

DISSERTAÇÃO DE MESTRADO

**IDENTIFICAÇÃO AUTOMÁTICA DOS ESTÁGIOS DE VIDA DA
Bemisia tabaci spp. POR VISÃO COMPUTACIONAL
E APRENDIZAGEM DE MÁQUINA**

GUILHERME ALBUQUERQUE BARBOSA SILVA

Brasília, Dezembro de 2019

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**IDENTIFICAÇÃO AUTOMÁTICA DOS ESTÁGIOS DE VIDA DA
Bemisia tabaci spp. POR VISÃO COMPUTACIONAL
E APRENDIZAGEM DE MÁQUINA**

GUILHERME ALBUQUERQUE BARBOSA SILVA

*Dissertação submetida ao Departamento de Engenharia Mecânica
da Faculdade de Tecnologia da Universidade de Brasília como requisito parcial
para obtenção do grau de Mestre em Sistemas Mecatrônicos.*

Banca Examinadora

Prof. Dr. Díbio Leandro Borges, ENM/UnB

Orientador

Prof. Dr. José Maurício S. T. da Motta, ENM/UnB

Examinador interno

Prof. Dr. Geraldo P. R. Filho, CIC/UnB

Examinador externo

FICHA CATALOGRÁFICA

SILVA, GUILHERME ALBUQUERQUE BARBOSA
IDENTIFICAÇÃO AUTOMÁTICA DOS ESTÁGIOS DE VIDA DA *Bemisia tabaci spp.* POR
VISÃO COMPUTACIONAL E APRENDIZAGEM DE MÁQUINA

[Distrito Federal] 2019.

xii, 49., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2019).

Dissertação de Mestrado - Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- | | |
|--------------------------|------------------------------|
| 1. Mosca branca | 2. Aprendizagem de Máquina |
| 3. <i>Random Forests</i> | 4. Classificação de ínstares |
| I. ENM/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

SILVA, G. A. B. (2019). IDENTIFICAÇÃO AUTOMÁTICA DOS ESTÁGIOS DE VIDA DA *Bemisia tabaci spp.* POR VISÃO COMPUTACIONAL E APRENDIZAGEM DE MÁQUINA, Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-71A/14, Departamento de Engenharia Mecânica, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 49.

CESSÃO DE DIREITOS

AUTOR: GUILHERME ALBUQUERQUE BARBOSA SILVA

TÍTULO: IDENTIFICAÇÃO AUTOMÁTICA DOS ESTÁGIOS DE VIDA DA *Bemisia tabaci spp.* POR VISÃO COMPUTACIONAL E APRENDIZAGEM DE MÁQUINA.

GRAU: Mestre ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse trabalho de conclusão de curso pode ser reproduzida sem autorização por escrito do autor.

GUILHERME ALBUQUERQUE BARBOSA
SILVA

Quadra 204 lote 05

Águas Claras

CEP 71939-540 - Brasília - DF - Brasil

Dedicatória

Dedico este trabalho para todos os trabalhadores do campo, incansáveis, que nos alimentam todos os dias, trabalhando de sol a sol

GUILHERME ALBUQUERQUE BARBOSA SILVA

Agradecimentos

Agradeço primeiramente a Deus, pelas oportunidades e direcionamento. Muitas pessoas durante esta pesquisa contribuíram significativamente para sua conclusão. Sem essas pessoas nada que está escrito aqui teria chegado a lugar nenhum. Em primeiro lugar agradeço ao meu pai Rogério, pelo seu companheirismo e dedicação e pela força que me deu nesta jornada. A minha mãe Carla, por poder proporcionar todo o amor, conforto e segurança que sempre tive. Ao meu outro pai, Ricardo pela compreensão e ajuda. Não posso me esquecer também, da minha irmã Danielle, que sempre me incentivou e acreditou no meu potencial mesmo quando eu já tinha duvidado. Quero agradecer também aos profissionais que me apoiaram imensamente neste trabalho. Em primeiro lugar ao meu orientador professor Dívio, pela sua imensa paciência, disponibilidade para orientar e olhar criterioso para com o trabalho. Ao Dr. Edson Hirose, que contribuiu com o tema do trabalho e com os materiais utilizados, sempre solícito e disponível para ajudar. Ao professor Eugênio pelo aconselhamento e incentivo. Sou eternamente grato por ter tido a oportunidade de vivenciar junto destas pessoas, que me ensinaram que apesar de existirem muitos problemas a serem resolvidos, há pessoas igualmente dispostas e interessadas a resolvê-los.

GUILHERME ALBUQUERQUE BARBOSA SILVA

RESUMO

Pragas agrícolas têm sofrido um aumento de incidência nas últimas décadas. Pesticidas como a maior forma de controle de pragas tem declinado em eficiência, na medida que a resistência das pragas aumenta. Métodos de controle mais naturais tem sido testados com grande sucesso em controlar estas populações de pragas. Tais métodos se beneficiam e necessitam de informações sobre essas pragas. Neste trabalho é apresentada uma metodologia de reconhecimento e contagem da mosca branca (*Bemisia tabaci spp*) em 6 diferentes estágios de seu ciclo de vida usando modelo de Aprendizagem de Máquina, Random Forests e regras morfológicamente derivadas. Fornecidas pela Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) 240 imagens foram utilizadas, obtidas com câmera de espectro comum. As imagens foram rotuladas e cada uma comparada com as regras entomológicas para a contagem e classificação de cada objeto: exúvia, ínstar 1 ao ínstar 4 e a mosca adulta. O sistema foi treinado em 100 imagens e performou a mesma tarefa de classificação nas 140 imagens restantes previamente não vistas pelo sistema e os resultados comparados. Os resultados mostraram acurácia de 95% de classificação por pixel e 86% para a classificação dos objetos. Sendo essa metodologia compatível com métodos similares.

