

---

**Método automático para detecção de  
nematóides em lavoura cafeeira  
usando imagens aéreas**

---

**Alexandre de Jesus Oliveira**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**Alexandre de Jesus Oliveira**

**Método automático para detecção de  
nematóides em lavoura cafeeira  
usando imagens aéreas**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Jefferson Rodrigo de Souza

Coorientador: Elaine Ribeiro de Faria Paiva

Uberlândia

2019

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU  
com dados informados pelo(a) próprio(a) autor(a).

O48 Oliveira, Alexandre de Jesus, 1991-  
2019 Método automático para detecção de nematóides em lavoura  
cafeeira [recurso eletrônico] / Alexandre de Jesus Oliveira. - 2019.

Orientador: Jefferson Rodrigo de Souza.  
Coorientadora: Elaine Ribeiro de Faria Paiva.  
Dissertação (Mestrado) - Universidade Federal de Uberlândia,  
Pós-graduação em Ciência da Computação.  
Modo de acesso: Internet.  
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2019.2314>  
Inclui bibliografia.  
Inclui ilustrações.

1. Computação. I. de Souza, Jefferson Rodrigo, 1985-, (Orient.).  
II. Ribeiro de Faria Paiva, Elaine, 1980-, (Coorient.). III.  
Universidade Federal de Uberlândia. Pós-graduação em Ciência da  
Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:  
Gizele Cristine Nunes do Couto - CRB6/2091  
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

**ATA DE DEFESA**

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado Acadêmico, 05/2019, PPGCO				
Data:	8 de julho de 2019	Hora de início:	14:00	Hora de encerramento:	16:35
Matrícula do Discente:	11722CCP001				
Nome do Discente:	Alexandre de Jesus Oliveira				
Título do Trabalho:	Detecção automática de nematoides em lavoura cafeeira usando imagens aéreas				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se na sala 1B132, Bloco 1B, Campus Santa Mônica, da Universidade Federal de Uberlândia, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Maria Camila Nardini Barioni - FACOM/UFU, João Fernando Mari - CRP/UFV, Elaine Ribeiro de Faria Paiva - FACOM/UFU (Coorientadora) e Jefferson Rodrigo de Souza - FACOM/UFU, orientador do candidato.

Ressalta-se que o Prof. Dr. João Fernando Mari participou da defesa por meio de videoconferência desde a cidade de Rio Paranaíba-MG. Os outros membros da banca e o aluno participaram in loco.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Jefferson Rodrigo de Souza, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

**Aprovado.**

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Jefferson Rodrigo de Souza, Professor(a) do Magistério Superior**, em 09/07/2019, às 09:20, conforme horário oficial de Brasília, com fundamento no art. 6º,



§ 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Elaine Ribeiro de Faria Paiva, Professor(a) do Magistério Superior**, em 09/07/2019, às 09:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Maria Camila Nardini Barioni, Professor(a) do Magistério Superior**, em 09/07/2019, às 10:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **João Fernando Mari, Usuário Externo**, em 12/07/2019, às 10:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1376534** e o código CRC **05688A8F**.

*Dedico este trabalho aos meus pais, Ricardo e Claudemira  
e minha irmã Ana Carolina por todo apoio, amor e incentivo durante esta jornada.*





---

# Agradecimentos

Agradeço imensamente a Deus que me ampara a cada passo desta jornada.

Minha gratidão ao meu orientador Prof<sup>o</sup>. Jefferson Rodrigo de Souza pela oportunidade de trabalhar e me aprofundar nesta temática, pelos ensinamentos, incentivo e disponibilidade.

Meu sincero agradecimento a minha coorientadora Prof<sup>a</sup>. Elaine Ribeiro Faria pelos ensinamentos, paciência e incentivo, além dos *feedbacks* que foram fundamentais para o meu crescimento.

Obrigado aos professores e funcionários da PPGCO, em especial aos professores Fabiano Dorça, Gina Maira Barbosa, Márcia Aparecida e André Backes e também ao Erisvaldo e a Sônia pelo grande auxílio nas questões administrativas.

Minha profunda gratidão a minha família Ana Carolina, Claudemira e Ricardo pelo carinho, cuidado e amor. Em especial, a minha mãe Claudemira pelo empenho e apoio incondicional em toda minha carreira acadêmica.

Obrigado aos meus amigos, em especial ao Antônio Martins pelas valiosas conversas e conselhos, além de toda a amizade.

Agradeço à CNPq pelo apoio financeiro durante a realização do trabalho.

A todos, o meu muito obrigado!



*“A última medida de um homem não é onde ele se posiciona em momentos de conforto e conveniência, mas onde ele se posiciona em momentos de desafio e controvérsia.”*  
*(Martin Luther King Jr.)*



---

## Resumo

A agricultura de precisão apresenta diversos desafios, dentre eles a detecção de doenças e pragas em ambientes agrícolas. Este trabalho descreve um método automático para a detecção de nematóides em lavouras cafeeiras usando imagens aéreas. Atualmente, a coleta de imagens aéreas aliada ao uso de técnicas computacionais baseadas em aprendizado de máquina (AM) apresentam um grande potencial em tarefas envolvendo a detecção de pragas em diversas culturas, como eucalipto, soja, entre outras. No entanto, há uma escassez de trabalhos para a cultura cafeeira. Neste trabalho é usado um veículo aéreo não tripulado (VANT) para obter imagens RGB de alta resolução de uma plantação cafeeira. O método proposto utiliza duas estratégias distintas para a extração de características e identificação das regiões com a presença dos nematóides. A primeira estratégia é baseada em blocos com os algoritmos KNN, RF e SVM para segmentação das regiões em praga e não-praga. A segunda estratégia é baseada em redes neurais convolucionais (em inglês *Convolutional Neural Network* (CNN)) como as arquiteturas do estado da arte U-Net e PSPNet para a segmentação das regiões em regiões com praga, saudáveis e solo. Também avaliou-se a influência na variação da altura utilizando a arquitetura U-Net. Os resultados desta pesquisa científica demonstram a viabilidade do método proposto, com uma média para a medida F de 0,64 para o RF usando validação cruzada com 10-pastas e 0,69 como média para a medida F usando a arquitetura U-Net para o conjunto de teste.

**Palavras-chave:** Detecção, Nematóides, Café, VANT, U-Net.



---

# Abstract

Precision agriculture presents several challenges, amongst them the detection of diseases and pests in agricultural environments. This work describes an automatic method for the detection of nematodes pest in coffee crops using aerial images. Currently, aerial imagery collection combined with the use of machine learning (ML) based computational techniques have great potential in tasks involving pest detection in several crops, such as eucalyptus, soybean, among others. However, there is a scarcity of studies for the coffee culture. The proposed method uses two distinct strategies for the feature extraction and identification of the regions with the presence of nematodes. The first one based on blocks with the KNN, RF and SVM algorithms to classify the regions in pest and non-pest. The second one based on Convolutional Neural Networks (CNN) with state-of-the-art architectures U-Net and PSPNet to classify areas into healthy, pest and soil. The influence on height variation was also evaluated using the U-Net architecture. Results demonstrate the viability of the proposed method, with an average F-measure of 0.64 for the RF using 10-fold cross validation and an average F-measure of 0.69 for the U-Net architecture for the test set.

**Keywords:** Detection, Pest, Nematodes, Coffee, UAV, U-Net.





---

## Lista de ilustrações

Figura 1	– (a) Raízes de café com galhas atípicas causadas por <i>M. exigua</i> (CHAVES, 2019). (b) Sintomas resultantes do parasitismo ao cafeeiro como queda de folhas, amarelamento e definhamento (STARBUCKS, 2016).	24
Figura 2	– À esquerda, folha com sintoma de cercóspora (manchas marrom). À direita, avaliação das regiões com sintoma da doença pelo software <i>Leaf Doctor</i> . Imagem adaptada de (PETHYBRIDGE; NELSON, 2015).	30
Figura 3	– Segmentação de imagens digitais por meio do método de Otsu. Imagem adaptada de (VASSALLO-BARCO et al., 2017).	34
Figura 4	– (a) Imagem de folhas doentes com sintomas, representada pelo espaço de cor RGB. (b) Imagem de folhas doentes com sintomas, representada pelo canal do espaço de cor $L^*a^*b^*$ , que destaca sintomas e veias foliares. Fonte: (BARBEDO, 2014)	35
Figura 5	– Resultado dos diferentes algoritmos de segmentação: (a) Imagem original, (b) algoritmo de agrupamento <i>k-means</i> , (c) algoritmo <i>Otsu</i> e (d) algoritmo proposto pelo autor. Fonte: (MA et al., 2017).	37
Figura 6	– Abordagem baseada em blocos para extração de características. Fonte: (SOUZA et al., 2015).	38
Figura 7	– (a) Imagem original particionada por meio de uma grade de tamanho 4x4. (b) Imagem dividida após aplicação do algoritmo de <i>Canny</i> . (c) A imagem em tons de cinza (à esquerda) é quantizada (à direita). (d) A imagem quantizada (à esquerda) é particionada em 64 segmentos (à direita). Fonte: Adaptado de (VASSALLO-BARCO et al., 2017).	40
Figura 8	– Tipos de aprendizado: (a) Supervisionado. (b) Não supervisionado. (c) Semi-supervisionado. Fonte: Elaborada pelo autor	41
Figura 9	– Tarefas envolvendo imagens. Fonte: Adaptado de (KARPATHY et al., 2016)	42
Figura 10	– Classificação de novas instâncias usando KNN. Fonte: Elaborada pelo autor.	43

Figura 11 – Classificação de novas instâncias usando RF. Fonte: Elaborada pelo autor. . . . .	44
Figura 12 – Classificadores lineares; (a) Duas classes de pontos (pontos laranjados e verdes) e um separador linear. (b) Um separador linear com maior margem $m$ distante dos vetores de suporte. Fonte: Elaborada pelo autor	45
Figura 13 – Exemplo de uma operação de convolução com <i>stride</i> 1 e <i>kernel</i> 3x3. Fonte: (EREMENKO, 2018) . . . . .	47
Figura 14 – Exemplo de uma operação de deconvolução com <i>kernel</i> 3x3 e sem padding. Fonte: Adaptada de (PRÖVE, 2017) . . . . .	47
Figura 15 – Exemplo de uma operação de max pooling. Fonte: Adaptada de (KARPATHY et al., 2016) . . . . .	48
Figura 16 – Exemplo da arquitetura U-Net. Cada bloco azul corresponde a um mapa de características. O número de blocos é indicado na parte superior do bloco. O tamanho do bloco é mostrado no canto inferior. Os blocos brancos representam os mapas de característica copiados. As setas denotam diferentes operações. Fonte: (RONNEBERGER; FISCHER; BROX, 2015) . . . . .	49
Figura 17 – Visão geral da arquitetura da rede PSPNet. (a) A imagem é fornecida como entrada. (b) Usa uma CNN pré-treinada para obter o mapa de características da última camada. (c) Em seguida, um módulo PP é aplicado para obter diferentes representações de sub-regiões, seguido por camadas de upsampling e concatenação para formar a representação de característica final, que carrega informações de contexto locais e globais. (d) Finalmente, a representação é alimentada em uma camada de convolução para obter a previsão final por pixel. Fonte: (ZHAO et al., 2017) . . . . .	50
Figura 18 – Particionamento dos dados usando a técnica <i>holdout</i> . Fonte: Elaborada pelo autor. . . . .	51
Figura 19 – Exemplo de validação cruzada com $K$ -pastas ( $K = 4$ ). Fonte: (BACKES, 2018) . . . . .	51
Figura 20 – Resultado da segmentação para o GP. As regiões em vermelho representam as áreas com a presença da praga e em azul a estrada. Fonte: (SOUZA et al., 2015) . . . . .	53
Figura 21 – (a) Imagem original. (b) Anotação manual. (c) Segmentação utilizando o método 1. (d) Segmentação utilizando o método 2. (e) Segmentação usando o método proposto Fonte: Adaptado de (CHEN et al., 2018) . .	54
Figura 22 – Metodologia para detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas. Fonte: Elaborada pelo autor . . . . .	57

Figura 23 – <i>Phanton 4 Pro</i> usado para coletar as imagens. Fonte: Elaborada pelo autor . . . . .	58
Figura 24 – Vista aérea das áreas obtidas na etapa da coleta das imagens da Fazenda Novo Mundo - Monte Carmelo (MG/BR) localizada por meio das coordenadas -18°49'36.98", -47°20'19.29". (a) Áreas A e B coletadas à 20m em Abril/2018. (b) Área C coletada em 3 alturas em Dezembro/2018. Fonte: Google Earth, imagens obtidas em 19/04/2017 com altitude de 1.23km. . . . .	59
Figura 25 – Exemplo de nematóide em lavoura cafeeira. Fonte: Elaborada pelo autor	60
Figura 26 – (a) e (d) Imagem original 1 e rótulo. (b) e (e) Imagem original 2 e rótulo. (c) e (f) Imagem original 3 e rótulo. . . . .	61
Figura 27 – Exemplo de imagem rotulada pelo <i>Labelme</i> . Fonte: Elaborada pelo autor	61
Figura 28 – Exemplo de regiões que evidenciam os desafios no processo de rotulagem. Em (a) e (b) há uma dificuldade na identificação das regiões com praga, por conta da sobreposição com o solo. Em (c) e (d) a distribuição das pragas ocorre de maneira esparsa, dificultando a rotulagem. . . . .	62
Figura 29 – Exemplo de pré-processamento aplicado para uma imagem. (a) Imagem original. (b) Imagem Segmentada. . . . .	63
Figura 30 – Exemplo de uma matriz 20×15 em menor escala para o rótulo, onde as regiões de solo (vermelho) são preenchidas com valor 0, saudáveis (azul) com o valor 1, praga (verde) com o valor 2 e o restante com o valor -1. Fonte: Elaborada pelo autor . . . . .	64
Figura 31 – Processo de extração de característica usando a abordagem por bloco. A Seta vermelha indica a direção da janela deslizante. Fonte: Elaborada pelo autor . . . . .	66
Figura 32 – A etapa original de contração da U-Net (destacada em amarelo) é substituída por uma VGG16, considerando somente a parte relativa a extração de característica (destacada em amarelo). Fonte: VGG adaptada de (FERGUSON et al., 2017) e U-Net adapta de (RONNEBERGER; FISCHER; BROX, 2015) . . . . .	69
Figura 33 – Modificação na arquitetura U-Net proposta. Fonte: Elaborada pelo autor	69
Figura 34 – Arquitetura de rede PSPNet usando uma ResNet101 pré-treinada. Fonte: Adaptada de (ZHAO et al., 2017) . . . . .	70
Figura 35 – Exemplos de imagens resultantes após executar a etapa de predição, na qual as regiões em azul representam verdadeiro positivo (classe real e predita iguais a praga), regiões em lilás falso-positivo (classe real não-praga e predita igual a praga) e em vermelho falso-negativo (classe real praga e classe predita não-praga). . . . .	75

Figura 36 – Imagens resultantes após executar a etapa de predição. (a)-(d) Imagem 1 utilizando SVM para as 4 configurações de <i>BInt</i> . (e)-(h) Imagem 4 utilizando RF para as 4 configurações de <i>BInt</i> . Em azul verdadeiro positivo (classe real e predita iguais a praga), em lilás falso-positivo (classe real não-praga e predita como praga) e em vermelho falso-negativo (classe real praga e predita não-praga). . . . .	81
Figura 37 – Resultado da predição pela comparação entre KNN, RF e SVM para 40 características sem segmentação (Tabela 10). (a) Resultado imagem 3 para KNN. (b) Resultado imagem 3 para RF. (c) Resultado imagem 3 para SVM. (d) Resultado imagem 9 para KNN. (e) Resultado imagem 9 para RF. (f) Resultado imagem 3 para SVM. . . . .	87
Figura 38 – Resultado da predição para as imagens obtidas pela comparação da U-Net e PSPNet. (a) Imagem 1. (b) Resultado imagem 1 para U-Net. (c) Resultado imagem 1 para PSPNet. (d) Imagem 2. (e) Resultado imagem 2 para U-Net. (f) Resultado imagem 2 para PSPNet. (g) Imagem 3. (h) Resultado imagem 3 para U-Net e (i) Resultado imagem 3 para PSPNet . . . . .	89
Figura 39 – Resultado da predição entre os modelos gerados com 10 e 20 metros. itens (a), (d) e (g) exibe as imagens originais fornecidas para predição com 10m, 15m e 20m respectivamente. Itens (b), (e) e (h) apresenta a predição fornecida pelo modelo gerado com 10m. Itens (c), (f) e (i) a predição fornecida pelo modelo gerado com 20m . . . . .	92

---

## Lista de tabelas

Tabela 1 – Sensores para VANTs em aplicações agrícolas. Adaptado de (SHAKHATREH et al., 2018) . . . . .	32
Tabela 2 – Matriz de confusão para duas classes. Extraído de (TAN; STEINBACH; KUMAR, 2005) . . . . .	52
Tabela 3 – Comparação entre os principais trabalhos correlatos a proposta de trabalho. . . . .	55
Tabela 4 – Avaliação do desempenho dos algoritmos para cada configuração de <i>BInt</i>	80
Tabela 5 – Número de amostras por classe considerando imagens sem segmentação e com segmentação. . . . .	82
Tabela 6 – Resultado KNN, RF e SVM para 230 características . . . . .	83
Tabela 7 – Resultado KNN, RF e SVM para 230 características com segmentação	83
Tabela 8 – Resultado KNN, RF and SVM para 34 características . . . . .	84
Tabela 9 – Resultado KNN, RF and SVM para 34 características com segmentação	84
Tabela 10 – Resultado KNN, RF e SVM para 40 características . . . . .	85
Tabela 11 – Resultado KNN, RF e SVM para 40 características com segmentação. .	85
Tabela 12 – Comparação das arquiteturas U-Net e PSPNet usando imagens com diferentes resoluções. . . . .	88
Tabela 13 – Avaliação dos modelos para diferentes alturas . . . . .	90



---

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>23</b>
<b>1.1</b>	<b>Motivação . . . . .</b>	<b>24</b>
<b>1.2</b>	<b>Objetivos e Desafios da Pesquisa . . . . .</b>	<b>26</b>
<b>1.3</b>	<b>Hipótese . . . . .</b>	<b>26</b>
<b>1.4</b>	<b>Contribuições . . . . .</b>	<b>27</b>
<b>1.5</b>	<b>Organização da Dissertação . . . . .</b>	<b>27</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>29</b>
<b>2.1</b>	<b>Coleta das Imagens . . . . .</b>	<b>29</b>
<b>2.2</b>	<b>Pré-Processamento . . . . .</b>	<b>32</b>
<b>2.3</b>	<b>Extração de Características . . . . .</b>	<b>36</b>
<b>2.4</b>	<b>Técnicas de Aprendizado de Máquina . . . . .</b>	<b>40</b>
2.4.1	K-Nearest Neighbors - KNN . . . . .	42
2.4.2	Random Forest - RF . . . . .	43
2.4.3	Support Vector Machine - SVM . . . . .	44
2.4.4	Deep Learning - DL . . . . .	45
<b>2.5</b>	<b>Partição e Avaliação de Desempenho . . . . .</b>	<b>50</b>
<b>2.6</b>	<b>Trabalhos Relacionados . . . . .</b>	<b>52</b>
<b>3</b>	<b>A SUA PROPOSTA . . . . .</b>	<b>57</b>
<b>3.1</b>	<b>Coleta de dados . . . . .</b>	<b>57</b>
3.1.1	Áreas Coletadas . . . . .	58
3.1.2	Análise nematológica . . . . .	59
<b>3.2</b>	<b>Seleção e Rotulagem das Imagens . . . . .</b>	<b>59</b>
3.2.1	Rótulo como Imagem . . . . .	60
3.2.2	Rótulo como Anotação . . . . .	60
3.2.3	CrITÉrios para Rotulagem . . . . .	62
<b>3.3</b>	<b>Pré-Processamento . . . . .</b>	<b>62</b>

3.3.1	Segmentação Manual . . . . .	63
3.3.2	Extração das Anotações . . . . .	63
<b>3.4</b>	<b>Extração de Características . . . . .</b>	<b>64</b>
3.4.1	Baseada em Blocos . . . . .	64
3.4.2	Baseada em Redes Neurais Convolucionais (CNN) . . . . .	68
<b>3.5</b>	<b>Algoritmos de Classificação . . . . .</b>	<b>70</b>
<b>3.6</b>	<b>Método e Medida de Avaliação . . . . .</b>	<b>71</b>
3.6.1	Treinamento para a estratégia baseada em blocos . . . . .	72
3.6.2	Treinamento para a estratégia baseada em CNN . . . . .	72
3.6.3	Resultado e Avaliação . . . . .	74
<b>3.7</b>	<b>Considerações finais . . . . .</b>	<b>75</b>
<b>4</b>	<b>EXPERIMENTOS E ANÁLISE DOS RESULTADOS . . . . .</b>	<b>77</b>
<b>4.1</b>	<b>Objetivos dos Experimentos . . . . .</b>	<b>77</b>
4.1.1	Experimento 1 - Avaliação da estratégia baseada em blocos . . . . .	77
4.1.2	Experimento 2 - Avaliação da estratégia baseada em CNN . . . . .	78
4.1.3	Experimento 3 - Avaliação de alturas usando a estratégia baseada em CNN . . . . .	78
<b>4.2</b>	<b>Bases de Dados . . . . .</b>	<b>78</b>
4.2.1	Base <i>Alfa</i> . . . . .	78
4.2.2	Base <i>Alfa-S</i> . . . . .	78
4.2.3	Base <i>Beta</i> . . . . .	79
4.2.4	Base <i>Gama</i> . . . . .	79
<b>4.3</b>	<b>Análise dos Experimentos . . . . .</b>	<b>80</b>
4.3.1	Experimento 1 - Avaliação da estratégia baseada em blocos . . . . .	80
4.3.2	Experimento 2 - Avaliação da estratégia baseada em CNN . . . . .	87
4.3.3	Experimento 3 - Avaliação de diferentes alturas usando a estratégia ba- seada em CNN . . . . .	89
<b>4.4</b>	<b>Considerações finais . . . . .</b>	<b>91</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>93</b>
<b>5.1</b>	<b>Principais Contribuições . . . . .</b>	<b>94</b>
<b>5.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>94</b>
<b>5.3</b>	<b>Contribuições em Produção Bibliográfica . . . . .</b>	<b>95</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>97</b>



---

## Introdução

Com a área plantada de 10,9 milhões de hectares e produção mundial de 153,6 milhões de sacas de 60 kg de café beneficiado na safra 2016, a cafeicultura representa uma importante atividade agrícola no cenário mundial (FAOSTAT, 2018). Em 2017, o volume total de café verde exportado no mundo atingiu 119,5 milhões de sacas de 60 kg (ICO, 2018).

O Brasil é o maior produtor e exportador de café e segundo maior consumidor da bebida no mundo, tendo como as principais espécies plantadas a arábica (80% da área) e o conilon. Além disso, o café ocupa a 5ª posição na pauta de exportação brasileira (MAPA, 2017), gerando uma receita cambial para o Brasil de US\$ 5,23 bilhões de dólares (CECAFE, 2018) e contribuindo para geração de mais de 8 milhões de empregos no país, proporcionando renda, acesso à saúde e educação aos trabalhadores e suas famílias (MAPA, 2017).

Dentre os principais estados produtores de café estão Minas Gerais, São Paulo, Espírito Santo, Bahia, Rondônia, Paraná, Rio de Janeiro, Goiás, Mato Grosso, Amazonas e Pará (MAPA, 2017). Minas Gerais é o maior produtor nacional de café, concentrando sua produção na espécie arábica e sendo responsável por aproximadamente 50% da safra brasileira. O café é o principal produto de exportação do agronegócio mineiro, sendo comercializado para mais de 60 países do mundo (SEAPAMG, 2017).

Apesar do Brasil ser o maior produtor de café, representando 32,7% do total produzido no mundo, a produtividade média de 25 sacas  $ha^{-1}$  ainda é considerada baixa (FAOSTAT, 2018). Dentre os fatores limitantes ao processo de crescimento e produção cafeeira, destacam-se a ocorrência de pragas e doenças. Nesse sentido, os nematóides do gênero *Meloidogyne* (*M.*) acarretam grandes prejuízos econômicos que variam em função da espécie presente na área (BOISSEAU et al., 2009), apresentando ampla disseminação, alta capacidade reprodutiva e agressividade (CAMPOS; VILLAIN, 2005).

Estimar precisamente as perdas provocadas unicamente pelos nematóides é uma tarefa difícil por vários fatores, dentre eles, a impossibilidade de isolar apenas o efeito causado pelo nematóide em condições de campo, onde o cafeeiro está sob a influência de variáveis

climatológicas, biológicas e bienalidade produtiva, característica do cafeeiro (SALGADO; CARNEIRO; PINHO, 2011). Entretanto, alguns estudos apontam redução de 30% a 45% na produção cafeeira do Brasil (CAMPOS; VILLAIN, 2005).

Dentre as espécies de nematóides que são parasitas de café-arábica destaca-se o *M. paranaensis*, com ocorrência no Brasil, Guatemala e Havaí (CARNEIRO et al., 2004). Essa espécie tem sido a mais severa (SALGADO; REZENDE; NUNES, 2014), ocasionando danos às raízes das plantas (Figura 1(a)), levando aos sintomas reflexos na parte aérea tais como amarelecimento, definhamento (Fig. 1(b)) e reboleiras<sup>1</sup> com consequente morte de plantas de cultivares suscetíveis.



Figura 1 – (a) Raízes de café com galhas atípicas causadas por *M. exigua* (CHAVES, 2019). (b) Sintomas resultantes do parasitismo ao cafeeiro como queda de folhas, amarelamento e definhamento (STARBUCKS, 2016).

Como é praticamente impossível erradicar os nematóides, o ideal é evitar sua introdução e disseminação nas áreas cafeeiras, por meio da adoção de medidas adequadas de manejo e o controle de nematóides. O controle dos nematóides é dificultado pela falta de conhecimento dos agricultores sobre a presença destes parasitas em suas plantações e os danos econômicos que podem ocasionar ao cafeeiro. Outra dificuldade é a falta de um método eficaz no controle das diversas espécies que parasitam o cafeeiro (SALGADO; CARNEIRO; PINHO, 2011).

## 1.1 Motivação

Sabendo da dificuldade em se eliminar os nematóides, uma vez que a área tenha sido contaminada, a adoção de medidas de manejo integrado são essenciais para conter a dispersão dos nematóides. Tais medidas envolvem o monitoramento permanente do cafeeiro, incluindo a coleta de uma grande quantidade de amostras e envio ao laboratório para realização do teste nematológico, que permite apontar com precisão a espécie e a quantidade de nematóides.

<sup>1</sup> Reboleiras consiste na manifestação dos sintomas da praga em regiões, visualizadas no campo.

O acompanhamento do cafeeiro normalmente é realizado por um agrônomo ou responsável pela lavoura. Por se tratar de áreas normalmente extensas, há a necessidade de caminhar pela plantação por longas horas para coleta de uma boa amostragem, tornando esse trabalho cansativo, oneroso e passível de falhas. Isso ocorre muitas vezes por conta de amostras insuficientes ou que não representam a população. Outro ponto está na dificuldade em uniformizar os critérios para identificação dos nematóides, o que torna essa tarefa constantemente subjetiva e por último a falta de conhecimento do responsável pelo monitoramento.

A adoção de técnicas como sensoriamento remoto (SR) permite auxiliar no processo de monitoramento de grandes áreas. De acordo com (NOVO, 1989), o SR baseia-se na utilização de sensores, equipamentos, aeronaves, entre outros, com objetivo de estudar o ambiente terrestre por meio do registro e da análise das interações entre a radiação eletromagnética e os componentes do planeta terra. Algumas das principais aplicações envolvendo o SR são o controle de desmatamento, identificação de focos de incêndios e queimadas, análise de crescimentos urbanos e mudanças climáticas. Além do mais, percebe-se que o uso do SR pode ser ampliado para a agricultura, possibilitando mapear grandes áreas em um curto intervalo de tempo e permitindo fornecer uma visão aérea provendo detalhes não observáveis pela visão no nível do solo.

Como forma de agilizar o processo de reconhecimento das áreas afetadas por pragas, muitas pesquisas estão sendo desenvolvidas com a utilização de Veículos Aéreos Não-Tripulados (VANTs) em SR. As principais vantagens do sensoriamento remoto por VANTs, esta na redução dos custos para obtenção das imagens, capacidade de se obter dados em qualquer momento da pesquisa, inclusive de serem obtidos mais de uma vez por dia.

Em (OTOBONI, 2014) houve uma análise temporal por meio de imagens aéreas obtidas por satélite (*Google Earth*), juntamente com o uso de aplicativo para demarcação de regiões críticas (também chamado de “*field scouting*”) para a demarcação de reboleiras causados por nematóides (*M. incognita*) nas culturas de café e soja.

A vantagem dessa técnica, está na simplicidade e rapidez para execução. No entanto, considerando que a atividade de monitoramento necessite ser realizada constantemente e avaliada por um longo período para se manter um histórico daquela área, alguns fatores podem influenciar esta análise, como a periodicidade com que tais imagens são atualizadas. Em geral as imagens não são atualizadas com frequência em bases públicas. Além disso, após a coleta das imagens, todo o restante do processo é feito manualmente, onerando tempo e esforço dos especialistas. Por último, o uso dessa técnica mantém o caráter subjetivo da atividade, dado que a identificação e demarcação das áreas é feita por um ser humano.

Deste modo, é fundamental o uso de técnicas que reduzam significativamente esse caráter subjetivo na identificação das regiões infestadas por nematóides e sirvam de apoio para especialistas do café. Além de permitir a coleta de imagens adequadas e com a

regularidade necessária para o efetivo monitoramento da lavoura. Sendo assim, a aplicação de técnicas que aliem a coleta de imagens aéreas de alta resolução com baixo custo e o emprego de técnicas computacionais baseadas em aprendizado de máquina (AM) possuem um grande potencial para detecção automática de plantas com sintomas de nematóides em lavouras cafeeiras, propiciando a redução de custos de produção e otimização das medidas de manejo integrado.

