de fundos em que um objeto pode aparecer.
- *Oclusão parcial*: várias vezes, em uma imagem, apenas parte do objeto aparece ou, ainda, outros objetos aparecem na imagem obstruindo a imagem do objeto de interesse.



Figura 2.2: Resultado da detecção de objetos utilizando descritores globais, neste exemplo utilizou-se um histograma de níveis de cinza com 15 intervalos.

- *Alteração de ponto de vista dos objetos:* a projeção de objetos 3D em cenários 3D, em um plano 2D depende da posição da câmera durante a captura da imagem. Esta posição da câmera altera sensivelmente a aparência do objeto. A invariância quanto ao ponto de vista não é possível para objetos de formato arbitrário [Mun06]. Porém pode ser obtida uma invariância parcial para certas variações de pontos de vista.

2.3 Saco de Palavras

O saco de palavras, em inglês *bag-of-words* ou BoW, é uma técnica proposta por [SZ03] para ser utilizada na área de reconhecimento de objetos. Ela é derivada da categorização de textos [Joa98]. Posteriormente, foi utilizada em classificação de texturas [CD01, LM01]. Tornou-se uma das abordagens mais eficazes para reconhecimento de objetos.

A ideia de que textos podem ser classificados a partir da frequência de palavras é a base da técnica do saco de palavras. Cria-se um dicionário com o conjunto de palavras presentes na coleção de textos. Um modelo é criado para textos de economia, por exemplo. Outro modelo é criado para textos de biologia e, assim, sucessivamente. Um texto sobre economia terá um vocabulário específico sobre este assunto, enquanto que um texto sobre biologia terá uma frequência bem maior de vocábulos desta área. Cada texto é representado pela frequência normalizada das palavras presentes nele. Um classificador é treinado para criar modelo de textos de economia e biologia, por exemplo. Um texto desconhecido é comparado aos modelos e o resultado é o rótulo do assunto com a maior possibilidade do texto pertencer. Textos com frequência maior de palavras ligadas a economia serão classificados como de economia. A adaptação para a categorização de objetos partiu desta ideia. O saco de palavras começa com a extração de descritores locais nas imagens. Estes descritores são agrupados em um número pré-determinado de centros. O algoritmo de agrupamento geralmente utilizado é o k-médias [SI84]. O ponto central de cada agrupamento é chamado de *palavra visual*. A coleção de palavras visuais forma o *dicionário visual*. Cada descritor local é traduzido para o centro do agrupamento de que aquele descritor faz parte. O descritor final na abordagem do saco de palavras é o histograma normalizado de frequência das palavras visuais. Nenhuma informação espacial é mantida no descritor final, uma abordagem totalmente diferente em relação a modelagem baseada em partes.

A abordagem do saco de palavras possui, tipicamente, as seguintes etapas:

- *Extração de características:* características são extraídas em alguns pontos. Estes pontos po-

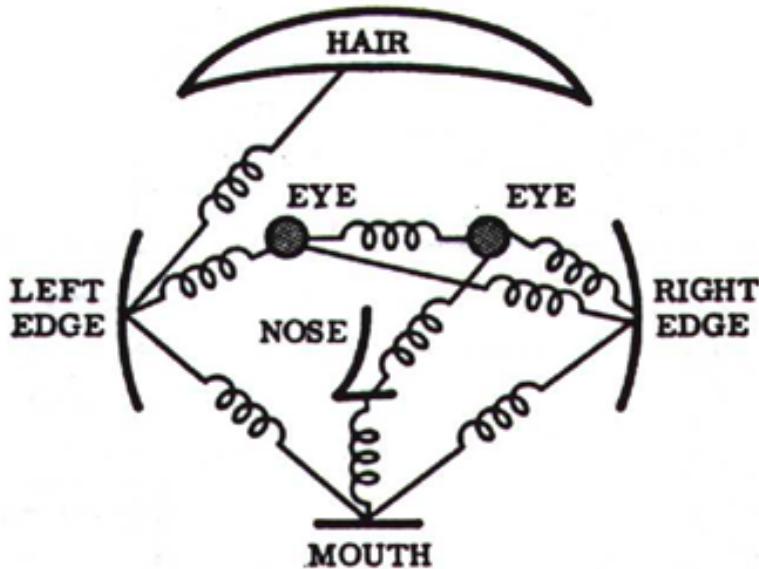


Figura 2.3: Imagem representando a abordagem de modelagem baseada em partes. Imagem do trabalho de [FE73].

dem ser de interesse, numa grade regular de pontos ou pontos aleatórios na imagem.

- *Representação das características*: um descritor é calculado usando as características extraídas na etapa anterior.
- *Construção do dicionário visual*: um número N de palavras visuais é escolhido. Os descritores são agrupados em N diferentes agrupamentos. O centro de cada agrupamento é um vetor que representa os descritores daquele agrupamento, cujo ponto central é a palavra visual. A coleção das N palavras visuais forma o dicionário visual.
- *Representação da imagem*: cada descritor da imagem é representado pela palavra visual mais próxima. A representação final da imagem é o histograma normalizado de frequência de palavras visuais.
- *Treinamento*: um modelo específico de categoria é aprendido usando amostras rotuladas e contra-exemplos para treinar o classificador.
- *Reconhecimento*: o modelo construído na etapa anterior é utilizado para determinar se uma imagem com rótulo desconhecido possui ou não um objeto da categoria de interesse.

Mais formalmente, o modelo do saco de palavras pode ser definido da seguinte maneira. Dado um conjunto de treinamento D contendo n imagens representadas por $D = d_1, d_2, d_3, \dots, d_n$, onde d são as características visuais extraídas. Um algoritmo de aprendizado não supervisionado, como o k-médias, é utilizado para agrupar D em um número fixo de categorias W representadas por $W = w_1, w_2, w_3, \dots, w_v$, onde v é o número de aglomerações. Pode-se sumarizar os dados em uma tabela de co-ocorrências de tamanho $v \times n$ com contagens $N_{ij} = n(w_i, d_j)$, onde $n(w_i, d_j)$ é o número de ocorrências da aglomeração w_i na imagem d_j [Tsa12].

A Figura 2.4 mostra as etapas da abordagem do saco de palavras. A Figura 2.4(a) mostra a extração de características em alguns pontos de interesse. No caso da imagem mostrada, foi utilizado o detector SIFT. O descritor SIFT é calculado nos pontos de interesse encontrados na etapa anterior (Figura 2.4(b)). Cada descritor pode ser visto como um ponto no espaço multidimensional (Figura 2.4(c)). Os descritores, pontos no espaço multidimensional, são agrupados em um número pré-determinado de N centros (Figura 2.4(d)). A representação final da imagem é o histograma normalizado de frequência da ocorrência destes centros, chamados de palavras visuais ou, em inglês, *codewords* (Figura 2.4(e)).

2.3.1 Detecção de Pontos

A escolha dos pontos onde serão extraídas características visuais que serão utilizadas para o cômputo dos descritores locais é a primeira etapa do modelo do saco de palavras. Existem vários detectores de pontos de interesse que poderiam ser utilizados nesta etapa [TM08]:

1. *Detector de Harris-Laplace*: regiões são detectadas por uma adaptação para escalas da função de Harris e selecionados em espaço-escala por um operador Laplaciano de Gaussianas. Este detector detecta estruturas em forma de cantos.
2. *Detector Diferença de Gaussianas*: regiões são localizadas na máxima espaço-escala local de diferenças de gaussianas. Este detector é mais adequado a encontrar regiões de manchas (*blob*). Os pontos localizados por este detector tem bom desempenho e o detector é rápido e existem, geralmente, menos pontos em uma imagem.
3. *Detector de Hessian-Laplace*: regiões são localizadas no espaço no máximo local de um determinante de Hessian e na escala no máximo local de um Laplaciano de Gaussiana.
4. *Entropia*: regiões salientes são detectados no espaço-escala em uma máxima local de entropia. A entropia de histogramas de intensidade de pixels são medidas por regiões circulares de vários tamanhos em cada posição da imagem.
5. *Regiões Extremas Estáveis Maximais*: MSER, sigla em inglês para *Maximally Stable Extremal Regions*, são componentes conexos de uma imagem limiarizada. Um algoritmo similar ao *Watershed* é aplicada às intensidades da imagem e fronteiras de segmentos que são estáveis em uma variedade grande de limiares definem uma região.

Os pontos que serão utilizados para extração de características podem ser calculados utilizando uma abordagem esparsa ou uma abordagem densa. A abordagem esparsa utiliza algum algoritmo de detecção de pontos de interesses, como os descritos aqui. A abordagem densa utiliza uma grade densa de pontos na imagem. [FFP05] mostrou que a abordagem densa de pontos possui desempenho superior a abordagem esparsa. As características também podem ser extraídas em pontos aleatórios. A abordagem densa tem mostrado melhor desempenho para tarefas de reconhecimento de objetos [vdSGS10b].

2.3.2 Extração dos Descritores Locais

O SIFT 2.6.1 é o descritor mais utilizado para representar as características visuais locais. O descritor SURF [BTVG06] vem ganhando popularidade por possuir um melhor desempenho computacional e possuir um número menor de dimensões no descritor final (64). [MS05] mostrou que o descritor SIFT e seus derivados possuem desempenho superior e eles têm sido utilizados com mais frequência no modelo do saco de palavras. Poucas alternativas têm sido utilizadas como descritores locais no modelo do saco de palavras [Tsa12].

Os descritores locais são calculados em pontos de interesse, em pontos aleatórios ou em nós de uma grade densa de pontos, como descrito na Seção 2.3.1. Após esta etapa, cada imagem I é representada por um conjunto de descritores de características locais:

$$X = [x_1, x_2, \dots, x_M]^T \in \mathcal{R}^{M \times D},$$

onde M é o número de descritores locais na imagem e D é o número de dimensões do descritor local.

2.3.3 Codificação

A representação das imagens a partir do conjunto de descritores locais extraídos contém muita informação, porém é muito grande para ser eficiente computacionalmente. Esta etapa, chamada

de codificação, traduz os descritores locais em uma representação mais eficiente. Uma codificação popular é traduzir os descritores locais em palavras visuais de um dicionário visual. A construção do dicionário visual é descrita na Seção 2.3.4. A representação da imagem, utilizando esta abordagem, ficaria da seguinte maneira [YZ11]:

$$V = [v_1, v_2, \dots, v_K]^T \in \mathcal{R}^{K \times D},$$

onde v_1, v_2, \dots, v_K são as palavras visuais do dicionário visual. Assim, cada descritor local é quantizado em um ou mais palavras visuais para formar uma matriz $A \in \mathcal{R}^{M \times K}$ onde cada elemento da matriz A $\alpha_{i,j}$ define se o i -ésimo descritor está representado pelo j -ésima palavra visual. Existem duas maneiras de traduzir o descritor local em uma palavra visual: atribuição rígida e atribuição leve [PVT12]. A atribuição rígida é a mais utilizada até agora:

$$\alpha_{i,j} = \begin{cases} 1 & \text{se } j = \arg \min_k \|x_i - v_k\|_2^2, \\ 0 & \text{caso contrário.} \end{cases}$$

A matriz A possui um número diferente de zero, igual a um, para cada linha da matriz. Uma outra atribuição é a atribuição leve. A atribuição leve possui vantagens em relação a atribuição rígida [LWL11]. Um descritor local que esteja na fronteira entre duas palavras visuais será representada por aquela palavra visual que esteja ligeiramente mais próxima pelo método original. Uma divisão da influência do descritor local entre as palavras visuais mais próximas seria uma representação mais fiel. A influência do descritor local é diluída entre estas palavras visuais. Quanto mais próximo do descritor local, maior será a influência dele na palavra visual. A atribuição leve é definida por:

$$\alpha_{i,j} = \frac{\exp(-\beta \|x_i - v_j\|_2^2)}{\sum_{k=1}^K \exp(-\beta \|x_i - v_k\|_2^2)},$$

onde β é o fator de atribuição leve.

Uma outra técnica de codificação é a codificação esparsa. A matriz $A \in \mathcal{R}^{M \times K}$ é definida pelo seguinte problema de otimização:

$$\arg \min_{A,V} \sum_{i=1}^M \|x_i - \alpha_i V\|^2 + \lambda |\alpha_i|,$$

tal que: $\|v_k\| \leq 1, \forall k = 1, 2, \dots, K$, onde λ é a regulação esparsa assegurando que a maioria dos valores de A sejam iguais a zero.

TF-IDF

Dar peso aos termos é uma estratégia para penalizar palavras que sejam muito comuns e dar mais importância a palavras pouco comuns. Esta técnica é derivada da técnica TF-IDF do inglês *Term Frequency-Inverse Document Frequency* ou Frequência de Termos-Inverso da Frequência de Documentos. A fórmula para TF-IDF é: $tf_i \log(\frac{N}{N_i})$, onde tf_i é o termo de frequência da i -ésima palavra, N é o número de documentos e N_i é o número de documentos na base de dados contendo a i -ésima palavra [OD11].

2.3.4 Geração do Dicionário Visual

Após a definição dos pontos a serem utilizados na imagem e do cômputo dos respectivos descritores locais, a construção do dicionário visual começa com a quantização dos descritores locais em um número pré-determinado de aglomerações. Geralmente, o k-médias é utilizado. O centro de cada aglutinação de descritores locais forma uma palavra visual. A coleção de palavras visuais forma o dicionário visual. Para representar a imagem utilizando o dicionário visual, os descritores locais de cada imagem são traduzidos em palavras visuais (utilizando a atribuição rígida ou a leve). A

representação final de uma imagem é, geralmente, o histograma de frequência de palavras visuais. Existem vários trabalhos que exploram o fato de que o modelo do saco de palavras não possuir nenhuma informação espacial. Estes trabalhos tentam adicionar algum tipo de informação espacial ao modelo original. O principal trabalho desta categoria é [LSP06] que introduz as pirâmides espaciais.

[vdSGS11] mostrou que o principal gargalo de desempenho computacional do modelo do saco de palavras é a aglutinação utilizando o k-médias. Alguns trabalhos propõem alternativas utilizando árvores aleatórias [USS10], k-médias hierárquico e k-médias aproximado [PCI⁺07]. Outra alternativa ao uso do k-médias é a amostragem dos descritores locais para gerar o dicionário visual. [VL08] mostrou que a amostragem de descritores locais gera dicionários visuais de qualidade comparável ao uso de aglomeração a um custo computacionável muitas vezes menor.

2.3.5 Agrupamento

Após a etapa de codificação, vem a etapa de agrupamento, em inglês *pooling*. O objetivo desta etapa é resumir as características codificadas em cada imagem para formar a representação final da imagem. A etapa de agrupamento deve gerar uma representação da imagem que seja invariante a transformações da própria imagem, que seja robusta a ruídos e fundos confusos e que possua uma representação compacta [YZ11]. Os métodos de agrupamento fazem uma transformação da matriz $A \in \mathcal{R}^{M \times R}$ em um vetor de dimensão K (histograma) $H \in \mathcal{R}^K$. Existem duas técnicas tradicionais de agrupamento:

- *Agrupamento pela média*: no agrupamento pela média, H é a soma ou a média de todos os valores das colunas da matriz A .
- *Agrupamento pelo máximo*: no agrupamento pelo máximo, H é o valor máximo das colunas da matriz A .

2.3.6 Modelos de Aprendizado

Após a construção do modelo do saco de palavras e a geração da representação final das imagens através das técnicas de codificação e agrupamento, existem duas alternativas. A primeira utiliza o vetor de características do modelo do saco de palavras como entrada em um classificador para treinamento e teste. São chamados modelos discriminativos. Os modelos generativos são utilizados para descobrir categorias utilizando análise semântica latente.

Modelos Discriminantes

Modelos discriminantes são baseados em aprendizado supervisionado de máquinas. Aprendizado supervisionado de máquinas podem ser vistos como aprendizado por exemplos. Para que o modelo seja capaz de detectar a presença de um objeto de uma determinada categoria em uma imagem, o modelo é construído utilizando uma série de exemplos positivos e negativos. A tarefa de aprendizado consiste em construir um modelo \hat{f} que crie um mapeamento entre os vetores de características de treinamento e os rótulos corretos com algum nível de acurácia. Esta etapa é chamada de treinamento. Após isto, o modelo é capaz de atribuir um rótulo para uma amostra desconhecida. O modelo calcula o grau de similaridade entre o vetor de característica da amostra de teste e todos os dados de treinamento das diversas categorias conhecidas e atribui o rótulo da categoria com maior grau de similaridade.

Modelos Generativos

Duas técnicas (pLSA e LDA) utilizadas em análise de textos para descobrir tópicos em documentos utilizando o modelo do saco de palavras foram adaptadas para descobrir categorias presentes em uma imagem [Hof99]. Em imagens contendo vários objetos, estas combinações de objetos são

utilizadas para classificar as cenas presentes na imagem. Por exemplo, água com ondas, areia e céu com nuvens, seriam objetos presentes em uma cena de praia [BMM07].

É uma técnica não supervisionada. O modelo gerativo utilizada na análise de textos possui dois níveis. O primeiro nível considera o documento uma mistura de tópicos e cada tópico tem sua própria distribuição de palavras.

$$p(w_i|d_j) = \sum_{k=1}^K p(w_i|z_k)p(z_k|d_j),$$

a distribuição de palavras w_i em um documento d_j é função da distribuição de palavras em um tópico z_k . A distribuição $p(w_i|d_j)$ é conhecida e as distribuições $p(w_i|z_k)$ e $p(z_k|d_j)$ são desconhecidas. Para encontrar o tópico mais provável em uma imagem:

$$z^* = \arg \max_z p(z|d) = \arg \max_z \frac{p(w|z)p(z|d)}{\sum_{z'} p(w|z')p(z'|d)}.$$

2.4 Pirâmides Espaciais

Pirâmide Espacial é uma técnica baseada em correspondência aproximada da geometria global entre imagens. A técnica funciona particionando uma imagem em sub-regiões menores e computando um histograma de características locais em cada sub-região. Uma correspondência de características locais é calculada em cada sub-região e a correspondência final é calculada utilizando as correspondências locais. A abordagem do saco de palavras despreza quaisquer informações espaciais ao contrário da modelagem baseada em partes. A pirâmide espacial é uma alternativa que tem mostrado um ganho na acurácia da classificação em aplicações de reconhecimento de objetos a um custo computacional eficiente. A técnica acrescenta informações espaciais de maneira grosseira ao modelo do saco de palavras. Esta informação espacial, apesar de grosseira, é suficiente para melhorar o desempenho da técnica original. A pirâmide espacial foi proposta em 2006 por [LSP06].

A imagem é particionada recursivamente em sub-regiões cada vez menores. A técnica original implementou a subdivisão duplicando o número de divisões em cada eixo a cada nível. Em um nível inicial, nível 0, a imagem original permanece sem divisões, ou seja, existe apenas uma região. O próximo nível, nível 1, subdivide a única região do nível 0 em 4 outras regiões dividindo o eixo x em dois e o eixo y também em dois formando as quatro regiões do nível 1. O nível 2 subdivide cada uma das regiões do nível 1 em 4 outras regiões, totalizando 16 regiões.

Formalizando esta abordagem, temos que a pirâmide possui os níveis $0, \dots, L$. Em um nível l , a pirâmide possui 2^l divisões em cada dimensão, o que dá um total de $D = 2^{2l}$ regiões em cada nível. Seja I_1 e I_2 duas imagens digitais (Seção 2.1), H_I^l o histograma de descritores da imagem I no nível l . O casamento de descritores entre duas regiões de duas imagens diferentes é igual ao número de descritores das duas imagens que estejam em uma mesma região de um mesmo nível. O casamento de descritores de um nível é igual a soma dos casamentos dos descritores das regiões deste nível. Formalizando, o casamento para o nível l entre as imagens I_1 e I_2 pode ser calculado:

$$M(I_1^l, I_2^l) = \sum_{i=1}^D \min(H_{I_1}^l(i), H_{I_2}^l(i)),$$

onde D é o número de regiões em um nível. Pode-se denotar a equação acima por M^l . Os casamentos do nível l incluem os casamentos que ocorrem no nível $l+1$. Assim os novos casamentos no nível l são dados por $M^l - M^{l+1}$, para $l = 0, \dots, L-1$. A cada nível são associados pesos inversamente proporcionais à largura da célula. No nível l associa-se o peso $\frac{1}{2^{L-l}}$. Assim, é dado maior peso para casamentos que ocorrem nos níveis mais altos, menores células. Estes casamentos ocorrem em uma

região espacial menor. A função núcleo de casamento de pirâmides pode ser escrita por:

$$k^L(I_1, I_2) = \frac{1}{2^L} M^0 + \sum_{l=1}^L \frac{1}{2^{L-l+1}} M^l.$$

Repetindo a equação anterior para cada uma das N palavras visuais do dicionário, o esquema de casamento das pirâmides espaciais pode ser escrito por:

$$K^L(I_1, I_2) = \sum_{n=1}^N k^L(I_1, I_2).$$

As pirâmides espaciais têm sido bastante utilizadas em aplicações de reconhecimento de objetos. Várias vezes as pirâmides são utilizadas para definir o descritor final apenas como o histograma normalizado de frequência das palavras visuais em cada nível. A Seção 3.1 propõe uma extensão à pirâmide espacial tradicional com relaxamento da subdivisão dos níveis e da subdivisão em células. As células de um nível l não precisam ser necessariamente subdivisões das células do nível $l - 1$. Este trabalho propõe ainda um relaxamento nas fronteiras das células podendo existir uma sobreposição nas regiões de fronteira entre duas células consecutivas. A ideia é que a melhor configuração de divisão espacial pode ser encontrada durante as etapas de treinamento e validação e, assim, a divisão espacial se adaptaria à forma dos objetos cujo modelo está sendo criado. O trabalho de [ZM10] propôs efetuar as divisões espaciais de uma maneira diferente, utilizando coordenadas polares e com ganho de desempenho, segundo os autores, em reconhecimento de objetos. A Seção 3.3 propõe uma divisão espacial da imagem centrada em um ponto e com círculos concêntricos e esta divisão se mostra bem mais robusta a variações na rotação das imagens que a divisão espacial tradicional.

A Figura 2.5 mostra um exemplo do uso da pirâmide espacial para composição do descritor final da imagem. A pirâmide espacial utilizada possui 2 níveis: 1 célula no nível 0 e 4 células no nível 1.

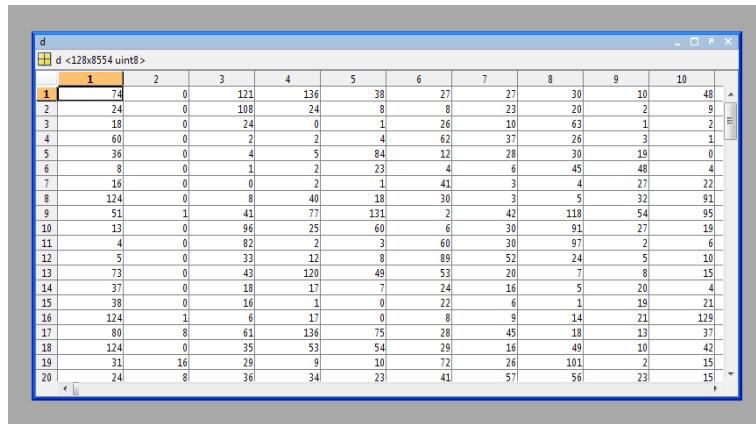
2.5 Categorização Refinada

Categorização refinada é a categorização de objetos que pertencem a uma mesma classe básica, por exemplo, diferentes espécies de pássaros [YBFF12]. Ainda não existe uma definição formal de categorização refinada. Em [BWS⁺10], categorização refinada é considerada uma categorização que necessita de uma perícia, conhecimento ou treinamento especial para ser realizado. Em [MMLM⁺09], a categorização refinada de diferentes espécies de artrópodes é definida como extremamente desafiadora e mesmo especialistas treinados têm dificuldade neste tipo de classificação. O artigo caracteriza os objetos da base de dados como possuindo grande variabilidade intra-classe e pequena diferença inter-classe. Categorização refinada é a classificação de objetos de categorias que possuem significante similaridade visual. Esta é a definição utilizada em [NZ06]. O trabalho [FOZ⁺11] situa a categorização refinada como um problema que fica a meio caminho entre a classificação de objetos tradicional e a classificação de indivíduos. A falta de uma definição formal do problema e as diferentes definições encontradas na literatura mostram que se trata de um problema novo que só recentemente despertou o interesse da comunidade. O interesse recente se deve, principalmente, a rápida evolução de desempenho em tarefas de categorização de objetos básicas, consideradas mais fáceis que o problema da categorização refinada. Daí a comunidade se volta a questões mais desafiadoras. Neste trabalho, definiremos categorização refinada como a categorização de objetos de uma mesma classe básica que possuem grande similaridade visual entre si.

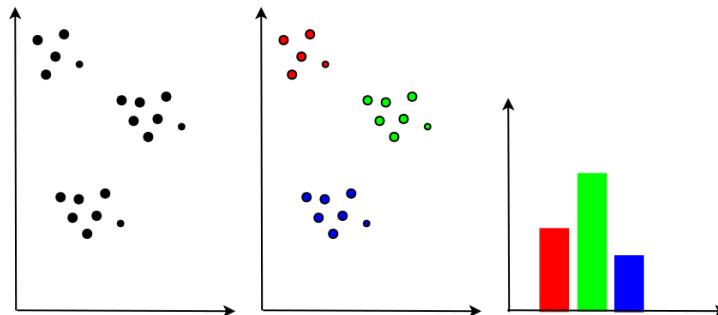
Este problema é bastante difícil pois objetos de categorias diferentes possuem várias características visuais parecidas. Os melhores algoritmos de categorização de objetos não apresentam bom desempenho quando os objetos são de uma mesma categoria básica. Algoritmos de categorização trabalham em generalizações para classificar os objetos em diferentes categorias. Essas generalizações, como a abordagem do saco de palavras, perdem informações sutis que diferenciam um objeto de uma categoria de outra categoria muito similar. A categorização de objetos de classes muito



(a) Pontos de interesse encontrados pelo detector SIFT.

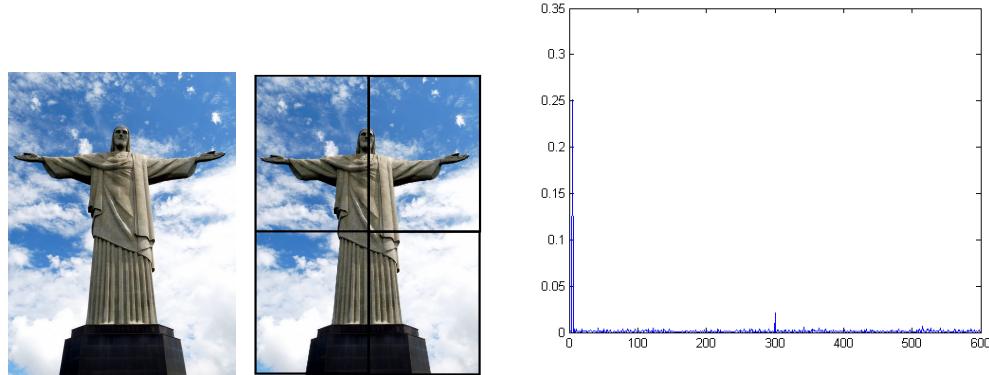


(b) Descriptor SIFT calculado nos pontos de interesse.



(c) Descriptor visto (d) Descritores aglomerados em torno da imagem: his-
com pontos no es- merados em torno da imagem: histo-
paço N-dimensional. N centros: dicionário
paço N-dimensional. N centros: dicionário
visual é construído
visual é construído
de frequência das pa-
lavras visuais.

Figura 2.4: Imagem do hotel Burj Al Arab da base de instâncias de monumentos utilizada nesta tese com pontos de interesse localizados pelo detector SIFT e ilustração das etapas da abordagem do saco de palavras.



0, a imagem inteira dividida em 4 regiões.
forma uma região.

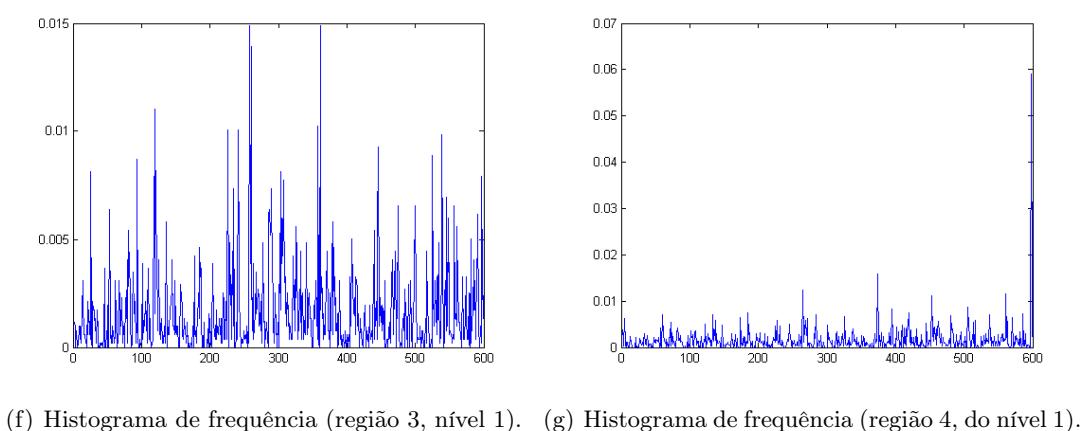
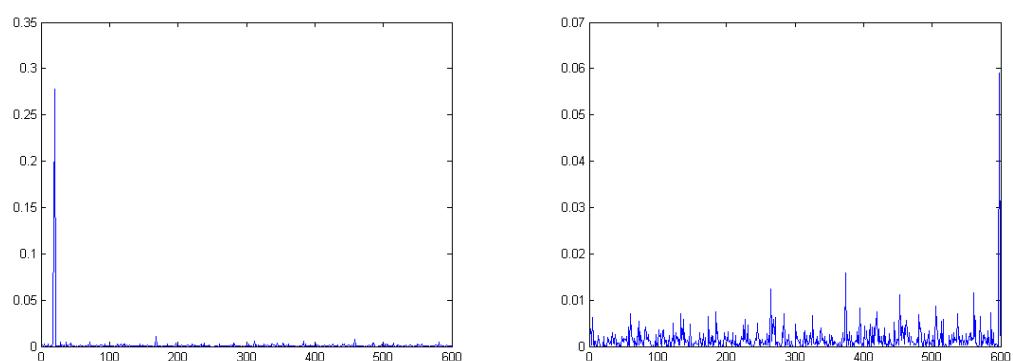


Figura 2.5: Imagem do Cristo Redentor da base de instâncias de monumentos utilizando uma pirâmide espacial de 2 níveis para composição do descriptor final.

parecidas pode ter aplicações reais importantes, como: identificação de plantas tóxicas em um ambiente selvagem; identificação de quais frutas são comestíveis e quais não são ou identificação de animais peçonhentos, como serpentes. Todas estas tarefas listadas têm algo em comum: são inspeções visuais feitas por especialistas, ou alguém especialmente treinado para a tarefa, o que difere da categorização de objetos tradicional que é uma classificação visual que qualquer pessoa tem condições de fazê-la. Mesmo uma criança consegue classificar corretamente as 20 categorias que a base de dados do desafio VOC Pascal utiliza. A Figura 1.2(b) mostra uma imagem para cada uma das categorias da base de dados VOC Pascal [EGW⁺10].

A categorização refinada demanda que o algoritmo consiga discriminar pequenas diferenças entre classes muito parecidas. A Figura 1.2(c) mostra a dificuldade da tarefa. São mostrados 7 pássaros de 7 categorias diferentes da base de pássaros Caltech-UCSD Birds-200 2011. São todos pássaros do gênero *Vireo*. Uma das hipóteses para o desempenho mediano da abordagem do saco de palavras para a categorização refinada é a generalização ocorrida na técnica quando os descritores locais são aproximados ao serem representados por uma palavra visual e também o fato da técnica não levar em consideração que a localização espacial dos descritores é uma generalização. O saco de palavras pode utilizar as pirâmides espaciais ou outro tipo de quantização espacial, porém estas quantizações espaciais adicionam informações espaciais de maneira grosseira. Estas generalizações do saco de palavras fazem com que diferenças sutis e importantes neste tipo de categorização sejam perdidas e a acurácia final da categorização é prejudicada [YBFF12].

Uma alternativa à construção de dicionários visuais automáticos é utilizar anotação humana de pontos de interesse. Estes pontos de interesse são anotados por especialistas e são altamente discriminantes. A base de pássaros Caltech-UCSD Birds-200 2011 [WBW⁺11] possui anotada para cada imagem, a localização de até 15 partes diferentes do pássaro, caso estas partes estejam visíveis. Estes pontos podem ser utilizados na construção do dicionário visual mais discriminante, já que são pontos localizados em partes importantes dos pássaros. A Seção 5.6.1 descreve esta base de dados em detalhes.

A base de pássaros Caltech-UCSD Birds-200 2011 não é a única utilizada para categorização refinada. O trabalho de [NZ06] utiliza uma base de dados de flores para categorização refinada, [MMLM⁺09] utiliza imagens de larvas de mosca para uma categorização de objetos muito parecidos. Alguns trabalhos utilizam ramos de classes similares da estrutura hierárquica de categorias da base de dados ImageNet [DDS⁺09]. A base ImageNet é a maior base de dados de reconhecimento de objetos com aproximadamente 18000 categorias. Trabalhos como [DBLFF10] utilizam as categorias de fungos desta base para desenvolver algoritmo de categorização refinada.

Várias abordagens são utilizadas em trabalhos com foco em categorização refinada. Entretanto o problema permanece aberto e nenhuma abordagem se mostrou nitidamente superior às demais. A utilização da abordagem do saco de palavras para categorização refinada é natural, visto que esta abordagem é o estado da arte em categorização de objetos. Trabalhos recentes como [WYY⁺10] utilizam uma codificação esparsa na construção do dicionário para obter um incremento no desempenho final da categorização. Outros trabalhos utilizam mais informações de anotações manuais, como a utilização das partes anotadas na base de pássaros Caltech-UCSD Birds-200 2011. Exemplos de trabalhos que utilizam esta abordagem altamente focada na anotação manual são [BWS⁺10] e [WPB11]. Além do alto custo de anotação destas bases, já discutido anteriormente, existe ainda o custo de adaptar o algoritmo desenvolvido para estas bases de dados para resolver o problema em outra base de dados, já que ele é dependente das anotações manuais específicas desta base.

O artigo que teve o melhor resultado até agora na categorização refinada de pássaros da base Caltech-UCSD Birds-200 2011, pelo que sabemos, é o artigo de [YBFF12]. Não foi utilizada a base inteira, mas apenas pássaros do gênero *Vireo* (são 7 categorias diferentes na versão 2011 da base) e 6 espécies de pica-paus. A escolha por este subconjunto da base de pássaros também foi feita nos experimentos do trabalho de [FOZ⁺11] e também foi utilizada na parte experimental deste trabalho descrita na Seção 5.6.2. O trabalho de [YBFF12] utilizou uma técnica baseada no casamento de modelos, uma abordagem diferente dos trabalhos anteriores. Um número grande de modelos é gerado a partir das imagens do conjunto de treinamento. Estes modelos são retângulos de

altura e largura aleatórios. Uma imagem de teste é representada pelas notas de casamento entre elas e os modelos gerados. O algoritmo de casamento de modelos foi proposto nos trabalhos [HCl⁺12] e [HHC⁺11] e utiliza as cores em formato RGB e o gradiente da imagem. Cada modelo é utilizado em várias escalas diferentes. Esta série de casamentos forma um mapa de respostas e deste mapa de respostas calcula-se um vetor de características com 7 dimensões:

1. Maior valor do mapa;
2. Segundo maior valor do mapa;
3. Terceiro maior valor do mapa;
4. Maior valor da região I (o mapa de respostas é dividido em 4 quadrantes, numerados a partir do quadrante superior direito em sentido horário);
5. Maior valor da região II;
6. Maior valor da região III;
7. Maior valor da região IV.

Existe uma restrição quanto aos maiores valores do mapa: eles devem estar afastados a uma distância de um décimo da largura ou altura, o que for menor. A representação final da imagem de teste é a concatenação destes resultados para cada modelo e para cada escala. A dimensão final do vetor de características é igual a Número de Modelos × Número de Escalas × 7. A representação da imagem é redundante visto que inúmeros modelos possuem sobreposição de região. Além disto, existem modelos que não possuem elementos discriminantes. O classificador SVM tende a ter um super ajuste aos dados de treinamento para descritores com número muito grande de dimensões. A dimensão do descritor final neste trabalho tem perto de 900000 dimensões. Para evitar isto, o artigo propõe usar vários classificadores SVM. Um método derivado do *bagging* [DHS01] é proposto pelo artigo para utilizar partes diferentes do conjunto de treinamento e combinar os resultados dos classificadores.

2.6 Descritores

Esta seção descreve os descritores utilizados neste trabalho.