Palavras Chaves: Mosca branca; *Random Forests*; Aprendizagem de máquina; Classificação de ínstares

ABSTRACT

Agricultural pests have been increasing in the past few decades. Pesticides as the major control method is declining in efficiency as pest resistance is increasing. More natural methods have been tested with great success in controlling pest population, such methods require and benefits with more information about the pests. Our work is to present a methodology for the recognition and counting of the silverleaf whitefly (*Bemisia tabaci spp*) in 6 different stages of its life cycle using the Machine Learning model, Random Forests and morphological derived rules. Given by Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), 240 images were used, taken with a consumer entry level camera. The images were labeled and each one compared to entomological rules to count and classify each object: *exuviae*, instar 1 to 4 and adult flies. The system was trained in 100 images and performed the same classification task in the remaining unseen 140 images and the results compared. The results showed accuracy of 95% for pixel classification and 86% for object classification. Being this methodology comparable with similar methods.

Keywords: Whitefly; Random Forests; Machine Learning; Instars classification

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	APRENDIZAGEM DE MÁQUINA (<i>Machine Learning</i>)	2
1.3	A MOSCA BRANCA (<i>Bemisia tabaci spp.</i>)	2
1.4	OBJETIVOS DA PESQUISA	5
1.4.1	HIPÓTESES DA PESQUISA	6
2	Referencial Teórico	7
2.1	TREINAMENTO SUPERVISIONADO	7
2.2	TREINAMENTO NÃO SUPERVISIONADO	8
2.3	TREINAMENTO SEMI-SUPERVISIONADO	8
2.4	<i>Naive BAYES</i>	8
2.5	FUNÇÃO DE BASE RADIAL, <i>Radial Basis Function</i> (RBF)	9
2.6	ÁRVORES DE DECISÃO	9
2.7	FLORESTAS RANDÔMICAS (<i>Random Forests</i>)	10
2.8	VISÃO COMPUTACIONAL	12
2.9	TRANSFORMADA CIRCULAR DE HOUGH	12
2.10	NORMALIZAÇÃO DE IMAGEM	13
2.11	REVISÃO DA LITERATURA	14
3	Metodologia	17
3.1	SELEÇÃO DE ATRIBUTOS E TREINAMENTO	19
3.2	BALANCEAMENTO E ALGORITMO SMOTE (CHAWLA ET AL., 2002)	22
3.3	EXTRAÇÃO DE OBJETOS E FILTRAGEM DE RUÍDOS	23
4	Resultados	29
5	Considerações finais	34
	REFERÊNCIAS BIBLIOGRÁFICAS	36
	Anexos	39
I	Anexo 1	40

LISTA DE FIGURAS

1.1	Exemplo de uma mosca branca numa folha contendo outros ínstares	3
1.2	Exemplo do fungo fumagina	4
1.3	Gráfico Dias x tamanho (mm) da mosca branca. Adaptado de (BAIG et al., 2015)..	5
1.4	Exemplo dos objetos nas seis classes avaliadas (Esta imagem é para fins de ilustração e por isso os objetos não possuem tamanhos proporcionais)	5
2.1	Exemplo de uma árvore de decisão a partir de sintomas	10
2.2	Esquema de ilustração do algoritmo de florestas randômicas	11
3.1	Exemplo de uma imagem fonte, marcada, que produz o vetor da Tabela 3.1.	18
3.2	Fluxograma da metodologia proposta.	19
3.3	Imagem exemplificando cada saída de cada etapa da metodologia. S é a imagem fonte, O1 - O5 são as saídas das etapas marcadas na Figura 3.3. As imagens individuais com mais detalhes podem ser visualizadas no Anexo 1.	20
3.4	Imagens resultantes dos algoritmos testados no pré-modelo (30 imagens).	21
3.5	A é a imagem original e A1 a saída do sistema: 1 exúvia, 1 primeiro ínstar, 2 segundo ínstar, 3 terceiros ínstar e 4 quarto ínstar.	27
3.6	B é a imagem original e B1 a saída do sistema: 6 exúvias.	27
3.7	C é a imagem original e C1 é a saída do sistema: 4 exúvias, 1 primeiro ínstar e 2 segundo ínstar.	27
3.8	D é a imagem original e D1 é a saída do sistema: 3 exúvias, 3 primeiro ínstar, 1 terceiro ínstar, 2 quarto ínstars.	28
3.9	E é a imagem original e E1 é a saída do sistema: 17 exúvias, 1 primeira ínstar, 3 segundo ínstars, 3 quarto ínstars and 1 adulta	28
3.10	F é a imagem original e F1 é saída do sistema: 8 exúvias, 9 primeiro ínstar, 5 segundo ínstar, 10 quarto ínstar e 4 adultas.	28
4.1	Curva ROC do modelo nos dados utilizados (verdadeiros positivos x falso positivos)	31
I.1	Imagem fonte (<i>Source</i>).	40
I.2	Imagem fonte após o uso da transformada circular de Hough (<i>Output 1</i>).....	41
I.3	Imagem fonte após o preenchimento humanamente supervisionado (<i>Output 2</i>).	42
I.4	Imagem fonte após o preenchimento automático (<i>Output 3</i>).	43
I.5	Imagem anterior após a filtragem de ruído (<i>Output 4</i>).....	44
I.6	Imagem anterior após a classificação dos objetos (<i>Output 5</i>).....	45

LISTA DE TABELAS

3.1	Exemplo de vetor que representa os achados na Figura 3.1	17
3.2	Vetor de exemplo da amostra retirada de cada pixel	20
3.3	Resultados dos testes no pré-modelo (30 imagens) no algoritmos testados	22
3.4	Regras de RGB e tamanhos para classificação dos objetos	26
4.1	Validação cruzada nas 140 imagens de validação.	32
4.2	Resultados por classe dos 2360 objetos encontrados nas 140 imagens de validação ...	33
4.3	Matriz de confusão dos objetos encontrados	33

