



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Classificação Automática do *Diaphorina citri* em Imagens de Microscopia

José Leonardo dos Santos Melo

Feira de Santana

2016



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

José Leonardo dos Santos Melo

Classificação Automática do *Diaphorina citri* em Imagens de Microscopia

Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Orientadora: Dr.^a Michele Fúlvia Angelo

Feira de Santana

2016

Ficha Catalográfica – Biblioteca Central Julieta Carteado

M485c Melo, José Leonardo dos Santos
Classificação automática do *Diaphorina citri* em imagens de
microscopia / José Leonardo dos Santos Melo. – Feira de Santana, 2016.
96 f. : il.

Orientadora: Michele Fúlvia Angelo.

Dissertação (mestrado) – Universidade Estadual de Feira de Santana,
Programa de Pós-Graduação em Computação Aplicada, 2016.

1. Computação – Classificação de imagens – *Diaphorina citri*. I.
Angelo, Michele Fúlvia, orient. II. Universidade Estadual de Feira de
Santana. III. Título.

CDU: 004.65:632.752

Está página deverá ser substituída por uma folha contendo as assinaturas dos membros da banca.

Abstract

The *Huanglongbing* (HLB) is the disease of greatest concern for growers because they spread quickly and cause severe symptoms. The *Diaphorina citri* insect is the main vector of the HLB. The application of insecticides is a control measure of the vector insect of the HLB widely adopted. The amount of pesticides needed for an effective control of this insect is better estimated if such application is combined with a monitoring of its population by yellow sticky traps. These insects are captured for a manual count in research centers. So, this research aims to discover a computational approach of classification of *Diaphorina citri* insect images with higher accuracy rate that the classification rate currently used in manual counting procedure and thus enable the automation of this important counting procedure. For this, have been tried and combined computational methods for features extraction (ORB, SIFT, SURF, BRISK and FREAK), grouping of characteristics (*Mini Batch K-Means*) and features classification for machine learning (KNN and SVM), using a generated bank with 1152 images of insects. The best found classification approach (extractor SURF/SIFT, BoF with *Diaphorina citri* features and SVM with core RBF) generated classification performance results for the metric accuracy, which outperformed the best measured result in research that evaluated the counting manual process. In this approach, the highest achieved accuracy, in the cross validation process, was 98.17% and was 2.54% as standard deviation and the accuracy of the final test of generalization model was 99.14%. The achieved result is of great importance for the control of HLB. The achieved classification accuracy rates were higher than rates reported in the manual procedure, making possible the construction of computer systems to high accuracy for the control of this insect. This automated control can provide significant savings of funds.

Keywords: *Diaphorina Citri*, Yellow Sticky Traps, Machine Learning, Image Classification, Image Processing.

Resumo

O *Huanglongbing* (HLB) é a doença de maior preocupação para os citricultores por se propagar com rapidez e provocar severos sintomas. O inseto *Diaphorina citri* é o principal vetor do HLB. A aplicação de inseticidas é uma medida de controle do inseto vetor do HLB amplamente adotada. A quantidade de inseticidas necessária para um controle efetivo desse inseto é melhor estimada se essa aplicação for combinada a um monitoramento de sua população por meio de armadilhas adesivas amarelas. Esses insetos são capturados para uma contagem manual em centros de pesquisa. Então, esta pesquisa tem por objetivo descobrir uma abordagem computacional de classificação de imagens de insetos *Diaphorina citri* com taxa de acurácia de classificação maiores que a taxa de classificação utilizada atualmente no procedimento manual de contagem e, assim, possibilitar a automação desse importante procedimento de contagem. Para isso, foram experimentados e combinados métodos computacionais para a extração de características (ORB, SIFT, SURF, BRISK e FREAK), agrupamento de características (*Mini Batch K-Means*) e classificação de características por aprendizagem de máquina (KNN e SVM), utilizando um banco gerado com 1152 imagens de insetos. A melhor abordagem de classificação encontrada (extrator SURF/SIFT, BoF com características do *Diaphorina citri* e SVM com núcleo RBF) gerou resultados de desempenho de classificação, pela métrica da acurácia, que superaram o melhor resultado medido na pesquisa que avaliou o processo de contagem manual. Nessa abordagem, a maior acurácia atingida no processo de validação cruzada foi de 98,17% e teve 2,54% como desvio padrão e a acurácia do teste final de generalização de modelo foi de 99,14%. O resultado alcançado é de grande importância para o controle do HLB. As taxas de acurácia de classificação alcançadas foram superiores as taxas relatadas no procedimento manual, tornando viável a construção de sistemas computacionais de alta acurácia para o controle desse inseto. Esse controle automatizado pode proporcionar uma economia significativa de recursos financeiros.

Palavras-chave: *Diaphorina Citri*, Armadilhas Adesivas Amarelas, Aprendizagem de Máquina, Classificação de Imagens, Processamento de Imagens.

Prefácio

Esta dissertação de mestrado foi submetida à Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

A dissertação foi desenvolvida dentro do Programa de Pós-Graduação em Computação Aplicada (PGCA) tendo como orientadora a Dr.^a **Michele Fúlvia Angelo**.

Esta pesquisa foi financiada pela Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB).

Agradecimentos

Agradeço primeiramente a Deus, força maior que nos protege e conduz nossos caminhos.

À Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) pelo auxílio financeiro, tornando viável a geração dos resultados de desta pesquisa nos prazos estabelecidos pelo Programa de Pós-Graduação em Computação Aplicada (PGCA) da Universidade Estadual de Feira de Santana (UEFS) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

Ao Fundo de Defesa da Citricultura (FUNDECITRUS) pelo apoio científico-tecnológico de alta qualidade e colaboração de pesquisadores de excelência. Esse apoio tornou possível a criação do banco de imagens com representativa amostragem da população dos insetos da citricultura paulista estudados, dentro dos prazos do projeto.

Aos professores doutores que direta ou indiretamente contribuíram para a realização deste projeto de pesquisa. Em especial aos professores e pesquisadores: Dr.^a Michele Fúlvia Angelo (PGCA/UEFS), Dr. Marcelo Pedreira de Miranda (FUNDECITRUS) e Dr. Angelo Amâncio Duarte (PGCA/UEFS).

À professora Michele pela excelência na orientação desta pesquisa, tornando possível os resultados alcançados.

Ao professor Marcelo pela transmissão de conhecimento sobre o problema de pesquisa, suporte tecnológico e científico em todas fases do projeto e pelo auxílio na aquisição e cessão do banco de imagens.

Ao professor Angelo pelos excelentes e diferenciados ensinamentos a respeito da metodologia científica e pelas conversas sobre projeção de carreira e suas lições de vida.

À família pela paciência e compreensão nos momentos de ausência e aos colegas de curso (PGCA/UEFS) que contribuíram com a troca de experiências e conhecimentos.

Sumário

Abstract	i
Resumo	ii
Prefácio	iii
Agradecimentos	iv
Sumário	vi
Lista de Publicações	vii
Lista de Tabelas	ix
Lista de Figuras	xii
Lista de Abreviações	xiii
Lista de Símbolos	xiv
1 Introdução	1
1.1 Objetivos	4
1.1.1 Específicos	4
1.2 Contribuição	5
1.3 Organização do Trabalho	5
2 Revisão Bibliográfica	6
2.1 Psilídeo dos Citros Asiático	6
2.1.1 Plantas Hospedeiras	7
2.1.2 Danos Causados pela Alimentação do Psilídeo dos Citros Asiático	7
2.1.3 Psilídeo dos Citros Asiático e o <i>Greening</i>	10
2.1.4 Monitoramento do Psilídeo dos Citros Asiático	12
2.2 Representação e Descrição de Características em Imagens	14
2.2.1 ORB	15

2.2.2	SIFT	17
2.2.3	SURF	20
2.2.4	BRISK	22
2.2.5	FREAK	24
2.3	Bolsas de Características	26
2.4	Aprendizagem de Máquina	28
2.4.1	Agrupamento (K-Means e Mini Batch K-Means)	31
2.4.2	K Vizinhos Mais Próximos	31
2.4.3	Máquinas de Vetores de Suporte	32
2.5	Trabalhos Relacionados	33
3	Métodos e Experimentos	37
3.1	Construção e Validação do Banco de Imagens	37
3.2	Rotulagem de Amostras e Extração de Características	42
3.3	Criação dos Livros de Código e das Bolsas de Características	43
3.4	Classificação das Amostras	45
3.5	Aprimoramento dos Classificadores	48
4	Resultados e Discussões	51
4.1	Primeira Fase de Experimentação Geral	52
4.2	Segunda Fase de Experimentação Geral	57
4.3	Estudo dos Agrupamentos das Bolsas de Características	62
4.4	Aprimoramento dos Classificadores	64
4.4.1	Discussão dos Resultados	67
5	Considerações Finais	69
5.1	Aplicações	70
5.2	Pesquisas Futuras	70
	Referências Bibliográficas	72

Lista de Publicações

[Melo e Ragni 2014] Melo, J. L. d. S. e Ragni, M. (2014). Processamento digital de imagens e reconhecimento de padrões no controle de pragas e doenças da citricultura: Diferenciação do psíldeo em armadilhas adesivas amarelas. Anais da ERBASE-Escola de Computação Bahia-Alagoas-Sergipe.

[Melo e Angelo 2015] Melo, J. L. d. S. e Angelo, M. F. (2015). Reconhecimento automático do inseto diaphorina citri em imagens de microscopia. In Rios, R. A. e Paiva, A., editors, Proceedings..., Porto Alegre. Conference on Graphics, Patterns and Images, 28. (SIBGRAPI), Sociedade Brasileira de Computação.

[Melo et al. 2015] Melo, J. L. d. S., Angelo, M. F., e Miranda, M. P. (2015). Classification of diaphorina citri into microscopy images. In Vieira, M. A. d. C., Rodrigues, E. L. L., e Escarpinati, M. C., editors, Proceedings..., São Carlos (SP). XI Workshop de Visão Computacional, LAVI - EESC/USP.

Lista de Tabelas

4.1	Dados gerados a partir de extrações de características com descritores e seus próprios detectores de pontos.	52
4.2	Dados gerados a partir de classificações de características com bolsas de características (Geral) e detectores de pontos próprios.	53
4.3	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Geral) com detectores de pontos próprios.	54
4.4	Dados gerados a partir de classificações de características com bolsas de características (<i>Diaphorina citri</i>) e detectores de pontos próprios.	54
4.5	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (<i>Diaphorina citri</i>) com detectores de pontos próprios.	55
4.6	Dados gerados a partir de classificações de características com bolsas de características (Por Classe) e detectores de pontos próprios.	55
4.7	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Por Classe) com detectores de pontos próprios.	56
4.8	Dados gerados a partir de extrações de características com descritores combinados ao detector de pontos SURF.	57
4.9	Dados gerados a partir de classificações de características com bolsas de características (Geral) e detectores de pontos SURF.	58
4.10	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Geral) com detectores de pontos SURF.	59
4.11	Dados gerados a partir de classificações de características com bolsas de características (<i>Diaphorina citri</i>) e detectores de pontos SURF.	59
4.12	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (<i>Diaphorina citri</i>) com detectores de pontos SURF.	60
4.13	Dados gerados a partir de classificações de características com bolsas de características (Por Classe) e detectores de pontos SURF.	60
4.14	Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Por Classe) com detectores de pontos SURF.	61

4.15	Relatório de teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo KNN.	65
4.16	Matriz de Confusão gerada a partir do teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo KNN.	65
4.17	Relatório de teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo SVM.	66
4.18	Matriz de Confusão gerada a partir do teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo SVM.	67

Lista de Figuras

2.1	Psilídeo dos citros asiático adulto, <i>Diaphorina citri</i> Kuwayama.	7
2.2	Crescimento de fungos devido a secreção excretada por ninfas.	8
2.3	Pontas torcidas de citros devido a alimentação do psilídeo.	8
2.4	Folhas queimadas de citros devido à alimentação do psilídeo.	9
2.5	Malformação de folhas de citros devido à alimentação do psilídeo.	9
2.6	Folhas de citros torcidas devido à alimentação do psilídeo.	10
2.7	Amarelamento das folhas devido a doença dos citros Greening.	11
2.8	Clorose das folhas dos citros devido ao Greening.	11
2.9	Frutos e sementes deformados devido ao Greening.	11
2.10	Pontas das folhas dos citros torcidas devido ao Greening.	12
2.11	Inclinação do corpo do psilídeo dos citros adulto.	12
2.12	Secreção, fungos e túbulos de cera produzidos por psilídeos.	13
2.13	Grande infestação de psilídeos em jasmim laranja.	13
2.14	Armadilha adesiva amarela para a detecção de psilídeos.	13
2.15	Dois testes binários do descritor BRIEF.	17
2.16	SIFT - Exemplo de magnitude de gradiente e orientação em imagem.	18
2.17	SIFT - Gradientes de imagem (esquerda) e descritor de pontos-chave (direita).	19
2.18	Da esquerda para a direita: a derivada parcial Gaussiana de segunda ordem nas direções y e xy , respectivamente; e a aproximação para a derivada parcial Gaussiana de segunda ordem nas direções y e xy . A intensidade da região cinza é zero.	21
2.19	Descritor de características SURF-64 - Gradientes de imagem.	21
2.20	Padrão de amostragem do BRISK.	24
2.21	Padrão de amostragem do FREAK.	26
2.22	Quatro passos para a construção de BoF para a representação de imagens.	27
3.1	Etapas da metodologia utilizada.	37
3.2	Equipamento de aquisição das imagens e armadilhas adesivas amarelas.	38
3.3	Algumas amostras do banco de imagens do <i>Diaphorina citri</i>	39
3.4	Algumas amostras do banco de imagens dos outros insetos.	39
3.5	Imagem do <i>Diaphorina citri</i> carregada com um espaço de cores baseado em modelo RGB.	40

3.6	Imagem do <i>Diaphorina citri</i> carregada com o canal K de um espaço de cores baseado em modelo CMYK.	41
3.7	Histograma do canal K de um espaço de cores baseado em modelo CMYK - Imagem do <i>Diaphorina citri</i>	41
3.8	Limiarização da imagem do canal K utilizando o <i>threshold</i> encontrado - Imagem do <i>Diaphorina citri</i>	41
3.9	Máscara binária feita a partir do contorno do objeto principal encontrado - Imagem do <i>Diaphorina citri</i>	42
3.10	Recorte feito a partir do máscara binária encontrada - Imagem (RGB) do <i>Diaphorina citri</i>	42
3.11	Divisão do banco de amostras com características extraídas das imagens dos insetos, processo de validação cruzada e avaliação final do desempenho de classificação.	47
3.12	Etapas detalhadas do processo de classificação e geração de resultados.	48
4.1	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.2.	54
4.2	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.2.	54
4.3	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.4.	55
4.4	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.4.	55
4.5	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.6.	56
4.6	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.6.	56
4.7	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.9.	58
4.8	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.9.	58
4.9	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.11.	59
4.10	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.11.	59
4.11	Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.13.	60
4.12	Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.13.	60
4.13	Estudo da variação do parâmetro <i>K</i> (número de agrupamentos) para bolsas de características “ <i>Diaphorina</i> ”, extrator “SURF/SIFT” e algoritmo KNN.	63

4.14 Estudo da variação do parâmetro K (número de agrupamentos) para bolsas de características “Diaphorina”, extrator “SURF/SIFT” e algoritmo SVM.	63
--	----

Lista de Abrebiações

Abreviação	Descrição
PIB	Produto Interno Bruto (<i>Gross Domestic Product</i>)
HLB	Doença Greening (<i>Huanglongbing</i>)
EUA	Estados Unidos da América (<i>USA - United States of America</i>)
ADN	Ácido Desoxirribonucleico (<i>DNA - Deoxyribonucleic Acid</i>)

Lista de Símbolos

Símbolo	Descrição
US\$	Dólar dos Estados Unidos (<i>United States Dollar</i>)
kg	Quilograma (<i>kilogram or kilogramme</i>)
mm	Milímetro (<i>Millimeter</i>)
”	Polegadas (<i>in - Inch</i>)
°	Grau (<i>Degree</i>)
°C	Grau Celsius (<i>Degree Celsius</i>)
%	Porcentagem (<i>Percentage</i>)

Capítulo 1

Introdução

*“Imaginar é o princípio da criação.
Nós imaginamos o que desejamos,
queremos o que imaginamos e,
finalmente, criamos aquilo que
queremos.”*

– George Bernard Shaw

O Brasil é o maior produtor mundial de laranja e somente na safra de 2013/14 a produção brasileira foi de 17.340 toneladas, aproximadamente 34,22% de toda a produção mundial. O país também se destacou no consumo dessa fruta com 5.788 toneladas no mesmo período, sendo assim o segundo maior consumidor do mundo [Mendes 2015].

A produção de suco de laranja no Brasil, na safra 2013/14, correspondeu a 57,05% de toda produção mundial, 95,82% dessa produção foi exportada e, desta forma, a participação brasileira no mercado mundial foi de 78,60% [Mendes 2015].

Lopes et al. [2011] afirmaram que a citricultura ¹ tem grande importância econômica para o Brasil e contribui com o PIB (Produto Interno Bruto) nacional. Essa cultura gerou 9 bilhões de reais, contribuindo com a balança comercial brasileira. Além disso, gerou 400.000 empregos diretos e indiretos na área rural .

O “cinturão citrícola brasileiro” é a região de produção citrícola de destaque do país e está localizado nos estados de São Paulo, Paraná e Triângulo Mineiro. Apesar do estado de São Paulo ser a maior região produtora de laranja do mundo, existem diversos estados brasileiros produzindo essa fruta [Mendes 2015].

O estado de São Paulo, na safra de 2013/14, foi responsável por cerca de 73% da produção brasileira de laranja, o que corresponde a 399 milhões de caixas de 40,8

¹Citricultura é o cultivo de plantas cítricas. Planta cítrica é a que produz frutos contendo ácido cítrico. Exemplo: A laranjeira e o limoeiro [Ferreira 2010].

kg ou 16 milhões de toneladas. No mesmo período, esse estado foi responsável por cerca de 64% da área colhida no país, o que corresponde a aproximadamente 707 mil hectares [Mendes 2015].

As pragas² e as doenças trazem grandes preocupações para a citricultura de modo geral e, por consequência, para o cultivo da laranja. Elas são responsáveis por significativas perdas no processo produtivo citrícola [Bieras e Santos 2003].

Neves et al. [2010] afirmaram que as pragas e as doenças foram a causa da erradicação³ de 39 milhões de árvores do parque citrícola de São Paulo e Triângulo Mineiro, no período entre 2000 e 2009. No último ano desse período, a taxa de mortalidade de árvores, causadas por pragas e doenças, correspondeu ao preocupante percentual de 7,5%.

Das doenças presentes na citricultura, e em especial na citricultura brasileira, o *Greening*, também conhecido como *Huanglongbing* (HLB), é a doença de maior preocupação para citricultores e pesquisadores do setor por se propagar com rapidez e provocar severos sintomas. O HLB é uma doença que afeta os citros e é causada por bactérias (*Candidatus Liberibacter*). As plantas cítricas, quando infectadas por alguma dessas bactérias, não desenvolvem resistência genética para combater essa doença. O inseto conhecido como psíldeo dos citros asiático (*Diaphorina citri* Kuwayama), que mede de 3 e 4 mm de comprimento, é o principal transmissor das bactérias causadoras do HLB no estado de São Paulo, ou seja, é o seu inseto vetor [Grafton-Cardwell et al. 2005, Teixeira et al. 2005, Neves et al. 2010, Agnelli 2011, Mendes 2015].

Ainda não existe a cura do HLB e as medidas para controlar essa doença não são efetivas. Então, prevenir a infecção das plantas é de extrema importância para o controle dessa doença [Belasque Junior et al. 2009, Mendes 2015].

Na tentativa de controlar o HLB foram aplicadas injeções de antibióticos em plantas infectadas, mas essa ação não obteve resultados promissores. A aplicação de antibióticos traz consigo riscos ao meio ambiente e quando se interrompe esse tipo de tratamento os sintomas dessa doença voltam a aparecer, ou seja, as plantas continuam doentes [Bové 2006]. Outro tipo de medida de controle do HLB também foi testado, a poda de ramos sintomáticos e assintomáticos, e essa medida se mostrou uma prática ineficaz [Lopes et al. 2007].

O controle recomendado, na atualidade, envolve a erradicação de plantas doentes, o plantio de mudas livres da doença e o controle do inseto vetor. Essas medidas resultam em gastos com recursos financeiros e humanos e requerem tempo, além da coordenação dos agentes envolvidos na cadeia de produção de frutas e sucos (governo, produtores e indústrias processadoras) [Belasque Junior et al. 2009].

²Praga é o nome genericamente dado a insetos ou a doenças que atacam plantas e animais [Ferreira 2010].

³Erradicar é o ato de extrair as plantas em sua totalidade, inclusive a raiz [Ferreira 2010].

Roistacher [1996] demonstrou que a manutenção de plantas doentes tornou-se inviável economicamente em menos de uma década após a constatação do primeiro caso de HLB em uma propriedade produtora monitorada. Por outro lado, Belasque Junior et al. [2009] demonstraram que as aplicações de inseticidas para o controle de psilídeos dos citros asiático estabilizou o número de erradicações à perda anual máxima de 2% das plantas produtoras, permitindo que pomares contaminados se mantivessem economicamente viáveis e produtivos.

A aplicação de inseticidas (controle químico) é uma medida de controle do inseto vetor do HLB amplamente adotada pelos citricultores e tem maior eficiência quando utilizada em larga escala, cobrindo grandes regiões citrícolas. A quantidade de inseticidas necessária para um controle efetivo do psilídeo dos citros asiático é melhor estimada se essa aplicação for combinada a um monitoramento da população desse inseto por meio de armadilhas adesivas amarelas e avaliações visuais dos ramos das plantas [Belasque Junior et al. 2009, Sala 2011].

As armadilhas adesivas amarelas são utilizadas na atração e captura do psilídeo dos citros asiático em sua fase adulta. Os insetos capturados nessas armadilhas podem ser encontrados em diferentes posições, rotacionados no espaço tridimensional e com seu exoesqueleto movido de diversas maneiras. Essas armadilhas são presas aos galhos das plantas e permanecem expostas nos pomares por dias. Então, elas são recolhidas e levadas para laboratórios em centros de pesquisa. Nesses laboratórios são realizadas contagens manuais dos psilídeos dos citros asiáticos capturados e, com isso, são calculadas estimativas populacionais relacionadas à região de onde essas armadilhas foram retiradas [Sala 2011, Oliveira 2013].

Um efeito colateral ao monitoramento de psilídeos dos citros asiático por meio de armadilhas adesivas amarelas é que elas atraem e capturam outros tipos de insetos [Yamamoto et al. 2001]. Além disso, uma dificuldade encontrada na realização de contagens manuais de psilídeos dos citros asiático nessas armadilhas é o tamanho desse inseto, 3 a 4 mm de comprimento, e isso dificulta o seu reconhecimento através das suas características quando observadas sem o auxílio de lentes de aumento [Grafton-Cardwell et al. 2005]. Um outro fator que dificulta o reconhecimento desses insetos é a grande variação de posições em que eles são encontrados presos (capturados) nessas armadilhas.

A contagem de psilídeos dos citros asiáticos em armadilhas adesivas amarelas é comumente feita por um técnico previamente treinado. Empresas do setor citrícola paulista realizam esse procedimento em tempos médios de 3 a 5 minutos e algumas delas realizam auditorias para verificar a acurácia de inspeção de seus técnicos. A característica de processo manual introduz o fator humano a essa atividade de contagem. Segundo Dejours [1997], o fator humano está associado a ideia de faltas, falhas ou erros cometidos por seres humanos na prática de operações manuais e muitas vezes repetitivas. Esse fator pode introduzir imprecisão de resultados de contagem, análises de reconhecimento de insetos subjetivas e lentidão nas etapas desse processo.

Leonardo [2014] realizou um estudo comparativo sobre a eficiência do reconhecimento manual do *Diaphorina citri* em armadilhas adesivas amarelas utilizando técnicos treinados. Nesse estudo foi constatado que o procedimento de reconhecimento tem melhor desempenho nos laboratórios do que no próprio campo e que a melhor acurácia média de reconhecimento alcançada no experimento foi de aproximadamente 93%, a maior acurácia atingida por um técnico foi de aproximadamente 95% e a menor foi de aproximadamente 91%, demonstrando assim boas taxas de acurácia nesse procedimento. Esse estudo mostrou ainda a importância da inspeção das armadilhas adesivas amarelas no processo de monitoramento do HLB e que existem custos de tempo, recursos humanos e financeiros associados a esse processo de inspeção manual.

Então, classificar automática e corretamente imagens de psíldeos dos citros asiáticos é uma importante etapa, e objetivo central de pesquisa, para viabilizar a construção de sistemas computacionais que resultem na automação do processo de reconhecimento desses insetos em armadilhas adesivas amarelas e assim reduzir consideravelmente os custos operacionais, imprecisões e instabilidades de resultados de classificação associados ao processo manual. Esses sistemas computacionais, quando viabilizados pelos resultados desta pesquisa, poderão, adicionalmente, enviar informações com dados desses insetos e suas estimativas de quantidade em armadilhas para sistemas de informações de monitoramento populacional de psíldeos dos citros asiáticos, atualmente existentes, e com isso proporcionar um monitoramento mais ágil e preciso do principal vetor do HLB.