2.6.1 SIFT

A Transformada Característica Invariante a Escala ou SIFT, da sigla em inglês de *Scale Invariant Feature Transform*, é o descritor mais utilizado atualmente, contando com suas variações. Ele foi introduzido por [Low99]. O SIFT é formado por uma região da imagem localizada pelo detector de pontos de interesse e por um descritor associado a esta região. Ambos, detector e descritor, podem ser utilizados separadamente [RW08]. Um dos objetivos do detector de pontos de interesse do SIFT é encontrar pontos de interesse reconhecíveis em diferentes visões e escalas [Low04]. Seja I uma imagem digital e $I(x, y)$ é a amplitude de I no ponto $(x, y) \in \mathbb{E}$ (Seção 2.1). Para encontrar estes pontos, é necessário construir um espaço escala utilizando um núcleo gaussiano:

$$L(x, y, \sigma) = g(x, y, \sigma) * I(x, y),$$

onde $g(x, y, \sigma)$ é um núcleo gaussiano $g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$ e $*$ é o operador de convolução.

Uma maneira efetiva de encontrar pontos de interesse estáveis, invariantes a escala e a diferentes pontos de vista é detectar extremos locais em uma diferença de gaussianas (DoG) [VF08]:

$$DoG(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma).$$

Um grande número de pontos de interesse gerados possuem contraste baixo ou são localizados em bordas. Estes pontos são eliminados. O próximo passo é localizar a orientação θ e a magnitude m do gradiente para cada pixel na vizinhança do ponto de interesse:

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1, \sigma) - L(x, y - 1, \sigma)}{L(x + 1, y, \sigma) - L(x - 1, y, \sigma)}\right),$$

$$m(x, y) = \sqrt{\frac{L(x, y + 1, \sigma) - L(x, y - 1, \sigma)}{L(x + 1, y, \sigma) - L(x - 1, y, \sigma)}}.$$

Um histograma com as orientações é computado. Pesos para as magnitudes e uma função gaussiana circular são aplicados para o cômputo deste histograma. O pico do histograma é a orientação dominante. Se existir mais de uma com pelo menos 80% do valor do pico, um novo ponto de interesse é criado com a nova orientação e na mesma posição e escala do ponto de interesse original.

A vizinhança do ponto de interesse é dividida em regiões 4×4 . Cada área tem um histograma de 8 intervalos de orientação. O descritor final é a concatenação dos histogramas, o que totaliza 128 dimensões (4×4 regiões \times 8 intervalos de orientações).

O descritor SIFT tem uma motivação biológica [Tre10]. Observações biológicas têm mostrado que os neurônios no córtex visual têm uma resposta maior para algumas orientações de gradiente em particular. A Figura 2.6 mostra graficamente alguns dos parâmetros levados em consideração no cálculo do descritor SIFT. Um descritor SIFT é uma região circular (com seu ponto central (x, y) e um raio, que é a escala s) juntamente com a orientação θ [VF08]. A Figura 2.6(a) mostra os parâmetros que compõem o descritor SIFT, região circular formada pelos parâmetros x, y e raio s e a orientação θ . A Figura 2.6(b) mostra as 16 regiões com os histogramas espaciais de orientação em 8 intervalos que compõem o descritor final SIFT.

Uma variação do SIFT tradicional aplica o descritor em pontos aleatórios ou em uma grade densa de pontos sobre a imagem [NJT06].

2.6.2 PHOW

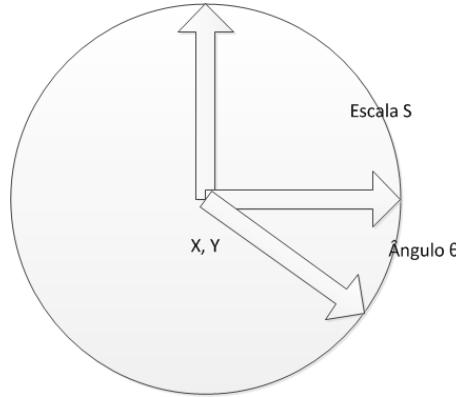
O Histograma Piramidal de Palavras Visuais ou PHOW do inglês *Pyramidal Histogram of Visual Words* é um descritor baseado no SIFT introduzido em [BZM07a]. Ele usa uma grade densa de pontos na imagem e calcula, para cada ponto, o descritor SIFT usando uma ou mais escalas diferentes. Para cada ponto o descritor SIFT pode ser aplicado em cada um dos 3 canais RGB perfazendo 384 (128 dimensões do SIFT \times 3 canais do espaço de cor RGB). O PHOW foi originalmente proposto para ser utilizado em pirâmides espaciais descritas na Seção 2.4. Em [WYY⁺10], o descritor PHOW é citado como um dos melhores descritores para aplicações de reconhecimento de objetos.

2.6.3 SURF

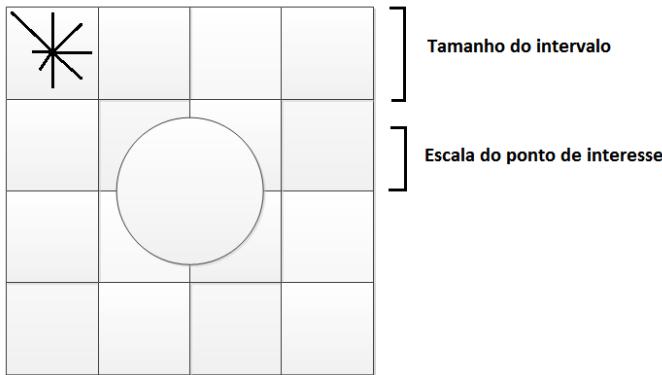
O descritor chamado de Características Robustas Aceleradas ou SURF, da sigla em inglês de *Speeded Up Robust Features*, é um descritor invariante a escala e a rotação proposto em [BTVG06]. Como o SIFT, ele consiste de um detector de pontos de interesse e de um descritor. O detector é uma aproximação da matrix de Hessian [MS01] que usa imagem integral [VJ01]. O descritor é baseado na resposta da ondaleta de Haar [GW02] e usa apenas 64 dimensões no vetor final. Os autores dizem que o descritor SURF é mais robusto e tem um custo computacional menor quando comparado ao SIFT.

2.6.4 PHOG

O Histograma Piramidal de Gradientes Orientados ou PHOG, da sigla em inglês de *Pyramidal Histogram of Oriented Gradients*, é um descritor baseado no HOG (Histograma de Gradientes Orientados ou em inglês *Histogram of Oriented Gradients*). O HOG foi proposto por [DT05].



(a) Ponto de interesse SIFT.



(b) Descritor SIFT.

Figura 2.6: Gráficos mostrando os principais parâmetros levados em consideração para calcular o ponto de interesse SIFT e o descritor SIFT. O histograma das orientações com 8 intervalos mostrado na primeira região da Figura 2.6(b) é calculado em cada uma das 16 regiões utilizadas no cômputo do descritor.

PHOG é uma adaptação do HOG proposto em [BZM07b] para ser utilizados em diferentes níveis, construindo uma pirâmide espacial. A pirâmide espacial foi descrita na Seção 2.4. A ideia por trás do descritor HOG é que um objeto pode ser descrito a grosso modo pela distribuição da orientação de seus gradientes. Para calcular o descritor, a imagem é dividida em várias regiões chamadas de células e em cada célula é compilado um histograma da orientação dos gradientes dentro daquela célula. A combinação de todos estes histogramas forma o descritor final.

2.6.5 LBP

Descriptor LBP proposto em [OPH94] bastante utilizado na classificação de texturas. A imagem é dividida em células com tamanho fixo. Para cada pixel da célula é feita uma comparação com os seus 8 vizinhos, em cada comparação, se o valor do pixel central for maior que o seu vizinho, ele é marcado com 1, 0 caso contrário. Estas comparações criam padrões binários de 8 dígitos para cada pixel. Um histograma é calculado sobre estes padrões binários. Os histogramas são normalizados e o descritor final é a concatenação dos histogramas normalizados de todas as células da imagem.

2.6.6 Histograma RGB

O histograma de cores quantifica quantas vezes um pixel com determinado valor aparece na imagem [GW02]. O histograma quase sempre é utilizado normalizado para facilitar comparações entre imagens com número de pixels diferentes. O histograma RGB utiliza a frequência normalizada em cada um dos canais RGB. Um parâmetro que foi utilizado é o número de agrupamentos dos valores dos pixels. Estes agrupamentos são utilizados por canal. Valores de pixels entre 0 e 255 para

um canal e 16 agrupamentos implica 16 valores para cada canal e 3 canais diferentes, totalizando 48 dimensões.

2.7 Classificadores

O classificador Máquinas de Vetores de Suporte ou SVM, da sigla em inglês de *Support Vector Machines*, foi proposto em [CV95] e é o classificador mais utilizado em problemas reconhecimento de objetos [Sze11]. SVM são, em sua forma mais simples, classificadores lineares binários supervisionados. SVM constrói um hiperplano em um espaço multidimensional que divide as duas classes do conjunto de treinamento. O hiperplano é construído de forma a maximizar a distância aos pontos dos dados de treinamento mais próximos de ambas as classes. Qualquer hiperplano pode ser construído como um conjunto de pontos \mathbf{x} que satisfazem:

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

onde \mathbf{w} é o vetor normal do hiperplano e $\frac{b}{\|\mathbf{w}\|}$ o deslocamento do hiperplano em relação à origem.

A tarefa de aprender uma máquina de vetores de suporte converte-se em um problema de otimização quadrática [SSSS07]. Dado um conjunto de treinamento $S = \{(x_i, y_i)\}_{i=1}^m$, onde $x_i \in \mathbb{R}^n$ e $y_i \in \{+1, -1\}$, é necessário encontrar o mínimo do problema:

$$\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{(\mathbf{x}, y) \in S} l(\mathbf{w}; (\mathbf{x}, y)),$$

onde $l(\mathbf{w}; (\mathbf{x}, y)) = \max\{\mathbf{0}, \mathbf{1} - \mathbf{y} \langle \mathbf{w}, \mathbf{x} \rangle\}$.

Se os dados não forem linearmente separáveis, é necessário usar o chamado *truque do núcleo* que utiliza uma função especial, a função núcleo, que mapeia pontos dos dados em um espaço de dimensão superior onde os dados são linearmente separáveis. A função núcleo depende da natureza do problema. Para resolver o problema de otimização, pode-se utilizar o solucionador Pegasos [SSSS07]. A função núcleo mais utilizada em problemas de reconhecimento de objetos é a qui-quadrada [RTG00].

2.7.1 Validação Cruzada

São técnicas utilizadas com o intuito de validar o quanto acurado é o modelo criado pelo classificador. A ideia é dividir o conjunto de dados de treinamento e utilizar parte deste conjunto para criar um modelo e a outra parte é utilizada para validar este modelo [Koh95]. Os principais métodos de particionamento do conjunto de treinamento são:

1. *holdout*: divide o conjunto de treinamento em 2 conjuntos distintos (treinamento e validação). As divisões mais comuns são meio a meio entre treinamento e validação e $\frac{2}{3}$ para treinamento e $\frac{1}{3}$ para validação.
2. *k-fold*: divide o conjunto de treinamento em k conjuntos distintos de mesmo tamanho. O processo de validação terá k etapas. Em cada etapa, um dos k subconjuntos serão utilizados para validação e os outros $k - 1$ subconjuntos serão utilizados como treinamento. A medida final é a média das medidas atingidas em cada uma das k etapas.
3. *leave-one-out*: caso especial do *k-fold* onde k é igual ao tamanho do conjunto de treinamento.

2.7.2 Combinação de Classificadores

Combinação de classificadores é uma técnica que usa o resultado de vários classificadores para obter a resposta final do problema. Ao utilizar vários classificadores e tomar a combinação de seus resultados, evita-se que seja utilizado apenas um classificador fraco [Pon11].

Quanto à arquitetura, os classificadores podem ser organizados de 2 formas:

- *Paralelo*: quando os classificadores são treinados em paralelo e a combinação ocorre após o treinamento.
- *Serial*: os classificadores são treinados de maneira sequencial, o segundo classificador melhora o desempenho do primeiro e, assim, sucessivamente.

Os principais métodos para criar uma combinação de classificadores [Pon11]:

1. *Aleatório*: o classificador utiliza um subconjunto aleatório do conjunto de treinamento ou o classificador utiliza algum tipo de inicialização aleatória.
2. Os principais métodos para se obter *diferentes subconjuntos de treinamento* são:
 - divisão do conjunto de treinamento em subconjuntos disjuntos;
 - validação cruzada no conjunto de treinamento;
 - escolha de amostras aleatórias do subconjunto de treinamento gerando subconjuntos de treinamento um pouco diferentes;
 - combinação de uma sequência de classificadores fracos, apenas um pouco melhores que a escolha aleatória, aumentando a acurácia da classificação final, método chamado de incremento adaptativo ou *adaptive boosting* en inglês.
3. *Diferentes subconjuntos do vetor de características*: treinar os classificadores usando subespaços escolhidos aleatoriamente do vetor de características.
4. *Diferentes subconjuntos do problema*: cada classificador resolve um subconjunto do problema de classificação de N classes.

As saídas de um classificador podem ser [Pon11]:

- *Rótulo*: o classificador retorna o rótulo da classe mais provável.
- *Lista*: o classificador retorna uma lista ordenada com os rótulos mais prováveis.
- *Nota*: o classificador retorna uma nota, probabilidade ou nível confiança de que o objeto classificado pertença a cada uma das classes.

Existem diversas maneiras de combinar os classificadores para ter o resultado final, as maneiras mais utilizadas são:

- *Mínimo*: acha as menores notas para cada classe de cada classificador e escolhe a classe com a maior nota.
- *Máximo*: acha as maiores notas para cada classe de cada classificador e seleciona a classe com maior nota.
- *Produto*: multiplica a nota de cada classificador para cada classe e escolhe a classe com maior produto.
- *Soma*: soma a nota de cada classificador para cada classe e escolhe a classe com maior soma.
- *Média*: acha a média das notas para cada classe e seleciona aquela com maior média.
- *Mediana*: acha a média das notas para cada classe e seleciona a classe com maior mediana.
- *Votação*: escolha a classe que foi a mais retornada como saída entre todos os classificadores.
- *Votação com peso*: a votação é multiplicada por um peso calculado durante a fase de validação do classificador.

ECOC

ECOC, do inglês *Error-Correcting Output Coding*, é um método de combinação de classificadores que utiliza classificadores diferentes para solucionar subconjuntos diferentes do problema [Pon11]. Este método foi proposto em [DB95] e é um método que transforma um problema multiclasse em vários problemas de 2 classes. A estratégia para combinar os classificadores consiste em criar uma matriz onde cada rótulo possível de classe representa uma linha e cada classificador binário possível representa uma coluna. Para uma dada coluna j de um classificador h_j e uma linha i com rótulo ω_i , o elemento ij da matriz vai ser preenchida com 1 se o classificador h_j retornar rótulo 1 para elementos da classe i e 0, caso contrário. Ao testar um novo objeto, as saídas dos classificadores formam um *codeword*. O rótulo cujo *codeword* apresentar a menor distância de Hamming em relação ao novo objeto, será o rótulo do novo objeto. A tabela ECOC deve ser construída de tal forma que evite colunas que tenham alta correlação entre si e, ainda, que garanta que os *codewords* sejam distantes entre si.

A Tabela 2.1 mostra um exemplo com 6 classificadores e 5 classes. O padrão para o rótulo ω_1 é o *codeword* 000101, para o rótulo ω_2 é o *codeword* 101001 e, assim, sucessivamente.

	h_1	h_2	h_3	h_4	h_5	h_6
ω_1	0	0	0	1	0	1
ω_2	1	0	1	0	0	1
ω_3	0	0	0	1	1	0
ω_4	1	1	0	0	0	0
ω_5	0	1	1	1	0	0

Tabela 2.1: Exemplo de uma tabela ECOC

2.7.3 Seleção de Características

Várias vezes, em um aprendizado supervisionado, o melhor resultado possível é obtido utilizando apenas um subconjunto das características dos dados de treinamento para produzir um modelo. Ou seja, várias características do conjunto de treinamento são irrelevantes ou redundantes. O problema é que estas características geram um grande custo computacional em todo o processo de classificação. Outro problema é que características irrelevantes podem gerar um superajuste do modelo aos dados de treinamento [Aka73]. Menos características melhoram a capacidade de generalização dos modelos.

A abordagem força-bruta para seleção de características testa todas as combinações possíveis de características e escolhe o melhor subconjunto. Obviamente o custo computacional é muito alto. As abordagens mais utilizadas usam algum tipo de escolha gulosa. O algoritmo seleção adiante sequencial ou, em inglês, *sequential forward selection* (SFS), é um dos mais simples algoritmos gulosos e também um dos mais utilizados. Dado um conjunto de características $X = \{x_i | i = 1 \dots N\}$, o objetivo é encontrar Y_M um subconjunto de X de tamanho M com $M < N$ que maximiza a função objetivo $J(Y)$:

$$Y_M = \{x_{i1}, x_{i2}, \dots, x_{iM}\} = \arg \max_{M, iM} J\{x_i | i = 1 \dots N\}.$$

O Algoritmo 1 mostra o algoritmo SFS em pseudocódigo. No passo 1, o algoritmo inicializa um conjunto vazio de características escolhidas. No próximo passo, ele escolhe a melhor característica não escolhida até então que maximize a função objetivo. O passo 3 atualiza o conjunto de características e o passo 4 retorna ao passo 2 até que não reste mais características a serem escolhidas que melhorem a função objetivo.

2.8 Bases de Dados de Reconhecimento de Objetos

Existem vários bancos de dados focados em mensurar algoritmos de reconhecimentos de objetos. A disponibilidade e a qualidade destas bases de dados são um dos fatores que contribuem para o

```

Input:  $X, J$ 
Output:  $Y_k$ 
Inicialize o conjunto vazio:  $Y_0 \leftarrow \{\}$ ;
 $k \leftarrow 0$ ;
Selecione a próxima melhor característica  $x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$ ;
repeat
|    $Y_{k+1} \leftarrow Y_k + x^+$ ;
|    $k \leftarrow k + 1$ ;
|   Selecione a próxima melhor característica  $x^+ = \arg \max_{x \notin Y_k} J(Y_k + x)$ ;
until  $J(Y_k + x) \leq J(Y_k)$ ;
return  $Y_k$ 

```

Algoritmo 1: Algoritmo SFS

rápido avanço da área [Sze11]. As bases de dados são diferentes em alguns aspectos como o número de classes ou o número de amostras por classe. A base de dados LabelMe [RTMF08] tem uma ferramenta *online* interessante que permite que seus usuários façam novas anotações na base. Ela é composta por imagens (cerca de 12000) e vídeos (cerca de 19000 quadros de vídeos). O desafio VOC Pascal [EGW⁺10] é uma competição anual de reconhecimento de objetos. O VOC Pascal apresenta anualmente uma base de dados e a versão de 2011 apresentou cerca de 11500 imagens rotuladas em 20 classes diferentes. É um dos mais difíceis bancos de dados de reconhecimento de objetos e têm se tornado padrão na comparação de algoritmos de reconhecimento de objetos. A Figura 1.2(b) mostra uma imagem para cada uma das 20 categorias da base de dados VOC Pascal 2011. A base de dados Caltech-101 [FFFP06] tem 101 categorias diferentes. Foi um dos primeiros bancos de dados com um número grande de categorias (mais de 100). O principal objetivo é testar algoritmos de categorização de objetos em situações com um número grande de classes. A base de dados Caltech-256 [GHP07] é uma evolução natural do Caltech-101 com um número maior de classes, imagens mais difíceis e um número maior de imagens por categoria. A base de dados SUN [XHE⁺10] foca, especificamente, em cenários diferentes: naturais, exterior ou interior. Tem quase 900 categorias e 13000 imagens diferentes. É utilizada principalmente para testar algoritmos de categorização de cenas. O ImageNet [DDS⁺09] é a maior base de dados de reconhecimento de objetos disponível atualmente. Ela usa uma estrutura hierárquica de um banco de dados léxico, o WordNet [Fel98]. Ela possui mais de 12 milhões de imagens em quase 18000 categorias diferentes. Por causa de sua estrutura hierárquica, alguns ramos da base ImageNet já foram testados em algoritmos de categorização refinada. A base de dados Caltech-UCSD Birds-200 2011 [WBW⁺11] é a principal base de dados especializada em categorização refinada e é descrita em mais detalhes na Seção 5.6.1.

2.9 Medidas

Existem vários critérios para avaliar o desempenho de um algoritmo de categorização de objetos. A melhor medida para avaliar um algoritmo vai depender de algumas características do próprio algoritmo, como por exemplo se é feita uma classificação binária ou multiclasse e se o retorno do algoritmo é um rótulo, uma lista ou um conjunto de notas que definem a probabilidade da amostra testada pertencer a cada uma das classes da base de dados. A disponibilidade de várias bases de dados juntamente com seus gabaritos (*ground-truths*) facilita o processo de mensuração dos algoritmos de categorização de objetos avaliados.

Utilizando um classificador binário, uma amostra de teste e existindo um gabarito, a amostra pode estar em uma das quatro situações listadas:

- Positivo verdadeiro ou TP, do inglês, *true positive*: uma amostra é corretamente classificada como pertencente ao conjunto positivo.
- Negativo verdadeiro ou TN, do inglês, *true negative*: uma amostra é corretamente classificada

como pertencente ao conjunto negativo.

- Positivo falso ou FP do inglês *false positive*: a amostra é incorretamente classificada como pertencente ao conjunto positivo.
- Negativo falso ou FN do inglês *false negative*: uma amostra é incorretamente classificada como pertencente ao conjunto negativo.

Feitas estas medidas com o conjunto de amostras de teste, é possível calcular algumas medidas. A primeira delas é a *acurácia* (acc). Acurácia é o percentual de classificação correta total do conjunto de amostras (conjunto positivo e conjunto negativo) e é calculado pela seguinte fórmula:

$$acc = \frac{TP + TN}{TP + TN + FP + FN},$$

a principal desvantagem desta medida é o fato dela ser muito afetada pelos resultados no conjunto negativo. Suponha uma situação com 10 amostras no conjunto positivo e 990 amostras no conjunto negativo, e que o algoritmo avaliado retorne sempre que a amostra não pertença ao conjunto positivo em todas as 1000 amostras avaliadas. A acurácia nesta situação seria de 99% e daria a falsa impressão de se tratar de um bom algoritmo apesar de ter classificado incorretamente todas as amostras do conjunto positivo.

As próximas medidas tentam avaliar o algoritmo desprezando o peso de classificar corretamente o conjunto negativo e com isto dar maior ênfase na classificação correta do conjunto positivo e em classificações erradas. A *precisão* (prec) mede a fração das amostras marcadas como positivas pelo algoritmo que realmente pertencem ao conjunto positivo:

$$prec = \frac{TP}{TP + FP}.$$

A *recuperação* (rec) mede a fração do conjunto positivo que foi encontrado pelo algoritmo:

$$rec = \frac{TP}{TP + FN}.$$

A precisão mede a confiança que se tem no algoritmo. Qual o percentual das amostras que ele marcou como positivas que realmente são amostras positivas segundo o gabarito. Já a recuperação mostra qual o percentual do total de amostras positivas do gabarito foram reconhecidas pelo algoritmo. Geralmente existe um dilema entre estas duas medidas. Ao escolher uma parametrização que privilegie uma das duas medidas, a outra, quase certamente, será prejudicada. A Figura 2.7 mostra o dilema entre as duas medidas, através da *curva precisão × recuperação*. É um gráfico utilizando as medidas precisão e recuperação. O gráfico ideal seria aquele com precisão igual a 1 e recuperação igual a 1. Ou seja, todos os valores que o algoritmo classificou como conjunto positivo são realmente pertencentes ao conjunto positivo e nenhuma amostra foi classificada incorretamente. O gráfico usual mostra o dilema entre a precisão e a recuperação. Geralmente, ao obter uma precisão alta, a recuperação vai ser baixa (Ponto A da figura). Ou seja, todos os valores marcados pelo algoritmo como pertencentes ao conjunto positivo realmente são do conjunto positivo. Por outro lado, várias amostras do conjunto positivo não foram marcadas como tal. Na outra ponta, ao se obter uma recuperação alta onde todas as amostras do conjunto positivo foram recuperadas, a precisão geralmente é baixa (Ponto B da figura). Várias amostras foram incorretamente marcadas como pertencentes ao conjunto positivo: aumentando o número de falsos-positivos e diminuindo a precisão.

O método para traçar o gráfico de precisão × recuperação é o seguinte: o algoritmo retorna para cada amostra a probabilidade daquela amostra pertencer ao conjunto positivo. Ordena-se as probabilidades da saída do algoritmo em ordem decrescente. Para cada amostra que realmente pertence ao conjunto positivo, calcula-se o valor da precisão e da recuperação. Divide-se a recuperação em intervalos regulares, geralmente 10, $r_j \in \{0,0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1,0\}$. A precisão

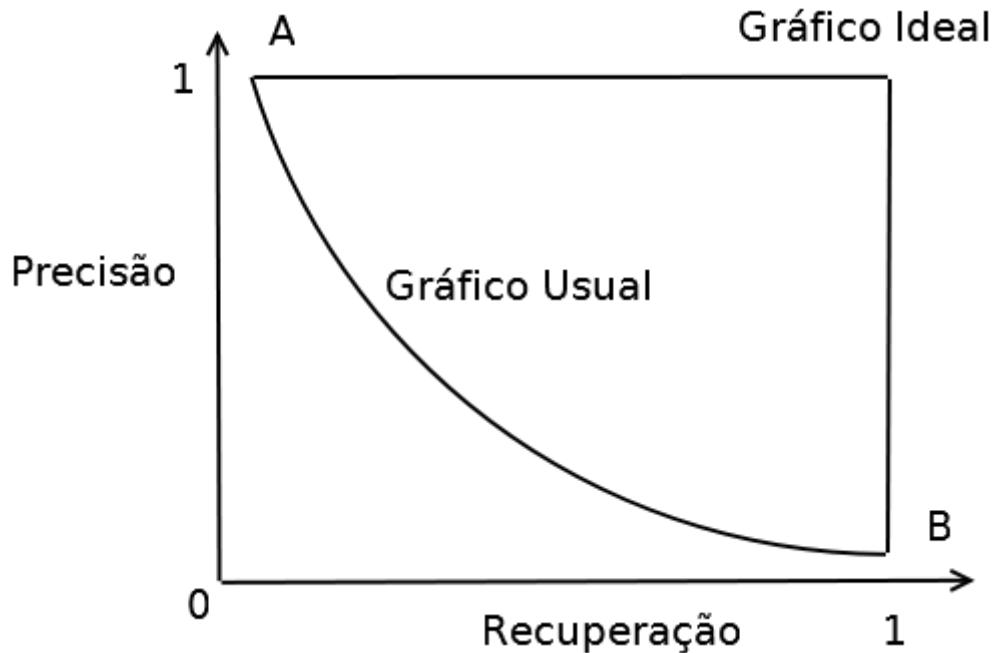


Figura 2.7: Gráfico precisão × recuperação.

é então calculada para cada r_j seguindo a seguinte fórmula:

$$Prec(r_j) = \max_{r_j \leq r \leq r_{j+1}} Prec(r),$$

ou seja, a precisão no ponto r_j é a maior precisão calculada entre r_j e r_{j+1} .

Uma outra medida é a *precisão média* que é a média das precisões obtidas em cada ponto a que uma amostra pertencente ao conjunto positivo é retornada. Uma maneira de calcular a precisão média ou AP do inglês *average precision* é calcular a área sob a curva precisão × recuperação. Este é o método utilizado no desafio VOC Pascal [EGW⁺10].

Tirando-se a média das precisões médias, geralmente obtidas em classes diferentes de uma base de dados, temos uma outra medida. Também conhecida pela sigla em inglês MAP de *Mean Average Precision*.

Em uma classificação multiclasses, quando o algoritmo retorna apenas o rótulo da classe mais provável, pode-se mostrar o resultado utilizando uma matriz de confusão. Matriz de confusão é uma tabela que facilita a visualização do desempenho do algoritmo. A matriz é quadrada, possuindo uma linha e uma coluna para cada classe possível do banco de dados utilizado. As colunas representam as classes preditas pelo algoritmo e as linhas as classes do gabarito. Se uma amostra pertence à classe 1 do gabarito e foi classificada como pertencendo à classe 2 pelo algoritmo, a posição linha 1, coluna 2 é incrementada em 1. A diagonal da matriz mostra os acertos do algoritmo e as posições fora da diagonal, os erros. A Tabela 5.16 mostra um exemplo de matriz de confusão utilizada nos experimentos deste trabalho.

2.10 Morfologia Matemática

A granulometria morfológica foi proposta por [Mat75] para caracterizar imagens granulares. A ideia é pensar na imagem como uma coleção de grãos que são peneirados por uma coleção de peneiras [BDFB⁺01]. A primeira peneira possui uma malha com furos bem pequenos. Assim, apenas grãos que tenham tamanho e forma adequados irão passar pela malha da primeira peneira. As próximas peneiras possuem malhas com formato similar e tamanhos maiores e, sendo assim, à medida que o tamanho dos furos nas malhas das peneiras aumentam, mais grãos passarão pela malhas. Este processo continua até que não restarão grãos na coleção original [DL03]. A quantização

da taxa com que os grãos passam pela malha das peneiras produz uma distribuição numérica que caracteriza a textura e formato da imagem [New91]. Granulometrias são implementadas por aplicações sucessivas de aberturas morfológicas usando uma sequência de elementos estruturantes convexos crescentes como círculos, quadrados ou losangos.

A Morfologia Matemática (MM) é um ramo da análise não-linear de imagens desenvolvida inicialmente por [Mat75] e [Ser82]. Eles trabalharam com mineralogia e petrografia na *Ecole des Mines de Paris*, França, na década de 60. O objetivo de seus trabalhos eram caracterizar propriedades físicas de alguns materiais. Eles desenvolveram uma análise quantitativa de imagens que mais tarde passou a ser conhecida como Morfologia Matemática. A MM também pode ser vista como uma sólida teoria algébrica que estuda transformações em reticulado completo [BB93]. A ideia básica é inspecionar as estruturas geométricas presentes em uma imagem, medindo como pequenos padrões geométricos se casam ou não na imagem principal [DL03]. Variando a forma, tamanho e a localização destes padrões, chamados de elementos estruturantes, é possível extrair características numéricas úteis.

A Seção 2.10.1 introduz alguns definições necessárias à definição de granulometria no arcabouço da MM.

2.10.1 Definições Preliminares

Seja B subconjunto de \mathbb{E} , b uma função de B em \mathbb{Z} , b é chamado de função estruturante e B de elemento estruturante (EE) que é um pequeno conjunto usado para inspecionar imagens [Soi02]. Um EE de duas dimensões é chamado de EE plano e um EE com 3 dimensões é chamado de não-plano ou volumétrico. A aplicação de EE planos é mais intuitiva e o custo computacional é menor que a aplicação de EE volumétricos. EE são definidos pelo seu formato, tamanho e um ponto que define a origem do EE e EE volumétricos possuem, ainda, pesos ou níveis de cinza.

São apresentados alguns conceitos básicos necessários às definições das operações morfológicas. Para as definições que seguem, assume-se que $\mathbb{E} \in \mathbb{Z} \times \mathbb{Z}$, $K \in \mathbb{Z}^+$, $f, g \in K^\mathbb{E}$ e $(x, y) \in \mathbb{E}$.

Seja $h = (h_x, h_y) \in \mathbb{E}$ um vetor, a *translação horizontal* de f por h , f_h , é definida por:

$$f_h(x, y) = f(x - h_x, y - h_y). \quad (2.1)$$

A *translação vertical* de f por $v \in K$, denotada $f + v$, é definida por:

$$(f + v)(x, y) = f(x, y) + v. \quad (2.2)$$

Denota-se por $Dom(f)$ e $Dom(g)$ os domínios das funções f e g , respectivamente.

Sejam $f, g \in K^\mathbb{E}$, diz-se que $f \leq g$:

- se $f(x, y) \leq g(x, y) \forall (x, y) \in Dom(f) \cup Dom(g)$,
- se $(x, y) \in Dom(f)$, mas $(x, y) \notin Dom(g) \Rightarrow g(x, y) = -\infty$,
- se $(x, y) \in Dom(g)$ e $(x, y) \notin Dom(f) \Rightarrow f(x, y) = -\infty$.

O *ínfimo* das funções f e g é uma função em $K^\mathbb{E}$ definida por:

$$(f \wedge g)(x, y) = \min\{f(x, y), g(x, y)\}, \text{ para } (x, y) \in \mathbb{E}. \quad (2.3)$$

A definição dual do ínfimo é o *supremo* das funções f e g é uma função em $K^\mathbb{E}$ definida por:

$$(f \vee g)(x, y) = \begin{cases} \max\{f(x, y), g(x, y)\} & \text{se } (x, y) \in Dom(f) \cap Dom(g), \\ f(x, y) & \text{se } (x, y) \in Dom(f) \text{ e } (x, y) \notin Dom(g), \\ g(x, y) & \text{se } (x, y) \in Dom(g) \text{ e } (x, y) \notin Dom(f), \\ \text{não definido} & \text{se } (x, y) \notin Dom(f) \cup Dom(g). \end{cases} \quad (2.4)$$

A *negação* de uma função f é denotada por f^c e é definida por:

$$f^c(x, y) = -f(x, y), \text{ para } (x, y) \in Dom(f). \quad (2.5)$$

A *reflexão* \check{f} de uma função f é dada por:

$$\check{f}(x, y) = f(-x, -y), \text{ para } (x, y) \in Dom(f). \quad (2.6)$$

A *subtração com saturação* entre duas funções f e g é dada por:

$$(f - g)(x, y) = \begin{cases} f(x, y) - g(x, y) & \text{se } f(x, y) \leq g(x, y), \\ 0 & \text{caso contrário.} \end{cases} \quad (2.7)$$

A *adição com saturação* de duas funções f e g é definida como:

$$(f + g)(x, y) = \begin{cases} f(x, y) + g(x, y) & \text{se } f(x, y) + g(x, y) < k, \\ k - 1 & \text{se } f(x, y) + g(x, y) \geq k, \end{cases} \quad (2.8)$$

onde $k - 1$ é o maior valor do contradomínio de f e g .

A *diferença simétrica* entre as funções f e g é denotada por:

$$f(x, y) \approx g(x, y) = (f(x, y) - g(x, y)) \vee (g(x, y) - f(x, y)). \quad (2.9)$$

Seja B um subconjunto de \mathbb{E} e b uma função de B em \mathbb{Z} , B é chamado de *elemento estruturante* e b de *função estruturante*.

Para definir dilatação e erosão de uma imagem digital f (definição na Seção 2.1) por um elemento estruturante b , é necessário definir algumas novas operações. Seja $\dot{+}$ uma operação definida em $K \times \mathbb{Z}$ para \mathbb{Z} , para, qualquer $t \in K$ e $v \in \mathbb{Z}$:

$$t \dot{+} v = \begin{cases} 0 & \text{se } t = 0, \\ 0 & \text{se } t > 0 \text{ e } t + v \leq 0, \\ t + v & \text{se } t > 0 \text{ e } 0 \leq t + v \leq k, \\ k & \text{se } t > 0 \text{ e } t + v > k. \end{cases} \quad (2.10)$$

De maneira análoga, pode-se definir a operação $\dot{-}$ como:

$$t \dot{-} v = \begin{cases} 0 & \text{se } t < k \text{ e } t - v \leq 0, \\ t - v & \text{se } t < k \text{ e } 0 \leq t - v \leq k, \\ k & \text{se } t < k \text{ e } t - v > k, \\ k & \text{se } t = k. \end{cases} \quad (2.11)$$

Com as duas novas operações é possível definir as operações de erosão e dilatação. A erosão da imagem f pelo elemento estruturante g é definida como:

$$(f \ominus g)(x, y) = \min\{f(z) \dot{-} g_{(x,y)}(z) \mid z \in Dom(g_{(x,y)}) \cap Dom(f)\}. \quad (2.12)$$

A dilatação de f pelo elemento estruturante g é definida como:

$$(f \oplus g)(x, y) = \max\{f(z) \dot{+} g_{(x,y)}(z) \mid z \in Dom(g_{(x,y)}) \cap Dom(f)\}. \quad (2.13)$$

A abertura morfológica é a composição trivial de erosão e dilatação. Ela é implementada erodindo uma imagem e usando o mesmo EE para dilatá-la. Algumas partes da imagem são recuperadas, porém, partes, em que o EE não se encaixam, são eliminadas pela erosão e não são mais recuperadas. A abertura é definida como:

$$f \circ b = (f \ominus b) \oplus b. \quad (2.14)$$

Fechamento é também uma composição de erosão e dilatação e é a operação dual da aber-

tura [DL03]. A dilatação é seguida por uma erosão usando o mesmo EE e é definida por:

$$f \bullet b = (f \oplus b) \ominus b. \quad (2.15)$$

A dilatação de uma imagem por um EE que contém a sua própria origem fará com que esta imagem se expanda continuamente a cada nova dilatação. Por vezes, pode ser interessante limitar este processo de expansão e para isto pode-se utilizar um marcador que limite a expansão da imagem original [DL03]. A dilatação de f pelo EE b condicionada ao marcador g é definida por:

$$f \oplus_g b = (f \oplus b) \wedge g. \quad (2.16)$$

A seqüência de n dilatações condicionais utilizando o marcador g é chamada de *dilatação geodésica de tamanho n* e é definida por:

$$(f \oplus_g b)^n = \underbrace{(((f \oplus_g b) \oplus_g b) \oplus_g \dots \oplus_g b)}_{n \text{ vezes}}. \quad (2.17)$$

Aplicando dilatações condicionais até a estabilidade, temos a operação de *reconstrução inf-geodésica*:

$$f \triangle_g b = (f \ominus_g b)^\infty, \quad (2.18)$$

onde ∞ é usado para indicar a aplicação do operação até a estabilidade.