LISTA DE SÍMBOLOS E ABREVIACÕES

Abreviações

MIP	Manejo integrado de pragas
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
<i>ROC</i>	<i>Receiver Operating Characteristics</i>
<i>SMOTE</i>	<i>Synthetic Minority Over-sampling Technique</i>
Px	pixel
Pos X	Posição ao longo do eixo das abcissas
Pos Y	Posição ao longo do eixo das ordenadas

Capítulo 1

Introdução

1.1 Contextualização

Atualmente, a demanda por comida tem crescido juntamente com o crescimento da população. Com o auxílio da tecnologia, a agricultura mundial em larga escala tem proporcionado um avanço significativo na produção de alimentos, principalmente, nos últimos 50 anos. Tecnologias de plantio e controle de pragas estão entre os maiores responsáveis por estes avanços (SCOTT; MAXWELL, 2018). Com o passar dos anos o combate a pragas utilizando apenas controles químicos tem perdido sua eficiência de modo rápido em plantações, o que ameaça seriamente a segurança alimentar (SPARKS; NAUEN, 2015).

A resistência biológica contra pesticidas tem aumentado a cada ano, especialmente para insetos. Considerando que, a adaptação natural e seleção genética nestas populações ocorrem, a despeito e em ritmo, que nossos controles químicos não possam acompanhar, é imperativo que outros métodos de controle sejam usados para o manejo biológico dessas populações.

Quanto mais informação se tem sobre uma determinada população de pragas, mais eficientemente um método de controle pode ser empregado. Técnicas de controle de pragas precisas dependem de informação precisa, completa e de alto nível. Um exemplo de um desses métodos que tem sido aplicado com grande sucesso em diferentes partes do mundo é o Manejo Integrado de Pragas (MIP). Esse método consiste na identificação de cada praga e a escolha informada e apropriada de um método adequado (biológico, químico, manual etc.). Com o objetivo de não erradicar, mas de manter esta população de pragas em um nível que não prejudique a produção de alimentos ou a sobrevivência dessa população no meio ambiente (EHLER; LESTER, 2006). A dinâmica populacional, assim como a maneira como esta se distribui, podem mudar a maneira mais apropriada de controlá-la. Conhecimento a respeito da estrutura da fase de vida de uma população de pragas é crítico para implementação de estratégias de controle que possam focar especificamente em cada fase (CHI, 1990).

Segundo (SAVARY et al., 2019), em todo mundo, é estimado que as pragas agrícolas sejam responsáveis pela perda de 21,5% da produção de trigo, 30,0% de arroz, 22,5% de milho e 21,4% do grão da soja. No caso dos custos de tratamento contra a mosca branca, por exemplo, em

platações de melão, o custo com defensivos agrícolas contra esta praga é de USD\$ 136,50 por hectare e com um potencial de perda de até 75% da produção (COSTA et al, 2019). A correta contagem e identificação dessas pragas são de grande interesse para a comunidade agrícola mundial e ainda é um problema em aberto na literatura científica.

Observando-se este cenário a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) em 2016 realizou uma competição (*hackaton*) direcionada ao público universitário com o título: "Manejo Integrado de Pragas para a sustentabilidade dos agroecossistemas". Esse evento teve como objetivo discutir ideias e soluções dentro dessa temática. O autor da presente pesquisa participou dessa competição obtendo o terceiro lugar. Durante o evento, foram realizadas discussões técnicas a cerca do tema, o que originou o contato com um dos entomologistas que participaram, Dr. Edson Hirose. O mesmo destacou a importância do tema e a necessidade da pesquisa utilizando métodos computacionais para reconhecimento e classificação de pragas agrícolas. Posteriormente o mesmo viria a proporcionar os dados (imagens) usados neste trabalho. Dentre as diversas metodologias na computação, com o foco em reconhecimento e classificação, a que mais vem sendo utilizada nesse contexto atualmente é Aprendizagem de Máquina.

1.2 Aprendizagem de Máquina (*Machine Learning*)

A Aprendizagem de Máquina é um campo de estudo da Inteligência Artificial, que é uma das áreas da Ciência da Computação. Este campo de estudo, tem como foco os algoritmos e modelos estatísticos com o objetivo de programar uma máquina a realizar tarefas que não foram explicitamente programadas. (BISHOP, C.M. 2006). Aprendizagem de Máquina é usada em diversas tarefas cujo objetivo, geralmente, são: classificação, clusterização (agrupamento), regressão e séries temporais. A Aprendizagem de Máquina funciona através da construção de um modelo estatístico/matemático utilizando dados amostrais que são compostos de atributos que caracterizam e separam essas amostras. Estes dados são chamados de dados de treinamento. O maior contra ponto desta abordagem é que modelos baseados em Aprendizagem de Máquina, geralmente, requerem grande quantidades de dados para serem úteis e poder computacional para serem implementados. Graças aos avanços no desenvolvimento de *hardware*, tal poder computacional, hoje pode ser facilmente obtido. Como os problemas tratados neste trabalho vem de uma natureza rica em dados (fotografias reais) esses contrapontos não são relevantes para escolha desta abordagem.

A partir das discussões ocorridas com a EMBRAPA representada na pessoa do Dr. Edson, foi informado que dentro das pragas que mais afetam a produtividade agrícola do Brasil e do mundo, a mosca branca tem sido o foco do esforço não somente da EMBRAPA mas de diversas entidades de pesquisa agrícola ao redor do mundo.

1.3 A mosca branca (*Bemisia tabaci spp.*)

A mosca branca é um pequeno inseto medindo em torno de 1 mm que se alimenta da seiva nutritiva das folhas em que pousa (DE BARRO et al., 2015). A mosca branca é uma praga comum

a todos os tipos de cultura em todas as partes do mundo. A mosca adulta e outras fases de vida em uma folha podem ser visualizados na Figura 1.1.