## 1.2 Objetivos e Desafios da Pesquisa

Desenvolver um método para detecção de plantas com sintomas de nematóides (*M. paranaensis*) em lavoura cafeeira, usando imagens aéreas RGB, coletadas por meio de um VANT e empregando técnicas de AM. Além disso, o método proposto deve permitir auxiliar os especialistas na cultura cafeeira de maneira eficaz e possibilitar que o mesmo seja comparável e reproduzível. Os objetivos específicos deste trabalho são:

- ❑ Aplicar e adaptar técnicas de pré-processamento e segmentação de imagens, além de fornecer as características que melhor discriminam regiões doentes e saudáveis;
- ❑ Construir e disponibilizar publicamente uma base de dados de imagens cafeeiras com a presença de pragas;
- ❑ Investigar a influência das imagens obtidas em diferentes altitudes com o VANT e definir a altura mais adequada para a detecção de pragas;
- ❑ Comparar diferentes técnicas de AM e definir quais são mais eficazes para a detecção do nematóide no cafeeiro.

## 1.3 Hipótese

- ❑ A presença do solo prejudica o desempenho do classificador na tarefa de detecção dos nematóides no cafeeiro;
- ❑ As imagens coletadas em período seco evidenciam melhor as características visuais da praga nematóide, quando comparados com período chuvoso;
- ❑ O erro de Falso negativo (FN) é menor usando redes neurais convolucionais, comparado com as técnicas tradicionais para detecção de nematóides;
- ❑ Um modelo treinado com imagens obtidas em baixas altitudes, generaliza o problema e alcança bom desempenho em imagens com maiores altitudes.

## 1.4 Contribuições

- ❑ Desenvolvimento de um método automático para detecção de nematóides em lavouras cafeeiras usando imagens aéreas, com potencial para ser aplicado em outras culturas (e.g. algodão, soja e milho);
- ❑ Criação e disponibilização de uma base de dados pública de imagens aéreas obtidas em lavouras cafeeiras, com o intuito de tornar a pesquisa reprodutível;
- ❑ Estudo comparativo da influência do uso de diferentes altitudes na detecção de pragas nematóides.

## 1.5 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos, como mostrado a seguir:

- ❑ Capítulo 2: apresenta os conceitos e as etapas fundamentais que compõem um sistema de detecção automática de pragas, dentre eles: coleta das imagens, pré-processamento, extração de características, técnicas de aprendizado de máquina e métodos de partição dos dados e avaliação de desempenho;
- ❑ Capítulo 3: apresenta o método para detecção automática de nematóides, detalhando cada uma das etapas que compõem o método proposto, desde a forma como as imagens são coletadas e rotuladas, as técnicas usadas para extração de características, bem como os algoritmos de classificação e medidas de avaliação;
- ❑ Capítulo 4: apresenta os resultados realizados para avaliação da estratégia baseada em blocos e CNN, bem como avaliar a influência na variação da altura;
- ❑ Capítulo 5: apresenta a conclusão desta dissertação com as considerações finais e propostas para possíveis trabalhos futuros. Também destaca-se as publicações geradas com a pesquisa desenvolvida.



---

## Fundamentação Teórica

Sabendo que a coleta de imagens de alta resolução e o uso de técnicas computacionais apresentam um grande potencial para detecção de pragas, incluindo os nematóides, neste capítulo é apresentado os componentes de um sistema automático de detecção de pragas, além de trabalhos relacionados. De uma maneira geral, um sistema de detecção automática de pragas é composto das seguintes etapas: I) Coleta das imagens da planta ou plantação; II) Pré-processamento (opcional); III) Extração de características de cor, forma e textura, etc. IV) Utilização de técnicas computacionais e AM para a detecção automática dos sintomas da doença na cultura por meio da análise de várias partes da planta como a raiz, frutos, caule e folhas e V) Avaliação de desempenho.

O capítulo está organizado da seguinte maneira: a Seção 2.1 descreve a etapa de coleta das imagens, especialmente para o contexto da agricultura, incluindo os sensores e equipamentos utilizados, com foco para os dados coletados por meio de VANTs; a Seção 2.2 descreve algumas técnicas de pré-processamento, em particular o redimensionamento e realce de imagens e apresenta 3 métodos de segmentação; a Seção 2.3 apresenta as principais técnicas para extração de características em imagens; a Seção 2.4 apresenta técnicas de AM usadas para adquirir conhecimento por meio de exemplos; a Seção 2.5 exibe formas de se avaliar as técnicas usadas na etapa anterior e finalmente na seção 2.6 são apresentados os trabalhos relacionados.

### 2.1 Coleta das Imagens

Existem inúmeras formas para se coletar as imagens da planta a ser analisada. Uma das formas mais acessíveis e populares é a coleta manual, que normalmente é feita por meio de câmeras digitais. A vantagem desse tipo de coleta, está na sua simplicidade, dado que hoje em dia os *smartphones* são equipados com câmeras digitais de alta resolução, podendo ser uma boa alternativa para realização da coleta de imagens. No entanto, essa coleta é limitada a pequenas áreas. Considerando áreas mais extensas, esta abordagem torna-se impraticável, visto que o responsável pela coleta teria que percorrer grandes distâncias

para obter uma amostra significativa e além disso, necessitando de um conhecimento a priori que lhe permita identificar visualmente os sintomas da doença a ser avaliada. Outra desvantagem é que muitos trabalhos impõem algumas restrições para a coleta das imagens com o intuito de diminuir o impacto dos ruídos, como o uso de um fundo escuro, boas condições de luminosidade e pouco reflexo (PETHYBRIDGE; NELSON, 2015).

DerwinSuhartono, Lestari e Yasin (2013) apresentam um sistema especialista que possibilita identificar diversas doenças na folha do café como ferrugens, nematóides e presença de fungos usando imagens obtidas de câmera digital. Vassallo-Barco et al. (2017) propõe uma análise utilizando folhas de café para detecção automática de deficiência nutricional dos elementos Boro (B), Cálcio (C), Ferro (Fe) e Potássio (K), que são importantes componentes na formação das plantas e auxiliam no processo metabólico. Nesse estudo, as imagens são coletadas manualmente por meio de uma câmera digital em um ambiente controlado. Barbedo (2014), propõe um método automático para detecção de doença nas folhas de diversas culturas, como café, maracujá, tomate, amendoim, beterraba, soja e milho, usando imagens digitais obtidas por meio de câmeras digitais e escâneres de alta resolução. Pethybridge e Nelson (2015) apresenta uma aplicação desenvolvida para *smartphones* que permite identificar doenças nas folhas de tomate, melancia, couve e limão e calcular o grau de severidade, como mostrado na Figura 2. Para a coleta das imagens, é utilizada uma câmera de *smartphone*.

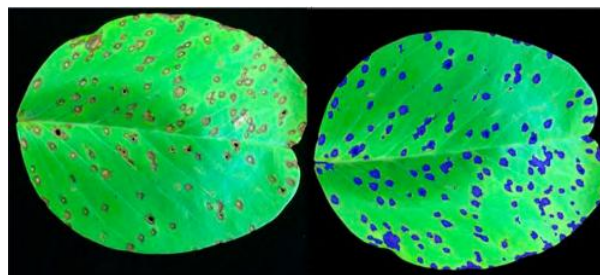


Figura 2 – À esquerda, folha com sintoma de cercóspora (manchas marrom). À direita, avaliação das regiões com sintoma da doença pelo software *Leaf Doctor*. Imagem adaptada de (PETHYBRIDGE; NELSON, 2015).

Considerando áreas extensas, outras abordagens se mostram mais adequadas como aquelas onde as imagens são coletadas por meio de satélites. Algumas vantagens dessa abordagem estão na sua alta disponibilidade e permitir cobrir áreas enormes. Em relação as desvantagens, destacam-se o altíssimo custo para se colocar um satélite em órbita, grande complexidade para implantação e operacionalização, além de ser altamente sensível as condições climáticas, necessitando de céu limpo para obtenção de imagens de qualidade (SHAKHATREH et al., 2018).

Gaertner et al. (2017) apresenta um método para análise da extensão espacial em grandes áreas, onde são cultivados cafeeiros da espécie arábica no norte e no sul do Havaí. De acordo com os autores, o método auxilia no manejo e controle de pragas,



planejamento econômico, além de permitir acompanhar as mudanças no café. Para o estudo, são utilizadas imagens aéreas obtidas pelo satélite WorldView-2 (WV2). Hicke e Logan (2009) propõe um método para mapeamento e classificação da mortalidade do pinheiro de casca branca, causada pela infestação de besouros em terrenos montanhosos e ingrimes usando imagens de satélite de alta resolução espacial. Em Reed et al. (1994), imagens coletadas por meio de satélite são utilizadas para avaliar de maneira eficiente e objetiva as características fenológicas em grandes áreas de vegetação.

Outra alternativa para a coleta de imagens em grandes áreas é a utilização de aeronaves tripuladas. A vantagem das aeronaves tripuladas consiste na cobertura de grandes áreas, permitindo a coleta de imagens mesmo em condições climáticas pouco favoráveis. Por outro lado, pode haver um custo alto para manutenção das aeronaves, risco a tripulação além de uma menor disponibilidade para realização dos voos se comparada com outras abordagens.

No trabalho de Akesson e Yates (1974), as imagens são coletadas por meio de aeronaves tripuladas, para o monitoramento de grandes plantações e controle de pragas. Huang, Lan e Hoffmann (2008) propõe um estudo baseado em imagens multi-espectrais obtidas através de um aeronave tripulada (*Cessna 206*), com grande potencial para controle de pragas, ervas daninhas e danos causados por insetos. A aplicação de aeronaves tripuladas também é amplamente utilizada no campo da agricultura para a tarefa de pulverização de grandes áreas. Em Teach (1994), é proposto um sistema de controle de aeronaves para aplicação de produtos químicos a um campo agrícola.

Por último, destaca-se o uso de VANTs para a detecção de imagens de plantas com sintomas decorrentes da presença de pragas. Na agricultura, os VANTs têm sido utilizados com diversas finalidades, dentre elas no diagnóstico de doenças em diferentes culturas. As principais vantagens da utilização de VANTs para coleta de imagens, em relação as abordagens de aquisição por satélite e aeronave tripulada, são o baixo custo, alta disponibilidade, permitindo cobrir áreas significativas e baixa complexidade para colocá-lo em operação. Em relação às desvantagens, está a baixa autonomia de bateria e maior susceptibilidade as condições climáticas, incluindo vento, chuva e neve.

Tetila et al. (2017) propõem um sistema de visão computacional, para identificar de maneira automática doenças foliares da soja usando imagens obtidas por VANT (*DJI Phantom 3*). A coleta foi realizada durante a safra, no período compreendido entre Setembro/2016 e Fevereiro/2017 em condições meteorológicas distintas. Nardari et al. (2018) apresenta um estudo comparativo entre técnicas de segmentação semântica para a tarefa de monitoramento e identificação de falhas em plantação de laranja com a utilização de imagens coletadas por VANT. Souza et al. (2015) propõe um método para detecção automática da doença murcha de *Ceratocystis* em plantações de eucalipto no Brasil por meio de imagens aéreas RGB de alta resolução, obtidas por VANT (*eBee robot, SenseFly*).

Durante o processo de coleta das imagens é possível utilizar diversos tipos de sensores

acoplados aos VANTs para obter as imagens aéreas, dentre eles podemos destacar o uso de câmeras digitais, multiespectrais, hiperspectrais, térmicas, além de espectrômetro, LIDAR e SONAR. Na Tabela 1 são exibidas algumas informações sobre os sensores, como a frequência de operação (espectro), aplicações e as desvantagens apresentadas por cada um deles.

Tabela 1 – Sensores para VANTs em aplicações agrícolas. Adaptado de (SHAKHATREH et al., 2018)

Tipo do Sensor	Freq. Operação (Espectro)	Aplicações	Desvantagens
Digital	Visível	Características visíveis, anomalias externas, verdura, crescimento.	- Limitado ao espectro visível pelo olho humano.
Multiespectral	Visível infravermelho	Deficiências nutricionais, estresse hídrico, doenças, entre outros.	- Limitado a poucas bandas espectrais.
Hiperspectral	Visível infravermelho	Stresse da planta, avaliar a qualidade do produto e controle de segurança.	- Processamento da imagem é um processo desafiador. - Sensores de alto custo.
Térmica	Termal infravermelho	Respostas das plantas ao estresse hídrico e doenças	- Condições ambientais podem influenciar no desempenho. - Diferenças de temperatura muito pequenas não são detectáveis.
Espectrômetro	Eletromagnético	Deteção de doenças, estresse e respostas das culturas.	- A presença do solo pode afetar na qualidade dos dados. - Possibilidade de mistura espectral. - Mais aplicável em sensores de solo.
LIDAR	Infravermelho próximo	Estimativas precisas de altura e volume da planta / árvore	- Sensível a pequenas variações no comprimento
SONAR	Som	Mapeamento e quantificação dos volumes da copa, espalhador de fertilizante.	- Sensível a ruídos.

No estudo proposto por Calderón et al. (2013), são utilizados sensores termiais, multiespectrais e hiperspectrais acoplados ao VANT para detecção dos sintomas da praga *murcha de Verticillium* em oliveiras (*Olea europaea L.*).

Desse modo é possível perceber que há várias formas para realizar a coleta das imagens, incluindo diferentes tipos de sensores e plataformas onde normalmente os sensores são acoplados. A escolha da forma mais adequada está fortemente relacionada ao contexto e o tipo de análise a ser realizada. A próxima seção descreve a etapa de Pré-processamento.

## 2.2 Pré-Processamento

Após o estágio de coleta das imagens, são aplicadas técnicas de pré-processamento que tem como objetivo melhorar a imagem, corrigindo algum defeito proveniente de sua aquisição e/ou realçando detalhes relevantes para a análise (GOMES, 2001). Segundo Oliveira (2007), nesta etapa devem ser corrigidas quaisquer distorções ocasionadas durante a fase de captura da imagem, incluindo distorções geométricas, variações na iluminação, entre outras. Além disso a imagem é adequada ao algoritmo responsável por realizar a extração das informações necessárias para a análise em questão. O realce dos detalhes da imagem também deve ser abordado nesta fase, com o uso de técnicas para a eliminação de partes indesejadas, ajustes de brilho, retirada de partes insignificantes, dentre outras, de modo a se obter as características que se deseja medir ou verificar.

Um dos primeiros passos realizados nessa etapa consiste no redimensionamento de imagens (redução ou ampliação) por meio de interpolação, que consiste no processo de estimar valores intermediários de uma função ou sinal discreto amostrado em posições no espaço contínuo. Em suma, o objetivo da interpolação é fornecer uma estimativa ótima para valores de uma imagem em qualquer posição do espaço bidimensional. Neste processo, são recomendadas aplicações de técnicas que não comprometam a qualidade da imagem, como a interpolação por vizinho mais próximo, bilinear e bicúbica. Em geral, a interpolação bicúbica apresenta os melhores resultados na preservação de detalhes, sendo o padrão utilizado em programas de edição de imagens, como *Adobe Photoshop* e o *Corel Photopaint* (GONZALEZ; WOODS, 2010).

Nesta etapa, também é comum o uso de operações para realçar as imagens, que consistem na alteração dos valores dos pixels de uma imagem, de forma a obter uma nova imagem, de melhor visualização. O realce da imagem, frequentemente é baseado em histogramas, que consiste em uma função que mostra a frequência com que cada nível de cinza aparece na imagem. Histograma é uma ferramenta bastante útil na etapa de pré-processamento, pois fornece uma visão estatística sobre a distribuição das intensidades dos pixels, sobre o contraste da imagem e os níveis de brilho. Além disso, o histograma é bastante utilizado na etapa de segmentação, principalmente em técnicas que se utilizam da similaridade entre os pixels (IGLESIAS, 2008).

Dentre as técnicas de realce, pode-se destacar a equalização do histograma, a qual procura redistribuir os valores dos níveis de cinza em uma imagem, para se obter um histograma uniforme, visando aumentar o intervalo dinâmico de uma imagem melhorando o contraste de imagens. No contexto de detecção de doenças em plantas, a remoção dos ruídos tem um papel importante para melhor identificação das características visíveis da doença.

Khirade e Patil (2015) apresentam um método automático para detecção de doenças em plantas por meio de imagens da folha. Um dos passos para detecção da doença compreende a etapa de pré-processamento, cujo objetivo neste trabalho é a remoção de ruídos. O primeiro passo consiste em aplicar uma operação de suavização. Em seguida, a imagem original (RGB) é convertida para a escala de cinza e por último é aplicado a equalização do histograma usando a função de distribuição acumulativa para distribuir os valores de intensidade.

No contexto de detecção de doenças de plantas, é muito comum a aplicação de técnicas como a segmentação para dividir uma imagem em regiões distintas, particionando os pixels em grupos de atributos similares (GONZALEZ; WOODS, 2010). Neste caso, pode-se segmentar uma imagem de uma folha, por exemplo, em dois grupos, um relacionado a regiões com sintomas da doença e outro com regiões saudáveis. O resultado da segmentação é uma imagem binária, na qual os pixels brancos formam a região de interesse e os pixels pretos o fundo, ou seja, de não interesse. Os algoritmos de segmentação geralmente são

baseados na busca pela descontinuidade (Exemplo: bordas) e similaridade (Ex: regiões) dos níveis de cinza (GONZALEZ; WOODS, 2010). A segmentação pode ser categorizada em 3 principais tipos, sendo limiarização (*Thresholding*), bordas e regiões.

Limiarização é um processo de segmentação de imagens que se baseia na diferença dos níveis de cinza que compõe diferentes objetos de uma imagem. A partir de um limiar estabelecido de acordo com as características dos objetos que se quer isolar, a imagem pode ser segmentada em dois grupos: o grupo de pixels com níveis de cinza abaixo do limiar e o grupo de pixels com níveis de cinza acima do limiar. Em uma imagem limiarizada, atribui-se um valor fixo para todos os pixels de mesmo grupo. A principal dificuldade nesta técnica está em escolher o limiar (ou os limiares) mais adequado para separar o objeto (ou os objetos) do fundo (GONZALEZ; WOODS, 2010). A escolha do limiar pode ser feita por:

- ❑ **Inspeção visual:** Usuário escolhe limiar na inspeção visual do histograma.
- ❑ **Tentativa e erro:** Aplicado a processos interativos. O usuário testa diferentes níveis até produzir um resultado satisfatório para o observador.
- ❑ **Obtenção automática do limiar:** Análise do formato do histograma.

Em alguns casos, a presença de sombras pode afetar negativamente o desempenho no processo de detecção de doenças, principalmente quando a avaliação é realizada por meio da folha da planta. Vassallo-Barco et al. (2017) aplica o processo de segmentação para remoção das sombras utilizando o algoritmo de *thresholding* global ótimo Otsu (1979), que tem como ideia maximizar a variância entre as classes (grupos) e minimizar a variância interna das classes (grupo). A Figura 3 exibe o esquema desse método de limiarização.

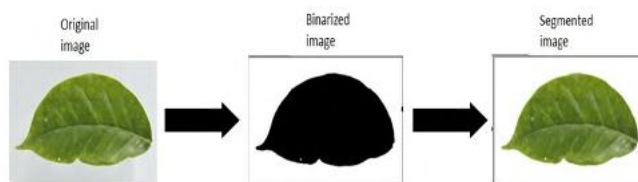


Figura 3 – Segmentação de imagens digitais por meio do método de Otsu. Imagem adaptada de (VASSALLO-BARCO et al., 2017).

Com o objetivo de diminuir o impacto causado pelas sombras nas imagens, alguns trabalhos impõem restrições mais rígidas na etapa de coleta das imagens, como a utilização de fundos claros ou escuros. Barbedo (2014) pressupõe que algumas premissas sejam satisfeitas para que a segmentação seja bem sucedida, como o fundo ser mais próximo possível do tom de branco ou preto. Para a segmentação, as imagens inicialmente são

convertidas para a escala de cinza com intervalo de (0 até 255) e aplica-se um *threshold* simples. Em seguida, elimina-se alguns ruídos da imagem oriundos da coletas, nos extremos da imagem. De acordo com o autor, o descarte das regiões espúrias corresponde a menos de 0,05% da área total da folha, o que é insuficiente para causar qualquer erro perceptível nos resultados finais. Por último, o algoritmo converte a imagem da representação de cor (geralmente RGB) para um formato  $L^*a^*b^*$  que destaca os diferentes tipos de sintomas. A Figura 4 exibe a imagem original (a) e a imagem gerada após conversão para o formato  $L^*a^*b^*$  (b).



Figura 4 – (a) Imagem de folhas doentes com sintomas, representada pelo espaço de cor RGB. (b) Imagem de folhas doentes com sintomas, representada pelo canal do espaço de cor  $L^*a^*b^*$ , que destaca sintomas e veias foliares. Fonte: (BARBEDO, 2014)

Algoritmos baseados em um limiar podem ser severamente afetados pela presença de ruídos ou iluminação não uniforme. Nestes casos, o uso de limiares variáveis são mais adequados. A aplicação de limiares variáveis pode ser feita de acordo com duas técnicas distintas. Na primeira chamada de particionamento, a imagem é subdividida em vários segmentos não sobrepostos e em seguida é aplicado algoritmos como *Otsu* (OTSU, 1979), baseado na maximização da variância entre classes, em cada um dos segmentos. Na segunda, a limiarização é baseada nas propriedades locais da imagem. Desta forma, calcula-se um limiar para cada ponto  $(x,y)$  da imagem com base em uma ou mais propriedades que são calculadas em sua vizinhança (GONZALEZ; WOODS, 2010).

Já os métodos de segmentação baseados em detecção de discontinuidades, pressupõem que as fronteiras das regiões são suficientemente diferentes entre si e em relação ao fundo da imagem para permitir a detecção de limite com base nas discontinuidades locais em intensidade (GONZALEZ; WOODS, 2010). A detecção de bordas é o método mais usado para segmentar as imagens com base nas variações abruptas (locais) de intensidade. Outro método empregado consiste na transformada de *Hough*, que permite encontrar formas em imagens (Retas, círculos e elipses). Em (KHIRADE; PATIL, 2015), a segmentação utiliza o algoritmo de detecção de bordas para identificar as manchas, que são sintomas de alguma doença na folha das plantas. Inicialmente, a imagem é convertida do formato RGB para

o modelo HSI, depois disso aplica-se um algoritmo de detecção de bordas considerando uma conectividade de 8 *pixels*.

Por último, destacam-se os métodos de segmentação que são baseados em encontrar regiões de forma direta. Um dos principais algoritmos representantes dessa categoria é o crescimento de regiões, procedimento que agrupa os pixels ou as sub-regiões em regiões maiores com base em critérios predefinidos para o crescimento. A abordagem básica é começar com um conjunto de pontos “semente” e, a partir deles, fazer as regiões crescerem anexando a cada semente aqueles pixels vizinhos que têm propriedades predefinidas semelhantes às das sementes (como os intervalos específicos de intensidade, cor ou textura) (GONZALEZ; WOODS, 2010).

Ma et al. (2017) propõem um método usando informações de cor e crescimento de região para segmentação de imagens de folhas de pepino cultivadas em estufas com manchas provenientes da doença míldio. O método é executado em duas etapas, na primeira delas são extraídas as seguintes características: *Excess Red Index* (ExR), componente H do sistema de cor HSV e o componente \*b do espaço de cor L\*a\*b\*, que de acordo com o autor, melhor discriminam as manchas das folhas. Em seguida, um método de crescimento de região é usado para obter a segmentação de pontos de doença. A Figura 5 exibe uma comparação do algoritmo proposto pelo autor baseado em crescimento de região com o algoritmo de agrupamento *K-means* e *Otsu*.

Também é possível combinar dois ou mais tipos de segmentação. Pang et al. (2011) apresenta um algoritmo de segmentação adaptativa, chamado de LTSRG que combina os métodos de linearização local e crescimento de regiões. O algoritmo é aplicado utilizando imagens digitais de folhas de milho para identificação de seis tipos de doenças que são manifestadas através de lesões na folha. Inicialmente identifica-se todos os pixels para os quais o nível do canal vermelho (R) é maior que o nível do canal verde (G) para o nível de cinza e seleciona esses pontos como “sementes” iniciais para aplicar o crescimento de regiões. Em seguida um *threshold* local é calculado para cada semente conectada e aplica-se o algoritmo *Otsu* (OTSU, 1979). Repete-se o processo, selecionando novos pixels como sementes e aplica o *threshold* local até que nenhum novo pixel semente possa ser incluído.

## 2.3 Extração de Características

A etapa de extração de características é uma das bases fundamentais dos sistemas de detecção automática de praga. A sua relevância está relacionada ao fato de que são as características extraídas nesta etapa que são fornecidas para discriminar entre regiões com a presença de praga e saudáveis.

O processo de extração de características consiste no cálculo de representações numéricas que podem ser utilizadas para caracterizar um determinado dado. De maneira geral,

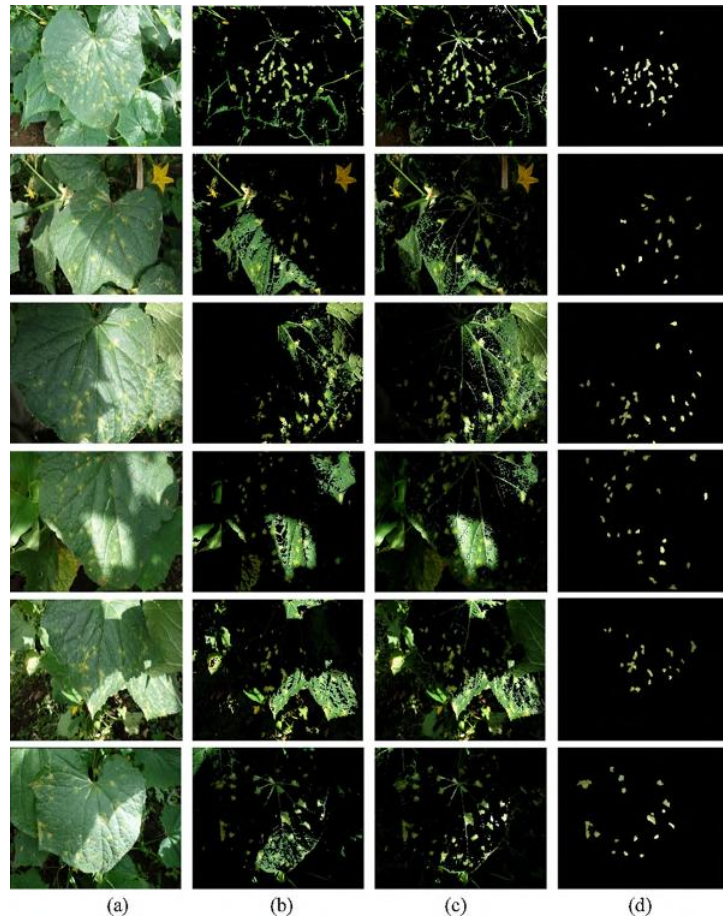


Figura 5 – Resultado dos diferentes algoritmos de segmentação: (a) Imagem original, (b) algoritmo de agrupamento *k-means*, (c) algoritmo *Otsu* e (d) algoritmo proposto pelo autor. Fonte: (MA et al., 2017).

existem duas abordagens para a extração dessas representações numéricas, sendo a extração de características a partir de dados brutos (*raw data*), ou a extração de características a partir de dados transformados, ou seja, as representações numéricas são obtidas a partir de um domínio de transformação comprimido (TRAINA; JR, 2003). Considerando o domínio de dados de imagens, por exemplo, é usual a extração das características a partir dos dados brutos, i.e., diretamente dos pixels da imagem.

Nesse processo, um dos principais desafios está na escolha de características que descrevam os objetos adequadamente, uma vez que essa escolha é altamente dependente do contexto. As informações individuais de cada objeto são obtidas por meio de descritores, capazes de descrever características baseadas em cor, textura e forma.

Os dados extraídos por meio dos descritores são representados através de um vetor de características (do inglês *feature vector*) que armazena características de alto nível representados pela imagem. Considerando a identificação de pragas, é interessante a seleção de características discriminativas, ou seja, que caracterizem as diferenças em relação as plantas doentes e saudáveis. Pode-se destacar os descritores baseados em cor e textura:

□ **Cor:** É provavelmente a característica mais utilizada para recuperação visual, por



apresentar relativa robustez no que concerne a independência do tamanho da imagem e da orientação da mesma. O RGB (*Red*, *Green* e *Blue*) é o espaço de cor mais utilizado para imagens digitais, com 3 canais ou componentes: vermelho (*red*), verde (*green*) e azul (*blue*). Outros sistemas de cores também são amplamente utilizados como HSI, HSV, YCbCr,  $L^*a^*b$ ,  $L^*u^*v$ , entre outros.

□ **Textura:** A textura constitui uma característica diretamente relacionada com as propriedades físicas que a superfície de um objeto apresenta (lisa, rugosa, macia, áspera ou ondulada). Ela descreve o padrão de variação de tons de cinza ou cor numa determinada área. Trata-se de um termo intuitivo e de grande utilidade, mas que, apesar de sua grande importância, não possui uma definição precisa.

Souza et al. (2015) propõem uma abordagem para extração de características baseada em blocos, onde utiliza-se um par de blocos de tamanhos diferentes e sobrepostos: o interno, usado para classificação e o externo que armazena as características do contexto. Além disso, utiliza-se descritores de cor e textura para extrair as características da imagem. Inicialmente o espaço de cor RGB é convertido em CIELab e escala de cinza, desta forma cada pixel produz um vetor com 4 componentes e são extraídos os valores de média e variância para cada componente. Já para as características texturais, o descritor *Local binary patterns* (LBP) é utilizado para extração de características da imagem, considerando apenas a escala de cinza com 4 vizinhos. De acordo com o autor, o LBP é capaz de gerar características distinguíveis considerando diferenças na iluminação, facilitando a distinção de regiões de estradas e árvores. Foram extraídas um conjunto de 25 características por bloco, permitindo discriminar satisfatoriamente as regiões de plantação saudáveis, com a incidência de pragas e o solo. A Figura 6 exibe a abordagem de extração de característica apresenta por (SOUZA et al., 2015).