De modo dual, define-se *erosão condicional*, *erosão geodésica de tamanho n* e *reconstrução sup-geodésica* por, respectivamente:

$$f \ominus_g b = (f \ominus b) \vee g, \quad (2.19)$$

$$(f \ominus_g b)^n = \underbrace{(((f \ominus_g b) \ominus_g b) \ominus_g \dots \ominus_g b)}_{n \text{ vezes}}, \quad (2.20)$$

$$f \nabla_g b = (f \oplus_g b)^\infty. \quad (2.21)$$

O gradiente de uma imagem pode ser calculado utilizando as operações morfológicas básicas. O gradiente morfológico externo é calculado pela seguinte fórmula:

$$\text{grad}_B^{\text{ext}}(f) = (f \oplus g) - f, \quad (2.22)$$

onde b é um EE, geralmente circular, que contém a origem.

2.10.2 Definição de Granulometria Morfológica

Granulometria é a família de transformações de imagens $\{\psi_t\}$ dependentes do parâmetros t , com $t > 0$ e as seguintes propriedades são satisfeitas [Vin00, BDS00]:

- $\forall t \geq 0, \psi_t$ é crescente. ψ_t é crescente se $X \subset Y$, então $\psi_t(X) \subset \psi_t(Y)$;
- $\forall t \geq 0, \psi_t$ é antiextensivo. ψ_t é antiextensivo se $\psi_t(X) \subset X$;
- $\forall t \geq 0, v \geq 0, \psi_t \psi_v = \psi_v \psi_t = \psi_{\max(t,v)}$;
- Da última propriedade, tem-se a transformação idempotente: $\psi_t \psi_t = \psi_t$;
- Invariante a translação: $\psi(X + a) = \psi(X) + a$;
- Para $t = 0, \psi_t(X) = X$.

Em [Mat75], caracterizou-se granulometrias baseadas em aberturas morfológicas. Os EEs usados na família de aberturas devem ser convexos.

$$\psi_t(X) = X \circ tB, \quad (2.23)$$

onde B é um EE convexo e é chamado de gerador da granulometria [DL03], $tB = \{t\mathbf{b} | \mathbf{b} \in B\}$ e $t \geq 0$. Se $t > v > 0$, então tB é vB -abertura, ou, $tB \circ vB = tB$. Então, para uma imagem X , $(X \circ tB) \subset (X \circ vB)$. Enquanto t aumenta, $X \circ tB$ diminui e para t suficientemente grande, $X \circ tB = \emptyset$.

Seja $\Omega(t)$ a área, no caso binário, ou o volume, no caso de níveis de cinza, que permanece na imagem após a aplicação da abertura por tB . Então, para imagem X , $\Omega(0) = v[X]$, onde $v[X]$ denota a área ou volume da imagem X se X é uma imagem binária ou uma imagem em níveis de cinza, respectivamente. $\Omega(t)$ é chamada *distribuição de tamanhos* [DL03, BDFB⁺01]. A *distribuição normalizada de tamanhos* mantém os valores da função entre 0 e 1 e mede quanto da área ou volume totais sumiram da imagem original até t : $\Phi(t) = 1 - \frac{\Omega(t)}{\Omega(0)}$. A derivada de $\Phi(t)$ é uma função densidade probabilidade e é conhecida como *padrão de espectro*. O padrão de espectro é importante para caracterizar uma imagem. A granulometria pode ser calculada globalmente ou localmente usando uma janela [DL03]. Para o caso discreto, o padrão de espectro $PS(t)$ pode ser calculado como:

$$PS(t) = \Phi(t+1) - \Phi(t). \quad (2.24)$$

[DKP89] usou granulometria para segmentar imagens de sensoriamento remoto. Classificação de textura foram feitas utilizando granulometria em vários trabalhos, como em [New91, KGM11]. Detecção de rodovias em imagens ortoretificadas de sensoriamento remoto foram feitas em [ZMB99]. Em [BDFB⁺01], foram feitas classificações de diferentes tipos de solos, usando características extraídas da caracterização dos grãos utilizando granulometria morfológica. Neste trabalho, é proposto um novo descritor que utiliza a granulometria morfológica de um modo diferente, aplicando-a no mapa de bordas e construindo modelo de objetos para ser aplicado em categorização de objetos. Este descritor é descrito na Seção 3.2.

Capítulo 3

Desenvolvimento

Este capítulo descreve as técnicas propostas neste trabalho. A primeira destas técnicas, descrita na Seção 3.1, é um incremento à técnica do Casamento de Pirâmides Espaciais ou SPM da sigla em inglês de *Spatial Pyramid Matching*. Neste trabalho, algumas restrições propostas por [LSP06], no artigo original, foram relaxadas, o que torna a técnica mais adaptável a uma categoria específica durante a etapa de treinamento. A Seção 3.2 descreve um descritor baseado na aplicação da granulometria morfológica no mapa de bordas de uma imagem. Modelos para aplicações de categorização de objetos podem ser criados utilizando este descritor com desempenho similar aos descritores tradicionais da área. Outros dois descritores também são propostos: um baseado em um histograma normalizado de padrões 3×3 e o outro gerando uma coleção de imagens binárias a partir da imagem original em níveis de cinza e aplicando granulometria morfológica em cada uma destas imagens binárias. Finalmente, a Seção 3.3 descreve as Pirâmides de Regiões Circulares Concêntricas (PRCC) que é uma nova maneira de dividir espacialmente uma imagem mantendo o ganho de desempenho das pirâmides espaciais retangulares e aumentando a invariância à rotação dos objetos presentes na imagem. O conteúdo desta seção foi publicado em [LH13b]. Os conteúdos das seções 3.1 e 3.2 foram publicados em [LH13a, LH11b].

3.1 Quantização Espacial Flexível

As pirâmides espaciais de [LSP06] (Seção 2.4) introduziram informações espaciais à abordagem do saco de palavras (Seção 2.3) incrementando o desempenho final da técnica em aplicações de reconhecimento de objetos. Neste trabalho, é proposta uma nova quantização espacial sem algumas restrições do artigo original.

A primeira restrição eliminada é em relação aos níveis e ao número de regiões em cada nível. Em [LSP06], foi proposta uma divisão onde cada região de um nível será dividida em quatro regiões no próximo nível. Em uma pirâmide espacial com 3 níveis, o primeiro nível começa com apenas uma região. O segundo nível terá quatro regiões e o terceiro nível terá 16 regiões. A Quantização Espacial Flexível (QEF), proposta aqui, não possuem este tipo de restrição. A escolha do número de níveis e o número de regiões em cada nível é livre. As divisões em cada uma das duas dimensões da imagem em um determinado nível também não precisam ser idênticas. Assim, pode-se ter níveis com divisões como: 3×1 , 2×4 , 1×2 , entre outras. A motivação para este relaxamento nas restrições originais de divisão de níveis em regiões é tornar a quantização espacial mais adaptável ao formato dos objetos. Várias configurações de QEFs podem ser tentadas durante as etapas de treinamento e validação. A ideia é que a configuração da QEF que obtenha o melhor desempenho durante a validação de parâmetros seja utilizada na criação do modelo final desta categoria de objetos. A Figura 3.1 mostra imagens de duas classes da base de dados Caltech-101. QEFs com número diferente de níveis e número diferente de divisões em cada nível são utilizadas em cada uma das classes. A Figura 3.1(a), da classe *dollar_bill*, mostra uma nota de um dólar. A imagem mostra uma divisão espacial com um nível e 1×3 regiões neste nível. Pode-se notar que as regiões 1 e 3 da nota possuem certa simetria. A região 2 (central) possui algumas particularidades, como o



(a) Imagem da classe *dollar_bill* dividida por uma quantização espacial com 1 nível e regiões 1×3 .



(b) Imagem da classe *faces_easy* com uma divisão espacial de 2 níveis: 1×2 e 3×1 .

Figura 3.1: A figura mostra imagens de duas classes da base de dados Caltech-101 com divisões espaciais que exploram a geometria dos objetos.

desenho do rosto de George Washington. Esta divisão pode auxiliar na construção de modelo para a classe *dollar_bill*. O outro exemplo (Figura 3.1(b)) mostra uma imagem da classe *faces_easy* com uma QEF de 2 níveis. O primeiro nível mostra regiões 1×2 e o segundo nível mostra regiões 3×1 . Nota-se que as regiões de um nível não precisam ser divisões das regiões de um nível anterior, restrição esta que existia na proposta original. Motivo, também, pelo qual a extensão das pirâmides espaciais, proposta daqui, não podem ser chamadas de *pirâmides*. Não existe necessariamente uma ordem crescente de divisões dos níveis. Assim, preferiu-se adotar o termo *quantização espacial*. Esta divisão espacial da Figura 3.1(b) pode auxiliar a construção de um modelo para a classe, já que o primeiro nível explora a simetria do rosto e o segundo nível apresenta áreas distintas do rosto em cada uma das regiões do nível 2: boca na primeira região, olhos, sombrancelhas e nariz na segunda região e, na última região, a testa e o alto da cabeça.

Outra restrição do artigo original, eliminada desta nova proposta, é em relação ao peso associado com cada nível. Na proposta original, o peso associado ao nível l é $\frac{1}{2^{L-l}}$, onde L é o número do último nível, e $l \in \{0, \dots, L-1\}$. Ou seja, o peso do nível é maior quanto maior for o nível e ele é inversamente proporcional à largura de uma região daquele nível. As QEFs possuem pesos livres associados a cada nível, uma vez que a divisão dos níveis é arbitrária, também é arbitrária a escolha dos pesos associados aos níveis. A única restrição é que a somatória dos pesos seja igual a 1. A ideia é mais uma vez utilizar o período de treinamento e validação para obter os melhores pesos

associados aos níveis da quantização espacial escolhida. A QEF foi proposta para ser utilizada em um modelo discriminante.

O artigo original propõe um esquema de casamento piramidal entre duas imagens onde os casamentos entre palavras visuais que ocorrem um nível maior são descontados dos casamentos de um nível menor. Isto é feito para evitar uma contagem dupla deste casamento. Ainda, como o peso aplicado a um nível mais profundo é maior que o de um nível menor, o casamento que ocorre em um nível maior ocorre dentro de uma célula de menor largura então seria mais importante. O peso maior aplicado no nível mais inferior evidenciaria esta maior importância do casamento em níveis maiores. Esta proposta de casamento não é utilizada nas QEFs.

As QEFs utilizam, dentro de um nível, regiões que podem se sobrepor em suas fronteiras. A quantificação desta sobreposição é controlada por um parâmetro que determina o número de pixels que uma região irá avançar sobre a outra região adjacente. Este número de pixels será o mesmo para as quatro direções. A motivação para utilizar esta sobreposição de regiões é diminuir a quebra de estruturas importantes para discriminação de imagens que, por acaso, estejam sobre a fronteira de duas regiões adjacentes. A Figura 3.2 mostra uma imagem da classe *umbrella* da base de dados Caltech-101 no canto superior esquerdo, em seguida, no canto superior direito, é mostrado o mapa de bordas da respectiva imagem. Na parte inferior da imagem, são mostradas duas divisões espaciais do mapa de bordas da imagem original: a primeira, do lado esquerdo, mostra uma divisão em regiões 3×1 sem sobreposição de regiões e a segunda, do lado direito, mostra a mesma divisão com uma sobreposição de 30 pixels. Ou seja, cada região avança 30 pixels para dentro da região adjacente. Pode-se notar que principalmente a borda inferior do guarda-chuva ficou mais identificável na divisão espacial com sobreposição. Houve uma quebra da região de fronteira do guarda-chuva na divisão espacial que não teve sobreposição de regiões. A ideia é que a divisão espacial com sobreposição preserva melhor as regiões de fronteira entre regiões adjacentes e com isto ajuda a construir modelos de objetos mais eficazes. A quebra abrupta nas regiões fronteiriças prejudica a interpretação das formas nestas regiões. As QEFs são utilizadas em modelos discriminantes do saco de palavras.

A implementação da QEF foi feita de modo que ela é independente do descritor. Os experimentos descritos na Seção 5.3 utilizam diferentes descritores em conjunto com as QEFs. Estes experimentos também mostram que existe uma configuração ideal de quantização espaciais para cada classe. Esta configuração deve ser encontrada durante o período de treinamento e validação dos parâmetros. Em [LH13a], foi publicado um trabalho descrevendo a QEF e sua aplicação em problemas de categorização de objetos e categorização refinada de objetos.

O artigo original utiliza a intersecção de histograma na construção da função núcleo a ser utilizado no classificador SVM. No trabalho proposto, o descritor final é construído concatenando os descritores extraídos em cada região de cada nível. Este descritor final é aplicado a um classificador SVM com função núcleo qui-quadrado que é reportada pela literatura como a função núcleo mais utilizada em aplicações de reconhecimento de objetos [RTG00].

A Seção 5.2 mostra os experimentos realizados comparando as QEFs com as Pirâmides Espaciais tradicionais propostas pelo trabalho original em uma aplicação de categorização de objetos e uma base de dados pública.

O Algoritmo 2 mostra o algoritmo em pseudocódigo da QEF. O algoritmo começa inicializando *Vetor_Final* como vazio. Na próxima linha, inicia-se um laço entre todos os níveis da QEF. Para cada nível, calcula-se o número de células nas dimensões 1 (altura) e 2 (largura). Divide-se a altura da imagem em *num_celulas1* + 1 pontos igualmente espaçados. O mesmo é feito para a largura. Para cada célula do nível *l*, faz-se um corte retangular na imagem, com o tamanho da célula correspondente. Esta nova imagem é passada como parâmetro para a função que calcula o descritor (função que também foi recebida como parâmetro). Ao resultado é aplicado o peso correspondente do nível *l* e concatenado ao vetor final. Ao fim dos laços, a variável *Vetor_Final* é retornada.

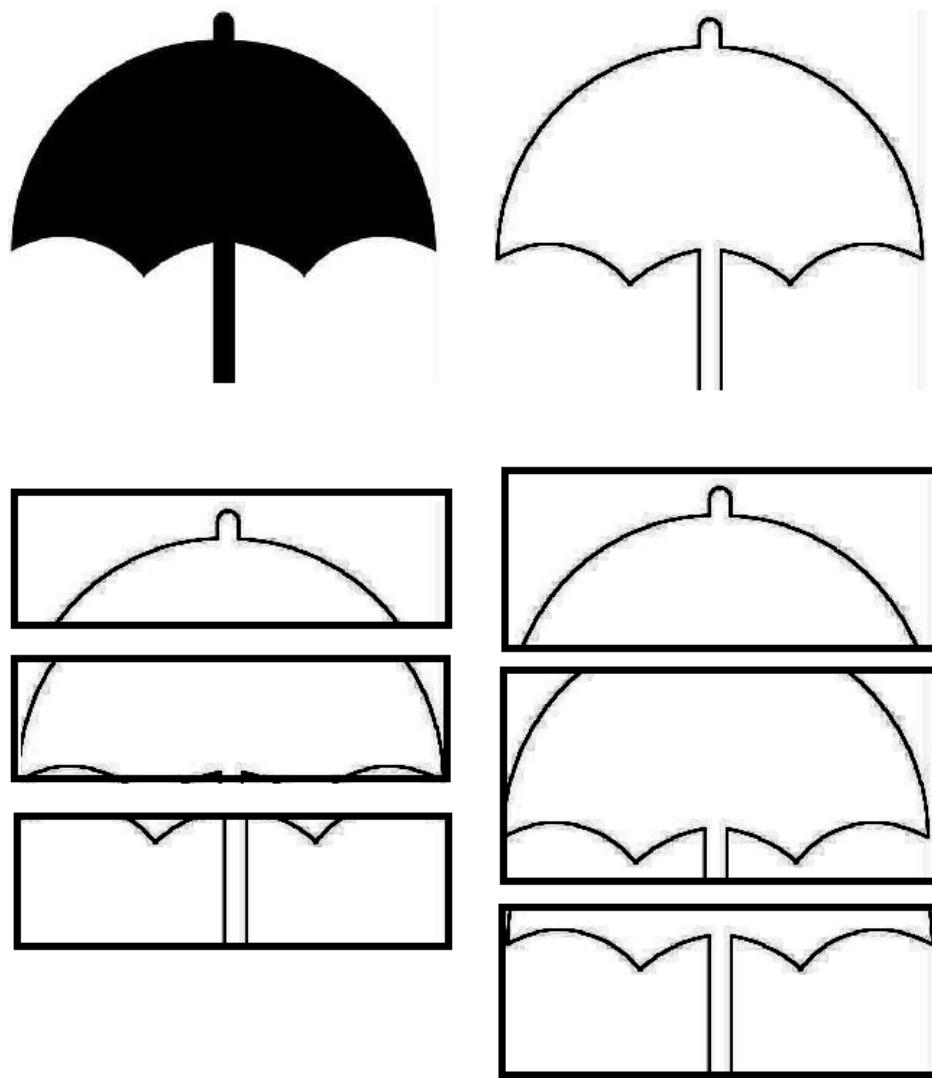


Figura 3.2: A figura mostra a imagem de um guarda-chuva (classe umbrella da base de dados Caltech-101), o mapa de bordas e duas divisões espaciais. Uma sem utilizar sobreposição de regiões e outras utilizando-a.

3.2 Descriptor GRABED

É proposto, neste trabalho, um novo descriptor que usa granulometria morfológica aplicada ao mapa de bordas. O descriptor é chamado de GRABED, do inglês *GRAnulometry Based-descriptor applied in map of EDges* ou descriptor baseado em granulometria aplicado ao mapa de bordas. Granulometria morfológica (Seção 2.10.2) é geralmente utilizada em imagens binárias ou em níveis de cinza para medir a distribuição de formas e tamanhos [Lef07]. Por ser aplicado ao mapa de bordas, o descriptor mede a orientação das bordas e também sua extensão ao longo de uma orientação. Esta última característica o difere do HOG e do PHOG (Seção 2.6.4), por exemplo.

Para computar o descriptor, o primeiro passo é produzir um conjunto de famílias de EEs. Cada família de EEs é usada em uma sequência de aberturas morfológicas produzindo um padrão de espectro (Seção 2.10.2). A coleção de padrões de espectro é utilizada para caracterizar as bordas (tamanhos e orientação). Uma família de EEs é composta por EEs lineares com a mesma orientação e tamanhos crescentes. Cada família produz um padrão de espectro diferente. Um parâmetro, R , define quantos ângulos diferentes são utilizados para rotacionar o EE linear entre 0° e 180° . Por exemplo, se $R = 10$, o conjunto de ângulos de rotação é $\{0^\circ, 18^\circ, 36^\circ, 54^\circ, 72^\circ, 90^\circ, 108^\circ, 126^\circ, 144^\circ, 162^\circ\}$. O parâmetro L_0 define o tamanho em pixels do primeiro EE de cada família de EEs. O parâmetro

```

Input: IMG, QEF, Peso, Sobreposicao, Ptr _ Descritor
Output: Vetor _ Final
Vetor _ Final  $\leftarrow \emptyset$ ;
for l  $\leftarrow 1$  to numero_niveis(QEF) do
    num_celulas1  $\leftarrow$  numero_celulas(QEF, l, 1);
    num_celulas2  $\leftarrow$  numero_celulas(QEF, l, 2);
    lista_pontos1  $\leftarrow$  espacamento_linear(altura(IMG), num_celulas1 + 1);
    lista_pontos2  $\leftarrow$  espacamento_linear(largura(IMG), num_celulas2 + 1);
    for y  $\leftarrow 1$  to num_celulas1 do
        for x  $\leftarrow 1$  to num_celulas2 do
            retangulo  $\leftarrow \{ lista_pontos2(x) - Sobreposicao, lista_pontos1(y) - Sobreposicao, lista_pontos2(x + 1) - lista_pontos2(x) + (2 * Sobreposicao), lista_pontos1(y + 1) - lista_pontos1(y) + (2 * Sobreposicao) \};
            IMG_CROP  $\leftarrow$  corte(IMG, retangulo);
            Vetor  $\leftarrow$  Ptr _ Descritor(IMG_CROP);
            Vetor_Final  $\leftarrow$  Vetor_Final + Peso(l) * Vetor;
        end
    end
end
return Vetor_Final$ 
```

Algoritmo 2: Algoritmo QEF

Max_Axis é usado para computar o tamanho do último EE de cada família. O tamanho do primeiro EE de uma família é L_0 . O próximo EE é duas vezes maior. O terceiro EE é duas vezes maior que o segundo e, assim, sucessivamente. O último EE da família é aquele que possui tamanho em pixels maior que metade do parâmetro *Max_Axis*. Seja *tam(i)* o tamanho do i-ésimo EE da família de EEs, a extensão em pixels do i-ésimo EE linear de uma família é:

$$tam(i) = \begin{cases} L_0 & \text{se } i = 1; \\ 2 * tam(i - 1) & \text{se } i > 1 \text{ e } tam(i - 1) < \frac{Max_Axis}{2}. \end{cases} \quad (3.1)$$

O mapa de bordas pode ser produzido usando-se o detector de bordas Canny ou a abordagem morfológica descrita na Seção 3.2.3. Para cada diferente ângulo de orientação (R ângulos diferentes), o padrão de espectro é calculado de acordo com a Equação 2.24. O descritor final concatena todos os padrões de espectros calculados.

A Figura 3.3 mostra as etapas de cálculo da família de EEs para uma imagem da classe *Faces_easy* da base de dados Caltech-101. A família de EEs são compostos por EEs lineares com 0° de orientação e tamanho inicial de 2 pixels. A Figura 3.3(a) mostra a imagem original e a Figura 3.3(b) mostra o mapa de bordas calculado usando Canny. O total de pixels do mapa de bordas (9470 pixels) é armazenado para ser utilizado no cálculo do padrão de espectro. O primeiro EE tem 2 pixels de tamanho. O resultado da abertura pode ser visto na Figura 3.3(c). O número de pixels restantes nesta etapa é 3892, $PS(1) = \frac{9470 - 3892}{9470} = 0,5890$. A Figura 3.3(d) mostra o resultado da abertura pelo EE linear com 4 pixels de tamanho. O total de 2489 pixels é contados no resultado, $PS(2) = \frac{3892 - 2489}{9470} = 0,1482$. As Figuras 3.3(e), 3.3(f), 3.3(g) e 3.3(h) mostram o resultado da abertura pelo EE linear com tamanho de 8, 16, 32 e 64 pixels, respectivamente, e os padrões de espectro calculados nestes passos são: 0,1653; 0,0700; 0,0099 e 0,0056. O resultado da abertura pelo EE linear de 128 pixels é mostrado na Figura 3.3(i). Nenhum pixel permanece nos resultados. O padrão de espectro é calculado como 0,0120 (a fração dos pixels que foram eliminados nesta etapa) e a última etapa (com EE linear com 256 pixels) não é calculada, neste caso, pois os resultados sempre permanecerão 0. Assim, o algoritmo começa a calcular o padrão de espectro da próxima orientação. O padrão de espectro para esta família de EEs é: $\{0,5890; 0,1482; 0,1653; 0,0700; 0,0099; 0,0056; 0,0120; 0\}$. Pode-se notar que a soma dos valores de

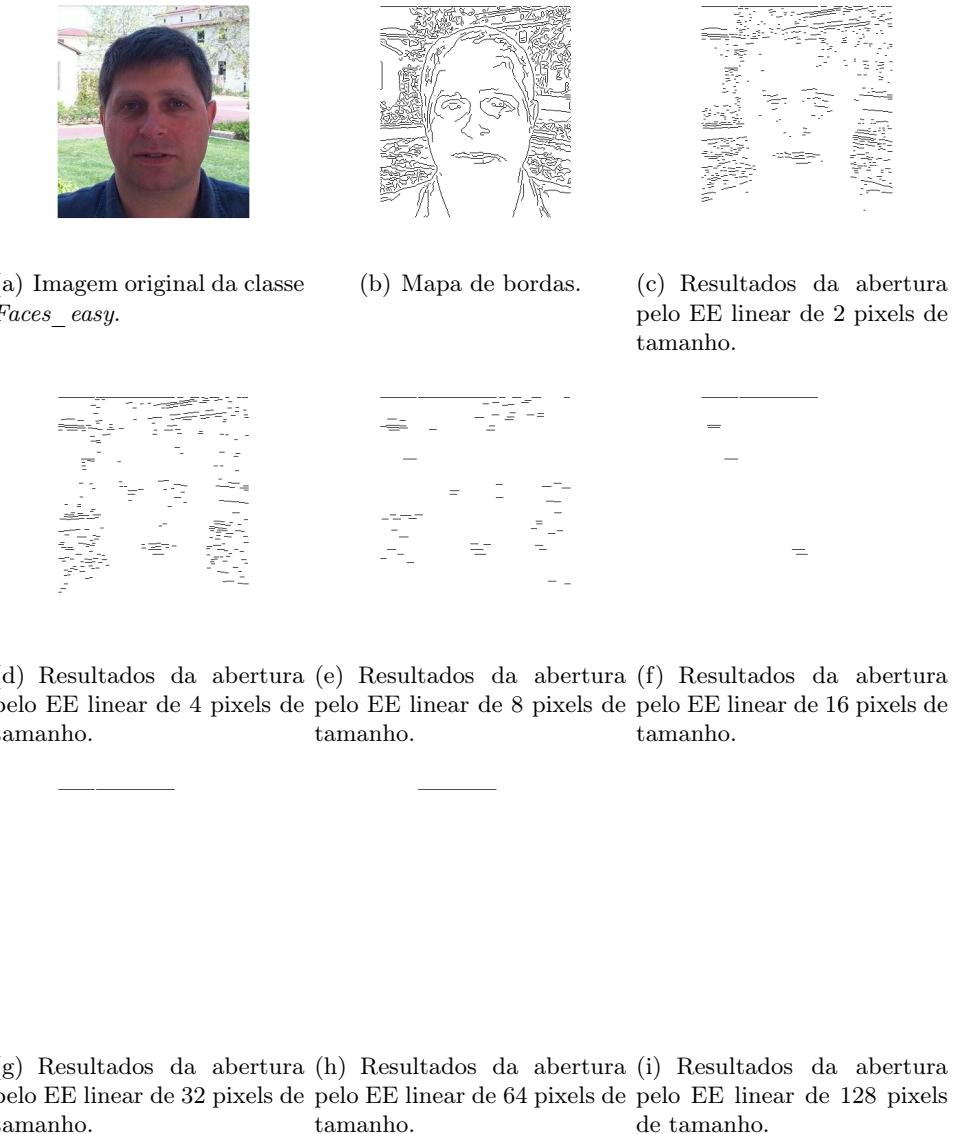


Figura 3.3: Passos para o cálculo do padrão de espectro para uma imagem da classe *Faces_easy*. A família de EEs usada para calcular o padrão de espectro tem 0° de orientação.

um padrão de espectro é igual a 1.

Duas extensões do descritor básico são propostas. A primeira é a utilização de múltiplas escalas. A primeira escala usa a imagem em seu tamanho original e as próximas escalas reduzem o tamanho original da imagem por fator F e assim sucessivamente. A ideia é modelar estruturas morfológicas de diferentes escalas de detalhes. Para cada escala, o descritor explanado no último parágrafo é calculado. O descritor de cada escala é concatenado. A Figura 3.4 mostra a imagem da classe *Leopards* da base Caltech-101 em 3 escalas diferentes. Cada escala tem metade do tamanho em cada dimensão da escala anterior. A outra extensão é o uso em conjunto com as QEFs descritas na Seção 3.1.

A Seção 5.3 descreve os experimentos realizados utilizando o descritor GRABED para modelar categoria de objetos. A descrição do descritor GRABED e os experimentos da Seção 5.3 foram publicados em [LH13a]. A Seção 5.4 descreve o descritor GRABED utilizado em uma combinação de descritores para reconhecimento de instâncias de monumentos. Estes experimentos foram publicados em [LH11b].

O Algoritmo 3 mostra a implementação do descritor em pseudocódigo. Logo no início, a variável PS que armazena o descritor é inicializada com valor vazio. O algoritmo 4 é chamado na

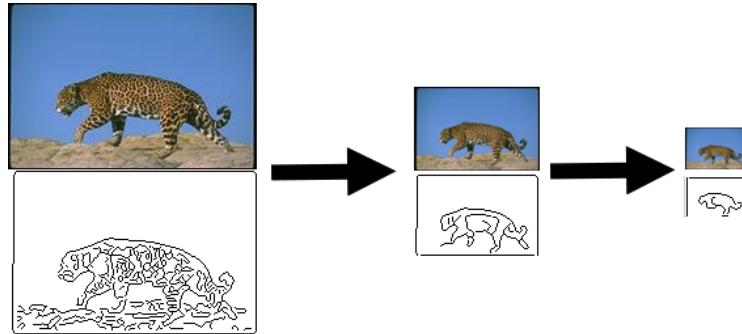


Figura 3.4: A figura mostra um exemplo da imagem da classe Leopards em 3 escalas diferentes.

linha seguinte retornando uma lista de EEs com tamanhos começando em L_0 , aumento em 2 vezes o tamanho do EE anterior até que atinja tamanho maior que metade de Max_Axis e orientações formadas por R divisões entre 0° e 180° . A seguir, para cada escala, calcula-se o mapa de bordas utilizando a função passada como parâmetro. Calcula-se a área inicial deste mapa de bordas armazenando-a na variável *area_inicial*. Recebe-se uma lista de EEs em ordem crescente de tamanho e com uma mesma orientação. Os EEs são aplicados em uma sequência de aberturas. A área da imagem resultante destas aberturas e o percentual de redução em relação a área inicial são, então, calculados. Este percentual é armazenado no vetor de retorno *PS*.

```

Input: IMG, Num_Escalas, F, Max_Axis, L0, R, ptr_borda
Output: PS
PS  $\leftarrow \emptyset$ ;
lista_EEs  $\leftarrow$  GERAR_EEs(Max_Axis, L0, R);
for escala  $\leftarrow 1$  to Num_Escalas do
    IR  $\leftarrow$  redimensionar(IMG, Fescala-1);
    MAPA_BORDAS  $\leftarrow$  ptr_borda(IR);
    area_inicial  $\leftarrow$  area(MAPA_BORDAS);
    area_ant = area_inicial;
    lista_rotacao  $\leftarrow$  conjunto de R ângulos entre  $0^\circ$  e  $180^\circ$ ;
    for angulo  $\in$  lista_rotacao do
        familia_EE  $\leftarrow$  lista ordenada de EEs com orientação angulo;
        for EE  $\in$  familia_EE do
            IO  $\leftarrow$  abertura(MAPA_BORDAS, EE);
            area_temp  $\leftarrow$  area(IO);
            PS  $\leftarrow$  PS +  $\frac{\text{area\_ant} - \text{area\_temp}}{\text{area\_inicial}}$ ;
            area_ant  $\leftarrow$  area_temp;
        end
    end
end
return PS

```

Algoritmo 3: Algoritmo GRABED

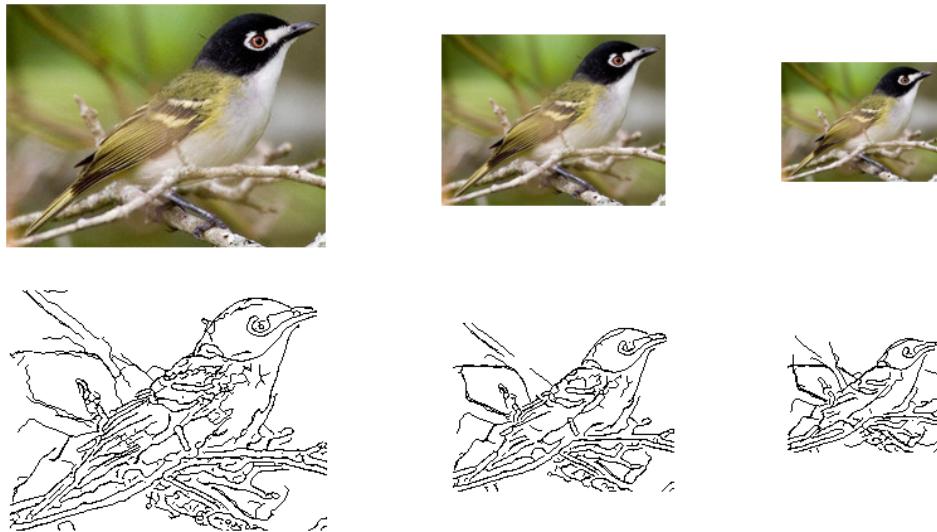
3.2.1 Descriptor PPE

O descriptor PPE, do inglês, *Pyramidal Patterns of Edges*, é um descriptor proposto neste trabalho com similaridades ao descriptor GRABED. O PPE também utiliza uma série de aberturas morfológicas em seu cômputo e ele também é aplicado ao mapa de bordas da imagem. A ideia do descriptor é criar um histograma normalizado de todos os padrões 3×3 pixels que aparecem no mapa de bordas de uma imagem. Este histograma normalizado junto com histogramas de outras

```

Input: Max_Axis,  $L_0$ ,  $R$ 
Output: lista_EEs
tamanhos  $\leftarrow \emptyset$ ;
 $l \leftarrow L_0$ ;
while  $l \leq \text{Max\_Axis}$  do
| tamanhos  $\leftarrow$  tamanhos +  $l$ ;
|  $l \leftarrow l * 2$ ;
end
lista_angulos  $\leftarrow$  conjunto de  $R$  ângulos entre  $0^\circ$  e  $180^\circ$ ;
lista_EEs  $\leftarrow \emptyset$ ;
for angulo  $\in$  lista_angulos do
| for tamanho  $\in$  tamanhos do
| | EE  $\leftarrow$  criar EE em forma de linha com tamanho tamanho e orientação angulo;
| | lista_EEs  $\leftarrow$  lista_EEs + EE;
| end
end
return lista_EEs

```

Algoritmo 4: Algoritmo GERAR_EEs**Figura 3.5:** Imagem da base de dados Caltech-UCSD Birds-200 2011 em 3 escalas diferentes, fator de redução 0,7 e seus respectivos mapas de bordas calculados utilizando o detector de bordas Canny.

imagens com mesmo rótulo seria utilizado para criar um modelo daquela categoria. O descritor utiliza as QEFs descritas na Seção 3.1 e também é aplicado em diferentes escalas, da mesma forma que o descritor GRABED. Um fator F é utilizado para redimensionar a imagem inicial gerando uma imagem em uma nova escala e este mesmo fator é reaplicado gerando uma terceira imagem com F^2 do tamanho da primeira imagem. O parâmetro `num_escalas` controla o número de imagens em escalas diferentes que são utilizadas. Em cada uma dessas imagens, o mapa de bordas é calculado utilizando o detector de bordas Canny [Can86]. A Figura 3.5 ilustra esta etapa mostrando uma imagem da base de dados Caltech-UCSD Birds-200 2011 em 3 escalas diferentes com fator de redução de 0,7 e seus respectivos mapas de bordas.

Um conjunto de EEs é criado representando cada um dos padrões possíveis em um quadrado 3×3 . O total de padrões possíveis é $512 = 2^9$. Para cada mapa de bordas calculado, a área inicial do mapa de bordas é armazenada. A área no mapa de bordas é igual ao número de pixels com valor 1. Uma coleção de aberturas morfológicas são aplicadas ao mapa de bordas utilizando cada um dos 512

EEs. A área do resultado da abertura é calculada. A fração entre área do resultado da abertura e a área inicial é armazenada no vetor de características. Para cada escala, calculam-se as 512 aberturas morfológicas e a aplicação de cada abertura resulta em um valor no vetor de características. Este processo se repete para cada região de cada nível na quantização espacial utilizada. Utilizando, por exemplo, uma QEF com 2 níveis (1×1 no primeiro nível e 3×3 no segundo nível) teremos 10 regiões da quantização espacial \times 3 escalas \times 512, o que dá 15360 dimensões no vetor de características.

Devido ao grande número de dimensões do descriptor final e também à presença de inúmeras características redundantes, uma etapa de seleção de características é utilizada (Seção 2.7.3). As características redundantes são por causa da presença de EEs que possuem basicamente a mesma estrutura. Alguns EEs que produzem características semelhantes:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Utilizando o algoritmo SFS, as 50 melhores características são selecionadas. O número foi pré-determinado para diminuir o custo computacional desta etapa. As 10 características mais discriminantes são listadas abaixo:

$$\begin{array}{ccccc} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

O número de dimensões do descriptor final, utilizando apenas as 50 características selecionadas para o caso da quantização espacial com 2 níveis e 10 regiões, é 10 regiões \times 3 escalas \times 50 características = 1500 dimensões.