Figura 1.1: Exemplo de uma mosca branca numa folha contendo outros ínstares

Essa praga leva a três principais problemas nas plantações. O primeiro, a retirada da seiva da folha leva a uma subnutrição da planta que acaba por se desenvolver abaixo do esperado, produzindo produtos que não são viáveis para o comércio. Este problema em especial afeta principalmente as produções de soja e de algodão. O segundo problema é que a mosca branca pode ser um vetor de transmissão de um conjunto de doenças, que geralmente afetam com mais rigor as plantações de tomate e grãos em geral (MORALES; ANDERSON, 2011). O terceiro problema são os excrementos da mosca branca. Estes excrementos propiciam o aparecimento de fungos, principalmente o conhecido como fumagina (Figura 1.2). Este fungo, que possui uma coloração escura, atrapalha o processo de fotossíntese da folha e conseqüentemente sua obtenção de nutrientes (HOROWITZ; ISHAAYA, 2016).

Do ciclo de vida da mosca branca, as seis principais classes identificadas e classificadas neste trabalho são: exúvia, ínstares 1 a 4 e a mosca adulta. As seguintes características físicas observadas são baseadas em observação empírica e descrições do Dr Edson Hirose, entomologista que ajudou neste trabalho.

A Figura 1.3 ilustra as fases de vida da mosca branca consideradas. A exúvia é o resto que permaneceu do nascimento de uma adulta. É predominantemente branca e de forma oval. O primeiro ínstar são larvas muito jovens, muito pequenas e com formato de gota. Neste trabalho



Figura 1.2: Exemplo do fungo fumagina

foram identificados a fase dos ovos e o primeiro estágio larval como “ínstar 1”, como ambos são muito pequenos. O segundo ínstar tem um tamanho maior que o anterior, possui menos pixels claros e a cor verde menos intensa. O terceiro ínstar continua com o crescimento de tamanho (o maior salto de tamanho ocorre entre o segundo e o terceiro ínstars) a coloração começa a ficar mais amarelada e os olhos são mais proeminentes. O quarto ínstar possui olhos mais notáveis cores verde e amarelo mais intensos e uma forma mais rígida, que não se espalha tão facilmente como os outros ínstars. A mosca branca adulta possui uma cabeça amarelada com a predominância total de pixels claros e uma forma comprida. Na Figura 1.3 é possível visualizar o crescimento (largura e comprimento) da mosca branca e a divisão por ínstars baseado nessas características de seu ciclo de vida em relação ao tempo.

Crescimento da mosca branca por dias avaliados

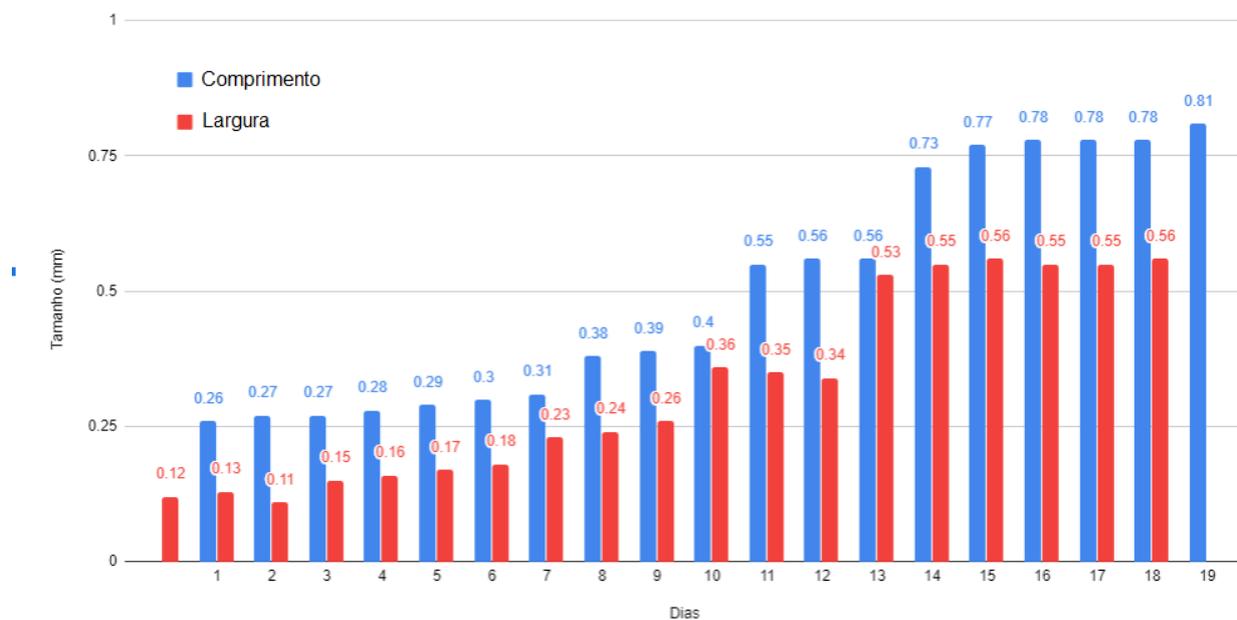


Figura 1.3: Gráfico Dias x tamanho (mm) da mosca branca. Adaptado de (BAIG et al., 2015).