1.1 Objetivos

O objetivo geral desta pesquisa é realizar um estudo que possibilite a classificação de psíldeos dos citros asiáticos (*Diaphorina citri* Kuwayama) por procedimento computacional, distinguindo-os de outros insetos que também possam ser capturados por armadilhas adesivas amarelas, através de imagens de insetos extraídas dessas armadilhas.

1.1.1 Específicos

Abaixo estão listados os objetivos específicos:

- Criar um banco de imagens com amostras dos insetos a serem estudados;
- Encontrar abordagens computacionais (conjuntos de técnicas - algoritmos) para a extração de características em imagens que possuem variação de escala ou rotação de objetos e que sejam aplicáveis a extração de características de insetos;

- Encontrar abordagens computacionais (conjuntos de técnicas - algoritmos) para a classificação de características em imagens que sejam aplicáveis a classificação de características de insetos;
- Descobrir um procedimento computacional aprimorado capaz de classificar o psilídeo dos citros asiático em imagens de microscopia com taxas médias de acurácia iguais ou superiores as taxas alcançadas no procedimento manual.

1.2 Contribuição

A principal contribuição deste trabalho é possibilitar a construção de sistemas automatizados de visão computacional para o reconhecimento e a contagem de psilídeos dos citros asiáticos, gerando assim inovações tecnológicas.

1.3 Organização do Trabalho

No capítulo 2 é apresentada a revisão bibliográfica necessária para a compreensão desta pesquisa, no capítulo 3 é descrita a metodologia empregada, no capítulo 4 são apresentados e analisados os resultados alcançados e no capítulo 5 são apresentadas as conclusões obtidas através dos resultados, as possibilidades de aplicações e apontadas sugestões de futuras pesquisas.

Capítulo 2

Revisão Bibliográfica

“A ignorância gera mais frequentemente confiança do que o conhecimento: são os que sabem pouco, e não aqueles que sabem muito, que afirmam de uma forma tão categórica que este ou aquele problema nunca será resolvido pela ciência.”

– Charles Darwin

Este capítulo apresenta temas necessários à compreensão desta pesquisa e os temas abordados são: detalhes sobre o psilídeo *Diaphorina citri* Kuwayama e a doença dos citros *Greening*, descrição de técnicas de Processamento Digital de Imagens e Aprendizagem de Máquina utilizadas e os trabalhos correlatos analisados.

2.1 Psilídeo dos Citros Asiático

O psilídeo dos citros asiático ou *Diaphorina citri* Kuwayama (Homoptera: Psyllidae) (Figura 2.1) é uma praga dos citros e plantas parentes próximas. Esse inseto prejudica diretamente essas plantas através de suas atividades de alimentação. Um novo broto em fase de crescimento e que esteja fortemente infestado por psilídeos dos citros asiáticos não cresce ou se desenvolve normalmente e está mais vulnerável a um rompimento das partes mais afetadas. Os danos diretos são graves, mas existe uma preocupação ainda maior relacionada aos danos indiretos. E essa preocupação se dá pelo fato desse inseto ser um eficiente vetor (transmissor) de bactérias que causam a doença dos citros, economicamente devastadora, conhecida como *Greening* ou *Huanglongbing* (HLB) [Grafton-Cardwell et al. 2005].



Figura 2.1: Psilídeo dos citros asiático adulto, *Diaphorina citri* Kuwayama.

Fonte: Foto de Jeffrey W. Lotz, disponível em Mead e Fasulo [1998].

Psilídeos dos citros asiáticos foram encontrados em diversas áreas do mundo, como: na Ásia tropical e subtropical, Afeganistão, Arábia Saudita, ilha francesa de Reunião, República de Maurício, partes das Américas do Sul e Central, México e no Caribe. Esses insetos foram identificados pela primeira vez dentro dos Estados Unidos em Palm Beach, Flórida, no mês de julho do ano de 1998, quando foram descobertos em jasmim laranja (*Murraya paniculata*), planta geralmente cultivada em quintais americanos. O transporte de mudas infestadas por psilídeos dos citros asiáticos contribuiu fortemente para a propagação desses insetos [Halbert et al. 2002].

Na primavera do hemisfério norte do ano de 2001, o psilídeo dos citros asiático foi acidentalmente introduzido em Vale do Rio Grande, no estado do Texas, transportado em vasos contendo mudas de *Murraya paniculata*, vindos do estado da Flórida [French et al. 2001].

2.1.1 Plantas Hospedeiras

Dentre os gêneros de plantas que podem servir de hospedeiros para o psilídeo dos citros asiático pode-se citar 25 da família Rutaceae, mas nem todos esses gêneros são bons hospedeiros para esse inseto [Halbert e Manjunath 2004]. Os gêneros de plantas hospedeiras mais comuns e preferidos por esse psilídeo são: *Citropsis*, *Citrus* e *Murraya*. Sendo um conjunto pequeno de gêneros de plantas hospedeiras e sabendo que esse inseto se aloja mais comumente em hospedeiros preferenciais, os esforços de monitoramento dessa praga focam nos citros e em seus parentes próximos [Grafton-Cardwell et al. 2005].

2.1.2 Danos Causados pela Alimentação do Psilídeo dos Citros Asiático

O psilídeo dos citros asiático danifica diretamente citros e plantas ornamentais que são parentes próximas dos citros [Halbert e Manjunath 2004]. Os psilídeos extraem

grandes quantidades de seiva quando eles se alimentam e, por consequência, produzem grandes quantidades de secreção. Essa secreção cobre as folhas das árvores, facilitando o crescimento de fungos (Figura 2.1.2).



Figura 2.2: Crescimento de fungos devido a secreção excretada por ninfas.
Fonte: Foto de M. E. Rogers, disponível em Grafton-Cardwell et al. [2005].

Adicionalmente, quando os psíldeos dos citros asiáticos se alimentam, eles injetam uma toxina salivar para o alongamento daquele canal da planta, causando a malformação das folhas e brotos (Figura 2.1.2) [Michaud 2004].



Figura 2.3: Pontas torcidas de citros devido a alimentação do psíldeo.
Fonte: Foto de Grafton-Cardwell et al. [2005].

Uma simples alimentação, de menos de 24 horas, de uma ninfa de psíldeo provoca a malformação permanente de uma folha. No período de hibernação, psíldeos adultos se agregam sobre novos brotos em formação, onde eles se alimentam e acasalam [Halbert e Manjunath 2004].

Frequentemente, infestações iniciais de psíldeos dos citros asiáticos são altamente concentradas sobre árvores isoladas, dentro dos pomares de citros. Essas concentrações e alimentações causam distorções dos brotos que proporciona

melhores locais de oviposição. A nutrição dos citros é, muitas vezes, seriamente danificada (Figuras 2.1.2 e 2.1.2), resultando em queda das folhas e brotos [Halbert e Manjunath 2004] ou folhas maduras malformadas (Figura 2.1.2). Árvores adultas podem tolerar estes danos já que a perda de folhas ou brotos acontece apenas em uma pequena parte da copa total da árvore. As mudas e as novas plantações podem requerer proteção química [Grafton-Cardwell et al. 2005].



Figura 2.4: Folhas queimadas de citros devido à alimentação do psilídeo.
Fonte: Foto de M. E. Rogers, disponível em Dara [2011].



Figura 2.5: Malformação de folhas de citros devido à alimentação do psilídeo.
Fonte: Foto de M. E. Rogers, disponível em Warnert et al. [2012].



Figura 2.6: Folhas de citros torcidas devido à alimentação do psíldeo.

Fonte: Foto de M. E. Rogers, disponível em Warnert et al. [2012].

2.1.3 Psíldeo dos Citros Asiático e o *Greening*

Os danos mais graves que o psíldeo dos citros asiático pode causar estão relacionados com sua capacidade de ser um eficiente vetor para as bactérias *Candidatus Liberibacter africanus*, *Candidatus Liberibacter americanus* e *Candidatus Liberibacter asiaticus*. Essas bactérias causam a doença dos citros, conhecida como *Greening* ou *Huanglongbing* (HLB). No passado, essas bactérias eram extremamente difíceis de serem detectadas e caracterizadas. Nos últimos anos, no entanto, a hibridização de DNA (*Deoxyribonucleic Acid*) com sonda, a microscopia eletrônica e testes ELISA (*Enzyme-Linked Immunosorbent Assay*) foram desenvolvimentos que melhoram a detecção dessas bactérias [Roistacher 1991, Garnier e Bové 1993, Bové et al. 1996].

Os sintomas do *Greening* incluem brotos amarelados (Figura 2.1.3), manchas e insuficiência de clorofila (clorose) das folhas [Capoor et al. 1974]. As manchas se assemelham, superficialmente, a deficiência de zinco. No entanto, as manchas associadas ao *Greening* não são encontradas somente ao longo das veias, como na deficiência de zinco, mas atravessam as nervuras (Figura 2.1.3). As árvores infectadas são raquíticas, com folhas esparsadas, e podem florescer fora da estação. Além disso, há pericínio dos galhos, queda de folhas e frutos, produção de frutos duros, pequenos e tortos com suas sementes duras pequenas e atrofiadas (Figura 2.1.3). O suco da fruta infectada tem um gosto amargo. As frutas não são corretamente coloridas, levando ao nome *Greening* (“Esverdeamento”). Essa doença é uma das mais devastadoras doenças dos citros no mundo. A injeção de antibióticos em árvores cítricas proporcionou uma remissão, apenas temporária, dos sintomas. Em partes da Ásia, o *Greening* reduziu substancialmente a quantidade de citros cultivados [Salibe e Cortez 1966, Catling et al. 1970].



Figura 2.7: Amarelamento das folhas devido a doença dos citros Greening.
Fonte: Foto de Susan E. Halbert, disponível em Dara [2011].



Figura 2.8: Clorose das folhas dos citros devido ao Greening.
Fonte: Foto de Grafton-Cardwell et al. [2005].

A forma asiática do *Greening* é encontrada por todas as áreas mais quentes da Ásia, Índia, e da península da Arábia Saudita (Figura ??). Em junho de 2004, as bactérias *Candidatus Liberibacter asiaticus* e *Candidatus Liberibacter americanus* isoladas foram encontradas no Brasil, afetando pomares do estado de São Paulo [Teixeira et al. 2005].



Figura 2.9: Frutos e sementes deformados devido ao Greening.
Fonte: Foto de M. E. Rogers, disponível em Dara [2011].

2.1.4 Monitoramento do Psilídeo dos Citros Asiático

O monitoramento é conduzido através da inspeção visual de novos brotos de citros (Figura 2.1.4) e mudas de plantas ornamentais tais como a jasmim laranja (*Murraya paniculata*) que potencialmente atuam como hospedeiros do psilídeo dos citros asiático. É recomendada uma inspeção detalhada buscando por esses insetos em todos os seus estágios, incluindo insetos adultos com cores de cinza a castanho (Figura 2.1.4) e ovos e ninfas brilhantes com cores entre o amarelo e o laranja [Grafton-Cardwell et al. 2005].



Figura 2.10: Pontas das folhas dos citros torcidas devido ao Greening.
Fonte: Foto de Grafton-Cardwell et al. [2005].



Figura 2.11: Inclinação do corpo do psilídeo dos citros adulto.
Fonte: Foto de Grafton-Cardwell et al. [2005].

Os insetos adultos saltam facilmente e são coletados, geralmente, com um aspirador. Os ovos e as ninfas pequenas são difíceis de se ver sem o auxílio de lentes de aumento. Ninfas ficam próximas umas das outras nos brotos e os ovos são colocados dentro de cavidades ou dobras das folhas. Uma atenção especial deve ser dada aos danos causados pelos psilídeos dos citros asiáticos, como: pontas das folhas dos brotos torcidas ou com ondulações, presença de fungos e depósitos brancos de cera sobre as folhas (Figuras 2.1.4 e 2.1.4). Armadilhas adesivas amarelas podem ser usadas como método de detecção do inseto adulto. Porém, essas armadilhas capturam outros insetos que são atraídos pela mesma cor (efeito colateral) e trazem dificuldades ao reconhecimento dos insetos pela presença de cola, gerando efeitos visuais como brilhos (Figura 2.1.4) [Grafton-Cardwell et al. 2005, Leonardo 2014].



Figura 2.12: Secreção, fungos e túbulos de cera produzidos por psilídeos.
Fonte: Foto de M. E. Rogers, disponível em Webb [2010].



Figura 2.13: Grande infestação de psilídeos em jasmim laranja.
Fonte: Foto de Grafton-Cardwell et al. [2005].



Figura 2.14: Armadilha adesiva amarela para a detecção de psilídeos.
Fonte: Foto de Robnett [2014].

O Fundo de Defesa da Citricultura (FUNDECITRUS) realiza frequentemente o monitoramento do HLB no cinturão citrícola paulista. Para isso, armadilhas adesivas amarelas são espalhadas quinzenalmente pelas propriedades citrícolas desta área. Após esse período, as armadilhas são recolhidas e levadas para laboratórios do FUNDECITRUS na cidade de Araraquara (São Paulo). Nos laboratórios, as armadilhas são inspecionadas manualmente, por técnicos treinados, com a finalidade de obter as quantidades de psilídeos presentes. Então, são realizados cálculos e estimadas as populações de psilídeos presentes no cinturão citrícola paulista. Essas informações de monitoramento populacional são cadastradas no sistema de informações de monitoramento do HLB criado pelo FUNDECITRUS [Leonardo 2014].

2.2 Representação e Descrição de Características em Imagens

Um dos mais poderosos sistemas de percepção da natureza é composto pelos olhos e pelo cérebro. Esse sistema de percepção traz benefícios no auxílio da resolução de muitas tarefas sofisticadas da vida real. Objetivando equipar as máquinas com essas habilidades, pesquisas científicas são constantemente realizadas. Como resultados dessas pesquisas, foi possível identificar objetos em uma imagem e encontrar as partes mais descritivas dela. Então, algoritmos foram criados na tentativa de encontrar características descritivas em uma imagem para várias tarefas de visão computacional tais como o reconhecimento de objetos ou a união de imagens que compõem a mesma cena. Técnicas foram criadas com a finalidade de detectar pontos significativos em imagens e nas regiões próximas a esses pontos foram descritas informações úteis, tal como a textura. Características em uma imagem (ou características de imagens) podem ser cantos, pontos ou pedaços os quais são frequentemente obtidos por alguma operação de vizinhança [Derntl 2014].

Como exemplo, em um procedimento de correspondência e rastreamento de objetos em imagens os seguintes passos deveriam ser aplicados: a) detectar e extrair as características com a finalidade de encontrar áreas significativas (*patches*) em imagens as quais proveem informação descritiva; b) representar uniforme e descritivamente as características extraídas das imagens para que elas possam ser comparadas; c) descrever a informação contida nessas representações de maneira apropriada (descrição de características); e d) usar a correspondência de características para obter a correspondência entre as características descritas [Derntl 2014].

A representação e a descrição de características são etapas do processamento digital de imagens que, geralmente, acontecem após algum tipo de segmentação de região dentro da imagem. O agregado de pixels resultante de algum processo de segmentação é, em geral, representado e descrito de forma adequada para um futuro processamento computacional. Normalmente, a representação de uma região da imagem envolve duas alternativas: a representação baseada em suas características externas (sua fronteira) ou internas (os pixels que compõem a região). No entanto, a escolha de uma alternativa de representação é apenas parte da tarefa de tornar os dados úteis para um computador. A tarefa seguinte é descrever a região da imagem em conformidade com a representação selecionada. Como exemplo, uma determinada região da imagem pode ser representada pela sua fronteira e ser descrita por características como a sua extensão, a orientação de uma linha reta que cruze seus pontos mais externos e o número de concavidades dessa borda. Quando o foco principal está nas características de forma, a escolha adequada é utilizar uma representação externa. A escolha de uma representação interna se dá quando o foco principal está nas propriedades regionais, como cor e textura. Algumas vezes, pode ser necessário utilizar esses dois tipos de representação combinados. Em todos os casos, as características escolhidas como descritores devem ser insensíveis, quando possível, às

mudanças de escala, translação e rotação, ou seja, devem ser invariantes a essas mudanças ao descreverem uma região da imagem. Essas variações podem afetar a taxa de reconhecimento em aplicações de visão computacional [Gonzalez e Woods 2010].

Os descritores são algoritmos computacionais empregados na descrição de características que possam estar presentes em imagens digitais.

Robustos descritores de características locais de imagem foram desenvolvidos na última década. Esses descritores são insensíveis a pelo menos uma das três importantes variações que podem ocorrer em uma imagem (escala, translação e rotação). Alguns desses descritores são: ORB (*Oriented FAST and Rotated BRIEF*), SIFT (*Scale Invariant Feature Transform*), SURF (*Speeded Up Robust Features*), BRISK (*Binary Robust Invariant Scalable Keypoints*) e FREAK (*Fast Retina Keypoint*) [Lowe 2004, Rublee et al. 2011, Schaeffer 2013]. Cada um desses descritores será apresentado a seguir.

2.2.1 ORB

Rublee et al. [2011] propuseram um descritor binário muito rápido baseado no descritor BRIEF (*Binary Robust Independent Elementary Features*), chamado ORB (*Oriented FAST and Rotated BRIEF*), o qual é invariante a rotação e resistente a ruídos. Os autores demonstraram, através de experimentos, que o ORB é, em duas ordens de magnitude, mais rápido (tempo de execução) que o algoritmo SIFT. A eficiência do ORB foi testada em diversas aplicações do mundo real, incluindo aplicações de detecção de objetos e rastreamento de partes (pedaços) de imagens em dispositivos móveis. A investigação sobre a variância na orientação foi fundamental para a construção do ORB e auxiliou na decorrelação de seus componentes, objetivando obter bom desempenho de descrição em aplicações similares. Um aspecto interessante é que os autores também distribuíram uma implementação do ORB sob a licença BSD (*Berkeley Software Distribution*) para a comunidade científica, através da biblioteca de visão computacional OpenCV [Initiative et al. 2006].

O descritor ORB é construído sobre o detector de pontos-chave FAST (*Features from Accelerated Segment Test*) [Viswanathan 2009]. Além disso, outro descritor de características é usado como base para a construção do descritor ORB, o descritor BRIEF (*Binary Robust Independent Elementary Features*) [Calonder et al. 2010].

O algoritmo FAST, proposto originalmente, implementa um conjunto de testes binários sobre um pedaço (*patch*) de uma imagem, variando o limiar de intensidade entre o pixel central e outros pixels presentes em um anel circular em torno do centro. A medida de canto (*corner*) de Harris e Stephens [1988] tem sido usada para fornecer uma avaliação da intensidade de canto. No ORB a falta de informação de orientação do FAST é complementada com a intensidade de canto (*corner*) de Rosin [1999].

Uma métrica simples e eficaz de orientação de canto (*corner*) é utilizada, a intensidade do centroide. Essa medida assume que a intensidade de um canto (*corner*) de

um pedaço (*patch*) de imagem é obtida a partir do seu centro e o vetor resultante dessa operação pode ser utilizado para determinar uma orientação para um pedaço (*patch*) de imagem. Rosin [1999] define os momentos de um pedaço (*patch*) de uma imagem. Em particular, o momento m_{pq} de um pedaço (*patch*) com intensidade I pode ser computado como [Ruble et al. 2011]:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \quad (2.1)$$

O centroide C pode ser calculado da seguinte maneira [Ruble et al. 2011]:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.2)$$

Através da construção de um vetor que vai do centro do pedaço (*patch*), O , até o centroide, C , pode-se definir a orientação relativa desse pedaço (*patch*) de imagem da seguinte maneira [Grana et al. 2013]:

$$\omega = \arctan 2(m_{01}, m_{10}), \quad (2.3)$$

onde, $\arctan 2$ é a versão de um quadrante de \arctan . Rosin [1999] fala em considerar se o canto (*corner*) é escuro ou claro. Contudo, para o propósito do cálculo de orientação no algoritmo ORB, pode-se ignorar isso, pois as medidas angulares são consistentes, independentemente do tipo de canto (*corner*). Para melhorar a invariância a rotação, certifica-se que os momentos são computados com as posições x e y remanescentes, dentro de uma região circular de raio r . Empiricamente, o raio r é escolhido para ser o tamanho do pedaço (*patch*), assim x e y são percorridos a partir de $[r, -r]$. Quanto mais $|C|$ se aproxima de 0, essa métrica se torna mais instável. Com cantos (*corners*) FAST, descobriu-se que isso raramente acontece [Ruble et al. 2011].

O descritor BRIEF (*Binary Robust Independent Elementary Features*) é uma descrição (cadeia - *string*) binária e é utilizado, no ORB, para representar um pedaço (*patch*) da imagem construída a partir de um conjunto de testes (de intensidade) binários. Considerando um pedaço (*patch*) de imagem suavizado, p , um teste binário τ é definido por [Calonder et al. 2010, Ruble et al. 2011]:

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases}, \quad (2.4)$$

onde $p(x)$ é a intensidade de p em um ponto x . A característica final é definida como um vetor de n testes binários [Grana et al. 2013]:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(p; x_i, y_i) \quad (2.5)$$

As localizações dos pedaços internos (*patches*) para os testes, portanto a escolha do vetor

$$L = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix}, \quad (2.6)$$

influencia na qualidade do próprio descritor [Ruble et al. 2011]. A solução poderia ser uma amostragem de grade baseada em conjuntos de conjuntos levando em consideração a orientação do pedaço (*patch*) de imagem e, em seguida, multiplicar essas localizações com a matriz de rotação. Contudo, através da análise de distribuição dos testes, essa solução traz uma baixa variância e melhora a correlação entre os testes binários (desde que os testes ao longo da orientação de margem estatisticamente produzam resultados semelhantes). Isso afeta fortemente a eficácia do descritor, descrevendo mais a redundância que o carácter distintivo. Para resolver esse problema, Rublee et al. [2011] empregaram um algoritmo de aprendizagem, amostrando testes com sub-janelas de 5 x 5 e essas sub-janelas com 31 x 31 pedaços (*patch*) para um descritor escolhido, rodando cada teste contra todos os pedaços (*patch*) de treinamento. O resultado é um conjunto predefinido de 256 testes chamado de rBRIEF [Grana et al. 2013].

O descritor BRIEF, base para a descrição do ORB, é construído usando um conjunto de testes binários entre pixels, calculados com base em um ponto chave central [Calonder et al. 2010]. A Figura 2.2.1 mostra dois desses testes binários.

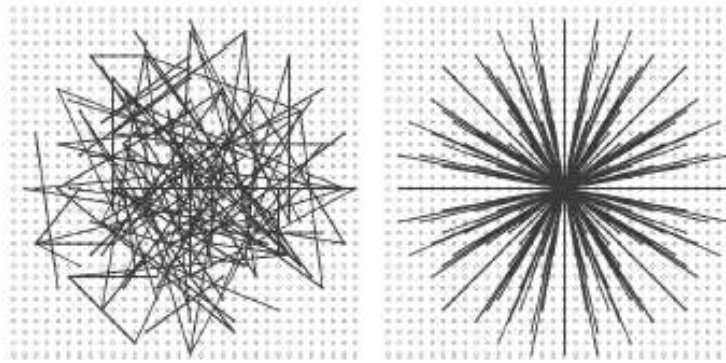


Figura 2.15: Dois testes binários do descritor BRIEF.
Fonte: Figura de Calonder et al. [2010].

2.2.2 SIFT

O algoritmo SIFT (*Scale Invariant Feature Transform*) foi proposto por Lowe et al. [2004] e provê a detecção e a descrição de características (em imagens) que são invariantes a escala, rotação, ruído, mudanças de iluminação, mudanças de ponto de vista em três dimensões (3D) e distorções afim (*affine*).

Na formulação original do algoritmo SIFT, uma região da imagem com escala normalizada é representada com a concatenação de histogramas de orientação de gradientes relativos a várias sub-regiões retangulares. Primeiramente, para obter pedaços (*patches*) de imagem com escala normalizada, um procedimento de detecção de região saliente fornece as vizinhanças de pontos correspondentes a essa região. A função de saliência é a diferença de Gaussianas (DoG - *Difference of Gaussians*) para escalas de espaço. As regiões da imagem (posição e escala) são selecionadas com base nos extremos locais gerados por essa função. Com a finalidade de computar o descritor local, as regiões são escalas normalizadas para computar as derivadas I_x e I_y da imagem I com diferenças de pixel [Moreno et al. 2009]:

$$I_x(x, y) = I(x + 1, y) - I(x - 1, y), \quad (2.7)$$

$$I_y(x, y) = I(x, y + 1) - I(x, y - 1). \quad (2.8)$$

A magnitude de gradiente e a orientação são calculadas para cada pixel na região da imagem [Moreno et al. 2009]:

$$M(x, y) = \sqrt{I_x(x, y)^2 + I_y(x, y)^2}, \quad (2.9)$$

$$\Theta(x, y) = \tan^{-1}\left(\frac{I_y(x, y)}{I_x(x, y)}\right). \quad (2.10)$$

Então, a região de interesse é dividida em sub-regiões em uma grade retangular. A Figura 2.2.2 mostra exemplos da magnitude de gradiente e da orientação para uma região da imagem. Essa região apresenta 16 sub-regiões (4 por eixo) [Lowe 2004].