3.2.2 PPD

O descriptor PPD, do inglês, *Pyramidal Patterns of Disks*, é proposto neste trabalho e, ao contrário do GRABED e do PPE, não é aplicado ao mapa de bordas da imagem. Como os outros descritores, são utilizadas várias escalas e a quantização espacial flexível. Dois parâmetros, F e *num_escala*, controlam o fator de redução do tamanho da imagem e o número de escalas diferentes a serem utilizadas. Outro parâmetro dita o número de agrupamentos dos valores de níveis de cinza. Este parâmetro divide os valores de níveis de cinza, que vão de 0 a 255, em intervalos de mesmo tamanho. Para cada tamanho de imagem e para cada intervalo de agrupamento dos valores dos pixels, a imagem em níveis de cinza é convertida para binário utilizando o seguinte critério: pixels com valores que estiverem dentro do intervalo são convertidos para 1 e os demais pixels com valores fora do intervalo são convertidos para 0. A esta imagem binária é aplicada uma granulometria morfológica com elemento estruturante em forma de disco. A primeira abertura da granulometria utiliza um EE circular com raio igual ao valor inicial passado por parâmetro. A segunda abertura utiliza um EE circular com raio igual a duas vezes o raio do primeiro. Este processo continua até que o raio dos EEs atinja um valor final, também passado como parâmetro. A cada abertura o percentual da área da imagem, que foi eliminado após a aplicação da abertura morfológica, é armazenada no vetor de características.

Resumindo, aplica-se uma granulometria com EEs circulares em cada uma das imagens binárias geradas a partir da imagem original em níveis de cinza e dos intervalos de valores de níveis de cinza. O descriptor final possui tamanho igual ao número de aberturas morfológicas aplicadas \times o número de intervalos de valores de pixels \times o número de escalas da imagem \times o número de regiões da QEF.

3.2.3 Mapas de Bordas: Canny e a Abordagem Morfológica

O mapa de bordas de uma imagem é necessário para o cálculos dos descritores GRABED (Seção 3.2) e PPE (Seção 3.2.1). A detecção de bordas é um problema fundamental de VC e existem várias abordagens para detectá-las. Segundo [Can86], um bom detector de bordas deve seguir os seguintes critérios:

- *Detecção*: a probabilidade de detectar pontos em bordas reais deve ser maximizada enquanto que a probabilidade de detectar de maneira incorreta pontos em não bordas deve ser minimizada.
- *Localização*: as bordas detectadas devem ser tão próximas quanto possíveis das bordas reais.
- *Número de respostas*: uma borda real não deve resultar em mais de uma borda detectada.

O método Canny foi proposto em [Can86]. É um dos algoritmos de detecção de bordas mais utilizados. O algoritmo possui 5 etapas:

1. *Suavização*: o algoritmo começa convoluindo a imagem com um filtro gaussiano com o intuito de suavizar a imagem e diminuir o ruído nos resultados.
2. *Encontrar gradientes*: o detector de Canny, basicamente, encontra bordas onde a intensidade em níveis de cinza da imagem sofre uma grande variação. Estas áreas são determinadas como o gradiente da imagem. São utilizados quatro filtros para detectar bordas em 4 direções diferentes: horizontal, vertical e nas duas diagonais.
3. *Eliminar pontos não máximos*: para ter um resultado com bordas finas, os pontos que não forem máximos locais são eliminados dos resultados.
4. *Limiar duplo*: após a etapa anterior, são utilizados dois limiares. Um limiar mais alto é aplicado aos resultados e todos os pixels com intensidade maior que este limiar mais alto são considerados bordas fortes. Um limiar mais baixo é utilizado para eliminar todos os pixels com intensidade menor que este limiar mais baixo. Todos os demais pixels, com intensidade entre o limiar mais baixo e o limiar mais alto, são considerados bordas fracas.
5. *Eliminação de bordas fracas*: as bordas fortes são confiáveis e pertencem às bordas reais. São, então, preservadas. As bordas fracas que estão conectadas a bordas fortes também são preservadas. Bordas fracas não conectadas a bordas fortes têm grande probabilidade de serem ruídos e serão, então, eliminadas nesta etapa.

Neste trabalho, o detector de bordas Canny foi utilizado em cada um dos 3 canais do modelo de cores RGB. Os 3 mapas de bordas calculados são somados formando o mapa de bordas resultante.

Foi utilizada também uma abordagem morfológica para cômputo do mapa de bordas. Em uma etapa inicial, são aplicados os operados HMIN e HMAX que eliminam, respectivamente, mínimos e máximos locais com valores menores que o parâmetro H passado aos operadores. Os operadores HMIN e HMAX são filtros conexos. Um filtro conexo é um operador crescente que possui a propriedade de unir regiões planas adjacentes em uma imagem. A aplicação de um filtro conexo nunca aumenta o número de regiões planas em uma imagem. Todas as regiões planas que aparecem no resultado de um filtro conexo estavam na imagem original. Outra propriedade importante dos filtros conexos é a preservação das bordas da imagem original. Os operadores HMIN e HMAX são definidos por:

$$HMIN_{H,B}(X) = (X + H)\nabla_B X, \quad (3.2)$$

$$HMAX_{H,B}(X) = (X - H)\Delta_B X, \quad (3.3)$$

onde ∇ é a reconstrução sup-geodésica e Δ é a reconstrução inf-geodésica (Seção 2.10.1). O valor utilizado para o parâmetro H nos filtros conexos HMIN e HMAX foi 30. Esta etapa elimina ruídos

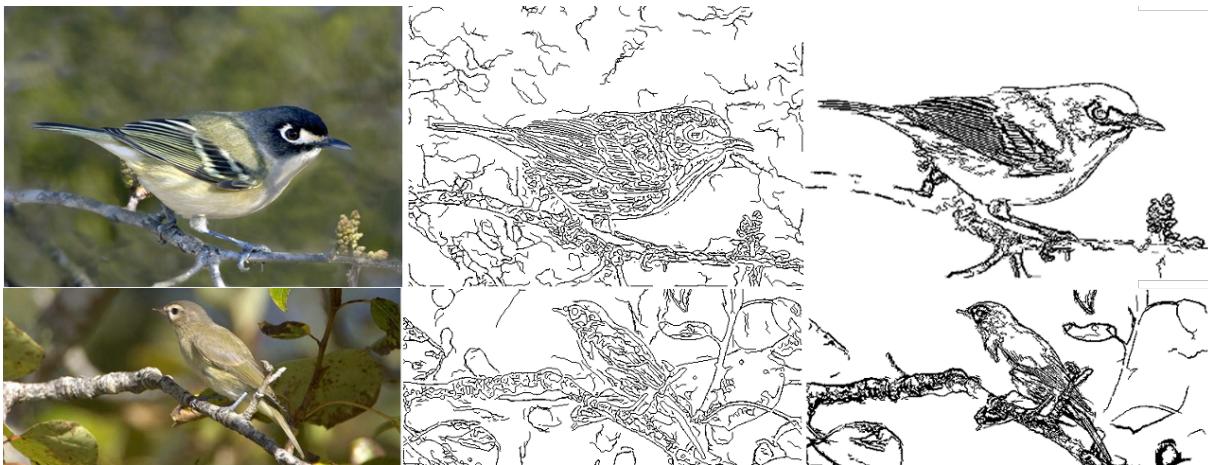


Figura 3.6: Aplicação do detector de bordas Canny e do gradiente morfológico externo filtrado por operadores conexos em duas imagens da base de dados Caltech-UCSD Birds-200 2011.

e diminui o número de regiões planas da imagem original. Pode-se ver que esta etapa é similar a etapa de suavização do detector de bordas Canny.

Em seguida, um EE em forma de disco com raio 1 é criado e o gradiente morfológico externo é calculado usando este EE:

$$\text{grad}_B^{\text{ext}}(X) = (X \oplus B) - X. \quad (3.4)$$

Como a imagem possui 3 canais de cores no modelo RGB, o gradiente também possui estes 3 canais. Então, o gradiente resultante é dividido em cada canal, formando 3 mapas de bordas distintos.

Cada um dos resultados são convertidos para binário utilizando um limiar. Usou-se limiar igual a 15 nos experimentos deste trabalho. O mapa de bordas resultante é a adição dos 3 mapas binários. Esta etapa elimina bordas com limiares abaixo do parâmetro passado, similar à etapa de eliminação de bordas fracas do detector de Canny.

A Figura 3.6 mostra o mapa de bordas resultante aplicando Canny e aplicando o gradiente morfológico externo filtrado por operadores conexos. Pode-se notar que o mapa de borda detectado por Canny apresenta bordas mais finas devido à etapa de eliminação de pontos não máximos locais. O mapa de bordas produzido pela abordagem morfológica apresenta bordas mais grossas. Pelas características apresentadas, o descriptor PPE funciona melhor utilizando as bordas produzidas por Canny e o descriptor GRABED, as bordas produzidas pela abordagem morfológica.

3.3 Pirâmide de Regiões Circulares Concêntricas

Uma desvantagem no uso das pirâmides espaciais é que elas não são invariantes a rotação. A invariância a rotação não é importante para o reconhecimento de vários objetos como montanhas, carros, casas, edifícios, etc. Estes objetos estão, geralmente, presentes na cena sem variações na rotação. Contudo, latas, garrafas, sapatos, canetas ou livros são exemplos de objetos que estão presentes em uma imagem em diferentes ângulos de rotação. Para um ser humano, é fácil identificar uma garrafa ou lata independente de sua inclinação na cena. Porém o reconhecimento de objetos não trata estas diferenças na pose do objeto sem um custo.

A Pirâmide de Regiões Circulares Concêntricas (PRCC) foi proposta neste trabalho, para substituir as pirâmides espaciais tradicionais (Seção 2.4) quando a invariância a rotação é uma propriedade desejável. A PRCC é construída com L níveis: sendo que os $L - 1$ primeiros níveis são regiões circulares de raios crescentes centradas no centro da imagem ou em um ponto P , opcional, passado como parâmetro e o último nível engloba a imagem inteira. O primeiro nível utilizado apresenta raio com tamanho igual a $R\%$ da altura ou largura da imagem, o que for menor, onde R é passado como argumento. O segundo nível é a região circular com raio igual a $2 \times R\%$ do menor eixo da imagem e

o nível l , com $l \in [1, L - 1]$ é uma região circular com raio igual a $l \times R\%$ do menor eixo da imagem. Para $l = L$, a última região é a imagem inteira. Cada nível engloba a área dos níveis anteriores. Para cada região, um conjunto de características locais são extraídas. Descritores locais são calculados a partir destas características e traduzidos em palavras visuais de um dicionário visual pré-calculado com N palavras visuais (Seção 2.3). Cada região é representada por um histograma normalizado de frequência das palavras visuais que representam os descritores. Seja $\omega_i \in (0, 1)$ um peso aplicado aos vetores do nível i com $i \in (1, L)$ e $\sum_{i=1}^L \omega_i = 1$.

O descritor final do PRCC é a concatenação dos histogramas de palavras visuais de cada região. Uma máscara circular é utilizada para delimitar as regiões circulares da pirâmide. A máscara circular é calculada utilizando a distância euclidiana:

$$mascara = \begin{cases} 1 & \text{se } \sqrt{(x - C_x)^2 + (y - C_y)^2} \leq R, \\ 0 & \text{se } \sqrt{(x - C_x)^2 + (y - C_y)^2} > R, \end{cases} \quad (3.5)$$

onde (C_x, C_y) é o ponto central da máscara circular, R é o raio do círculo, $x \in \{0, \dots, W\}$, $y \in \{0, \dots, H\}$, W é a largura da máscara e H é a altura da máscara.

A palavra visual utilizada para mapear a região de fundo, fora da área circular de interesse, é identificada. A frequência correspondente à palavra visual usada para mapear esta região de fundo é excluída do descritor final. Utilizando um dicionário visual de 600 palavras, parâmetros $R = 15$ e $L = 4$, o descritor final possui dimensão igual a 2396 ($= 4 \times (600 - 1)$).

Figura 3.7 mostra uma imagem da classe *BACKGROUND_Google* da base de dados Caltech-101 e a mesma classe rotacionada em 196° . A segunda linha mostra os gráficos dos descritores de ambas as imagens na abordagem do saco de palavras. A próxima linha mostra os gráficos de descritores de ambas as imagens usando saco de palavras em uma quantização espacial em 2 níveis. Cada descritor totaliza 3000 dimensões usando o mesmo dicionário visual de 600 palavras visuais. A última linha mostra gráficos de descritores usando a PCRR com 2396 dimensões para cada descritor.

Para calcular a diferença entre o descritor calculado usando a tradicional abordagem do saco de palavra acrescido da quantização espacial com regiões retangulares e a PRCC, foi aplicada a distância qui-quadrada entre os dois descritores finais (da imagem original e da imagem rotacionada). A distância qui-quadrada entre os vetores x e y com tamanho n é igual a $\sum_{i=1}^n \frac{(x(i) - y(i))^2}{(2*(x(i) + y(i)))}$. Para a abordagem do saco de palavras acrescido da quantização espacial, a diferença entre os dois descritores é 3,4055 e, para a PRCC, a diferença é 1,2146. Portanto, para uma mesma rotação, a PRCC produz uma diferença menor entre os descritores. A pirâmide proposta tem 2396 dimensões e a quantização espacial usada para comparação possui 3000 dimensões. Calculando-se a diferença média para cada dimensão, tem-se, para a abordagem da quantização espacial com regiões retangulares, $3,4055 \div 3000 = 1,1352 \times 10^{-3}$. Para a PRCC o valor médio da diferença entre a imagem original e a rotacionada para cada dimensão é $1,2146 \div 2396 = 5,0693 \times 10^{-4}$. A PRCC possui também uma diferença média menor, para cada dimensão.

Figura 3.8 mostra a mesma imagem da Figura 3.7 e cada região usada para calcular o descritor final.

O Algoritmo 5 mostra PRCC em pseudocódigo. O PRCC recebe a imagem *IMG*, a variável *Divisoes* com os raios que serão utilizados nas máscaras circulares, a variável *pesos* com os pesos que serão aplicados em cada nível, um ponto *P*, opcional, que será o centro das máscaras circulares concêntricas e um ponteiro *Ptr_Descriptor* para a função que calcula o descritor. Para o último nível, não existe máscara circular e os descritores são extraídos da imagem inteira. Para os demais níveis, uma máscara circular é calculada e aplicada à imagem original. Os descritores são calculados utilizando a abordagem saco de palavras. Os descritores são extraídos na região da máscara e o peso correspondente ao nível é aplicado. A palavra que mapeia a região de fora da máscara circular é identificada. A dimensão correspondente a esta palavra é retirada do vetor de características. O vetor final é a concatenação dos descritores de cada região.

A Seção 5.5 descreve os experimentos realizados utilizando a PRCC onde é feita uma comparação da PRCC com a Pirâmide Espacial e o modelo original do saco de palavras. Estes experimentos foram publicados em [LH13b].

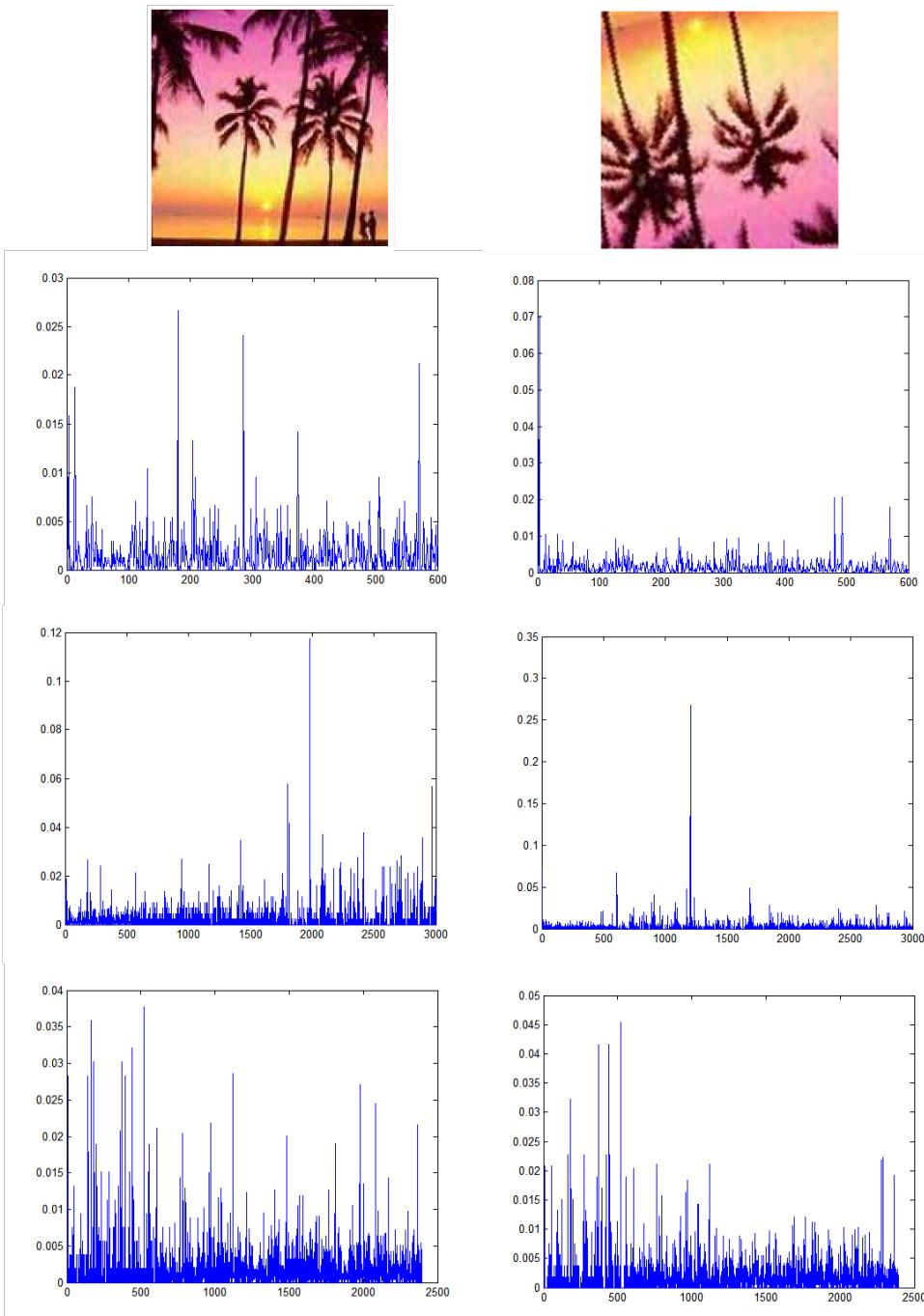


Figura 3.7: Imagem original da classe BACKGROUND_Google da base de dados Caltech-101 e a imagem rotacionada (196° de rotação). A segunda linha mostra o descritor de cada imagem na abordagem do saco de palavras com 600 palavras visuais. A terceira linha mostra o descritor de ambas as imagens na abordagem do saco de palavras junto com uma quantização espacial de 2 níveis que totaliza 3000 dimensões e a última linha mostra os descritores das imagens usando a PRCC com 2396 dimensões.



Figura 3.8: A figura mostra a mesma imagem da Figura 3.7. A imagem original da classe BACKGROUND_Google da base de dados Caltech-101 e a imagem rotacionada em 196°. São mostradas 4 regiões da PRCC para cada imagem. A primeira coluna mostra a primeira região circular com raio de 15% da altura da imagem, a próxima coluna mostra a segunda região e assim sucessivamente, até a última coluna que mostra a imagem inteira que é a quarta região utilizada para calcular o descritor final.

```

Input: IMG, Divisoes, Pesos, P, Ptr_Descriptor
Output: Vetor_Final
Vetor_Final ← ∅;
for l ← 1 to numero_niveis(Divisoes) + 1 do
    if l = numero_niveis(Divisoes) + 1 then
        | IMGR ← IMG;
    end
    else
        | Raio ← Divisoes(l);
        | if P = nulo then
        |     | P ← centro(IMG);
        | end
        | Mascara ← circulo(P, Raio);
        | IMGR ← IMG * Mascara;
    end
    Vetor ← Ptr_Descriptor(IMGR);
    Palavra ← palavra que mapeia a região de fora da máscara circular;
    Vetor ← Vetor – Vetor(dimensao(Palavra));
    Vetor_Final ← Vetor_Final + Peso(l) * Vetor;
end
return Vetor_Final

```

Algoritmo 5: Algoritmo PRCC

Capítulo 4

Categorização Refinada

Este capítulo descreve o uso das QEFs propostas na Seção 3.1, e dos descritores propostos neste trabalho na Seção 3.2, juntamente com outras técnicas para a resolução do problema da categorização refinada. A categorização refinada (Seção 2.5) consiste na categorização de objetos pertencentes a classes muito parecidas. Neste trabalho, foi utilizada a base de pássaros Caltech-UCSD Birds-200 2011 (Seção 5.6.1). O método proposto aqui cria um modelo específico para cada classe testada utilizando a abordagem do saco de palavras junto com as QEFs, os descritores baseados em abertura morfológicas e outros descritores usualmente utilizados em aplicações de reconhecimento de objetos. A configuração de todos os parâmetros envolvidos é específica por classe e calculando a média de resultados de todas as classes foram obtidos resultados similares aos melhores resultados reportados na literatura [YBFF12]. A Seção 5.6 descreve estes resultados. A ideia de criar um modelo com abordagens e configurações específicas para cada classe a ser testada na base de dados é intuitiva. Quando as pessoas tentam reconhecer objetos bem parecidos, são os detalhes que vão identificar unicamente cada objeto. Por exemplo, um gato e uma onça da mesma cor, o tamanho seria a diferença entre um e outro. A diferença entre um cavalo e uma zebra, a padronagem de listas da zebra diferenciaria uma espécie da outra. O novo método de categorização refinada foi publicado em [LH13a].

4.1 Etapas

O método de categorização refinada proposto neste trabalho é dividido em cinco etapas: inicialização, geração de descritores, criação dos modelos, testes e o cálculo das medidas de desempenho.

4.1.1 Inicialização

Esta é a etapa onde são carregadas as parametrizações utilizadas e também informações da base de dados, como a localização das imagens, rótulo das imagens, divisão treinamento/teste da base de dados, anotações das partes, caixa envoltória dos objetos, entre outras. É gerada uma semente diferente a cada execução da função de números pseudo-aleatórios do Matlab a partir da hora do sistema. Assim, a cada execução, números diferentes são gerados aumentando o grau de aleatoriedade das rotinas que dependem disto, como, por exemplo, a seleção dos descritores que serão utilizados na construção de um dicionário visual.

Uma rotina opcional durante a inicialização é a geração de imagens de treinamento expandidas. Utilizando uma abordagem similar a [LF06], geram-se novas imagens a partir de transformações nas imagens de treinamento originais, aumentando o número de imagens disponíveis na criação de um modelo para classe. Para cada imagem de treinamento, F fatores de redimensionamento são aplicados gerando imagens em tamanhos diferentes. Em cada tamanho, ou se mantém a imagem ou se aplica uma função de espelhamento e, finalmente, R ângulos de rotações são aplicados em cada um dos resultados da etapa anterior. Para cada imagem de treinamento original, são gerados $F \times 2 \times R$ imagens de treinamento expandidas.

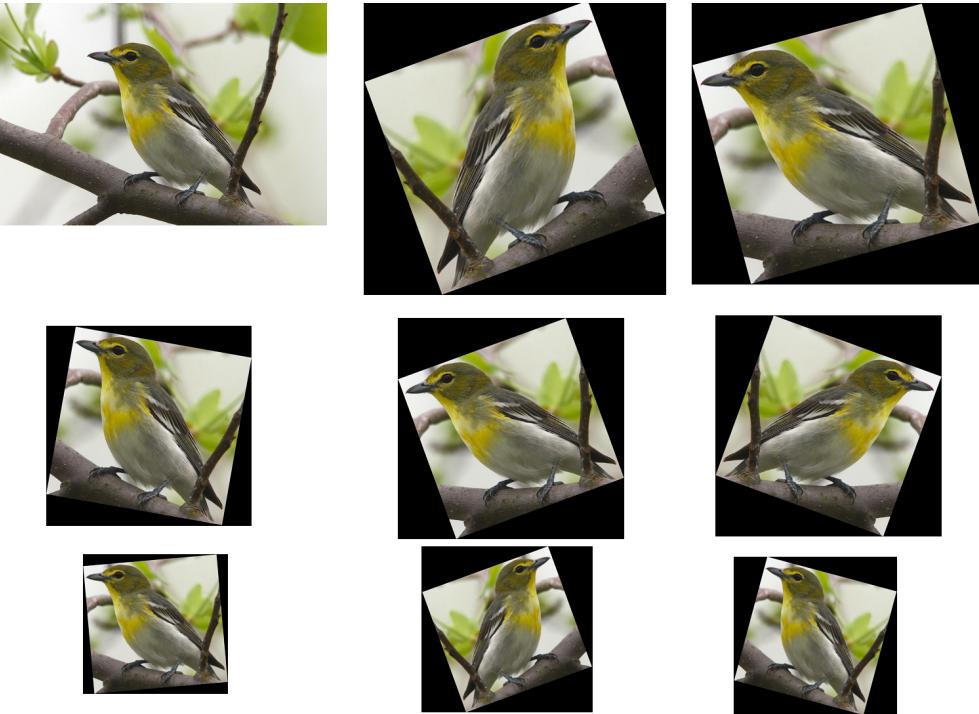


Figura 4.1: A imagem mostra, no canto superior esquerdo, a foto original da base de dados de um pássaro da classe 157 e as fotos seguintes mostram 8 exemplos de imagens geradas pela rotina de geração de imagens de treinamento expandidas.

Nos testes da Seção 5.6, foram utilizados três fatores de redimensionamento, espelhamento ou não da imagem e 9 ângulos de rotações entre -20° e 20° . Das 389 imagens de treinamento original, são geradas 21006 imagens ($389 \text{ imagens} \times 3 \text{ escalas} \times 2 \text{ espelhamentos} \times 9 \text{ rotações}$). As transformações foram aplicadas em uma imagem resultante de um recorte feito utilizando a caixa envoltória dos pássaros. A Figura 4.1 mostra uma imagem original da classe 157 e o exemplo de oito imagens geradas pela rotina de geração de imagens expandidas.

4.1.2 Geração dos descritores

Nesta etapa, são construídos os dicionários visuais (Seção 2.3) utilizados e testados na aplicação de categorização refinada. Dois tipos de dicionários foram criados: um local e outro global.

- **Dicionário global:** é construído a partir de descritores amostrados aleatoriamente, com distribuição uniforme, entre todas as classes da base de dados. Este dicionário pode ser utilizado em uma classificação binária ou em uma classificação multiclasse.
- **Dicionário local:** é específico para uma classe. Metade dos descritores aleatoriamente escolhidos para serem utilizados na construção do dicionário visual é de imagens rotuladas como a classe do dicionário e a outra metade dos descritores é escolhida uniformemente entre todas as outras classes utilizadas na aplicação. É necessário construir um dicionário local para cada classe e este dicionário só poderá ser utilizado em uma classificação binária. A vantagem é que, por possuir metade dos descritores vindos de uma única classe, ele tem maior probabilidade de descrever melhor esta classe. A desvantagem dos dicionários locais em relação aos globais é o custo computacional exigido para construí-los.

Por causa do custo computacional de construir dicionários locais, optou-se por utilizar a abordagem global na construção dos dicionários visuais nos experimentos descritos na Seção 5.6.

A função de construção do dicionário recebe como entrada o número de palavras do dicionário visual, o número de descritores que será utilizado nesta construção, qual o descritor utilizado e uma

variável que armazena informações relativas à base de dados. A partir da variável que armazena informações da base de dados, cria-se uma lista das imagens de treinamento. A base de dados possui um marcador que indica se a imagem deve ser utilizada durante o treinamento ou durante o teste. Para imagens de mesmo rótulo, a ordem destas imagens é aleatorizada. O descritor é chamado, passando a imagem como parâmetro e os descritores extraídos são armazenados em uma matriz. Este processo continua até que o número de descritores ultrapasse o limite calculado pela razão número de descritores pelo número de classes. Ao ultrapassar este limite, a matriz passa a ser preenchida por descritores da próxima classe até atingir o limite de descritores da última classe. Novamente, a ordem dos descritores é embaralhada e os descritores são agrupados em N centros diferentes utilizando o algoritmo k-médias [HE02], onde N é o número de palavras do dicionário visual. A lista destes N centros forma o dicionário visual. A função de construção do dicionário visual é independente do descritor utilizado que é passado como parâmetro.

O Algoritmo 6 mostra a função de construção de um dicionário visual em pseudo-código.

```

Input: bd, num_palavras, total_descr, Ptr_Descriptor
Output: Dicionario
lista_imgs ← retorna_imgs_treinamento(bd);
num_descr_por_classe ← total_descr/num_classes(bd);
descritores ← ∅;
for classe ← 1 to num_classes(bd) do
    descritores_classe ← ∅;
    ids_arqs_classe = retorna_arqs_classe(lista_imgs, classe);
    for id_arq ← 1 to tamanho(ids_arqs_classe) do
        | imagem ← ler(lista_imgs(id_arq));
        | descritores_imagem ← Ptr_Descriptor(imagem);
        | descritores_classe ← descritores_classe + descritores_imagem;
    end
    if tamanho(descritores_classe) > num_descr_por_classe then
        | descritores_classe ← subconjunto(descritores_classe, num_descr_por_classe);
    end
    descritores ← descritores + descritores_classe;
end
descritores ← aleatoriza_colunas(descritores);
dicionario ← kmedias(descritores, num_palavras);
return dicionario

```

Algoritmo 6: Algoritmo Constrói Dicionário

Construídos os dicionários visuais necessários, começa a geração das matrizes de descritores. É gerada uma matriz de descritores para cada descritor visual utilizado. Assim, existirá uma matriz de descritores para o PHOW, por exemplo, onde o número de colunas possui o mesmo número de dimensões do PHOW e o número de linhas é igual ao número de imagens de treinamento. Uma lista das imagens de treinamento, a lista dos dicionários visuais disponíveis e um ponteiro para a função que chama os descritores visuais a serem utilizados para a geração das matrizes são passados como parâmetros para a geração das matrizes de descritores. A cada imagem, a função que chama os descritores é chamada junto com a lista de dicionários. Nesta função, ocorre a parametrização da chamada do descritor e a chamada da quantização espacial apropriada: SPM [LSP06], QEF (Seção 3.1) ou PRCC (Seção 3.3). Nos experimentos deste trabalho, os descritores utilizados na etapa de geração são: PHOW nos 3 canais RGB (Seção 2.6.2), PHOW nos níveis de cinza (Seção 2.6.2), PHOG (Seção 2.6.4), SURF (Seção 2.6.3), LBP (Seção 2.6.5), Histograma RGB (Seção 2.6.6), GRABED (Seção 3.2), PPE (Seção 3.2.1) e PPD (Seção 3.2.2).

4.1.3 Criação de Modelos

Esta etapa utiliza as matrizes de descritores de treinamento gerados na etapa anterior para criar um modelo. Para cada descritor visual utilizado na geração das matrizes, são criados vários modelos diferentes. Para cada matriz de descritores de um descritor visual, tem-se:

1. Modelo de nível 1: os descritores das imagens de treinamento são agrupados de alguma maneira a respeitar alguma hierarquia que exista entre as classes da base de dados. Por exemplo, na base de dados de pássaros, utilizada nos experimentos deste trabalho, o subconjunto de pássaros utilizados nos experimentos podem claramente ser divididos em 2 grupos diferentes. Os descritores das espécies do gênero *Vireo* e os descritores das espécies de pica-paus. Ou seja, descritores de 7 espécies são rotulados como pertencentes à classe 1 e descritores de 6 espécies são rotulados como pertencentes à classe 2. Chamamos esta classificação de classificação de 1º nível. Assim, para os experimentos deste trabalho, tem-se um modelo de nível 1 com 2 rótulos diferentes.
2. Modelo binário de nível 2: a mesma matriz de descritores é utilizada para criar um modelo para cada classe. Fixando a cada momento uma classe diferente, os descritores pertencentes a esta classe são rotulados como descritores do rótulo 1 e os descritores das outras classes diferentes são rotulados como descritores do rótulo 2. No final deste processo serão criados um número de modelos igual ao número de classes envolvida na categorização. Para o subconjunto de classes utilizadas nos experimentos deste trabalho, são criados 13 modelos nesta etapa. Ao todo, nos experimentos deste trabalho, são criados 14 modelos para cada tipo de descritor utilizado (SIFT, SURF, GRABED, etc).

Para cada modelo, é utilizada a técnica dos mapas explícitos de características proposta em [VZ12] e implementada em [VF10]. Esta técnica recebe as matrizes de descritores e cria um mapa que aproxima a função núcleo qui-quadrada com desempenho praticamente similar ao da aplicação da função núcleo completa, mas com custo computacional até 19 vezes menor. A função núcleo qui-quadrada é aplicada a um classificador SVM linear (Seção 2.7) e o algoritmo Pégasos [SSSS07] é utilizado para resolver o problema de otimização do classificador SVM.

Após a criação do modelo pelo classificador SVM, calcula-se a sua confiança utilizando o método *k-fold* de validação cruzada (Seção 2.7.1). A medida utilizada para validação do modelo é a precisão. Nos experimentos deste trabalho, foi utilizado o parâmetro $k = 5$.

4.1.4 Testes

Durante a fase de testes, para cada amostra, cada um dos modelos é aplicado. Cada modelo retorna a lista de probabilidades da amostra pertencer a cada uma das classes que foram utilizadas em sua construção. Assim, a resposta para cada tipo de modelo é:

1. Modelo de nível 1: resposta de qual a probabilidade pertencer a cada uma das categorias agrupadas. No caso dos experimentos efetuados neste trabalho, este modelo possui dois rótulos; classe 1 (espécies do gênero *Vireo*) ou classe 2 (espécies de pica-paus).
2. Modelo binário de nível 2: resposta com a probabilidade da amostra pertencer à classe 1 que é a classe do modelo ou a classe 2, ou seja, pertencer às outras classes diferentes da classe do modelo. Nos testes efetuados, são 13 modelos binários construídos na fase de treinamento.

Quando o rótulo respondido pelo modelo corresponde a mais de uma classe final, é feita uma divisão uniforme entre estas probabilidades e o número real de classes. Por exemplo, suponha que um modelo de nível 1 retorne que a amostra tenha 40% de pertencer à classe 1 e 60% de pertencer à classe 2. O rótulo 1 do modelo de nível 1 corresponde, nos experimentos efetuados, a 7 classes do gênero *Vireo* e o rótulo 2 corresponde a 6 classes de pica-paus, então a probabilidade de cada

classe do gênero *Vireo* será igual a $\frac{40}{7}\%$ e para cada classe de pica-pau será $\frac{60}{6}\%$, de acordo com este modelo.

A soma das probabilidades de resposta dos modelos para uma amostra é sempre 1. Modelos de nível 1 com precisão calculada durante a criação do modelo pela técnica de validação cruzada inferior a um limiar L_1 e os outros modelos de nível 2 com precisão inferior a um limiar L_2 são eliminados. Os resultados dos modelos restantes são combinados utilizando a maior probabilidade média para obter o resultado final (Seção 2.7.2). Nos experimentos deste trabalho, foram utilizados os limiares $L_1 = 0,75$ e $L_2 = 0,30$.

4.1.5 Medidas de Desempenho

Para medir o resultado da aplicação, utilizou-se a medida precisão média (AP) para cada classe (Seção 2.9) e a implementação é a função utilizada no desafio VOC Pascal [EGW⁺10]. Calculando a média das AP de cada classe, obtém-se a MAP (média das precisões médias).

Capítulo 5

Experimentos e Resultados

Este capítulo descreve alguns dos experimentos realizados e dos resultados atingidos. A Seção 5.1 descreve os ambientes de hardware e software utilizados nos experimentos. A Seção 5.2 descreve a comparação entre a Quantização Espacial Flexível e a pirâmide espacial tradicional proposta no artigo original de [LSP06]. A Seção 5.3 descreve o descritor GRABED e o compara com outros descritores. A Seção 5.4 descreve a combinação de descritores para resolver a categorização de instâncias de monumentos. A Seção 5.5 apresenta experimentos que demonstram a maior robustez a rotação de objetos da Pirâmide com Regiões Circulares Concêntricas em relação à Pirâmide Espacial com divisão retangular tradicional. A Seção 5.6 descreve os experimentos realizados utilizando as técnicas descritas neste trabalho na resolução de um problema de categorização refinada. Os experimentos das seções 5.3 e 5.6 foram publicados em [LH13a]. Em [LH11b], foi publicado o experimento da Seção 5.4. O experimento da Seção 5.5 foi publicado em [LH13b].

5.1 Ambiente dos Experimentos

As implementações foram feitas utilizando o Matlab¹. Foi utilizada a versão 7.8 (Release 2009a) de 64 bits rodando em um computador de mesa com processador Intel Core i5 2.8 GHz, arquitetura de 64 bits e 16 GB de memória RAM.