Figura 6 – Abordagem baseada em blocos para extração de características. Fonte: (SOUZA et al., 2015).



Seguindo a mesma ideia para extração usando blocos, Shinde et al. (2015) apresentam uma abordagem para classificação de imagens, onde cada imagem pode ser categorizada em 6 classes distintas, sendo elas: flores, aeronaves, automóveis, faces, elefantes e dinossauros. Para a extração de características, utiliza-se somente características baseadas em cor. Inicialmente a imagem é dividida em 16 blocos de mesmo tamanho e na sequência, para cada bloco é obtida a média para cada componente dos espaços de cor RGB e LUV, totalizando 6 atributos. De acordo com o autor, o espaço de cor LUV é usado para eliminar o impacto na dominância de cores.

Kusumo et al. (2018) apresentam uma abordagem para detecção automática de doenças na folha do milho. Neste trabalho são investigadas várias características com o intuito de identificar aquelas que melhor discriminam as características da doença e apresentam maior robustez em relação a variabilidade dos dados. Desta forma, são utilizadas descritores de cor como RGB, descritores baseados em características locais que são relacionados a um padrão encontrado em uma imagem, como um ponto ou uma borda. Exemplos de características locais são: SIFT (LOWE, 2004), SURF (BAY; TUYTELAARS; GOOL, 2006) e ORB (RUBLEE et al., 2011), além de descritores usados para detecção de objetos como HOG (TOMASI, 2012). Os resultados do experimento indicam que descritores baseados em cor, como RGB são altamente discriminativos, apresentando os melhores resultados na identificação de pragas na folha do milho.

Vassallo-Barco et al. (2017) utilizam dois descritores populares na literatura, conhecidos como *Blurred shape model* (BSM) e *Gray level co-occurrence matrix* (GLCM) para obter as características mais significativas das imagens contendo folhas de café. O BSM é usado para particionar a imagem por meio de uma grade com o intuito de extrair as características da imagem. A imagem original particionada é exibida na Figura 7(a). Esse processo pode ser dividido em duas etapas:

- ❑ O primeiro passo dessa técnica consiste na aplicação de um algoritmo de detecção de borda como *Canny* (CANNY, 1986), gerando como saída uma imagem binarizada.
- ❑ Na etapa seguinte, a imagem binária é particionada em múltiplas regiões por meio de uma grade de tamanho ajustável. Para o artigo, o autor utilizou uma grade de tamanho  $4 \times 4$ . A saída do descritor representa uma probabilidade distribuição da forma do objeto considerando distorções, onde o nível de distorção é determinado pelo tamanho da grade. A imagem resultante desse processo é exibida na Figura 7(b).

Segundo o autor, a vantagem dessa técnica está na alta tolerância a deformações (irregularidades) do objeto. Já o descritor GLCM consiste em um método estatístico de análise de textura que considera a relação espacial de pixels e a matriz de co-ocorrência de nível cinza, também conhecido como matriz de dependência espacial. De acordo com o autor, a imagem inicialmente é quantizada e particionada em 64 segmentos, de acordo

com as Figuras 7(c) e 7(d) respectivamente. O descritor é aplicado em cada segmento e as características de todos os segmentos ao final irão compor as características da imagem. Um dos pontos positivos dessa técnica é que uma vez que a matriz de co-ocorrência é construída, ela pode ser usada para a derivação de algumas medidas. Neste trabalho, a entropia, contraste e correlação são utilizadas. Um dos pontos negativos desta técnica, consiste no alto custo computacional para a geração das matrizes.

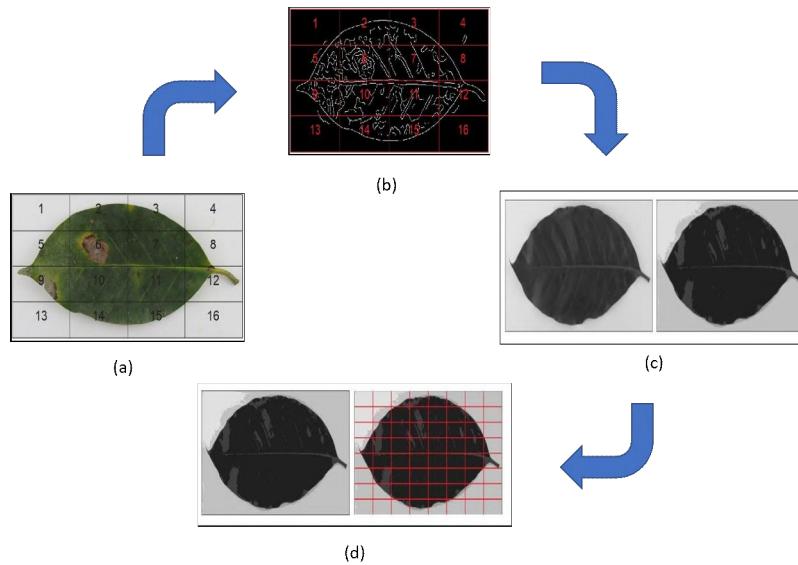


Figura 7 – (a) Imagem original particionada por meio de uma grade de tamanho 4x4. (b) Imagem dividida após aplicação do algoritmo de *Canny*. (c) A imagem em tons de cinza (à esquerda) é quantizada (à direita). (d) A imagem quantizada (à esquerda) é particionada em 64 segmentos (à direita). Fonte: Adaptado de (VASSALLO-BARCO et al., 2017).

## 2.4 Técnicas de Aprendizado de Máquina

Até o momento foram detalhadas formas de coletar os dados (Seção 2.1), no contexto de imagens, em seguida a aplicação de técnicas de pré-processamento (Seção 2.2) e por último a utilização de técnicas para extração manual das características de interesse (Seção 2.3). Em posse desses dados, deseja-se obter informações úteis que permitam tomar decisões assertivas. Entretanto na etapa de extração de características, frequentemente são obtidos conjunto de dados, com inúmeras amostras (linhas) e características (colunas) ou dimensões, impossibilitando desse modo a extração de informações úteis por parte dos humanos (MARSLAND, 2009).

Por outro lado, nos últimos anos percebe-se um crescimento no poder computacional, aumento na capacidade de se obter e armazenar dados, bem como um avanço em áreas da Inteligência Artificial (IA) como o Aprendizado de Máquina (AM), cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado assim como a construção

de sistemas capazes de adquirir conhecimento de forma automática (REZENDE, 2003), gerando soluções genéricas, flexíveis o suficiente para se adaptar aos dados coletados. Dessa maneira, o uso de AM tem sido amplamente utilizado em problemas de várias áreas, incluindo a detecção de regiões com praga na agricultura.

O AM é composto por 3 paradigmas de aprendizado, incluindo supervisionado (AS), não supervisionado e semi-supervisionado, que se diferenciam pela presença ou não de rótulos nos dados (MITCHELL, 1997).

- ❑ **Aprendizado supervisionado:** baseia-se em um conjunto de dados, mais especificamente um conjunto de exemplos de treinamento, em que as respostas (classes) ou saídas corretas (desejadas) são fornecidas e, com base neste conjunto de treinamento, os algoritmos em geral generalizam para responder corretamente a todas as entradas possíveis. Este tipo de aprendizado é chamado de aprendizado a partir de exemplos (MARSLAND, 2009). A Figura 8 (a) apresenta uma representação de base de dados supervisionada.
- ❑ **Aprendizado não supervisionado:** No aprendizado não supervisionado, tem-se um conjunto de exemplos, mas não se conhece a categorias (classes) envolvidas, sendo necessária a extração de propriedades estatisticamente relevantes. A Figura 8 (b) apresenta uma representação de base de dados não-supervisionada.
- ❑ **Semi-supervisionada:** A abordagem semi-supervisionada combina uma pequena quantidade de amostras classificadas com um grande número de amostras não classificadas para se produzir melhores classificadores. Normalmente um pequeno número de exemplos é rotulado com a respectiva classe. O restante dos exemplos têm valores ausentes no valor da classe, conforme ilustrado na Figura 8 (c)

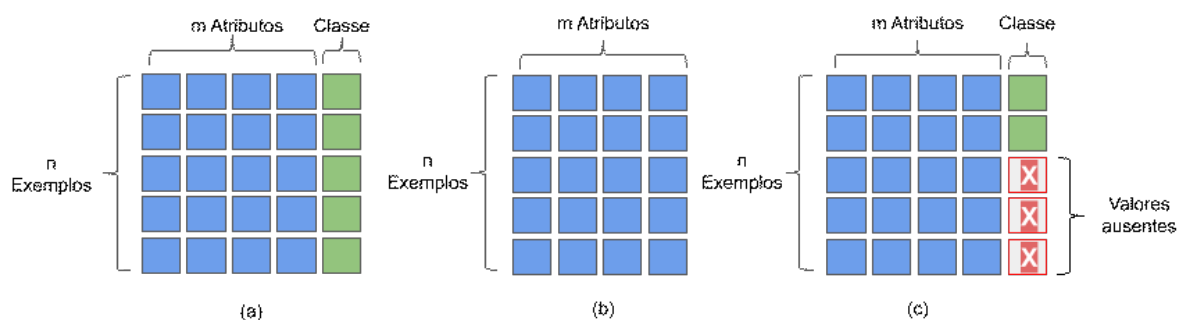


Figura 8 – Tipos de aprendizado: (a) Supervisionado. (b) Não supervisionado. (c) Semi-supervisionado. Fonte: Elaborada pelo autor

A classificação é um dos problemas comumente encontrados em AM supervisionado. A classificação consiste na generalização de experiências passadas  $X$  (espaço das entradas ou instâncias) para prever algum valor discreto (conhecido como classe ou rótulo) em novos exemplos (MITCHELL, 1997). No contexto de imagens, a classificação é o processo

de determinar a classe ou categoria de imagem. O processo de classificação é exemplificado pela Figura 9 (a), onde uma das questões a serem respondidas pelo classificador é identificar qual a classe identificada através da imagem, neste caso um gato.

Mas em alguns casos, a classificação pode não ser suficiente e precisamos também saber a localização de um objeto em uma imagem. Nesse caso, além de prever a classe (gato, cachorro, avião, pessoa), o modelo deve prever as coordenadas da área (chamada de caixa delimitadora ou *bounding box*) onde o objeto está situado dentro da imagem, de acordo com a Figura 9 (b). A combinação Classificação + Localização funciona muito bem quando temos apenas um objeto em uma imagem, no entanto quando há múltiplos objetos em uma cena ou mesmo nenhum, é mais adequado o uso de técnicas de Detecção de Objetos (KARPATHY et al., 2016). Essa tarefa é apresentada na Figura 9 (c).

Por fim, temos a segmentação, também chamada de segmentação semântica. Mais precisamente, a segmentação da imagem é o processo de atribuição de um rótulo a cada pixel em uma imagem, de modo que os pixels com o mesmo rótulo compartilhem certas características (KARPATHY et al., 2016). A tarefa de segmentação semântica é exemplificada na Figura 9 (d).

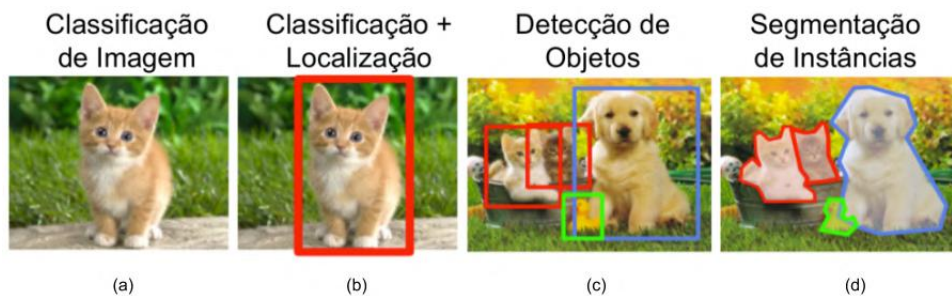


Figura 9 – Tarefas envolvendo imagens. Fonte: Adaptado de (KARPATHY et al., 2016)

No paradigma do aprendizado não-supervisionado não há a presença de rótulos nos dados, sendo que o algoritmo de AM deve aprender a agrupá-los por similaridade, processo que é chamado de *clustering* (TAN; STEINBACH; KUMAR, 2005).

Neste trabalho, foi utilizado o paradigma de AM supervisionado. As técnicas escolhidas são descritas nas subseções seguintes. São elas: *K-Nearest Neighbors*, *Random Forest*, *Support Vector Machine* e *Deep Learning*.

### 2.4.1 K-Nearest Neighbors - KNN

O *K-Nearest Neighbors*, também chamado de K-Vizinhos-Mais-Próximos, é um dos algoritmos de aprendizado supervisionado mais simples e difundidos da literatura. O que caracteriza esse algoritmo é o fato do aprendizado ser baseado em instâncias (SHAKH-NAROVICH; DARRELL; INDYK, 2006). É um algoritmo não paramétrico, ou seja, não faz nenhuma suposição sobre a distribuição de dados subjacente. Além disso é conhecido como *lazy* ou preguiçoso, ou seja, método de aprendizagem em que a generalização, além

dos dados de treinamento, é adiada até que uma consulta seja feita ao sistema, ou seja, não possui etapa de treinamento.

Uma nova instância de teste é classificada por uma votação majoritária de seus  $k$  vizinhos mais similares. Os vizinhos mais próximos de  $K$  são computados usando uma função de distância, em geral definida em termos da distância Euclidiana (TAN; STEINBACH; KUMAR, 2005). Também pode ser usada a distância de Mahalanobis. A Figura 10 apresenta um exemplo onde uma nova instância pode ser classificada como classe 1 ou classe 2, dependendo do valor de  $K$  vizinhos a ser considerado.

Uma das vantagens deste algoritmo está na simplicidade da implementação e por não requerer uma etapa de treinamento. Por outro lado, uma das desvantagens destes métodos é o alto custo computacional para classificar novas instâncias, pois todos os cálculos são feitos apenas na etapa de classificação. Para minimizar esses problemas, técnicas de indexação de exemplos de treinamento podem reduzir o tempo de processamento. Outra desvantagem é que são considerados todos os atributos das instâncias para classificar instâncias similares, ou seja, está fortemente relacionado ao número de dimensões, sendo suscetível à Maldição da Dimensionalidade (Curse of Dimensionality) (TAN; STEINBACH; KUMAR, 2005).

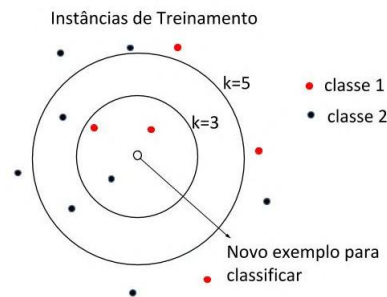


Figura 10 – Classificação de novas instâncias usando KNN. Fonte: Elaborada pelo autor.

## 2.4.2 Random Forest - RF

*Random Forest* (RF) (BREIMAN, 2001) é uma técnica de aprendizado supervisionada baseada em comitês de classificadores, normalmente Árvore de decisão (AD) e seleção aleatória de características. O RF é composto por um conjunto de classificadores estruturados em árvore, em que cada árvore lança um voto unitário para a classe mais popular na instância de entrada. A Figura 11 exemplifica esse processo.

Sua popularidade é impulsionada principalmente por sua alta eficiência computacional durante o treinamento (SAFFARI et al., 2009). Nas árvores padrão, cada nó é dividido usando a melhor divisão entre todas as características. Em contrapartida, em uma floresta

aleatória, o nó de cada AD é dividido usando um subconjunto aleatório de características. Cada árvore de classificação é construída usando uma amostra de *bootstrap* do conjunto de treinamento (em geral, apenas dois terços), e os dados restantes servem para validar o modelo. O RF apresenta uma baixa tendência em sobreajustar (em inglês, *overfitting*), sendo considerado um dos algoritmos de aprendizado mais precisos. Além disso, ele é executado eficientemente em grandes conjuntos de dados e pode lidar com um grande número de características.

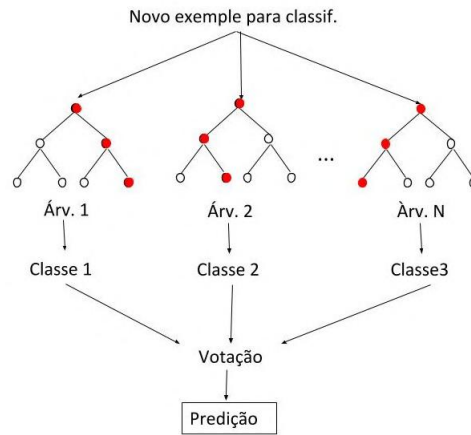


Figura 11 – Classificação de novas instâncias usando RF. Fonte: Elaborada pelo autor.

### 2.4.3 Support Vector Machine - SVM

O classificador *Support Vector Machine* (SVM), proposto por (CORTES; VAPNIK, 1995), tem como ideia básica encontrar uma função que separe os dados rotulados, em um hiperespaço  $d$ -dimensional. Em problemas de duas classes linearmente separáveis, SVMs determinam um classificador linear  $f$  que melhor separa os dados de treinamento. A Figura 12 (a) ilustra um exemplo de classificador linear.

Vários hiperplanos de separação podem ser usados em um classificador linear, no entanto, a seleção do limite pode ser melhor determinada através da maximização da margem  $m$  ilustrada na Figura 12 (b), que é a distância entre os valores mais próximos de cada classe, também chamados de vetores de suporte (em inglês, *support vectors*).

Em casos onde as classes não são linearmente separáveis, é necessário o cálculo de uma função *kernel*, que recebe dois pontos  $x_i$  e  $x_j$  do espaço de entradas e computa o produto escalar  $\Phi(x_i)\Phi(x_j)$  no espaço de características. Para tanto, é necessário escolher uma função de *kernel* adequada, bem como os parâmetros dessa função. Um dos *kernels* mais usados é polinomial, que possui o parâmetro  $D$  que determina o grau do polinômio.

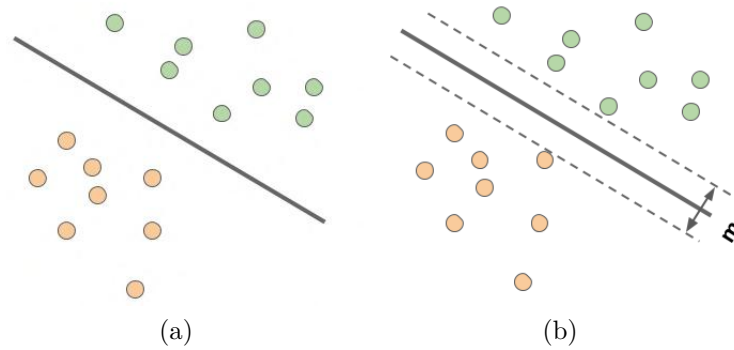


Figura 12 – Classificadores lineares; (a) Duas classes de pontos (pontos laranjados e verdes) e um separador linear. (b) Um separador linear com maior margem  $m$  distante dos vetores de suporte. Fonte: Elaborada pelo autor

#### 2.4.4 Deep Learning - DL

Métodos de Aprendizado Profundo (*Deep Learning*) são atualmente o estado-da-arte em muitos problemas possíveis de se resolver via AM, em particular problemas de classificação, detecção de objetos e segmentação semântica (PONTI; COSTA, 2018). Os principais fatores que levaram a popularização do DL são: a disponibilidade de bases de dados com milhões de imagens (RUSSAKOVSKY et al., 2015) e de computadores capazes de reduzir o tempo necessário para realizar o processamento dessas bases de dados (PONTI; COSTA, 2018). Além disso, anteriormente alguns estudos se dedicaram a explorar a aplicação de representações hierárquicas com redes neurais, tais como o Neocognitron de Fukushima (FUKUSHIMA, 1988) e a rede neural para reconhecimento de dígitos de LeCun (LECUN et al., 1998).

Métodos que utilizam DL buscam descobrir um modelo (exemplo: regras e parâmetros) utilizando um conjunto de dados (exemplos) e um método para guiar o aprendizado do modelo a partir desses exemplos. Ao final do processo de aprendizado é gerada uma função capaz de receber como entrada, os dados de entrada e fornecer como saída uma representação adequada para o problema. A diferença em relação às técnicas tradicionais de AM está na forma como se aprende a função  $f(\cdot)$ . De forma geral, métodos tradicionais de AM buscam diretamente por uma função. Em contrapartida, em DL é usado uma composição de funções, da seguinte forma:

$$f(x) = f_c(\dots f_2(f_1(x_1))\dots),$$

onde cada função  $f_c(\cdot)$  (o índice  $c$  se refere comumente a uma “camada”), recebe como entrada um vetor de dados  $x_c$  (preparado para a camada  $c$ ), gerando como saída o próximo vetor  $x_{c+1}$ . Portanto, uma das ideias centrais em DL é aprender sucessivas representações dos dados, intermediárias, ou seja, a profundidade permite aprender uma sequência de funções que transformam vetores mapeando-os de um espaço a outro, até al-

cançar o resultado desejado (PONTI; COSTA, 2018). Outra diferença do DL em relação aos métodos tradicionais de AM, está na extração de característica. Em DL a extração de característica é realizada nas camadas intermediárias, ao passo que nos métodos convencionais a extração de características é feita manualmente.

#### 2.4.4.1 Convolutional Neural Networks - CNN

Um dos métodos de DL mais difundidos em atividades de visão computacional consiste nas redes convolucionais, também conhecido na literatura como *Convolutional Neural Networks* (CNN). O que caracteriza esse tipo de rede é ser composta basicamente de camadas convolucionais, que processa as entradas considerando campos receptivos locais. A principal aplicação das CNNs é para o processamento de informações visuais, em particular imagens, pois a convolução permite filtrar as imagens considerando sua estrutura bidimensional (espacial) (PONTI; COSTA, 2018).

As CNNs são um tipo de rede neural *feed-forward*. Elas recebem uma entrada (pixels de uma imagem), realizam operações matemáticas entre os dados de entrada e os pesos (multiplicação de matrizes) e calculam as probabilidades para cada possível classe de saída. Em geral, uma camada de uma CNN é composta por 3 operações, sendo a convolução, função de ativação não-linear e uma camada de *pooling* respectivamente.

A convolução é uma função linear que se baseia na multiplicação de duas matrizes. Uma operação de convolução possui 3 componentes principais, conforme ilustrado na Figura 13, sendo: a imagem de entrada, o detector de características (*kernel*) e o mapa de características (*feature map*) resultante desta operação.

O tamanho do *kernel*, varia de acordo com o parâmetro escolhido, sendo muito comum na literatura o uso de uma janela de tamanho 3x3. Outros 2 parâmetros que determinam diretamente no *feature map* resultante é o *stride* e o *padding*. O *stride* é o tamanho do passo do *kernel* ao percorrer a imagem e o *padding* define como a borda da imagem de entrada é tratada. O *kernel* é uma janela deslizante que irá percorrer a imagem de entrada da esquerda para a direita e de cima para baixo de acordo com o valor do *stride* definido, realizando a multiplicação entre a sua matriz e o contexto atual em que a janela se encontra, somando os valores da multiplicação resultante. Na maioria das vezes, os valores do *kernel* são inicializados aleatoriamente o que resulta em mapas desconexos para a visão humana. Contudo, a medida que a rede apreende, os valores são ajustados e a rede decide por si própria a configuração mais adequada.

Em algumas arquiteturas de rede é possível encontrar operações de deconvolução ou transposição convolucional, que consiste na operação inversa da convolução. Esta nomenclatura gera uma certa divergência, pois matematicamente, a deconvolução gera os valores originais de uma matriz antes da convolução. Porém, no contexto das CNNs, a deconvolução apenas reconstroí a resolução espacial da imagem, de forma que os valores originais antes da convolução não são obtidos. A Figura 14 é uma representação de uma



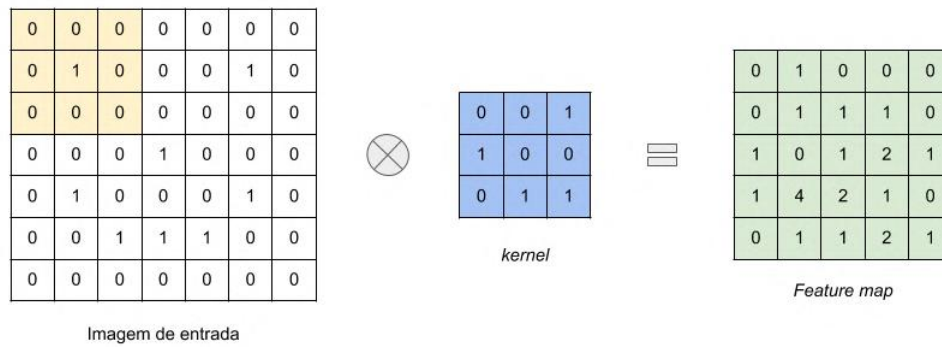


Figura 13 – Exemplo de uma operação de convolução com *stride* 1 e *kernel* 3x3. Fonte: (EREMENKO, 2018)

operação de deconvolução. A parte em verde localizada no centro da entrada é o mapa de característica gerado pela última operação realizada.

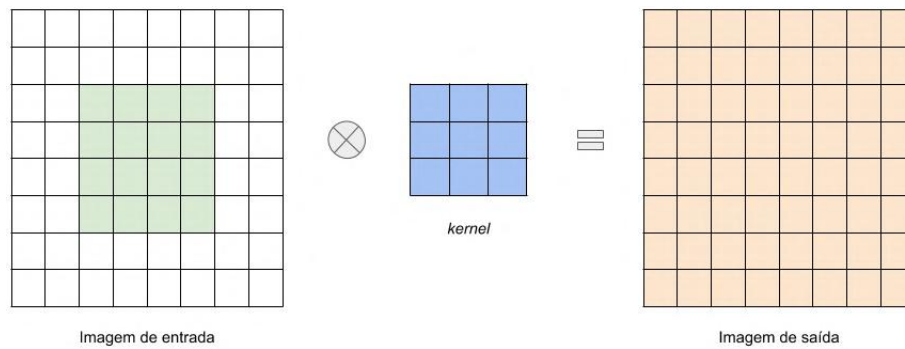


Figura 14 – Exemplo de uma operação de deconvolução com *kernel* 3x3 e sem padding. Fonte: Adaptada de (PRÖVE, 2017)

Outra operação encontrada em uma cada de CNN é a função de ativação não-linear. Em DL, uma das mais usadas é a ReLU (*Rectified Linear Unit*) que é descrita da seguinte forma  $f(z) = \max(0, z)$ , onde  $z$  é o valor de entrada e o valor propagado será  $z$  se ele for maior que 0, ou 0 caso contrário. Até o momento, não existe uma justificativa razoável pela ReLU ser melhor que outras funções de ativação na literatura (GOODFELLOW; BENGIO; COURVILLE, 2016).

Por último, a camada de *pooling* é usada para reduzir o tamanho dos *feature maps* usando alguma função que sumariza uma sub-região (DUMOULIN; VISIN, 2016). Embora exista uma diversidade de funções para a operação de *pooling*, a mais usada e aceita na literatura no momento é a *max pooling*, que propaga para as próximas camadas o maior valor encontrado em uma janela  $N \times N$  do *feature map*. A Figura 15 ilustra a operação de max pooling.

Embora as CNNs sejam amplamente utilizadas para classificação, em muitas tarefas visuais, especialmente em processamento de imagens biomédicas, a saída desejada deve incluir localização, exigindo a atribuição de um rótulo de classe para cada pixel. Esta

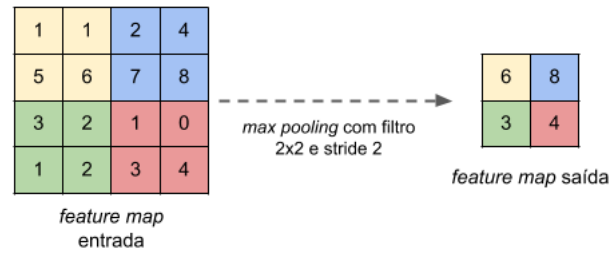


Figura 15 – Exemplo de uma operação de max pooling. Fonte: Adaptada de (KARPATHY et al., 2016)

é a principal ideia da aplicação de segmentação semântica em imagens usando CNNs (RAZZAK; NAZ; ZAIB, 2018).

Algoritmos de segmentação semântica recentes, convertem a arquitetura CNN existente construída para classificação, uma arquitetura totalmente convolucional (*Fully Convolutional Network* – FCN). Eles obtêm um mapa de rótulos da rede classificando cada região local na imagem e realizando uma simples deconvolução, que é implementada como interpolação bilinear, para rotulagem em nível de pixel. Além disso, novas propostas, introduzem a ideia da rede de deconvolução para gerar um mapa de probabilidade de classe *pixel-wise* densa, por operações consecutivas de descompactação, deconvolução e retificação. Uma dessas arquiteturas CNN modificadas chamada de U-Net (RONNEBERGER; FISCHER; BROX, 2015), vem sendo utilizada com sucesso especialmente em tarefas de segmentação de imagens biomédicas (RAZZAK; NAZ; ZAIB, 2018). Outra rede do estado-da-arte também é detalhada como a PSPNet (ZHAO et al., 2017), que se sagrou campeã no desafio ImageNet (ZHOU et al., 2019) na categoria análise de cena no ano de 2016.

#### 2.4.4.2 U-Net

A arquitetura da U-Net é caracterizada por duas etapas, sendo elas a **contração** e a **expansão**. A rede recebe como entrada uma imagem que é propagada pela rede, gerando como resultado uma imagem segmentada. A contração consiste em uma sucessão de camadas que são alteradas por operações de convolução e *max pooling*, de forma que a resolução espacial da imagem é diminuída por um fator de dois, ao passo que o número de filtros ou mapas de característica aumentam por este mesmo fator. Na expansão a imagem é reconstruída novamente, usando operações de convolução e convolução transposta, revertendo os efeitos da primeira parte, reduzindo o número de filtros e aumentando a resolução espacial, ambos em um fator de dois.