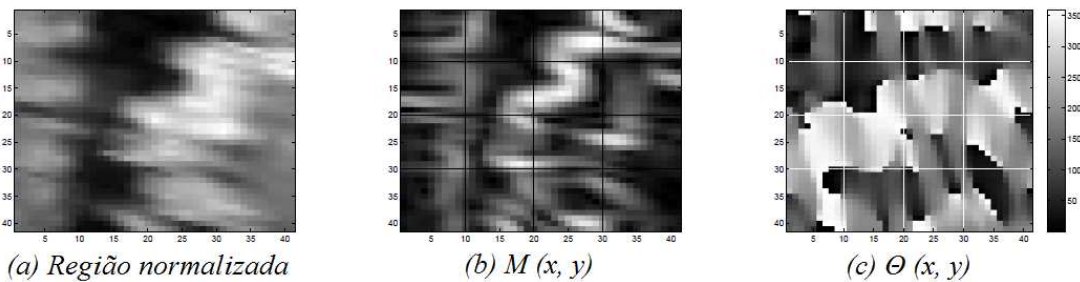


Figura 2.16: SIFT - Exemplo de magnitude de gradiente e orientação em imagem.
Fonte: Figura de Moreno et al. [2009].

O próximo passo é computar o histograma de orientação de gradiente, ponderado por magnitude de gradiente, para a sub-região. A orientação é dividida em 8 caixas

(*bins*) e cada caixa é definida pelo somatório da diferença de orientação da janela para o centro da caixa, ponderado por magnitude de gradiente [Moreno et al. 2009]:

$$h_{r(l,m)}(k) = \sum_{x,y \in r(l,m)} M(x,y) \left(\frac{1 - |\Theta(x,y) - c_k|}{\Delta k} \right), \quad (2.11)$$

$$\Theta(x,y) \in \text{bin } k, \quad (2.12)$$

onde c_k é o centro da orientação da caixa (*bin*), Δk é a largura da orientação da caixa e (x,y) são as coordenadas dos pixels na sub-região $r(l,m)$ [Moreno et al. 2009].

O descritor local SIFT é a concatenação de vários histogramas de orientação de gradiente para todas as sub-regiões [Lowe 2004]:

$$u = (h_{r(1,1)}, \dots, h_{r(l,m)}, \dots, h_{r(4,4)}) \quad (2.13)$$

O passo final é normalizar o descritor para a norma de unidade, objetivando reduzir os efeitos de mudanças de iluminação uniformes. A orientação de gradiente não é invariante a rotações da região da imagem, de modo que o descritor não é invariável. Para prover a invariância a rotação, Lowe [2004] propôs computar a orientação da região da imagem e definir a orientação de gradiente relativa a orientação da região. A orientação é dada através do pico mais alto do histograma de orientação de gradiente da região da imagem [Moreno et al. 2009].

A Figura 2.2.2 mostra, como exemplo, uma matriz do descritor de 2 x 2 (à direita) computada a partir de um conjunto de amostras (matriz) de 8 x 8 (à esquerda). O comprimento de cada seta (direção) (Figura 2.2.2 - Direita) corresponde a soma das magnitudes de gradiente próximas que possui o mesmo sentido [Lowe 2004].

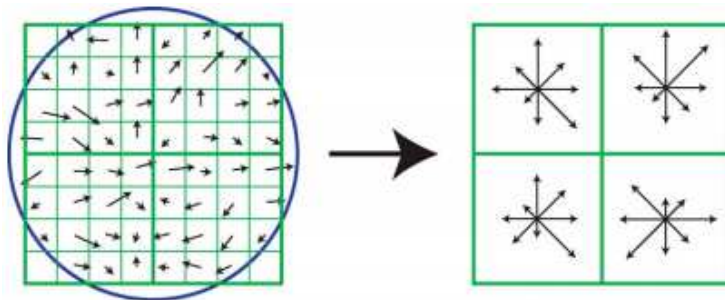


Figura 2.17: SIFT - Gradientes de imagem (esquerda) e descritor de pontos-chave (direita).

Fonte: Figura de Lowe [2004].

2.2.3 SURF

O algoritmo SURF (*Speeded Up Robust Features*) é similar ao algoritmo SIFT e é um detector e descritor de características de imagens invariante a rotação e escala. Além disso, o SURF é muito mais rápido de computar sem perder a robustez, a repetibilidade e a capacidade de distinção [Bay et al. 2006, Bay et al. 2008].

A extração de características utilizando o algoritmo SURF possui duas etapas (sub-algoritmos): a localização (detecção) de pontos-chave e a descrição de características ao redor dos pontos-chave detectados [Bay et al. 2006, Bay et al. 2008].

Na localização de pontos-chave do algoritmo SURF, uma aproximação de uma Matriz Hessiana (*Hessian Matrix*) de segunda ordem é usada para a detecção de pontos-chave, ao redor dos quais, serão descritas as características. Estruturas na forma de manchas (*blobs*) são detectadas onde o determinante da Matriz Hessiana (*Hessian Matrix*) tem seu valor máximo. Um filtro com janela (matriz ou caixa) de tamanho 9 x 9 faz aproximação Gaussiana (*Gaussian*) dos valores contidos nessa janela e é usado para computar mapas de resposta dessas estruturas (*blobs*). Bay et al. [2008] atribuem os nomes de D_{xx} , D_{yy} e D_{xy} para esses mapas. O determinante de uma dada Matriz Hessiana (*Hessian Matrix*) é computado usando o determinante de Hessiano (*Hessian*) e esse determinante representa a estrutura de resposta (*blob*) dentro da imagem na localização X . Os resultados encontrados são armazenados em mapas de resultados de estruturas (*blobs*) em diferentes escalas, onde o local de máximo tenha sido detectado. Então, o espaço de escala é analisado através do aumento de escala (*up-scaling*) do filtro devido ao uso de imagens integrais (*integral images*). Um custo computacional (tempo) muito baixo é obtido através do uso dessas imagens integrais [Pang et al. 2012, Derntl 2014].

Na Figura 2.2.3, filtros de janela (*box*) de 9 x 9 são aproximações de uma função Gaussiana (*Gaussian*) com escala igual a 1,2 e representam a escala mais baixa para a computação dos mapas de resposta das manchas (*blobs*) detectadas. A aproximação por Matriz Hessiana (*Hessian Matrix*) leva-se à utilização de imagens integrais, que é uma técnica muito útil [Agrawal et al. 2008, Ebrahimi e Mayol-Cuevas 2009, Juan e Gwun 2009]. Assim, o tempo de computação é reduzido drasticamente.

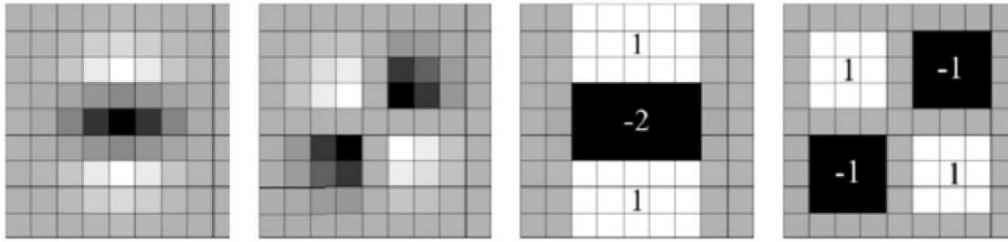


Figura 2.18: Da esquerda para a direita: a derivada parcial Gaussiana de segunda ordem nas direções y e xy , respectivamente; e a aproximação para a derivada parcial Gaussiana de segunda ordem nas direções y e xy . A intensidade da região cinza é zero.

Fonte: Figura de Bay et al. [2008].

A construção de pirâmide de imagens em escalas distintas no algoritmo SURF é similar a construção realizada no algoritmo SIFT. Como é mostrado na Figura 2.2.3, o espaço de escala é dividido em oitavas e existem 4 níveis de escala em cada oitava. Cada oitava representa uma série de mapas de respostas dos filtros obtidos pela convolução da mesma imagem de entrada com um tamanho de filtro crescente. E, a diferença de escala mínima entre as duas escalas subsequentes depende dos lóbulos (*lobes*) de tamanho negativo ou positivo da derivada parcial de segunda ordem na direção de derivação. Por fim, é aplicada a Supressão de Não-máximos (*Non-maximum Suppression*) com uma vizinhança (*neighbourhood*) de $3 \times 3 \times 3$ sobre as imagens e escalas [Pang et al. 2012, Derntl 2014].

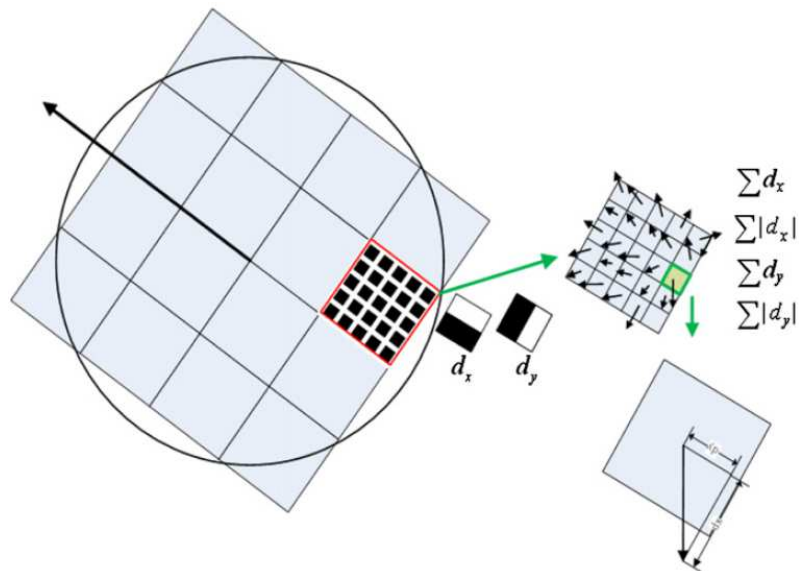


Figura 2.19: Descritor de características SURF-64 - Gradientes de imagem.

Fonte: Figura de Pang et al. [2012].

Com a finalidade de extrair características de imagens invariantes à rotação, respostas de transformada Haar (*Haar wavelet*) são calculadas nas direções x e y das imagens integrais (*integral images*) dentro de uma vizinhança circular de raios $6s$ ao redor de um ponto de característica, s é a escala na qual o ponto de característica foi detectado. As respostas da transformada Haar são representadas como vetores. Então, resume todo o vetor de direção x e y das respostas de transformada Haar dentro de uma janela de orientação deslizante, cobrindo um ângulo de tamanho $\frac{\pi}{3}$ ao redor de um ponto de característica. As duas respostas sumarizadas resultam em um novo vetor. E, o vetor mais longo é a orientação dominante do ponto de característica [Pang et al. 2012].

Para a extração do descritor, constrói-se uma região quadrada com um tamanho de $20s$ e divide-se a região de interesse em 4×4 sub-regiões quadradas e dentro delas 5×5 pontos de amostragem espaçados regularmente [Bay et al. 2008].

Como é mostrado na Figura 2.2.3, realiza-se a computação para obter a resposta da transformada Haar na direção x (d_x) e a resposta da transformada Haar na direção y (d_y). Pondera-se a resposta com um núcleo Gaussiano (*Gaussian kernel*) centralizado no ponto de interesse. E então, sumariza-se a resposta sobre cada sub-região de d_x e d_y separadamente. De modo a gerar informações sobre a polaridade das alterações de intensidade, extrai-se a soma do valor absoluto dessas respostas. Portanto, cada sub-região é formada por um vetor de 4 dimensões:

$$Vec = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (2.14)$$

Finalmente, normaliza-se o vetor em unidade de comprimento para obter a invariância de contraste [Pang et al. 2012].

Concatena-se o vetor do descritor para todas as sub-regiões quadradas de 4×4 , resultando um vetor de características de 64 dimensões. E, computa-se a soma de d_x e $|d_x|$ separadamente para $d_y < 0$ e $|d_y| \geq 0$ (similarmente para d_y e $|d_y|$), resultando num vetor com o dobro de dimensões [Bay et al. 2008].

2.2.4 BRISK

O algoritmo BRISK (*Binary Robust Invariant Scalable Keypoints*) estende o projeto do algoritmo BRIEF adicionando invariância de escala e transformações de rotação a descritores de imagens. O BRISK subdivide-se em dois algoritmos menores: um algoritmo de detecção de pontos-chave e outro algoritmo de descrição de características [Redondi et al. 2013].

O detector de pontos-chave do algoritmo BRISK é uma versão multi-escala do detector popular FAST (*Fast Accelerated Segment Test*) [Viswanathan 2009]. O FAST representou um avanço claro na detecção de cantos de alta velocidade. Ele classifica

um ponto candidato p (com intensidade I_p) como um canto (*corner*) se n pixels contínuos no círculo de Bresenham de raio 3 ao redor de p são todos mais brilhantes que $I_p + t$, ou mais escuros que $I_p - t$, com um limiar t predefinido. Então, para cada canto (*corner*) é atribuída uma pontuação s , definido como o limite máximo ainda classificando p como um canto [Redondi et al. 2013]. Adicionalmente, Leutenegger et al. [2011] apresentaram uma abordagem de aprendizagem de máquina para criar árvores de decisões que permitem classificar um ponto candidato com apenas alguns testes de pixel, acelerando, assim, o processo de detecção. Isso mostrou que o uso dessa abordagem resultou, em média, menos que 2,3 testes por pixels para determinar se é ou não uma característica.

O descritor de características do algoritmo BRISK utiliza um padrão de pontos p_i para amostragem de vizinhança de cada ponto-chave. O padrão define dois conjuntos de amostragem de pares de pontos, ou seja, pares de longa e curta distância. O conjunto de longa distância é composto por todos os pares (i, j) tal que $|p_i - p_j|_2 > \delta_{min}$ e eles são usados para estimar a orientação do ponto-chave pela média de gradiente local. Uma vez que a orientação do ponto-chave é estimada, o padrão de amostragem é rotacionado adequadamente e os pares de curta distância (cuja distância do ponto de amostragem é inferior a um limiar δ_{max}) são usados para construir o descritor. Em primeiro lugar, para cada p_i , um valor de intensidade é obtido por suavização Gaussiana (*Gaussian*) com o desvio padrão σ_i , proporcional à distância entre p_i e o centro do padrão. Para recuperar eficientemente as localizações dos pontos do padrão e informação de suavização, uma tabela para a pesquisa de todas as orientações é construída *offline* e armazenado na memória. Como exemplo, para 1024 ângulos pré-computados, o tamanho da tabela de pesquisa é tão grande quanto 40 Megabytes. Para construir o descritor real, uma cadeia binária é construída concatenando o resultado de todas as comparações de intensidade de pares de pontos de curta distância (p_i, p_j) de tal modo que cada bit b corresponde a [Leutenegger et al. 2011]:

$$b = \begin{cases} 1, & I(p_i, \sigma_i) > I(p_j, \sigma_j) \\ 0, & \text{caso contrario} \end{cases} \quad (2.15)$$

Onde $I(p_i, \sigma_i)$ denota o valor de intensidade suavizada em p_i . Na implementação original do BRISK dos autores Leutenegger et al. [2011], δ_{max} é ajustado de modo que o descritor resultante tem 512 bits [Redondi et al. 2013].

O padrão de amostragem do descritor BRISK é apresentado na Figura 2.2.4 [Leutenegger et al. 2011].

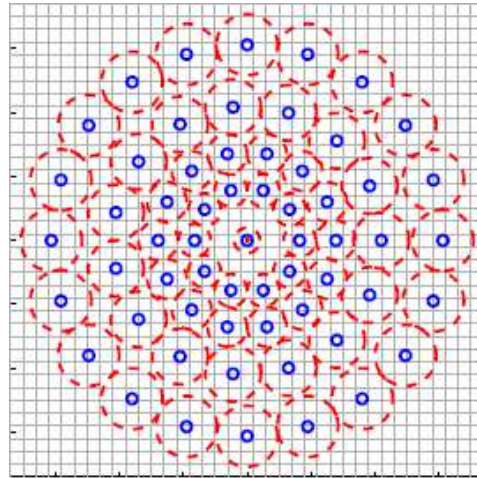


Figura 2.20: Padrão de amostragem do BRISK.
Fonte: Figura de Leutenegger et al. [2011].

2.2.5 FREAK

O algoritmo FREAK (*Fast Retina Keypoint*) tenta imitar matematicamente partes da função de reconhecimento de imagens da visão humana. Para isso, esse algoritmo proposto por Alahi et al. [2012] detecta pontos-chave em uma amostragem da retina que é circular. Além disso, os pontos mais centrais dessa amostragem têm uma densidade maior. A quantidade desses pontos diminui exponencialmente a medida que essas detecções se aproximam das bordas do círculo de amostragem. A diferença de tamanhos de núcleo Gaussiano (*Gaussian kernel*) é usada para cada ponto-chave amostrado. Foi empiricamente observado que o tamanho dos núcleos (*kernels*) que correspondem ao padrão de retina *log-polar* e à sobreposição de campos receptivos aumenta o desempenho desse descritor de características. Um poder de discriminação maior é alcançado através do aumento da redundância desses descritores [Derntl 2014].

O FREAK possui apenas o algoritmo de descrição de características e não possui a etapa de detecção, sendo associado a outros detectores no momento de sua utilização [Alahi et al. 2012].

A descrição no algoritmo FREAK pode ser dividida em duas partes: uma etapa de refinamento e outra de orientação dos descritores [Derntl 2014].

Na etapa de refinamento, um descritor binário F é gerado através da limiarização (*thresholding*) da diferença entre pares de campos receptivos com um núcleo Gaussiano (*Gaussian*) correspondente. F é uma cadeia (*string*) de caracteres binários formada por uma sequência de bits de valor 1 que codifica diferenças em função Gaussiana (DoG - *Difference of Gaussian*). A diferença de Gaussiana é dada pela fórmula a seguir [Derntl 2014]:

$$F = \sum_{0 \leq a \leq N} 2_a T(P_a) \quad (2.16)$$

Onde, P_a é um par de campos receptivos, N é o tamanho projetado do descritor e $T(P_a)$ é dada pela seguinte fórmula [Derntl 2014]:

$$T(P_a) = \left\{ \begin{array}{l} 1, \text{ se } (I(P_a^{r_1}) - I(P_a^{r_2})) > 0, \\ 0, \text{ caso contrario} \end{array} \right\} \quad (2.17)$$

Onde, $I(P_a^{r_1})$ é a intensidade suavizada (*smoothed intensity*) do primeiro campo receptivo do par P_a . Para um melhor desempenho computacional, os pares que não descrevem a imagem de uma forma suficientemente eficiente têm que ser removidos. Portanto, os passos para aprender os melhores pares a partir dos dados de treinamento é sempre mantendo a melhor coluna (média de 0,5) e interativamente adicionar as colunas restantes com baixa correlação com as colunas selecionadas [Alahi et al. 2012].

Um padrão simétrico é capturado devido à orientação do padrão ao longo do gradiente global. Os primeiros grupos de descritores envolvem principalmente campos receptivos periféricos enquanto que os últimos desses grupos implicam em campos mais centralizados. Esse comportamento é muito próximo ao comportamento que o olho humano utiliza [Alahi et al. 2012].

Na etapa de orientação é estimada a rotação dos pontos-chave. Os autores sugerem sumarizar os gradientes locais estimados para todos os pares selecionados. Essa orientação é dada pela seguinte fórmula [Derntl 2014]:

$$O = \frac{1}{M} \sum_{P_o \in G} (I(P_o^{r_1}) - I(P_o^{r_2})) \frac{P_o^{r_1} - P_o^{r_2}}{\|P_o^{r_1} - P_o^{r_2}\|} \quad (2.18)$$

Onde, G é o conjunto de todos os pares usados para computar os gradientes locais, M é o número de pares em G e $P_o^{r_i}$ é um vetor de duas dimensões (2D) contendo coordenadas espaciais do centro do campo receptivo [Alahi et al. 2012].

O padrão de amostragem do algoritmo FREAK é ilustrado da Figura 2.2.5 [Alahi et al. 2012].

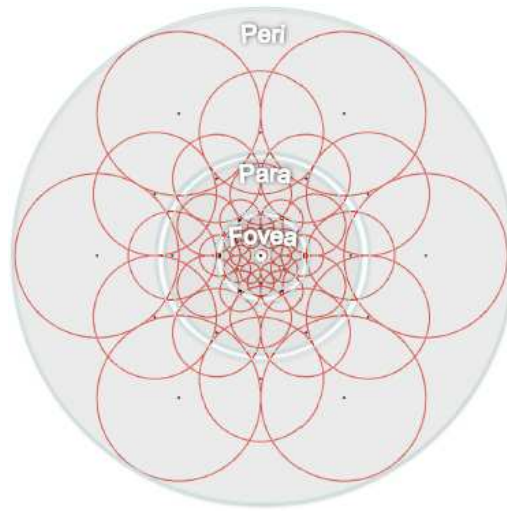


Figura 2.21: Padrão de amostragem do FREAK.
Fonte: Figura de Alahi et al. [2012].

2.3 Bolsas de Características

Sistemas de Recuperação de Imagens Baseados em Conteúdo (CBIR - *Content-Based Image Retrieval*) necessitam que os usuários consultem as imagens por meio do seu conteúdo visual de baixo nível. Isso não só é difícil para os usuários ao realizar a formulação de consultas, mas também pode levar a resultados de recuperação não satisfatórios. Por causa disso, a anotação de imagem foi proposta. O objetivo dessa anotação é atribuir automaticamente palavras-chave para as imagens, de tal forma que os usuários sejam capazes de consultar as imagens através desses códigos [Bosch et al. 2007].

A anotação de imagem pode ser considerada como um problema de classificação de imagem. Nesse processo de classificação, as imagens são representadas por algumas características de baixo nível e, em seguida, técnicas de aprendizagem de máquina podem ser usadas para aprender um mapeamento entre as características de baixo nível e os conceitos de alto nível. Um dos métodos de anotação de características de imagem utilizados é a Bolsa de Palavras (Visuais) (BoW - *Bag-of-Words*) [Tsai 2012]. O método BoW foi primeiramente proposto em problema do domínio da recuperação de texto e mais tarde foi adaptado para aplicações de visão computacional [Bosch et al. 2007]. Esse método também é conhecido como Bolsa de Características (BoF - *Bag-of-Features*), uma vez que o termo “palavra” está associado a representação de uma parte da imagem (uma característica) [Hackeling 2014].

No modelo BoF, o conceito de “palavras visuais” se assemelha ao conceito de “palavra” que é usado no modelo BoW. Essas palavras visuais (características visuais) são geradas por processo de quantização vetorial que utiliza agrupamento (*Cluste-*

ring) de características visuais de baixo nível extraídas de pontos ou regiões locais. Exemplos dessas características visuais são: cores e texturas [Tsai 2012].

O processo de geração e representação de características visuais no modelo BoF possui os seguintes passos: (i) detecção automática de regiões ou pontos de interesse; (ii) computação de descritores locais sobre essas regiões ou pontos de interesse; (iii) quantização dos descritores agrupando-os em palavras visuais para formar o vocabulário visual (formação do livro de código, *Codebook*); e (iv) contagem das ocorrências na imagem de cada palavra visual específica do vocabulário visual para a construção do vetor de características BoF (ou um histograma de frequência de palavras visuais) [Bosch et al. 2007]. A Figura 2.3 mostra esses quatro passos do processo de geração e representação de características BoF a partir de imagens.

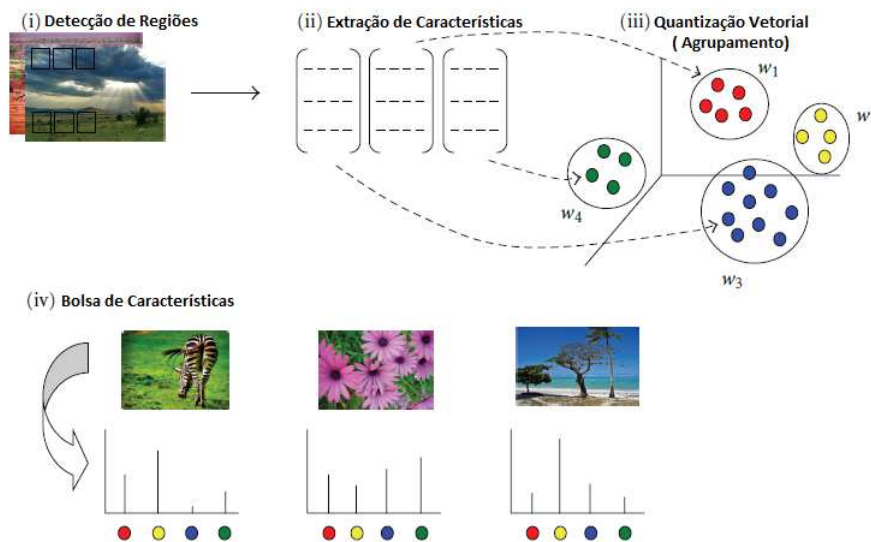


Figura 2.22: Quatro passos para a construção de BoF para a representação de imagens.