Foram utilizadas as seguintes bibliotecas:

- *VLFeat*: biblioteca de Visão Computacional com licença GPL². Possui interfaces em C e Matlab e possui vários algoritmos modernos de VC e reconhecimento de padrões implementados [VF10].
- *OpenSURF*: biblioteca que implementa o descritor SURF [Eva09].
- *PRTtools*: é uma *toolbox* para o ambiente Matlab focada em reconhecimento de padrões e sem custo para pesquisa [DJP+07]. Ela implementa cerca de 200 funções diferentes na área de reconhecimento de padrões e processamento de imagens.
- *slmetric_pw*: conjunto de funções escritas em Matlab que calcula a distância entre dois vetores utilizando 20 medidas diferentes, como distância euclidiana, distância qui-quadrada, distância L1, distância de Minkowski, distância de Hamming, entre outras. Faz parte da *toolbox* para o Matlab *Statistical Learning Toolbox*³.

5.2 Experimentos com a Quantização Espacial Flexível

As Quantizações Espaciais Flexíveis (QEF) descritas na Seção 3.1 propõem algumas flexibilizações, introduzem alguns parâmetros e eliminam algumas restrições que existem nas Pirâmides

¹www.mathworks.com

²www.gnu.org/copyleft/gpl.html

³www.mathworks.com/matlabcentral/fileexchange/12333-statistical-learning-toolbox

#	Nome científico	# treinamento	# teste
2	Diomedea immutabilis	30	30
12	Xanthocephalus xanthocephalus	30	26
14	Passerina cyanea	30	30
17	Cardinalis cardinalis	30	27
160	Dendroica coerulescens	30	29

Tabela 5.1: Espécies de pássaros utilizadas no experimento de comparação da QEF e SPM.



Figura 5.1: A imagem mostra as 5 espécies de pássaros utilizadas neste experimento. Começando no canto superior esquerdo, em sentido horário, a ordem é a mesma da Tabela 5.1.

Espaciais tradicionais (SPM) propostas no artigo de [LSP06]. Para validar estas modificações, foi feita uma comparação da QEF com a SPM em uma base pública de imagens para aplicações de categorização de objetos. Os experimentos desta seção mostram que a QEF obteve uma acurácia maior que a SPM nos testes efetuados.

A base de dados utilizada foi a Caltech-UCSD Birds-200 2011 [WBW⁺11]. Uma descrição detalhada desta base de dados está na Seção 5.6.1. Foram utilizada 5 espécies distintas de pássaros desta base de dados. Cada espécie é uma categoria diferente na base de dados. A Tabela 5.1 apresenta cada uma das 5 espécies de pássaros da base de dados utilizadas no experimento. São mostrados o número da classe na base de dados, o nome científico da espécie, o número de imagens para treinamento e o número de imagens para teste segundo o indicador sugerido pela própria base de dados. A Figura 5.1 mostra uma imagem para cada uma das 5 espécies utilizadas no experimento. Iniciando no canto superior esquerdo, em sentido horário, a ordem é a mesma da Tabela 5.1.

Para a implementação da SPM, foi utilizado o código disponível no sítio da autora do artigo original em <http://www.cs.illinois.edu/homes/slazebni/>. Foi construído um dicionário visual de 600 palavras visuais. O descriptor utilizado é o SIFT (Seção 2.6.1), calculado em uma grade regular de 8 pixels de espaçamento. A pirâmide espacial possui 3 níveis com 1, 4 e 16 regiões respectivamente. O peso de cada nível é $\frac{1}{8}$, $\frac{1}{4}$ e $\frac{1}{2}$, respectivamente. A QEF foi implementada de acordo com o ambiente descrito na Seção 5.1. Foi criado um dicionário visual com 600 palavras. O descriptor utilizado foi o PHOW (Seção 2.6.2), calculado em uma grade regular com 3 pixels de espaçamento. Foi utilizada a seguinte divisão para a quantização espacial: 2 níveis, sendo que o nível 1 apresenta 1 região e o nível 2 apresenta 9 regiões (formato 3×3 regiões). O peso aplicado a cada nível é $\frac{1}{2}$. Foi utilizado um parâmetro de sobreposição de regiões igual a 15 pixels. Foi criado um modelo para cada classe utilizando as imagens de treinamento desta classe como conjunto positivo e as

Classe	SPM		QEF	
	acc.(%)	AP	acc.(%)	AP
2	84,5	0,764	80,3	0,718
12	85,9	0,738	88,7	0,853
14	79,6	0,413	78,7	0,558
17	81,7	0,436	90,1	0,719
160	78,9	0,637	83,1	0,654
Média	82,1	0,598	84,2	0,700

Tabela 5.2: Resultados da comparação entre a SPM e a QEF em 5 classes da base de dados Caltech-UCSD Birds-200 2011.

imagens de treinamento das outras 4 classes como conjunto negativo. Todas as imagens de teste das 5 classes foram utilizadas para testar o modelo. Foi utilizado uma classificador binário SVM com função núcleo qui-quadrada para a QEF e função núcleo intersecção de histograma para a SPM. A Tabela 5.2 apresenta os resultados da comparação entre o SPM e a QEF para cada uma das 5 classes e a média das 5 classes. Foram utilizadas as medidas acurácia e a precisão média - AP (Seção 2.9). A QEF apresentou uma média das precisões médias (MAP) para as 5 classes 17% melhor que a pirâmide tradicional.

Um outro experimento foi efetuado comparando a SPM com a QEF. O código desenvolvido neste trabalho foi utilizado para categorizar as mesmas 5 classes do experimento anterior. Foi utilizada uma abordagem multiclasse e a medida utilizada para avaliar o desempenho foi a acurácia. O descriptor utilizado foi o PHOW em uma grade regular com 3 pixels de espaçamento. Em cada ponto da grade foi calculado o SIFT utilizando 2, 4, 6 e 10 pixels de largura. Um dicionário visual com 600 palavras visuais foi calculado utilizando os descritores locais PHOW. Foi utilizado o classificador SVM com núcleo qui-quadrado. A única diferença entre as duas abordagens foi a utilização de diferentes divisões espaciais. O código para a SPM utilizava uma divisão espacial de 3 níveis. Uma região no primeiro nível, 4 regiões no segundo e 16 regiões no terceiro com pesos de $\frac{1}{8}$, $\frac{1}{4}$ e $\frac{1}{2}$. Não existe sobreposição de regiões em um mesmo nível. A QEF foi configurada para utilizar dois níveis: uma região no primeiro nível e 9 regiões no segundo nível. Foram utilizados 15 pixels de sobreposição entre regiões de um mesmo nível. Foram utilizadas 15 imagens de cada classe para treinamento e as demais imagens de cada classe foram utilizadas no teste. A divisão entre as imagens de treinamento e teste são definidas pela própria base de dados. Foram efetuadas 5 categorizações e a acurácia média obtida foi 53,8% para a SPM e 57,0% para o QEF. O QEF obteve um desempenho aproximadamente 6% melhor que a SPM nos testes efetuados.

5.2.1 Discussão

A QEF foi utilizada em vários experimentos ao longo deste trabalho. Uma de suas características marcantes é a flexibilidade na utilização. Vários parâmetros podem ser escolhidos e as possibilidades são inúmeras. Este também pode ser visto como um ponto fraco da QEF. Até agora, não se conseguiu definir uma regra clara para determinar quantos níveis utilizar em quais situações ou qual a divisão de regiões dentro de um nível. Usar ou não sobreposição e quantos pixels de sobreposição utilizar. Basicamente, ao longo dos experimentos neste trabalho, foram utilizadas diversas configurações e algumas se saíram melhor que outras. Estas configurações que obtiveram melhor desempenho foram escolhidas para a criação dos modelos das categorias.

Como observado nos experimentos nesta seção, verificou-se que ao utilizar uma configuração padrão, os resultados obtidos utilizando a QEF superaram os resultados sem utilizar qualquer quantização espacial ou, ainda, superaram os resultados utilizando uma SPM. Uma configuração padrão para a QEF é com dois níveis, sendo uma região no nível 1 e 4 ou 9 regiões no nível 2 e sobreposição de regiões com aproximadamente 15 pixels de tamanho e pesos iguais nos dois níveis. A QEF foi proposta para ser utilizada em um modelo discriminante da abordagem do saco de palavras.

Classe	# de imagens
BACKGROUND_Google	467
Faces	435
Faces_easy	435
Leopards	200
Motorbikes	798

Tabela 5.3: Classes da base de dados Caltech-101 utilizadas neste experimento e o respectivo número de imagens por classe



Figura 5.2: Duas imagens para cada classe da base de dados Caltech-101 utilizadas neste experimento. Da esquerda para a direita, as classes são: BACKGROUND_Google, Faces, Faces_Easy, Leopards e Motorbikes.

5.3 Experimentos com o Descriptor GRABED

O descriptor GRABED (Seção 3.2) foi testado em um subconjunto de 5 classes da base Caltech-101. Esse subconjunto apresenta classes com centenas de imagens, ao contrário das outras 96 classes da base de dados que apresenta um número de imagens da ordem das dezenas. As imagens da base de dados Caltech-101 são fáceis e apresentam pouca variabilidade intra-classe. Alguns trabalhos reportam resultados neste subconjunto da base Caltech-101, como [SRE+05]. A Tabela 5.3 apresenta as 5 classes do subconjunto da base de dados utilizada e o número de amostras por classe. A Figura 5.2 mostra dois exemplos por classe usada neste experimento. A classe BACKGROUND_Google é composta por imagens aleatórias. Não existe nenhum tipo de padrão nas imagens desta classe. A classe Faces e a classe Faces_easy são compostas por imagens contendo rostos humanos. Porém, na classe Faces_easy, os rostos ocupam uma porção maior da imagem. A classe Leopards mostra diferentes imagens de leopardo em diferentes situações na natureza e a classe Motorbikes mostra a visão lateral de motocicletas. Os experimentos desta seção mostram que o descriptor GRABED possui um desempenho similar aos descritores mais utilizados em categorização de objetos nos testes efetuados.

Todas as imagens testadas foram redimensionadas para 500 pixels de altura mantendo a mesma razão de aspecto, caso elas possuíssem mais de 500 pixels de altura, caso contrário, o tamanho original foi mantido. Para cada uma das 5 classes, um conjunto de 200 imagens foram aleatoriamente escolhidos para fazer parte do conjunto de treinamento. Cem imagens formam o conjunto positivo e 100 imagens são escolhidas dentre as outras 4 classes formando o conjunto negativo. Um modelo da classe é construído usando o classificador SVM com núcleo qui-quadrado (Seção 2.7).

Foram utilizados neste experimento os descritores: SIFT (Seção 2.6.1), PHOW (Seção 2.6.2), PHOG (Seção 2.6.4), SURF (Seção 2.6.3) e o GRABED (Seção 3.2).

Foi utilizada a implementação do SIFT da biblioteca VLFeat. Os descritores SIFT foram calculados nos pontos de interesse detectados pelo detector SIFT. Foi utilizada a abordagem do saco

Parâmetros		Medidas		
R	L ₀	Acc.(%)	Prec.(%)	Rec.(%)
12	2	84	82	91
16	2	84	82	93
12	3	84	82	91
16	3	83	80	94
12	5	85	82	92
16	5	84	82	91

Tabela 5.4: Médias das medidas para as 5 classes utilizando o descritor GRABED básico (sem o uso de múltiplas escalas e sem o uso de quantização espacial)

de palavras (Seção 2.3) e foram utilizadas 100 palavras no dicionário visual do descritor. O PHOW utilizado foi a implementação da VLFeat. Foi utilizada a abordagem do saco de palavras e foi utilizado um dicionário visual com 600 palavras como relatado em [LH11b]. A implementação do SURF utilizada foi a disponível na biblioteca OpenSURF library [Eva09]. Também foi utilizada a abordagem do saco de palavras com um dicionário visual com 100 palavras visuais. Foi utilizada a implementação do PHOG disponível no site dos autores do descritor⁴.

5.3.1 Resultados

Esta seção apresenta os resultados obtidos nos experimentos com descritor GRABED. A Tabela 5.4 apresenta os resultados do descritor GRABED básico (a média das 5 classes) sem o uso de diferentes escalas e sem o uso de quantizações espaciais. Foram testados dois valores para o parâmetro R (12 e 16) e 3 valores para o parâmetro L_0 (2, 3 e 5 pixels). Para $R = 12$, são usados 12 divisões entre 0° e 180° : 0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150 e 165 graus. No caso que $R = 16$, são 16 divisões entre 0° e 180° : começando em 0° , a próxima orientação é $11,25^\circ$ e, assim sucessivamente até a última orientação, $168,75^\circ$. O parâmetro *Max_Axis* usado é 500 pixels. Assim, com o parâmetro $L_0 = 2$, os tamanhos utilizados para criar os EEs são: 2, 4, 8, 16, 32, 64, 128 e 256. A regra estabelece que o tamanho do próximo EE linear é duas vezes o anterior até ultrapassar a metade do parâmetro *Max_Axis*. É criado um EE linear para cada combinação de tamanho e orientação.

A Tabela 5.4 não apresentou nenhuma diferença significativa entre $R = 12$ ou $R = 16$ e também nenhuma diferença significativa quando $L_0 = 2, 3$ ou 5 pixels. Todas as combinações possuem acurácia em torno de 84% (valor médio das 5 classes testadas). O melhor valor obtido é 85% de acurácia usando $R = 12$ e $L_0 = 5$ pixels.

A Tabela 5.5 apresenta a média das medidas para as 5 classes utilizando múltiplas escalas para o descritor GRABED. Cada escala divide a imagem anterior por um fator 2. Usando 3 escalas, a acurácia atinge o valor de 90%. Uma melhora de desempenho de cerca de 6% comparada ao descritor básico (Tabela 5.4).

Tabela 5.6 apresenta os resultados da média de medidas para as 5 classes. Neste experimento foram utilizados os mesmos parâmetros R e L_0 . Foi usado $R = 12$ e $L_0 = 2$ para todas os testes. Foram utilizadas quantizações espaciais de 2 níveis e múltiplas escalas e foi observada uma melhora significativa dos resultados.

Os melhores resultados foram obtidos por uma quantização espacial com 3×3 regiões no segundo nível. Foram obtidos 97% de acurácia. Esta medida é 14% melhor que o descritor básico e 8% melhor que o descritor utilizando múltiplas escalas. Estes resultados corroboram a literatura que reporta uma melhora na taxa de acertos em aplicações de reconhecimento de objetos com o uso das quantizações espaciais [WYY⁺¹⁰].

A Tabela 5.7 apresenta o resultados do descritor GRABED para cada uma das 5 classes utilizando a melhor parametrização do experimento apresentado na Tabela 5.6. Foram utilizadas duas

⁴<http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html>

Parâmetros			Medidas		
Escalas	R	L ₀	Acc. (%)	Prec. (%)	Rec. (%)
1	12	2	84	82	91
1	16	2	84	82	93
1	12	3	84	82	91
1	16	3	83	80	94
1	12	5	85	82	92
1	16	5	84	82	91
2	12	2	88	86	94
2	16	2	88	85	98
2	12	3	85	82	96
2	16	3	87	84	97
2	12	5	87	84	95
2	16	5	89	86	96
3	12	2	89	87	94
3	16	2	89	89	92
3	12	3	85	86	90
3	16	3	88	87	94
3	12	5	90	88	94
3	16	5	87	84	96

Tabela 5.5: Média das medidas das 5 classes para o descritor GRABED utilizando múltiplas escalas

Parâmetros		Medidas		
QEF	Escalas	Acc. (%)	Prec. (%)	Rec. (%)
1x1;2x2	1	93	90	96
1x1;2x2	2	94	92	97
1x1;2x2	3	94	92	97
1x1;3x3	1	93	96	91
1x1;3x3	2	97	97	97
1x1;3x3	3	95	92	98
1x1;4x4	1	93	92	96
1x1;4x4	2	95	95	96
1x1;4x4	3	96	97	95

Tabela 5.6: Média das medidas para as 5 classes utilizando múltiplas escalas e as QEFs propostas neste trabalho.

escalas diferentes e a QEF com 2 níveis (3×3 regiões no segundo nível). Os outros parâmetros são: $R = 12$, $L_0 = 2$ e $Max_Axis = 500$ pixels.

A Tabela 5.8 apresenta os melhores resultados para cada classe usando a sua melhor parametrização. Cada classe apresenta uma parametrização ideal. A média das medidas das 5 classes teve um pequeno incremento neste experimento quando comparada com os resultados da Tabela 5.7 que mostra a mesma parametrização para todas as classes.

A Tabela 5.9 apresenta resultados comparativos entre o descritor GRABED e os outros descritores testados.

5.3.2 Discussão

O descritor GRABED, proposto neste trabalho, se mostrou tão eficiente quanto os melhores descritores testados neste experimento para aplicações de reconhecimento de objetos. Ele possui similaridades ao PHOG, ambos medem características ligadas às bordas das imagens. Porém o GRABED mede além da distribuição de orientação, a distribuição das extensões das linhas dos

Classe	Medidas		
	Acc.(%)	Prec.(%)	Rec.(%)
BACKGROUND_Google	91	90	92
Faces	98	96	100
Faces_easy	99	98	100
Leopards	99	98	100
Motorbikes	97	100	94
Média	97	97	97

Tabela 5.7: Resultados do descriptor GRABED para cada uma das classes testadas utilizando a parametrização do melhor resultado médio apresentado na Tabela 5.6.

Classe	Parâmetros				Medidas		
	QEF	Escalas	R	L ₀	Acc.(%)	Prec.(%)	Rec.(%)
BACKGROUND_Google	1x1;3x3	3	12	2	91	87	96
Faces	1x1;4x4	3	12	2	100	100	100
Faces_easy	1x1;2x2	2	12	2	100	100	100
Leopards	1x1;3x3	2	12	2	99	98	100
Motorbikes	1x1;2x2	3	12	2	98	100	96
Média					98	97	98

Tabela 5.8: Resultados do descriptor GRABED com a melhor parametrização para cada uma das 5 classes.

gradientes em uma certa direção. Por estas características, o GRABED pode ser visto como uma alternativa para aplicações de reconhecimento de objetos e ser utilizado em uma abordagem de combinação de descritores [Pon11]. O PHOW, descritor que obteve o melhor desempenho nos testes efetuados, mede características mais ligadas à textura dos objetos, assim, tanto o PHOG quanto o GRABED podem ser uma excelente opção para ser combinado ao PHOW e tornar o modelo do objeto mais eficiente e robusto. Os descritores SIFT e SURF foram testados usando a abordagem do saco de palavras, porém sem utilizar quantizações espaciais. O dicionário utilizado também possui significativamente menos palavras (100 contra 600 do dicionário utilizado no descritor PHOW). O tamanho reduzido do dicionário visual e a não utilização da quantização espacial contribuíram para um desempenho inferior de ambos os descritores. Os resultados obtidos nestes experimentos corroboram com a literatura que afirma que o uso das quantizações espaciais aumenta a acurácia dos descritores nos problemas de reconhecimento de objetos. Por outro lado, o seu uso implica perda de robustez quanto a variações de escala e orientação. Estes pontos devem ser pesados antes da adoção ou não das quantizações espaciais.

Descriptor	QEF	Multiescalas	BoW	Medidas		
				Acc.(%)	Prec.(%)	Rec.(%)
PHOG	Sim	Sim	Não	98	97	99
PHOW	Sim	Sim	Sim	98	97	98
GRABED	Sim	Sim	Não	97	97	97
SURF	Não	Não	Sim	88	86	93
SIFT	Não	Não	Sim	86	84	90

Tabela 5.9: Resultados comparativos entre o descritor GRABED e os outros descritores testados: PHOG, PHOW, SURF and SIFT. A tabela apresenta a média das 5 classes testadas.

#	Monumento	Cidade	# de amostras
1	Cristo Redentor	Rio de Janeiro, Brasil	21
2	Estátua da Liberdade	Nova Iorque, EUA	20
3	Torre Eiffel	Paris, França	20
4	Burj Al Arab	Dubai, EAU	21
5	Ópera House	Sidnei, Austrália	20
6	Torres Petronas	Kuala Lumpur, Malásia	21
7	Catedral de São Basílio	Moscou, Rússia	20
8	Congresso Nacional	Brasília, Brasil	20
9	Big Ben	Londres, Reino Unido	20
10	Capitólio	Washington, EUA	21
11	Cidade Proibida	Pequim, China	20
12	Parthenon	Atenas, Grécia	20
13	Coliseu	Roma, Itália	20
14	Templo da Sagrada Família	Barcelona, Espanha	20
15	Casa Rosada	Buenos Aires, Argentina	20
16	Catedral de Colônia	Colônia, Alemanha	20
17	Taj Mahal	Agra, Índia	20
18	Pirâmide de Queops	Cairo, Egito	20
19	Taipei 101	Taipei, Taiwan	20
20	Torre CN	Toronto, Canadá	20

Tabela 5.10: Base de dados de instâncias de monumentos.

5.4 Reconhecimento de Instâncias de Monumentos

Esta seção descreve experimentos realizados que utilizam o descritor GRABED (Seção 3.2) em conjunto com outros descritores, principalmente o PHOW (Seção 2.6.2). Foi montada uma base de dados de instâncias de monumentos e utilizado o descritor PHOW e também uma combinação de descritores para resolver o problema da categorização de instâncias de monumentos. Alguns experimentos também foram feitos na base de dados Caltech-101.

5.4.1 A Base de Dados de Instâncias de Monumentos

Para este experimento, foi montada uma base de dados com foco em categorização de instâncias de monumentos. Para isto, coletaram-se imagens de importantes monumentos do sítio Flickr⁵. Todas as imagens coletadas possuem licença *Creative Commons*⁶. Como o foco da base de dados é o reconhecimento de instâncias, cada monumento diferente foi rotulado como uma categoria diferente na base de dados. São 20 categorias e aproximadamente 20 imagens de cada categoria. As fotos são vistas externas dos monumentos e foram coletadas com a maior resolução disponível. Foram coletadas fotos de diferentes pontos de vista, diferentes resoluções e a diferentes distâncias. Algumas foram tiradas à noite e outras durante o dia. Algumas apresentam oclusões parciais, principalmente com pessoas em frente aos monumentos. A Tabela 5.10 apresenta as 20 instâncias de monumentos. A Figura 5.3 mostra uma imagem para cada monumento da base de dados. Os experimentos desta seção mostram que o descritor GRABED combinado a um descritor de textura e outro descritor de cor melhoram o desempenho final da classificação de instâncias de monumentos nos testes efetuados.

5.4.2 Parametrização do GRABED

Foram testados três métodos para obter o mapa de bordas: Canny, Prewitt [GW02] e o gradiente morfológico externo (GME) (Seção 2.10.1). O Canny e o Prewitt retornam um mapa binário de

⁵www.flickr.com

⁶www.creativecommons.org



Figura 5.3: Um exemplo de cada instância da base de dados de instâncias de monumentos. Da esquerda para a direita, de cima para baixo, as imagens apresentam a mesma sequência apresentada na Tabela 5.10.

bordas e o GME retornam um mapa de bordas em níveis de cinza. Os métodos foram testados utilizando duas abordagens diferentes:

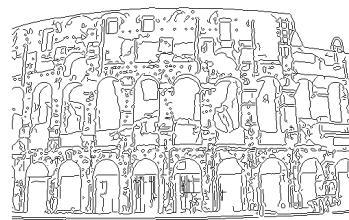
- *RGB*: o gradiente para cada um dos 3 canais do espaço de cor RGB é computado e o resultado final é a adição destes 3 gradientes intermediários.
- *Níveis de cinza*: a imagem original é convertida para níveis de cinza e o gradiente é computado.

A Figura 5.4 mostra uma imagem da categoria Coliseu e três mapas de bordas computados usando métodos diferentes: Canny, Prewitt e o GME.

Os resultados dos testes são apresentados na Tabela 5.11. O método Canny mostrou o melhor desempenho. Foram utilizados 5 imagens no conjunto de treinamento e 5 imagens no conjunto de testes neste experimento.



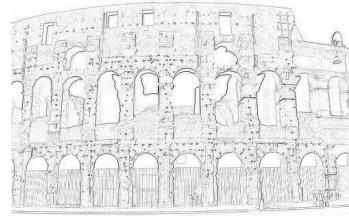
(a) Imagem original do Coliseu.



(b) Mapa de bordas calculado utilizando Canny.



(c) Mapa de bordas calculado utilizando Prewitt.



(d) Mapa de bordas calculando utilizando GME.

Figura 5.4: O mapa de bordas de uma imagem do Coliseu foi computada utilizando os três métodos testados diferentes: Canny, Prewitt e GME.

Método	Abordagem	Acc. (%)
Prewitt	RGB	32
Prewitt	Gray	32
Ext.Morph.Gradient	RGB	38
Ext.Morph.Gradient	Gray	30
Canny	RGB	40
Canny	Gray	45

Tabela 5.11: Acurácia do descriptor GRABED usando diferentes métodos para computar as bordas usando canais RGB ou imagens em níveis de cinza.

5.4.3 Parametrização do descriptor PHOW

O descriptor PHOW é calculado usando uma grade de pontos com espaçamento de 5 pixels entre os pontos horizontais e verticais. Nestes pontos, o descriptor SIFT é calculado utilizando várias regiões vizinhas. Nestes testes utilizou-se 2, 4, 6 e 10 pixels de diâmetros como escala dos descritores SIFT calculados pelo PHOW. Os parâmetros foram escolhidos empiricamente. Foram testados duas maneiras diferentes de criar o dicionário visual. No primeiro método, 100 imagens da base de dados inteira é escolhida aleatoriamente. No segundo método, 5 imagens de cada uma das 20 instâncias de monumentos são escolhidas aleatoriamente. Depois disto, 600000 descritores são escolhidos aleatoriamente entre os descritores extraídos das 100 imagens. Estes descritores são utilizados para serem aglomerados na construção do dicionário. O segundo método incrementa a acurácia em cerca de 8%.

A Figura 5.5 mostra os gráficos em alguns experimentos comparando a acurácia do descriptor PHOW no problema de categorização de instância com parâmetros utilizados na construção do dicionário visual. O Gráfico 5.5(a) mostra a acurácia dos resultados na base de dados de instância de monumentos para o descriptor PHOW em relação ao número de palavras visuais do dicionário

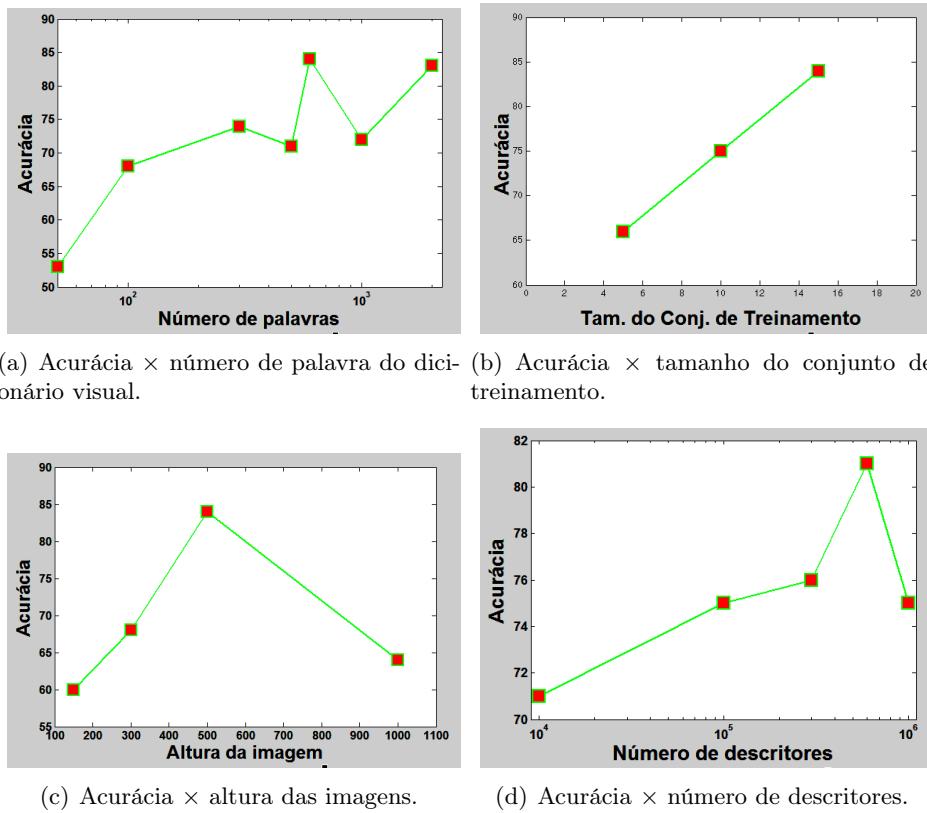


Figura 5.5: Gráficos mostrando a acurácia × número de palavras do dicionário visual, a acurácia × tamanho do conjunto de treinamento, a acurácia × altura das imagens e a acurácia × número de descritores

visual. Nos testes efetuados, a acurácia aumenta até 600 palavras, cai em torno de 1000 palavras e aumenta novamente em torno de 2000 palavras no dicionário. No restante dos testes, utilizaram-se 600 palavras no dicionário. O segundo gráfico mostra a acurácia do problema de categorização em relação ao tamanho do conjunto de treinamento. Em todas as execuções, o tamanho do conjunto de testes foi 5. A acurácia aumenta com o número de imagens no conjunto de treinamento. O terceiro gráfico mostra a acurácia do problema de categorização em relação ao tamanho da imagem. O melhor resultado foi obtido em imagens com altura de 500 pixels. O tamanho das imagens está diretamente conectado aos parâmetros de amostragem do descritor. O último gráfico mostra a influência do número de descritores usados para construir o dicionário. O melhor desempenho foi obtido utilizando 600000 descritores escolhidos aleatoriamente entre os descritores extraídos das 100 imagens do conjunto de treinamento.

A implementação da aplicação da categorização de objetos, utilizando o descritor PHOW, foi testada também na base de dados Caltech-101. Foram utilizadas 2000 palavras no dicionário e testado em um subconjunto de 5 classes. Estas 5 classes possuem centenas de imagens de amostras: BACKGROUND_Google, Faces, Faces_easy, Leopards e Motorbikes. Todas as imagens foram redimensionadas para terem 300 pixels no maior eixo, caso tenham mais de 300 pixels neste eixo. Os resultados deste experimento são apresentados na Tabela 5.12. Foram testados o subconjunto de 5 classes e a base de dados inteira com 102 classes.

5.4.4 Experimentos com Descritor Único

A Tabela 5.13 apresenta a acurácia da categorização de instâncias na base de dados de instâncias de monumentos para cada um dos descritores testados. Foram utilizadas 15 imagens para treinamento e 5 para testes, escolhidas aleatoriamente dentre as imagens de cada monumento. O descritor PHOW teve o melhor desempenho. O descritor GRABED obteve o segundo melhor desempenho, sendo superior ao outro descritor baseado em bordas testado, o PHOG.

Conjunto	# imagens treinamento	acc. (%)
5 classes	10	84,00
5 classes	50	96,00
5 classes	100	98,00
5 classes	200	98,80
5 classes	300	98,80
Completo	100	69,93

Tabela 5.12: Experimentos utilizando o descritor PHOW em categorização de objetos na base de dados Caltech-101.

Descriptor	Categoria	Acc. (%)
SIFT em pontos de interesse	textura	47
SIFT em pontos aleatórios	textura	41
PHOW	textura	81
SURF	textura	51
PHOG	bordas	38
GRABED	bordas	53
Histograma no espaço de cor AB	cor	15

Tabela 5.13: Acurácia dos descritores no problema de categorização na base de instâncias de monumentos.

5.4.5 Combinação de Descritores

Neste experimento, foram combinados três descritores que utilizam características diferentes da imagem: textura, bordas e cor. Foi escolhido o melhor descritor de cada categoria. Um modo intuitivo de criar um modelo de objetos é utilizar descritores que caracterizam sua textura, suas bordas e forma e sua cor. Usando a combinação dos descritores PHOW, GRABED e do histograma de cores no espaço AB, foi obtida uma acurácia de 84%. A Tabela 5.14 apresenta o desempenho individual de cada instância da base de dados. Novamente, foram escolhidas aleatoriamente 15 imagens para o conjunto de treinamento e 5 imagens para o conjunto de testes. O classificador utilizado foi o SVM utilizando núcleo qui-quadrado (Seção 2.7). Parâmetros utilizados: 600 palavras no dicionário visual formado a partir da aglomeração de 600000 descritores. Imagens redimensionadas para ter 500 pixels de altura, no máximo, e mantendo a razão de aspecto.

A Tabela 5.15 apresenta a matriz de confusão (Seção 2.9) dos resultados. As classes seguem a mesma numeração da Tabela 5.14. O pior desempenho individual foi obtidos pelas categorias Taipei 101 e Torre Eiffel (40% de acerto nas amostras de teste). Onze classes obtiveram acurácia total: Cristo Redentor, Torres Petronas, Congresso Nacional, Cidade Proibida, Partenon, Coliseu, Casa Rosada, Catedral de Colônia, Pirâmide de Queóps e Torre de CN. As maiores confusões foram Taipei 101, confundida como Torre Eiffel e Catedral de São Basílio, confundida como Coliseu em 40% dos casos, em ambas as situações.

5.4.6 Discussão

Neste experimento, utilizamos o descritor PHOW e uma combinação de descritores, incluindo um dos descritores propostos neste trabalho, baseado em famílias de granulometrias, para resolver o problema da categorização de instâncias de monumentos em uma base de dados criada a partir de imagens coletadas no sítio Flickr. A principal diferença entre esta base e outras bases de reconhecimento de objetos é que cada classe, na base de dados testada neste experimento, está relacionada a um objeto único, singular, uma instância. Cada classe é representada por imagens de um mesmo monumento. A base de dados SUN [XHE+10] é uma base de dados de cenários diferentes, um deles é a classe *Church/outdoor* com 921 imagens externas de diferentes igrejas ao redor do mundo. A Figura 5.6 mostra algumas imagens das duas bases para ressaltar a diferença entre elas. A Figura 5.6(a) mostra algumas igrejas da classe de igrejas, vista externa (classe *Church/outdoor*) da

#	Classe	Acc. (%)
1	Cristo Redentor	100
2	Estátua da Liberdade	60
3	Torre Eiffel	80
4	Burj Al Arab	80
5	Ópera House	60
6	Torres Petronas	100
7	Catedral de São Basílio	60
8	Congresso Nacional	100
9	Big Ben	60
10	Capitólio	80
11	Cidade Proibida	100
12	Parthenon	100
13	Coliseu	100
14	Templo da Sagrada Família	80
15	Casa Rosada	100
16	Catedral de Colônia	100
17	Taj Mahal	100
18	Pirâmide de Queóps	100
19	Taipei 101	20
20	Torre CN	100
Média		84

Tabela 5.14: Acurácia da classificação utilizando combinação dos descritores PHOW, GRABED e histograma de cores AB.

Gabarito	Resultados																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	3	0	0	0	0	0	2	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0
10	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
19	0	0	2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5

Tabela 5.15: Matriz de confusão para a classificação utilizando a combinação de descritores.

(a) Imagens da classe *Church/outdoor* da base de dados SUN.(b) Imagens da classe *Catedral de Colônia* da base de dados de instâncias de monumentos.(c) Imagens da classe *Templo da Sagrada Família* da base de dados de instâncias de monumentos.

Figura 5.6: São mostradas imagens que ilustram a diferença entre a base de dados SUN e a base de dados de instâncias de monumentos. A primeira figura mostra imagens de diferentes igrejas ao redor do mundo, pertencentes à classe de igrejas da base SUN. A segunda figura mostra diferentes imagens da mesma igreja, a Catedral de Colônia, que é rotulada como uma classe única na base de monumentos e a terceira figura mostra diferentes imagens do Templo da Sagrada Família, uma outra igreja que é rotulada como uma classe diferente na base de dados de instâncias de monumentos.

base de dados SUN. São imagens de igrejas diferentes ao redor do mundo. As Figuras 5.6(b) e 5.6(c) mostram imagens de igrejas porém da base de dados de instâncias de monumentos. Apesar de todas as imagens serem de igrejas, elas estão rotuladas com duas classes diferentes. A Figura 5.6(b) mostra imagens da Catedral de Colônia e estas imagens estão rotuladas como pertencentes a esta classe e a Figura 5.6(c) mostra imagens do Templo da Sagrada Família e estas imagens estão com este rótulo.

Neste experimento, foram testados vários parâmetros na construção de um dicionário visual para o descriptor PHOW, usando a abordagem do saco de palavras. O descriptor GRABED mostrou-se mais efetivo que o outro descriptor baseado em bordas, o PHOG, na base de dados testada. A combinação de um descriptor baseado em textura, um baseado em borda e um baseado em cor, melhorou o desempenho geral da categorização de monumentos.

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	42	3	1	1	3	
Faces	1	49	0	0	0	
Faces_easy	0	0	50	0	0	
Leopards	0	0	0	50	0	
Motorbikes	0	0	0	0	50	

Tabela 5.16: Matriz de confusão da abordagem saco de palavras.