Um aspecto importante desta rede é o uso de conexões de salto ou “skip”, que concatenam no canal os resultados de convoluções executadas antes de cada conjunto máximo para os efeitos de cada convolução transposta. Essas conexões de saltos são correspondidas para que o resultado antes da primeira camada do conjunto máximo seja concatenado

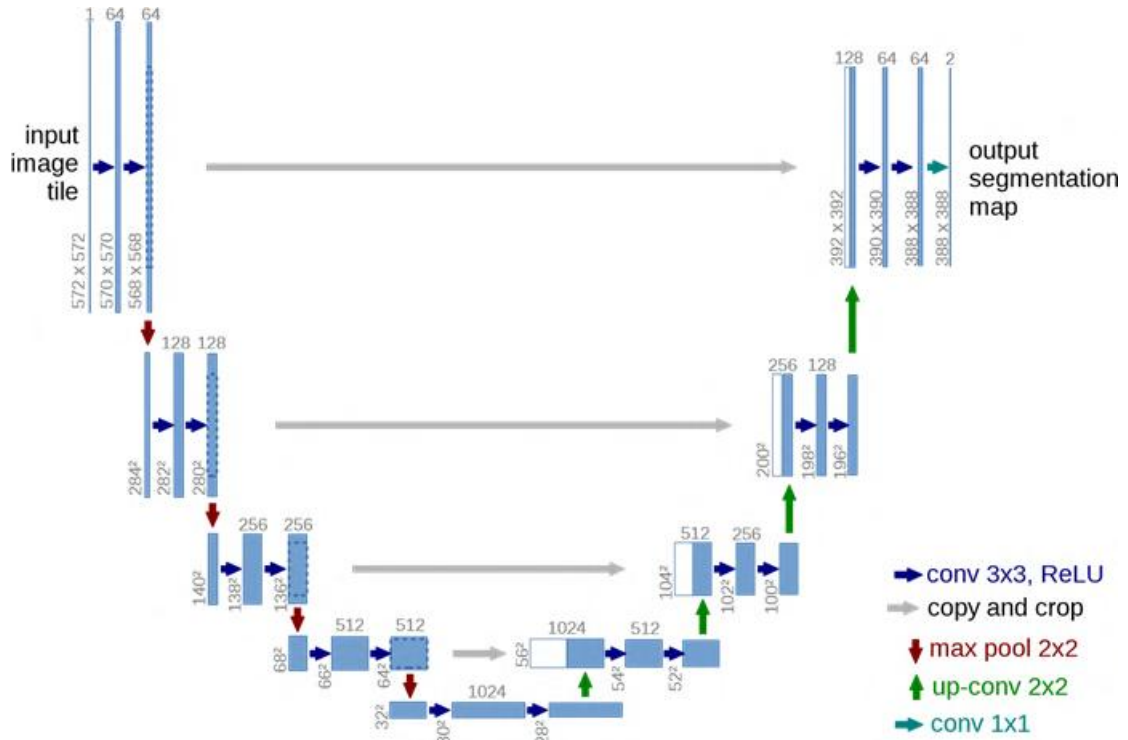


Figura 16 – Exemplo da arquitetura U-Net. Cada bloco azul corresponde a um mapa de características. O número de blocos é indicado na parte superior do bloco. O tamanho do bloco é mostrado no canto inferior. Os blocos brancos representam os mapas de característica copiados. As setas denotam diferentes operações. Fonte: (RONNEBERGER; FISCHER; BROX, 2015)

ao resultado da última convolução transposta. A vantagem das conexões é que elas permitem que a rede aprenda um resultado inicial em poucas etapas de treinamento, sem a necessidade de modificar a maioria dos parâmetros de rede. De certa forma, eles permitem que o processo de treinamento seja incremental, onde os parâmetros mostrados no meio da arquitetura podem ser os últimos a serem aprendidos e, como resultado, ajudam a evitar os mínimos locais.

#### 2.4.4.3 PSPNet

Essa arquitetura trabalha com informações de contexto global e, em seguida, agrega esse contexto a diferentes regiões usando o módulo de agrupamento piramidal (em inglês, *Pyramid Pooling Module*). Este módulo demonstra ser um contexto global útil, que funde recursos em quatro níveis distintos de pirâmide.

O primeiro nível é responsável por produzir uma saída (bin). O segundo nível separa o mapa de características em várias sub-regiões e faz representação em pool para vários locais. Em seguida, a saída desses níveis apresentam mapas de características com tamanhos diferentes. Para manter o peso do recurso global, de repente os autores aplicam uma camada de convolução de  $1 \times 1$  após cada nível de pirâmide para diminuir a dimensão do design de contexto para  $\frac{1}{N}$  do original se o tamanho do nível do A pirâmide é N. Ter-

ceiro nível é aplicado um upsample, que o recurso de baixa dimensão mapeia para obter o mesmo recurso de tamanho usando interpolação bilinear. O quarto nível concatena os recursos para produzir o recurso global PP final. A Figura 17 exibe uma visão geral da arquitetura PSPNet.

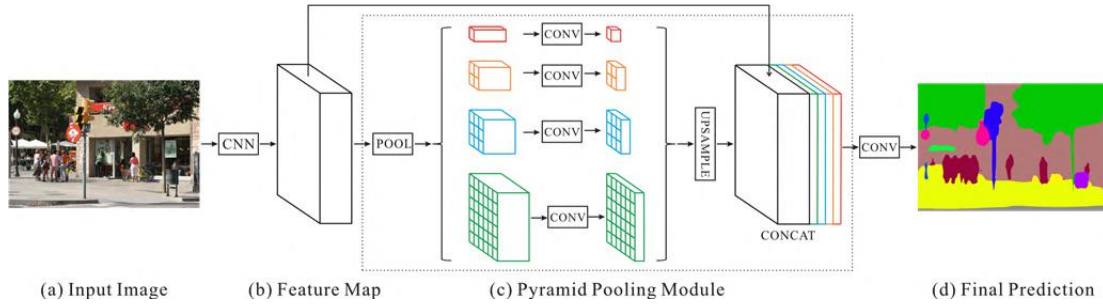


Figura 17 – Visão geral da arquitetura da rede PSPNet. (a) A imagem é fornecida como entrada. (b) Usa uma CNN pré-treinada para obter o mapa de características da última camada. (c) Em seguida, um módulo PP é aplicado para obter diferentes representações de sub-regiões, seguido por camadas de upsampling e concatenação para formar a representação de característica final, que carrega informações de contexto locais e globais. (d) Finalmente, a representação é alimentada em uma camada de convolução para obter a previsão final por pixel. Fonte: (ZHAO et al., 2017)

O número de níveis de pirâmide e o tamanho de cada nível podem ser modificados, associados ao tamanho do mapa de recursos. A formação de sub-regiões com núcleos de agrupamento variável em menos passos. Dada uma imagem de entrada, a PSPNet usa uma ResNet (HE et al., 2016) pré-treinada com a estratégia de rede dilatada (CHEN et al., 2014) para extrair o mapa de características. O tamanho final do mapa de características é  $\frac{1}{8}$  da imagem de entrada. Aplicando a pirâmide de 4 níveis, os kernels de pool cobrem a imagem. Eles são combinados como o global, concatenando o anterior com o mapa de recursos original, acompanhado por uma camada de convolução para produzir o mapa de previsão final.

## 2.5 Partição e Avaliação de Desempenho

Espera-se que um classificador apresente desempenho adequado para dados não vistos. A literatura propõe algumas técnicas para particionar os dados, bem como avaliar o desempenho dos classificadores. Uma das técnicas mais simples para particionamento é conhecida como *Holdout* (ou *split-sample*), no qual a maior parte dos dados, normalmente dois terços, é usada como dados de treinamento, e o restante é usado como dados de teste (TAN; STEINBACH; KUMAR, 2005). A Figura 18 exibe um exemplo da partição dos dados usando o *holdout*. Essa técnica é indicada para bases com grande quantidade de dados. Por outro lado, considerando uma quantidade pequena de dados, problemas como

classes sub-representadas ou super-representadas podem surgir, além de que quanto menor o conjunto de treinamento, maior a variância do classificador.

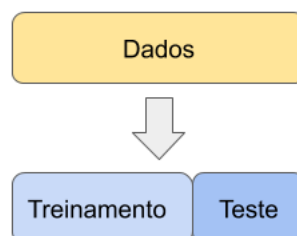


Figura 18 – Particionamento dos dados usando a técnica *holdout*. Fonte: Elaborada pelo autor.

Uma alternativa em relação ao *holdout* pode ser a técnica *Random Subsampling* ou subamostragem aleatória, que consiste na repetição do *holdout* várias vezes, na qual em cada iteração uma amostragem aleatória é gerada para os dados de treinamento e teste (TAN; STEINBACH; KUMAR, 2005).

Já a técnica validação cruzada (em inglês, *Cross validation*) (KOHAVI et al., 1995) consiste em particionar o conjunto de dados em subconjuntos mutuamente exclusivos, utilizando alguns subconjuntos para treinamento e o restante para teste. O processo de divisão dos dados em treinamento e teste segue o método k-pastas, que consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos de mesmo tamanho. A cada iteração uma das k partições é usada para testar o modelo e as outras k-1 são usadas para treinar o modelo, conforme ilustrado na Figura 19.

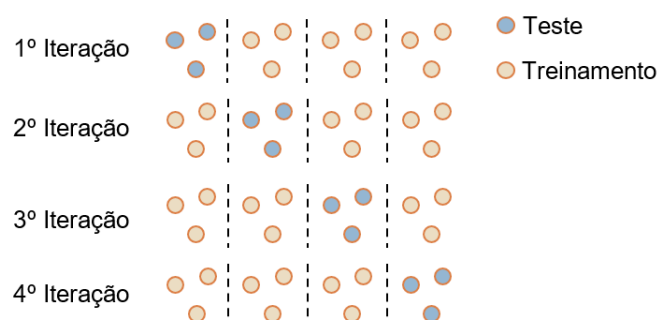


Figura 19 – Exemplo de validação cruzada com  $K$ -pastas ( $K = 4$ ). Fonte: (BACKES, 2018)

A avaliação do desempenho de um modelo de classificação é baseada nas contagens de registros de teste corretamente e incorretamente previstas pelo modelo. Essas contagens são tabuladas em uma tabela conhecida como matriz de confusão. A Tabela 2 descreve a matriz de confusão para um problema de classificação binária.

Tabela 2 – Matriz de confusão para duas classes. Extraído de (TAN; STEINBACH; KUMAR, 2005)

		Classe predita	
		<i>Classe = 1</i>	<i>Classe = 0</i>
Classe	<i>Classe = 1</i>	$f_{11}$ (TP)	$f_{10}$ (FN)
Real	<i>Classe = 0</i>	$f_{01}$ (FP)	$f_{00}$ (TN)

Cada entrada  $f_{ij}$  da tabela denota o número de registros da classe  $i$  preditos como da classe  $j$ . Desta forma, o número total de predições corretas feitas pelo modelo é ( $f_{11} + f_{00}$ ) e o número total de predições incorretas é igual a ( $f_{10} + f_{01}$ ). A partir destes dados, pode-se calcular a acurácia (ACC) e a taxa de erro (ERR):

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (1.1)$$

$$ERR = \frac{FN + FP}{TP + FN + FP + TN} \quad (1.2)$$

Entretanto, é necessário fazer uso de outras métricas para a avaliação do modelo. Considerando um desbalanceamento entre as classes, a acurácia e taxa de erro podem não ser as medidas mais adequadas (SOMEREN; WIDMER, 1997). Desta forma, (LEWIS; GALE, 1994) desenvolveram a medida-F (*F-Measure*), que combina as medidas de precisão (*Precision*) que representa a proporção entre classes preditas corretas, e revocação (*Recall*) que demonstra a taxa de falsos negativos, de acordo com as seguintes equações:

$$Precisão = \frac{TP}{TP + FP} \quad (1.3)$$

$$Revocação = \frac{TP}{TP + FN} \quad (1.4)$$

$$Medida-F = 2 * \left( \frac{Precisão * Revocação}{Precisão + Revocação} \right) \quad (1.5)$$

## 2.6 Trabalhos Relacionados

Em (SOUZA et al., 2015) o conjunto de dados é composto por 154 imagens e desse conjunto foram selecionadas 10 imagens para treinamento e 5 para teste. Após extração manual das características, aplicou-se uma normalização nos dados (para cada característica  $f$  é subtraído pela média do valor do atributo  $f$  na base de dados e o resultado é dividido pelo desvio padrão do valor do atributo  $f$  na base de dados). Em seguida os dados são usados para comparação entre 4 técnicas de AM supervisionado, sendo elas KNN, RF, GP e *Multilayer Perceptron* (MLP). A classificação considerando todos os algoritmos é aplicada pixel à pixel na imagem, ou seja, como saída é obtida uma imagem segmentada,

exibida na Figura 20. Os dados foram particionados usando a técnica *holdout*, ou seja, a maior parte para treinamento e o restante para teste e utilizou-se a medida-F para avaliação dos resultados.

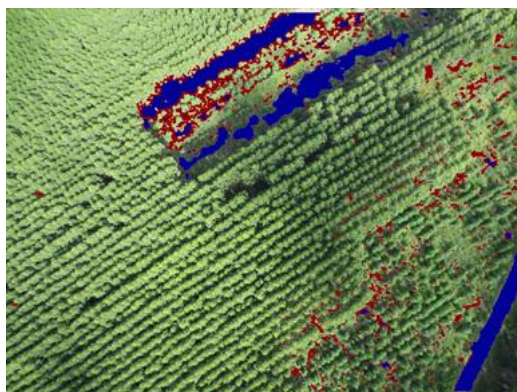


Figura 20 – Resultado da segmentação para o GP. As regiões em vermelho representam as áreas com a presença da praga e em azul a estrada. Fonte: (SOUZA et al., 2015)

Para encontrar o modelo mais adequado o autor utilizou diferentes tamanhos de blocos (configuração do bloco interno) e para cada algoritmo avaliou diferentes valores de hiper-parâmetros. Os resultados mostram que o GP obteve o maior F-measure (0.80) na identificação das regiões com a presença de praga. Além disso são fornecidos os melhores parâmetros para cada algoritmo. Também é avaliada a influência do bloco contextual utilizando o melhor modelo, obtido através do GP. Observa-se que a presença do bloco contextual é importante para se obter um modelo adequado.

Joalland et al. (2018) apresenta um método para identificação de nematóides em plantações de beterraba utilizando imagens aéreas hiperspectrais. Algumas plantações susceptíveis e tolerantes aos nematóides foram selecionadas e avaliadas, indicando que as características que melhor diferenciam as plantações susceptíveis e tolerantes aos nematóides são os índices espectrais relacionados a clorofila, nitrogênio e a água e a temperatura da copa da planta. A combinação das características utilizando AD (usando a implementação J48 do *Weka* com poda) permitiram melhorar a discriminação das regiões susceptíveis e tolerantes aos nematóides.

Vassallo-Barco et al. (2017) descreve que para classificar as amostras obtidas com base nas características extraídas pelos descritores, foram escolhidos três classificadores: KNN com hiper-parâmetro ( $K = 1$ ), ANN com 3 camadas treinadas com o algoritmo backpropagation e o classificador bayesiano Naive Bayes (NB). Segundo o autor, para avaliação dos dados utilizou-se as medidas de Precisão, Revocação e medida-F. Para particionar os dados utilizou-se a técnica de validação-cruzada com 10-pastas considerando todos os cenários.

Avaliando os resultados, é possível inferir que o sistema obteve melhores resultados na identificação de deficiência nutricional dos elementos Ferro (Fe) e Boro (B). De acordo



com o autor, esses resultados podem estar relacionados ao fato de os sintomas associados às deficiências de Boro e Ferro serem mais notáveis do que aqueles associados aos demais nutrientes, desta forma, sendo identificados mais facilmente. Outra correlação observada pelo autor está no fato do descritor GLCM apresentar melhores resultados na identificação na deficiência de Ferro.

Em relação à comparação direta entre os desempenhos relacionados ao BSM e aos descritores do GLCM, não há uma clara superioridade de um descritor em relação ao outro. Finalmente, uma comparação direta entre as três abordagens de classificação supervisionada conclui que o uso de informações associadas ao descritor BSM, para todas as deficiências o classificador com melhores resultados de precisão foi o classificador NB, obtendo uma precisão geral de 65.05%, ao passo que o ANN obteve 59.11% e 46.09%. Por outro lado, utilizando o descritor GLCM, nenhum dos classificadores se destacou.

Chen et al. (2018) apresenta um método para detecção automática da praga da espécie *aphid nymphs*, sendo uma das pragas que causam maior degradação em plantas mantidas em estufas, principalmente pela rápida capacidade de reprodução. No trabalho o autor não detalha qual cultura foi utilizada. Inicialmente os dados são coletados por meio de cameras digitais e são obtidas imagens (RGB) da folha de uma planta. Neste trabalho, foi proposto um método para segmentação usando a arquitetura da CNN U-Net modificada, e em seguida é aplicada uma etapa para contagem da ocorrência da praga na folha. Além disso é realizada uma comparação na etapa de segmentação entre o método proposto e outros dois métodos baseados em limiar por cor (neste caso utilizou-se uma transformação de cor para o espaço YCbCr e foi utilizado o canal Cr) e o segundo baseado na abordagem proposta em (MAHARLOOEI et al., 2017) que tem como objetivo identificar a presença de pragas da espécie *Aphis glycines* em folhas de soja, respectivamente.

Os experimentos apontam inequivocamente que o método proposto é mais eficiente na identificação da praga da espécie *aphid nymphs*. Aplicando a medida-F, o método proposto alcança 0.96, enquanto os outros métodos ambos ficam em torno de 0.39. A Figura 21 exibe a imagem segmentada, considerando o método proposto e os outros métodos usados como comparação.

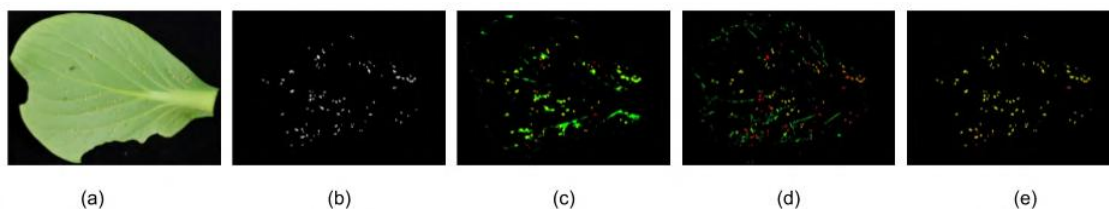


Figura 21 – (a) Imagem original. (b) Anotação manual. (c) Segmentação utilizando o método 1. (d) Segmentação utilizando o método 2. (e) Segmentação usando o método proposto Fonte: Adaptado de (CHEN et al., 2018)

A Tabela 3 apresenta uma comparação entre os principais trabalhos relacionados e a nossa proposta. Cada um dos trabalhos, conta com alguma das etapas que compõem um



sistema de detecção automático de pragas, citadas neste capítulo. Em relação a nossa proposta, a cultura escolhida é a cafeeira, que também é investigada em (VASSALLO-BARCO et al., 2017), no entanto, nesta proposta utiliza-se imagens aéreas coletadas por VANT, enquanto que em (VASSALLO-BARCO et al., 2017) a coleta é feita de forma manual por meio das folhas do café.

Para a extração de características, a proposta deste trabalho realiza a extração baseada em características como cor e textura quanto a extração baseada em redes convolucionais. Em relação aos algoritmos, neste trabalho usa-se 3 algoritmos clássicos da literatura além de duas redes convolucionais para segmentação como U-Net e PSPNet. Por último, este trabalho avalia as 3 medidas, precisão, revocação e medida-F.

Tabela 3 – Comparação entre os principais trabalhos correlatos a proposta de trabalho.

Referência	Cultura	Coleta/ Tipo imagem	Extração Característica	Classificação Segmentação	Métricas Avaliação
(SOUZA et al., 2015)	Eucalipto	VANT/ digital (RGB)	cor e textura	KNN, RF, ANN e GP	Precisão, Revocação e medida-F
(VASSALLO-BARCO et al., 2017)	Café	manual (folha)/ digital (RGB)	textura	K-NN, NB e ANN	Precisão, Revocação e medida-F
(JOALLAND et al., 2018)	Beterraba	VANT/ hiperspectral	-	AD	Acurácia e Kappa
(CHEN et al., 2018)	-	manual (folha) digital (RGB)	baseada em CNN	U-Net	Precisão, Revocação, medida-F e <i>Dice</i>
Nossa proposta	Café	VANT/ digital (RGB)	cor e textura / baseada em CNN	KNN, RF, GP, U-net e PSPNet	Precisão, Revocação e medida-F



## Materiais e Métodos

Neste capítulo é apresentado o método de pesquisa adotado para a tarefa de detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas obtidas por VANT. O método é composto pelas etapas de coleta de dados, seleção das imagens e rótulo, a etapa de pré-processamento é opcional, extração de característica, método de treinamento e predição e medidas de avaliação. As etapas são apresentadas na Figura 22 ilustrada a seguir.

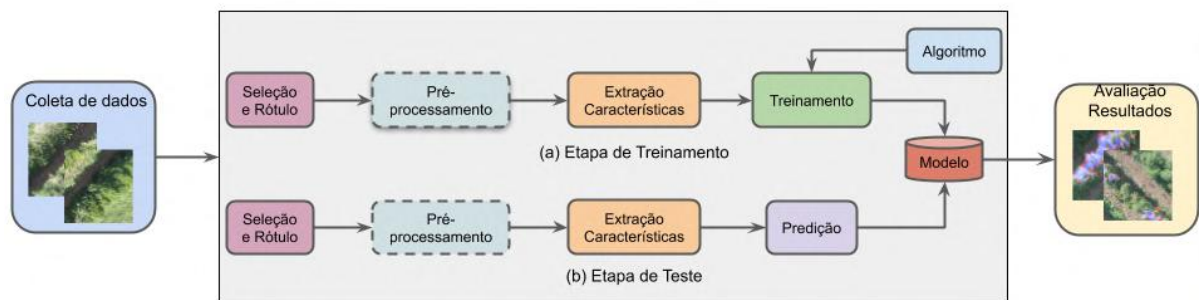


Figura 22 – Metodologia para detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas. Fonte: Elaborada pelo autor

### 3.1 Coleta de dados

A coleta das imagens foi realizada em uma lavoura cafeeira na fazenda Novo Mundo, situada no município de Monte Carmelo do estado de Minas Gerais (Brasil) por meio de um VANT *Phantom 4 Pro* (Figura 23). O VANT é equipado com uma câmera digital RGB permitindo a aquisição de imagens de altíssima resolução ( $4864 \times 3648$ ) 72 dpi. Além disso, o VANT é equipado com sensores de detecção de obstáculos, evitando uma eventual colisão com a plantação.

Nesta etapa houve a colaboração de uma especialista na cultura cafeeira que foi responsável por indicar as áreas cafeeiras que estavam infectadas pelos nematóides, além



Figura 23 – *Phanton* 4 Pro usado para coletar as imagens. Fonte: Elaborada pelo autor

de guiar a todos até a plantação para mostrar os aspectos visíveis da praga na planta, como o amarelamento, surgimento de galhas e reboleiras. Em seguida foi realizado um planejamento visando definir quais os períodos do ano seriam os mais adequados para realização dos vôos e em quais horários haveria menor impacto de ruídos como sombras, por exemplo.

### 3.1.1 Áreas Coletadas

A coleta das imagens foi realizada em 3 áreas, sendo as áreas A e B, exibidas na Figura 24(a) com altura de 20 metros (m) no período compreendido em Maio de 2018 entre às 10h e 12h da manhã e a área C, na qual as imagens foram coletadas em 3 alturas diferentes, sendo 10, 15 e 20 m em Dezembro de 2018 entre às 12h e 13h da tarde, ilustrada na Figura 24(b). Nesse contexto, altura é a distância vertical entre o solo e a altura sobrevoada pelo VANT.

A área A é composta por uma área de 0,35 h que corresponde a 3494 m<sup>2</sup> e perímetro de 257 m; A área B conta com uma área de 0,67 h, correspondendo a 6657 m<sup>2</sup> e 341 m de perímetro. Ao todo, considerando a intersecção das duas áreas, a união das áreas A e B equivale a 0,84 h ou 8345 m<sup>2</sup>. Já a área C é composta por uma área de 0,1 h que equivale a 569 m<sup>2</sup> e 96m de perímetro.

Em relação ao número de imagens coletadas por área, na área A foram obtidas 77, na área B 144, ambas considerando uma altura de 20 m. Já para área C foram obtidas 86 imagens coletadas a uma altura de 10 m, 37 imagens com altura de 15 m e 25 imagens com altura de 20 m. A subseção 3.1.2 detalha o processo de análise nematológica que consiste em uma análise laboratorial das amostras do solo com o objetivo de constatar a presença do parasita nematóide, bem como a quantidade. O teste foi realizado somente na Área C (Figura 24(b)).

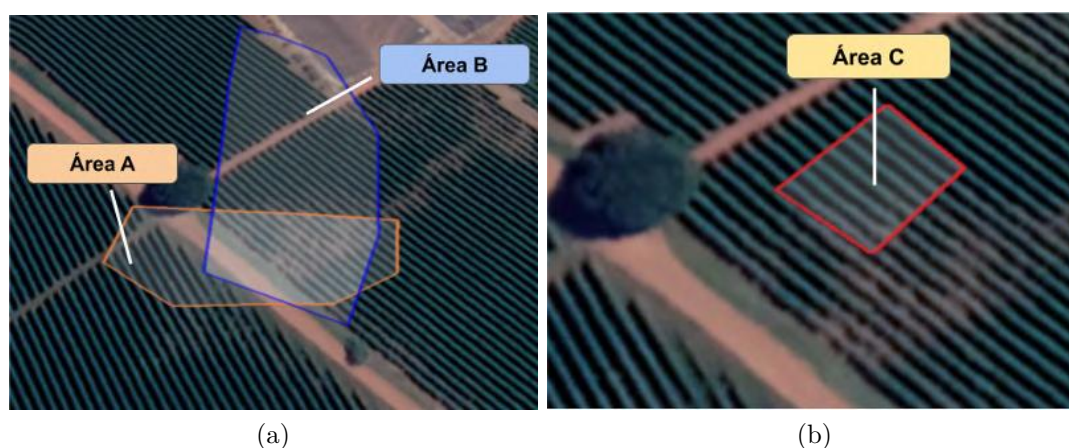


Figura 24 – Vista aérea das áreas obtidas na etapa da coleta das imagens da Fazenda Novo Mundo - Monte Carmelo (MG/BR) localizada por meio das coordenadas - 18°49'36.98", -47°20'19.29". (a) Áreas A e B coletadas à 20m em Abril/2018. (b) Área C coletada em 3 alturas em Dezembro/2018. Fonte: Google Earth, imagens obtidas em 19/04/2017 com altitude de 1.23km.

### 3.1.2 Análise nematológica

As amostras para análise nematológica foram coletadas em uma área de 0,1 ha plantada (Área C ilustrada na Figura 24(b)). Foi efetuado o caminhamento em zigue-zague no talhão, coletando-se as amostras de solo e raiz de 15 plantas de toda a área analisada. O solo foi amostrado na projeção da copa dos cafeeiros na profundidade de 20 cm. Posteriormente as amostras simples foram homogeneizadas, resultando em uma amostra composta contendo 500 g de solo e 100 g de raiz, sendo posteriormente analisadas no Laboratório de Nematologia Agrícola da Universidade Federal de Uberlândia.

Uma alíquota de 150 cm<sup>3</sup> do solo homogeneizado foi processada pelo método da flutuação centrífuga em solução de sacarose para a extração de nematóides (JENKINS, 1964). A suspensão obtida foi utilizada para a determinação do número de juvenis de segundo estágio no solo, com auxílio de câmara de contagem de Peter.

Verificou-se no resultado da análise nematológica da área a presença de 2.400 juvenis e/ou adultos de *Pratylenchus coffeae*/150 cm<sup>3</sup> de solo e 406 ovos e/ou juvenis de 2º estágio de *Meloidogyne exigua*/g de raízes. Ambas as espécies detectadas no talhão são nematóides parasitos do cafeeiro, sendo o gênero *Meloidogyne* promotor de galhas radiculares e o gênero *Pratylenchus* associado à formação de lesões radiculares.

## 3.2 Seleção e Rotulagem das Imagens

Após concluída a etapa de coleta das imagens, a resolução de todas elas foi reduzida para (1280×960) por meio da ferramenta de código livre GIMP (em inglês, *GNU Image Manipulation Program*), usando o método de interpolação bicúbica (GONZALEZ; WO-

ODS, 2010). Em seguida, foram selecionadas as imagens manualmente, com o apoio da especialista na cultura cafeeira, priorizando a escolha daquelas onde melhor se destacavam os aspectos visíveis dos nematóides, ilustrado na Figura (25).



Figura 25 – Exemplo de nematóide em lavoura cafeeira. Fonte: Elaborada pelo autor

Posteriormente, as imagens foram rotuladas manualmente, sob a supervisão da especialista. O processo de rotulagem (em inglês, *ground truth* (GT)), consiste em delimitar as regiões de interesse na imagem e indicar qual a sua respectiva classe ou categoria. A rotulagem das imagens pode ser feita de acordo com dois métodos. No primeiro deles gera-se uma nova imagem correspondente à imagem original com o rótulo e no segundo método é gerado um arquivo de anotações para cada imagem. Os métodos são detalhados nas respectivas subseções 3.2.1 e 3.2.2. Por fim, são definidos dois critérios para rotulagem das imagens, o primeiro deles adota um critério flexível e o segundo um critério rígido. Os critérios são descritos na subseção 3.2.3.

### 3.2.1 Rótulo como Imagem

Para esta rotulagem, gera-se uma nova imagem chamada de imagem de rótulos, com o mesmo tamanho da imagem original. Nesta etapa, são identificadas duas classes, sendo praga e não-praga. Os pixels na imagem original, correspondentes às regiões com praga (infestação dos nematóides), são destacados na cor branca na imagem de rótulo e o fundo na cor cinza, representando as áreas onde não há a infestação de pragas. Esse processo é ilustrado na Figura 26, com o exemplo de 3 imagens, juntamente com o seus respectivos rótulos. Para a criação dos rótulos foi utilizada a ferramenta GIMP na versão 2.8.