Fonte: Figura de Tsai [2012].

O modelo BoF pode ser definido da seguinte maneira: dado um conjunto de treinamento D contendo n imagens representadas por $D = d_1, d_2, \dots, d_n$, onde d é uma característica visual extraída, um algoritmo específico de aprendizagem de máquina não supervisionada, tal como *K-Means*, é usado para agrupar D baseado em um número fixo de palavras visuais W (ou categorias) representado por $W = w_1, w_2, \dots, w_v$, em que V é o número de grupos. Então, pode-se resumir os dados em uma tabela de $V \times N$ ocorrências de contagens $N_{ij} = n(w_i, d_j)$, onde $n(w_i, d_j)$ denota com que frequência a palavra visual w_i ocorre em uma imagem d_j [Tsai 2012].

2.4 Aprendizagem de Máquina

O campo de pesquisa da Aprendizagem de Máquina tem por objetivo principal criar algoritmos que aprendem como executar uma função sem serem explicitamente programados para isso e prover conhecimento para possibilitar projetos de *softwares* que usam a experiência prévia para tomar decisões futuras, ou seja, o estudo de programas que aprendem a partir dos dados. Ao mesmo tempo que a Aprendizagem de Máquina é uma disciplina relativamente nova, também são encontrados muitos exemplos práticos de sua utilização. Programas de filtragem inteligente de e-mails não desejados (filtros de *spam*), termostatos com regulação de temperatura automatizada e personalizada, programas que utilizam imagens para reconhecimento de textos escritos a mão, programas detectores de fraudes em transações com cartão de crédito, programas de predição de preços de imóveis e programas de classificação de tumores em imagens são alguns exemplos de aplicação das técnicas dessa área do conhecimento [Hackeling 2014]. O campo da Aprendizagem de Máquina está situado como uma subárea da Inteligência Artificial [Garreta e Moncecchi 2013].

Existem três tipos principais de algoritmos de Aprendizagem de Máquina: a aprendizagem não supervisionada, a aprendizagem por reforço e a aprendizagem supervisionada. Esses tipos são identificados com base no seu comportamento, seus dados de entrada e suas informações de saída [Hackeling 2014].

Na aprendizagem não supervisionada, um agente inteligente, ou programa de computador, aprende padrões a partir dos dados de entrada embora não sejam fornecidas, previamente, informações de saída para uma comparação do desempenho do sistema e adequação das funções de aprendizagem. O agrupamento é a tarefa mais comum desse tipo de aprendizagem [Russel e Norvig 2013].

Na aprendizagem por reforço, a aprendizagem se baseia em estratégias de séries de reforços que podem ser estímulos positivos ou negativos, ou seja, recompensas ou punições [Russel e Norvig 2013].

Na aprendizagem supervisionada, um agente inteligente observa alguns exemplos de dados de entrada e informações de saída e, a partir dessa observação, aprende uma função que faz o mapeamento entre essas entradas e saídas. Assim, quando são fornecidas entradas de dados com saídas desconhecidas essa função de mapeamento é capaz de prever as informações de saída [Russel e Norvig 2013].

Existe um outro tipo mais amplo de Aprendizagem de Máquina que se caracteriza pela utilização de algoritmos tanto de aprendizagem supervisionada quanto de não supervisionada, a aprendizagem semissupervisionada. A aprendizagem por reforço pode ser considerada uma subárea da aprendizagem semissupervisionada [Hackeling 2014, Russel e Norvig 2013].

Duas das tarefas mais comuns da aprendizagem de máquina supervisionada são a classificação e a regressão. Nas tarefas de classificação, o programa deve aprender a prever valores de saída discretos, ou seja, o resultado da predição são valores

dentro de categorias (rótulos ou classes) predefinidas. Em problemas de regressão, o programa deve aprender a prever valores de saída contínuos, isto é, as informações de saída geralmente podem ser qualquer valor dentro do conjunto dos números reais. No caso da aprendizagem de máquina não supervisionada algumas tarefas comuns são o agrupamento e a redução de dimensionalidade das características fornecidas ou os dados de entrada [Hackeling 2014]. O agrupamento consiste na detecção de grupos potencialmente úteis a partir dos dados de entrada [Russel e Norvig 2013].

Quando o número de características ou dados de entrada é alto, isso pode afetar o desempenho dos sistemas de Aprendizagem de Máquina. A redução de dimensionalidade é utilizada para diminuir o número de características de entrada tentando não reduzir o desempenho de regressão ou classificação no programa. Além disso, essa tarefa de redução é utilizada para tornar possível a visualização de dados, transformando dados de alta em baixa dimensionalidade [Hackeling 2014].

Em problemas de aprendizagem de máquina supervisionada, é comum existir uma divisão das amostras em 2 conjuntos: um conjunto de dados de treinamento e um conjunto de dados de teste. Quando um modelo de aprendizagem memoriza o conjunto de treinamento, mas não prevê bem o conjunto de dados de testes, dizemos que o modelo está superajustado, ou seja, não generaliza bem para novos conjuntos de amostras. Para testes mais precisos sobre a generalização de um modelo, muitas vezes é recomendada a divisão do conjunto de dados em 3 partes: um conjunto de treinamento, um conjunto de validação e um conjunto de testes final. Nessa abordagem, o conjunto de treinamento e o conjunto de validação são utilizados para treinamento e validação do modelo de aprendizagem utilizando sucessivas divisões distintas entre eles (validação cruzada). Uma vez que o modelo de aprendizagem foi selecionado, esses dois conjuntos são combinados para a realização de um último treinamento e o conjunto de teste final é utilizado para testar a real generalização do modelo aprendido [Hackeling 2014].

Russel e Norvig [2013] afirmam que os números de iterações em processos de validação cruzada comumente utilizados são 5 e 10. Esses valores possibilitam divisões do banco de amostras em 5 e 10 partes e apenas uma dessas partes é reservada para testes a cada iteração. Então, podemos observar divisões comuns do banco de amostras em percentuais de 80% para treinamento e 20% para testes ou 90% para treinamento e 10% para testes, respectivamente. Considerando, por exemplo, 10 iterações de validação cruzada em divisões do banco de amostras com 3 partes, há uma divisão proporcional do número de amostras para que se tenham sempre os mesmos percentuais de treinamento e testes em todo processo. No exemplo de utilização de validação cruzada com 10 divisões (e iterações), 90% das amostras iniciais é destinado ao conjunto de desenvolvimento (conjunto de treinamento e conjunto de validação) e 10% é reservado a um conjunto de testes final. Dentro do conjunto de desenvolvimento é realizada uma nova divisão proporcional de 90% para o conjunto de treinamento e 10% para o conjunto de validação [Scikit-learn 2014b, James et al. 2013].

Muitas métricas podem ser utilizadas para mensurar se um programa está ou não

aprendendo a realizar suas tarefas eficientemente. Para problemas de aprendizagem de máquina supervisionada, muitas medidas de desempenho mensuram o número de erros de predição. Existem duas causas fundamentais de erros de predição: o viés (tendência) de um modelo e sua variância. Um modelo com alto viés é inflexível, mas um modelo com alta variância pode ser tão flexível ao ponto de modelar ruídos em um conjunto de treinamento. Isto é, um modelo com alta variância super-ajusta (*Over-fitting*) os dados de treinamento e um modelo com alto viés sub-ajusta (*Under-fitting*) esse conjunto de dados. Idealmente, um modelo deve ter baixo viés e baixa variância, mas os esforços de diminuir um aumenta o outro. Isso é conhecido como a ponderação (*trade-off*) entre viés e variância [Garreta e Moncecchi 2013].

Em muitas aplicações de classificação, o custo associado com todos os tipos de erros podem ser o mesmo e a acurácia é uma das medidas de desempenho mais comuns em problemas de classificação. O cálculo da acurácia é dado pela fórmula [Hackeling 2014]:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.19)$$

Onde TP (*True Positive*) representa o número de classificações corretas para a classe positiva (verdadeiro-positivo), TN (*True Negative*) é o número de classificações corretas para a classe negativa (verdadeiro-negativo), FP (*False Positive*) é o número de classificações incorretas para a classe negativa (falso-positivo) e FN (*False Negative*) é o número de classificações incorretas para a classe positiva (falso-negativo).

Outras métricas mais específicas são a precisão (*Precision*) e a sensibilidade (*Recall*). A primeira mede o número de amostras verdadeiras positivas identificadas sobre o número total de predições positivas e a segunda mede o número de amostras verdadeiras positivas identificadas sobre o número total de amostras verdadeiras positivas existentes no conjunto predito. A precisão está relacionada com a taxa de falsos positivos e a sensibilidade com a taxa de falsos negativos do sistema. Quanto maior é a taxa de precisão menor é o número de falsos positivos e quanto maior é a taxa de sensibilidade menor é o número de falsos negativos. O cálculo da precisão é exibido abaixo [Hackeling 2014]:

$$P = \frac{TP}{TP + FP} \quad (2.20)$$

O cálculo da sensibilidade é dado pela seguinte fórmula [Hackeling 2014]:

$$R = \frac{TP}{TP + FN} \quad (2.21)$$

Uma média harmônica entre a precisão e a sensibilidade pode ser calculada, conhecida pelo nome de *F1-score*. Essa medida tenta combinar a precisão e

sensibilidade em uma única taxa. O cálculo dessa média é dado pela fórmula [Garreta e Moncecchi 2013]:

$$F1_{score} = \frac{2 * P * R}{P + R} \quad (2.22)$$

2.4.1 Agrupamento (K-Means e Mini Batch K-Means)

O agrupamento (*clustering*) é um tipo de aprendizagem de máquina não supervisionada que forma automaticamente grupos (*clusters*) de coisas similares. O *K-Means* é um tipo de algoritmo de agrupamento. Esse algoritmo encontra k grupos para um dado conjunto de dados. O parâmetro de número de grupos k é definido pelo usuário. Cada grupo é descrito por um ponto simples conhecido como centroide. No centro de todos os pontos de um grupo está o seu centroide. O algoritmo *K-Means* funciona da seguinte maneira: os k centroides são randomicamente atribuídos a pontos. Em seguida, cada ponto do conjunto de dados é atribuído a um grupo. Essa atribuição é feita encontrando-se o centroide mais próximo e associando um determinado ponto ao grupo daquele centroide. Em seguida, os centroides são todos atualizados pelo valor médio de todos os pontos daquele grupo. Esse procedimento iterativo é realizado até que não existam mais pontos que mudem de grupo entre uma iteração. Uma vantagem do algoritmo *K-Means* é a sua fácil implementação. Algumas desvantagens são que esse algoritmo pode convergir para um mínimo local e, em conjuntos de dados grandes, ele se torna lento. O *K-Means* trabalha apenas com valores numéricos [Harrington 2012].

O *Mini Batch K-Means* é uma variação do algoritmo *K-Means* que usa amostragem randômica de instâncias em cada iteração. O *Mini Batch K-Means* converge mais rapidamente para seus agrupamentos, mas suas distorções nos grupos aprendidos podem ser maiores que no *K-Means*. Na prática, os resultados são similares e essas distorções são aceitáveis quando a velocidade de execução do algoritmo é um importante requisito do sistema de visão computacional projetado [Hackeling 2014, Scikit-learn 2014a].

2.4.2 K Vizinhos Mais Próximos

O K Vizinhos Mais Próximos (KNN - *K Nearest Neighbors*) é um algoritmo de aprendizagem de máquina supervisionada e funciona da seguinte maneira: dado um conjunto de amostras de treinamento existente (conjunto de treinamento) é sabida a classe (rótulo) de cada uma dessas amostras. Quando uma nova amostra com rótulo desconhecido é inserida no algoritmo, essa nova amostra é comparada com cada amostra de dados existente. Então, obtém-se a parte das amostras mais similar (vizinhos mais próximos) a essa amostra desconhecida e verificam-se os seus rótulos. Essa parte mais similar possui k amostras. Finalmente, realiza-se a contagem de

votos (rótulos) das k amostras e o rótulo votado majoritariamente é selecionado como a nova classe para a amostra não rotulada. O parâmetro k é um valor inteiro usualmente menor que 20 [Harrington 2012].

As vantagens do algoritmo K Vizinhos Mais Próximos são: a alta acurácia de classificação, indiferença a valores extremos (*outliers*) e não calcular funções de hipótese sobre os dados. Suas desvantagens são: o alto custo computacional e requerer grandes quantidades de memória. Esse algoritmo trabalha com valores numéricos e valores nominais [Harrington 2012].

2.4.3 Máquinas de Vetores de Suporte

Máquinas de Vetores de Suporte (SVMs - *Support Vector Machines*) é o nome atribuído a um conjunto de algoritmos de aprendizagem de máquina, podendo ser usados em aprendizagem supervisionada. Após a etapa de treinamento, esses algoritmos realizam boas previsões para os dados desconhecidos e seus resultados, geralmente, possuem pequenas taxas de erro [Harrington 2012].

Algoritmos de SVMs utilizam o conceito de Hiperplano de Separação (*Separating Hyperplane*) e funções de separação de dados que são chamadas de núcleo (*Kernel*). Essas funções podem ser lineares, polinomiais ou baseadas em raio (RBF - *Radius Basis Function*). Quando os dados são bidimensionais e o núcleo de um algoritmo de SVMs é linear, precisa-se de uma reta para separar o conjunto de dados. Se os dados são tridimensionais, precisa-se de um plano. E, em altas dimensões, precisa-se de algo denominado Hiperplano (*Hyperplane*) [Harrington 2012].

O Hiperplano é a fronteira de decisão (classificação). Todos os dados de um lado do Hiperplano pertencem a uma classe e todos os dados do outro lado à outra. Um algoritmo de SVMs possui o conceito de Margem (*Margin*) que se caracteriza pela distância do Hiperplano de Separação aos pontos de dados mais próximos, em ambos os lados dessa fronteira. A Margem ideal é aquela que maximiza a distância do Hiperplano aos pontos mais próximos, igualmente nos dois lados dessa separação. Para que isso seja possível, são usados Vetores de Suporte (*Support Vectors*) que consistem nesses pontos mais próximos. Esses algoritmos possuem uma constante C , definida pelo usuário, que controla uma ponderação entre o objetivo de ter a maior Margem possível e assegurar que a maior parte das amostras (pontos de dados) tenha uma Margem funcional. Essa margem é a área formada entre o Hiperplano de separação e os vetores de suporte [Harrington 2012].

As vantagens dos algoritmos de Máquinas de Vetores de Suporte são: o pequeno erro de generalização, o baixo custo computacional e a facilidade em interpretar seus resultados. Suas desvantagens são: a sensibilidade de resultados de classificação a ajustes de parâmetros e seleção de núcleo e, nativamente, a capacidade de realizar apenas classificações binárias. Esse algoritmo trabalha com valores numéricos e valores nominais [Harrington 2012].

2.5 Trabalhos Relacionados

No trabalho de Larios et al. [2008] foi utilizada uma abordagem de bolsa de características para automatizar a identificação taxonômica de larvas da ordem Plecoptera. Foram coletadas 263 dessas larvas em riachos de Willamette e Cascade Range, próximo a Oregon nos Estados Unidos. Aproximadamente 10 imagens foram obtidas para cada espécie. Essas imagens foram examinadas manualmente e todas as imagens que proporcionavam uma visão dorsal dos insetos, dentro de 30 graus verticais, foram selecionadas para análise. As imagens foram classificadas através de um processo que envolve: identificação de regiões de interesse, representação dessas características como vetores SIFT (*Scale-Invariant Feature Transform*), transformação dos vetores SIFT em um histograma de características detectadas e a classificação do histograma por um conjunto de árvores de modelo logístico. Foram aplicados três algoritmos de detecção de regiões: Hessian-affine, Kadir e PCBR (*Principal Curvature-Based Region*). A construção do livro de código (*Codebook*) foi realizada através de GMM (*Gaussian Mixture Model*). Os autores afirmam que seu detector PCBR supera os outros dois descritores e mostram uma acurácia de classificação de 82% para quatro classes e 95% para três classes.

No trabalho de Shiguo et al. [2009] foi apresentado um conjunto de abordagens baseadas em ontologias para o reconhecimento de insetos em imagens. Esse conjunto de abordagens incluiu etapas como: uma camada de treinamento, uma camada de ontologia e uma camada de reconhecimento. Além disso, incluiu métodos para a comunicação entre essas camadas. A camada de treinamento continha um banco com as amostras de imagens dos insetos e outro com os tipos e as características desses insetos. A camada de ontologia continha as ontologias das características visuais, das características de mídia, do módulo de classificação, da morfologia e da taxonomia dos insetos.

No trabalho de Xiao-Lin et al. [2009] foi usada a regressão espectral com LDA (*Linear Discriminant Analysis*) para reduzir espaços de alta dimensionalidade de imagens de insetos e obter um subespaço de características dos insetos estudados. Em seguida, o vetor de características, gerado a partir das características de subespaço, foi usado como entrada para o algoritmo de aprendizagem de máquina KNN. Assim, imagens de insetos não rotuladas foram classificadas. O percentual médio de acurácia foi de 90%. Essa abordagem de classificação se mostrou superior a utilização de PCA (*Principal Component Analysis*) e matriz comprimento de corrida (*Run Length Matrix*).

No trabalho de Le-Qing e Zhen [2010] foi apresentado um método para classificar insetos em imagens, considerando as regiões das asas desses insetos, por meio da análise do histograma de cor e GLCM (*Gray-Level Co-occurrence Matrices*). A imagem da asa de insetos da ordem Lepidoptera foi pré-processada para se obter a região de interesse (ROI - *Region of Interest*). Em seguida, a imagem colorida foi convertida de um espaço de cor no modelo RGB (*Red-Green-Blue*) para o espaço de

cor HSV (*Hue-Saturation-Value*) e os histogramas de cor em uma dimensão (1D) de regiões de interesse foram gerados a partir de distribuições de tom (*Hue*) e saturação (*Saturation*). Em seguida, a imagem colorida foi convertida para tons em escala de cinza, rotacionada e transformada para uma posição padrão e foram extraídas suas características GLCM. A correspondência foi primeiramente calculada pela correlação dos vetores (histogramas) entre imagens de exemplo e imagens de teste. Se a correlação fosse mais elevada do que um determinado limiar, então as características GLCM das imagens comparadas correspondiam. A política *winner-take-all* foi adotada na decisão de espécies mais similares dentro dos resultados do KNN (*K Nearest Neighbors*). Foi utilizado um banco de imagens com 100 espécies de Lepidoptera. O percentual de reconhecimento foi de 71,1%.

No trabalho de Zhang e Hu [2010] foi projetado um classificador de imagens para o reconhecimento de insetos de grãos estocados. Essa classificação utilizou várias características de imagens como parâmetros de entrada e o grau de mistura entre as características de várias espécies de insetos foi considerável. A teoria de extensão foi proposta como alternativa eficaz de classificação automática de insetos. O método que construiu a matriz de elemento dos insetos foi apresentado com base na média e variância das características das imagens. Após calcular o grau de correlação entre o inseto a ser reconhecido e nove espécies de insetos, o inseto pôde ser reconhecido pelo critério do grau de correlação integrado máximo. A experiência confirma que o reconhecimento dos insetos com base na teoria de extensão é prático e viável pelo treinamento e análise das amostras dos insetos. Esse método alcançou percentuais de acurácia de 93,33%.

No trabalho de Yang et al. [2010] foi apresentado um método de reconhecimento de imagens de insetos utilizando tecnologias de visão computacional. Primeiramente, foram extraídas 14 características de cada imagem para algumas espécies de insetos. Essas características foram: a retangularidade, alongamento, circularidade, excentricidade, esfericidade, compacidade, quantidade de núcleos interconectados dentro uma região de interesse (*lobation*) e os sete momentos invariantes de Hu. Como meio de classificação, um algoritmo de aprendizagem de máquina chamado de Árvores Randômicas foi empregado. O classificador foi treinado com as características extraídas e o resultado do treinamento foi salvo em um banco de dados. Em seguida, os dados puderam ser carregados a partir dessa base de dados para o algoritmo e, assim, o algoritmo pôde ser utilizado para reconhecer as espécies de insetos para as quais ele foi treinado. A fim de implementar esses algoritmos, módulos de *software* foram desenvolvidos para: a extração de características, treinamento de algoritmo de aprendizagem de máquina e realização de etapas de reconhecimento. Na implementação desses módulos, funções da biblioteca de código-fonte aberto chamada OpenCV foram utilizadas. O percentual médio de reconhecimento foi de 97,14%.

No trabalho de Larios et al. [2011] foi proposta a melhoria de um método de reconhecimento de classes de objetos em imagens que foi demonstrado na classificação de espécies de insetos para fins de monitoramento biológico e determinação da qualidade do meio em que esses insetos estavam inseridos. Essa nova abordagem de

empilhamento de características (*stacking*) quantificou, de forma eficiente, os dados de entrada e aumentou a acurácia de classificação, ao mesmo tempo em que permitiu o uso de informação espacial. Foi utilizado um banco com 4722 imagens classificadas em 29 espécies comumente presentes em água corrente, muitas delas de difícil distinção. Os insetos estavam em diferentes posições do espaço tridimensional, em diferentes rotações e em diferentes estágios de degradação e desenvolvimento, com larga variação de características das amostras dentro das classes. Esta pesquisa mostrou os benefícios do uso de informações espaciais no âmbito de empilhamento com vários tipos diferentes de características de objetos em imagens. O percentual médio de acurácia foi de 88,06%.

No trabalho de Samanta e Ghost [2012] foi criado um sistema para a classificação de insetos usando CFS (*Correlation-based Feature Selection*) e IBPLN (*Incremental Back Propagation Learning Network*). Foi criado um banco de imagens concentrando-se sobre oito maiores pragas (insetos) registradas em diferentes jardins de chá de distritos da Índia. O banco possui 609 imagens pertencentes a oito classes descritas por 11 atributos. A classificação foi realizada utilizando redes neurais artificiais. O estudo demonstrou que o CFS pode ser usado para a redução do vetor de características e a combinação do CFS e IBPLN pode ser usada em outros problemas de classificação.

No trabalho de Lu et al. [2012] foi proposta uma abordagem híbrida chamada DLSoft (*Discriminative Local Soft Coding*), que combinou estratégias de codificação locais e discriminativas juntas. Esse método usou palavras de código próximas para obter codificação leve local e classes de palavras de código específicas para uma representação discriminativa. Na obtenção de uma representação vetorial de imagem via *spatial pyramid pooling*, uma SVM linear foi usada para classificar as imagens dentre as espécies treinadas. Os resultados experimentais mostraram que o método proposto teve um percentual médio de acurácia de 68,80% e superou os métodos do estado da arte quanto a classificação de objetos genéricos. .

No trabalho de Mundada e Gohokar [2013] foi criado um sistema para detectar moscas brancas e pulgões em culturas infectadas, dentro de estufas. Imagens da folha infectada foram capturadas por câmeras e processadas previamente usando técnicas de processamento de imagens tais como conversão de espaço de cores baseado no modelo RGB (*Red-Green-Blue*) para espaço de cores em escala de cinza e filtragem para melhoria das imagens. Na extração de características foram consideradas uma variedade de propriedades de região e de matriz de covariância cinza. Alguns exemplos de propriedades extraídas foram: entropia, média, desvio padrão, contraste, energia, correlação e excentricidade. A classificação foi realizada com o uso de SVM. Os autores afirmam que o protótipo do sistema proveu detecção rápida de pragas e teve o mesmo nível de desempenho que uma abordagem manual clássica. A tarefa de classificar imagens de folhas como “infectada” ou “não infectada” resultou em acurácia de 100%.

No trabalho de Venugoban e Ramanan [2014] foi criado um sistema de classificação

de imagens de pragas (insetos) presentes em plantações de arroz. Foram usadas características baseadas em gradientes através da abordagem de bolsa de características. Imagens de 20 classes desses insetos foram obtidas a partir do Google e fotografias cedidas pela Faculdade de Agricultura, Universidade de Jafna, Sri Lanka. As imagens foram classificadas por um sistema que envolveu identificação de regiões de interesse, representação dessas regiões como descritores SURF ou SIFT, criação de Codebooks e classificação dessas características com SVM linear. Além disso, características HOG (*Histogram of Oriented Gradient*) foram extraídas e classificadas. O algoritmo *Nearest Neighbor* foi utilizado na comparação com o algoritmo SVM. Os resultados mostraram que o descritor HOG superou os descritores SIFT e SURF. A combinação de SURF e HOG, quando classificados com SVM linear, resultou em aproximadamente 90% de acurácia.

Capítulo 3

Métodos e Experimentos

“O método científico é comprovado e verdadeiro. Não é perfeito, é apenas o melhor que temos. Abandoná-lo, junto com seus protocolos céticos, é o caminho para uma idade das trevas.”

– Carl Sagan

O desenvolvimento do trabalho foi dividido em 5 etapas: a “Construção e Validação do Banco de Imagens”, a “Rotulagem de Amostras e Extração de Características”, a “Criação dos Livros de Código e das Bolsas de Características”, “Classificação das Amostras” e a “Aprimoramento dos Classificadores”, conforme apresentado na Figura 3.1. Nas subseções a seguir são descritas cada uma destas etapas.