5.5 Experimentos das Pirâmides de Regiões Circulares Concêntricas

Nesta seção, são descritos os experimentos feitos utilizando a Pirâmide de Regiões Circulares Concêntricas - PRCC (Seção 3.3). O primeiro experimento utiliza a tradicional abordagem do saco de palavras, o descriptor PHOW e o classificador SVM com núcleo qui-quadrado. Foram utilizadas 5 classes da base de dados Caltech-101: *BACKGROUND_Google*, *Faces*, *Faces_easy*, *Leopards* e *Motorbikes*. Foram utilizadas 100 imagens de cada uma das 5 classes na etapa de treinamento. Para a etapa de testes, foram utilizadas 50 imagens para cada classe totalizando 250 imagens. PRCC é testado em um problema de categorização de objetos usando as 5 classes citadas da base de dados Caltech-101. O descriptor PHOW foi utilizado com 5 pixels de distância entre os pontos da grade retangular e 2, 4, 6 e 10 pixels de vizinhança para calcular os 4 respectivos descritores SIFT por posição da grade. Foi obtida uma acurácia final de 96,4%. A Tabela 5.16 apresenta a matriz de confusão dos resultados. As linhas mostram as classes do gabarito e as colunas as classes reportadas pelo classificador. Os experimentos desta seção mostram que a PRCC é mais robusta a rotação de objetos que a quantização espacial utilizando regiões retangulares nos testes efetuados.

No segundo experimento, todos os parâmetros do primeiro experimento são repetidos. Uma rotação aleatória é aplicada ao conjunto de teste. A Figura 5.7 mostra algumas imagens usadas neste experimento. As imagens são redimensionadas por um fator F calculado de acordo com a seguinte fórmula:

$$F = \sqrt{\left(\frac{\text{altura}}{2}\right)^2 + \left(\frac{\text{largura}}{2}\right)^2},$$

onde altura e largura são as medidas originais da imagem. Depois que a imagem é redimensionada por um fator F , ela é rotacionada por um ângulo aleatório e cortada na parte central para a imagem manter o mesmo tamanho da imagem original. Este procedimento evita que regiões de fundo da imagem estejam na imagem final de teste. A acurácia deste experimento caiu para 42%. A Tabela 5.17 apresenta a matriz de confusão.

No terceiro experimento é utilizada a abordagem do saco de palavras acrescida da quantização espacial. Os parâmetros do primeiro experimento são repetidos aqui. A quantização espacial possui 2 níveis: o primeiro nível é formado pela imagem inteira e, no segundo nível, a imagem é dividida em quatro regiões. Nenhuma rotação foi aplicada no conjunto de teste. Este experimento atingiu 96,8% de acurácia. Tabela 5.18 apresenta a matriz de confusão dos resultados.

No próximo experimento, foram aplicados os parâmetros do experimento 3 e o conjunto de teste foi rotacionado por um ângulo aleatório (como no experimento 2). A acurácia final deste experimento foi 43,2%. A Tabela 5.19 apresenta a matriz de confusão dos resultados.

Este experimento utiliza a abordagem proposta neste trabalho para adicionar informação espacial ao saco de palavras. A Pirâmide de Regiões Circulares Concêntricas usa 4 regiões. Três regiões circulares centradas no centro da imagem e com raios de tamanho 15%, 30% e 45% da altura da

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	37	0	11	0	2	
Faces	8	1	40	0	1	
Faces_easy	2	0	48	0	0	
Leopards	37	0	1	11	1	
Motorbikes	35	0	7	0	8	

Tabela 5.17: Matriz de confusão do experimento usando saco de palavras e testado em imagens aleatoriamente rotacionadas.

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	43	3	2	2	0	
Faces	1	49	0	0	0	
Faces_easy	0	0	50	0	0	
Leopards	0	0	0	50	0	
Motorbikes	0	0	0	0	50	

Tabela 5.18: Matriz de confusão usando a abordagem do saco de palavras adicionada à quantização espacial.

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	45	0	5	0	0	0
Faces	14	5	31	0	0	0
Faces_easy	5	0	45	0	0	0
Leopards	42	0	3	5	0	0
Motorbikes	35	0	7	0	8	

Tabela 5.19: Matriz de confusão do experimento utilizando o saco de palavras acrescido da quantização espacial com imagens de teste rotacionadas.

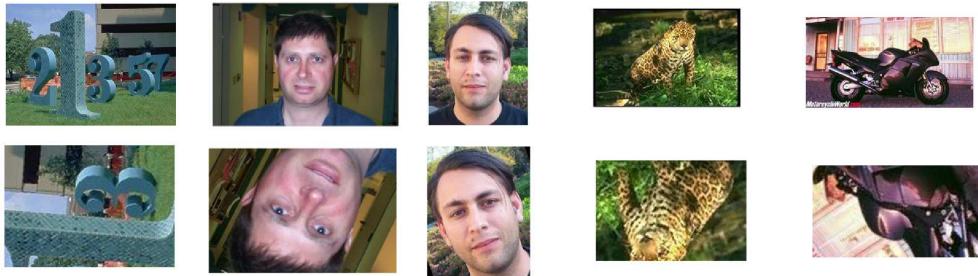


Figura 5.7: Esta figura apresenta uma amostra de cada classe do conjunto de teste da base de dados Caltech-101. A primeira linha mostra a imagem original e a segunda linha mostra a mesma imagem rotacionada. Da esquerda para direita, as classes são: BACKGROUND_Google, Faces, Faces_Easy, Leopards e Motorbikes.

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	47	1	0	1	1	
Faces	0	50	0	0	0	
Faces_easy	0	0	50	0	0	
Leopards	1	0	0	49	0	
Motorbikes	0	0	0	0	50	

Tabela 5.20: Matriz de confusão da abordagem PRCC com imagens de teste não rotacionadas.

imagem. A quarta região é a imagem inteira. O conjunto de testes não é rotacionado neste experimento. A acurácia obtida foi de 98,4%. A Tabela 5.20 apresenta a matriz de confusão deste experimento.

Finalmente, o último experimento mostra a aplicação da abordagem usando o PRCC em imagens aleatoriamente rotacionadas. Este experimento atingiu 76,8% de acurácia e a matriz de confusão deste experimento é apresentada na Tabela 5.21.

5.5.1 Discussão dos Resultados

Pode-se ver nos experimentos usando o saco de palavras com ou sem a quantização espacial (primeiro e terceiro experimento) que a categorização de objetos tem um índice de acerto bastante alto para as 5 classes testadas nesta base de dados. Foram menos de 10 erros em 250 imagens testadas, uma acurácia de mais de 96%. Três classes testadas, *Faces_easy*, *Leopards* e *Motorbikes*, obtiveram 100% de acertos. As duas classes de faces, *Faces* e *Faces_easy*, possuem uma grande

Gabarito	Resultados	BCK_Google	Faces	Faces_easy	Leopards	Motorbikes
BCK_Google	36	3	1	7	3	
Faces	7	43	0	0	0	
Faces_easy	8	0	42	0	0	
Leopards	4	0	0	46	0	
Motorbikes	13	0	0	12	25	

Tabela 5.21: Matriz de confusão da abordagem PRCC com imagens de teste rotacionadas.

similaridade extra-classe. Porém, ainda assim, obtiveram-se 98% e 100% de acerto respectivamente nos dois experimentos. A classe *BACKGROUND_Google* é uma classe complexa. Ela tenta modelar qualquer tipo de categoria, o que é impossível. Contudo, neste problema em um mundo fechado, esta classe obteve 84% e 86% de acurácia nos dois experimentos.

Os resultados são completamente diferentes quando o conjunto de testes é rotacionado. A acurácia final da do problema de categorização, nas 5 classes testadas, cai para meros 42% sem o uso da quantização espacial (segundo experimento) e 43,2% usando a quantização espacial (quarto experimento). Um resultado completamente diferente, quando comparado com os resultados dos experimentos 1 e 3.

O experimento 5 mostra a quantização espacial proposta neste trabalho, em conjunto de teste normal, sem a rotação. O resultado foi muito bom e atingiu 98,4% de acurácia. O resultado da categorização não diminuiu usando-se a quantização espacial proposta neste trabalho. Pelo contrário, até aumentou um pouco. A vantagem da utilização da PRCC, em relação à quantização espacial com regiões retangulares, é mais evidente quando uma rotação aleatória é aplicada nas imagens do conjunto de testes. A PRCC obteve 76,8% de acurácia comparados aos 43,2% de acurácia da abordagem do saco de palavras junto com a quantização espacial. Um incremento de quase 78% quando comparado com a quantização espacial em um cenário onde a rotação está presente.

5.6 Experimentos de Categorização Refinada

Esta seção descreve os experimentos realizados utilizando o novo método de categorização refinada, proposto no Capítulo 4. A Seção 5.6.1 descreve detalhadamente a base de dados utilizada nos experimentos. A Seção 5.6.2 descreve os experimentos e mostra os resultados e a Seção 5.6.3 discute os resultados obtidos. Os experimentos desta seção mostra a utilização do descritor GRABED e do novo método de categorização refinada proposta neste trabalho melhoraram o desempenho da abordagem do saco de palavras no problema de categorização refinada de pássaros.

5.6.1 Base de Dados Caltech-UCSD Birds-200 2011

A base de pássaros Caltech-UCSD Birds-200 2011 foi descrita em [WBW⁺¹¹] e é uma extensão da base de dados descrita em [WBM⁺¹⁰]. É uma base de dados focada em reconhecimento de categorias muito parecidas, a categorização refinada. São 200 espécies diferentes de pássaros, cada espécie é um rótulo diferente na base de dados.

A classificação de pássaros é uma tarefa extremamente difícil, seja para algoritmos de categorização ou para seres humanos. As espécies de pássaros podem variar bastante de aparência visual, por exemplo, um albatroz e um rouxinol (Figura 5.8). Por outro lado, espécies diferentes podem ser absolutamente iguais, como, por exemplo, algumas espécies de pardais (Figura 5.9). Uma mesma espécie de pássaro pode apresentar uma diferença visual grande: o mesmo pássaro pode apresentar formas bem diversas quando está em vôo, nadando ou em terra. A Figura 5.10 mostra três imagens da mesma espécie de albatroz em situações diferentes: em vôo, na água e em terra. A ave apresenta formas bem diferentes em cada uma das situações mostradas. Algumas fotos da base de dados mostram os pássaros parcialmente oclusos.

A base de dados contém 11788 imagens rotuladas em 200 espécies diferentes de pássaros. Cada foto apresenta apenas um pássaro. As imagens foram recolhidas do sítio Flickr⁷ e anotadas por usuários do Mechanical Turk⁸ que é um serviço da Amazon⁹ onde pessoas são contratadas para tarefas que necessitam da intervenção humana. Foram anotados manualmente, para cada imagem:

- A caixa envoltória em torno do pássaro.
- 28 atributos visuais. Os atributos incluem informações como padrão das asas, formato dos bicos, formato dos rabos e cores das principais partes dos pássaros.

⁷ www.flickr.com

⁸ www.mturk.com

⁹ www.amazon.com



Figura 5.8: Esta figura mostra a imagem de um albatroz e de um rouxinol da base de dados Caltech-UCSD Birds-200 2011



Figura 5.9: Esta figura mostra imagens de duas espécies diferentes de pardais da base de dados Caltech-UCSD Birds-200 2011

- A localização e a visibilidade de 15 partes dos pássaros:

1. costas;
2. bico;
3. barriga;
4. peito;
5. coroa (alto da cabeça);
6. testa;
7. olho esquerdo;
8. olho direito;

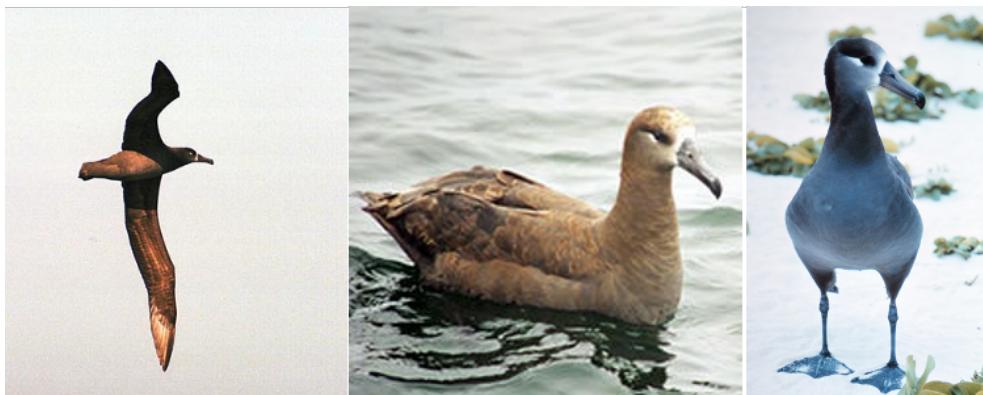


Figura 5.10: Esta figura mostra imagens da mesma espécie de albatroz em situações diferentes: em vôo, na água e na terra.

9. perna esquerda;
10. perna direita;
11. asa esquerda;
12. asa direita;
13. cauda;
14. garganta;
15. nuca.

Os principais objetivos da base é se tornar uma base de dados de referência para:

- Testar algoritmos de categorização refinada: as espécies de pássaros são visualmente muito parecidas. Algumas espécies são indistinguíveis por pessoas leigas. Assim, esta base se torna um desafio para algoritmos de categorização de objetos.
- Categorização multiclasse: 200 categorias é uma quantidade razoável, o que se torna também um desafio. São 10 vezes mais categorias que a base de dados do desafio VOC Pascal.
- Métodos de detecção de partes: a detecção de partes tem-se tornado uma importante abordagem de reconhecimento de objetos [FMR08]. Entretanto não existem muitas bases com partes anotadas. Esta base ajuda a preencher esta lacuna.
- Algoritmos baseados em atributos: uma abordagem muito recente que utiliza atributos visuais [LNH09]. Esta base se diferencia por, além de ter atributos anotados, ter também suas localizações.

Neste experimento, foi utilizado um subconjunto da base de dados contendo um total de 13 classes distintas. Foram utilizadas as 7 espécies distintas do gênero *Vireo* existentes na base de dados e 6 espécies de pica-paus. Este subconjunto específico foi utilizado nos experimentos de [YBFF12] e de [FOZ⁺11]. A Tabela 5.22 apresenta informações sobre as 13 espécies utilizadas nestes experimentos: o número da espécie na base de dados, o nome científico da espécie e o número de imagens disponíveis na base para treinamento e teste, respectivamente, segundo a sugestão da própria base de dados. A Figura 1.2(c) mostra uma imagem para cada uma das espécies de pássaros do gênero *Vireo* utilizadas nos experimentos de categorização refinada deste trabalho. De cima para baixo, da esquerda para a direita, os pássaros estão na mesma ordem das 7 primeiras linhas da Tabela 5.22. A Figura 5.11 mostra as 6 espécies de pica-paus utilizadas neste experimento. Da esquerda para a direita e de cima para baixo, são imagens das 6 últimas espécies listadas na Tabela 5.22.

Para ilustrar a anotação das partes dos pássaros, a Figura 5.12 mostra uma imagem de um pássaro do gênero *Vireo* da classe 151 da base de dados com a anotação referente ao bico assinalada com uma marca quadrada em vermelho.

5.6.2 Experimentos

Foi realizado um número grande de experimentos (cerca de 60) com variações quanto aos classificadores utilizados, descritores utilizados, variações na configuração dos dicionários visuais e configurações dos descritores. Nenhuma configuração obteve o melhor desempenho em todas as classes, porém algumas configurações foram bem sucedidas em classes específicas. Apesar de ter sido feitos testes utilizando outros classificadores, o classificador linear SVM com função núcleo qui-quadrado sempre obteve o melhor desempenho. A classificação das espécies em 2 níveis também foi uma abordagem bem sucedida que obteve os melhores resultados. A utilização das quantizações especiais flexíveis (Seção 3.1) também esteve presente em todas as configurações que obtiveram os melhores resultados. As principais diferenças foram em relação aos descritores utilizados, configuração destes descritores e configuração dos dicionários visuais utilizados.



Figura 5.11: Esta figura mostra as 6 espécies de pica-paus utilizados neste experimento. Juntamente com as 7 espécies de pássaros do gênero Vireo, perfazem o total de 13 espécies utilizadas nos experimentos de categorização refinada.



Figura 5.12: Pássaros da classe 151 da base de pássaros com o bico marcado com um quadrado vermelho.

#	Nome científico	# treinamento	# teste
151	Vireo atricapilla	30	21
152	Vireo solitarius	30	30
153	Vireo philadelphicus	30	29
154	Vireo olivaceus	30	30
155	Vireo gilvus	30	30
156	Vireo griseus	30	30
157	Vireo flavifrons	30	29
187	Picoides dorsalis	30	20
188	Dryocopus pileatus	30	30
189	Melanerpes carolinus	30	30
190	Picoides borealis	29	29
191	Melanerpes erythrocephalus	30	30
192	Picoides pubescens	30	30

Tabela 5.22: *Espécies de pássaros utilizadas nos experimentos de categorização refinada deste trabalho.*

Classe	AP(%)	Configuração
151	33,4	7
152	58,8	1
153	36,3	1
154	21,2	4
155	24,9	7
156	20,3	5
157	24,7	2
187	46,4	6
188	40,4	3
189	56,5	3
190	70,0	3
191	43,9	3
192	45,8	3
MAP	40,2	

Tabela 5.23: *A precisão média (AP) para cada classe testada e a configuração utilizada.*

A Tabela 5.23 apresenta a melhor precisão média obtida para cada uma das 13 classes e também a média das precisões médias obtida com estes resultados. As configurações utilizadas nos experimentos que obtiveram os melhores resultados são descritas nas tabelas 5.24, 5.25, 5.26, 5.27, 5.28, 5.29 e 5.30.

A Tabela 5.31 apresenta a comparação dos resultados obtidos neste trabalho com outros resultados de categorização refinada utilizando o mesmo subconjunto reportado em [YBFF12].

A Tabela 5.32 mostra o tempo de processamento de uma configuração da aplicação de categorização refinada de pássaros que utiliza imagens de treinamento expandidas, 2600 imagens de treinamento, 368 imagens de teste e descritores PHOW colorido, PHOW em níveis de cinza, GRABED, PPE, LBP e SURF e um dicionário visual com 1000 palavras visuais. O tempo total de processamento foi de 12 horas, 17 minutos e 14,29 segundos.

Foi feito um outro experimento que utiliza as anotações das partes e também as PRCCs. As imagens da classe 151 foram rotuladas como pertencentes à classe 1 e as imagens das outras 12 classes foram mapeadas como pertencentes à classe 2. Foi utilizado um dicionário local de 600 palavras para a classe 151, construído utilizando-se descritores PHOW colorido. Foi criado um modelo para cada parte das aves utilizando as partes visíveis. Usou-se a PRCC centrada no ponto de localização da parte com circunferências iguais a 15%, 30% e 45% da altura da imagem. Assim, a PRCC para cada parte, tem 4 regiões: 3 circunferências e a última região é a imagem inteira.

Descritores	Parâmetros
•GRABED	bordas Canny, $L_0 = 2$, $R = 12$, 3 escalas, fator de redução 0,7, QEF com 2 níveis (1 e 4 regiões)
•Histograma RGB	15 intervalos, QEF com 2 níveis (1 e 4 regiões)
•PPE	3 escalas, fator de redução 0,7, bordas Canny, QEF com 2 níveis (1 e 4 regiões)
•PHOW colorido	QEF com 2 níveis (1 e 4 regiões), intervalo de 3 pixels na grade e SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•PHOG	um nível e 8 intervalos de ângulos
Dicionário	Parâmetros
•PHOW Colorido	600 palavras, 1000000 de descritores aleatórios
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.24: Parâmetros da Configuração 1.

Descritores	Parâmetros
•GRABED	Todos os descritores da Configuração 1 bordas Canny, $L_0 = 2$, $R = 15$, 2 escalas, fator de redução 0,5, QEF com 2 níveis (1 e 9 regiões)
•Histograma RGB	5 intervalos, sem QEF
•PPE	2 escalas, fator de redução 0,5, bordas Canny, sem QEF
•PHOW colorido	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•PHOW níveis de cinza	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•LBP	células de 15 pixels e QEF de 2 níveis (1 e 4 regiões)
•SURF	QEF de 2 níveis (1 e 9 regiões)
Dicionário	Parâmetros
•PHOW Colorido	200 palavras, 1000000 de descritores aleatórios
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.25: Parâmetros da Configuração 2.

Usando classificado SVM linear com função núcleo qui-quadrada, a confiança dos modelos criados obtidos por validação cruzada dos dados de treinamento utilizando o método k-fold com $k = 5$ estão apresentados na Tabela 5.33. Obviamente, os resultados não podem ser comparados com os da Tabela 5.23 visto que a localização da parte nas amostras testadas também foi utilizada. Porém o experimento é válido para mostrar a adequação do uso da PRCC para descrever as partes e também para mostrar que, na categorização refinada, as partes podem ser mais discriminantes que o todo.

Ao comparar apenas as 7 espécies do gênero *Vireo*, que são bem mais parecidas entre si, pode-se entender a diferença de resultados entre as partes. As PRCCs utilizam descritores de 3 regiões circulares centradas no ponto definidido como a parte que a pirâmide está modelando e, por último, descritores da imagem inteira são utilizados. Assim, pode-se entender porque a perna não demonstrou ser discriminante. As regiões em torno das pernas das 7 espécies de pássaros não possuem diferenças significativas. A espécie da classe 151 possui uma diferença em relação às outras 6 espécies na região da cabeça. A cabeça de cor preta, com olhos em tom vermelho circundados por uma mancha branca, é a marca que distingue estes pássaros dos outros. Pode-se perceber que as partes mais discriminantes estão próximas da região da cabeça (bico, coroa, testa e nuca). Os olhos foram a exceção, esperava ser mais discriminante que o resultado obtido. A pouca discriminação do corpo do pássaro (costas, barriga, asa e garganta) é explicada pelas semelhanças que existem

Descritores	Parâmetros
•GRABED	Todos os descritores da Configuração 1 bordas Canny, $L_0 = 2$, $R = 15$, 2 escalas, fator de redução 0,5, QEF com 2 níveis (1 e 9 regiões)
•Histograma RGB	5 intervalos, sem QEF
•PPE	2 escalas, fator de redução 0,5, bordas Canny, sem QEF
•PHOW colorido	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•PHOW níveis de cinza	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•LBP	células de 15 pixels e QEF de 2 níveis (1 e 4 regiões)
•SURF	QEF de 2 níveis (1 e 9 regiões)
Dicionário	Parâmetros
•PHOW Colorido	1000 palavras, 2000000 de descritores aleatórios
•PHOW níveis de cinza	1000 palavras, 2000000 de descritores aleatórios
•SURF	1000 palavras, 2000000 de descritores aleatórios
•LBP	1000 palavras, 2000000 de descritores aleatórios
Imagens expandidas	Sim
Sobreposição	15 pixels

Tabela 5.26: Parâmetros da Configuração 3.

Descritores	Parâmetros
•GRABED	bordas Canny, $L_0 = 2$, $R = 12$, 3 escalas, fator de redução 0,7, QEF com 2 níveis (1 e 4 regiões)
•PPE	3 escalas, fator de redução 0,7, bordas Canny, QEF com 2 níveis (1 e 4 regiões)
•PPD	3 escalas, fator de redução 0,7, os elementos estruturantes disco têm raio inicial igual 1 e incrementando até 500 pixels, 15 intervalos de níveis de cinza
Dicionário	Parâmetros
•Sem dicionários	
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.27: Parâmetros da Configuração 4.

dos corpos dos pássaros da classe 151 com os corpos dos pássaros das outras 6 classes. Todos os pássaros possuem o corpo com formas semelhantes e em tons aproximados de amarelo e cinza.

5.6.3 Discussão

Este trabalho propôs a utilização das Quantizações Espaciais Flexíveis e de 3 descritores baseados em granulometria morfológica, em um esquema de classificação de 2 níveis, com modelos específicos para cada uma das classes testadas para resolver um problema de categorização refinada em um subconjunto de 13 classes da base de pássaros Caltech-UCSD Birds-200 2011. O método proposto, baseado na abordagem do saco de palavras, obteve um desempenho superior ao de outros trabalhos que também utilizaram a abordagem do saco de palavras para resolver o problema e obteve resultado praticamente igual ao de um trabalho que utilizou as anotações manuais disponíveis na base de dados. O trabalho proposto aqui não utilizou os atributos visuais ou a anotação manual da localização das partes das aves para obtenção do resultado final da Tabela 5.23. Utilizou-se apenas a caixa envoltória na fase de treinamento. As imagens de teste foram utilizadas por inteiro, não foram utilizadas as caixas envoltórias das imagens de teste. A vantagem de não utilizar as anotações

Descritores	Parâmetros
•GRABED	bordas morfológicas, $L_0 = 2$, $R = 12$, 3 escalas, fator de redução 0,7, QEF com 2 níveis (1 e 9 regiões)
•PPE	3 escalas, fator de redução 0,7, bordas Canny, QEF com 2 níveis (1 e 9 regiões)
•PHOW colorido	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
•PHOG	um nível e 8 intervalos de ângulos
Dicionário	Parâmetros
•PHOW colorido	600 palavras, 1500000 descritores aleatórios
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.28: Parâmetros da Configuração 5.

Descritores	Parâmetros
•GRABED	bordas morfológicas, $L_0 = 2$, $R = 12$, 3 escalas, fator de redução 0,7, QEF com 2 níveis (1 e 9 regiões)
Dicionário	Parâmetros
•Sem dicionários	
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.29: Parâmetros da Configuração 6.

manuais, como a técnica proposta aqui, é que existe um alto custo em produzi-las e o fato das anotações serem específicas para uma determinada base de dados prejudica a generalização das técnicas que as utilizam. Outra vantagem de nossa técnica em relação à técnica proposta por [YBFF12] que obteve o melhor resultado reportado pela literatura, é que por utilizar a abordagem do saco de palavras, a mesma técnica poderia ser utilizada em categorização refinada e também em aplicações de categorização de objetos comuns. A técnica que foi proposta neste trabalho é basicamente incremental em relação a abordagem clássica do saco de palavras. Não existe perda de generalidade na técnica nem aumento absurdo no custo computacional. Além disto, a proposta deste trabalho não é específica para categorização refinada nem para uma determinada base de dados.

Como descrito no próprio trabalho de [YBFF12], a abordagem do saco de palavras faz algumas generalizações que levam à perda de informações. Para combater isto, utilizou-se um modelo específico por classe. Assim, pode-se ter uma configuração de dicionário, combinação de descritores e parametrização de descritores que consigam captar as pequenas diferenças entre a classe que está sendo modelada e as demais classes testadas da base de dados. O aumento do desempenho com modelo específico por classe foi observado já nos experimentos da Seção 5.3 que antecederam os experimentos desta seção.

Uma observação interessante é que as classes com melhores resultados são praticamente as

Descritores	Parâmetros
•PHOW colorido	QEF com 2 níveis (1 e 9 regiões), intervalo de 3 pixels na grade SIFT calculado com raio de 2, 3, 5, 7 e 10 pixels
Dicionário	Parâmetros
•PHOW colorido	600 palavras, 1500000 descritores aleatórios
Imagens expandidas	Não
Sobreposição	15 pixels

Tabela 5.30: Parâmetros da Configuração 7.

Abordagem	MAP(%)	Referência
Casamento de padrões	44,73	[YBFF12]
Anotações	40,25	[FOZ ⁺¹¹]
Saco de palavras	40,20	proposta neste trabalho
Casamento de padrões	39,76	[YBFF12]
Saco de palavras	37,12	[VDSGS10a]
Saco de palavras	37,02	[BWS ⁺¹⁰]

Tabela 5.31: Resultados de abordagens de categorização refinada utilizando o mesmo subconjunto da base de pássaros Caltech-UCSD Birds-200 2011 reportados na literatura.

Etapa	Tempo (em segs.)
Construção do dicionário visual	8960,28
Extração dos descritores de treinamento	28626,68
Extração dos descritores de teste	6114,49
Construção do modelo	524,83
Teste	8,01
Tempo total	44234,29

Tabela 5.32: Tempo de processamento em cada etapa do processo da aplicação de categorização refinada de pássaros

mesmas em quase todas as configurações testadas e as classes com os piores resultados também não se alteram muito de uma configuração para outra. O que leva a intuir que algumas classes são mais fáceis de serem reconhecidas, enquanto que outras são mais difíceis. A classe que sempre obteve os melhores resultados foi a classe 190 e as classes com piores resultados sempre foram as classes 154, 155, 156 e 157. Visualmente, pode-se notar esta diferença. As espécies do gênero *Vireo* são mais difíceis de serem reconhecidas que as espécies de pica-paus.

Outra técnica que se mostrou válida no ganho de desempenho da aplicação foi a geração de novas imagens de treinamento a partir de transformações aplicadas às imagens originais. A desvantagem desta técnica é o aumento significativo do tempo de treinamento.

Outro resultado interessante é que o descritor GRABED esteve presente no melhor modelo criado em 11 das 13 classes utilizadas neste experimento. O descritor PHOW, reportado como um dos melhores descritores para reconhecimento de objetos, também esteve presente em 11 modelos de classes. O GRABED mostrou melhores resultados que o outro descritor baseado em bordas utilizados nestes testes, o PHOG. Apenas para quantificar esta diferença, o GRABED utilizado

Parte	AP(%)
Costas	40
Bico	80
Barriga	20
Peito	40
Coroa	73
Testa	100
Olhos	20
Perna	0
Asa	0
Nuca	67
Rabo	60
Garganta	0

Tabela 5.33: Resultados da validação cruzada de modelos criados a partir da localização das partes utilizando a PRCC, junto com dicionário visual de 600 palavras construídos com descritores PHOW colorido.

como único descritor na Configuração 6, que foi o melhor modelo para a classe 187. Nesta mesma configuração, o MAP geral, para todas as classes, foi igual a 18,7%. Em outro experimento, com configuração praticamente idêntica, porém utilizando apenas o PHOG, o MAP geral para todas as classes foi de 16,6%. Uma diferença de 12,7% em prol do GRABED. Em várias outras configurações esta diferença se mostrou similar. O PHOG apareceu na composição de 9 dos melhores modelos para classes específicas, porém, sempre em conjunto com vários outros descritores enquanto que o GRABED esteve sozinho em uma configuração e em conjunto com os outros 2 descritores propostos neste trabalho em outra configuração.

O experimento, que utiliza as localizações das partes anotadas manualmente para criar modelos, mostra a adequação da PRCC para descrever estas partes e mostra que a melhora de desempenho nesta base de dados deve passar pela utilização destas partes anotadas durante a fase de treinamento.

Capítulo 6

Conclusões

Este trabalho trata o problema do reconhecimento de objetos, um dos mais ativos da área de VC. Foram propostas duas novas técnicas de quantizações espaciais para serem utilizadas junto com a abordagem do saco de palavras em aplicações de reconhecimento de objetos. Foram propostos, ainda, três novos descritores que utilizam granulometria morfológica e são utilizados na criação de modelos de classes para aplicações de reconhecimento de objetos. Estas propostas foram utilizadas em uma nova técnica para tratar um dos problemas mais desafiadores da área de reconhecimento de objetos, a categorização refinada que é a categorização de objetos de classes muito parecidas. O novo método proposto, neste trabalho, para categorização refinada mostrou o melhor resultado reportado até o presente momento utilizando a abordagem do saco de palavras. O resultado foi praticamente idêntico ao de um trabalho que se utiliza da localização das partes das aves anotadas manualmente e ficou atrás da técnica proposta por [YBFF12], que utiliza casamento de padrões e um custo computacional bem maior, lidando com vetores de características de até quase 900000 dimensões.

Uma das novas quantizações espaciais proposta neste trabalho é a Quantização Espacial Flexível (QEF). Ela é baseada na tradicional SPM proposta por [LSP06] e elimina várias de suas restrições. A QEF permite qualquer número de níveis e quaisquer divisões espaciais retangulares dentro de um mesmo nível. Os pesos utilizados nos diferentes níveis também não seguem ao proposto pelo artigo original. Não existe a eliminação dos casamentos de descritores que ocorrem em níveis mais profundos da contagem de casamentos dos níveis mais altos, como proposto pelo artigo original e adicionou-se um parâmetro que determina o número de pixels em que a área de uma região avança sobre uma outra região adjacente dentro de um nível. A ideia destas flexibilizações é que, durante a etapa de treinamento, a melhor configuração da quantização espacial poderá ser testada para a criação de um modelo de classe mais efetivo. Utilizou-se um subconjunto de classes de uma base de dados pública, a base de pássaros Caltech-UCSD Birds-200 2011, onde foram feitas comparações entre as diferenças propostas pela QEF e a SPM tradicional em uma aplicação de categorização de objetos. A QEF mostrou uma acurácia 6% melhor nos resultados.

Este trabalho propõe também 3 descritores baseados em aberturas morfológicas. O primeiro descritor proposto, chamado de GRABED, é calculado aplicando-se uma família de granulometrias no mapa de bordas da imagem o que difere do uso tradicional da granulometria que geralmente é aplicada diretamente nos níveis de cinza da imagem original. Os EEs utilizados são todos lineares e possuem comprimento crescente. Um conjunto de EEs lineares crescentes, com uma mesma orientação, forma uma família de EEs que gera um padrão de espectro. São utilizadas várias orientações diferentes e vários padrões de espectro são gerados no processo de cálculo do descritor. O GRABED utiliza a QEF e também calcula os mapas de bordas utilizando diferentes escalas. O descritor mede a distribuição da orientação, como o HOG, e também a distribuição das extensões das bordas. O GRABED foi utilizado, em combinação com outros descritores, no reconhecimento de instâncias de monumentos em uma base coletada do sítio Flickr. O GRABED foi comparado a outros descritores utilizando cinco classes da base Caltech-101 e obteve resultado similar aos melhores descritores reportados na literatura e também utilizados nesta comparação. Finalmente, o GRABED foi utili-

zado como descritor no novo método proposto de categorização refinada e esteve presente no melhor modelo de 11 das 13 classes testadas neste experimento. Outra importante observação é que, no experimento da categorização refinada que utilizou a base de dados Caltech-UCSD Birds-200 2011, o descritor GRABED obteve resultados superiores ao PHOG, outro descritor que caracteriza o gradiente das imagens e é listado na literatura como um dos melhores descritores para reconhecimento de objetos.

O descritor PPE computa um histograma normalizado de todos os padrões 3x3 presentes no mapa de bordas de uma imagem. O descritor também utiliza a QEF e várias escalas em seu cômputo. Por causa do tamanho final do descritor, usou-se uma técnica de seleção de características para limitar os histogramas apenas aos 50 padrões 3x3 mais discriminantes. Ele também foi utilizado na aplicação de categorização refinada nos melhores modelos de 10 das 13 classes testadas da base de pássaros.

Finalmente, o último descritor proposto, neste trabalho, é o PPD que, ao contrário dos outros dois, aplica uma série de aberturas em imagens binárias calculadas através de intervalos dos valores de níveis de cinza da imagem original. As aberturas utilizam EEs circulares com raios crescentes onde o raio atual vale duas vezes o raio anterior. Este descritor é utilizado na criação do melhor modelo da classe 154 na aplicação de categorização refinada testada neste trabalho. Aliás, o melhor modelo da classe 154 utiliza os 3 descritores propostos neste trabalho, e apenas eles.

As Quantizações Espaciais tradicionais, como a SPM, melhoraram a abordagem do saco de palavras porém não são invariantes a rotação. Vários objetos presentes em uma cena podem estar rotacionados, como, por exemplo, um livro, uma caneta ou uma lata de refrigerante. As Pirâmides de Regiões Circulares Concêntricas (PRCC), propostas neste trabalho, são uma nova divisão espacial que, utilizadas em conjunto com o saco de palavras e sem perder o incremento de desempenho das quantizações espaciais, aumentam a robustez a variações de orientações dos objetos presentes em cena. A comparação entre a quantização espacial com divisões retangulares e a PRCC foi efetuada utilizando 5 classes de uma base de dados pública, a Caltech-101, e os resultados foram surpreendentes. Além de melhorar os resultados da categorização de objetos em situação normal, sem rotação dos objetos, em 1,7%, melhorou os resultados em um cenário de rotação dos objetos em cena, em 78%.

Por último, este trabalho utilizou as técnicas propostas para resolver um dos problemas mais difíceis da área de reconhecimento de objetos, a categorização refinada. Utilizou-se a base de pássaros Caltech-UCSD Birds-200 2011 para efetuar os experimentos em um subconjunto de 13 classes distintas, utilizado em vários trabalhos anteriores. O método de categorização refinada, proposto aqui, utiliza uma classificação em 2 níveis e cria 14 modelos para cada descritor visual utilizado na aplicação. Um dos modelos classifica a amostra como pertencente ao gênero *Vireo* ou às espécies de pica-paus e os outros 13 modelos classificam a amostra na classe final. Modelos, que obtiveram um nível de confiança baixo durante uma etapa de validação cruzada, na fase de treinamento, são eliminados. Os resultados dos modelos restantes são combinados. A maior média das probabilidades é utilizada para o rótulo final da amostra. Além desta classificação em 2 níveis utilizando vários descritores e modelos, utilizou-se uma configuração ideal por classe testada. O resultado final desta técnica foi superior a todas as outras técnicas que utilizaram a abordagem do saco de palavras, empatou com um trabalho que utilizou a localização das partes anotadas manualmente e só obteve resultado inferior a de uma técnica que utiliza casamento de padrões e uma versão modificada da técnica de *bagging*, a um custo computacional alto: são utilizados vetores de características de quase 900000 dimensões. A QEF está presente em todos os modelos com melhor desempenho, e pelo menos, um dos 3 descritores propostos neste trabalho está presente em 11 dos 13 modelos criados para as classes. Foi feito também um pequeno experimento utilizando o PRCC para descrever as partes dos pássaros anotadas. Foram utilizadas aves da classe 151 como conjunto positivo e as demais classes como conjunto negativo. Foi criado um modelo para cada parte do pássaro e, utilizando-se a confiança calculada durante a validação cruzada na fase de treinamento, tivemos precisão de 100% para a testa, por exemplo. Estes resultados, apesar de não poderem ser comparados aos resultados finais obtidos anteriormente, mostram a adequação do uso da PRCC para descrever as

partes anotadas.