### 3.2.2 Rótulo como Anotação

Nesta rotulagem, para cada imagem é gerado um arquivo de anotações correspondente. Cada anotação é formada por um conjunto de polígonos que são usados para delimitar a região de interesse e para cada polígono é atribuído o valor de uma classe. Neste método, são identificadas 3 classes, sendo regiões com praga (infestação nematóides), regiões saudáveis (ausência dos nematóides) e o solo. Para a criação dos rótulos foi utilizada a



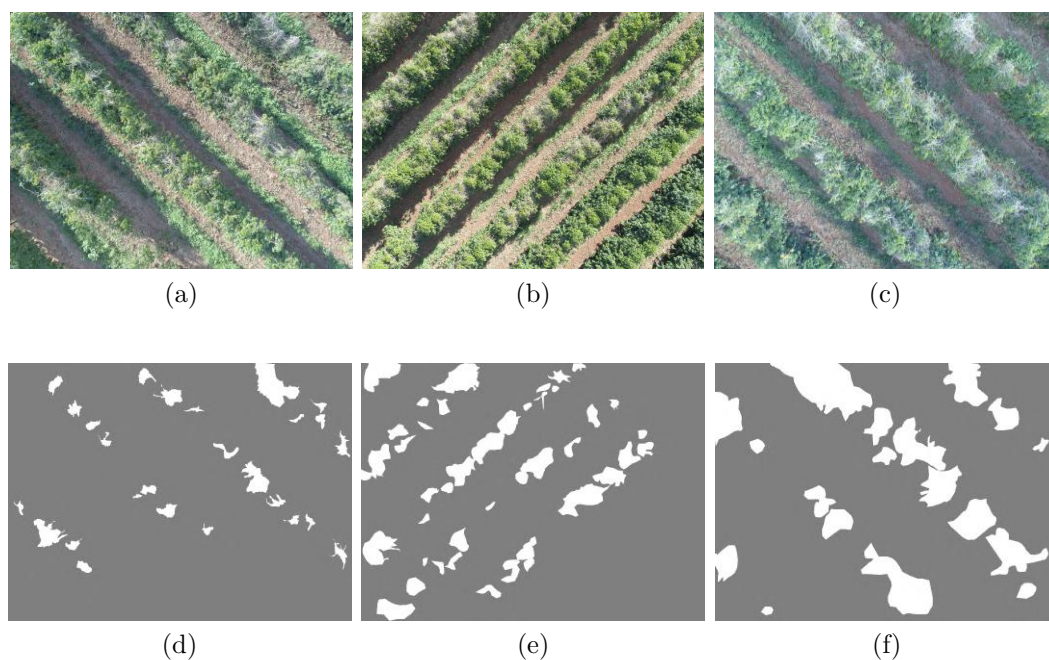


Figura 26 – (a) e (d) Imagem original 1 e rótulo. (b) e (e) Imagem original 2 e rótulo. (c) e (f) Imagem original 3 e rótulo.

ferramenta gráfica de anotações *Labelme* (WADA, 2016) na versão 3.5.0, disponibilizada na plataforma Anaconda com *Python* na versão 3.6.

Utilizando a ferramenta *Labelme*, inicialmente a imagem deve ser carregada e em seguida as regiões de interesse são delimitadas por meio de polígonos e identificadas com o rótulo de uma das classes. A Figura 27, ilustra a imagem rotulada no *Labelme*, na qual os polígonos em verde delimitam as regiões com praga, em azul as partes saudáveis, em vermelho o solo e o restante não é rotulado. Após concluir a rotulagem, a ferramenta permite que as anotações contendo os rótulos sejam geradas para a imagem no formato *JSON* (*JavaScript Object Notation*). Assim, rótulos são gerados com mesmo nome da imagem, formato .json.

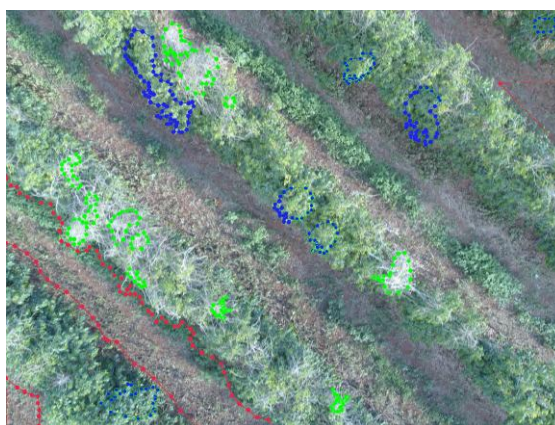


Figura 27 – Exemplo de imagem rotulada pelo *Labelme*. Fonte: Elaborada pelo autor

### 3.2.3 Critérios para Rotulagem

Durante o processo de rotulagem, em muitos casos houve uma dificuldade em se identificar a presença de pragas em determinadas regiões, mesmo com o apoio da especialista na cultura cafeeira. E em outros, a maior dificuldade encontrada foi na forma como as regiões de praga estavam distribuídas, muitas vezes de maneira esparsa. A Figura 28 ilustra os desafios na identificação das regiões com praga.

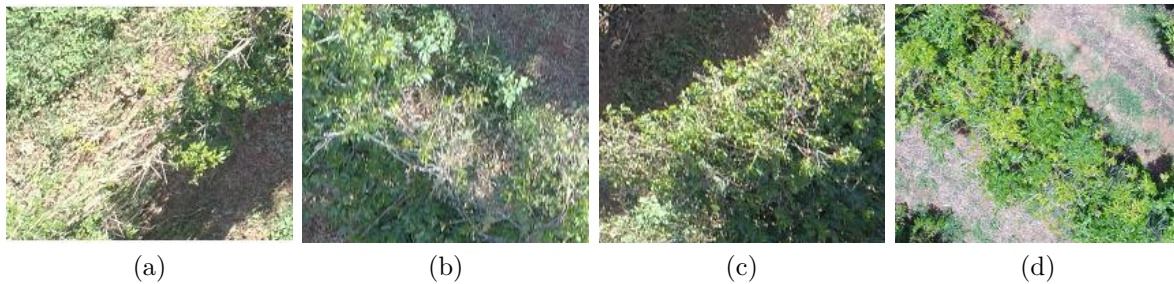


Figura 28 – Exemplo de regiões que evidenciam os desafios no processo de rotulagem. Em (a) e (b) há uma dificuldade na identificação das regiões com praga, por conta da sobreposição com o solo. Em (c) e (d) a distribuição das pragas ocorre de maneira esparsa, dificultando a rotulagem.

Com o intuito de reduzir possíveis equívocos no momento de realizar o processo de rotulagem, adota-se dois critérios para criação dos rótulos. O primeiro deles é o critério flexível, no qual havendo alguma dúvida na real categoria (ou classe) de uma determinada região da imagem a ser rotulada (como nos exemplos ilustrados na Figura 28), então essas regiões serão ignoradas no momento da rotulagem. Caso não haja nenhuma dúvida ou impasse na identificação da região, então a mesma é rotulada com sua respectiva classe. Portanto, no critério flexível, alguns pixels da imagem podem não serem rotulados.

O segundo é chamado de critério rígido, pois todas as regiões da imagem são rotuladas, sem exceção, mesmo havendo algum impasse na identificação, todos os pixels da imagem são rotulados com alguma classe.

## 3.3 Pré-Processamento

Ao finalizar a etapa de seleção e rotulagem, as imagens e os respectivos rótulos gerados podem ser pré-processados de forma opcional, conforme ilustrado na Figura 22, onde esta etapa é exibida de maneira tracejada. Nesta etapa, podem ser realizadas duas etapas de pré-processamento. A primeira consiste na segmentação manual para remoção do solo e a segunda na execução de um script para redimensionamento e extração dos rótulos gerados como anotações. Cada uma das etapas são aplicadas de forma independente. A seguir as etapas são descritas.



### 3.3.1 Segmentação Manual

Esta etapa consiste na segmentação manual para remoção das regiões de solo nas imagens. A Figura 29, ilustra um exemplo da imagem original e após a segmentação. A segmentação é realizada por meio da ferramenta GIMP na versão 2.8, onde utiliza-se a ferramenta de seleção livre para delimitar as regiões de solo. Desta forma, os pixels correspondentes às regiões com solo são definidos como preto e o restante dos pixels referente às regiões de plantação e praga são mantidos intactos.



Figura 29 – Exemplo de pré-processamento aplicado para uma imagem. (a) Imagem original. (b) Imagem Segmentada.

O intuito do passo de segmentação é compreender o real impacto que o solo causa na identificação das regiões com nematóides, pois dependendo do estágio ou grau de severidade da doença, os aspectos visíveis dos nematóides manifestados na plantação apresentam coloração semelhante ao solo, podendo dificultar a tarefa de detectar as pragas. Ademais, a ideia é estabelecer o teto que o método para detecção de pragas pode alcançar, considerando o melhor cenário, que é quando o solo foi removido manualmente.

### 3.3.2 Extração das Anotações

Nesta etapa são fornecidas as imagens com os respectivos rótulos gerados como anotações (descrito na subseção 3.2.2). Para extração das anotações foi criado um script em Python na versão 3.7.0, juntamente com as bibliotecas openCV na versão 3.4.1, Torch na versão 0.4.1, json na versão 2.0.9 e NumPy na versão 1.15.4.

O *script* recebe como parâmetros o caminho das imagens juntamente com o local dos rótulos no formato .json, além da configuração da resolução e o dicionário que tem como chave a classe e o respectivo valor. Inicialmente a imagem é carregada para um *array* com 3 dimensões e aplica-se uma função para redimensionar a imagem na altura e largura especificada. Em seguida os valores do *array* contendo a imagem são alterados para formato *int32* e são colocados no intervalo compreendido entre [0 e 255].

Considerando os rótulos, primeiramente é criada uma matriz chamada *geraLabel* com 2 dimensões com a altura e largura especificada e inicializada com todas as posições do array

com valor igual a -1. Logo após, o arquivo no formato .json é carregado e são extraídas as anotações dos polígonos criados no momento da rotulagem. Para cada polígono, os pontos que o delimitam são transformados em um *array* e em seguida a respectiva posição do polígono na matriz *geraLabel* é preenchida com o valor definido para a classe no dicionário. Neste caso, se o polígono delimitar o solo, a posição do polígono é preenchida com 0, se for saudável é preenchida com o valor 1 e se for praga com o valor 2. O preenchimento da matriz *geraLabel* é ilustrado na Figura 30. Finalmente o *array* contendo a imagem e a matriz *geraLabel* são armazenado em um dicionário de amostras.

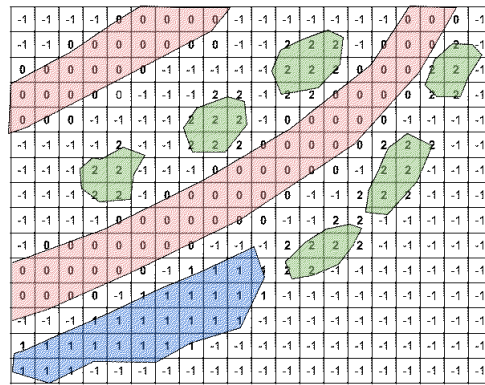


Figura 30 – Exemplo de uma matriz  $20 \times 15$  em menor escala para o rótulo, onde as regiões de solo (vermelho) são preenchidas com valor 0, saudáveis (azul) com o valor 1, praga (verde) com o valor 2 e o restante com o valor -1. Fonte: Elaborada pelo autor

## 3.4 Extração de Características

Em posse das imagens após a etapa de pré-processamento e seus respectivos rótulos, a próxima etapa é caracterizada como extração de características. Neste trabalho foram aplicadas duas técnicas distintas para extração de características, sendo elas a abordagem baseada em blocos, de forma análoga à descrita em (SOUZA et al., 2015), que utiliza esta técnica para obter as características de pragas no contexto da cultura do eucalipto e a extração baseada em DL, que atualmente é o estado da arte na extração de características envolvendo imagens. Ambas as técnicas de extração são descritas a seguir.

### 3.4.1 Baseada em Blocos

A abordagem de extração baseada em blocos permite receber como entrada as imagens originais, ou seja, sem aplicação de nenhuma etapa de pré-processamento, bem como imagens segmentadas manualmente (descrito na subseção 3.3.1). Os rótulos gerados devem ser fornecidos como imagens (descrito na subseção 3.2.1). Além disso, devem ser

fornechas as configurações dos parâmetros como tamanho dos blocos, quais os extratores de características a serem usados e as características a serem extraídas.

Nessa abordagem um par de blocos sobrepostos e com tamanhos distintos são usados para obter as características de uma amostra. O conjunto das amostras obtidas é conhecido como matriz de característica, usada para geração dos conjuntos de dados de treinamento e teste.

O par de blocos é composto pelo Bloco Interno (BInt) e pelo Bloco externo (BExt). O objetivo do BInt além de fornecer as características primárias da amostra, tem como função estabelecer a classe da amostra. Desta forma o BInt também pode ser chamado de bloco de classificação. Já o BExt também chamado de Bloco Contextual fornece as informações referentes à borda do BInt (sem sobreposição), ou seja, propriedades contextuais.

Selecionar a configuração mais adequada de tamanho para o bloco BInt pode ser desafiador, pois essa escolha esta intimamente relacionada ao contexto do problema a ser solucionado. O tamanho adequado de bloco deve garantir o menor número possível de perda de exemplos rotulados como pragas. Ao analisar as imagens, constatou-se que a distribuição das pragas se dá em reboleiras (concentrado) e em outros casos de maneira esparsa. Desta forma, utilizou-se 4 configurações de tamanho distintas para o bloco BInt, sendo  $10 \times 10$ ,  $20 \times 20$ ,  $30 \times 30$  e  $40 \times 40$  pixels com o intuito de avaliar qual o tamanho mais adequado. Em relação ao BExt, a configuração é definida em função do tamanho do BInt, isto é, o BExt é configurado com o dobro do tamanho do BInt.

O processo de extração das amostras consiste em percorrer a imagem por meio de uma janela deslizante, que movimenta-se da esquerda para a direita e de cima para baixo, usando o par de blocos BInt e BExt. Cada passo realizado pela janela na imagem é obtido de acordo com o tamanho do BInt. Em seguida, são extraídas as mesmas características do BInt e do BExt e finalmente o conteúdo de ambos são concatenados juntamente com a respectiva classe para formar uma nova amostra. A composição de todas as amostras obtidas pode ser agrupada por meio da tabela de vetores de característica. A Figura 31 ilustra o processo de extração de característica descrito.

Durante a extração de característica, algumas estratégias são usadas para a seleção das amostras, para evitar que exemplos ambíguos sejam fornecidos ao classificador. Ao iniciar a extração, o segmento do bloco BInt é obtido, isto é, a posição da janela na imagem. Essa mesma posição é verificada na respectiva imagem rotulada para obter uma tupla (T) com dois valores, sendo a proporção de pixels definidos como saudáveis (S) e doentes (D) para o segmento, onde a soma de S e D é menor ou igual 100%. Caso 100% dos pixels dessa janela estiverem definidos como saudável, ou seja ( $S = 100\%$ ) ou no mínimo 80% (definido de maneira arbitrária) dos pixels estiverem configurados como doente ( $D \geq 80\%$ ), então as características dessa janela são extraídas e a classe da amostra é atribuída a categoria (S ou D) do maior valor da tupla T. Caso contrário, o bloco BInt e consequentemente seu

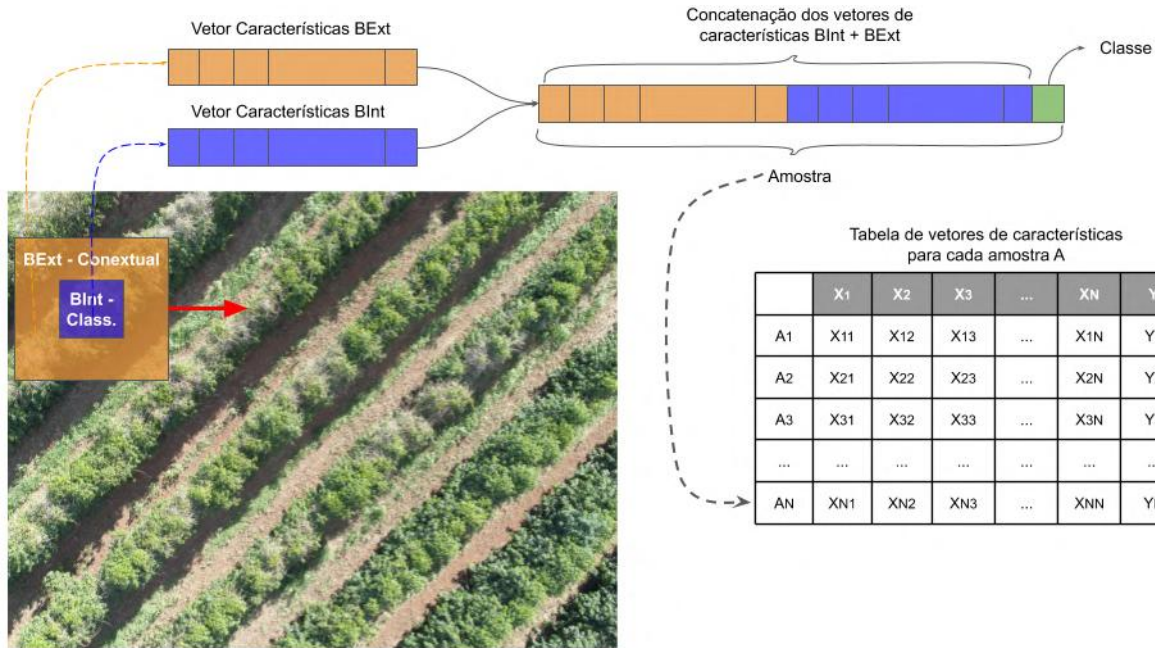


Figura 31 – Processo de extração de característica usando a abordagem por bloco. A Seta vermelha indica a direção da janela deslizante. Fonte: Elaborada pelo autor

BExt (contexto) são desconsiderados para extração. Neste caso, há uma maior tolerância em relação aos exemplos doentes, visto que há um número muito menor de amostras com praga em relação às saudáveis.

Considerando que a imagem seja segmentada, a princípio é obtido o segmento do bloco BExt, em seguida verifica-se na respectiva imagem convertida em tons de cinza (em inglês, *gray*). Caso o valor de algum pixel do segmento seja menor que 25 (cor preta) indicando a presença do solo neste segmento avaliado, então não é extraída as características deste par de blocos. Caso contrário, as características são extraídas.

#### 3.4.1.1 Informações Obtidas para os Blocos

Para ambos os blocos BInt e BExt, as mesmas 115 características foram extraídas, considerando descritores baseados em cor e textura, da seguinte forma:

- ❑ média e desvio padrão para cada componente de cada um dos seguintes sistemas de cor, RGB, HSV, LAB, YCbCr, XYZ, YIQ, YUV e o valor mínimo apenas para o HSV, contribuindo com 45 características;
- ❑ média e desvio padrão do nível de cinza e entropia, além do valor máximo considerando a escala de cinza, fornecendo 5 características;
- ❑ diferença da média do nível de cinza da imagem e a média de cinza no bloco a ser extraído, produzindo uma única característica;

- ❑ LBP com 4 vizinhos, resultando em 16 características para cada um dos componentes (LBP GRAY, LBP RED, LBP GREEN and LBP BLUE), contribuindo 64 características.

### 3.4.1.2 Redução da Dimensionalidade

Ao todo o vetor de característica é composto por 230 características. Pensando na composição da matriz de característica, é possível perceber o alto grau de dimensionalidade devido à elevada quantidade de características, tornando o modelo complexo e susceptível à maldição da dimensionalidade. Deste modo, foram utilizados dois métodos para a redução da dimensionalidade baseados na seleção de características disponíveis no software Weka (HALL et al., 2009), visando melhorar a eficiência computacional dos algoritmos e simplificando o modelo gerado.

O primeiro algoritmo de seleção utilizado é baseado no uso de um *wrapper* (*wrapperSubsetEval*) (KOHAVI; JOHN, 1997), que avalia o conjunto de características usando um esquema de aprendizado. Alguns parâmetros são configurados para utilização do método, que aplica o esquema chamado de *BestFirst*, que busca o espaço de características utilizando uma abordagem gulosa. Além disso o classificador J48 baseado em árvores é aplicado. Esse método fornece 20 características para cada bloco BInt e BExt, totalizando 40 ao todo, sendo:

- ❑ média para os seguintes componentes pertencentes aos sistema de cor: R e B (RGB), Y (YCbCr), Y e Z (XYZ), entropia, contribuindo com 6 características;
- ❑ média de desvio padrão para os componentes S (HSV), A e B (LAB), Q (YIQ), fornecendo 8 características;
- ❑ desvio padrão e valor máximo para o nível de cinza, fornecendo 2 características;
- ❑ diferença da média do nível de cinza da imagem e a média de cinza no bloco a ser extraído, produzindo uma única característica;
- ❑ uma característica para cada um dos componentes LBP GRAY, LBP RED e LBP GREEN, contribuindo com 3 características.

Outro método usado é baseado na correlação dos atributos (*CorrelationAttributeEval*) (HALL, 2000). Esse método gera uma classificação das melhores características ordenadas do maior para o menor valor de correlação entre o atributo em análise e o atributo alvo (classe). Foram selecionados as características com valor de correlação maior ou igual 0,20. Portanto, todas as características com valor menor que 0,20 são removidas. Esse método fornece 17 características para cada bloco BInt e BExt, totalizando 34 ao todo, sendo:

- ❑ média para cada um dos componentes RGB e XYZ, contribuindo com 6 características;
- ❑ média para os seguintes componentes dos sistemas de cor: V (HSV), L (LAB), Y (YCbCr), Y (YIQ) e Y (YUQ), fornecendo 5 características;
- ❑ desvio padrão para cada um dos componentes XYZ, contribuindo com 3 características;
- ❑ média e valor máximo para o nível de cinza, fornecendo 2 características;
- ❑ diferença da média do nível de cinza da imagem e a média de cinza no bloco a ser extraído, produzindo uma única característica.

Inicialmente o conjunto de dados é gerado contendo apenas as características relativas ao bloco BInt, neste caso com 115 características. Em seguida é aplicado o método de redução de dimensionalidade para seleção das características mais relevantes e finalmente é gerado um novo conjunto de dados com as características fornecidas pelos algoritmos de seleção, em seguida, são adicionadas as características correspondentes do bloco BExt.

### 3.4.2 Baseada em Redes Neurais Convolucionais (CNN)

Diferentemente da abordagem baseada em blocos, nas redes convolucionais a extração de características é feita de forma automática. As características são extraídas pelos filtros (em inglês, *kernels*) presentes em cada camada por meio de operações de convolução e *pooling*. Normalmente nas primeiras camadas são extraídas características mais simples e de difícil compreensão para seres humanos, como por exemplo as bordas. À medida que as camadas superiores vão sendo ativadas, características de mais alto nível são extraídas, ou seja, a tendência é que sejam obtidas características mais discriminativas. Neste trabalho, foram utilizadas duas arquiteturas próprias para segmentação semântica, detalhadas a seguir.

#### 3.4.2.1 U-Net

Neste trabalho é proposto uma modificação na rede U-Net, especificamente na etapa de contração, conforme ilustrado na Figura 32. Por padrão, a etapa de contração pode ser vista como uma sucessão de 5 blocos formados por operações de convolução e *pooling*. Cada bloco é composto por duas convoluções com kernel de tamanho  $3 \times 3$  e passo (em inglês, *stride*) igual a 1, seguido por uma operação de *max pooling*  $2 \times 2$  e passo igual a 2. Esta etapa é semelhante à arquitetura VGG com 13 camadas proposta em (SIMONYAN; ZISSERMAN, 2014).

A modificação proposta na arquitetura consiste em substituir a etapa da contração original da U-Net, por uma arquitetura do estado da arte, sendo a VGG16 (SIMONYAN;



ZISSERMAN, 2014), considerando somente a parte relativa a extração de características. Um dos pontos que motivaram esta alteração está no fato de que Simonyan e Zisserman (2014) concluíram que a medida que aumenta-se o número de camadas, há uma diminuição do erro de classificação. Neste caso, a VGG16 conta com 16 camadas e a etapa de contração original com 13 camadas. Esse processo é ilustrado na Figura 32, onde a etapa original de contração da U-Net destacada em amarelo, recebe (indicado pela seta laranja) a etapa de extração de características de uma VGG16.

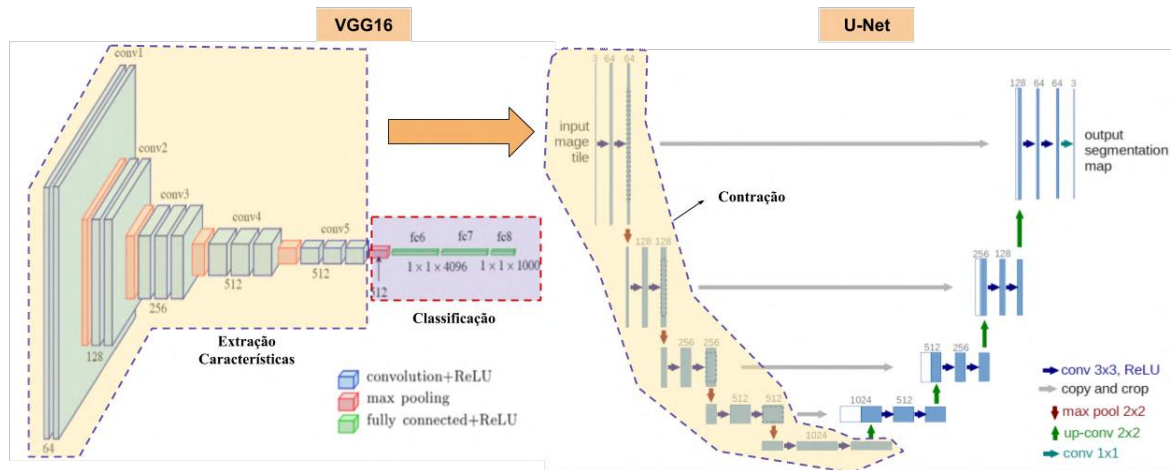


Figura 32 – A etapa original de contração da U-Net (destacada em amarelo) é substituída por uma VGG16, considerando somente a parte relativa a extração de característica (destacada em amarelo). Fonte: VGG adaptada de (FERGUSON et al., 2017) e U-Net adaptada de (RONNEBERGER; FISCHER; BROX, 2015)

Devido a relativa baixa quantidade de imagens, optou-se por utilizar uma VGG16 pré-treinada na base de dados ImageNet (DENG et al., 2009) que conta com 1 milhão e 200 mil imagens, rotuladas em 1000 classes, processo descrito como transferência de aprendizado (do inglês *transfer learning*). A arquitetura final da U-Net após a combinação com a VGG16 é ilustrada na Figura 33.

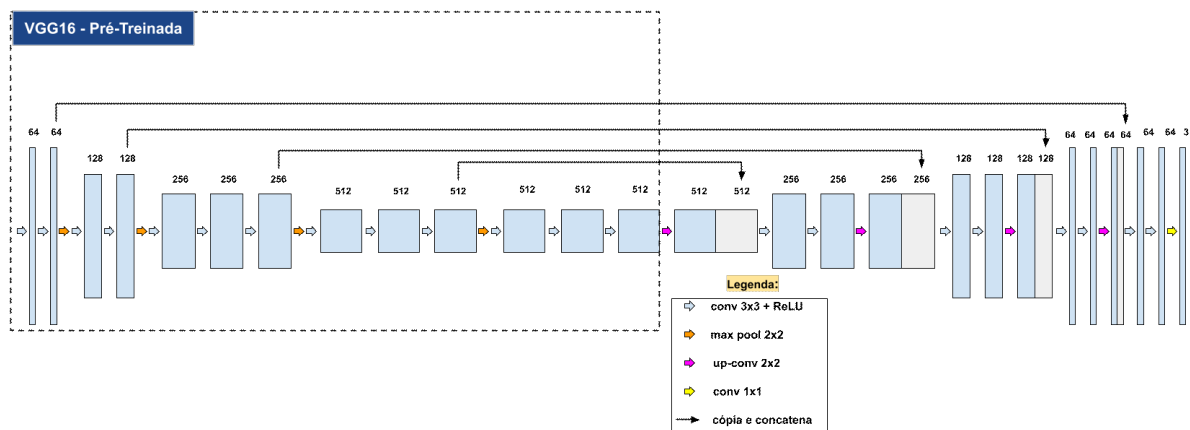


Figura 33 – Modificação na arquitetura U-Net proposta. Fonte: Elaborada pelo autor

### 3.4.2.2 PSPNet

Em relação à PSPNet é proposta uma modificação na rede convolucional utilizada para fornecer os mapas de características para a etapa chamada de módulo de agrupamento piramidal (em inglês, *Pyramid Pooling Module*). Zhao et al. (2017) sugere em seu artigo a utilização da arquitetura do estado da arte ResNet50 pré-treinada usando a base de dados ImageNet (DENG et al., 2009). Desta forma, a extração de características é feita pela ResNet50, desconsiderando a camada completamente conectada, fornecendo 2048 características.

Neste trabalho, optou-se por utilizar a rede ResNet101 pré-treinada. A principal diferença entre as redes está no número de blocos convolucionais na terceira camada (destacada em roxo na Figura 34). A ResNet101 apresenta 23 blocos convolucionais, sendo cada um deles formado por 3 operações convolucionais, enquanto que a ResNet50 é composta por 6 blocos convolucionais, também incluindo 3 operações de convolução para cada bloco. A arquitetura da rede PSPNet é ilustrada na Figura 34.