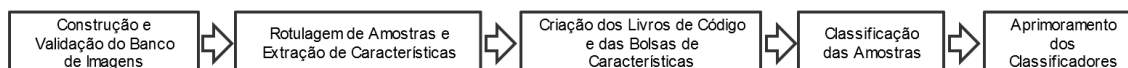


Figura 3.1: Etapas da metodologia utilizada.

Os procedimentos computacionais deste trabalho de pesquisa, que serão mencionados nas seções a seguir, foram todos implementados na linguagem de programação *python*.

3.1 Construção e Validação do Banco de Imagens

A aquisição das imagens foi realizada pelo Fundo de Defesa da Citricultura (FUNDECITRUS), instituição de pesquisa, que, no período entre janeiro e agosto de

2015, coletou mais de 700 armadilhas adesivas amarelas que estavam posicionadas em diversas regiões citrícolas do estado de São Paulo, Brasil.

Em laboratório, com a utilização de uma câmera fotográfica (Olympus SC30) acoplada a um microscópio (Olympus SZ61) e luminária de led (LED 72T) (Figura 3.2), foram reunidas 576 imagens do *Diaphorina citri* (Figura 3.3) e 576 imagens de outros insetos (Figura 3.4) que são, geralmente, capturados em armadilhas adesivas amarelas nessas regiões, formando assim um banco de imagens com uma representativa amostragem da região citrícola paulista. Esse banco de imagens foi criado e cedido exclusivamente para atender as necessidades desta pesquisa e atualmente não se encontra acessível publicamente.



Figura 3.2: Equipamento de aquisição das imagens e armadilhas adesivas amarelas.

O equipamento utilizado para essa aquisição foi ajustado com uma ampliação de imagem fixa de 0.67x e uma distância focal sutilmente variável que possibilitasse o enquadramento de um inseto por imagem, independente do seu tamanho na armadilha. Esse tipo de abordagem de aquisição gerou imagens de insetos em diversas escalas, inclusive para a mesma espécie. Isso simulou a aquisição de imagens através de dispositivos com ajuste de distância focal variável e introduziu um grau maior de dificuldade e realismo ao problema de classificação proposto nesta pesquisa. Todas as imagens resultantes do processo de aquisição tiveram a largura de 2048 *pixels* e a altura de 1532 *pixels*.

Na medida em que as imagens foram adquiridas, as mesmas foram separadas em dois diretórios nomeados como “Diaphorina” e “Outros”. Isso possibilitou que os pesquisadores do FUNDECITRUS, especialistas no objeto de estudo, realizassem a separação das imagens desse banco em duas classes.

Uma vez que o trabalho de aquisição e separação de imagens foi concluído, foram realizadas verificações de integridade do banco de imagens com a finalidade de ratificar a inexistência de imagens repetidas ou a inexistência de imagens separadas de forma incorreta, devido a eventuais erros humanos. Para a análise de imagens repetidas foi executado um procedimento computacional para a comparação de imagens, todas comparadas contra todas. E, para a análise de imagens separadas de forma incorreta foram inspecionadas cuidadosa e manualmente cada imagem dos dois diretórios, 4

vezes. No resultado dessas verificações não foram encontradas imagens repetidas ou imagens em diretórios incorretos.

Antes de dar início a etapa de extração de características, todas as imagens utilizadas foram recortadas de forma a aproximar suas laterais dos corpos dos insetos. Esse procedimento computacional foi realizado para reduzir a área de fundo da imagem e, conseqüentemente, diminuir a captura de informação ruidosa presente nesses locais. Esse procedimento gerou imagens em resoluções distintas.



Figura 3.3: Algumas amostras do banco de imagens do *Diaphorina citri*.

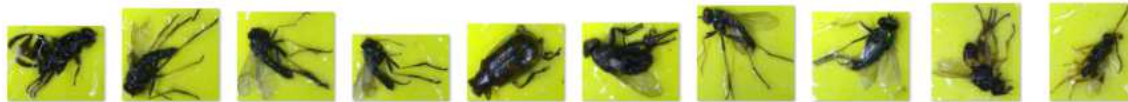


Figura 3.4: Algumas amostras do banco de imagens dos outros insetos.

O procedimento computacional de recorte (*Crop*) de imagens foi composto pelas seguintes etapas:

- (a) carregar duas imagens em espaços de cores baseados nos modelos RGB (*Red-Green-Blue*) (Figura 3.5) e CMYK (*Cyan-Magenta-Yellow-Black [Key]*);
- (b) obter o histograma do canal K da imagem carregada em um espaço de cores baseado no modelo CMYK (Figura 3.6);
- (c) percorrer os índices do vetor representante desse histograma (Figura 3.7) em ordem decrescente comparando os seus valores de conteúdo, iniciando essa busca no índice 150 (ou no índice final se o vetor tiver tamanho menor ou igual a 150) e tendo como índice de parada dessa busca o primeiro índice cujo valor de seu conteúdo for igual a 10000 (o índice final encontrado é usado como ponto de limiarização ou *threshold*);
- (d) realizar a limiarização (Figura 3.8) da imagem do canal K utilizando o índice encontrado como o valor de *threshold* (resultando em uma imagem binária);
- (e) encontrar o contorno do objeto principal dessa imagem binária criando, assim, uma máscara binária (Figura 3.9) desse objeto;
- (f) e realizar o recorte (Figura 3.10) da imagem em espaço de cores baseado em RGB através da máscara binária gerada.

Espaços de cores baseados no modelo RGB são implementações comumente encontradas nas funções de bibliotecas de visão computacional atuais e por esse motivo esse modelo foi testado, analisando os resultados de seus canais [Bradski e Kaehler 2008, Van Der Walt et al. 2014]. Um espaço de cores baseado no modelo CMYK foi utilizado porque separa completamente em dois de seus canais as cores amarela e preta. A cor amarela é predominante no fundo das armadilhas e a cor preta (ou algum tom de cinza) destaca a maioria dos insetos. O isolamento dessas cores em canais distintos auxiliou no procedimento de recorte de imagens acima descrito.

O valor 10000, usado como valor de conteúdo buscado no vetor que representa o histograma gerado, é utilizado para descobrir o índice que será utilizado como valor de *threshold* e pode ser diferente para cada imagem. Esse valor de conteúdo procurado resultou em índices que trouxeram recortes satisfatórios para todas as imagens do banco utilizado, sendo que esses recortes resultantes foram verificados em todas as imagens desse banco por procedimento de inspeção visual. Além disso, foram analisados os histogramas do canal K de todas as imagens do banco de amostras utilizado e após essa análise foi definido o valor de 150 como um índice inicial de busca adequado. Foram considerados satisfatórios os recortes que selecionavam apenas o inseto na imagem sem excluir alguma parte de seu exoesqueleto, mesmo não eliminando por completa a área de fundo da armadilha adesiva amarela.

Após a execução do (*Crop*) de imagens, as imagens resultantes tiveram variação de largura e altura. A largura das imagens recortadas variou de 211 *pixels* a 2020 *pixels* e a altura variou de 223 *pixels* a 1511 *pixels*.

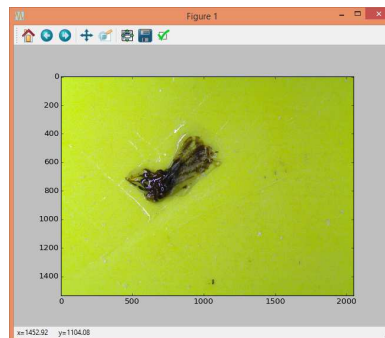


Figura 3.5: Imagem do *Diaphorina citri* carregada com um espaço de cores baseado em modelo RGB.

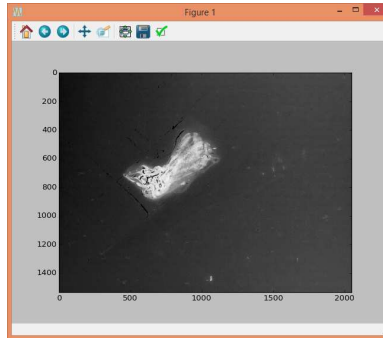


Figura 3.6: Imagem do *Diaphorina citri* carregada com o canal K de um espaço de cores baseado em modelo CMYK.

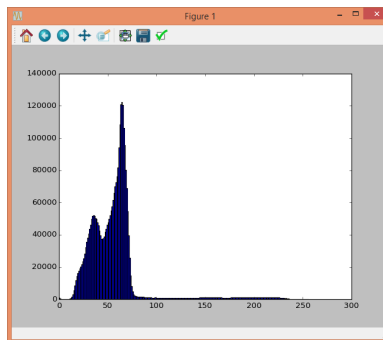


Figura 3.7: Histograma do canal K de um espaço de cores baseado em modelo CMYK - Imagem do *Diaphorina citri*.

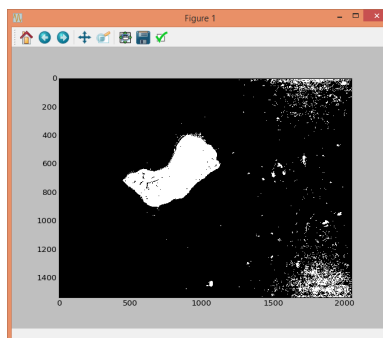


Figura 3.8: Limiarização da imagem do canal K utilizando o *threshold* encontrado - Imagem do *Diaphorina citri*.

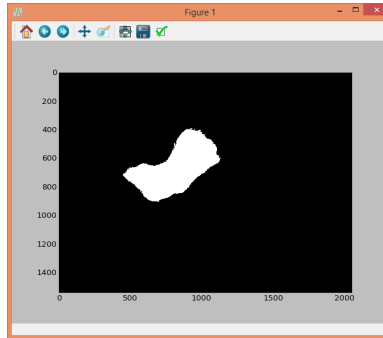


Figura 3.9: Máscara binária feita a partir do contorno do objeto principal encontrado - Imagem do *Diaphorina citri*.

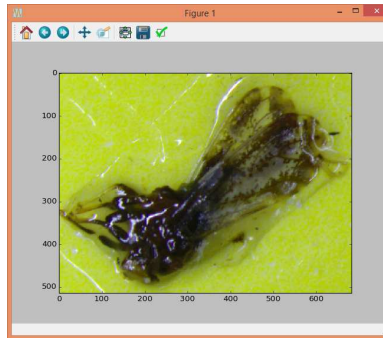


Figura 3.10: Recorte feito a partir do máscara binária encontrada - Imagem (RGB) do *Diaphorina citri*.

3.2 Rotulagem de Amostras e Extração de Características

Na rotulagem, um procedimento computacional realizou a leitura das imagens nos diretórios “Diaphorina” e “Outros” e, como resultado, gerou arquivos contendo as informações dos rótulos dessas imagens e dos seus endereços de armazenamento. Como rótulos, o número 0 (zero) foi atribuído a classe “Outros” e o número 1 (um) foi atribuído a classe “Diaphorina”.

Em seguida, foram executadas rotinas para a extração de características invariantes a rotação ou escala. Inicialmente, foram utilizados os algoritmos de descrição de características ORB (*Oriented FAST and Rotated BRIEF*) [Rublee et al. 2011], SIFT (*Scale-Invariant Feature Transform*) [Lowe 2004], SURF (*Speeded Up Robust Features*) [Bay et al. 2006], BRISK (*Binary Robust Invariant Scalable Keypoints*) [Leutenegger et al. 2011] e FREAK (*Fast Retina Keypoint*) [Alahi et al. 2012] com seus respectivos detectores de pontos, presentes na biblioteca de visão computacional OpenCV. Esses algoritmos foram

executados em suas configurações de parâmetros padrão. Os algoritmos de descrição de características ORB, SIFT, SURF e BRISK possuem detectores de pontos locais próprios. Já o descritor FREAK não possui um algoritmo de detecção de pontos associado. Baseada na análise comparativa de descritores (SURF, FREAK e BRISK), feita por Khvedchenia [2012], foi escolhido o detector de pontos do algoritmo SURF para ser combinado ao descritor FREAK pela rapidez e robustez de detecção.

Em uma segunda fase de extração de características, os 4 descritores utilizados e que possuem detectores de pontos próprios tiveram seus detectores substituídos pelo detector de pontos do algoritmo SURF para uma comparação mais adequada utilizando a mesma quantidade de pontos detectados para cada extração de características. Considerando a primeira etapa (descritores combinados com detectores de pontos próprios) e a segunda etapa (descritores combinados com detector de pontos SURF) de extração de características, foram realizados 9 tipos de extração de características distintas. A escolha do detector de pontos do algoritmo SURF, na segunda fase de extração de características, se deu pela apresentação de melhores resultados de acurácia em testes de classificação iniciais decorrentes da primeira etapa de extração de características.

A escolha desses descritores de características, invariantes a rotação ou escala, está relacionada à possibilidade de tornar as abordagens de classificação, propostas e experimentadas, extensíveis a uma variação maior de escalas e rotações das imagens dos insetos.

Para a implementação de cada um dos 9 tipos de extração de características foi criado um procedimento computacional que executou as tarefas de: ler os arquivos contendo os endereços das imagens, carregar cada imagem na memória, convertê-la para a escala de cinza, extrair (detectar e descrever) suas características e salvar em arquivo o resultado dessas extrações, indexando-as de acordo com seus devidos rótulos.

Para evitar resultados de classificação tendenciosos, as amostras resultantes das extrações de características, mantendo a indexação de seus respectivos rótulos, foram dispostas em ordem aleatória, de forma que essa ordenação (ordem pseudo-aleatória) pudesse ser reproduzida em todos os testes. Esse processo foi repetido 20 vezes sequencialmente para cada uma dessas extrações, objetivando tornar as amostras aleatórias. A reprodutibilidade dessa ordenação nas amostras foi feita para tornar as classificações das características extraídas adequadamente comparáveis.

3.3 Criação dos Livros de Código e das Bolsas de Características

Na etapa de criação de livros de código (*Codebooks*) e bolsas de características (*Bag-of-Features*) [Tsai 2012], as amostras de extrações de características geradas, com

seus respectivos rótulos, foram divididas em duas partes: 90% das amostras foram utilizadas para a geração de cada livro de código (conjunto de desenvolvimento) e 10% das amostras foram reservadas para os testes finais (conjunto de testes final) de classificação [Harrington 2012, Hackeling 2014]. Tanto o conjunto de 90% das amostras quanto o conjunto de 10% foram divididos de forma que cada parte contivesse 50% dos rótulos positivos (Classe 1) e 50% dos rótulos negativos (Classe 0), ou seja, os conjuntos das amostras foram divididas de forma estratificada.

O recurso de ordenação e estratificação de amostras (“*Stratified Shuffle Split*”) utilizado está disponível na biblioteca de aprendizagem de máquina *python* Scikit-Learn [Pedregosa et al. 2011]. O parâmetro “*Random State*”, muito comum em diversos algoritmos no Scikit-Learn, é utilizado para evitar a aleatoriedade na execução de muitos de seus algoritmos que possuem estados iniciais variáveis, fixando esses estados. O valor desse parâmetro pode ser qualquer número do tipo inteiro escolhido a critério de quem utiliza algum algoritmo presente nessa biblioteca. Nesta pesquisa, foi sempre atribuído o valor 1234 para esse parâmetro por questões de padronização, em todos os algoritmos que necessitam desse tipo de configuração.

Os livros de código foram criados usando o algoritmo de aprendizagem de máquina não-supervisionada *Mini Batch K-Means*, uma variação do algoritmo *K-Means* [Hackeling 2014, Scikit-learn 2014a].

A quantidade de agrupamentos configurada no *Mini Batch K-Means* foi de 300 e esse valor de agrupamento permaneceu fixo durante os primeiros experimentos para possibilitar comparações adequadas. Além disso, a fixação do número de agrupamentos reduz o número de análises necessárias devido a extensão de combinações de parâmetros possíveis proporcionada pela variação desses grupos.

Essa opção por 300 agrupamentos se baseou no exemplo de implementação bem sucedida de livro de código e bolsas de características descrito na obra denominada “*Mastering Machine Learning with Scikit-Learn*” [Hackeling 2014].

Foram criados três tipos distintos de implementação de livro de código. No primeiro, foi criado um livro de código geral (*Codebook* “Geral”), que engloba tanto as características do *Diaphorina citri* quanto as características dos outros insetos, agrupando essas características em 300 conjuntos. No segundo, foi criado um livro de código somente com características do *Diaphorina citri* (*Codebook* “*Diaphorina*”), distribuídas em 300 grupos. E, no terceiro, foi criado um livro de código com 300 agrupamentos de características para cada classe (*Codebook* “Duas Classes” [Classes “*Diaphorina*” e “Outros”]), totalizando 600 agrupamentos.

Uma vez que os livros de código foram gerados, bolsas de características foram criadas, uma por amostra, tanto para o conjunto de 90% quanto para o conjunto de 10% das amostras. Nos casos das bolsas de características geral e com características somente do *Diaphorina citri*, o vetor de características resultante, por amostra, é de 300 dimensões. Porém, na bolsa de características por classe, o vetor de características resultante é de 600 dimensões, 300 por classe.

3.4 Classificação das Amostras

As bolsas de características geradas e seus respectivos rótulos foram utilizados como entrada para os procedimentos computacionais de classificação. Os algoritmos de classificação escolhidos foram *Support Vector Machine* (SVM) (implementado na classe “SVC” do Scikit-Learn) e *K Nearest Neighbor* (KNN). O algoritmo SVM foi escolhido devido a sua ampla utilização em abordagens bem sucedidas de classificação de imagens [Boiman et al. 2008, Chaplot et al. 2006, Kotsia e Pitas 2007]. O algoritmo KNN foi escolhido pela sua simplicidade de implementação, por ser uma maneira efetiva de classificar dados e por ter sido aplicado com sucesso na classificação de imagens [Harrington 2012, Hmeidi et al. 2008].

O algoritmo KNN foi executado nas configurações de parâmetros padrão do Scikit-Learn, com valor do parâmetro k igual a 5. O algoritmo SVM, também executado em suas configurações padrão, teve o valor do parâmetro c igual a 1. O núcleo (kernel) SVM escolhido foi o linear. Essas escolhas deveram-se a simplicidade de classificação desse núcleo e a diminuição do número de análises comparativas que seriam necessárias pelo uso de outros núcleos ou outros intervalos de valores de parâmetros. Em procedimentos posteriores de aprimoramento de classificador esses parâmetros e núcleos foram variados.

As bolsas de características de cada um dos 9 tipos de extração de características executados (ORB/ORB, SIFT/SIFT, SURF/SURF, BRISK/BRISK, SURF/FREAK, SURF/ORB, SURF/SIFT, SURF/SURF e SURF/BRISK) foram classificadas tanto com KNN quanto com SVM, nas 3 abordagens de bolsas de características utilizadas. Ou seja, foram combinados 9 tipos de extração de características com 3 tipos de abordagens de bolsas de características e 2 algoritmos de classificação, totalizando 54 experimentos testados somente nessa etapa.

A métrica de desempenho selecionada, que é comumente utilizada para avaliar os resultados de classificação, foi a acurácia [Harrington 2012]. Essa métrica foi usada em pesquisa que envolveu o cálculo de desempenho de classificação do processo de contagem manual do *Diaphorina citri* em armadilhas adesivas amarelas [Leonardo 2014], e também foi utilizada em trabalhos correlatos a esta pesquisa [Zhang e Hu 2010, Yang et al. 2010, Larios et al. 2011, Lu et al. 2012, Mundada e Gohokar 2013, Venugoban e Ramanan 2014]. Além da acurácia, outras métricas como a precisão, a sensibilidade e o *F1-score* foram calculadas e apresentadas, nesta pesquisa, apenas de forma quantitativa.

O conjunto de desenvolvimento (90% das amostras) foi dividido novamente para possibilitar o processo de validação cruzada. Esse conjunto foi dividido em 10 partes (dobras) iguais e estratificadas. Ou seja, foi utilizada a validação cruzada com k (k -fold) igual a 10 (*10-fold CV*). Cada um dos *folds* (ou dobras) tinha 10% das amostras do conjunto de desenvolvimento.

Na validação cruzada com *10-fold CV*, 9 *folds* (conjunto de treinamento) são usados para treinamento do modelo de aprendizagem de máquina e 1 *fold* (conjunto de

validação) é usado na avaliação de desempenho de classificação do modelo. Para esse valor k especificamente escolhido, a validação cruzada é realizada com 10 iterações (treinamento e avaliação) e em cada uma dessas repetições o conjunto (*fold*) de validação é substituído por um dos *folds* que ainda não foi selecionado no conjunto de treinamento. Após as 10 iterações é calculada a média dos desempenhos de classificação e o seu desvio padrão.

O processo de validação cruzada foi utilizado tanto com o algoritmo KNN quanto no algoritmo SVM. Russel e Norvig [2013] afirmam que os valores populares de k (*k-fold CV*) são 5 e 10 e que esses números de iterações são suficientes para fornecer uma estimativa que é estatisticamente provável que seja precisa, à um custo razoável de computação. Usando um conjunto de dados com amostras simuladas, onde já era sabida a taxa de erro de classificação real, James et al. [2013] mostraram que a validação cruzada com k igual a 10 (*10-fold CV*) estimava modelos com taxas de erro de classificação mínimas muito próximas das taxas de erro de classificação mínimas reais. Essas justificativas são a base para a utilização de validação cruzada com *k-fold CV* igual a 10 neste projeto de pesquisa.

A escolha dos percentuais dos conjuntos de desenvolvimento e de testes final baseou-se nos percentuais gerados para treinamento e testes utilizando a abordagem de validação cruzada escolhida (*10-fold CV*), 90% para treinamento e 10% para teste.

Por fim, o conjunto de desenvolvimento (90% das amostras) é reunido novamente, após os resultados de desempenho de classificação gerados pela validação cruzada, para um treinamento final dos modelos de aprendizagem de máquina escolhidos. Nesse momento, o conjunto de testes finais (10% das amostras) é finalmente utilizado para uma última predição de desempenho de classificação, mostrando a generalização real desses modelos. Os percentuais de 90% para o conjunto de desenvolvimento e de 10% para o conjunto de testes finais foram escolhidos baseados nas divisões e justificativas do processo de validação cruzada utilizado. Ou seja, buscou-se manter as mesmas divisões proporcionais das etapas de treinamento de modelo e avaliação de desempenho, em todo processo.

O processo de divisão do banco de amostras, as iterações da validação cruzada utilizada e a reunião do conjunto de desenvolvimento, após a validação cruzada, para uma avaliação do desempenho de classificação e avaliação de generalização dos modelos encontram-se ilustrados na Figura 3.11.

Divisão do Banco de Amostras, Validação Cruzada e Avaliação Final de Desempenho

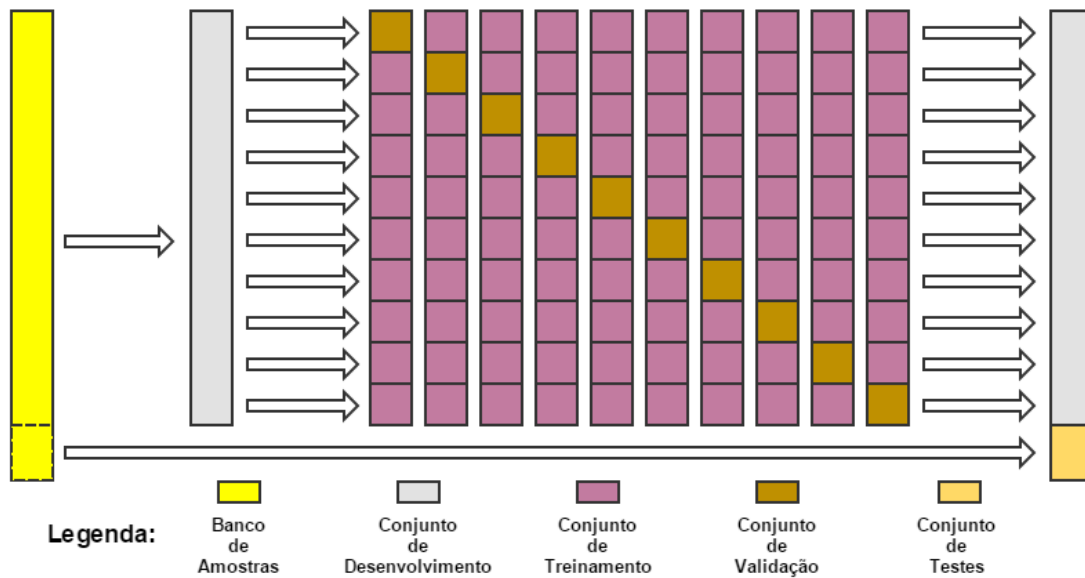


Figura 3.11: Divisão do banco de amostras com características extraídas das imagens dos insetos, processo de validação cruzada e avaliação final do desempenho de classificação.