6.1 Sugestões para Pesquisas Futuras

Uma extensão da QEF é a utilização de regiões não retangulares. Uma etapa de segmentação pode ser utilizada para criar quantizações espaciais que se encaixam mais naturalmente às fronteiras e bordas presentes nas imagens. Estas quantizações espaciais teriam, em um primeiro nível, apenas uma região, ou seja, a imagem inteira. Uma segmentação hierárquica pode ser aplicada e dividiria o primeiro nível em duas regiões, por exemplo, seguindo bordas naturais presentes nas imagens. Estas subdivisões seriam aplicadas recursivamente até a obtenção de regiões planas. O uso de operadores morfológicos conexos parece ser uma boa opção para a construção deste tipo de quantização espacial.

O mapa de bordas bem como as imagens em níveis de cinza apresentam uma grande variedade de características que podem ser exploradas na construção de descritores que captam propriedades discriminantes de classes de objetos. Os 3 descritores apresentados neste trabalho dão uma pequena amostra deste poder. Várias outras características podem ser exploradas, como, por exemplo, as ramificações das bordas, as terminações e as junções. Terminações e junções podem facilmente ser caracterizadas por uma versão modificada do descritor PPE.

As PRCCs mostraram resultados promissores nas bases testadas. Uma modificação a ser testada é seu comportamento utilizando um descritor invariante a rotação como o SURF.

Uma extensão óbvia ao método de categorização refinada mostrada aqui é a utilização do PRCC para descrever as partes anotadas dos pássaros. Foi feito um teste neste sentido utilizando a PRCC para criar um modelo da parte e este modelo foi validado no próprio conjunto de treinamento utilizando a anotação da parte para extrair o descritor da amostra a ser testada. O resultado utilizando a anotação no teste foi excelente. Uma das partes obteve uma precisão de 100% para validação cruzada usando o método k-fold com $k = 5$. Este resultado valida o descritor, o classificador e a divisão espacial proposta neste trabalho, usados no experimento. Porém, para aplicar esta técnica em amostras de teste sem utilizar a anotação da parte, será necessário criar um algoritmo de localização da parte no teste. O problema de categorização refinada passa a ser visto como um problema de localização da parte, visto que, após a localização da parte na amostra de teste, a classificação implica um desempenho muito bom. A criação deste algoritmo de localização de partes, a partir de amostras das mesmas partes no conjunto de treinamento, é uma interessante extensão deste trabalho. Cada classe seria representada por uma série de modelos e cada modelo representa uma parte. A combinação dos resultados para a rotulação final da amostra pode dar pesos diferentes a partes diferentes para cada classe testada. Assim, espécies com características discriminantes nos olhos, por exemplo, teriam o modelo respectivo com um peso maior na combinação de classificadores.

Uma outra modificação seria a escolha automática dos pesos dos modelos na combinação final de classificadores, a partir de validações efetuadas durante a fase de treinamento.

Uma extensão, um pouco mais complexa para este trabalho, seria a utilização dos sons e vídeos dos pássaros no processo de categorização refinada. Existem disponíveis na Web catálogos¹ de pássaros com sons de seus cantos e vídeos. A utilização do som dos cantos e dos movimentos dos pássaros nos vídeos poderia trazer mais elementos a serem utilizados em uma aplicação de suas categorizações.

¹www.allaboutbirds.org

Apêndice A

Detalhes de Codificação e o Arcabouço Orientado a Objetos

Este apêndice mostra alguns detalhes da codificação em Matlab, dando ênfase aos recursos utilizados para tornar o código reutilizável em diferentes aplicações, acessando diferentes bases de dados e utilizando descritores e classificadores diversos. Descrevemos também a evolução do código escrito em Matlab para um arcabouço orientado a objetos, escrito em Java, de aplicações de reconhecimento de objetos.

A.1 Codificação em Matlab

Durante o desenvolvimento deste trabalho, tivemos uma preocupação constante de evitar, ao máximo, reescrita de código. Uma constatação que tivemos, ao longo dos anos trabalhando com Visão Computacional, é que pouco ou nenhum código é reutilizado em aplicações diferentes. Mesmo que estas aplicações sejam, a grosso modo, muito parecidas: a grande parte das aplicações de categorização de objetos segue um mesmo molde. As aplicações possuem, basicamente, os seguintes passos:

1. *Parametrização*: neste passo a parametrização é lida de algum arquivo externo ou recebida como argumento da linha de comando.
2. *Base de dados*: a base de dados é lida e uma estrutura interna armazena as categorias, arquivos e outras informações inerentes à base de dados.
3. *Geração de matrizes de descritores*: os descritores do conjunto de imagens de treinamento são gerados. Pode ser necessário construir dicionários visuais neste passo.
4. *Geração de modelos*: são criados modelos para as categorias da base de dados.
5. *Testes*: são gerados descritores para cada amostra de teste, estes descritores são aplicados aos modelos criados com o conjunto de treinamentos. Um rótulo final ou uma lista contendo a possibilidade da amostra de teste pertencer a cada uma das categorias da base de dados é gerada.
6. *Medidas de desempenho*: uma comparação entre os rótulos produzidos pela aplicação e os rótulos do gabarito da base de dados é feita e é gerada uma medida que mede a eficiência desta aplicação permitindo uma comparação com outras reportadas pela literatura.

Muitas vezes, ao se alterar a base de dados, descritores, classificadores ou a abordagem utilizada (multiclasse ou classificação binária) nos vemos reescrevendo a aplicação inteira, inclusive porções de códigos bem parecidos. Para evitar isto, utilizamos características novas da linguagem Matlab como células, introspecção, ponteiros para funções e, ainda, tentamos modularizar ao máximo as

funções definindo as responsabilidades, diminuindo o acoplamento e aumentando a coesão. Tudo isto nos permitiu, em uma mesma aplicação, utilizar diferentes bases de dados, gerar diferentes descritores e combiná-los de diferentes maneiras, utilizar diferentes classificadores, criar modelos baseados em descritores e classificadores, combinar estes modelos de diferentes maneiras para obter a rotulação final e, ainda, utilizar diferentes medidas de avaliação de desempenho. Utilizamos, ainda, abordagens multiclasse e de classificação binária em uma mesma aplicação Matlab dependendo da parametrização passada.

A listagem A.1 mostra um trecho de código em Matlab que carrega os parâmetros padrões, recebe como parâmetro uma configuração específica que modifica a parametrização padrão e mostra, ainda, a chamada a carga de informações do banco de dados de acordo com a parametrização e a configuração anteriores. Cada base de dados diferente possui uma função de carga específica que é atribuída para ao campo *init_db* da variável *params*. A listagem A.2 mostra um exemplo de configuração específica para uma execução de uma aplicação de categorização de objetos. Foram utilizados três descritores diferentes (linha 17) e apenas o primeiro descritor faz uso de um dicionário visual de 600 palavras (linha 26).

```

1 function main(varargin)
2 % Main program
3 % Author: Arnaldo Câmara Lara
4 % Date: 03/31/2011
5 % Database: Instance Monuments Search
6 % Objective: build a classifier to find an instance of a monument
7
8 % initialize parameters
9 params = init_params;
10
11 if ~isempty(varargin)
12     config = varargin{1};
13 end;
14
15 params = set_config(config, params);
16
17 % initialize database
18 if isfield(params, 'init_db') && (~isempty(params.init_db))
19     DB = params.init_db(params);
20 else
21     fprintf(1, 'DB must be setup!\n');
22     return;
23 end;
```

Listing A.1: Trecho de código mostrando o passo inicial de uma aplicação de categorização de objetos carregando parâmetros e informações da base de dados.

```

1 % run combined descriptors
2
3 clear;
4
5 config.verbose = 1;
6 config.init_db = @init_landmarks_db;
7
8 % features
9 config.phow.step = 5;
10 config.phow.sizes = [2 4 6 10];
11 config.feat.save = 1;
12 config.feat.load = 0;
13 config.feat.axis_value = 500;
14 config.feat.axis = 1;
15 config.trainset_size = 15;
16 config.testset_size = 5;
17 config.feature_extractor = {@phow_quant, @ppe3, @histogram_ab};
18 config.edge_method = @edge_canny;
19 config.edge_calc = 'rgb2gray';
```

```

20 config.ppe3.nscales = 3;
21 config.ppe3.list_strels = [];
22 config.histogram_ab.nbins = 30;
23 config.feat.quantize_location = {1, 0, 0};
24 config.feat.dim1 = {2, 0, 0};
25 config.feat.dim2 = {2, 0, 0};
26 config.build_dic = {@build_visual_dic_class, [], []};
27 config.visual_dic.max_descriptors = {600000, NaN, NaN};
28 config.visual_dic.num_codewords = {600, NaN, NaN};
29 config.visual_dic.max_search_dic = {15, 15, 15};
30 config.visual_dic.dicset_size = {15, 15, 15};
31 config.feature_descriptor_builder = {@feat2bow_quant_loc, @cell2feat, @cell2feat
    };
32 config.feature_selector = [];
33 config.segmentation_method = [];
34
35 % approach
36 config.build_model = @build_model_combined_pegasos;
37 config.kernel_method = @chi2_kernel;
38 config.homkermap.N = 1;
39 config.svm.C = 10;
40 config.svm.biasMultiplier = 1;
41 config.test = @test_combined_pegasos;
42
43 % name
44 config.feat_filename = 'feat_cd_landmarks';
45 config.model_filename = 'model_cd_pegasos_landmarks';
46 config.temp_model_filename = 'temp_model_cd_pegasos_landmarks';
47
48 main(config);

```

Listing A.2: Configuração específica para uma execução da aplicação.

A.2 Arcabouço Orientado a Objetos

Uma evolução natural do código descrito na seção anterior é implementar uma aplicação de categorização de objetos em um arcabouço orientado a objetos. Este arcabouço permitiria a utilização de modernas técnicas de Engenharia de Software [Pre09] com o intuito de atingir alto grau de produtividade, qualidade, confiabilidade, suportabilidade e manutenibilidade. Além disto, utilizando um ambiente de desenvolvimento sem custo, possibilitaria executar aplicações de Visão Computacional fora do ambiente Matlab que é uma ferramenta comercial de alto custo.

A.2.1 Escolha do Java

A codificação de aplicações de Categorização de Objetos em Matlab é natural, pois grande parte da pesquisa de ponta nesta área é feita utilizando Matlab. Os sítios de alguns dos principais autores da área, como Torralba¹, Felzenswalb², Lazebnik³ e Fei-Fei⁴, disponibilizam códigos de suas mais recentes pesquisas em Matlab. Os principais desafios da área, como VOC Pascal [EGW⁺10] e TRECVID [OAF⁺10], também disponibilizam códigos aos participantes em Matlab.

Assim, um dos requisitos do arcabouço orientado a objetos é a capacidade de executar funções escritas em Matlab. Algoritmos recém disponibilizados nos sítios dos autores em Matlab poderiam ser rapidamente utilizados no arcabouço para incorporar as novas funcionalidades ou para serem testados juntamente com outras funções similares. Se for necessária, uma reimplementação da função utilizando uma linguagem de programação mais eficiente, como o C, por exemplo, poderia ser feita

¹<http://web.mit.edu/torralba/www/>

²<http://www.cs.brown.edu/~pff/>

³<http://www.cs.illinois.edu/homes/slazebni/>

⁴<http://vision.stanford.edu/~feifeili/>

futuramente e a interface da chamada da função original seria mantida e, assim, os códigos de chamada a esta função continuariam preservados.

A linguagem de programação Java foi escolhida para implementar o arcabouço orientado a objetos pelas seguintes razões:

- A linguagem de programação Java⁵, desenvolvida por James Gosling na *Sun Microsystems*, durante a década de 90, que foi posteriormente adquirida pela Oracle⁶, é uma das linguagens de programação mais utilizadas do mundo [Tio12].
- Java é uma linguagem de programação orientada a objetos que roda em uma máquina virtual com implementações em diferentes arquiteturas. O código binário roda em qualquer máquina virtual sem a necessidade de ser recompilado.
- A linguagem de programação e o ambiente de execução Java estão disponíveis sem custo. Existe, ainda, uma infinidade de bibliotecas, ambientes de desenvolvimento, ferramentas e recursos disponíveis, sem qualquer custo, facilitando o processo de desenvolvimento, manutenção e instalação de aplicações.
- Existem inúmeros meios de comunicação entre o Matlab e o Java e desde a versão 6 (R12), o Matlab possui uma máquina virtual Java incorporada.
- Existe uma API nativa, a JNI, que permite ao Java executar código escrito nativamente, na linguagem C.
- A OpenCV [BK08] é uma biblioteca de Visão Computacional com uma extensa lista de funções escritas em C e C++ com o intuito de serem utilizadas em aplicações de VC em tempo-real. Além do C e C++, existe uma interface para acesso às funções em Python e um projeto de terceiros que acessa a biblioteca em Java, o JavaCV [Aud12].

A.2.2 Comunicação Java-Matlab

Executar uma aplicação Java do Matlab é trivial. Existe uma máquina virtual Java, a JVM, incluída no ambiente de programação Matlab e qualquer aplicação Java pode ser executada neste ambiente. Executar aplicações escritas em Matlab da plataforma Java é um pouco mais complexo. Existem algumas opções:

- O *Matlab Builder JA* é uma ferramenta oficial da *Mathworks*⁷, empresa proprietária do Matlab, que permite criar classes Java a partir de programas escritos em Matlab. As classes Java geradas são empacotadas em arquivos jar (bibliotecas dinâmicas da plataforma Java) e podem ser distribuídos sem custo e sem a necessidade de qualquer instalação do Matlab. A principal desvantagem desta opção é o custo da ferramenta. Esta ferramenta não faz parte do pacote básico do Matlab e tem um alto custo.
- *Java-to-Matlab Interface* é uma biblioteca não documentada disponibilizada pela própria Mathworks que permite que aplicações Java chamem código Matlab. Esta biblioteca está no arquivo jmi.jar e é disponibilizada em todas as versões do Matlab.. A grande desvantagem de utilizá-la é, obviamente, a falta de documentação e de suporte oficiais. Alguns sítios na Web disponibilizam tutoriais sobre sua utilização, como em [Alt12].
- *JAMAL*, do inglês *Java Matlab Linking*, é uma biblioteca que torna possível a chamada de funções Matlab de classes Java passando e retornando tipos primitivos do Java e seus vetores. Esta biblioteca é baseada na tecnologia Java RMI (*Remote Method Invocation*). Mais detalhes sobre sua utilização podem ser encontrados em [Kha12].

⁵www.java.com

⁶www.oracle.com

⁷www.mathworks.com

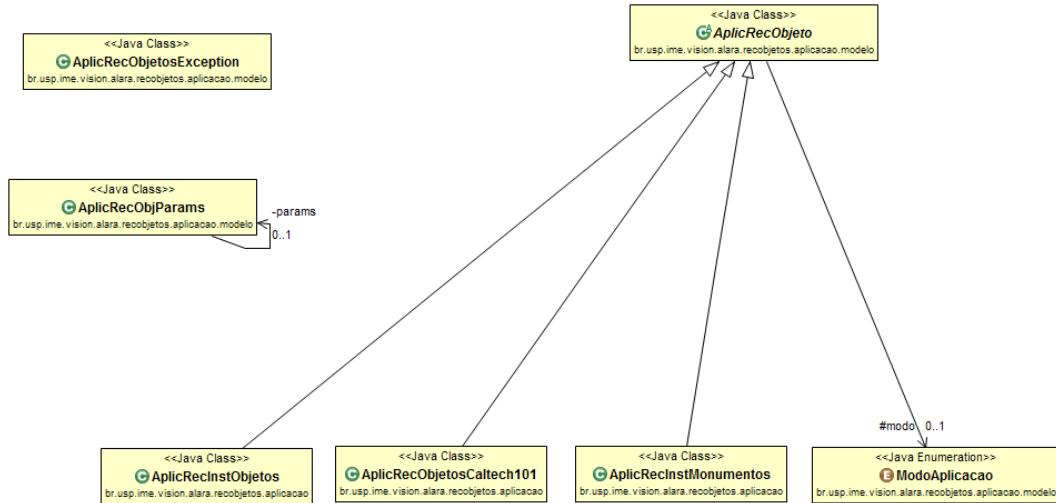


Figura A.1: Diagrama de classes mostrando a classe abstrata AplicRecObjetos e suas classes derivadas.

- *Matlabcontrol* é uma biblioteca Java que permite controlar uma ou mais sessões Matlab de uma aplicação Java de fora do ambiente Matlab. As aplicações Matlab são executadas sem intervenção manual e é construída utilizando a interface RMI do Java e a biblioteca JMI disponibilizada pela Mathworks. O projeto está hospedado em [Whi12].

No sitio <http://code.google.com/p/matlabcontrol/wiki/ApproachesToControl> estão descritas algumas outras formas de uma classe Java controlar uma sessão Matlab ou executar um programa escrito em Matlab. Em nossa implementação utilizamos a biblioteca *Matlabcontrol* desenvolvida por K. Whitehouse.

A.2.3 Implementação do Arcabouço Orientado a Objetos

O arcabouço orientado a objetos foi implementado utilizando linguagem Java. A comunicação entre a aplicação Java e o Matlab, onde várias funções são executadas, é implementada utilizando a biblioteca Matlabcontrol, descrita na seção anterior. Utilizamos diagramas de classes do UML para descrever o arcabouço. A UML é um conjunto de notações gráficas que auxiliam na descrição de sistemas de software, especialmente aqueles construídos utilizando o paradigma de orientação a objetos [Fow04].

A Figura A.1 apresenta um diagrama de classes mostrando a classe abstrata *AplicRecObjetos*. Quaisquer aplicações do arcabouço precisam derivar desta classe. São mostradas três classes derivadas da classe abstrata *AplicRecObjetos*. Cada uma destas classes é uma aplicação concreta utilizando uma base de dados instanciada, uma lista de descritores e seus respectivos parâmetros e uma lista de classificadores. O método inicializa é reescrito e objetos da base de dados, descritores e classificadores, são instanciados neste método. A classe *ModoAplicacao* controla se a aplicação está em treinamento ou teste. A classe *AplicRecObjParams* armazena parâmetros relativos a aplicação.

Figura A.2 mostra o diagrama de classes de bases de dados. A classe *ServicoBaseDados* possui o código necessário para carregar informações das bases de dados. A classe *BaseDados* é abstrata e todas as bases de dados estendem esta classe. A classe *Classe* modela as diferentes categorias da base de dados. As categorias podem estar em diferentes níveis.

A Figura A.3 mostra o diagrama das classes de imagens. A classe *ServicoConjImagen* gera os conjuntos de treinamento e teste. Cada conjunto é uma instância da classe *ConjImagen*. A classe *TipoConjImagen* define se o conjunto de imagem é teste ou treinamento. O conjunto de imagem possui uma lista de instâncias de *RegConjImagen* e cada instância de *RegConjImagen* possui uma instância de imagem, classe e informações sobre a rotulação daquela imagem. Cada objeto da classe *Imagen* pode ter uma máscara ou uma caixa envoltória para ser utilizada na etapa de treinamento.

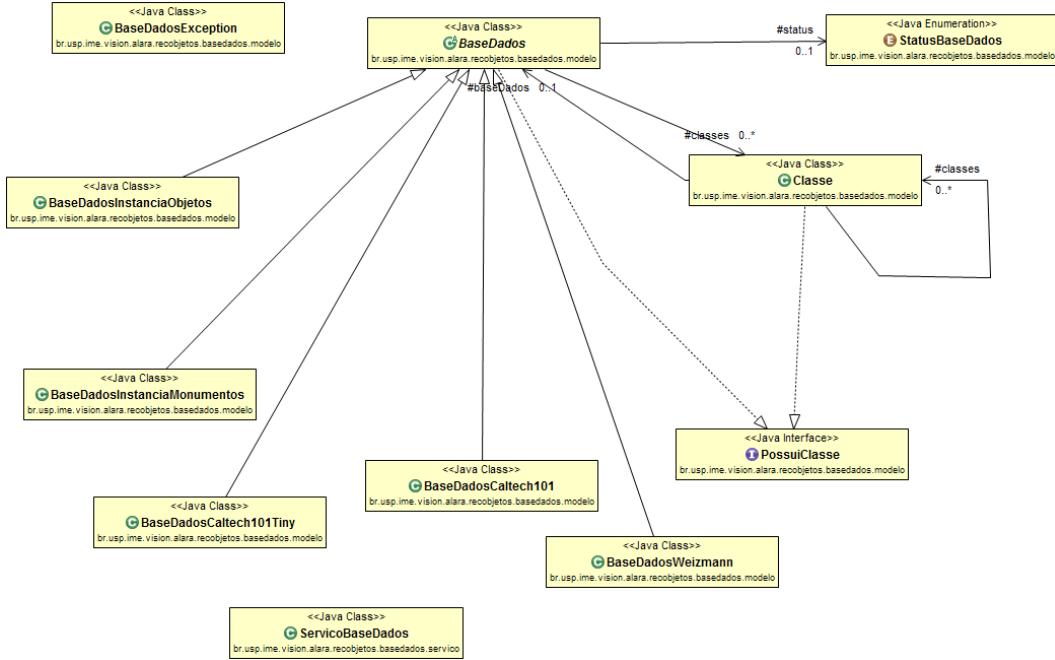


Figura A.2: Diagrama de classes mostrando as classes de bases de dados.

A Figura A.4 mostra o diagrama das classes relativas aos descritores. A classe abstrata *Descriptor* é a classe ancestral de todas as classes de descritores, como a LBP, PHOW ou SURF. Um descendente da classe *Descriptor* recebe no construtor um objeto de alguma classe descendente da classe abstrata *DescriptorParam* com as parametrizações específicas daquele descritor. O método da classe *Descriptor* extraiDescritores cria um objeto da classe *FachadaMatlab* e repassa os parâmetros do descritor e a lista de imagens com suas categorias respectivas. Um objeto descritor pode possuir um objeto da classe *DicionarioVisual* e ainda um objeto da classe *QuantizacaoEspacial*. Existem duas classes derivadas da *QuantizacaoEspacial*, a *PiramideEspacial* e a *PiramideCircular*. Objetos da classe *PiramideCircular* implementam as Pirâmides de Regiões Circulares Concêntricas descritas na Seção 3.3.

O diagrama de classes dos classificadores são mostrados na Figura A.5. A classe abstrata *Classificador* possui um descendente, o *ClassificadorGenerico*. O *ClassificadorGenerico* é a classe pai da classe *SVM* que possui como descendente a classe *SVMBinario*. Finalmente, a classe *SVM_Pegasos* descendente de *SVMBinario* é a primeira classe concreta nesta hierarquia de classes. A classe *SVM* possui um atributo da classe *Kernel*. Implementamos a classe concreta *KernelChi2* descendente de *Kernel*. A classe *ServicoModelo* possui métodos que permitem o treinamento, teste e cálculo do resultado final da classificação. O classificador aplicado aos descritores do conjunto de imagens de treinamento gera um objeto da classe *Modelo*. Uma coleção de modelos é aplicada a amostras do conjunto de testes e, utilizando objetos da classe *CombinacaoClassificadores*, os resultados são combinados e a rotulação final da amostra é encontrada.

A Figura A.6 mostra o diagrama de classes utilitárias. A principal classe deste diagrama é a classe *FachadaMatlab* que implementa a interface *Fachada* e recebe como parâmetro um objeto da classe *PropriedadesMatlab* que armazena vários parâmetros necessários para a comunicação com o Matlab. A classe *FachadaMatlab* implementa o padrão de projeto *Fachada*. Padrões de projeto são classes e objetos que se comunicam e são adaptados para resolver um problema geral em um contexto específico [GHJV94]. O padrão de projeto *Fachada* tem como objetivo fornecer uma interface unificada para um conjunto de interfaces em um subsistema. Em nosso contexto, toda comunicação com o Matlab é feita através da classe *FachadaMatlab*, portanto é a única que conhece os detalhes da comunicação. Uma das vantagens da utilização deste padrão de projeto é que, caso exista a necessidade de trocar a tecnologia de comunicação com o Matlab, apenas a classe de fachada é alterada. Todas as outras classes que disparam funções no Matlab continuam

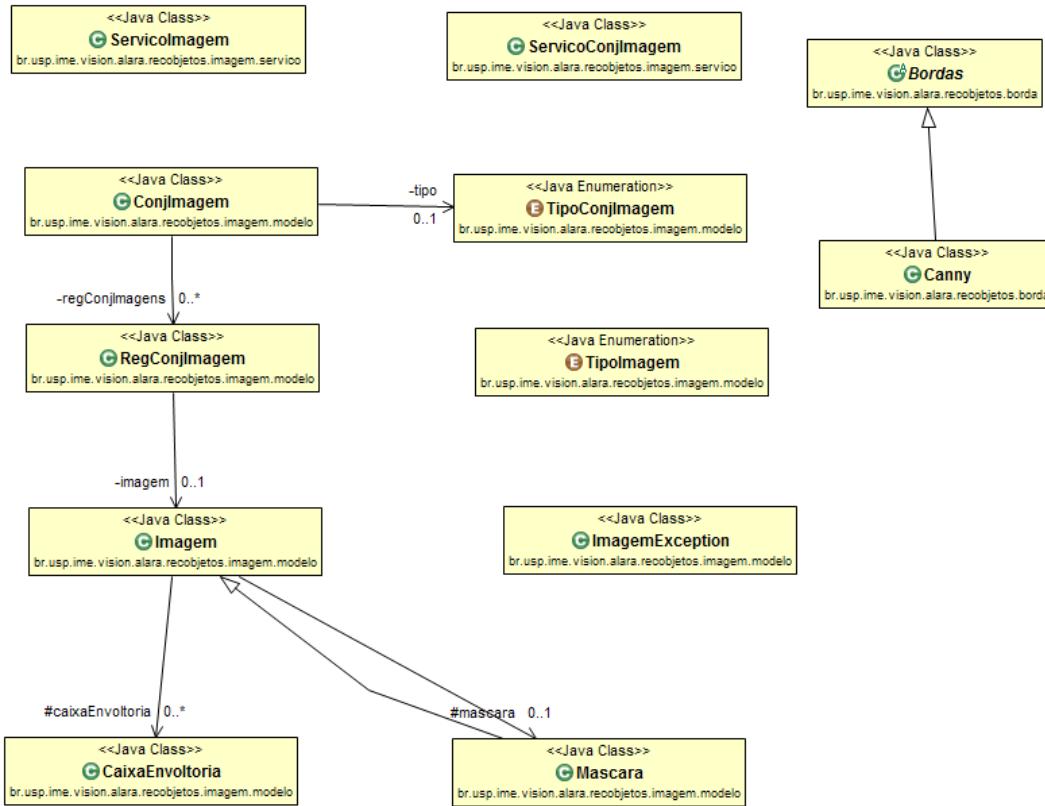


Figura A.3: Diagrama de classes mostrando as classes de relacionadas com as imagens.

preservadas, chamando pela interface da chamada. Apenas os detalhes internos da classe de fachada seriam alterados.

A Figura A.7 mostra o diagrama das classes de resultados. A classe *ResultadoModeloImagen* armazena os resultados individuais das imagens por modelos. A classe *CalculaResultados* calcula as medidas de desempenho e as classes *ResultadoModelo* e *ResultadoModeloMultiClasse* calculam as medidas para cada modelo utilizado.

A.2.4 Direções Futuras

Pretendemos disponibilizar publicamente o arcabouço de reconhecimento de objetos em breve. Porém alguns ajustes ainda são necessários. O principal ajuste é em relação a configuração da aplicação. É necessária uma extensa parametrização para que uma aplicação de reconhecimento de objetos rode. É preciso facilitar a parametrização do arcabouço para que seu uso se torne viável. Uma nova funcionalidade bastante interessante seria o acesso a funções da biblioteca OpenCV através do projeto JavaCV.

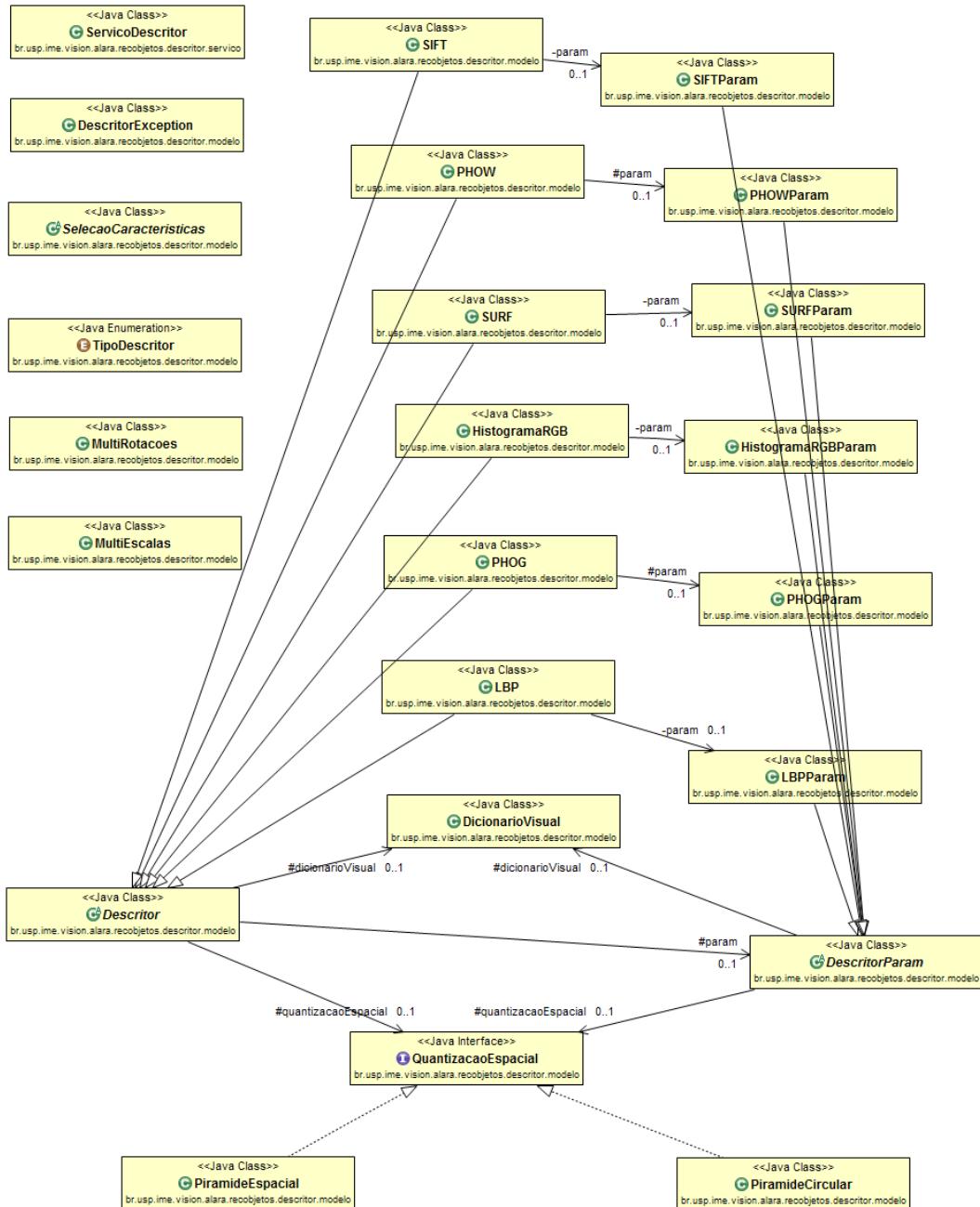


Figura A.4: Diagrama de classes mostrando as classes de descritores.

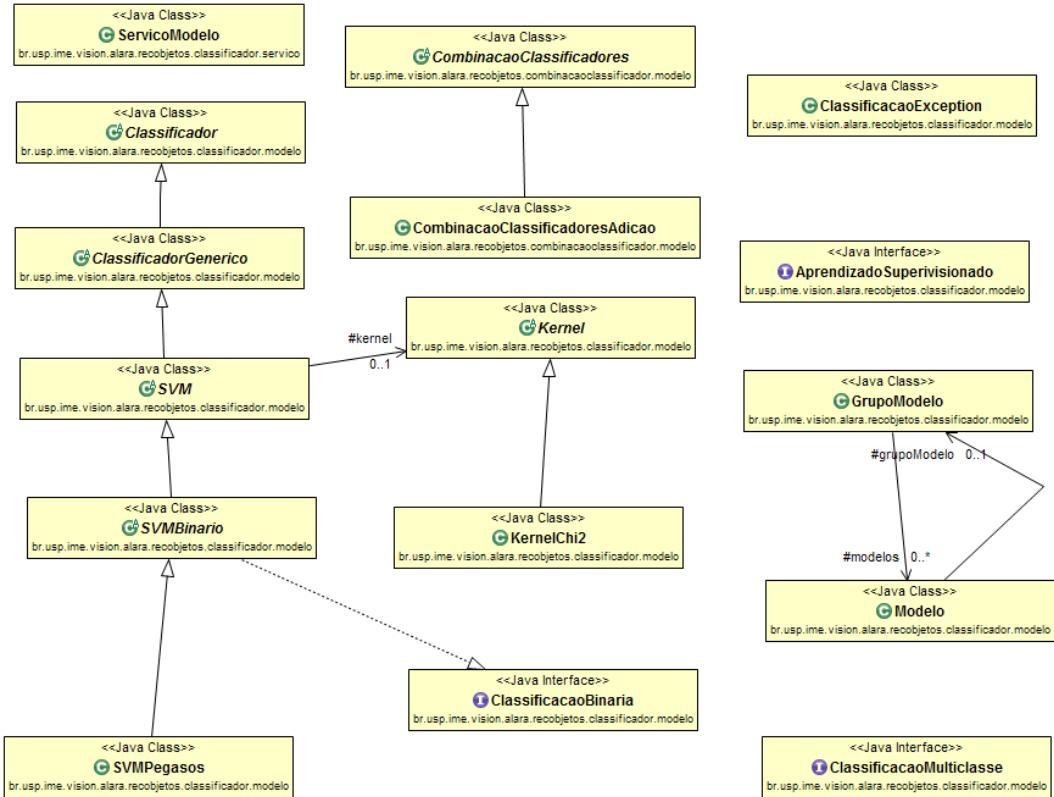


Figura A.5: Diagrama de classes mostrando as classes de classificadores.

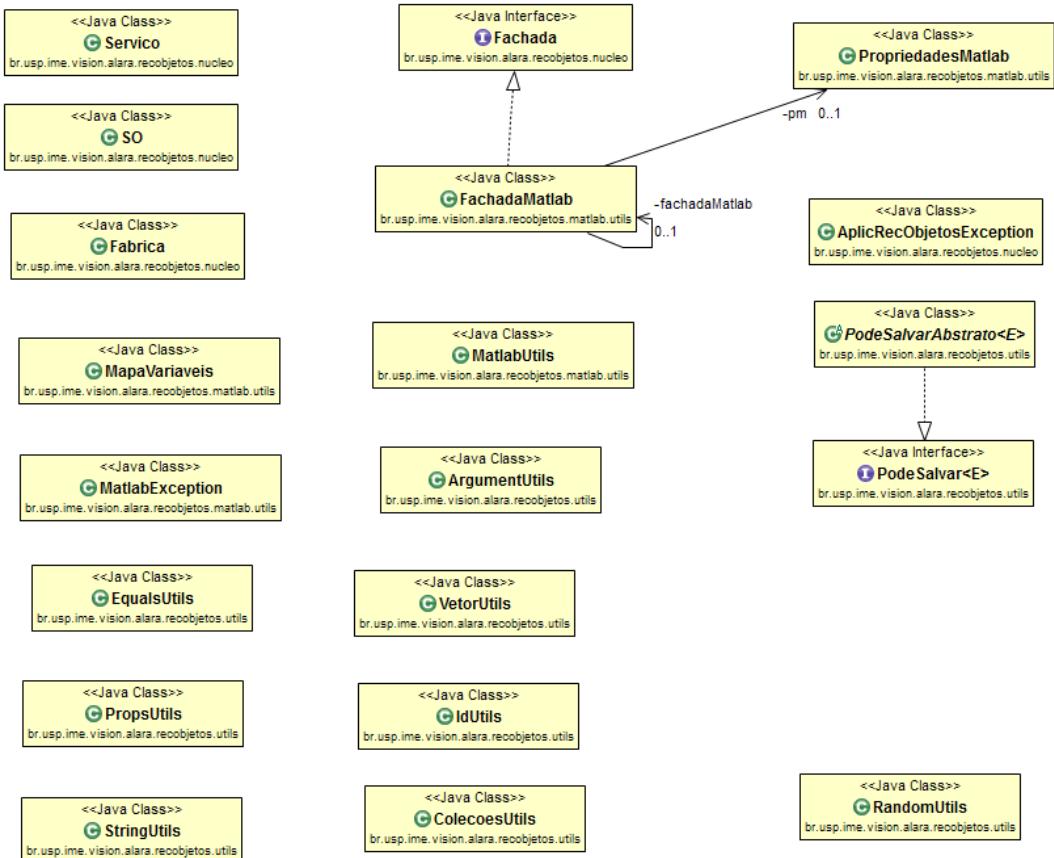


Figura A.6: Diagrama de classes mostrando as classes de utilitários.