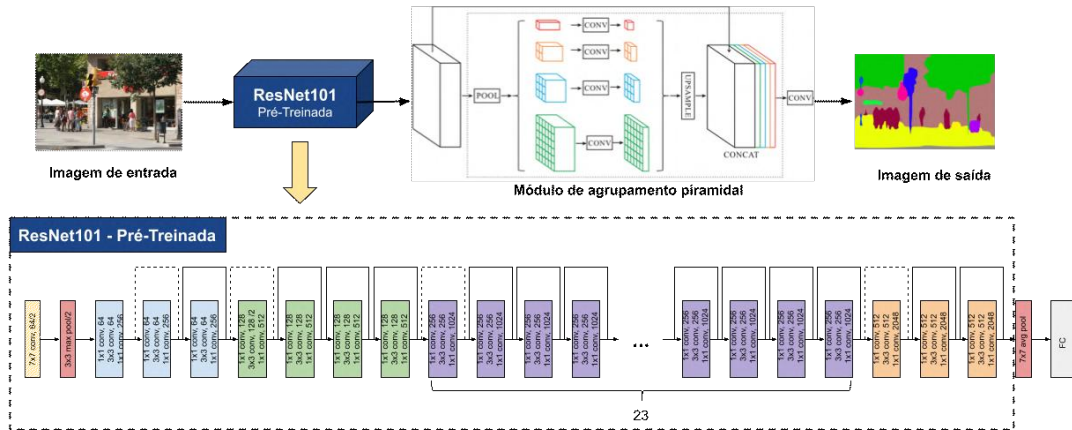


Figura 34 – Arquitetura de rede PSPNet usando uma ResNet101 pré-treinada. Fonte: Adaptada de (ZHAO et al., 2017)

## 3.5 Algoritmos de Classificação

Após concluída a etapa de extração de características, a etapa seguinte consiste em selecionar os algoritmos de classificação que serão usados para conduzir a etapa de treinamento (conforme ilustrado na Figura 22, que descreve a metodologia adotada). Foram selecionados 5 algoritmos, sendo o KNN, RF, SVM e duas arquiteturas CNN, sendo U-Net e PSPNet que fornecem a classificação pixel à pixel para a imagem fornecida. A seguir, detalha-se o motivo da escolha de cada algoritmo, bem como as configurações e parâmetros definidos, além da implementação adotada.

A escolha do KNN foi motivada principalmente pelo fato deste algoritmo ser ideal para conjuntos pequenos ou médios. Neste caso, há uma escassez no número de imagens



utilizadas e consequentemente no volume de dados fornecidos para o treinamento. Outra questão que encorajou a escolha por esta técnica é que as predições fornecidas pelo algoritmo são baseadas em informações locais. Neste contexto, como a praga se distribui de maneira esparsa, esse pode ser um bom indicativo. Para este algoritmo, adotou-se a implementação fornecida pelo Weka na versão 3.8.2, usando a configuração padrão, sendo 1 vizinho ( $K = 1$ ) e com distância euclidiana.

Em relação ao RF optou-se por utilizá-lo especialmente pela capacidade que este algoritmo tem em lidar com bases de dados de alta dimensionalidade e pela baixa tendência em sobreajustar (em inglês, *overfitting*) os modelos gerados. Neste caso, para a extração de características baseada em blocos (descrita na subseção 3.4.1) utiliza-se um número considerável de características inicialmente. Para este algoritmo, adotou-se a implementação fornecida pelo Weka na versão 3.8.2, utilizando a configuração de 100 árvores.

Para o SVM a escolha foi feita pois este algoritmo tem sido frequentemente mencionado na literatura por obter bons resultados de classificação em diversos domínios de aplicação, incluindo sensoriamento remoto (MOUNTRAKIS; IM; OGOLE, 2011). Além do mais é um dos algoritmos mais eficientes para problemas de elevada dimensionalidade. Para este algoritmo, adotou-se a implementação fornecida pelo Weka na versão 3.8.2, com *kernel* polinomial de grau 1 (definido arbitrariamente).

Por último, outros dois algoritmos baseados em CNN foram usados, sendo as arquiteturas U-Net e PSPNet. Que consiste em métodos baseados em DL, que são atualmente o estado da arte em tarefas de classificação e segmentação semântica em imagens.

A escolha da rede U-Net partiu de uma característica interessante adotada durante o treinamento, na qual utiliza-se uma estratégia de *data augmentation* nas imagens rotuladas (RONNEBERGER; FISCHER; BROX, 2015). Essa característica permite que não seja necessário fornecer um número muito grande de amostras para o treinamento. Além do mais, esta arquitetura apresentou importantes resultados na segmentação de imagens médicas. Em relação a PSPNet, pelo fato desta rede usar sugestões de contextos, o que tende a melhorar consideravelmente o desempenho em tarefas de segmentação semântica.

Ambas as arquiteturas U-Net e PSPNet foram implementadas em *Python* na versão 3.7.0, usando a biblioteca *torch* (*PyTorch*) na versão 0.4.1, que consiste em uma plataforma de DL de código aberto, além da biblioteca *Numpy* na versão 1.5.4.

## 3.6 Método e Medida de Avaliação

Após concluída a etapa de extração de características, bem como a definição dos algoritmos de classificação, a etapa seguinte consiste em usar essas informações extraídas e por meio dos algoritmos de classificação, executar a etapa de treinamento para gerar um modelo preditivo, que permita inferir se há ou não a presença de nematóides ao fornecer uma nova amostra desconhecida, etapa conhecida como predição ou teste. Em seguida, é

realizada a avaliação dos resultados. Este processo é ilustrado pela Figura 22.

Neste trabalho foram aplicados dois métodos distintos para as etapas de treinamento e predição. No primeiro as amostras são fornecidas com base na extração baseada em blocos (descrita na subseção 3.4.1), e no segundo usando a estratégia baseada em CNN (descrita na subseção 3.4.2) juntamente com o conjunto de imagens. Os métodos são detalhados nas respectivas subseções 3.6.1 e 3.6.2. Por fim, são utilizadas 3 medidas de avaliação para avaliar o desempenho dos classificadores, descrita na subseção 3.6.3.

### 3.6.1 Treinamento para a estratégia baseada em blocos

Nesta subseção, as amostras fornecidas ao treinamento e predição são obtidas com base na extração baseada em blocos (descrita na subseção 3.4.1), na qual gera-se uma tabela de vetores de características para todas as amostras obtidas, incluindo um índice que permita identificar de qual imagem a amostra foi extraída.

Em seguida os dados são subdivididos em dois conjuntos mutuamente exclusivos, sendo o conjunto de treinamento e o conjunto de teste ou predição. Essa divisão entre os conjuntos é aplicada de acordo com a técnica de validação cruzada (descrita em 2.5) com  $N$ -pastas, onde  $N$  é o número de imagens fornecidas. Então parte dos dados é particionada em treinamento e o restante para teste. Neste caso, são realizadas  $N$  iterações, representadas pelo seguinte conjunto  $\{i, i+1, i+2, \dots, N-2, N-1, N\}$  de imagens, na qual para cada iteração  $i$  o conjunto de teste é composto pelas amostras da imagem  $i$  e o conjunto de treinamento pelas amostras pertencentes às  $N-1$  imagens restantes, de forma que as amostras de cada imagem são avaliadas pelo menos uma vez no conjunto de teste.

A cada iteração, os conjuntos de treinamento e teste são normalizados de acordo com a técnica Z-score (JAIN; NANDAKUMAR; ROSS, 2005). Essa normalização é importante para que nenhum atributo domine os outros no momento da geração do modelo. Por último, são utilizados os algoritmos de classificação KNN, RF e SVM para conduzir o processo de treinamento. Neste caso, um modelo é gerado com base no algoritmo selecionado e em seguida as amostras fornecidas no conjunto de teste são avaliadas por esse modelo. Os valores para as  $N$  iterações são computados.

### 3.6.2 Treinamento para a estratégia baseada em CNN

Nesta subseção, as imagens e os rótulos são fornecidas em 3 subconjuntos distintos, sendo eles treinamento, validação e teste. O conjunto de treinamento é responsável por permitir que a rede aprenda os padrões com base nos dados fornecidos. Já o conjunto de validação é responsável por fornecer uma avaliação imparcial do modelo no conjunto de dados de treinamento, enquanto ajusta os hiperparâmetros da rede. Por fim, o conjunto de teste é usado para avaliar o modelo final gerado após o treinamento.

Para o conjunto de treinamento e validação os rótulos são fornecidos como anotações (descrito na subseção 3.2.2) e adota-se o critério flexível de rotulagem (descrito na subseção 3.2.3), processo no qual são rotuladas as regiões onde há a certeza da classe real. Em caso de dúvida ou impasse da real classe, essa região não é rotulada. Desta maneira, essa medida ajuda a não confundir a rede durante o treinamento, pois evita-se o uso de amostras ambíguas.

Em relação ao conjunto de testes, os rótulos são fornecidos como anotações (descrito na subseção 3.2.2) e adota-se o critério rígido de rotulagem (descrito na subseção 3.2.3), no qual todas as regiões da imagem são identificadas com alguma classe. Como o conjunto de teste será avaliado, a única forma de fornecer uma avaliação justa para a imagem é rotulando todos os pixels com alguma classe. Caso fosse utilizado um critério flexível para este conjunto, muito provavelmente os resultados obtidos poderiam estar enviesados, visto que nos rótulos poderiam estar apenas as regiões onde a rede identifica com sucesso as regiões de nematóides. A seguir é descrito o processo de treinamento da CNN.

Inicialmente um novo modelo com a arquitetura da rede é instanciado passando como parâmetro o número de classes e em qual dispositivo o modelo será gerado, possibilitando a utilização de CPU ou GPU. Neste caso utilizou-se GPU.

Nesta etapa, são usados modelos pré-treinados para a U-Net e PSPNet (descrito na subseção 3.4.2) como um processo de *fine-tuning*, ou seja, ao usar um modelo pré-treinado na base de dados Imagenet, os parâmetros iniciais da rede são inicializados e em seguida o treinamento é retomado a partir destes pesos iniciais, mas agora utilizando o conjunto de dados de treinamento fornecido, isto é, as amostras com regiões com pragas, regiões saudáveis, etc. A abordagem adotada para realizar o *fine-tuning* permite que o algoritmo ajuste todos os pesos da rede com base nas imagens fornecidas para treinamento. Para ajustar os parâmetros da rede, permitindo que a função de custo seja reduzida durante a etapa de treinamento é usado o método Gradiente Descendente Estocástico (do inglês *Stochastic Gradient Descent*, SGD).

Para a rede U-Net as camadas são configuradas com taxas de aprendizado diferentes. As camadas da etapa de contração, considerando os pesos e o *bias* de cada uma são definidas com o valor de  $1e-6$  para taxa de aprendizado. Já as camadas da etapa de expansão, ambos os pesos e o *bias* são configurados com valor de 0.02 para taxa de aprendizado e os pesos de cada camada são definidos com decaimento (do inglês, *weight decay*) de  $5e-4$ . Para todos os parâmetros é usado *momentum* com valor de 0.9. A escolha destes parâmetros baseou em um trabalho cujo objetivo era a segmentação semântica de árvores e frutos. Em relação a rede PSPNet todos os parâmetros da rede são configurados com taxa de aprendizado igual a  $1e-2$ , *momentum* com valor de 0.9 e com decaimento de peso com valor  $1e-4$ . Neste caso, utilizou-se os parâmetros padrões.

Ambas as redes são treinadas por 100 épocas utilizando os dados de treinamento. Uma época é um passo completo no treinamento, ou seja, quando toda a base de dados

é processada.

Para cada época  $e^i$  de  $\{e^1, e^2, e^3, \dots, e^{98}, e^{99}, e^{100}\}$ , é gerado um novo modelo  $M^i$  onde é calculado a função de custo e a acurácia. Em seguida, é fornecido o conjunto de validação para  $M^i$  e calcula-se a função de custo e a acurácia para o conjunto de validação. Se a acurácia do conjunto de validação em  $e^i$  for maior que o maior valor de acurácia de validação obtido até o momento, considerando os modelos gerados nas épocas anteriores  $\{e^{i-1}, e^{i-2}, \dots, e^2, e^1\}$  então o modelo  $M^i$  é gerado e armazenado. Caso contrário o modelo é descartado e prossegue para a próxima época. O treinamento será interrompido se a acurácia calculada com a base de validação não tiver sido melhorada por 30 épocas.

Ao final do treinamento as imagens de teste poderão ser fornecidas ao modelo gerado, em seguida aplica-se a função *softmax* que resulta na probabilidade dos pixels pertencerem a cada uma das classes. Então a classe com maior valor de probabilidade é atribuída ao pixel como a classe predita.

Ambas as arquiteturas U-Net e PSPNet foram implementadas em *Python* na versão 3.7.0, usando a biblioteca *torch* (*PyTorch*) na versão 0.4.1, que consiste em uma plataforma de DL de código aberto, além da biblioteca *Numpy* na versão 1.5.4. Por razões de eficiência de processamento, foi utilizada uma máquina com GPU. Caso contrário, o treinamento poderia levar dias para ser finalizado. O GPU utilizado foi uma GeForce GTX TITAN X 33MHz.

### 3.6.3 Resultado e Avaliação

As medidas usadas para avaliar o desempenho dos classificadores foram acurácia, precisão, revocação e medida F (descritas na seção 2.5). As medidas de precisão, revocação e medida F são indicadas para avaliar situações envolvendo duas ou mais classes e principalmente, quando existe um desbalanceamento entre as mesmas, pois as medidas são calculadas para cada classe.

Além das medidas, é gerada uma imagem resultante com a predição fornecida para a imagem de teste, destacando as regiões onde a praga foi identificada com sucesso e também as regiões onde eventualmente as regiões de praga foram classificadas erroneamente. Para isso, foi desenvolvido um script em *Python* na versão 3.7.0, no qual obtem-se o vetor (sequência de valores) de predição para a imagem ou amostras de teste e o respectivo rótulo da imagem de teste. Então é usado uma estrutura de repetição para percorrer as linhas e colunas para cada posição:

- ❑ Se o valor predito for **praga** e o valor rotulado for **praga**, então o pixel na imagem resultante é pintado na cor azul; (Verdadeiro-Positivo)
- ❑ Se o valor predito for **praga** e o valor rotulado for **diferente de praga**, então o pixel na imagem resultante é pintado na cor lilás; (Falso-Positivo ou Erro tipo-1);

- Se o valor predito for **diferente de praga** e o valor rotulado for **praga**, então o pixel na imagem resultante é pintado na cor vermelha; (Falso-Negativo ou Erro tipo-2)

Os pixels correspondentes às áreas saudáveis e solo não são destacados na imagem resultante. A Figura 35 ilustra alguns exemplos de segmentos de imagens nas quais as regiões de praga são preditas.

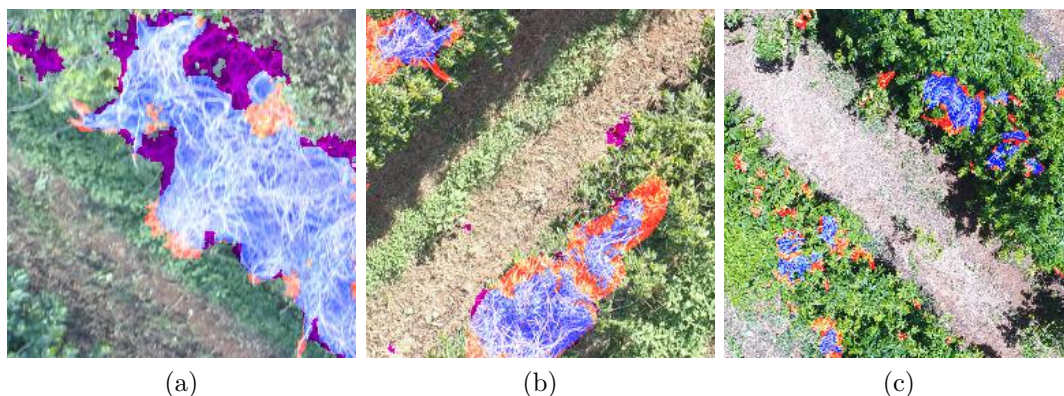


Figura 35 – Exemplos de imagens resultantes após executar a etapa de predição, na qual as regiões em azul representam verdadeiro positivo (classe real e predita iguais a praga), regiões em lilás falso-positivo (classe real não-praga e predita igual a praga) e em vermelho falso-negativo (classe real praga e classe predita não-praga).

## 3.7 Considerações finais

Este capítulo apresentou a descrição de um método para a detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas. O método proposto é composto pelas etapas de coleta de dados, seleção e rotulagem das imagens, pré-processamento, extração de características, treinamento e predição, bem como o resultado e avaliação. Esta metodologia avalia duas estratégias para as etapas de extração de características, treinamento e predição. A primeira abordagem consiste na extração de características baseada em blocos e utilização dos algoritmos de classificação KNN, RF e SVM. A segunda abordagem é baseada no uso das CNNs U-Net e PSPNet que são utilizadas tanto para a etapa de extração de características como para a etapa de treinamento e predição do modelo gerado. No próximo capítulo são apresentados os experimentos para as duas estratégias bem como a avaliação e discussão dos resultados.



---

## Experimentos e Análise dos Resultados

Este capítulo apresenta os experimentos considerando a metodologia proposta para a detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas. O capítulo está organizado da seguinte maneira: a Seção 4.1 descreve o objetivo dos experimentos; a Seção 4.2 detalha os conjuntos de dados considerados para os experimentos; a Seção 4.3 apresenta uma análise dos experimentos. Por fim, a Seção 4.4 apresenta as considerações finais do capítulo.

### 4.1 Objetivos dos Experimentos

Neste trabalho foram realizados três experimentos visando a detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas. Para isso foram usadas duas estratégias, a primeira baseada na extração por blocos e a segunda na extração baseada em CNN. Por último, avalia-se a influência da altura usando a estratégia baseada em CNN.

#### 4.1.1 Experimento 1 - Avaliação da estratégia baseada em blocos

Este experimento consiste em aplicar a estratégia de extração de características baseada em blocos (descrito na subseção 3.4.1) e avaliar qual a configuração mais adequada para o tamanho do bloco interno (*BInt*). Em seguida, utilizando a configuração mais adequada obtida no passo anterior, deseja-se analisar a influência da segmentação do solo, bem como a influência dos atributos extraídos do bloco na tarefa de detecção automática de nematóides. Neste experimento serão empregadas as técnicas de classificação KNN, RF e SVM (descritas na seção 3.5). Ao final, pretende-se compreender a quantidade de atributos mais adequada para geração de um modelo robusto, ou seja, que apresente a menor variação entre os algoritmos avaliados e o algoritmo com melhor desempenho.

### 4.1.2 Experimento 2 - Avaliação da estratégia baseada em CNN

Este experimento consiste em avaliar a tarefa de identificação de nematóides em lavoura cafeeira por meio da estratégia baseada em CNN e comparar duas arquiteturas do estado da arte, sendo elas U-Net e PSPNet (descritas na subseção 3.4.2 para extração baseada em CNN e na seção 3.5 que detalha os algoritmos). Além disso, deseja-se analisar se há ou não algum impacto ao modelo gerado ao fornecer imagens de menor resolução espacial às CNN. No fim, deseja-se compreender qual das arquiteturas é a mais adequada nesta tarefa.

### 4.1.3 Experimento 3 - Avaliação de alturas usando a estratégia baseada em CNN

Este experimento avalia a influência da variação da altura na tarefa de detecção automática de nematóides. Além disso, deseja-se avaliar o quão robusto pode ser um modelo gerado com imagens de menores alturas (ex. 10m) e usado para predição em imagens obtidas de maiores alturas (ex. 15m ou 20m) e vice-versa, ou seja, utilizando modelo gerado com imagens de maiores alturas (ex. 20m) e usado para predição em imagens de menores alturas (ex. 10m ou 15m). Para isso, será utilizada a arquitetura com melhor desempenho, além da configuração de resolução mais adequada para as imagens, descritos no Experimento 2, na subseção 4.3.2). Ao final deste experimento, deseja-se compreender o impacto na variação das alturas na tarefa de detecção automática de nematóides.

## 4.2 Bases de Dados

### 4.2.1 Base *Alfa*

Esta base de dados, chamada de *Alfa* é formada por 10 imagens obtidas da área A (descrita na subseção 3.1.1), nas quais os rótulos gerados são fornecidos como imagens (descrito na subseção 3.2.1) utilizando o critério rígido de rotulagem (descrito na subseção 3.2.3) e identificando duas classes, sendo praga (infestação nematóides) e não-praga (regiões saudáveis e solo).

### 4.2.2 Base *Alfa-S*

Esta base de dados, chamada *Alfa-S* consiste nas imagens da base *Alfa* após aplicar a etapa de segmentação manual para remoção do solo (descrita na subseção 3.3.1). Os rótulos são fornecidos de forma semelhante à base *Alfa*.



### 4.2.3 Base *Beta*

Esta base de dados, chamada de *Beta* é formada por 18 imagens obtidas da área B (descrita na subseção 3.1.1). As imagens desta base são divididas em 3 conjuntos, sendo treinamento, validação e teste, descritos as seguir:

- ❑ **treinamento:** composto por 12 imagens, nas quais os rótulos gerados são fornecidos como anotações (descrito na subseção 3.2.2) usando o critério flexível de rotulagem (descrito na subseção 3.2.3).
- ❑ **validação:** composta por 3 imagens, onde os rótulos são gerados e fornecidos de forma similar ao conjunto de treinamento.
- ❑ **teste:** composta por 3 imagens, onde os rótulos são fornecidos como anotações usando o critério rígido de rotulagem (descrito na subseção 3.2.3).
- ❑ Os motivos para a escolha dos critérios de rotulagem dos conjuntos de treinamento, validação e teste é descrito na subseção 3.6.2.

Para os rótulos gerados, são identificadas 3 classes, sendo regiões com praga, regiões saudáveis e o solo. Em seguida, as imagens e rótulos desta base passam por uma etapa de pré-processamento para redimensionamento e extração das anotações (descrito na subseção 3.3.2).

### 4.2.4 Base *Gama*

Esta base de dados, chamada de *Gama* é formada por 39 imagens, sendo 13 delas selecionadas para cada altura (10m, 15m e 20m) obtidas da área C (descrita na subseção 3.1.1). Para cada altura, as imagens são divididas em 3 conjuntos, sendo treinamento, validação e teste, descritos a seguir.

- ❑ **treinamento:** Composto por 10 imagens nas quais os rótulos gerados são fornecidos como anotações (descrito na subseção 3.2.2) usando o critério flexível de rotulagem (descrito na subseção 3.2.3).
- ❑ **validação:** Formado por 2 imagens, onde os rótulos são gerados e fornecidos de forma similar ao conjunto de treinamento.
- ❑ **teste:** Formado por 1 imagem, onde os rótulos são gerados como anotações usando o critério rígido de rotulagem (descrito na subseção 3.2.3).
- ❑ Os motivos para a escolha dos critérios de rotulagem dos conjuntos de treinamento, validação e teste é descrito na subseção 3.6.2.

Para os rótulos gerados, são identificadas 3 classes, sendo regiões com praga (infestação nematóides), regiões saudáveis (ausência dos nematóides) e o solo. Em seguida, as imagens são pré-processadas de forma semelhante à base *Beta*.

## 4.3 Análise dos Experimentos

### 4.3.1 Experimento 1 - Avaliação da estratégia baseada em blocos

Inicialmente a base de dados *Alfa* é fornecida como entrada para a etapa de extração baseada em blocos (descrita na subseção 3.4.1) sem aplicar nenhum dos algoritmos para redução da dimensionalidade. Nesta etapa, são aplicadas 4 configurações distintas para o tamanho do bloco interno *BInt*, sendo elas ( $10 \times 10$ ,  $20 \times 20$ ,  $30 \times 30$  e  $40 \times 40$ ). O bloco externo *BExt* usado para descrever o contexto, por sua vez, tem o seu tamanho ajustado em função do bloco interno, sendo duas vezes maior que este. Para esta escolha, avaliou-se diferentes tamanhos para o bloco *BExt* e chegou-se a conclusão que essa proporção era a mais adequada neste contexto. A matriz de características resultante do processo de extração é formada por 230 características. Em seguida, os dados são fornecidos para 3 algoritmos, sendo eles KNN, RF e SVM (descritos na subseção 3.5). A Tabela 4, exibida a seguir, apresenta o valor da medida F para a avaliação do desempenho dos algoritmos para cada configuração do tamanho do bloco interno *BInt*, juntamente com o valor da média (M) e o desvio padrão (D) da medida F para as 10 imagens avaliadas (uso de validação cruzada de 10 pastas).

Tabela 4 – Avaliação do desempenho dos algoritmos para cada configuração de *BInt*

I	10×10			20×20			30×30			40×40		
	KNN	RF	SVM	KNN	RF	SVM	KNN	RF	SVM	KNN	RF	SVM
1	0,31	0,57	0,57	0,30	0,57	0,59	0,25	0,56	0,74	0,20	0,57	0,86
2	0,56	0,72	0,68	0,65	0,78	0,74	0,67	0,85	0,86	0,77	0,82	0,93
3	0,02	0,07	0,14	0,01	0,04	0,04	0	0	0	0	0	0
4	0,57	0,70	0,74	0,69	0,65	0,76	0,78	0,44	0,50	0,67	0,50	0,50
5	0,54	0,50	0,55	0,64	0,55	0,75	0,69	0,22	0,39	0,31	0	0,14
6	0,58	0,61	0,64	0,70	0,64	0,77	0,73	0,71	0,60	0,63	0,22	0
7	0,57	0,67	0,74	0,68	0,74	0,85	0,70	0,78	0,76	0,77	0,86	0,83
8	0,60	0,58	0,65	0,65	0,63	0,73	0,65	0,69	0,76	0,72	0,66	0,72
9	0,57	0,64	0,69	0,64	0,77	0,79	0,75	0,80	0,71	0,79	0,83	0,75
10	0,51	0,54	0,66	0,60	0,63	0,79	0,69	0,70	0,71	0,77	0,62	0,63
<b>M</b>	0,48	0,56	0,61	0,56	0,60	<b>0,68</b>	0,59	0,58	0,60	0,56	0,51	0,54
<b>D</b>	<b>0,17</b>	0,18	<b>0,17</b>	0,21	0,20	0,22	0,24	0,26	0,24	0,27	0,31	0,34

Analisando os resultados da Tabela 4, é possível constatar que os algoritmos RF e SVM alcançam o maior valor de média para a medida F, sendo 0,60 e 0,68 respectivamente, utilizando a configuração de  $20 \times 20$  como tamanho para bloco *BInt*. Já o algoritmo KNN exibe o maior valor de média para a medida F, sendo 0,59 para a configuração de  $30 \times 30$

para BInt. O maior valor da média obtido para o algoritmo SVM é destacado em negrito. Em relação ao desvio padrão, é possível perceber uma tendência nos dados, na qual a medida que aumenta-se o tamanho do bloco BInt, aumenta-se o valor do desvio padrão, indicando uma maior variação no desempenho dos algoritmos entre as iterações. O menor valor para o desvio padrão obtido para os algoritmos KNN e SVM é destacado em negrito.

Observando a Figura 36, é possível perceber que para ambas as imagens resultantes, há uma perda significativa de amostras avaliadas considerando os tamanhos de bloco correspondentes à  $30 \times 30$  e  $40 \times 40$  (Figura 36 (c)-(d) e (g)-(h)). Em relação as imagens resultantes considerando o tamanho de bloco  $20 \times 20$  (Figura 36 (b) e (f)), não há uma perda significativa de amostras avaliadas em relação as imagens obtidas com tamanho  $10 \times 10$  (Figura 36 (a) e (e)).

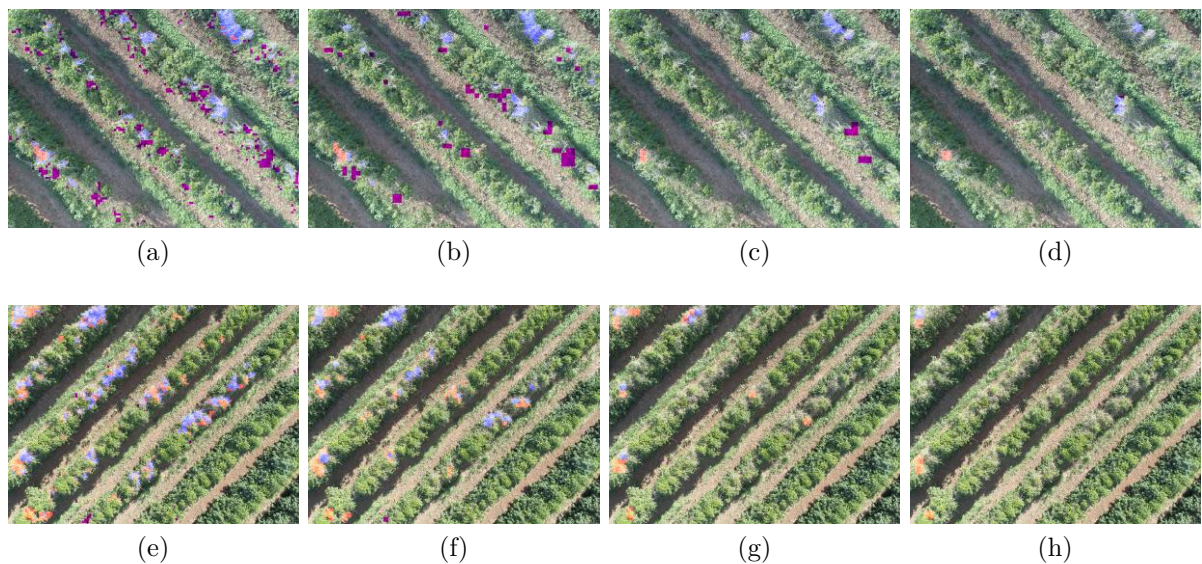


Figura 36 – Imagens resultantes após executar a etapa de predição. (a)-(d) Imagem 1 utilizando SVM para as 4 configurações de *BInt*. (e)-(h) Imagem 4 utilizando RF para as 4 configurações de *BInt*. Em azul verdadeiro positivo (classe real e predita iguais a praga), em lilás falso-positivo (classe real não-praga e predita como praga) e em vermelho falso-negativo (classe real praga e predita não-praga).