Considerando que empresas da região citrícola paulista especializadas no processo de contagem manual do *Diaphorina citri* em armadilhas adesivas amarelas realizam esse procedimento em tempos médios de 3 a 5 minutos, a medição dos tempos médios de execução das abordagens automáticas testadas é uma informação importante para uma comparação entre processos e a determinação da viabilidade dessas soluções apresentadas. Essa métrica foi calculada através da soma de três tempos médios: o tempo médio de extração de características por imagem, o tempo médio de geração de bolsas de características por imagem e o tempo médio de classificação dessas bolsas de características, também por imagem. A seguir a fórmula correspondente a essa métrica:

$$TME = \frac{TE}{TI} + \frac{TGBof}{TI} + \frac{TC}{TI}$$

Onde, TME é o tempo médio de execução da abordagem de classificação, TE é o tempo de extração de características, TGBof é o tempo de geração de uma bolsa de características, TC é o tempo de classificação das bolsas de características e TI é o total de imagens consideradas em cada etapa do cálculo. Essa métrica foi calculada para todas as abordagens testadas.

A Figura 3.12 apresenta, de forma detalhada, as etapas do processo de classificação das amostras, partindo do banco de imagens particionado, estratificado e rotulado até a classificação e geração dos resultados.

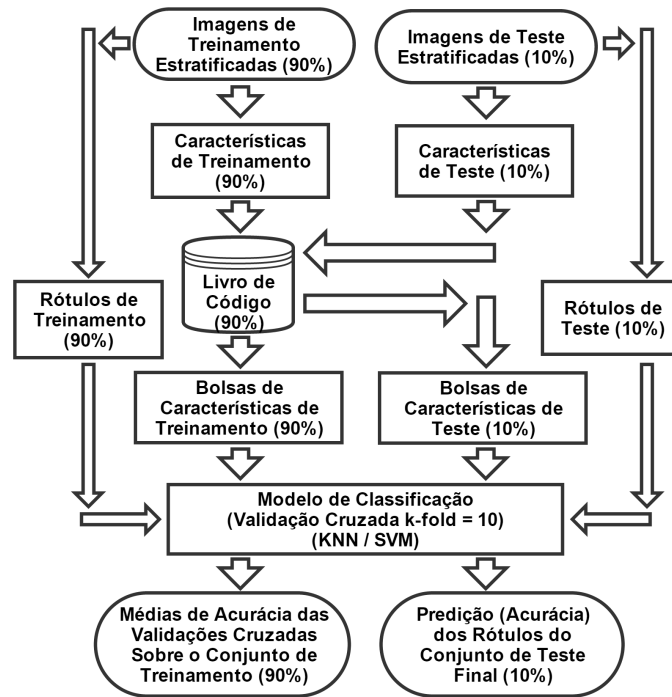


Figura 3.12: Etapas detalhadas do processo de classificação e geração de resultados.

3.5 Aprimoramento dos Classificadores

Na etapa de aprimoramento dos classificadores, tentou-se melhorar os resultados de desempenho de classificação das duas abordagens de reconhecimento automático resultantes que tiveram os melhores percentuais de acurácia na etapa de validação cruzada, selecionando uma abordagem resultante para cada algoritmo de aprendizagem de máquina utilizado. Para cada uma dessas abordagens de reconhecimento automático resultantes, foi executado um procedimento personalizado de seleção de modelos de aprendizagem de máquina.

Para reduzir o número de procedimentos de seleção de modelos a serem testados, um estudo prévio foi realizado com o objetivo de determinar qual o melhor tipo de extração de características e a melhor abordagem de bolsas de características.

Foram testados 9 tipos de extração de características nos 3 tipos de abordagens de bolsas de características combinados com os 2 algoritmos de aprendizagem de máquina selecionados. Com isso, foram gerados resultados de acurácia para todas essas combinações de abordagens, o que possibilitou a determinação e fixação das melhores abordagens de extração de características e de abordagens de bolsas.

Em testes subsequentes, após a definição de qual extrator de características e qual abordagem de bolsas de características rederam melhores acurácias de classificação, foi realizado um estudo da variação da quantidade de agrupamentos do algoritmo

Mini Batch K-Means, ou seja, um estudo do comportamento da acurácia em função da variação do número de grupos definidos para as bolsas de características. Os valores de agrupamento testados pertenceram ao intervalo de 25 a 1000, variando de 25 em 25 grupos. O intervalo escolhido e a variação de grupos selecionada se mostraram adequados para analisar o comportamento de desempenho de classificação observado. Esse intervalo de agrupamentos gerou resultados de classificação para os 2 algoritmos de aprendizagem de máquina utilizados. O trabalho de Venugoban e Ramanan [2014] mostrou um estudo simplificado do número de agrupamentos de bolsas de características aplicado a um contexto específico e serviu de base para a definição dos testes de variação de agrupamento realizados nesta pesquisa.

Uma vez encontrado o número k (número de agrupamentos) para o algoritmo *Mini Batch K-Means* que obteve a melhor acurácia na validação cruzada (em cada algoritmo de aprendizagem de máquina utilizado), foi iniciada a seleção de modelos de aprendizagem de máquina com o objetivo de melhorar a acurácia das duas melhores abordagens de classificação encontradas, uma abordagem resultante com melhor desempenho de classificação utilizando o algoritmo KNN e outra utilizando o algoritmo SVM.

A estratégia de seleção de modelos de aprendizagem de máquina utilizada é conhecida como *Grid Search* (Busca em Grade). Essa estratégia consiste basicamente em configurar intervalos de valores para os parâmetros selecionados nos algoritmos de aprendizagem de máquina e realizar treinamento e validação de cada um desses parâmetros (modelos) [Scikit-learn 2014b, Garreta e Moncecchi 2013]. Objetivo da utilização dessa seleção foi encontrar modelos de aprendizagem de máquina aprimorados para a classificação dos insetos estudados.

Os números de agrupamentos de bolsas de características aprimorados para os algoritmos KNN e SVM foram fixados e deu-se início o processo de seleção de modelos de aprendizagem de máquina para esses dois algoritmos.

O algoritmo KNN foi testado com o parâmetro k (ou n *neighbors*) variando de 1 a 114 (de 1 até 10% do número total de amostras do banco de imagens), com o parâmetro *weights* (pesos) em todas as suas variações (*'uniform'* e *'distance'*), com o parâmetro *algorithm* (algoritmo) em todas as suas variações (*'auto'*, *'ball tree'*, *'kd tree'* e *'brute'*), com o parâmetro *leaf size* (tamanho da folha) variando de 25 a 35 (em intervalo de números inteiros e variando em torno do valor padrão do parâmetro, 30) e com o parâmetro p (métricas de distância) em todas as suas métricas disponíveis (1 = *'manhattan distance'*, 2 = *'euclidean distance'*, 3 = *'minkowski distance'*) [Scikit-learn 2014d]. James et al. [2013] realizaram um estudo do comportamento dos erros de treinamento e teste em classificação utilizando o algoritmo KNN. Nesse estudo, demonstrou-se boas estimativas de erros de teste selecionando modelos com o parâmetro k variando de 1 a 100 e utilizando *10-fold CV*. O total de modelos testados para o algoritmo KNN foi de 27360.

O algoritmo SVM foi testado com o parâmetro *kernel* (núcleo) nas opções *'linear'* (núcleo com função linear) e *'rbf'* (núcleo com função baseada em raios). O nú-

cleo *rbf* foi testado com o parâmetro c (penalidade do termo de erro da função do núcleo) variando de 10^{-12} a 10^{11} (25 valores) e o parâmetro γ (coeficiente de ajuste do núcleo) variando de 10^{-12} a 10^{11} (25 valores). O núcleo *linear* foi testado com o parâmetro c variando de 10^{-12} a 10^{11} (25 valores) [Scikit-learn 2014e, Garreta e Moncecchi 2013]. Uma estratégia para a descoberta dos melhores valores para os parâmetros c e γ é testar intervalos pequenos iniciais e caso o melhor resultado de classificação esteja em uma das extremidades desse intervalo, deve-se expandir essa extremidade [Scikit-learn 2014c, Garreta e Moncecchi 2013]. Nesta pesquisa, o intervalo escolhido foi suficiente para manter um resultado ótimo de classificação entre os valores das extremidades. O total de modelos testados para o algoritmo SVM foi de 650.

Capítulo 4

Resultados e Discussões

“Se quisermos alcançar resultados nunca antes alcançados, devemos empregar métodos nunca antes testados.”

– Francis Bacon

Neste capítulo serão apresentados os resultados gerados a partir da metodologia apresentada no capítulo anterior. Esses resultados serão mostrados na ordem em que foram obtidos juntamente com as análises quantitativas e qualitativas, quando apropriadas.

Para auxiliar na análise dos resultados a serem apresentados é importante definir o que é um bom resultado em termos de acurácia para este trabalho. Baseando-se no trabalho de Leonardo [2014], que avaliou a acurácia do processo manual de inspeção em armadilhas adesivas amarelas, bons resultados de acurácia para as abordagens de classificação automática testadas, são os que atingem ou ultrapassam os percentuais médios do processo de contagem manual que é de aproximadamente 93%. O critério de seleção das melhores abordagens de classificação automática, durante todo processo de geração de resultados, foi a acurácia alcançada em seus correspondentes processos de validação cruzada.

Todas as abordagens de classificação testadas utilizaram um total de 1152 imagens (576 imagens do inseto *Diaphorina citri* e 576 imagens de outros insetos presentes em armadilhas adesivas amarelas coletadas na região citrícola do estado de São Paulo - Brasil) e validação cruzada com 10 *folds*.

O computador, no qual todos os testes foram executados, possuiu as seguintes configurações de hardware e software: 1) processador Intel (R) Core (TM) i5-4210U CPU @ 1.70GHz - 2394MHz; 2) memória RAM de 8GB; 3) armazenamento em SMART SSD SZ9STE 2.5 7mm 128GB; 4) sistema operacional de 64 bits *Windows 8.1 Enterprise*; 5) ambiente de desenvolvimento integrado (IDE) *PyCHARM Professional*

Edition 4.5; 6) e pacote de bibliotecas com funções em *Python*, *Anaconda* 2.1.0 64 bits.

4.1 Primeira Fase de Experimentação Geral

Primeiramente, foram extraídas características invariantes à rotação ou escala utilizando 5 tipos de descritores de características (ORB, SIFT, SURF, BRISK e FREAK), com seus respectivos detectores de pontos (exceto para o descritor FREAK, que não possui detector de pontos próprio e foi combinado com o detector de pontos do descritor SURF). A Tabela 4.1 mostra dados referentes a essas extrações. Os títulos das linhas apresentam os detectores de pontos e os descritores de características utilizados. Nas colunas, MaQCEI representa a maior quantidade de características extraídas em uma imagem, a coluna MeQCEI representa a menor quantidade de características extraídas em uma imagem, a coluna TPD representa o total de pontos detectados em todo processo, a coluna TCD representa o total de características descritas em todo processo, a coluna TTEE representa o tempo (em segundos) total de execução de todas as extrações e a coluna TMEC representa o tempo (em segundos) médio de extração de características de uma imagem.

Tabela 4.1: Dados gerados a partir de extrações de características com descritores e seus próprios detectores de pontos.

	MaQCEI	MeQCEI	TPD	TCD	TTEE	TMEC
ORB/ORB	29018	136	3184189	3184189	59,0469	0,0512
SIFT/SIFT	7685	64	916993	916993	385,4600	0,3346
SURF/SURF	11028	140	1952783	1952783	388,9790	0,3376
BRISK/BRISK	2455	9	277156	277156	44,9329	0,0390
SURF/FREAK	10371	68	1952783	1583039	182,0999	0,1580

Nesta pesquisa, foram analisados os custos computacionais de cada tipo de abordagem de extração, agrupamento e classificação de características. Assim, toda vez que for apresentado no texto o termo “custo computacional” isto se refere ao tempo em segundos gasto na execução dos procedimentos computacionais utilizados, envolvendo todas as amostras de imagens ou seu tempo médio para uma amostra.

Nos resultados dos experimentos apresentados na Tabela 4.1, a extração ORB/ORB teve o maior número de pontos detectados e descritos e o segundo melhor custo computacional. Já a extração BRISK/BRISK obteve o menor número de pontos detectados e descritos e o melhor custo computacional. O pior custo computacional foi da extração SURF/SURF, que obteve uma quantidade mediana de pontos detectados e descritos. Um caso particular é a extração SURF/FREAK que, ao usar detector de pontos diferente do descritor, obteve uma quantidade menor de pontos descritos do que a quantidade de pontos detectados nas imagens utilizadas.

Em seguida, foram utilizados os resultados de extração de características gerados como entrada para a geração dos resultados de classificação. As Tabelas 4.2, 4.4 e 4.6 mostram os resultados dessas classificações juntamente com as quantidades totais de características agrupadas na geração dos livros de código das bolsas de características. As Tabelas 4.3, 4.5 e 4.7 mostram os tempos (em segundos) de geração de bolsas de características e de classificação de características. Em todas essas tabelas, as linhas correspondem aos descritores utilizados e seus detectores de pontos correspondentes, quando existentes.

Nas tabelas com resultados de classificação (Tabelas 4.2, 4.4 e 4.6), as duas primeiras colunas correspondem aos resultados de acurácia obtidos no processo de validação cruzada para o algoritmo KNN com o parâmetro K igual a 5 (coluna “KNN $K=5$ (VC)”) e para o algoritmo SVM linear com o parâmetro C igual a 1 (coluna “SVM $L C=1$ (VC)”). A coluna “BoF QCA” corresponde a quantidade total de características agrupadas dentro dos procedimentos de geração de livros de código. Para as tabelas de registro de tempo (Tabelas 4.3, 4.5 e 4.7), a coluna “KNN TTP” corresponde ao tempo total de predição com o algoritmo KNN para todas as amostras utilizadas, “KNN TMPA” é o tempo médio de predição por amostra utilizando o algoritmo KNN, “SVM TTP” corresponde ao tempo total de predição com o algoritmo SVM para todas as amostras utilizadas, “SVM TMPA” é o tempo médio de predição por amostra utilizando o algoritmo SVM, “TTCB” representa o tempo total de criação de bolsas de características para todas as amostras utilizadas e “TMCB” é o tempo médio de criação de bolsas de características por amostra. Os tempos apresentados em “TTCB” e “TMCB” são valores compartilhados pelos dois procedimentos de classificação de características testados (KNN e SVM).

As Figuras 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6 apresentam gráficos com análises de tendência de desvio padrão em função da variação da acurácia alcançada no procedimento de validação cruzada. Esses gráficos apresentam essas tendências tanto para o algoritmo KNN quanto para o algoritmo SVM, nas 3 abordagens de bolsas de características utilizadas. No eixo horizontal desses gráficos está representado o intervalo com a acurácia variando de 60% a 100% e no eixo vertical está representado o intervalo com o desvio padrão variando de 0% a 12%. As linhas pontilhadas representam as tendências obtidas a partir dos gráficos produzidos.

Tabela 4.2: Dados gerados a partir de classificações de características com bolsas de características (Geral) e detectores de pontos próprios.

	KNN $K=5$ (VC)	SVM $L C=1$ (VC)	BoF QCA
ORB/ORB	76,94% (+/- 7,21%)	76,55% (+/- 3,63%)	2802210
SIFT/SIFT	84,16% (+/- 7,18%)	87,83% (+/- 5,00%)	808539
SURF/SURF	92,75% (+/- 6,01%)	90,92% (+/- 4,04%)	1735988
BRISK/BRISK	66,21% (+/- 9,05%)	67,57% (+/- 8,30%)	244495
SURF/FREAK	81,08% (+/- 6,16%)	82,74% (+/- 9,46%)	1404259

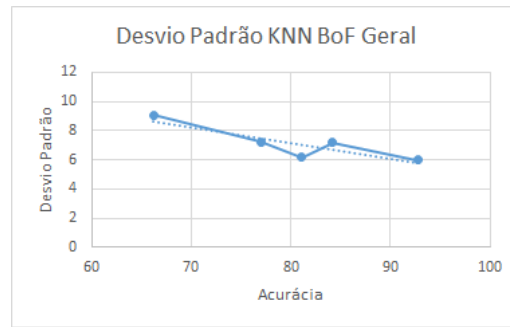


Figura 4.1: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.2.

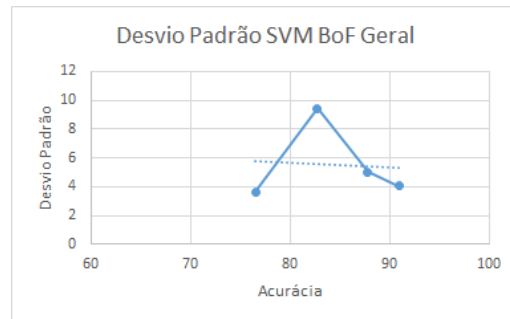


Figura 4.2: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.2.

Tabela 4.3: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Geral) com detectores de pontos próprios.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
ORB/ORB	0,0870	0,0007	0,0119	0,0001	43,6029	0,0378
SIFT/SIFT	0,0850	0,0007	0,0080	0,0000	15,5109	0,0134
SURF/SURF	0,0799	0,0006	0,0079	0,0000	27,2539	0,0236
BRISK/BRISK	0,0859	0,0007	0,0189	0,0001	4,3980	0,0038
SURF/FREAK	0,0850	0,0007	0,0090	0,0000	21,7679	0,0188

Tabela 4.4: Dados gerados a partir de classificações de características com bolsas de características (*Diaphorina citri*) e detectores de pontos próprios.

	KNN K=5 (VC)	SVM L C=1 (VC)	BoF QCA
ORB/ORB	75,96% (+/- 7,55%)	73,26% (+/- 7,75%)	918643
SIFT/SIFT	83,86% (+/- 6,13%)	88,41% (+/- 5,52%)	275590
SURF/SURF	92,56% (+/- 5,41%)	89,27% (+/- 4,69%)	616378
BRISK/BRISK	64,29% (+/- 6,06%)	68,74% (+/- 7,75%)	77159
SURF/FREAK	80,40% (+/- 7,63%)	82,83% (+/- 9,92%)	464935

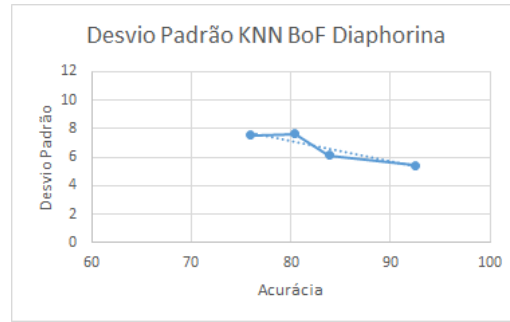


Figura 4.3: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.4.

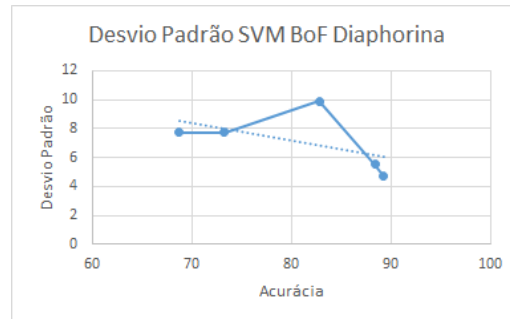


Figura 4.4: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.4.

Tabela 4.5: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (*Diaphorina citri*) com detectores de pontos próprios.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
ORB/ORB	0,0830	0,0007	0,0120	0,0001	40,4779	0,0351
SIFT/SIFT	0,0750	0,0006	0,0079	0,0000	14,5830	0,0126
SURF/SURF	0,0769	0,0006	0,0070	0,0000	27,3069	0,0237
BRISK/BRISK	0,0840	0,0007	0,0169	0,0001	4,3719	0,0037
SURF/FREAK	0,0820	0,0007	0,0099	0,0000	21,8190	0,0189

Tabela 4.6: Dados gerados a partir de classificações de características com bolsas de características (Por Classe) e detectores de pontos próprios.

	KNN K=5 (VC)	SVM L C=1 (VC)	BoF QCA
ORB/ORB	76,15% (+/- 7,07%)	76,46% (+/- 9,17%)	2802210
SIFT/SIFT	84,35% (+/- 5,62%)	90,34% (+/- 4,54%)	808539
SURF/SURF	93,14% (+/- 4,93%)	91,87% (+/- 6,45%)	1735988
BRISK/BRISK	65,54% (+/- 7,37%)	68,93% (+/- 7,50%)	244495
SURF/FREAK	80,98% (+/- 6,30%)	84,85% (+/- 7,77%)	1404259

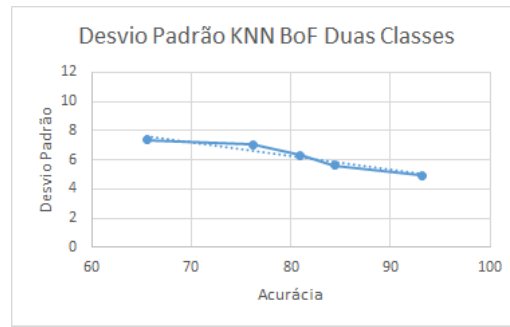


Figura 4.5: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.6.

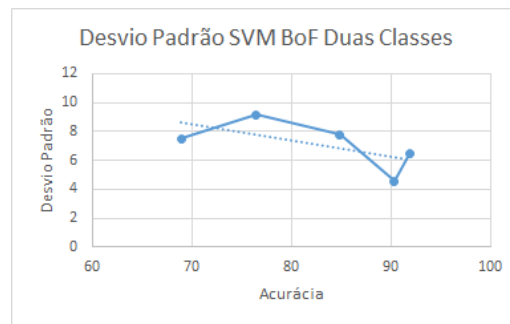


Figura 4.6: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.6.

Tabela 4.7: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Por Classe) com detectores de pontos próprios.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
ORB/ORB	0,1710	0,0014	0,0320	0,0002	85,3719	0,0741
SIFT/SIFT	0,1679	0,0014	0,0199	0,0001	29,1309	0,0252
SURF/SURF	0,1610	0,0013	0,0190	0,0001	53,7059	0,0466
BRISK/BRISK	0,1770	0,0015	0,0369	0,0003	8,4060	0,0072
SURF/FREAK	0,1689	0,0014	0,0260	0,0002	43,4169	0,0376

Um comportamento observado nos resultados de classificação iniciais é que o desvio padrão tende a diminuir a medida em que a acurácia no processo de validação cruzada aumenta (Tabelas 4.2, 4.4 e 4.6 e Figuras 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6).

É possível perceber, através das Tabelas 4.1, 4.3, 4.5 e 4.7, que todas as etapas individuais que compõe a classificação de uma imagem possuem tempos médios inferiores a 1 segundo.

Nas Tabelas 4.2, 4.4 e 4.6 são mostrados resultados de classificação inconclusivos pois há uma inadequação do número de características detectadas por cada extrator, apresentados na Tabela 4.1 e na coluna “TPD”. Para que exista uma comparação de resultados de classificação mais adequada, resolveu-se repetir todos os experimentos considerando a utilização de um único detector de pontos para todos os descritores utilizados. O detector de pontos escolhido foi o detector do descritor SURF. Esse detector foi escolhido por apresentar os melhores resultados de classificação, na etapa de validação cruzada, em todos os experimentos realizados (Tabelas 4.2, 4.4 e 4.6). Essa escolha também foi feita baseando-se na análise comparativa de descritores (SURF, FREAK e BRISK) realizada por Khvedchenia [2012], que considerou comparações de resultados de classificação utilizando o mesmo detector de pontos (SURF) para os descritores de características testados (SURF, FREAK e BRISK).

4.2 Segunda Fase de Experimentação Geral

A seguir, são apresentados os dados de extrações de características (Tabela 4.8), os resultados de classificação (Tabelas 4.9, 4.11 e 4.13) e as medições de tempo de execução (Tabelas 4.10, 4.12 e 4.14), considerando a utilização do detector de pontos do algoritmo SURF nestes experimentos.

Nessa segunda fase de experimentos, as Tabelas 4.9, 4.11 e 4.13 tem nomenclatura de linhas e colunas equivalentes e correspondentes as das Tabelas 4.2, 4.4 e 4.6, as Tabelas 4.10, 4.12 e 4.14 tem nomenclatura de linhas e colunas equivalentes e correspondentes as das Tabelas 4.3, 4.5 e 4.7 e os gráficos contidos nas Figuras 4.7, 4.8, 4.9, 4.10, 4.11 e 4.12 tem nomenclaturas equivalentes e correspondentes as das Figuras 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6, respectivamente. A única diferença nessas nomenclaturas é que, nas linhas das tabelas da segunda fase de experimentos, são apresentadas informações considerando o detector de pontos do algoritmo SURF em todos os resultados, substituindo os detectores de pontos próprios.

Tabela 4.8: Dados gerados a partir de extrações de características com descritores combinados ao detector de pontos SURF.

	MaQCEI	MeQCEI	TPD	TCD	TTEE	TMEC
SURF/ORB	10972	130	1952783	1913222	190,2699	0,1651
SURF/SIFT	11028	140	1952783	1952783	2167,0390	1,8811
SURF/SURF	11028	140	1952783	1952783	424,1620	0,3681
SURF/BRISK	10848	118	1952783	1828572	197,2020	0,1711
SURF/FREAK	10371	68	1952783	1583039	189,3719	0,1643

É possível perceber, através da Tabela 4.8, que o número de pontos detectados (coluna “TDP”) é o mesmo para todas as extrações realizadas. Os números de características descritas (coluna “TCD”) são iguais entre si ou tem valores próximos. A

detecção de quantidades equivalentes de pontos nas mesmas regiões locais dentro das imagens juntamente com as quantidades equivalentes ou aproximadas de características descritas proporcionou uma maior aproximação dos números de características extraídas, e com isso uma comparação mais adequada desses extratores.