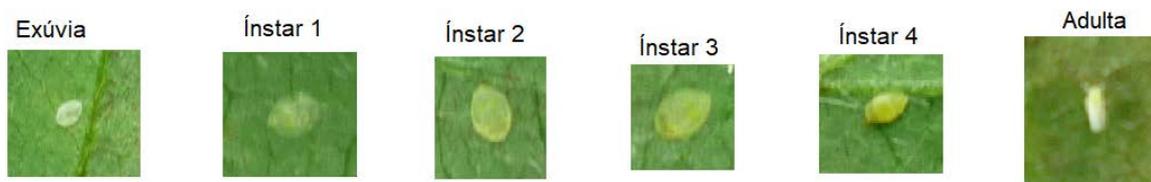


Figura 1.4: Exemplo dos objetos nas seis classes avaliadas (Esta imagem é para fins de ilustração e por isso os objetos não possuem tamanhos proporcionais)

1.4 Objetivos da pesquisa

Tendo em vista que o foco predominante na literatura científica atual é do reconhecimento apenas da fase adulta da mosca branca, ou apenas o reconhecimento de ninfas de alfédeos em geral, este trabalho levanta o seguinte questionamento: **é possível identificar e classificar os ínstars da mosca branca (*Bemisia tabaci* spp.) utilizando Aprendizagem de Máquina e Visão Computacional?** O tratamento de qualquer praga é significativamente enriquecido se a informação sobre sua dinâmica populacional está disponível e a depender dessa dinâmica, a escolha do tratamento mais eficaz pode ser completamente alterado. Desse modo, o objetivo deste trabalho é **propor uma metodologia para identificar e contar a mosca branca (*Bemisia tabaci* spp.) em suas diferentes fases de vida de forma automática.**

1.4.1 Hipóteses da pesquisa

Para responder ao problema levantado pela pesquisa, as seguintes hipóteses serão testadas:

1. Por meio da Visão Computacional e Aprendizagem de Máquina é possível identificar a mosca branca.
2. Por meio da Visão Computacional e Aprendizagem de Máquina é possível classificar as fases de vida da mosca branca (ínstares 1 a 4 e exúvias).

A metodologia proposta se divide em duas principais fases. A primeira fase é a identificação de pixels que possivelmente possam formar objetos de interesse relacionados a mosca branca, onde o teste de *hold-out* é utilizado para testar a hipótese 1. A segunda fase é a classificação dos grupos de pixels já encontrados nas 6 classes alvo desta pesquisa, sendo realizado um teste de validação cruzada (*cross-validation 10-fold*) para testar a hipótese 2.

Para o alcançar o objetivo geral o seguintes objetivos específicos foram traçados:

- Discernir computacionalmente em uma imagem, ínstares da mosca branca do restante da imagem.
- Mensurar a acurácia do algoritmo de Aprendizagem de Máquina utilizando o teste de *hold-out*.
- Diminuir o erro proveniente do modelo de Aprendizagem de Máquina utilizando procedimentos de filtragem.
- Destacar os objetos procurados e rotulá-los nas seis classes procuradas.
- Mesurar o nível de acerto dos objetos classificados utilizando o teste de validação cruzada (*cross-validation 10-fold*).

Capítulo 2

Referencial Teórico

Neste capítulo são apresentados os fundamentos teóricos usados para a construção da metodologia proposta. Inicialmente são apresentados os algoritmos e técnicas que foram comparados para a implementação da metodologia proposta neste trabalho proveniente do campo de estudo de Aprendizagem de Máquina. Em seguida, os conceitos de Visão Computacional utilizados são detalhados.

Dentro da área de Aprendizagem de Máquina existem diversas metodologias usadas com o objetivo de classificação. Existem os modelos baseados em redes neurais, os modelos baseados em redes profundas, modelos baseados em árvores e os modelos baseados em probabilidade. Todos esses modelos utilizam-se de um ou mais tipos de treinamento, que são recorrentes na área de Aprendizagem de Máquina, os principais tipo de treinamento são descritos a seguir.

2.1 Treinamento supervisionado

A abordagem de treinamento supervisionado requer que o conjunto de treinamento esteja corretamente e previamente rotulado, isto é, cada amostra deve possuir a informação a que classe ou categoria esta pertence. Utilizando-se esses dados já classificados inicialmente, separa-se parte deste conjunto de treinamento para a construção de um conjunto menor, geralmente chamado de conjunto de teste. Neste conjunto de teste o atributo que representa a classe a qual aquela amostra pertence é escondido do modelo, de forma que o modelo sendo treinado no conjunto de treinamento consiga preencher a informação faltante (geralmente a que classe cada amostra pertence) do conjunto de teste. Quanto do conjunto de teste o modelo conseguiu corretamente preencher a informação faltante é a acurácia do modelo, e este teste é chamado de "hold-out".

2.2 Treinamento não supervisionado

Na abordagem de treinamento não supervisionado, os dados não estão previamente rotulados. Dessa maneira, não há informação de qual classe ou categoria cada amostra pertence. Através de algoritmos de agrupamento (clusterização) como K-means (FORGY, 1965) por exemplo, é possível agrupar essas amostras afim de descobrir similaridades nos atributos para artificialmente agrupá-las e posteriormente rotulá-las. Assim, é possível a utilização do treinamento supervisionado posteriormente.

2.3 Treinamento semi-supervisionado

Na abordagem de treinamento semi-supervisionado os dados estão parcialmente rotulados e parcialmente não rotulados. Nessa abordagem tenta-se utilizar o treinamento supervisionado para treinar o modelos nas amostras rotuladas para que este posteriormente rotule o restante das amostras. Também são utilizadas informações dos atributos das amostras para que estas sejam agrupadas como na abordagem não supervisionada. O modelo treinado pode então rotular os grupos de amostras que foram formados por algoritmos de agrupamento.

2.4 *Naive* Bayes

O algoritmo *naive* bayes é pertencente ao grupo de classificadores baseados em probabilidade. Esta abordagem, assim como outras que levam o nome do teorema de bayes utilizam a lei de probabilidade condicional de Bayes, conforme equação 2.1 (STUART, 1994).