Analisando os resultados, temos que as duas melhores configurações para o tamanho do bloco *BInt* são  $10 \times 10$  e  $20 \times 20$ . Neste caso, é necessário obter um equilíbrio entre o melhor desempenho na média obtida para as imagens e uma menor variação entre os resultados das imagens. Os dados da tabela 4, indicam que a média da medida F para os algoritmos KNN, RF e SVM com o tamanho  $20 \times 20$  foram bem superiores aos valores obtidos para o tamanho  $10 \times 10$ . Por outro lado, ambas as configurações de tamanho apresentam valores de desvio padrão próximos. Portanto, o tamanho de bloco interno *BInt* mais adequado é igual a  $20 \times 20$ .

A Tabela 5 exibida a seguir, apresenta o número de amostras ou exemplos por classe

(não-praga e praga), incluindo o percentual de pragas em relação ao total de amostras para cada uma das imagens avaliadas na Tabela 4 com  $BInt$  igual a  $20 \times 20$ . Nesta tabela, exibe-se os valores considerando imagens sem segmentação (sem aplicar nenhum pré-processamento) e segmentadas manualmente.

Tabela 5 – Número de amostras por classe considerando imagens sem segmentação e com segmentação.

I	Sem segmentação			Segmentação		
	Não-praga	Praga	Perc. (%)	Não-praga	Praga	Perc. (%)
1	2318	60	2,5	452	58	11,4
2	2420	180	6,9	793	180	18,5
3	2204	2	0,0	873	1	0,1
4	1276	111	8,0	85	81	48,5
5	1356	155	10,3	103	121	54,0
6	1467	145	9,0	127	110	46,4
7	1956	304	13,5	299	252	45,7
8	2075	417	16,7	268	392	59,4
9	2058	340	14,2	401	274	40,6
10	1656	405	19,7	244	384	61,1
<b>T</b>	1874	2119	10,1	3645	1853	33,7

Analisando as imagens não segmentadas na Tabela 5, é possível perceber um desbalanceamento entre as classes não-praga e praga, ou seja, existem muito mais exemplos de amostras identificadas como não-praga (regiões saudáveis e solo) do que pragas (infestação de nematóides). A imagem 3 por exemplo, o percentual de amostras com praga é menor que 1%. Já a imagem 10 apresenta a maior proporção de exemplos com praga (19,7%). Em relação as 10 imagens, a proporção de amostras com praga é de 10% e os 90% restantes não-praga. Isto permite constatar que este é um problema de natureza desbalanceada, também chamado de *class imbalance*.

Com o intuito de analisar maneiras de equilibrar o número de amostras para as classes não-praga e praga, avalia-se as 10 segmentadas manualmente, ou seja, com a remoção do solo. Neste caso, as amostras da classe não-praga são identificadas somente por regiões saudáveis. A Tabela 5 apresenta o número de amostras por classe considerando as imagens segmentadas. Neste caso, é possível perceber um crescimento na proporção de amostras com praga principalmente para o conjunto de imagens 4,5,6,7,8,9,10. Em relação ao total de amostras para as 10 imagens, verifica-se que a proporção de amostras com praga é de 33,7% e 66,3% não-praga. É possível observar que a proporção de amostras com praga é consideravelmente maior considerando a remoção do solo.

Após encontrar o tamanho mais adequado para o bloco  $BInt$ , deseja-se avaliar 3 conjuntos de características distintos (descritos na subseção 3.4.1), sendo eles o conjunto inicial formado por 230 características (sem redução de dimensionalidade) e outros dois conjuntos de características, obtidos mediante a etapa de redução de dimensionalidade (descrita na subseção 3.4.1.2). Para cada conjunto de características são fornecidas ima-

gens originais (sem aplicar pré-processamento) e segmentadas. A ideia de analisar as imagens segmentadas é compreender qual o valor máximo os algoritmos podem alcançar na tarefa de identificação dos nematóides quando o melhor cenário é usado, ou seja, quando o solo é retirado e as características extraídas são apenas de regiões saudáveis e com praga. Este processo é descrito a seguir.

Inicialmente, as bases *Alfa* e *Alfa-S* que consiste nas imagens segmentadas, são fornecidas como entrada para a extração baseada em blocos com *BInt* igual a  $20 \times 20$  sem aplicar a etapa de redução de dimensionalidade. Para ambas as bases, o conjunto de características inicial é formado por 230 características. Em seguida, os dados são fornecidos para os algoritmos KNN, RF e SVM. As Tabelas 6 e 7 exibidas a seguir, apresentam o valor da precisão (P), revocação (R) e medida F para avaliação do desempenho dos algoritmos para o conjunto com 230 características, considerando as imagens sem segmentação e segmentadas respectivamente. Em ambas as tabelas destaca-se em negrito o maior valor para a medida F e o menor valor de desvio padrão.

Tabela 6 – Resultado KNN, RF e SVM para 230 características

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,20	0,72	0,31	0,45	0,80	0,57	0,45	0,87	0,59
2	0,42	0,84	0,56	0,60	0,91	0,72	0,60	0,96	0,74
3	0,01	0,65	0,02	0,04	0,89	0,07	0,02	1,00	0,04
4	0,58	0,56	0,57	0,96	0,55	0,70	0,86	0,69	0,76
5	0,61	0,49	0,54	0,98	0,33	0,50	1,00	0,60	0,75
6	0,62	0,55	0,58	0,95	0,45	0,61	0,92	0,66	0,77
7	0,63	0,52	0,57	0,86	0,55	0,67	0,93	0,78	0,85
8	0,75	0,50	0,60	0,96	0,42	0,58	0,97	0,59	0,73
9	0,63	0,52	0,57	0,98	0,48	0,64	0,99	0,67	0,79
10	0,75	0,39	0,51	0,98	0,37	0,54	0,95	0,67	0,79
<b>M</b>	0,52	0,57	0,48	0,78	0,58	0,56	0,77	0,75	<b>0,68</b>
<b>D</b>	0,24	0,13	<b>0,18</b>	0,32	0,21	0,19	0,32	0,15	0,23

Tabela 7 – Resultado KNN, RF e SVM para 230 características com segmentação

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,24	0,88	0,37	0,36	0,95	0,52	0,39	1,00	0,56
2	0,43	0,87	0,57	0,57	0,92	0,70	0,60	0,96	0,74
3	0,01	0,88	0,01	0,01	0,88	0,02	0,02	1,00	0,03
4	0,76	0,70	0,73	0,95	0,70	0,81	0,86	0,69	0,76
5	0,81	0,70	0,75	0,97	0,70	0,81	0,93	0,93	0,93
6	0,67	0,72	0,70	0,81	0,74	0,77	0,81	0,88	0,84
7	0,74	0,65	0,69	0,90	0,67	0,77	0,93	0,78	0,85
8	0,77	0,53	0,63	0,94	0,47	0,63	0,98	0,64	0,77
9	0,73	0,62	0,67	0,93	0,61	0,74	0,93	0,82	0,87
10	0,80	0,51	0,62	0,93	0,56	0,70	0,96	0,75	0,84
<b>M</b>	0,60	0,71	0,57	0,74	0,72	0,65	0,74	0,85	<b>0,72</b>
<b>D</b>	0,28	0,14	<b>0,23</b>	0,32	0,16	0,24	0,32	0,13	0,26

Ao analisar a Tabela 6, é possível observar que o algoritmo SVM alcançou o maior valor para a média da medida F, igual a 0,68, destacado na tabela. Por outro lado, o KNN e RF apresentaram uma menor variação na predição das imagens, observando os valores obtidos para o desvio padrão. Tendência igualmente observada na Tabela 7, onde o maior valor de média é atribuída ao SVM e os menores valores para o desvio padrão são obtidos

para o KNN e RF. Além disso, é possível perceber uma melhora razoável na média da medida F para os algoritmos KNN e RF, com um aumento próximo a 10% para ambos, quando comparado com os resultados obtidos sem segmentação, descritos na Tabela 6. Deste modo, considerando este conjunto de característica, aparentemente o impacto da segmentação para o SVM é muito menor quando comparado com o KNN e RF.

Em seguida, deseja-se avaliar os outros dois conjuntos de características obtidos mediante a etapa de redução de dimensionalidade (descrita na subseção 3.4.1.2). O primeiro deles, as bases *Alfa* e *Alfa-S* são fornecidas como entrada para a extração baseada em blocos com *BInt* igual a  $20 \times 20$  e em seguida aplica-se o algoritmo de seleção baseado na correlação de atributos selecionando o conjunto de atributos com valor maior ou igual a um limiar. Neste caso, adota-se o valor de 0,2 para o limiar. Este conjunto é formado por 34 características. Os dados são fornecidos para os algoritmos KNN, RF e SVM. As tabelas exibidas a seguir, apresentam o desempenho dos algoritmos para o conjunto com 34 características, considerando imagens sem segmentação (Tabela 8) e com segmentação (Tabela 9). Em ambas as tabelas destaca-se em negrito o maior valor para a medida F e o menor valor de desvio padrão.

Tabela 8 – Resultado KNN, RF and SVM para 34 características

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,14	0,65	0,23	0,54	0,73	0,62	0,86	0,63	0,73
2	0,44	0,93	0,60	0,63	0,91	0,74	0,75	0,89	0,81
3	0,01	1,00	0,01	0,01	0,50	0,02	0,01	1,00	0,02
4	0,32	0,32	0,32	1,00	0,10	0,18	1,00	0,09	0,17
5	0,54	0,34	0,42	1,00	0,10	0,18	1,00	0,06	0,11
6	0,41	0,30	0,34	0,75	0,04	0,08	1,00	0,08	0,15
7	0,57	0,63	0,59	0,70	0,55	0,62	0,96	0,45	0,62
8	0,87	0,51	0,64	0,97	0,44	0,61	0,98	0,47	0,63
9	0,65	0,57	0,61	0,94	0,54	0,69	0,94	0,43	0,59
10	0,69	0,44	0,54	0,98	0,42	0,59	0,99	0,36	0,53
<b>M</b>	0,46	0,57	0,43	0,75	0,43	0,43	0,85	0,45	<b>0,44</b>
<b>D</b>	0,26	0,24	<b>0,21</b>	0,31	0,28	0,28	0,31	0,33	0,29

Tabela 9 – Resultado KNN, RF and SVM para 34 características com segmentação

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,27	0,95	0,42	0,34	1,00	0,50	0,34	0,98	0,50
2	0,50	0,94	0,65	0,59	0,94	0,73	0,71	0,92	0,80
3	0,00	0,00	0,00	0,00	1,00	0,01	0,00	1,00	0,01
4	0,92	0,56	0,69	0,96	0,32	0,48	1,00	0,56	0,71
5	0,90	0,77	0,83	1,00	0,78	0,87	1,00	0,89	0,94
6	0,67	0,85	0,75	0,84	0,72	0,78	0,78	0,76	0,77
7	0,84	0,78	0,81	0,92	0,78	0,85	0,95	0,83	0,88
8	0,92	0,53	0,67	0,98	0,49	0,65	0,97	0,50	0,66
9	0,79	0,73	0,76	0,94	0,72	0,81	0,94	0,85	0,89
10	0,89	0,68	0,77	0,95	0,67	0,78	0,97	0,74	0,84
<b>M</b>	0,67	0,68	0,64	0,75	0,74	<b>0,65</b>	0,77	0,80	0,70
<b>D</b>	0,32	0,28	0,25	0,34	0,22	0,26	0,34	0,17	0,27

Analisando a Tabela 8, constata-se que nenhum dos 3 algoritmos avaliados se destacou em relação à média da medida F. Comparando os resultados da tabela 8 com a tabela 6, nota-se que para os algoritmos RF e SVM houve uma queda significativa na média

da medida F, além de um aumento considerável no desvio padrão quando utilizado o conjunto com 34 características. Isso indica que provavelmente este conjunto não é o mais adequado para esta tarefa. Por outro lado, o impacto da aplicação da segmentação neste conjunto é maior quando comparado com a configuração com 230 características. Observando a Tabela 9 é possível verificar um aumento significativo na média da medida F, superior a 20% para os 3 algoritmos, além de uma queda no desvio padrão para o RF e SVM quando comparado com os resultados obtidos sem segmentação, descritos na Tabela 8.

No segundo conjunto obtido por meio da etapa de redução de dimensionalidade (descrita na subseção 3.4.1.2), as bases *Alfa* e *Alfa-S* são fornecidas como entrada para a extração baseada em blocos com *BInt* igual a  $20 \times 20$  e em seguida aplica-se o algoritmo de seleção baseado em *wrapper* no qual o próprio algoritmo fornece o melhor conjunto de atributos por meio de um esquema de aprendizado. Este conjunto é formado por 40 características. Então os dados são fornecidos para os algoritmos KNN, RF e SVM. As tabelas exibidas a seguir, apresentam o desempenho dos algoritmos para o conjunto com 40 características, considerando imagens sem segmentação (Tabela 10) e com segmentação (Tabela 11). Em ambas as tabelas destaca-se em negrito o maior valor para a medida F e os menores valores para o desvio padrão.

Tabela 10 – Resultado KNN, RF e SVM para 40 características

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,19	0,82	0,31	0,37	0,78	0,50	0,50	0,83	0,63
2	0,54	0,98	0,70	0,67	0,96	0,79	0,55	0,93	0,69
3	0,00	0,50	0,01	0,03	1,00	0,06	0,03	1,00	0,05
4	0,72	0,82	0,77	0,97	0,60	0,74	0,82	0,63	0,71
5	0,86	0,70	0,77	0,99	0,48	0,65	1,00	0,41	0,58
6	0,72	0,75	0,73	0,97	0,60	0,74	0,99	0,48	0,65
7	0,76	0,74	0,75	0,89	0,69	0,78	0,91	0,67	0,77
8	0,90	0,58	0,71	1,00	0,46	0,63	0,95	0,54	0,69
9	0,77	0,67	0,72	0,97	0,67	0,79	0,97	0,57	0,72
10	0,89	0,61	0,72	0,97	0,56	0,71	0,94	0,57	0,71
<b>M</b>	0,64	0,72	0,62	0,78	0,68	<b>0,64</b>	0,77	0,66	0,62
<b>D</b>	0,31	0,14	0,25	0,33	0,18	0,22	0,32	0,20	<b>0,21</b>

Tabela 11 – Resultado KNN, RF e SVM para 40 características com segmentação.

I	KNN			RF			SVM		
	P	R	F	P	R	F	P	R	F
1	0,28	1,00	0,43	0,34	1,00	0,50	0,37	1,00	0,55
2	0,59	0,97	0,73	0,64	0,97	0,77	0,67	0,96	0,79
3	0,00	1,00	0,01	0,01	1,00	0,01	0,01	1,00	0,02
4	0,95	0,86	0,90	0,95	0,75	0,84	0,97	0,75	0,85
5	0,95	0,87	0,91	0,98	0,93	0,96	0,93	0,93	0,93
6	0,68	0,88	0,77	0,81	0,87	0,84	0,86	0,83	0,84
7	0,88	0,81	0,84	0,94	0,85	0,89	0,95	0,86	0,90
8	0,94	0,66	0,78	0,99	0,58	0,73	0,97	0,61	0,75
9	0,84	0,72	0,77	0,98	0,75	0,85	0,93	0,79	0,85
10	0,92	0,69	0,79	0,97	0,76	0,85	0,95	0,76	0,84
<b>M</b>	0,70	0,85	0,69	0,76	0,85	0,72	0,76	0,85	<b>0,73</b>
<b>D</b>	0,33	0,13	<b>0,27</b>	0,34	0,14	0,28	0,33	0,13	<b>0,27</b>

Ao analisar a Tabela 10, é possível perceber um certo equilíbrio na média da medida

F obtida para os 3 algoritmos. Comparando os resultados desta tabela com os valores obtidos para o conjunto com 230 características sem segmentação (Tabela 6) nota-se um aumento razoável na média da medida F para os algoritmos KNN e RF. Para o algoritmo SVM há uma queda na média da medida F. Isso indica que provavelmente exista um número considerável de características irrelevantes no conjunto com 230 atributos. Em relação ao desvio padrão, há um aumento para os algoritmos KNN e RF e uma queda considerando o SVM. Ao verificar os resultados desta tabela com os valores obtidos para o conjunto com 34 características sem segmentação (Tabela 8) é possível constatar um aumento considerável na média da medida F, igual ou superior a 18% para todos os algoritmos, além de um menor desvio padrão para os algoritmos RF e SVM.

Analisando a Tabela 11, resultante da avaliação dos algoritmos para 40 características com segmentação, verifica-se que há um aumento razoável na média da medida F para os 3 algoritmos comparando com os dados obtidos na Tabela 10.

De um modo geral, há um pequeno aumento na média da medida F para os algoritmos ao aplicar a segmentação para os 3 conjuntos de características analisados. O maior valor médio da medida F para os algoritmos KNN, RF e SVM, são obtidos para a configuração com 40 características (Tabela 11), alcançando valores próximos à 70%. Desta maneira, é possível constatar que o impacto do solo na tarefa de detecção automática de nematóides é baixa, uma vez que mesmo removendo as regiões de solo, o erro persiste em patamares próximos à 30%, indicando que a maior parte do erro obtido na identificação de regiões com nematóides é causado pela confusão com as regiões saudáveis.

O próximo passo, consiste em selecionar o conjunto de características (sem segmentação) mais adequado para a tarefa de detecção automática de nematóides, considerando desempenho e robustez, ou seja, aquele que apresente os melhores resultados para os algoritmos avaliados, considerando a média para a medida F, além de obter a menor variação entre os valores obtidos. Neste caso, o conjunto com 40 características (Tabela 10) é selecionado.

Ao analisar os resultados das imagens para os algoritmos KNN, RF e SVM com 40 características (Tabela 10), verifica-se que o algoritmo RF alcançou a maior média para a medida F entre os 3 algoritmos, com valor de 0,64.

Dentre as imagens, é possível perceber que para a imagem 3, os resultados obtidos para ambos os algoritmos ficaram bem abaixo da média, alcançando patamares abaixo dos 10% para a medida F. Ao examinar esta imagem, verificou-se uma grande dificuldade para o especialista identificar as regiões com a presença dos nematóides. Um dos fatores que pode ter influenciado é que na área coletada haviam pontos onde existiam um maior desnível do solo. A Figura 37 ilustrada pelos itens (a), (b) e (c) apresenta a predição para o KNN, RF e SVM.

Também é possível observar que o maior valor obtido para a medida F é encontrado para a imagem 9 com a predição fornecida pelo RF com valor de 0,79. A Figura 37



ilustrada pelos itens (d), (e) e (f) apresenta a predição para o KNN, RF e SVM, respectivamente. Uma característica a ser observada na predição para os 3 algoritmos é que a média da precisão para o KNN (0,64) é relativamente menor a média alcançada para o RF (0,78) e SVM (0,77). Para ambas as imagens 3 e 9 descritas anteriormente, é possível perceber a maior ocorrência de falsos positivos, representados pela cor roxa na predição fornecida pelo KNN (Figuras 37 (a) e (d)).

Desta forma, conclui-se que o algoritmo RF é o mais adequado para a configuração com 40 características, pois o algoritmo apresentou o maior valor para a média da medida F, precisão e revocação e um dos menores valores encontrados para o desvio padrão.

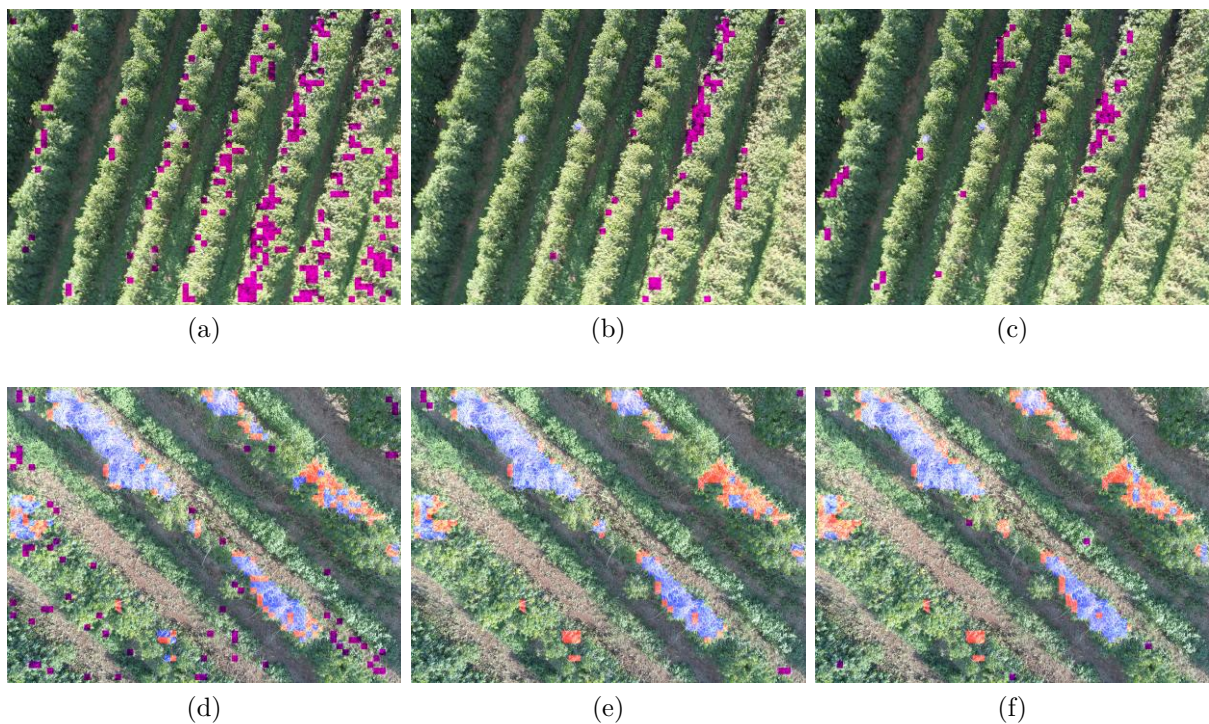


Figura 37 – Resultado da predição pela comparação entre KNN, RF e SVM para 40 características sem segmentação (Tabela 10). (a) Resultado imagem 3 para KNN. (b) Resultado imagem 3 para RF. (c) Resultado imagem 3 para SVM. (d) Resultado imagem 9 para KNN. (e) Resultado imagem 9 para RF. (f) Resultado imagem 3 para SVM.

### 4.3.2 Experimento 2 - Avaliação da estratégia baseada em CNN

Inicialmente, utiliza-se a estratégia de extração de características baseada em CNN (descrita na subseção 3.4.2) para as arquiteturas U-Net e PSPNet, utilizando redes pré-treinadas para ambas com o intuito de inicializar os pesos da rede. Então, os conjuntos de treinamento e validação gerados para a base *Beta* são fornecidos como entrada para as redes U-Net e PSPNet e em seguida, inicia-se o método para treinamento (descrito na subseção 3.6.2) e avalia-se as imagens do conjunto de teste para predição das áreas

identificadas como nematóides (descrito na seção 3.6.3). A Tabela 12 descrita a seguir, apresenta uma comparação das arquiteturas U-Net e PSPNet, na qual são usadas imagens em 3 resoluções diferentes ( $640 \times 480$ ,  $960 \times 640$  e  $1280 \times 960$ ).

Tabela 12 – Comparação das arquiteturas U-Net e PSPNet usando imagens com diferentes resoluções.

Img		640×480		960×640		1280×960	
		U-Net	PSPNet	U-Net	PSPNet	U-Net	PSPNet
1	P	0,67	0,70	0,68	0,57	0,72	0,58
	R	0,79	0,58	0,80	0,69	0,75	0,78
	F	<b>0,73</b>	0,63	<b>0,73</b>	0,63	<b>0,73</b>	0,66
2	P	0,61	0,41	0,54	0,38	0,58	0,45
	R	0,91	0,93	0,92	0,93	0,88	0,90
	F	<b>0,73</b>	0,57	<b>0,68</b>	0,54	<b>0,70</b>	0,60
3	P	0,70	0,15	0,73	0,34	0,67	0,95
	R	0,54	0,85	0,54	0,67	0,59	0,16
	F	<b>0,61</b>	0,27	<b>0,62</b>	0,45	<b>0,63</b>	0,28

Analisando a Tabela 12, é possível perceber que a U-Net obteve o melhor desempenho na avaliação das imagens de teste, considerando a medida F para as 3 resoluções analisadas. Os resultados são destacados em negrito. Observando o desempenho da U-Net e PSPNet para cada uma das 3 resoluções, percebe-se uma variação mínima nos resultados obtidos para as imagens, indicando que ambas são robustas a variações espaciais das imagens fornecidas. Para as análises subsequentes avalia-se os modelos para a U-Net e PSPNet obtidos com resolução de  $640 \times 480$ .

Ao observar a Tabela 12, verifica-se que as maiores discrepâncias nos valores obtidos para a U-Net e PSPNet podem ser percebidas na avaliação das imagens 3 e 2 respectivamente. Em ambos os casos, é possível observar uma diferença significativa na avaliação da precisão (P). Considerando a avaliação da imagem 3, a precisão alcançada para a U-Net é de 0,70 enquanto que para a PSPNet é 0,15. Já para a imagem 2, a precisão para a U-Net é 0,61 e para a PSPNet é 0,41.

Os resultados para as imagens 2 e 3 são ilustrados na Figura 38, para os itens (e), (f), (h) e (i), nas quais é possível perceber uma maior ocorrência de falsos positivos, representados pela cor roxa nas Figuras 38 (f) e (i) resultante da predição da PSPNet em comparação com a predição fornecida pela U-Net (Figuras 38 (e) e (h)). As regiões em azul destacam as regiões de praga identificadas corretamente.

Em relação a imagem 1, é possível verificar que a U-Net e PSPNet alcançaram valores próximos de precisão (P), porém houve uma diferença maior na avaliação da revocação (R), sendo 0,79 para a U-Net e 0,58 para a PSPNet. A Figura 38, considerando os itens (b) e (c) ilustra os resultados para a imagem 1, na qual é possível perceber maiores ocorrências de falsos negativos, representados na cor vermelha na Figura 38 (c) resultante



da predição da PSPNet em comparação com a predição fornecida pela U-Net (Figura 38 (b)).

Desta maneira, é possível concluir que a U-Net apresenta um desempenho superior comparado a PSPNet, para a tarefa de identificação de nematóides.

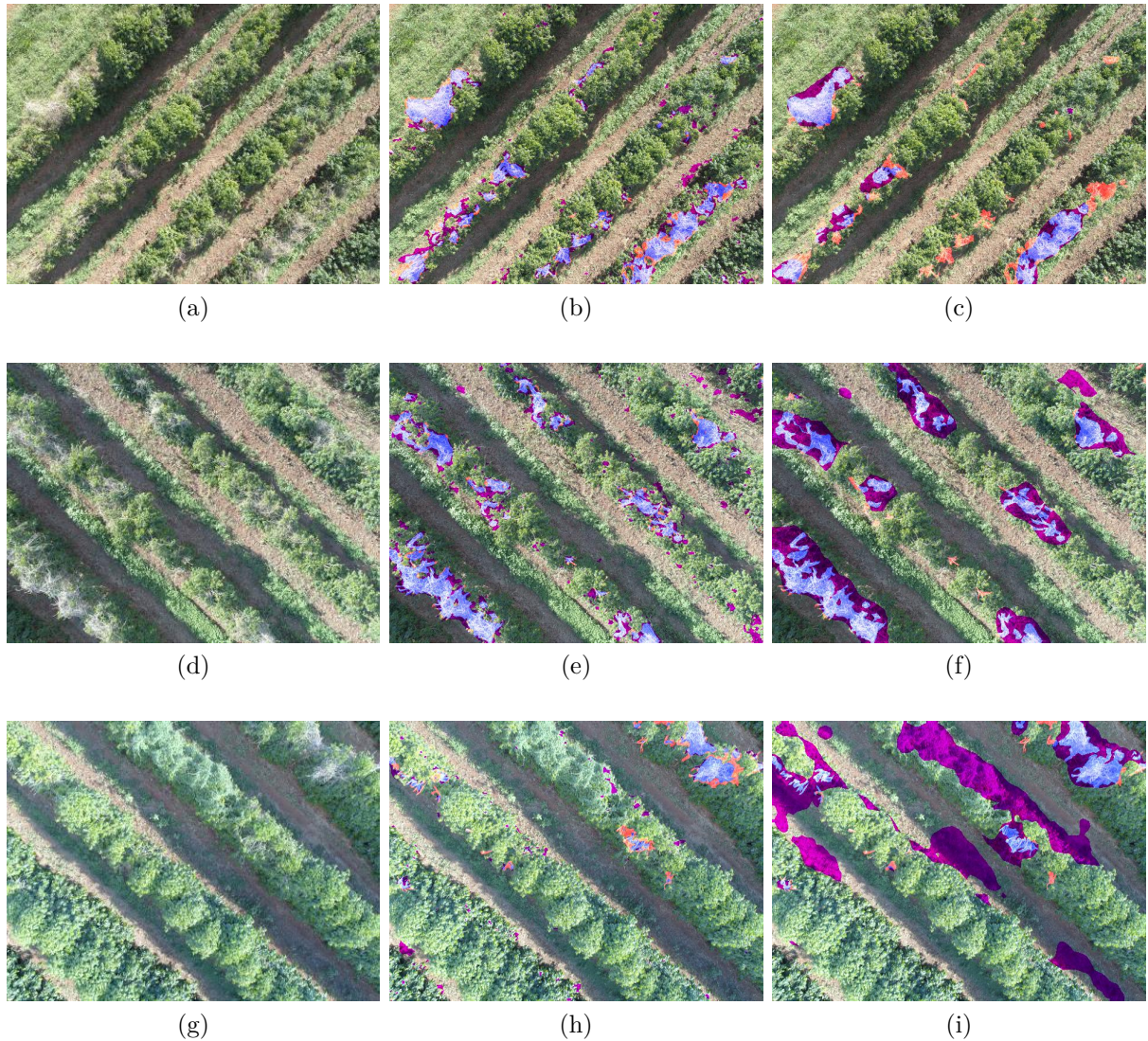


Figura 38 – Resultado da predição para as imagens obtidas pela comparação da U-Net e PSPNet. (a) Imagem 1. (b) Resultado imagem 1 para U-Net. (c) Resultado imagem 1 para PSPNet. (d) Imagem 2. (e) Resultado imagem 2 para U-Net. (f) Resultado imagem 2 para PSPNet. (g) Imagem 3. (h) Resultado imagem 3 para U-Net e (i) Resultado imagem 3 para PSPNet

### 4.3.3 Experimento 3 - Avaliação de diferentes alturas usando a estratégia baseada em CNN

Uma das principais motivações para este experimento, está no fato de que durante os vôos com o VANT, observou-se que dada uma área a ser coletada, sobrevoá-la à alturas

mais baixas demanda uma carga maior de bateria do VANT. Por exemplo, dada a área C (descrita na subseção 3.1.1) e considerando que com a bateria totalmente carregada, permite uma autonomia de 30 minutos de bateria ao VANT. Ao traçar a rota para a área coletada de forma automática, sobrevoar a 10m é estimado para este voo 20 minutos (correspondendo a 66% de bateria) enquanto que sobrevoar a mesma área a 20m é estimado para o voo 10 minutos (correspondendo a 33% de bateria). Desta maneira, em termos de consumo de bateria, é mais vantajoso sobrevoar em alturas maiores. No entanto, uma importante questão é saber se o desempenho preditivo dos classificadores é afetado à medida que coleta imagens em maiores alturas. A seguir é detalhada uma análise utilizando 3 alturas diferentes para avaliação.