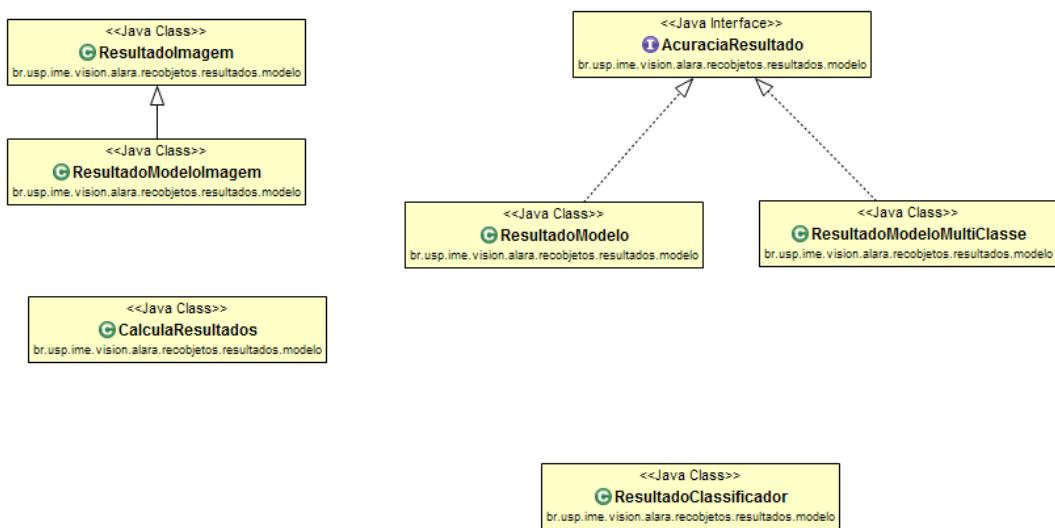


Figura A.7: Diagrama de classes mostrando as classes de resultados.

Apêndice B

Participação no TRECVID 2011

O TRECVID [OAF⁺10] é uma conferência e uma competição anual com o objetivo de promover o progresso na área de análise e avaliação de vídeos. Dentro da competição existem 6 tarefas e cada time participante escolhe uma ou mais tarefas para escrever uma solução e submeter os seus resultados. As tarefas nas edições de 2010 e 2011 foram: indexação semântica, procura por item conhecido, detecção de cópia baseada em conteúdo, detecção de eventos de segurança, detecção de eventos de multimídia e busca por instâncias. Em sua edição de 2010, foi proposta uma tarefa piloto de busca por instâncias em uma coleção de vídeos. Os resultados foram bem pobres e o problema ganhou a atenção da comunidade de VC. A tarefa consiste em receber uma ou mais imagens de uma instância e procurar as 1000 cenas mais prováveis de conter aquela instância, procurando em uma coleção de dezenas de milhares de vídeos. Apesar do tempo de busca e a localização exata da instância dentro do quadro serem fatores importantes para a avaliação das técnicas, estes fatores ainda não foram levados em consideração na avaliação dos resultados. Cada instância pode ser do tipo objeto, lugar, pessoa ou personagem. Técnicas diferentes podem ser utilizadas para cada tipo de instância. A nossa participação foi publicada em [LH11a].

B.1 Bases de Dados

Esta seção descreve as bases de dados utilizadas na edição 2011 da tarefa de busca por instâncias do TRECVID.

B.1.1 Sound & Vision

Esta base de dados do desafio TRECVID foi disponibilizada em 2009 e utilizada na tarefa busca por instâncias de 2010 e 2011. Em 2010, foi a base de teste e, em 2011, a base de treinamento. São 400 vídeos produzidos pela TV holandesa, disponibilizados em formato MPEG-1 e totalizando 114,8 GB de dados. Para o desafio, foram divididos em 61091 cenas. Os vídeos são protegidos por leis de direitos autorais e não estão disponíveis publicamente.

B.1.2 BBC Rushes

Base de dados disponibilizada para a tarefa busca por instância de 2011 do TRECVID foi a base de teste para a tarefa da edição de 2011. São vídeos de vários programas não editados da BBC, canal de TV do Reino Unido. A base de dados é dividida em cenas aproximadamente do mesmo tamanho, cerca de 30 segundos, perfazendo um total de 20982 vídeos e 56,8 GB de dados. Os vídeos são protegidos por leis de direitos autorais e não estão disponíveis publicamente, apenas para os participantes do desafio TRECVID. A Figura B.1 mostra alguns quadros das bases de dados utilizadas na tarefa busca por instâncias do desafio TRECVID.

(a) Alguns quadros da base *Sound & Vision*.(b) Exemplos de quadros da base *BBC Rushes*.**Figura B.1:** Exemplos de quadros das duas bases de dados utilizadas na tarefa busca por instâncias da edição 2011 do desafio TRECVID.

B.2 Revisão Bibliográfica

Esta seção descreve os métodos que obtiveram os melhores resultados na edição 2010 da tarefa.

O trabalho [LPW⁺10] submetido na edição 2010 do desafio TRECVID utiliza duas abordagens: uma para pessoas e personagens e outra para objetos e locais. Na abordagem para pessoas e personagens são extraídas características locais e globais dos quadros das consultas e das bases de dados. Um modelo é criado para cada consulta utilizando os quadros de exemplos. Para as características locais, são localizados 13 pontos marcantes nas faces e calculadas as intensidades dos pixels a um raio de 15 pixels de cada um dos pontos marcantes. Este descritor possui 1937 dimensões. O vetor de características globais desta abordagem utiliza o descritor LBP com regiões 5×5 e 30 intervalos de intensidades dos pixels totalizando 750 dimensões. Na abordagem para objetos e locais, são extraídos descritores SIFT e histograma RGB. São construídos dois dicionários visuais utilizando o descritor SIFT com 2048 e 16384 palavras. Uma medida de similaridade entre o modelo criado a partir das consultas e um modelo criado para cada cena é calculada utilizando distâncias L1 e L2 para identificar os quadros da base de dados com maior probabilidade de ter a instância procurada. Uma combinação utilizando pesos diferentes entre as características locais e globais é utilizada. Este trabalho obteve o melhor resultado em duas consultas de pessoas.

O trabalho descrito em [GCL⁺10] obteve o melhor resultado em duas consultas de pessoas. A técnica foca a busca por pessoas e personagens através do reconhecimento de faces. Foram utilizados vários descritores, como o histograma HSV, ondaleta de Gabor, HOG, LBP e o correlograma HSV.

Dicionários visuais de 4096 e 512 palavras, que usam o descritor SURF e implementados através da OpenCV [BK08], no trabalho de [SDK⁺10], obtiveram o melhor resultado em uma consulta de pessoa.

O trabalho proposto em [BMH⁺10] utiliza 5 descritores diferentes: ondaletas de Gabor para reconhecimento de faces; descritores SIFT são utilizados na abordagem do saco de palavras para a construção de dois dicionários visuais com 100 e 1000 palavras; descritores SIFT em pontos de interesse, HOG e descritor covariança de regiões que utiliza uma distância baseada em autovalores. Quatro abordagens diferentes de fusão dos resultados de cada descritor foram testadas e chegou-se à conclusão de que a utilização de descritores únicos, sem fusão, obteve resultados melhores que os resultados que utilizaram fusão de vários descritores. Esta técnica obteve o melhor resultado de

duas consultas: uma de pessoa e uma de objeto.

B.3 Submissão

Esta seção descreve a nossa participação no TRECVID edição 2011 na tarefa de busca por instância. De um total de 48 times inscritos na tarefa, apenas 13 times submeteram resultados a este desafio na edição 2011. Fomos o único time da América do Sul a submeter resultados. Na edição 2011 da busca por instância, foram 25 consultas no total, sendo 17 consultas de objetos, 6 consultas de pessoas e 2 consultas de locais. Cada consulta pode ter até 6 exemplos de quadros com a respectiva máscara contendo a instância a ser buscada na base de dados. Para cada consulta, a resposta deve ser as 1000 cenas mais prováveis de ter a instância buscada presente. Submetemos apenas uma execução.

Construímos um dicionário visual de 300 palavras utilizando 600000 descritores PHOW, aleatoriamente escolhidos dentre todos os descritores extraídos da base de dados. O descritor PHOW utilizou um grade regular de pontos com 5 pixels de intervalo entre os pontos. Os descritores do conjunto positivo para a construção do modelo da consulta foram extraídos apenas nas regiões delimitadas pelas máscaras dos quadros exemplos das consultas. Para cada cena da base de dados, apenas o quadro central, representando a cena inteira, foi utilizado. Os quadros representando as cenas são descritos como histograma normalizado de frequência do dicionário visual utilizado. Para cada modelo da consulta, uma distância qui-quadrada para cada quadro é calculado. As 1000 cenas com as menores distâncias são retornadas para cada consulta, em ordem crescente de distância.

Nossos melhores resultados foram nas duas consultas de locais. Obtivemos resultados acima da média nestas duas consultas. Em ambas as consultas, a textura do quadro é a característica mais importante e a abordagem utilizada (saco de palavras juntamente com o descritor PHOW) é muito adequada para descrever textura. Outro fator que melhorou nossos resultados, nestas duas consultas, é o fator de ambas as máscaras ocuparem quase que a imagem inteira e, assim, o modelo construído se tornou mais significativo. O método utilizado não funcionou quando a instância de interesse ocupava uma área pequena dos quadros de exemplo.

B.4 Conclusão

Utilizamos o mesmo método para todos os tipos de consultas (pessoa, personagem, objeto ou local). O nosso método teve um resultado bom apenas para consultas do tipo local: consulta número 9024 e consulta número 9029. Para consulta 9024, a medida MAP (média das precisões médias) de todos os times foi 0,241 e o nosso resultado obteve precisão média (AP) de 0,268. Para a consulta 9029, a medida MAP para todos os times submetidos foi 0,184 e o nosso resultado obteve AP de 0,276. Nestas duas consultas de locais, a textura da cena é uma característica marcante e a técnica utilizada que usa o descritor PHOW juntamente com o saco de palavras é adequada para caracterizar textura.

A abordagem do saco de palavras mais o descritor PHOW são listados como tendo um bom desempenho para aplicações de reconhecimento de objetos [WYY⁺¹⁰]. Porém o descritor PHOW juntamente com a abordagem saco de palavras não se mostraram suficientes para obter bons resultados na tarefa de busca de instâncias do desafio TRECVID. Para consultas de pessoas, resultados de outros times mostram que é necessário utilizar métodos de detecção e reconhecimento de faces. Para consultas de objetos, é necessário utilizar uma combinação de descritores para criar modelos mais efetivos.

Devido ao grande número de vídeos envolvidos durante as fases de treinamento e validação (mais de 61000 vídeos) e teste (quase 21000 vídeos), várias simplificações e adaptações no nosso método se fizeram necessárias. Utilizamos um hardware que não era o apropriado para lidar com o volume de dados que a tarefa exigia. Tínhamos restrição de memória, de disco e de processamento. A restrição de memória limitava o tamanho dos dicionários visuais que poderíamos criar. A limitação de disco restringia o volume de estruturas temporárias criadas que desafogaria a memória. Porém a principal

limitação foi de poder de processamento e tempo. Dependendo da configuração utilizada, o tempo de treinamento e validação de parâmetros poderia ser superior a uma semana.

Para contornar as limitações de processamento e tempo, restringimos o número de quadros analisados em cada cena para apenas um. Esta restrição é uma representação muito grosseira da cena. Cada cena tem um tempo médio de 30 segundos e a codificação utilizada usa 25 quadros por segundo, o que totaliza, em média, 750 quadros por cena que foi representada por apenas um quadro. Tivemos que utilizar um único descritor, o PHOW, e um dicionário visual de tamanho modesto. Isto também afetou o desempenho do método.

Com o intuito de melhorar o desempenho na tarefa de busca por instâncias, vários métodos utilizados na categorização refinada (Capítulo 4) podem ser utilizados:

- Utilizar mais quadros, de preferência todos, para representar a cena.
- Na categorização refinada foi gerado um conjunto expandido das imagens de treinamento através de transformações aplicadas às imagens originais. Para criar modelos das consultas mais efetivos, podemos aplicar transformações similares aos quadros exemplos que são, originalmente, muito poucos.
- A utilização de diversos descritores, incluindo os propostos neste trabalho, para modelar as consultas poderia melhorar o desempenho final da aplicação. A combinação de diversos descritores se mostrou útil na criação de modelos dos pássaros na aplicação de categorização refinada.
- Criação de modelos de vários níveis, explorando alguma hierarquia natural do problema, é uma tática que também se mostrou vitoriosa no problema de categorização e pode auxiliar na criação dos modelos das consultas.
- Uso de vários dicionários locais e globais com número diferenciados de palavras para serem utilizados na criação de modelos das consultas.
- A utilização de modelos específicos por tipos de consultas mostrou ser uma opção acertada, adotada por quase todos os times participantes.
- Por causa da baixa resolução dos vídeos, diminuir o intervalo de amostragem da grade regular do descritor PHOW.

Referências Bibliográficas

- [Aka73] H. Akaike. Information theory and an extension of the maximum likelihood principle. Em *International Symposium on Information Theory, 2 nd, Tsahkadsor, Armenian SSR*, páginas 267–281, 1973. [27](#)
- [Alt12] Yair Altman. Undocumented matlab: Jmi-java-to-matlab interface, Novembro 2012. <http://undocumentedmatlab.com/blog/jmi-java-to-matlab-interface/>. [90](#)
- [Aud12] Samuel Audet. Javacv: Java interface to opencv and more, Novembro 2012. <http://code.google.com/p/javacv/>. [90](#)
- [BB93] G.J.F. Banon e J. Barrera. Decomposition of mappings between complete lattices by mathematical morphology, part i. general lattices. *Signal Processing*, 30(3):299–327, 1993. [31](#)
- [BDFB⁺01] Y. Balagurunathan, E.R. Dougherty, S. Frančišković-Bilinski, H. Bilinski e N. Vdović. Morphological granulometric analysis of sediment images. *Image Analysis and Stereology*, 20(2):87–99, 2001. [30](#), [34](#)
- [BDS00] S. Batman, E.R. Dougherty e F. Sand. Heterogeneous morphological granulometries. *Pattern Recognition*, 33(6):1047–1057, 2000. [33](#)
- [BK08] G. Bradski e A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008. [90](#), [98](#)
- [BMH⁺10] W. Bailer, R. Morzinger, S. Holzl, F. Lee, H. Stiegler e R. Sorschag. Joanneum research and vienna university of technology at trecvid 2010. Em *Proceedings of TRECVID 2010 - National Institute of Standards and Technology*, Gaithersburg, EUA, novembro 2010. National Institute of Standards and Technology. [98](#)
- [BMM07] Anna Bosch, Xavier Muñoz e Robert Martí. Which is the best way to organize/classify images by content? *Image and vision computing*, 25(6):778–791, 2007. [17](#)
- [BTVG06] H. Bay, T. Tuytelaars e L. Van Gool. Surf: Speeded up robust features. *Lecture Notes in Computer Science*, páginas 404–417, 2006. [14](#), [23](#)
- [BWS⁺10] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona e S. Belongie. Visual recognition with humans in the loop. *Computer Vision-ECCV 2010*, páginas 438–451, 2010. [18](#), [21](#), [80](#)
- [BZM07a] A. Bosch, A. Zisserman e X. Munoz. Image classification using random forests and ferns. Em *11th International Conference on Computer Vision*, páginas 1–8, Rio de Janeiro, Brasil, Outubro 2007. [23](#)
- [BZM07b] A. Bosch, A. Zisserman e X. Munoz. Representing shape with a spatial pyramid kernel. Em *Proceedings of the 6th ACM international conference on Image and video retrieval*, páginas 401–408, Amsterdã, Holanda, Julho 2007. [5](#), [24](#)

- [Can86] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(6):679–698, 1986. [42](#), [44](#)
- [CD01] G. Cula e J. Dana. Compact representation of bidirectional texture functions. Em *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 1041–1047, Kauai, EUA, Dezembro 2001. IEEE Computer Society. [12](#)
- [CDF⁺04] G. Csurka, C. Dance, L. Fan, J. Willamowski e C. Bray. Visual categorization with bags of keypoints. Em *ECCV International Workshop on Statistical Learning in Computer Vision*, volume 1, página 22, Praga, República Tcheca, 2004. [2](#), [10](#)
- [CV95] C. Cortes e V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273, 1995. [25](#)
- [DB95] T. Dietterich e G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995. [27](#)
- [DBLFF10] J. Deng, A. Berg, K. Li e L. Fei-Fei. What does classifying more than 10,000 image categories tell us? *Computer Vision-ECCV 2010*, páginas 71–84, 2010. [2](#), [21](#)
- [DDS⁺09] J. Deng, W. Dong, R. Socher, L. Li, K. Li e L. Fei-Fei. Imagenet: A large-scale hierarchical image database. Em *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, páginas 248–255, Miami, EUA, Agosto 2009. [21](#), [28](#)
- [DHS01] R. Duda, P. Hart e D. Stork. *Pattern Classification*. John Wiley and Sons, 2001. [22](#)
- [DJP⁺07] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax e S. Verzakov. Prtools 4: A matlab toolbox for pattern recognition. [urllhttp://www.prtools.org/files/PRTTools4.1.pdf](http://www.prtools.org/files/PRTTools4.1.pdf), 2007. Online; acessado em 19 de julho de 2012. [55](#)
- [DKP89] E.R. Dougherty, E.J. Kraus e J.B. Pelz. Image segmentation by local morphological granulometries. Em *Geoscience and Remote Sensing Symposium, 1989. IGARSS'89. 12th Canadian Symposium on Remote Sensing., 1989 International*, volume 3, páginas 1220–1223. IEEE, 1989. [34](#)
- [DL03] E. R. Dougherty e R. A. Lotufo. *Hands-on Morphological Image Processing*. SPIE Press, 2003. [30](#), [31](#), [33](#), [34](#)
- [DT05] N. Dalal e B. Triggs. Histograms of oriented gradients for human detection. Em *International Conference on Computer Vision and Pattern Recognition (CVPR '05)*, páginas 886–893, San Diego, EUA, Junho 2005. [23](#)
- [EGW⁺10] M. Everingham, L. Gool, C. Williams, J. Winn e A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. [2](#), [3](#), [21](#), [28](#), [30](#), [53](#), [89](#)
- [Eva09] C. Evans. Notes on the opensurf library. *University of Bristol, Tech. Rep. CSTR-09-001, January*, 2009. [55](#), [59](#)
- [FE73] M. Fischler e R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22:67–92, 1973. [xv](#), [10](#), [13](#)
- [Fel98] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, EUA, 1998. [28](#)
- [FFFP06] L. Fei-Fei, R. Fergus e P. Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006. [28](#)

- [FFP05] Li Fei-Fei e Pietro Perona. A bayesian hierarchical model for learning natural scene categories. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, páginas 524–531. IEEE, 2005. [14](#)
- [FGM10] P.F. Felzenszwalb, R.B. Girshick e D. McAllester. Cascade object detection with deformable part models. Em *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, páginas 2241–2248. IEEE, 2010. [2](#)
- [FMR08] P. Felzenszwalb, D. Mcallester e D. Ramanan. A discriminatively trained, multiscale, deformable part model. Em *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, páginas 1–8, Anchorage, EUA, 2008. [74](#)
- [Fow04] M. Fowler. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004. [91](#)
- [FOZ⁺11] R. Farrell, O. Oza, N. Zhang, V.I. Morariu, T. Darrell e L.S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 161–168. IEEE, 2011. [3](#), [18](#), [21](#), [74](#), [80](#)
- [GCL⁺10] X. Guo, Y. Chen, W. Liu, Y. Mao, H. Zhang, K. Zhou, L. Wang, Y. Hua, Z. Zhao, Y. Zhao e A. Cai. Bupt-mcprl at trecvid 2010. Em *Proceedings of TRECVID 2010 - National Institute of Standards and Technology*. National Institute of Standards and Technology, novembro 2010. [98](#)
- [GHJV94] E. Gamma, R. Helm, R. Johnson e J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994. [92](#)
- [GHP07] G. Griffin, A. Holub e P. Perona. Caltech-256 object category dataset. Relatório Técnico CaltechAUTHORS:CNS-TR-2007-001, California Institute of Technology, Abril 2007. [28](#)
- [GW02] R. C. Gonzalez e R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, segunda edição, 2002. [9](#), [23](#), [24](#), [62](#)
- [HCI⁺12] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua e V. Lepetit. Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888, 2012. [22](#)
- [HE02] G. Hamerly e C. Elkan. Alternatives to the k-means algorithm that find better clusterings. Em *Conference on Information and Knowledge Management: Proceedings of the eleventh international conference on Information and knowledge management*, páginas 600–607, 2002. [51](#)
- [Hei95] H. Heijmans. Mathematical morphology: Basic principles. Em *Proceedings*, 1995. [9](#)
- [HHC⁺11] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab e V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 858–865. IEEE, 2011. [22](#)
- [Hof99] Thomas Hofmann. Probabilistic latent semantic analysis. Em *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, páginas 289–296. Morgan Kaufmann Publishers Inc., 1999. [16](#)
- [Joa98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, páginas 137–142, 1998. [12](#)

- [KGM11] M. Khatun, A. Gray e S. Marshall. Morphological granulometry for predicting evolution time of evolving grey-scale texture images. Em *19th European Signal Processing Conference-EUSIPCO 2011*, volume 19, páginas 759–763, 2011. [34](#)
- [Kha12] Maksim Khadkevich. Jamal, Novembro 2012. <http://jamal.khadkevich.org/documentation.html>. [90](#)
- [Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. Em *International joint Conference on artificial intelligence*, volume 14, páginas 1137–1145. Lawrence Erlbaum Associates Ltd, 1995. [25](#)
- [Lef07] S. Lefevre. Extending morphological signatures for visual pattern recognition. Em *IAPR International Workshop on Pattern Recognition in Information Systems.*, Junho 2007. [38](#)
- [LF06] V. Lepetit e P. Fua. Keypoint recognition using randomized trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1465–1479, 2006. [2](#), [49](#)
- [LH11a] A. Lara e R. Hirata Jr. Cauvis-ime-ust at trecvid 2011: instance search. Em *Proc. of NIST TRECVID Workshop*, 2011. [6](#), [97](#)
- [LH11b] A. Lara e R. Hirata Jr. Combining features to a class-specific model in an instance detection framework. Em Thomas Lewiner e Ricardo Torres, editors, *Sibgrapi 2011 (24 Conference on Graphics, Patterns and Images)*, páginas 165–172, Maceió, AL, Brasil, Agosto 2011. IEEE. [6](#), [35](#), [40](#), [55](#), [59](#)
- [LH13a] A. Lara e R. Hirata Jr. A granulometry based descriptor for object categorization. Em *Proc. of ISMM - International Conference on Mathematical Morphology*, Uppsala, Suécia, Maio 2013. Springer. [6](#), [35](#), [37](#), [40](#), [49](#), [55](#)
- [LH13b] A. Lara e R. Hirata Jr. A pyramid of concentric circular regions to improve rotation invariance in bag-of-words approach for object categorization. Em S. Battiatto e J. Braz, editors, *Proc. of VISAPP - International Conference on Computer Vision Theory and Applications*, Barcelona, Espanha, Fevereiro 2013. SCITEPRESS. [6](#), [35](#), [46](#), [55](#)
- [LJ11] S. Li e A. Jain. *Handbook of face recognition*. Springer, 2011. [2](#)
- [LM01] Thomas Leung e Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001. [12](#)
- [LNH09] C.H. Lampert, H. Nickisch e S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. Em *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, páginas 951–958. IEEE, 2009. [74](#)
- [Low99] D. Lowe. Object recognition from local scale-invariant features. Em *The 1999 7th IEEE International Conference on Computer Vision (ICCV'99)*, páginas 1150–1157, Setembro 1999. [22](#)
- [Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. [22](#)
- [LPW⁺10] D. Le, S. Poullot, X. Wu, B. Nouvel e S. Satoh. National institute of informatics, japan at trecvid 2010. Em *Proceedings of TRECVID 2010 - National Institute of Standards and Technology*. National Institute of Standards and Technology, novembro 2010. [98](#)

- [LSP06] S. Lazebnik, C. Schmid e J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Em *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, páginas 2169–2178, Nova Iorque, EUA, Junho 2006. 5, 16, 17, 35, 51, 55, 56, 83
- [LWL11] Lingqiao Liu, Lei Wang e Xinwang Liu. In defense of soft-assignment coding. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 2486–2493. IEEE, 2011. 15
- [Mat75] G. Matheron. *Random sets and integral geometry*, volume 261. Wiley New York, 1975. 30, 31, 33
- [MMLM⁺09] G. Martinez-Munoz, N. Larios, E. Mortensen, W. Zhang, A. Yamamuro, R. Paasch, N. Payet, D. Lytle, L. Shapiro, S. Todorovic et al. Dictionary-free categorization of very similar objects via stacked evidence trees. Em *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, páginas 549–556. IEEE, 2009. 18, 21
- [MS01] K. Mikolajczyk e C. Schmid. Indexing based on scale invariant interest points. Em *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, páginas 525–531, Vancouver, Canadá, Julho 2001. IEEE Computer Society. 23
- [MS05] Krystian Mikolajczyk e Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005. 14
- [Mun06] J. Mundy. Object recognition in the geometric era: A retrospective. *Toward category-level object recognition*, páginas 3–28, 2006. 12
- [NB77] R. Nevatia e T. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98, 1977. 10
- [NBE⁺93] C. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos e G. Taubin. Qbic project: querying images by content, using color, texture, and shape. Em *Storage and Retrieval for Image and Video Databases*, San Jose, EUA, 1993. SPIE. 10
- [New91] J. Newell III. *Pixel classification by morphological granulometric features*. Tese de Doutorado, Rochester Institute of Technology, 1991. 31, 34
- [NJT06] E. Nowak, F. Jurie e B. Triggs. Sampling strategies for bag-of-features image classification. Em *European Conference on Computer Vision 2006*, Graz, Áustria, maio 2006. Springer Berlin / Heidelberg. 23
- [NZ06] M.E. Nilsback e A. Zisserman. A visual vocabulary for flower classification. Em *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, páginas 1447–1454. IEEE, 2006. 18, 21
- [OAF⁺10] P. Over, G. Awad, J. Fiscus, B. Antonishek e M. Michel. Trecvid 2010 - an overview of the goals, tasks, data, evaluation mechanisms, and metrics. Em *Proceedings of TRECVID 2010 - National Institute of Standards and Technology*. National Institute of Standards and Technology, Novembro 2010. 89, 97
- [OD11] Stephen O’Hara e Bruce A Draper. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*, 2011. 15

- [OPH94] T. Ojala, M. Pietikainen e D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. Em *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, páginas 582–585. IEEE, 1994. [24](#)
- [PCI⁺07] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic e Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. Em *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, páginas 1–8. IEEE, 2007. [16](#)
- [Pon11] M. Ponti Jr. Combining classifiers: From the creation of ensembles to decision fusion. Em Dimas Martinez Adelailson Peixoto e Thales Vieira, editors, *Tutorials of Sibgrapi 2011*, Maceió, AL, Brasil, Agosto 2011. SBC. [25](#), [26](#), [27](#), [61](#)
- [Pre09] R.S. Pressman. *Software engineering: a practitioner's approach*. McGraw-hill New York, 7 edição, 2009. [89](#)
- [PVT12] Otavio AB Penatti, Eduardo Valle e Ricardo da S Torres. Are visual dictionaries generalizable? *arXiv preprint arXiv:1205.2663*, 2012. [15](#)
- [RTG00] Y. Rubner, C. Tomasi e J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. [25](#), [37](#)
- [RTMF08] C. Russell, A. Torralba, P. Murphy e T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. [28](#)
- [RW08] P. Roth e M. Winter. Survey of appearance-based methods for object recognition. Relatório Técnico ICG-TR-01/08, Graz University of Technology, Janeiro 2008. [22](#)
- [SDK⁺10] J. Schavemaker, P. Doets, W. Kraaij, S. Raaijmakers e M. Staalduin. Notebook paper: Tho instance search submissions. Em *Proceedings of TRECVID 2010 - National Institute of Standards and Tchnology*. National Institute of Standards and Tchnology, novembro 2010. [98](#)
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982. [31](#)
- [SI84] S. Selim e M. Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(1):81–87, Janeiro 1984. [12](#)
- [Soi02] P. Soille. *Morphological image analysis: principles and applications*. Springer, 2002. [31](#)
- [SRE⁺05] J. Sivic, R. Russell, A. Efros, A. Zisserman e W. Freeman. Discovering objects and their location in images. Em *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, páginas 370–377, San Diego, EUA, Junho 2005. IEEE Computer Society. [58](#)
- [SSSS07] S. Shalev-Shwartz, Y. Singer e N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. Em *Proceedings of the 24th international conference on Machine learning*, páginas 807–814. ACM, 2007. [25](#), [52](#)
- [SZ03] Josef Sivic e Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. Em *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, páginas 1470–1477. IEEE, 2003. [10](#), [12](#)

- [Sze11] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, 2011. 1, 2, 3, 25, 28
- [Tio12] Tiobe programming community index for november 2012, Novembro 2012. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. 90
- [TM08] Tinne Tuytelaars e Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008. 14
- [TMFR03] A. Torralba, K.P. Murphy, W.T. Freeman e M.A. Rubin. Context-based vision system for place and object recognition. Em *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, páginas 273–280. IEEE, 2003. 2
- [Tre10] M. Treiber. *An introduction to object recognition: Selected algorithms for a wide variety of applications*. Advances in pattern recognition. Springer, Nova Iorque, EUA, 2010. 1, 10, 11, 23
- [Tsa12] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 2012. 13, 14
- [USS10] Jasper RR Uijlings, Arnold WM Smeulders e Remko JH Scha. Real-time visual concept classification. *Multimedia, IEEE Transactions on*, 12(7):665–681, 2010. 16
- [VDSGS10a] K.E.A. Van De Sande, T. Gevers e C.G.M. Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010. 80
- [vdSGS10b] Koen EA van de Sande, Theo Gevers e Cees GM Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010. 14
- [vdSGS11] Koen EA van de Sande, Theo Gevers e Cees GM Snoek. Empowering visual categorization with the gpu. *Multimedia, IEEE Transactions on*, 13(1):60–70, 2011. 16
- [VF08] A. Vedaldi e B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. 22, 23
- [VF10] A. Vedaldi e B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. Em *International Conference on ACM Multimedia 2010*, páginas 1469–1472, Florença, Itália, Outubro 2010. 52, 55
- [Vin00] L. Vincent. Fast granulometric methods for the extraction of global image information. *The proceedings of PRASA*, páginas 119–140, 2000. 33
- [VJ01] P. Viola e M. Jones. Rapid object detection using a boosted cascade of simple features. Em *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on In Proceedings of the 2001*, páginas 511–518, Kauai, EUA, Abril 2001. IEEE Computer Society. 23
- [VL08] Ville Viitaniemi e Jorma Laaksonen. Experiments on selection of codebooks for local image feature histograms. Em *Visual Information Systems. Web-Based Visual Information Search and Management*, páginas 126–137. Springer, 2008. 16
- [VZ12] A. Vedaldi e A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012. 52

- [WBM⁺10] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie e P. Perona. Caltech-ucsd birds 200. Relatório Técnico CNS-TR-2010-001, California Institute of Technology, 2010. [4](#), [72](#)
- [WPB⁺11] C. Wah, S. Branson, P. Perona e S. Belongie. Multiclass recognition and part localization with humans in the loop. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 2524–2531. IEEE, 2011. [21](#)
- [WBW⁺11] C. Wah, S. Branson, P. Welinder, P. Perona e S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Relatório Técnico CNS-TR-2011-001, California Institute of Technology, 2011. [4](#), [21](#), [28](#), [56](#), [72](#)
- [Whi12] Kamin Whitehouse. Matlabcontrol: A java api to interact with matlab, Novembro 2012. <https://code.google.com/p/matlabcontrol/>. [91](#)
- [WYY⁺10] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang e Y. Gong. Locality-constrained linear coding for image classification. Em *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, páginas 3360–3367, San Francisco, EUA, Junho 2010. IEEE Computer Society. [5](#), [21](#), [23](#), [59](#), [99](#)
- [XHE⁺10] J. Xiao, J. Hays, K. Ehinger, A. Oliva e A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. Em *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, páginas 3484–3492, Junho 2010. [xv](#), [2](#), [3](#), [28](#), [66](#)
- [YAK00] M.H. Yang, N. Abuja e D. Kriegman. Face detection using mixtures of linear subspaces. Em *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, páginas 70–76. IEEE, 2000. [2](#)
- [YBFF12] B. Yao, G. Bradski e L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. Em *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, páginas 3466–3473. IEEE, 2012. [18](#), [21](#), [49](#), [74](#), [76](#), [79](#), [80](#), [83](#)
- [YZ11] Xinnan Yu e Yu-Jin Zhang. A 2-d histogram representation of images for pooling. Em *IS&T/SPIE Electronic Imaging*, páginas 787706–787706. International Society for Optics and Photonics, 2011. [15](#), [16](#)
- [ZBMM06] H. Zhang, A.C. Berg, M. Maire e J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. Em *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, páginas 2126–2136. IEEE, 2006. [2](#), [10](#)
- [ZM10] E. Zhang e M. Mayo. Enhanced spatial pyramid matching using log-polar-based image subdivision and representation. Em *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, páginas 208–213. IEEE, 2010. [18](#)
- [ZMB99] C. Zhang, S. Murai e E. Baltsavias. *Road network detection by mathematical morphology*, volume 8093. Institute of Geodesy and Photogrammetry, ETH-Hoenggerberg, 1999. [34](#)

Índice Remissivo

- Agrupamento, 16
Máximo, 16
Média, 16
- Bases de dados
Caltech-101, 28, 35–37, 39, 40, 46, 48, 58, 62, 69, 83, 84
Caltech-256, 28
Caltech-UCSD Birds-200, 4, 21, 28, 42, 49, 56, 72, 78, 83, 84
ImageNet, 21, 28
LabelMe, 28
SUN, 2, 3, 28, 66, 68
VOC Pascal, 2, 3, 21, 28, 30, 74, 89
- Categorização refinada, 3–6, 18, 21, 28, 37, 49, 50, 72, 74, 76–79, 83–85
- Descriptor
GRABED, 5, 6, 38, 40–45, 51, 52, 58–62, 65, 66, 68, 72, 80, 81, 83, 84
HOG, 24, 38, 83
LBP, 24, 51, 92
PHOG, 24, 38, 51, 58–61, 65, 68, 80, 81, 84
PHOW, 5, 24, 51, 56, 58, 59, 61, 62, 64–66, 68, 69, 76, 80, 92, 99, 100
PPD, 5, 43, 51, 84
PPE, 5, 41, 43–45, 51, 84, 85
SIFT, 13, 14, 22–24, 52, 56, 58, 59, 61, 64, 69
SURF, 14, 24, 51, 52, 55, 58, 59, 61, 85, 92
- Detector de Harris-Laplace, 14
- Detector de Hessian-Laplace, 14
- Dicionário visual, 5, 12, 13, 15, 16, 21, 46, 49–51, 56, 59, 61, 64–66, 68, 74, 87, 88, 98–100
- Diferença de Gaussianas, 14, 22
- Ecoc, 27
- Granulometria morfológica, 4, 5, 30, 33, 38, 83
- Holdout, 25
- K-fold, 25, 52, 77, 85
- Leave-one-out, 25
- Média
Média das Precisões Médias, 81
- Medidas
Acurácia, 29, 83
Média das Precisões Médias, 30, 53, 57
Precisão, 29, 84
Precisão Média, 30, 53, 57
Recuperação, 29
Modelo discriminante, 16, 37
Modelo generativo, 17
MSER, 14
- Núcleo qui-quadrado, 25, 37, 52, 57, 58, 66, 69, 74, 77
- PRCC, 5, 6, 45–48, 51, 69, 71, 72, 76, 77, 81, 84, 85, 92
- QEF, 4–6, 35–37, 42, 43, 51, 55–57, 59, 83–85
- Saco de palavras, 4–6, 10, 12–14, 16–18, 21, 35, 37, 46, 49, 59, 61, 68, 69, 71, 72, 78, 79, 83, 84, 98, 99
- SFS, 27
- SPM, 4, 5, 16–18, 20, 21, 24, 35–37, 45, 46, 51, 56, 57, 83, 84
- SVM, 22, 24, 37, 52, 57, 58, 66, 69, 74, 77, 92
- TF-IDF, 15
- WordNet, 28