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (2.1)$$

Onde A e B são eventos e $P(B) \neq 0$, dessa maneira $P(A|B)$ é a chance do evento A ocorrer sendo B verdadeiro. Da mesma maneira, $P(B|A)$ é a ocorrência do evento B se o evento A for verdadeiro. O classificador *naive* Bayes ou ingênuo (em tradução livre) assume uma forte independência entre os atributos, o que na prática pode não acontecer.

Esta abordagem pode ser implementada de diferentes formas, como em redes neurais ou implementando-se o cálculo do modelo probabilístico. Neste trabalho e para fins de comparação com outras abordagens, a implementação comparada foi em redes neurais. Desse modo, o objetivo dessa abordagem é classificar as amostras pela probabilidade condicional que cada valor de atributo contribui para que, de maneira independente, esta possua a maior probabilidade de pertencer a uma determinada classe dada aquela específica combinação de valores de seus atributos (RISH, 2001).

2.5 Função de base radial, *Radial Basis Function* (RBF)

Funções de base radial são funções do domínio real usadas para aproximar funções multivariáveis pela combinação linear de seus termos, criando assim, um espaço de aproximação onde é possível calcular a distância (geralmente euclidiana) entre um ponto qualquer e o centro desse espaço. Dentre as maneiras possíveis de implementação dessas funções, geralmente em Aprendizagem de Máquina utiliza-se a implementação na forma de redes neurais (BROOMHEAD; LOWE, 1988).

Neste trabalho, para fins de comparação, esta implementação (com kernel gaussiano) foi utilizada. Essa rede neural utiliza o vetor de atributos de entrada e calcula-se a distância entre este vetor e o vetor centro da função, também chamado de protótipo. Essa rede possui três camadas: A camada de entrada (que é o vetor de atributos), a camada escondida onde é calculada a aproximação dos dados de entrada com os pesos, a reposta de ativação de cada neurônio e a camada de saída que possui o número de neurônios de saída representando as classes procuradas. O neurônio de saída com a maior ativação é o resultado da classificação.

Cada neurônio ou unidade de cálculo é composto pelo vetor protótipo centro da função (ou kernel), uma função de base radial (geralmente gaussiana). O somatório das repostas de ativação desses neurônios multiplicado pelos pesos das conexões alimenta a camada de saída e ativação final. A cada etapa no processo de treinamento os pesos das ligações é atualizado, geralmente pela abordagem de descida do gradiente ou (*gradient descent*).

2.6 Árvores de Decisão

Árvore de decisão é uma estrutura de decisão que tem como objetivo discernir sobre uma instância iterativamente reduzindo o escopo de decisão pela partição de características daquela instância. Em Aprendizagem de Máquina, uma amostra (ou observação) é classificada pelos valores de seus atributos até que a combinação desses valores alcance o último nó da árvore também chamado de folha que representa a classificação final daquela amostra. Na etapa de treinamento, os valores que fornecem a separação dos dados dos atributos são testados, até que os melhores valores capazes de separar os dados são encontrados e a amostra pode percorrer a árvore para que se chegue na classificação final em uma das folhas (SHALEV-SHWARTZ et al., 2014).

Dentre as maneiras para que se escolha os valores que melhor separam os dados dos atributos, geralmente é utilizado o cálculo de ganho de informação ou entropia. Neste trabalho, a implementação utilizada é o algoritmo chamado *c4.5* que utiliza o cálculo de entropia para se determinar essa melhor separação ou "*best-split*" (QUINLAN, 1993).

$$H(T) = - \sum_{i=1}^J p_i \times \log_2 p_i \quad (2.2)$$

Sendo $H(T)$ entropia, p_i frações que representam a porcentagem de cada classe do nó filho

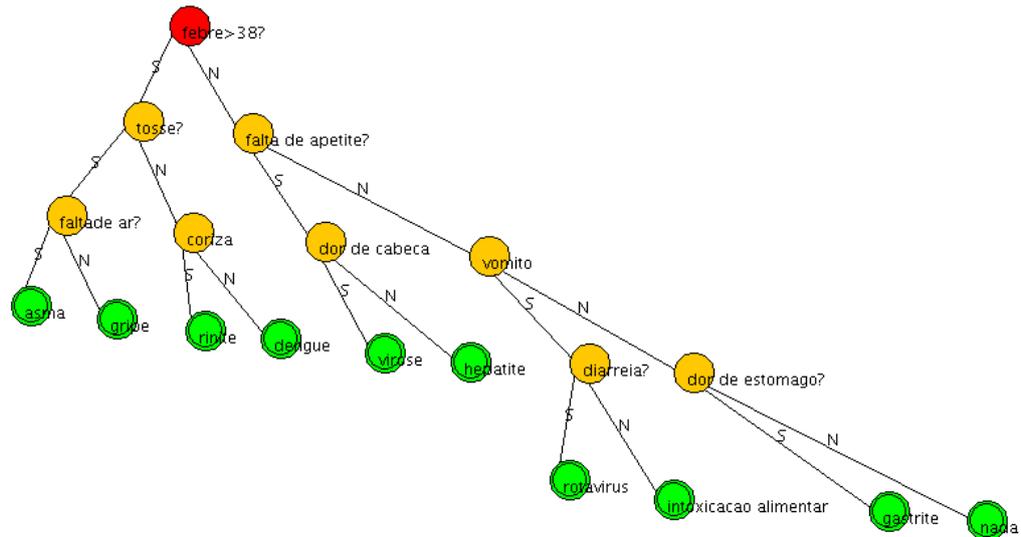


Figura 2.1: Exemplo de uma árvore de decisão a partir de sintomas

advindo de uma partição da árvore.