Para este experimento, utiliza-se a arquitetura U-Net e as imagens são fornecidas com tamanho de  $640 \times 480$ , sendo o algoritmo e a configuração com melhor desempenho obtidos no experimento 2 (descrito na subseção 4.3.2). Nesta avaliação, é utilizada a base *Gama*, onde para cada altura (10m, 15m e 20m), são obtidos os conjuntos de treinamento e validação correspondentes para geração do modelo. Nesse caso, gera-se 3 modelos, o primeiro com imagens coletadas a 10m, o segundo com imagens coletadas a 15m e o terceiro com imagens coletadas a 20m. Em seguida, para cada um dos modelos gerados é fornecido uma imagem de teste para cada uma das alturas coletadas e realiza-se a predição e avaliação das imagens.

Na Tabela 13 exibida a seguir, as colunas *Train-Val* apresentam os modelos gerados de acordo com o treinamento e validação, considerando cada uma das alturas (10m, 15m e 20m), e as linhas apresentam a predição para cada imagem fornecida a 10m, 15m e 20m para os modelos. Para avaliação é usada as medidas precisão (P), revocação (R) e medida F (F). Nesta tabela destaca-se os maiores valores da medida F para cada uma das alturas avaliadas na predição.

Tabela 13 – Avaliação dos modelos para diferentes alturas

Predição	<i>Train-Val-10m</i>			<i>Train-Val-15m</i>			<i>Train-Val-20m</i>		
	P	R	F	P	R	F	P	R	F
10m	0,43	0,77	<b>0,55</b>	0,48	0,45	0,47	0,31	0,63	0,42
15m	0,31	0,58	0,41	0,38	0,47	<b>0,42</b>	0,25	0,72	0,37
20m	0,40	0,40	0,40	0,51	0,34	0,41	0,33	0,58	<b>0,42</b>

A Tabela 13 mostra uma comparação de diferentes modelos utilizando a arquitetura U-Net em alturas distintas para a predição de regiões de nematóides. É possível observar que o modelo que apresentou o melhor desempenho, considerando o maior valor de medida F é o modelo gerado com imagens de 10m (*Train-Val-10m*) e avaliando imagens desta mesma altura, com valor de 0,55. No entanto, este modelo com 10 metros, quando distanciado de 15 metros e 20 metros, tende a diminuir sua eficácia (diminui os valores da medida F).

Considerando o modelo gerado para 20m (*Train-Val-20m*) apesar deste modelo não apresentar os maiores valores de predição para as alturas avaliadas, é possível perceber

uma menor variação entre as 3 alturas avaliadas, ou seja, uma constância para a medida F. Desta maneira, este modelo aparentemente apresenta uma tendência de maior robustez, pois não se degrada tanto, ou seja, pequena variação na medida F.

Por outro lado, ao comparar o resultado deste modelo (*Train-Val-20m*) na avaliação de uma imagem na mesma altura, temos que o resultado obtido foi de 0,42, que é muito inferior aos resultados obtidos com essa mesma altura no experimento 2 (descrito na subseção 4.3.2).

Investigando mais a fundo as bases *Beta*, usada no experimento 2, e *Gama*, usada neste experimento, tem-se que a base *Beta* foi coletada no mês de Maio que foi um período bastante seco, no qual a precipitação registrada foi de 27mm (representando 22,43% da média anual de precipitação registrada em 2018) (DATE.ORG, 2019). Essas condições, facilitaram a identificação das regiões de nematóides. Já a base *Gama* foi coletada no mês de Dezembro, que por outro lado foi o mês mais chuvoso de 2018 para a região de Monte Carmelo-MG, registrando 277mm de precipitação (DATE.ORG, 2019). Essas condições, em contrapartida, dificultam ainda mais a identificação das regiões de praga. Desta maneira, esse é um dos fatores que podem explicar a diferença na predição considerando as imagens das bases *Beta* e *Gama* usando a U-Net.

A Figura 39 ilustra uma comparação entre os modelos gerados a 10 e 20 metros. A primeira coluna da Figura 39, ilustrada pelos itens (a), (d) e (g), exibe as imagens originais fornecidas para predição com 10m, 15m e 20m respectivamente; a segunda coluna, itens (b), (e) e (h), apresenta a predição fornecida pelo modelo gerado com 10m; a terceira coluna, itens (c), (f) e (i), apresenta a predição fornecida pelo modelo gerado com 20m.

## 4.4 Considerações finais

Neste capítulo são apresentados três experimentos, o primeiro utilizando a estratégia de extração baseada em blocos em conjunto com os algoritmos de classificação KNN, RF e SVM usando validação cruzada de 10 pastas. O segundo usando a estratégia baseada em CNN com as redes U-Net e PSPNet para extração de características e avaliação das imagens de testes. Ambos os experimentos demonstram ser efetivos na tarefa de detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas.

Além disso, utilizando a estratégia baseada em CNN realizou-se um experimento com o objetivo de avaliar a influência da variação da altura na identificação dos nematóides. Os resultados não são conclusivos, porém é possível perceber uma tendência que indica que modelos gerados com imagens de maior altitudes podem ser mais robustos na detecção dos nematóides.



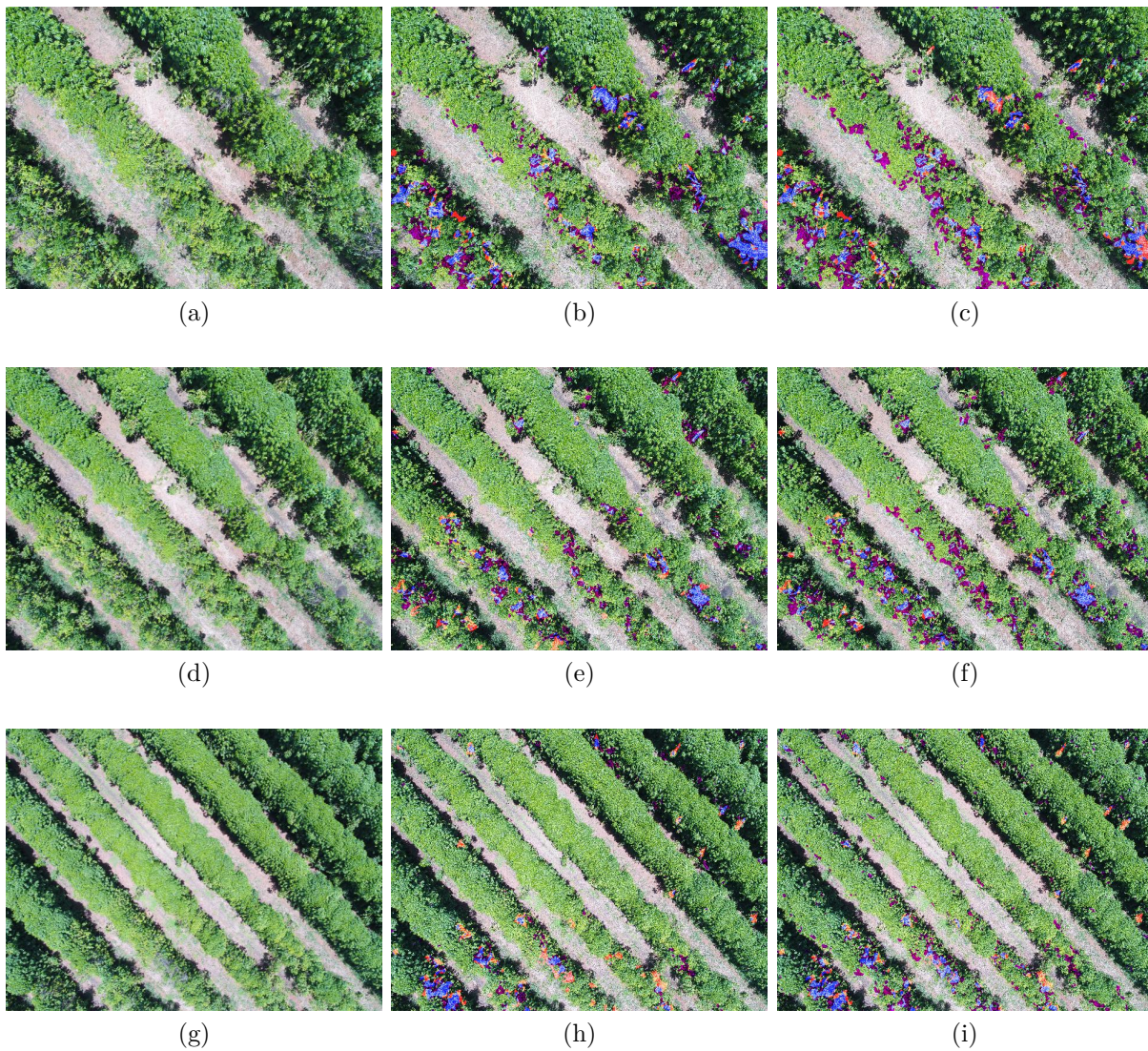


Figura 39 – Resultado da predição entre os modelos gerados com 10 e 20 metros. itens (a), (d) e (g) exibe as imagens originais fornecidas para predição com 10m, 15m e 20m respectivamente. Itens (b), (e) e (h) apresenta a predição fornecida pelo modelo gerado com 10m. Itens (c), (f) e (i) a predição fornecida pelo modelo gerado com 20m

---

## Conclusão

Esta dissertação propõe o desenvolvimento de um método para detecção automática de nematóides em lavouras cafeeiras usando imagens aéreas por meio de VANT. A metodologia proposta avalia duas estratégias para a extração de características, sendo a primeira baseada em blocos e a segunda baseada em CNN. O método propõem o uso de diferentes classificadores e o uso de medidas de avaliação para avaliar o desempenho.

A primeira estratégia utiliza a extração de característica baseada em blocos para obter as informações que permitam discriminar as regiões entre praga (nematóides) e não-praga. Para isso, utilizando um conjunto inicial com 230 características, avalia-se 4 configurações distintas para o bloco interno (*BInt*). Durante os experimentos constatou-se que o tamanho adequado para o bloco é igual a  $20 \times 20$ . Em seguida, avalia-se dois métodos de redução de dimensionalidade utilizando imagens sem segmentação e com segmentação. Verificou-se que o método de seleção baseado em *wrapper* que fornece 40 características foi o que apresentou o melhor valor para a medida F e menor desvio padrão entre os algoritmos avaliados KNN, RF e SVM com *kernel* polinomial. Dos algoritmos avaliados o RF apresentou a melhor média para a medida F considerando as imagens sem segmentação com 0,64.

A segunda estratégia utiliza a extração baseada em CNN, onde são utilizadas duas arquiteturas do estado da arte, sendo elas U-Net e PSPNet, ambas pré-treinadas com a base de dados ImageNet. Após inicializar os pesos iniciais da rede, os dados de treinamento e validação são fornecidos para ambas as redes U-Net e PSPNet e o treinamento é realizado para geração do modelo de predição. Constatou-se que a rede U-Net apresentou os melhores resultados, com valor médio de aproximadamente 70% para as imagens avaliadas. Também avaliou-se a influência na variação da altura utilizando esta estratégia. Verificou-se que o modelo gerado a partir das imagens coletadas a 10m, obteve o maior valor para a medida F, igual a 0,55. No entanto, o modelo gerado a partir de imagens coletadas a 20m, aparenta ser o mais robusto considerando a menor variação na predição para menores alturas.

## 5.1 Principais Contribuições

A seguir são apresentadas as principais contribuições alcançadas no desenvolvimento do trabalho:

- ❑ Desenvolvimento de um método automático para detecção de pragas, o qual pode ser aplicado em outros problemas envolvendo imagens aéreas;
- ❑ Criação de uma base de dados de imagens aéreas obtidas em lavoura cafeeira, juntamente com os respectivos rótulos, disponível em <<https://github.com/ufuairlab/nematode-detection-labels>>, com o intuito de tornar a pesquisa reprodutível;
- ❑ Estudo comparativo da influência do uso de diferentes altitudes na detecção dos nematóides.

## 5.2 Trabalhos Futuros

Os resultados obtidos no trabalho descrito aqui mostraram o bom desempenho do método e motivam novas investigações para trabalhos futuros. Algumas novas possíveis linhas de investigação são listadas a seguir:

- ❑ Em relação a estratégia baseada em blocos, é possível perceber que há um desbalanceamento entre amostras de praga e não-praga. Nos trabalhos futuros deseja-se avaliar métodos de subamostragem dos dados, bem como técnicas como abordagem de “detecção de novidades” (ou “detecção de anomalias”) para identificação das áreas infestadas;
- ❑ Em relação a avaliação da alturas é necessário avaliar um número maior de alturas, com maior variação entre elas para corroborar os resultados, que nos experimentos realizados neste trabalho, indicaram que modelos treinados em maiores altitudes são mais robustos;
- ❑ Deseja-se utilizar uma CNN pré-treinada para extração das características, e fornecer essas informações para outros classificadores, como RF e SVM, com o intuito de avaliar este conjunto de características;
- ❑ Avaliar e comparar outras arquiteturas de CNN do estado da arte, como a *SegNet* e *TernausNet*.



## 5.3 Contribuições em Produção Bibliográfica

A pesquisa realizada durante esta dissertação produziu a publicação de um artigo e a submissão de outro em conferência. A seguir são apresentados detalhes de dois artigos e dos respectivos meios de publicação:

- Artigo intitulado “Analysis of nematodes in coffee crops at different altitudes using aerial images” aceito na conferência “European Signal Processing Conference” (EU-SIPCO) em 2019. Neste artigo foram apresentados os resultados 2 e 3 descritos no capítulo de resultados.
- Artigo intitulado “Segmenting and detecting nematode in coffee crops using aerial images” submetido para a conferência “International Conference on Computer Vision Systems” (ICVS). Este artigo apresenta a estratégia baseada em blocos para identificação de nematóides em lavoura cafeeira usando imagens aéreas.



---

## Referências

- AKESSON, N. B.; YATES, W. E. **The use of aircraft in agriculture**. [S.l.]: Food & Agriculture Org., 1974.
- BACKES, A. **PGC204 - Reconhecimento de Padrões**, howpublished = <<http://www.facom.ufu.br/~backes/pgc204/Aula03-ClassificadoresElementares.pdf>>, note = Accessed: 2019-05-08. 2018.
- BARBEDO, J. G. A. An automatic method to detect and measure leaf disease symptoms using digital image processing. **Plant Disease**, Am Phytopath Society, v. 98, n. 12, p. 1709–1716, 2014. Disponível em: <<https://doi.org/10.1094/PDIS-03-14-0290-RE>>.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: SPRINGER. **European conference on computer vision**. 2006. p. 404–417. Disponível em: <[https://doi.org/10.1007/11744023\\_32](https://doi.org/10.1007/11744023_32)>.
- BOISSEAU, M. et al. Resistance to meloidogyne paranaensis in wild coffea arabica. **Tropical Plant Pathology**, v. 34, p. 38–41, 2009.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>.
- CALDERÓN, R. et al. High-resolution airborne hyperspectral and thermal imagery for early detection of verticillium wilt of olive using fluorescence, temperature and narrow-band spectral indices. **Remote Sensing of Environment**, Elsevier, v. 139, p. 231–245, 2013. Disponível em: <<https://doi.org/10.1016/j.rse.2013.07.031>>.
- CAMPOS, V. P.; VILLAIN, L. **Plant parasitic nematodes in subtropical and tropical agriculture**. [S.l.]: Wallingford: CAB International, 2005.
- CANNY, J. A computational approach to edge detection. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 8, n. 6, p. 679–698, jun. 1986. ISSN 0162-8828. Disponível em: <<https://doi.org/10.1109/TPAMI.1986.4767851>>.
- CARNEIRO, R. M. D. G. et al. Identification and genetic diversity of meloidogyne spp. (tylenchida: Meloidogynidae) on coffee from brazil, central america and hawaii. **Nematology**, v. 6, p. 287–298, 2004. Disponível em: <<https://doi.org/10.1163/1568541041217942>>.

CECAFE. **Relatório das exportações brasileiras de café. Conselho dos exportadores de café do Brasil**. 2018. Disponível em: <<http://www.cecafe.com.br>>.

CHAVES, R. L. **Roger Lopez-Chaves, Universidad de Costa Rica, Bugwood.org**. 2019. Disponível em: <<https://www.ipmimages.org/browse/detail.cfm?imgnum=1356089>>.

CHEN, J. et al. Automatic segmentation and counting of aphid nymphs on leaves using convolutional neural networks. **Agronomy**, Multidisciplinary Digital Publishing Institute, v. 8, n. 8, p. 129, 2018. Disponível em: <<https://doi.org/10.3390/agronomy8080129>>.

CHEN, L. et al. Semantic image seg. deep conv. nets and fully connected. **CVPR**, 2014.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995. Disponível em: <<https://doi.org/10.1023/A:1022627411411>>.

DATE.ORG climate. **Temperaturas e precipitações médias, howpublished = <<https://pt.climate-data.org/america-do-sul/brasil/minas-gerais/monte-carmelo-24982/>>, note = Accessed: 2019-06-17**. 2019.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DERWINSUHARTONO, W. A.; LESTARI, M.; YASIN, M. Expert system in detecting coffee plant diseases. **Int. J. Electr. Energy**, v. 1, n. 3, p. 156–162, 2013. Disponível em: <<https://doi.org/10.12720/ijoe.1.3.156-162>>.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. **arXiv preprint arXiv:1603.07285**, 2016.

EREMENKO, K. **Deep Learning A-Z™: Hands-On Artificial Neural Networks.**, howpublished = <<https://www.udemy.com/deeplearning/learn/v4/overview>>. 2018.

FAOSTAT. **Food and Agriculture Organization of the United Nations**. 2018. Disponível em: <<http://faostat.fao.org>>.

FERGUSON, M. et al. Automatic localization of casting defects with convolutional neural networks. In: IEEE. **2017 IEEE International Conference on Big Data (Big Data)**. 2017. p. 1726–1735. Disponível em: <<https://doi.org/10.1109/BigData.2017.8258115>>.

FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. **Neural networks**, Elsevier, v. 1, n. 2, p. 119–130, 1988. Disponível em: <[https://doi.org/10.1016/0893-6080\(88\)90014-7](https://doi.org/10.1016/0893-6080(88)90014-7)>.

GAERTNER, J. et al. Vegetation classification of coffea on hawaii island using worldview-2 satellite imagery. **Journal of Applied Remote Sensing**, International Society for Optics and Photonics, v. 11, n. 4, p. 046005, 2017.

GOMES, O. d. F. M. Processamento e análise de imagens aplicados à caracterização automática de materiais. **Rio de Janeiro**, 2001.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing (3rd Edition)**. [S.l.]: Prentice Hall, 2010.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

HALL, M. et al. The WEKA data mining software: an update. **SIGKDD Explorations**, v. 11, n. 1, p. 10–18, 2009. Disponível em: <<https://doi.org/10.1145/1656274.1656278>>.

HALL, M. A. Correlation-based feature selection of discrete and numeric class machine learning. University of Waikato, Department of Computer Science, 2000.

HE, K. et al. Deep residual learning for image recognition. **Computer Vision and Pattern Recognition**, 2016.

HICKE, J. A.; LOGAN, J. Mapping whitebark pine mortality caused by a mountain pine beetle outbreak with high spatial resolution satellite imagery. **International Journal of Remote Sensing**, Taylor & Francis, v. 30, n. 17, p. 4427–4441, 2009. Disponível em: <<https://doi.org/10.1080/01431160802566439>>.

HUANG, Y.; LAN, Y.; HOFFMANN, W. Use of airborne multi-spectral imagery in pest management systems. **Agricultural Engineering International: CIGR Journal**, 2008.

ICO. **Total Production Crop Year, International Coffee Organization**. 2018. Disponível em: <<https://goo.gl/Gn6A6z>>.

IGLESIAS, J. C. Á. Uma metodologia para caracterização de sínter de minério de ferro: Microscopia digital e análise de imagens. **PUC, Rio de Janeiro**, 2008.

JAIN, A.; NANDAKUMAR, K.; ROSS, A. Score normalization in multimodal biometric systems. **Pattern recognition**, Elsevier, v. 38, n. 12, p. 2270–2285, 2005. Disponível em: <<https://doi.org/10.1016/j.patcog.2005.01.012>>.

JENKINS, W. R. A rapid centrifugal-flotation technique for separating nematodes from soil. **Plant Disease Report**, v. 48, 1964.

JOALLAND, S. et al. Aerial and ground based sensing of tolerance to beet cyst nematode in sugar beet. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 10, n. 5, p. 787, 2018. Disponível em: <<https://doi.org/10.3390/rs10050787>>.

KARPATHY, A. et al. Cs231n convolutional neural networks for visual recognition. **Neural networks**, v. 1, 2016.

KHIRADE, S. D.; PATIL, A. Plant disease detection using image processing. In: IEEE. **Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on**. 2015. p. 768–771. Disponível em: <<https://doi.org/10.1109/ICCUBEA.2015.153>>.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. **Artificial intelligence**, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Disponível em: <[https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)>.

- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. **Ijcai**. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.
- KUSUMO, B. S. et al. Machine learning-based for automatic detection of corn-plant diseases using image processing. In: IEEE. **2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)**. [S.l.], 2018. p. 93–97.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998. Disponível em: <<https://doi.org/10.1109/5.726791>>.
- LEWIS, D. D.; GALE, W. A. A sequential algorithm for training text classifiers. In: **Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Springer-Verlag New York, Inc., 1994. (SIGIR '94), p. 3–12. ISBN 0-387-19889-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=188490.188495>>.
- LOWE, D. G. **Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image**. [S.l.]: Google Patents, 2004. US Patent 6,711,293.
- MA, J. et al. A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing. **Computers and Electronics in Agriculture**, Elsevier, v. 142, p. 110–117, 2017. Disponível em: <<https://doi.org/10.1016/j.compag.2017.08.023>>.
- MAHARLOOEI, M. et al. Detection of soybean aphids in a greenhouse using an image processing technique. **Computers and electronics in agriculture**, Elsevier, v. 132, p. 63–70, 2017. Disponível em: <<https://doi.org/10.1016/j.compag.2016.11.019>>.
- MAPA. **Café no Brasil**. 2017. Disponível em: <<http://www.agricultura.gov.br/assuntos/politica-agricola/cafe/cafeicultura-brasileira>>.
- MARSLAND, S. Machine learning, an algorithmic perspective, chapman & hall/crc machine learning & pattern recognition. **CRC, Boca Raton, Fla**, 2009.
- MITCHELL, T. M. Machine learning (mcgraw-hill international editions computer science series). McGraw-Hill, 1997.
- MOUNTRAKIS, G.; IM, J.; OGOLE, C. Support vector machines in remote sensing: A review. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 66, n. 3, p. 247–259, 2011. Disponível em: <<https://doi.org/10.1016/j.isprsjprs.2010.11.001>>.
- NARDARI, G. V. et al. Crop anomaly identification with color filters and convolutional neural networks. In: IEEE. **2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)**. [S.l.], 2018. p. 363–369.
- NOVO, E. Sensoriamento remoto—princípios e aplicações, são paulo, 2. **Edigard Bliicher Ltda**, 1989.

- OLIVEIRA, E. F. d. Estudo da granulometria em imagens via a modelagem das frequências espaciais. 2007.
- OTOBONI, C. Metodologia de demarcação de reboleiras para o manejo localizado de nematóides. In: **Congresso Brasileiro de Agricultura de Precisão-ConBAP. São Pedro-SP**. [S.l.: s.n.], 2014.
- OTSU, N. A threshold selection method from gray-level histograms. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 9, n. 1, p. 62–66, 1979. Disponível em: <<https://doi.org/10.1109/TSMC.1979.4310076>>.
- PANG, J. et al. Automatic segmentation of crop leaf spot disease images by integrating local threshold and seeded region growing. In: IEEE. **Image Analysis and Signal Processing (IASP), 2011 International Conference on**. [S.l.], 2011. p. 590–594.
- PETHYBRIDGE, S. J.; NELSON, S. C. Leaf doctor: A new portable application for quantifying plant disease severity. **Plant Disease**, Am Phytopath Society, v. 99, n. 10, p. 1310–1316, 2015. Disponível em: <<https://doi.org/10.1094/PDIS-03-15-0319-RE>>.
- PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. **arXiv preprint arXiv:1806.07908**, 2018.
- PRÖVE, P.-L. **An Introduction to different Types of Convolutions in Deep Learning**, howpublished = <<https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>>. 2017.
- RAZZAK, M. I.; NAZ, S.; ZAIB, A. Deep learning for medical image processing: Overview, challenges and the future. In: **Classification in BioApps**. Springer, 2018. p. 323–350. Disponível em: <[https://doi.org/10.1007/978-3-319-65981-7\\_12](https://doi.org/10.1007/978-3-319-65981-7_12)>.
- REED, B. C. et al. Measuring phenological variability from satellite imagery. **Journal of vegetation science**, Wiley Online Library, v. 5, n. 5, p. 703–714, 1994. Disponível em: <<https://doi.org/10.2307/3235884>>.
- REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [S.l.]: Editora Manole Ltda, 2003.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical image computing and computer-assisted intervention**. 2015. p. 234–241. Disponível em: <[https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)>.
- RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: CITESEER. **ICCV**. [S.l.], 2011. v. 11, n. 1, p. 2.
- RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015. Disponível em: <<https://doi.org/10.1007/s11263-015-0816-y>>.
- SAFFARI, A. et al. On-line random forests. In: **12th International Conference on Computer Vision Workshops**. [s.n.], 2009. p. 1393–1400. Disponível em: <<https://doi.org/10.1109/ICCVW.2009.5457447>>.

- SALGADO, S.; CARNEIRO, R.; PINHO, R. d. Aspectos técnicos dos nematoides parasitas do cafeeiro. **EPAMIG, Belo Horizonte**, 2011.
- SALGADO, S. M. L.; REZENDE, J. C.; NUNES, J. A. R. Selection of coffee progenies for resistance to nematode *meloidogyne paranaensis* in infested area. **Crop Breeding and Applied Biotechnology**, v. 14, p. 94–101, 2014. Disponível em: <<https://doi.org/10.1590/1984-70332014v14n2a17>>.
- SEAPAMG. **Certifica Minas Café**. 2017. Disponível em: <<http://www.agricultura.mg.gov.br/institucional/15-uncategorised/12-certifica-minas-cafe892>>.
- SHAKHATREH, H. et al. Unmanned aerial vehicles: A survey on civil applications and key research challenges. **arXiv preprint arXiv:1805.00881**, 2018.
- SHAKHNAROVICH, G.; DARRELL, T.; INDYK, P. **Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing)**. The MIT Press, 2006. Disponível em: <<https://doi.org/10.7551/mitpress/4908.001.0001>>.
- SHINDE, S. R. et al. Experiments on content based image classification using color feature extraction. In: IEEE. **2015 international conference on communication, Information & Computing Technology (ICCICT)**. 2015. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICCICT.2015.7045737>>.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SOMEREN, M. v.; WIDMER, G. (Ed.). **ECML '97: Proceedings of the 9th European Conference on Machine Learning**. London, UK, UK: Springer-Verlag, 1997. ISBN 3-540-62858-4.
- SOUZA, J. R. et al. Automatic detection of ceratocystis wilt in eucalyptus crops from aerial images. In: IEEE. **2015 IEEE international conference on robotics and automation (ICRA)**. [S.l.], 2015. p. 3443–3448.
- STARBUCKS, C. **Growing the future of coffee**. 2016. Disponível em: <<https://stories.starbucks.com/stories/2016/growing-the-future-of-coffee-one-tree-at-a-time-a/>>.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining, (First Edition)**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367.
- TEACH, T. L. **Agricultural aircraft control system using the global positioning system**. [S.l.]: Google Patents, 1994. US Patent 5,334,987.
- TETILA, E. C. et al. Identification of soybean foliar diseases using unmanned aerial vehicle images. **IEEE Geoscience and Remote Sensing Letters**, IEEE, v. 14, n. 12, p. 2190–2194, 2017. Disponível em: <<https://doi.org/10.1109/LGRS.2017.2743715>>.
- TOMASI, C. Histograms of oriented gradients. **Computer Vision Sampler**, p. 1–6, 2012.
- TRAINA, A.; JR, C. Similarity search in multimedia databases. In: \_\_\_\_\_. [S.l.: s.n.], 2003. p. 711–738.



VASSALLO-BARCO, M. et al. Automatic detection of nutritional deficiencies in coffee tree leaves through shape and texture descriptors. **Journal of Digital Information Management**, v. 15, n. 1, 2017.

WADA, K. **labelme: Image Polygonal Annotation with Python**. 2016.  
<<https://github.com/wkentaro/labelme>>.

ZHAO, H. et al. Pyramid scene parsing network. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 2881–2890.

ZHOU, B. et al. Semantic understanding of scenes through the ade20k dataset. **International Journal of Computer Vision**, v. 127, n. 3, p. 302–321, Mar 2019. ISSN 1573-1405. Disponível em: <<https://doi.org/10.1007/s11263-018-1140-0>>.