Tabela 4.9: Dados gerados a partir de classificações de características com bolsas de características (Geral) e detectores de pontos SURF.

	KNN K=5 (VC)	SVM L C=1 (VC)	BoF QCA
SURF/ORB	86,18% (+/- 6,70%)	86,00% (+/- 11,68%)	1700462
SURF/SIFT	92,85% (+/- 3,84%)	96,61% (+/- 4,80%)	1735988
SURF/SURF	92,75% (+/- 6,01%)	90,92% (+/- 4,04%)	1735988
SURF/BRISK	83,49% (+/- 5,90%)	83,68% (+/- 6,54%)	1624417
SURF/FREAK	81,08% (+/- 6,16%)	82,74% (+/- 9,46%)	1404259

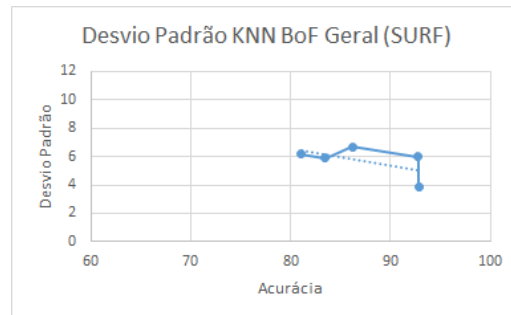


Figura 4.7: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.9.

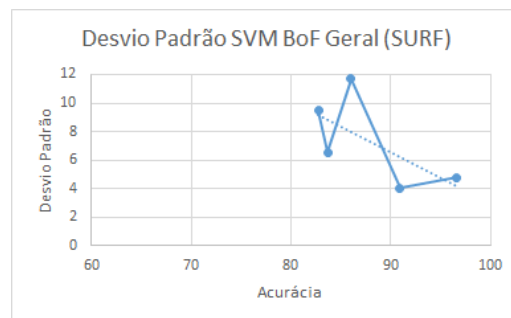


Figura 4.8: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.9.

Tabela 4.10: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Geral) com detectores de pontos SURF.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
SURF/ORB	0,0859	0,0007	0,0090	0,0000	25,6239	0,0222
SURF/SIFT	0,0859	0,0007	0,0059	0,0000	64,1960	0,0557
SURF/SURF	0,0810	0,0006	0,0079	0,0000	26,6070	0,0230
SURF/BRISK	0,0820	0,0007	0,0099	0,0000	25,1789	0,0218
SURF/FREAK	0,0850	0,0007	0,0109	0,0000	23,0700	0,0200

Tabela 4.11: Dados gerados a partir de classificações de características com bolsas de características (*Diaphorina citri*) e detectores de pontos SURF.

	KNN K=5 (VC)	SVM L C=1 (VC)	BoF QCA
SURF/ORB	85,52% (+/- 7,01%)	86,19% (+/- 9,08%)	599956
SURF/SIFT	94,11% (+/- 4,11%)	97,39% (+/- 2,74%)	616378
SURF/SURF	92,56% (+/- 5,41%)	89,27% (+/- 4,69%)	616378
SURF/BRISK	84,07% (+/- 8,23%)	84,75% (+/- 5,83%)	565983
SURF/FREAK	80,40% (+/- 7,63%)	82,83% (+/- 9,92%)	464935

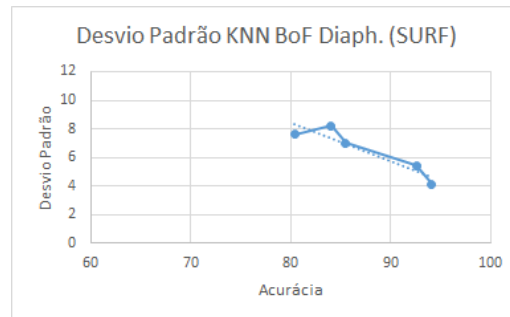


Figura 4.9: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.11.

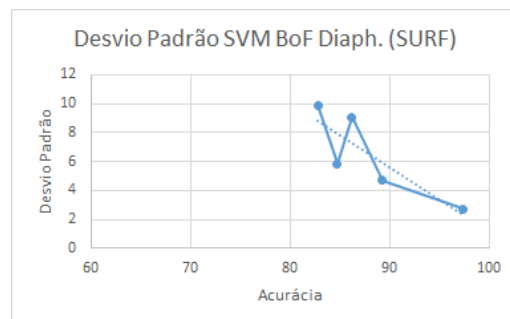


Figura 4.10: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.11.

Tabela 4.12: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (*Diaphorina citri*) com detectores de pontos SURF.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
SURF/ORB	0,1260	0,0010	0,0089	0,0000	24,3789	0,0211
SURF/SIFT	0,0759	0,0006	0,0069	0,0000	30,1730	0,0261
SURF/SURF	0,0839	0,0007	0,0080	0,0000	28,1769	0,0244
SURF/BRISK	0,0829	0,0007	0,0090	0,0000	25,5559	0,0221
SURF/FREAK	0,0850	0,0007	0,0130	0,0001	21,9289	0,0190

Tabela 4.13: Dados gerados a partir de classificações de características com bolsas de características (Por Classe) e detectores de pontos SURF.

	KNN K=5 (VC)	SVM L C=1 (VC)	BoF QCA
SURF/ORB	86,66% (+/- 7,12%)	87,55% (+/- 6,41%)	1700462
SURF/SIFT	93,53% (+/- 3,89%)	97,00% (+/- 4,46%)	1735988
SURF/SURF	93,14% (+/- 4,93%)	91,87% (+/- 6,45%)	1735988
SURF/BRISK	84,55% (+/- 6,33%)	86,68% (+/- 6,62%)	1624417
SURF/FREAK	80,98% (+/- 6,30%)	84,85% (+/- 7,77%)	1404259

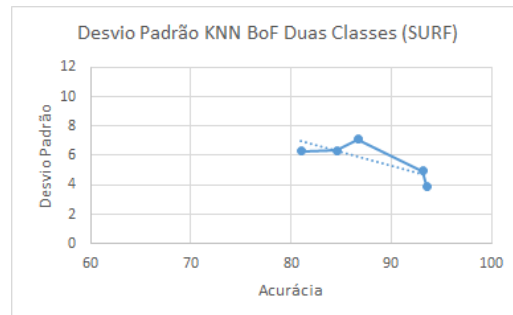


Figura 4.11: Gráfico de tendência de desvio padrão em função da acurácia (KNN), gerado a partir dos dados da Tabela 4.13.

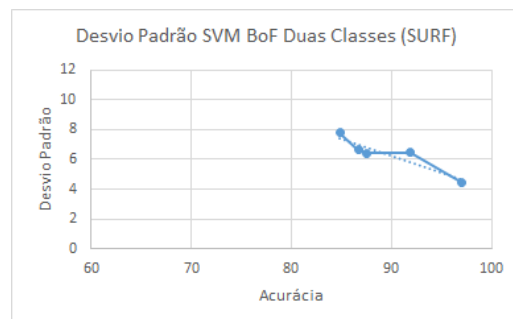


Figura 4.12: Gráfico de tendência de desvio padrão em função da acurácia (SVM), gerado a partir dos dados da Tabela 4.13.

Tabela 4.14: Dados de medição de tempo gerados a partir de classificações de características e geração de bolsas de características (Por Classe) com detectores de pontos SURF.

	KNN TTP	KNN TMPA	SVM TTP	SVM TMPA	TTCB	TMCB
SURF/ORB	0,1759	0,0015	0,0239	0,0002	48,1229	0,0417
SURF/SIFT	0,1890	0,0016	0,0139	0,0001	59,8730	0,0519
SURF/SURF	0,1619	0,0013	0,0190	0,0001	57,3619	0,0497
SURF/BRISK	0,1670	0,0014	0,0260	0,0002	50,2480	0,0436
SURF/FREAK	0,1769	0,0015	0,0249	0,0002	43,6010	0,0378

Ao analisar os dados temporais, nesta segunda fase de experimentos (Tabelas 4.8, 4.10, 4.12 e 4.14), novamente todas as etapas individuais que compõe a classificação de uma imagem possuem tempos médios inferiores a 1 segundo, exceto pela extração “SURF/SIFT” da Tabela 4.8. Nessa extração, o tempo médio de extração de características para uma imagem (coluna “TMEC”) é de aproximadamente 1,8 segundos. O aumento no tempo de execução dessa extração deveu-se ao aumento do número de pontos detectados, devido a troca do detector de pontos.

Novamente é observado, nesta segunda fase de experimentos, que o desvio padrão tende a diminuir a medida em que a acurácia no processo de validação cruzada aumenta (Tabelas 4.9, 4.11 e 4.13 e Figuras 4.7, 4.8, 4.9, 4.10, 4.11 e 4.12).

A partir da análise dos dados gerados por todos os experimentos anteriores, algumas definições de melhores abordagens puderam ser realizadas.

Comparando-se as tabelas com informações de desempenho de classificação da primeira fase de experimentos (Tabelas 4.2, 4.4 e 4.6) com as tabelas com informações de desempenho de classificação da segunda fase (Tabelas 4.9, 4.11 e 4.13), percebe-se uma melhoria, na segunda fase, dos resultados de acurácia nos seus processos de validação cruzada. Essa melhoria deveu-se a troca dos detectores de pontos dos próprios descritores pelo detector de pontos do algoritmo SURF.

Devido a melhoria de resultados de desempenho de classificação e a adequação do número de características extraídas, as análises seguintes basearam-se somente na segunda fase de experimentos.

As Tabelas 4.9, 4.11 e 4.13 mostram que o descritor SIFT combinado ao detector de pontos do algoritmo SURF teve as melhores acurácias nos processos de validação cruzada, tanto para o algoritmo de aprendizagem de máquina KNN quanto para o algoritmo SVM. Assim, o extrator “SURF/SIFT” contribuiu para o alcance de melhores resultados de desempenho de classificação.

Uma vez que o extrator “SURF/SIFT” foi selecionado como melhor abordagem de extração de características, realizou-se uma comparação dos resultados de acurácia dos processos de validação cruzada nas Tabelas 4.9, 4.11 e 4.13 com a finalidade de determinar qual a melhor abordagem de bolsas de características. Na Tabela

4.11, linha “SURF/SIFT”, foram encontrados os melhores resultados de acurácia em procedimento de validação cruzada. Assim, a abordagem de bolsas de características com características extraídas apenas das imagens do inseto *Diaphorina citri* contribuiu para os melhores resultados de desempenho de classificação. Em termos de custo computacional, a abordagem de bolsas de características selecionada possui menores tempos de execução (Tabela 4.12) pois utiliza o menor vetor de características gerado com a menor quantidade de características utilizadas. Isso pode ser observado comparando-se as colunas “TTCB” e “TMCB” das Tabelas 4.10, 4.12 e 4.14.

Em todos os resultados de testes apresentados até o momento, não foi possível determinar adequadamente qual algoritmo de aprendizagem de máquina gerou melhores resultados de desempenho de classificação para o extrator de características e a abordagem de bolsas de características selecionados. Pois, esses algoritmos possuem uma quantidade considerável de modelos passíveis de utilização e testar alguns deles contribui para encontrar os modelos aprimorados.

4.3 Estudo dos Agrupamentos das Bolsas de Características

Após fixar o extrator de características “SURF/SIFT” e a abordagem de bolsas de características com somente características do “*Diaphorina citri*” como melhores escolhas, foi dado início a um estudo da variação do parâmetro K das bolsas de características com a finalidade de encontrar valores de parâmetros que aprimoram os resultados de desempenho de classificação, tanto para o algoritmo KNN quanto para o algoritmo SVM.

As Figuras 4.13 e 4.14 apresentam gráficos com estudos da variação do parâmetro K (número de agrupamentos) para bolsas de características formadas com características apenas do inseto *Diaphorina citri* e extrator de características “SURF/SIFT”, considerando os algoritmos de aprendizagem de máquina KNN e SVM nas configurações de parâmetros iniciais da metodologia utilizada. Nesses gráficos, o eixo horizontal corresponde ao intervalo de números de agrupamentos (parâmetro K) das bolsas de características e o eixo vertical corresponde ao intervalo de valores de acurácia alcançados na etapa de validação cruzada. As linhas pontilhadas representam as tendências desses resultados de classificação.

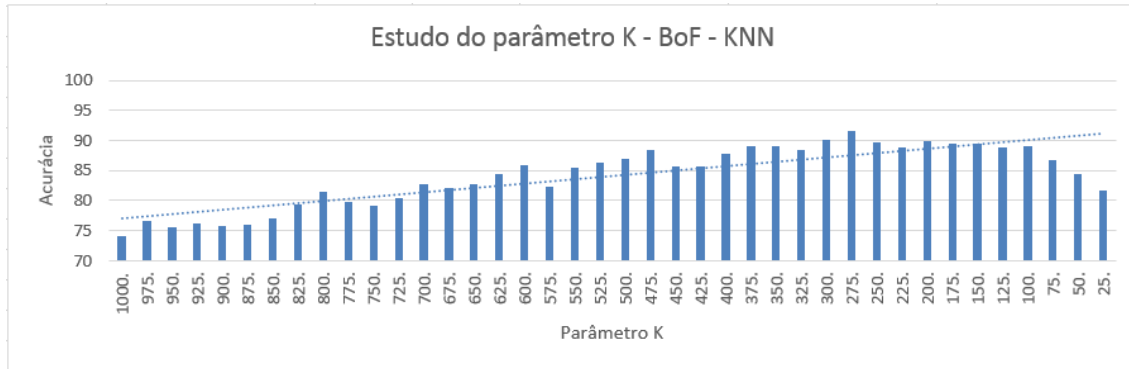


Figura 4.13: Estudo da variação do parâmetro K (número de agrupamentos) para bolsas de características “Diaphorina”, extrator “SURF/SIFT” e algoritmo KNN.

O gráfico da Figura 4.13 mostra que o melhor resultado de acurácia do intervalo experimentado foi encontrado ao testar 275 agrupamentos na abordagem de bolsas de características, utilizando o algoritmo KNN. O valor da acurácia usando 275 grupos foi de 91,46% e teve 3,51% de desvio padrão. Então, 275 agrupamentos foi considerado o parâmetro K ótimo dessa amostragem e foi fixado em testes subsequentes de aprimoramento de modelos de aprendizagem de máquina. A reta de tendência desse gráfico mostra que valores mais baixos de K tendem a atingirem melhores acurácias. Esse comportamento não acontece quando se utiliza valores de K abaixo de 275.

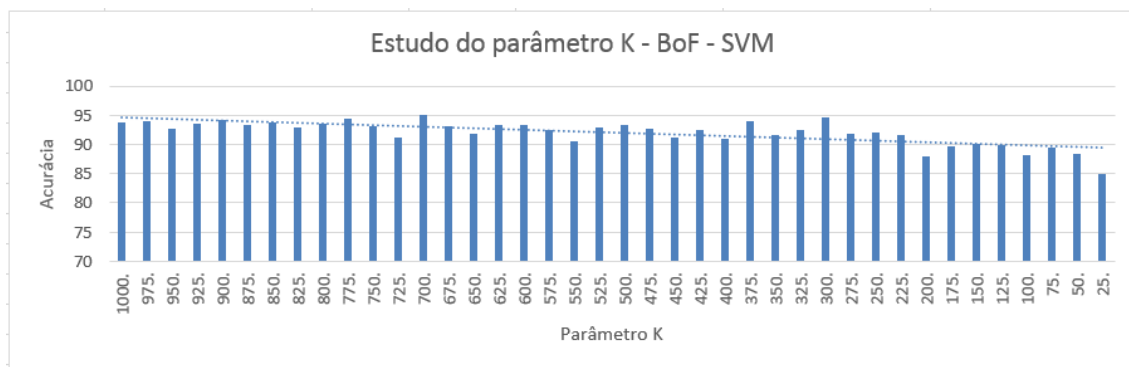


Figura 4.14: Estudo da variação do parâmetro K (número de agrupamentos) para bolsas de características “Diaphorina”, extrator “SURF/SIFT” e algoritmo SVM.

O gráfico da Figura 4.14 mostra que o melhor resultado de acurácia do intervalo experimentado foi encontrado ao testar 700 agrupamentos na abordagem de bolsas de características, utilizando o algoritmo SVM. O valor da acurácia usando 700 grupos foi de 94,96% e teve 2,62% de desvio padrão. Então, 700 agrupamentos foi considerado o parâmetro K ótimo dessa amostragem e foi fixado em testes subsequentes de aprimoramento de modelos de aprendizagem de máquina. A reta de tendência desse gráfico mostra que valores mais altos de K tendem a atingirem melhores acurácias.

Um caso particular a ser observado é que a acurácia alcançada com 300 agrupamentos chegou muito próximo ao resultado alcança com 700 grupos, com 94,65% de acurácia e desvio padrão de 2,74%.

4.4 Aprimoramento dos Classificadores

Ao encontrar o extrator “SURF/SIFT”, a abordagem de bolsas de características “Diaphorina” e os números de agrupamentos em bolsas de características de 275 para o algoritmo KNN e de 700 para o algoritmo SVM como abordagens e valores de parâmetros geradores de melhores resultados de classificação, essas abordagens e valores de parâmetros foram fixados para a realização de testes de aprimoramento de modelos de aprendizagem de máquina com o objetivo de descobrir modelos ótimos e determinar qual algoritmo alcançou melhores resultados.

Dois procedimentos de seleção de modelos de aprendizagem de máquina foram executados, um para o algoritmo KNN e outro para o algoritmo SVM. Na seleção de modelos utilizando o algoritmo KNN foram gerados resultados de desempenho de classificação para 27360 modelos testados. Já para o algoritmo SVM foram gerados resultados de desempenho de classificação para 650 modelos testados. Essas quantidades geradas de modelos de aprendizagem de máquina foram resultantes da variação dos parâmetros dos algoritmos utilizados.

Como resultado da execução do procedimento de Busca de Grade configurado para o algoritmo KNN, o modelo composto pelos hiper-parâmetros k (n neighbors ou número de vizinhos mais próximos) igual a 7, *weights* (pesos) igual a ‘uniform’ (uniforme), *leaf size* (tamanho da folha) igual a 25, *algorithm* (algoritmo) igual a ‘auto’ (automático) e p (métricas de distância) igual a 2 obteve o melhor desempenho de classificação desse primeiro procedimento de seleção de modelos. Outros parâmetros utilizados e executados em sua configuração padrão do Scikit-Learn foram: *metric* (métrica) com o valor igual a ‘minkowski’ e *metric params* (parâmetros da métrica) com o valor igual a ‘Nome’. Nesse procedimento de aprimoramento, a maior acurácia alcançada, considerando o processo de validação cruzada, foi de 93,73% e teve 3,82% como desvio padrão.

A Tabela 4.15 mostra o relatório do teste final de generalização para o modelo de aprendizagem de máquina selecionado, utilizando o algoritmo KNN. Nessa tabela, as linhas apresentam as classes “Outros” (0) e “Diaphorina” (1) e a média aritmética (*AVG*) (para as três primeiras colunas) ou o somatório total (*Total*) (para a última coluna). Nas colunas dessa tabela são apresentados os resultados de precisão (*Precision*), sensibilidade (*Recall*), média harmônica entre a precisão e a sensibilidade (*F1-score*) e o número de amostras utilizadas nesse teste por classe ou suporte (*Support*). Como resultado das três primeiras colunas, as médias aritméticas de precisão, sensibilidade e *F1-score* obtiveram no mesmo valor, 97,00%. Essas médias foram computadas considerando ambas as classes utilizadas em um momento como

classificações positivas e em outro momento como negativas. Nas quatro linhas após essa tabela encontram-se listadas informações como o número de classificações corretas sobre o total de imagens testadas e a acurácia, precisão, sensibilidade e *F1-score* obtidas considerando a classe “Diaphorina” como classificação positiva. Nesse teste de generalização houve 112 classificações de amostras (imagens) corretas em um total de 116 (4 erros de classificação), a acurácia alcançada foi de 96,55%, a precisão foi de 98,21%, a sensibilidade de foi de 94,83%, e 96,49% em *F1-score*.

Tabela 4.15: Relatório de teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo KNN.

	Precisão	Sensibilidade	F1-Score	Suporte
Outros (0)	95%	98%	97%	58
Diaphorina (1)	98%	95%	96%	58
AVG / Total	97%	97%	97%	116
Número de Classificações Corretas: 112/116 (Erro: 4)				
Acurácia: 96,55%				
Precisão: 98,21%				
Sensibilidade: 94,83%				
F1-Score: 96,49%				

Uma matriz de confusão é apresentada na Tabela 4.16 e mostra os resultados referentes ao procedimento de teste de generalização final para o modelo de aprendizagem de máquina selecionado utilizando Busca em Grade para o algoritmo KNN. Nessa matriz, na primeira linha e primeira coluna é apresentado o número de verdadeiros negativos, na primeira linha e segunda coluna o número de falsos positivos, na segunda linha e primeira coluna o número de falsos negativos e na segunda linha e segunda coluna o número de verdadeiros positivos. Em um total de 58 amostras por classe, o número de verdadeiros negativos foi de 57, o número de falsos positivos foi de 1, o número de falsos negativos foi de 3 e o número de verdadeiros positivos foi de 55.

A soma dos números apresentados nas 4 células da matriz de confusão descrita acima corresponde ao valor 116, o qual representa 10% do banco de amostras de imagens. Esse percentual foi reservado para os testes de generalização de modelo do conjunto de testes final.

Tabela 4.16: Matriz de Confusão gerada a partir do teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo KNN.

Matriz de Confusão	
57	1
3	55

Como resultado da execução do procedimento de Busca de Grade configurado para o algoritmo SVM, o modelo composto pelos hiper-parâmetros *kernel* (núcleo) igual a ‘rbf’ (núcleo com função baseada em raios), *c* (penalidade do termo de erro da função do núcleo) igual a 10 e *gamma* (coeficiente de ajuste do núcleo) igual a 0,0001 obteve o melhor desempenho de classificação desse segundo procedimento de seleção de modelos. Outros parâmetros utilizados e executados em sua configuração padrão do Scikit-Learn foram: *cache size* (tamanho da *cache*) com o valor igual a 200, *class weight* (peso da classe) com o valor igual a ‘Nome’, *coef0* (coeficiente 0) com o valor igual a 0, *degree* (grau) com o valor igual a 3, *max iter* (máximo de iterações) com o valor igual a -1, *probability* (probabilidade) com o valor igual a False, *shrinking* (encolhimento) com o valor igual a True, *tol* (tolerância) com o valor igual a 0,001 e *verbose* (verbosidade) com o valor igual a False. Nesse procedimento de aprimoramento, a maior acurácia alcançada, considerando o processo de validação cruzada, foi de 98,17% e teve 2,54% como desvio padrão.

A Tabela 4.17 mostra o relatório do teste final de generalização para o modelo de aprendizagem de máquina selecionado, utilizando o algoritmo SVM. Nessa tabela, as linhas e colunas apresentam definições de nomenclatura correspondentes as definições descritas para a Tabela 4.15. Como resultado das três primeiras colunas, as médias aritméticas de precisão, sensibilidade e *F1-score* obtiveram no mesmo valor, 99,00%. Essas médias foram computadas considerando ambas as classes utilizadas em um momento como classificações positivas e em outro momento como negativas. Nas quatro linhas após essa tabela encontram-se listadas informações considerando “Diaphorina” como classe positiva. Essas informações possuem equivalência de nomenclatura com as informações apresentadas na listagem da Tabela 4.15. Nesse teste de generalização houve 115 classificações de amostras (imagens) corretas em um total de 116 (1 erro de classificação), a acurácia alcançada foi de 99,14%, a precisão foi de 98,31%, a sensibilidade de foi de 100,00%, e 99,15% em *F1-score*.

Tabela 4.17: Relatório de teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo SVM.

	Precisão	Sensibilidade	F1-Score	Suporte
Outros (0)	100%	98%	99%	58
Diaphorina (1)	98%	100%	99%	58
AVG / Total	99%	99%	99%	116
Número de Classificações Corretas: 115/116 (Erro: 1)				
Acurácia: 99,14%				
Precisão: 98,31%				
Sensibilidade: 100,00%				
F1-Score: 99,15%				

Uma matriz de confusão é apresentada na Tabela 4.18 e mostra os resultados referentes ao procedimento de teste de generalização final para o modelo aprendiza-

gem de máquina selecionado utilizando Busca em Grade para o algoritmo SVM. As descrições das informações das linhas e colunas desse matriz foram apresentadas anteriormente (no texto anterior à Tabela 4.16). Em um total de 58 amostras por classe, o número de verdadeiros negativos foi de 57, o número de falsos positivos foi de 1, o número de falsos negativos foi de 0 e o número de verdadeiros positivos foi de 58.