2.7 Florestas Randômicas (*Random Forests*)

O modelo florestas randômicas é um algoritmo criado em 1995 por Tim Kam Ho. É baseado em dividir árvores de decisão profundas com hiperplanos oblíquos, a cada árvore é apresentada um subconjunto aleatório de atributos (do total de atributos) e a folha resultante no maior número de árvores é o resultado da classificação final (HO, 1995). Isso foi feito para se ganhar precisão sem correr o risco de *overfitting* (viés nos resultados por causa do treinamento). Uma atualização do algoritmo foi desenvolvida por Leo Breiman (BREIMAN, 2001). Este trabalho utiliza esta versão do algoritmo de floresta randômica para a tarefa de classificação de pixels.

Florestas Randômicas é um algoritmo baseado em árvores de decisão. Como o nome sugere, é um largo número de árvores de decisão (geralmente entre 100 e 300), cada árvore de decisão gera sua classe de saída, e a classe que foi a saída do maior número de árvores é considerada o resultado da classificação final. Cada árvore individual, aleatoriamente seleciona amostras do conjunto de treinamento (com substituição) para construir seu próprio modelo e só tem a acesso a uma parte aleatória do conjunto de atributos de cada amostra. Em árvores de decisão convencionais, o atributo selecionado para particionar cada nó de cada árvore, entre todos os atributos, é aquele que produz a maior distância entre os nós filhos da esquerda e da direita. No modelo de florestas randômicas, cada árvore só tem acesso apenas a um subconjunto aleatório dos atributos para se fazer essa partição, fazendo cada árvore diferente e tão diversa quanto possível. Neste modelo, tem-se árvores que não apenas foram treinadas em subconjuntos diferentes dos dados, mas que também usam atributos diferentes para fazer a classificação.

O número de árvores no modelo, assim como o número de atributos aleatórios que cada árvore

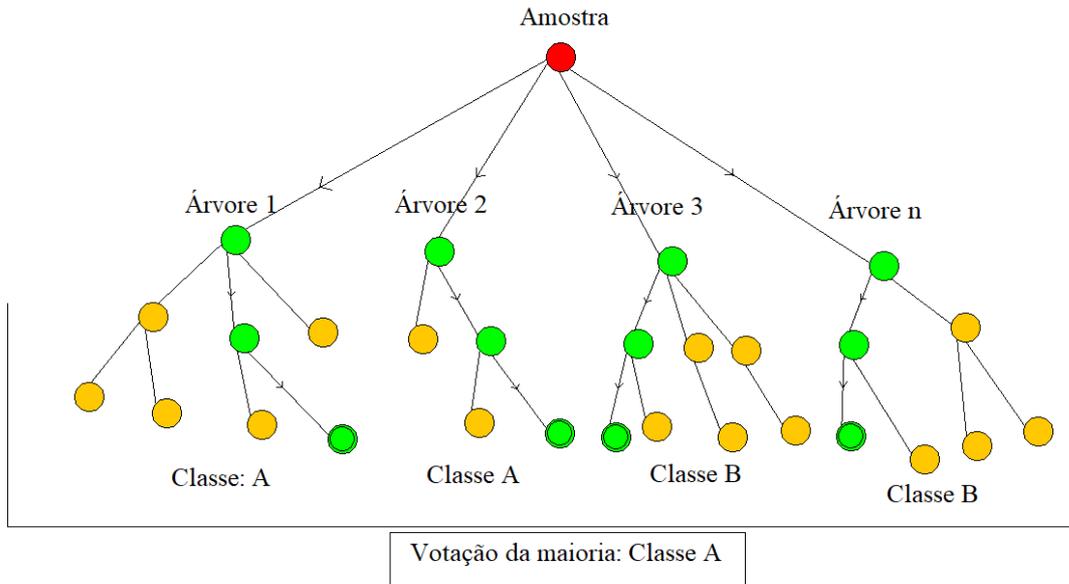


Figura 2.2: Esquema de ilustração do algoritmo de florestas randômicas

terá acesso, são aspectos que podem ser usados para ajustar e refinar o modelo. O algoritmo (que pode ser visto no Algoritmo 2) funciona da seguinte maneira: para cada árvore no conjunto (floresta), é selecionado uma amostra aleatória do conjunto de treinamento S (*dataset*) com substituição, onde $S(i)$ é a i -ésima seleção (*bootstrapping*). Então é aplicado, a árvore de decisão convencional com uma diferença. A diferença acontece da seguinte maneira: a cada nó de cada árvore não é analisado todas as possíveis partições de atributos. Ao invés disso, um subconjunto k , com $k \subseteq K$ de atributos, é aleatoriamente selecionado onde K é o conjunto de todos os atributos. O nó é então particionado usando o melhor atributo de k ao invés de K . Como k é consideravelmente menor que K , selecionar qual atributo oferece a melhor partição é computacionalmente mais eficiente. Cada árvore será diferente, mesmo que algumas árvores classifiquem erroneamente, possivelmente devido à combinação ao acaso de determinadas amostras com determinados atributos, a floresta (maioria) terá a tendência de classificar melhor e com menos vies que suas partes menores individuais.

Algoritmo 1 Pseudocódigo do algoritmo florestas randômicas

Requerido: Um conjunto de treinamento $S := (x_1, y_1), \dots, (x_n, y_n)$, Atributos K , e o número de árvores na floresta B .

função FlorestasRandômicas(S , K)

$H \leftarrow \text{null}$

for $i \leftarrow 1, \dots, B$ **do**

$S(i) \leftarrow$ A bootstrap sample from S

$h(i) \leftarrow$ FlorestasRandômicas($S(i), K$)

$H \leftarrow H \cup h(i)$

end for

return H

Fim-função

function AprendizagemRandomica(S , K)

A cada nó:

$k \leftarrow$ um sub conjunto de K

$node \leftarrow$ melhor partição em k

retorne: A árvore de decisão

end function

2.8 Visão Computacional

Visão Computacional é um campo de estudo científico da Inteligência Artificial que tem como objetivo simular a compreensão de imagens ou vídeos digitais do sistema visual humano utilizando artefatos computacionais (BALLARD; BROWN, 1982). Esta área de estudo provê, entre outros, técnicas e algoritmos para facilitar a captura e compreensão de objetos específicos e contextos em imagens digitais. Neste trabalho são utilizados os procedimentos de normalização de componente *Red Green Blue* (RGB) e a transformada circular de Hough.