De maneira equivalente a matriz de confusão apresentada para o algoritmo KNN, a soma dos números apresentados nas 4 células da matriz de confusão descrita acima (matriz de confusão para o algoritmo SVM) corresponde ao valor 116, o qual representa 10% do banco de amostras de imagens. Esse percentual foi reservado para os testes de generalização de modelo do conjunto de testes final.

Tabela 4.18: Matriz de Confusão gerada a partir do teste de Generalização final de modelo de Aprendizagem de Máquina para o algoritmo SVM.

Matriz de Confusão	
57	1
0	58

4.4.1 Discussão dos Resultados

Os tempos médios dos procedimentos individuais de extração, agrupamento e classificação de características (etapas das abordagens de classificação de características) não superaram 1 segundo, na maioria dos casos. Esses resultados de medição de tempo demonstraram a viabilidade de se empregar tais técnicas em sistemas computacionais a custos aceitáveis de tempo de processamento.

No conjunto de experimentos executados com o objetivo de determinar a melhor abordagem de extração (detecção de pontos mais descrição) de características, o detector de pontos do algoritmo SURF com o descritor de características do algoritmo SIFT tiveram os melhores resultados de desempenho de classificação.

A abordagem de bolsas de características que obteve o melhor desempenho de classificação foi a abordagem que agrupou características apenas do inseto *Diaphorina citri*, analisando os resultados gerados pela abordagem de extração de características SURF/SIFT nas 3 abordagens de bolsas de características testadas.

Uma vez determinadas, e fixadas, as melhores abordagens de extração de características e de bolsas de características, testes foram realizados com o objetivo de determinar o melhor número de agrupamentos na abordagem de bolsas de características selecionada, tanto para o algoritmo KNN quanto para o algoritmo SVM. Para o algoritmo KNN, o número de agrupamentos que atingiu o melhor resultado de desempenho de classificação, dentro do intervalo testado, foi 275. Para o algoritmo SVM, esse número de agrupamentos aprimorados foi 700.

Após a determinação dos números de agrupamentos de bolsas de características que geraram os melhores desempenhos de classificação, testes de aprimoramento (seleção) de modelos de aprendizagem de máquina foram executados com o objetivo de encontrar modelos com melhores desempenhos de classificação, tanto para o algoritmo KNN quanto para o algoritmo SVM. Para o algoritmo KNN foram testados 27360 modelos e para o algoritmo SVM foram testados 650 modelos. No procedimento de seleção de modelo para o algoritmo KNN, a maior acurácia atingida no processo de validação cruzada foi de 93,73% e teve 3,82% como desvio padrão e a acurácia do teste final de generalização do modelo foi de 96,55%. No procedimento de seleção de modelo para o algoritmo SVM, a maior acurácia atingida no processo de validação cruzada foi de 98,17% e teve 2,54% como desvio padrão e a acurácia do teste final de generalização do modelo foi de 99,14%.

O algoritmo KNN utilizando o modelo composto pelos hiper-parâmetros k (n neighbors ou número de vizinhos mais próximos) igual a 7, $weights$ (pesos) igual a ‘uniform’ (uniforme), $leaf\ size$ (tamanho da folha) igual a 25, $algorithm$ (algoritmo) igual a ‘auto’ (automático) e p (métricas de distância) igual a 2 e o algoritmo SVM utilizando o modelo composto pelos hiper-parâmetros $kernel$ (núcleo) igual a ‘rbf’ (núcleo com função baseada em raios), c (penalidade do termo de erro da função do núcleo) igual a 10 e $gamma$ (coeficiente de ajuste do núcleo) igual a 0,0001 obtiveram os melhores desempenhos de classificação.

Então, a melhor abordagem de classificação de imagens do inseto *Diaphorina citri* encontrada foi a abordagem composta por: extração de características SURF/SIFT com bolsas de características com somente características do inseto *Diaphorina citri* utilizando 700 grupos e algoritmo SVM com o modelo anteriormente mencionado.

Capítulo 5

Considerações Finais

*“Não me venham com conclusões!
A única conclusão é morrer.”*

– Fernando Pessoa

Esta pesquisa teve o objetivo de realizar um estudo que possibilitasse a classificação de psíldeos dos citros asiáticos (*Diaphorina citri* Kuwayama) por procedimento computacional, distinguindo-os de outros insetos que também possam ser capturados por armadilhas adesivas amarelas, através de imagens de insetos extraídas dessas armadilhas.

Para a extração de características foram testados os algoritmos ORB, SIFT, SURF, BRISK e FREAK com os seus detectores de pontos próprios e com o detector de pontos do algoritmo SURF.

Para o agrupamento de características foi testado o algoritmo *Mini Batch K-Means* e foi variado o número de agrupamentos neste algoritmo com o objetivo de encontrar quantidades de agrupamentos que aprimorassem os resultados de desempenho de classificação.

Para a classificação de características foram testados os algoritmos de aprendizagem de máquina KNN e SVM. Procedimentos de seleção de modelo de aprendizagem de máquina, utilizando o método de Busca em Grade, foram executados para encontrar modelos aprimorados, tanto para o algoritmo KNN quanto para o SVM. No aprimoramento de modelo para o algoritmo KNN, diversos valores em hiper-parâmetros foram testados. Para o algoritmo SVM, além da variação de valores em hiper-parâmetros, foram testados os núcleos linear e baseado em raios. Todos os resultados foram gerados utilizando um procedimento padronizado de validação cruzada.

Baseando-se no trabalho de Leonardo [2014], que avaliou a acurácia do processo manual de inspeção em armadilhas adesivas amarelas, alcançando aproximadamente

93%, os resultados desta pesquisa superaram os resultados apresentados nesse processo de contagem manual. No procedimento computacional de classificação de imagens de insetos *Diaphorina citri* descoberto foi atingida a acurácia de 98,17% utilizando validação cruzada e 99,14% de acurácia no teste final de generalização do modelo de SVM selecionado, portanto, alcançando o objetivo proposto.

5.1 Aplicações

O resultado alcançado por esta pesquisa é de grande importância para a citricultura e para o controle do HLB. Foi descoberto um procedimento computacional aprimorado para o reconhecimento de insetos *Diaphorina citri* em imagens digitais extraídas de armadilhas adesivas amarelas. As taxas de acurácia de classificação (reconhecimento) alcançadas no procedimento computacional descoberto são superiores as taxas alcançadas no procedimento manual praticado na citricultura paulista [Leonardo 2014]. Esse procedimento computacional viabiliza a construção de sistemas computacionais que resultam na automação do processo de reconhecimento desses insetos nessas armadilhas e podem reduzir consideravelmente os custos operacionais, imprecisões e instabilidades de resultados de classificação associados ao processo manual, resultando em uma economia relevante de tempo e, consequentemente, dinheiro.

Na construção de um sistema computacional para automatizar a contagem de insetos *Diaphorina citri* poderá ser aplicado o procedimento computacional aprimorado descoberto, gerando informações, com alta acurácia, das quantidades desses insetos presentes nas armadilhas adesivas amarelas. Esse sistema computacional poderá integrar-se ao sistema de informações de monitoramento populacional de insetos *Diaphorina citri* da citricultura paulista que encontra-se atualmente em funcionamento, automatizando assim todo o processo de monitoramento em larga escala do inseto vetor do *Huanglongbing* (HLB) [Fundecitrus 2015]. Essa integração de sistemas poderá proporcionar um monitoramento mais ágil e preciso do principal vetor do HLB, resultando no aumento da acurácia na aplicação de agrotóxicos em larga escala e economizando, consideravelmente, recursos financeiros.

5.2 Pesquisas Futuras

Em uma continuidade direta desta pesquisa, poderão ser realizados testes considerando o algoritmo de aprendizagem de máquina *Support Vector Machine* (SVM) [Scikit-learn 2014e]. Nesses testes poderão ser avaliados os desempenhos de classificação, pela métrica da acurácia, para modelos de SVM com núcleos polinomiais, utilizando os mesmos procedimentos de validação cruzada e aprimoramento de modelo (Busca em Grade) [Scikit-learn 2014b]. Além disso, o trabalho de Louppe

e Geurts [2012] demonstrou desempenhos de classificação promissores ao utilizar técnicas de combinação de classificadores, sendo assim um caminho a ser experimentado em busca de melhores resultados.

Poderão ser adquiridas imagens de armadilhas adesivas amarelas completas com os insetos capturados. Então, poderão ser reconhecidos esses insetos através de técnicas de processamento de imagem a serem pesquisadas juntamente com o procedimento de classificação aprimorado descoberto nesta pesquisa. Assim, poderão ser informadas as presenças do inseto *Diaphorina citri* nessas armadilhas e, em seguida, contadas as suas ocorrências. Com isso, um sistema computacional poderá ser criado para realizar a automação da tarefa de contagem de insetos *Diaphorina citri* em armadilhas adesivas amarelas.

Referências Bibliográficas

- [Agnelli 2011] Agnelli, A. R. (2011). Potencial de agentes indutores de resistência para o controle da bactéria *Candidatus Liberibacter asiaticus* em plantas cítricas. Master's thesis, Fundo de Defesa da Citricultura - FUNDECITRUS.
- [Agrawal et al. 2008] Agrawal, M., Konolige, K., e Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pp. 102–115. Springer.
- [Alahi et al. 2012] Alahi, A., Ortiz, R., e Vanderghenst, P. (2012). Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517. Ieee.
- [Bay et al. 2008] Bay, H., Ess, A., Tuytelaars, T., e Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.
- [Bay et al. 2006] Bay, H., Tuytelaars, T., e Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pp. 404–417. Springer.
- [Belasque Junior et al. 2009] Belasque Junior, J., Bergamin Filho, A., Bassanezi, R. B., Barbosa, J. C., Fernandes, N. G., Yamamoto, P. T., Lopes, S. A., Machado, M. A., Leite Junior, R. P., Ayres, A. J., et al. (2009). Base científica para a erradicação de plantas sintomáticas e assintomáticas de huanglongbing (hlb, greening) visando o controle efetivo da doença. *Tropical Plant Pathology*, 34(3):137–145.
- [Bieras e Santos 2003] Bieras, A. R. e Santos, M. (2003). Condições climáticas e incidência de pragas e doenças na cultura de citros nas principais regiões produtoras do estado de são paulo. *Ambientes estudos de Geografia*.
- [Boiman et al. 2008] Boiman, O., Shechtman, E., e Irani, M. (2008). In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8. IEEE.
- [Bosch et al. 2007] Bosch, A., Muñoz, X., e Martí, R. (2007). Which is the best way to organize/classify images by content? *Image and vision computing*, 25(6):778–791.
- [Bové et al. 1996] Bové, J., Chau, N. M., Trung, H. M., Bourdeaut, J., e Garnier, M. (1996). Huanglongbing (greening) in viet nam: Detection of liberobacter

- asiaticum by dna-hybridization with probe in 2.6 and pcr-amplification of 16s ribosomal dna. In *Proc. 13th Conference of the International Organization of Citrus Virologists (IOCV)*. University of California, pp. 258–266.
- [Bové 2006] Bové, J. M. (2006). Huanglongbing: a destructive, newly-emerging, century-old disease of citrus. *Journal of plant pathology*.
- [Bradski e Kaehler 2008] Bradski, G. e Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc."
- [Calonder et al. 2010] Calonder, M., Lepetit, V., Strecha, C., e Fua, P. (2010). Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pp. 778–792.
- [Capoor et al. 1974] Capoor, S., Rao, D., e Viswanath, S. (1974). Greening disease of citrus in the deccan trap country and its relationship with the vector, diaphorina citri kuwayama. In *Proceedings of the 6th Conference, International Organization of Citrus Virologists*, pp. 43–49.
- [Catling et al. 1970] Catling, H. et al. (1970). Distribution of the psyllid vectors of citrus greening disease, with notes on the biology and bionomics of diaphorina citri. *FAO Plant Protection Bulletin*, 18(8).
- [Chaplot et al. 2006] Chaplot, S., Patnaik, L., e Jagannathan, N. (2006). Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network. *Biomedical Signal Processing and Control*, 1(1):86–92.
- [Dara 2011] Dara, S. (2011). Asian citrus psyllid threat to santa barbara county. <http://ucanr.edu/blogs/blogcore/postdetail.cfm?postnum=4165>. Acessado: 2015-07-10.
- [Dejours 1997] Dejours, C. (1997). *Fator Humano, O*. FGV Editora.
- [Derntl 2014] Derntl, A. (2014). Survey of feature detectors and descriptors in surgical domain. In *IEEE GSC*.
- [Ebrahimi e Mayol-Cuevas 2009] Ebrahimi, M. e Mayol-Cuevas, W. W. (2009). Sussure: Speeded up surround extrema feature detector and descriptor for realtime applications. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 9–14. IEEE.
- [Ferreira 2010] Ferreira, A. B. d. H. (2010). *Míni Aurélio. O dicionário língua portuguesa*. Positivo.
- [French et al. 2001] French, J., Kahlke, C., e Da Graça, J. (2001). First record of the asian citrus psylla, diaphorina citri kuwayama (homoptera: Psyllidae) in texas. *Subtropical Plant Science*, 53:14–15.
- [Fundecitrus 2015] Fundecitrus, R. (2015). Alerta fitossanitário. <http://alerta.fundecitrus.com.br/>. Acessado: 2016-02-08.

- [Garnier e Bové 1993] Garnier, M. e Bové, J.-M. (1993). Citrus greening disease. In *Proceedings of the 12th Conference, International Organization of Citrus Virologists*, pp. 212–219.
- [Garreta e Moncecchi 2013] Garreta, R. e Moncecchi, G. (2013). *Learning scikit-learn: Machine Learning in Python*. Packt Publishing Ltd.
- [Gonzalez e Woods 2010] Gonzalez, R. C. e Woods, R. E. (2010). *Processamento de imagens digitais*. Pearson.
- [Grafton-Cardwell et al. 2005] Grafton-Cardwell, E. E., Godfrey, K., Rogers, M. E., Childers, C. C., e Stansly, P. A. (2005). *Asian citrus psyllid*. UCANR Publications.
- [Grana et al. 2013] Grana, C., Borghesani, D., Manfredi, M., e Cucchiara, R. (2013). A fast approach for integrating orb descriptors in the bag of words model. In *IS&T/SPIE Electronic Imaging*, pp. 866709–866709. International Society for Optics and Photonics.
- [Hackeling 2014] Hackeling, G. (2014). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd.
- [Halbert e Manjunath 2004] Halbert, S. E. e Manjunath, K. L. (2004). Asian citrus psyllids (sternorrhyncha: Psyllidae) and greening disease of citrus: a literature review and assessment of risk in florida. *Florida Entomologist*, 87(3):330–353.
- [Halbert et al. 2002] Halbert, S. E., Niblett, C., Manjunath, K., Lee, R. F., e Brown, L. G. (2002). Establishment of two new vectors of citrus pathogens in florida. In *Proc. International Soc. Citriculture IX Congress, ASHS Press, Alexandria, VA*, pp. 1016–1017.
- [Harrington 2012] Harrington, P. (2012). *Machine learning in action*. Manning Publications Co.
- [Harris e Stephens 1988] Harris, C. e Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, pp. 50. Citeseer.
- [Hmeidi et al. 2008] Hmeidi, I., Hawashin, B., e El-Qawasmeh, E. (2008). Performance of knn and svm classifiers on full word arabic articles. *Advanced Engineering Informatics*, 22(1):106–111.
- [Initiative et al. 2006] Initiative, O. S. et al. (2006). The bsd license.
- [James et al. 2013] James, G., Witten, D., Hastie, T., e Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.
- [Juan e Gwun 2009] Juan, L. e Gwun, O. (2009). A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152.
- [Khvedchenia 2012] Khvedchenia, I. (2012). A battle of three descriptors: Surf, freak and brisk. <http://computer-vision-talks.com/articles/2012-08-18-a-battle-of-three-descriptors-surf-freak-and-brisk/>. Acessado: 2015-11-24.

- [Kotsia e Pitas 2007] Kotsia, I. e Pitas, I. (2007). Facial expression recognition in image sequences using geometric deformation features and support vector machines. *Image Processing, IEEE Transactions on*, 16(1):172–187.
- [Larios et al. 2008] Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., Moldenke, A., Lytle, D. A., Correa, S. R., Mortensen, E. N., et al. (2008). Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects. *Machine Vision and Applications*, 19(2):105–123.
- [Larios et al. 2011] Larios, N., Lin, J., Zhang, M., Lytle, D., Moldenke, A., Shapiro, L., e Dietterich, T. (2011). Stacked spatial-pyramid kernel: An object-class recognition method to combine scores from random trees. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pp. 329–335. IEEE.
- [Le-Qing e Zhen 2010] Le-Qing, Z. e Zhen, Z. (2010). Auto-classification of insect images based on color histogram and glcm. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, volume 6, pp. 2589–2593. IEEE.
- [Leonardo 2014] Leonardo, A. (2014). Otimização da leitura de cartão adesivo amarelo para o monitoramento de adultos de diaphorina citri kuwayama (hemiptera: Liviidae). Master’s thesis, Fundo de Defesa da Citricultura - FUNDECITRUS.
- [Leutenegger et al. 2011] Leutenegger, S., Chli, M., e Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555. IEEE.
- [Lopes et al. 2011] Lopes, J., Déo, T., Andrade, B. M., Giroto, M., Felipe, A., Junior, C., Bueno, C., Silva, T., e Lima, F. (2011). Importância econômica do citros no brasil. *Rev. Cient. Eletrônica de Agronomia*, 20(1).
- [Lopes et al. 2007] Lopes, S. A., Frare, G. F., Yamamoto, P. T., Ayres, A. J., e Barbosa, J. C. (2007). Ineffectiveness of pruning to control citrus huanglongbing caused by candidatus liberibacter americanus. *European Journal of Plant Pathology*, 119(4):463–468.
- [Louppe e Geurts 2012] Louppe, G. e Geurts, P. (2012). Ensembles on random patches. In *Machine Learning and Knowledge Discovery in Databases*, pp. 346–361. Springer.
- [Lowe 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- [Lu et al. 2012] Lu, A., Hou, X., Liu, C.-L., e Chen, X. (2012). Insect species recognition using discriminative local soft coding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1221–1224. IEEE.
- [Mead e Fasulo 1998] Mead, F. W. e Fasulo, T. (1998). Asian citrus psyllid, diaphorina citri kuwayama (insecta: Hemiptera: Psyllidae). <http://entnemdept.ufl.edu/creatures/citrus/acpsyllid.htm>. Acessado: 2015-07-09.

- [Mendes 2015] Mendes, M. (2015). *Agriannual 2015: Anuário da Agricultura Brasileira*. Informa Economics FNP.
- [Michaud 2004] Michaud, J. (2004). Natural mortality of asian citrus psyllid (homoptera: Psyllidae) in central florida. *Biological Control*, 29(2):260–269.
- [Moreno et al. 2009] Moreno, P., Bernardino, A., e Santos-Victor, J. (2009). Improving the sift descriptor with smooth derivative filters. *Pattern Recognition Letters*, 30(1):18–26.
- [Mundada e Gohokar 2013] Mundada, R. G. e Gohokar, V. (2013). Detection and classification of pests in greenhouse using image processing. *IOSR Journal of Electronics and Communication Engineering*, 5(6):57–63.
- [Neves et al. 2010] Neves, M. F., Trombin, V., Milan, P., Lopes, F., Cressoni, F., e Kalaki, R. (2010). O retrato da citricultura brasileira. *CitrusBR*.
- [Oliveira 2013] Oliveira, J. M. C. d. (2013). Diaphorina citri e candidatus liberibacter spp: Associação que coloca em risco a citricultura baiana.
- [Pang et al. 2012] Pang, Y., Li, W., Yuan, Y., e Pan, J. (2012). Fully affine invariant surf for image matching. *Neurocomputing*, 85:6–10.
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- [Redondi et al. 2013] Redondi, A., Baroffio, L., Canclini, A., Cesana, M., e Tagliasacchi, M. (2013). A visual sensor network for object recognition: Testbed realization. In *DSP*, pp. 1–6.
- [Robnett 2014] Robnett, M. S. (2014). Disease-carrying insect threatens yuma citrus industry. <http://kawc.org/post/disease-carrying-insect-threatens-yuma-citrus-industry>. Acessado: 2015-07-09.
- [Roistacher 1991] Roistacher, C. (1991). Techniques for biological detection of specific citrus graft transmissible diseases. *Greening*. *FAO, Rome*, 286:35–45.
- [Roistacher 1996] Roistacher, C. (1996). The economics of living with citrus diseases: huanglongbing (greening) in thailand. In *Proc. 13th Conference of the International Organization of Citrus Virologists (IOCV)*. *University of California, Riverside*, pp. 279–285.
- [Rosin 1999] Rosin, P. L. (1999). Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307.
- [Rosten e Drummond 2006] Rosten, E. e Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pp. 430–443. Springer.

- [Rublee et al. 2011] Rublee, E., Rabaud, V., Konolige, K., e Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571. IEEE.
- [Russel e Norvig 2013] Russel, S. e Norvig, P. (2013). Inteligência artificial. *Editora Campus*.
- [Sala 2011] Sala, I. (2011). Avaliação do tempo de exposição em armadilha adesiva amarela e das condições de armazenamento de adultos de diaphorina citri na detecção de candidatus liberibacter asiaticus. Master’s thesis, Fundo de Defesa da Citricultura.
- [Salibe e Cortez 1966] Salibe, A. A. e Cortez, R. E. (1966). Studies on the leaf mottling disease of citrus in the philippines. *FAO Plant Protection Bulletin*, 14:141–144.
- [Samanta e Ghosh 2012] Samanta, R. e Ghosh, I. (2012). Tea insect pests classification based on artificial neural networks. *International Journal of Computer Engineering Science (IJCES)*, 2(6).
- [Schaeffer 2013] Schaeffer, C. (2013). A comparison of keypoint descriptors in the context of pedestrian detection: Freak vs. surf vs. brisk.
- [Scikit-learn 2014a] Scikit-learn, D. (2014a). Comparison of the k-means and mini batch k-means clustering algorithms. http://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html. Acessado: 2015-04-30.
- [Scikit-learn 2014b] Scikit-learn, D. (2014b). Parameter estimation using grid search with cross-validation. http://scikit-learn.org/stable/auto_examples/model_selection/grid_search_digits.html#example-model-selection-grid-search-digits-py. Acessado: 2015-12-14.
- [Scikit-learn 2014c] Scikit-learn, D. (2014c). Rbf svm parameters. http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html. Acessado: 2015-12-16.
- [Scikit-learn 2014d] Scikit-learn, D. (2014d). Sklearn neighbors kneighbors-classifier. <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>. Acessado: 2015-12-16.
- [Scikit-learn 2014e] Scikit-learn, D. (2014e). Sklearn svc. <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>. Acessado: 2015-12-16.
- [Shiguo et al. 2009] Shiguo, H., Mingquan, Z., Guohua, G., e Xiuli, W. (2009). Ontology-based insect recognition. In *Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on*, pp. 176–178. IEEE.
- [Teixeira et al. 2005] Teixeira, D. C., Saillard, C., Eveillard, S., Danet, J. L., da Costa, P. I., Ayres, A. J., e Bové, J. (2005). ‘candidatus liberibacter americanus’, associated with citrus huanglongbing (greening disease) in são paulo

- state, brazil. *International Journal of Systematic and Evolutionary Microbiology*, 55(Pt 5):1857–1862.
- [Tsai 2012] Tsai, C.-F. (2012). Bag-of-words representation in image annotation: A review. *International Scholarly Research Notices*, 2012.
- [Van Der Walt et al. 2014] Van Der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., e Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.
- [Venugoban e Ramanan 2014] Venugoban, K. e Ramanan, A. (2014). Image classification of paddy field insect pests using gradient-based features. *International Journal of Machine Learning and Computing*, 4(1).
- [Viswanathan 2009] Viswanathan, D. G. (2009). Features from accelerated segment test (fast).
- [Warnert et al. 2012] Warnert, J. et al. (2012). Research news: Asian citrus psyllid and huanglongbing disease threaten california citrus. *California Agriculture*, 66(4):127–130.
- [Webb 2010] Webb, C. M. (2010). Asian citrus psyllid task force. <http://ucanr.edu/blogs/blogcore/postdetail.cfm?postnum=2339>. Acessado: 2015-07-10.
- [Xiao-Lin et al. 2009] Xiao-Lin, L., Shi-Guo, H., Ming-Quan, Z., e Guo-Hua, G. (2009). Knn-spectral regression lda for insect recognition. In *Information Science and Engineering (ICISE), 2009 1st International Conference on*, pp. 1315–1318. IEEE.
- [Yamamoto et al. 2001] Yamamoto, P. T., Pria Jr, W., Roberto, S. R., Fellipe, M., e Freitas, E. (2001). Flutuação populacional de cigarrinhas (hemiptera: Cicadellidae) em pomar cítrico em formação. *Neotrop. Entomol.*, 30(1):175–177.
- [Yang et al. 2010] Yang, H., Liu, W., Xing, K., Qia, J., Wang, X., Gao, L., e Shen, Z. (2010). Research on insect identification based on pattern recognition technology. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 2, pp. 545–548. IEEE.
- [Zhang e Hu 2010] Zhang, H. e Hu, Y. (2010). Extension theory for classification of the stored-grain insects. In *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, pp. 758–760. IEEE.