2.9 Transformada circular de Hough

A transformada circular de Hough é um algoritmo utilizado em Visão Computacional e Processamento de Imagem com o objetivo de encontrar o maior círculo em uma imagem (ILLINGWORTH; KITTLER, 1987). Este algoritmo é uma especialização da transformada de Hough, que é um procedimento de extração de características específicas em imagens digitais por meio de uma votação dos valores de máximos locais que acontece no espaço paramétrico. No caso da transformada circular, os máximos locais são de círculos e um acumulador é usado para fazer a votação. No algoritmo 2, é possível visualizar este procedimento.

Algoritmo 2 Pseudocódigo do algoritmo da transformada circular de Hough

Requerido: Um vetor acumulador $A[x,y,r]$ sendo x e y as coordenadas na imagem e r os raios dos círculos

for cada $A[x,y,r] \leftarrow 0$ **do**
 Processe o filtro Gaussiano
 Converta a imagem em escala de cinza
 Utilize o operador de Canny
 Votação (A).
 Os círculos máximos locais no vetor A dão o espaço de Hough.
 O círculo mais votado no vetor A é o maior círculo na imagem.

end for

função Votação(A)

for cada pixel(x,y) **do**
 for para cada raio $r =$ menor raio possível esperado $r \leq$ maior raio possível esperado **do**
 for cada θ $t \leftarrow 0$ $t \leq 360$ **do**
 $a \leftarrow x - r * \cos(t * \pi / 180)$
 $b \leftarrow y - r * \sin(t * \pi / 180)$
 $A[a,b,r] += 1;$
 end for
 end for

end for

2.10 Normalização de imagem

O processo de normalização de imagem é muito utilizado nos campos de Inteligência Artificial em especial em Visão Computacional com o principal objetivo de destacar objetos de interesse em uma imagem ou suavizar (em alguns casos, eliminar) objetos fora da análise em questão (VANRELL et al., 2001). Em imagens não normalizadas a faixa de valores possível assumidos em um espaço de cor RGB é de 0 a 255 em cada uma dessas componentes. Muitos objetos podem já serem separados do restante da imagem com uma faixa menor de valores. Se essa faixa é reduzida, é reduzido também a influência de agentes externos como diferentes fontes de luz ou sombra que podem alterar os valores esperados para aquele objeto sob esta influência. Existem diversos procedimentos de normalização como: normalização de componentes específicas, normalização por histograma, escala de cinza e etc. Neste trabalho é utilizado a normalização da componente vermelha. Tendo em vista que para a classificação dos instares da mosca branca vários atributos são utilizado de cor e tamanho é interessante reduzir influências externas nessas características. Para a característica de cor desses objetos, a cor amarela e a cor verde são as principais cores usadas para determinação dessa classificação. Como o verde é uma cor primária no modelo RGB e sua faixa de valores fundamental, o amarelo é a cor advinda da mistura de verde com vermelho, optou-se pela normalização de uma das componentes da cor secundária que não reduzisse o escopo de informação da componente verde. Como os objetos não possuem a componente azul muito destacada, essa componente neste caso não possui uma faixa grande o suficiente para ser muito influenciada por fatores de iluminação externos.

O processo de normalização acontece da seguinte maneira: cada valor RGB de cada pixel

é substituído pela soma da componente vermelha individual dividida pelo total somado dessa componente (da imagem inteira) vezes 255. Dessa maneira, espera-se diminuir problemas de picos de valores ocasionados por fontes de luz arbitrárias ou vales nestes valores devido ao aparecimento de sombras que podem ocorrer nas situações nas quais as imagens que este trabalho utilizou foram tiradas.

2.11 Revisão da literatura

Muitos trabalhos tem abordado essa temática de reconhecimento e classificações de pragas sob condições diferentes. Em (CHO et al., 2007) foi utilizado armadilha de cola fixa em estufas para amostragem de mosca branca, áfídeos e tripses. Apenas insetos em fase adulta (que são capazes de voar) são capturadas. Neste caso, este método não suporta o tratamento preventivo contra a praga (antes que a planta sinta seus efeitos). Os autores utilizaram características morfológicas como cor e tamanho para encontrar separar os objetos e transformações no espaço de cor para encontrar os melhores valores dessas características que representassem cada espécie. Em 10 imagens a metodologia foi testada e os resultados de reconhecimento da mosca branca, nestas condições superam 90% (para a mosca adulta). Neste trabalho, as condições de captura das imagens são em condições menos restritivas sem o uso de armadilha de cola, bem como, os objetos foco neste trabalho não é somente a fase adulta.

No trabalho de (BARBEDO, 2014), a metodologia apresentou uma proposta para o reconhecimento e contagem da mosca branca. Neste trabalho a implementação da contagem foi semiautomática, onde correções manuais em várias imagens eram necessárias. Vários ângulos das folhas foram testados para se avaliar os possíveis impactos nos resultados. A metodologia também, foi testada em diferentes climas para que se pudesse aferir a interferência das condições de luz na imagem. O trabalho utilizou transformações no espaço de cor RGB para para destacar cores de determinados ínstares e o tamanho dos objetos encontrados. Diferenças nas tonalidades das folhas (que desviassem para o marrom ou cinza) não eram suportadas nesta metodologia. Neste trabalho a abordagem é totalmente automática, não necessitando de intervenção humana durante o processo. Os atributos utilizados neste trabalho são invariantes em relação a rotação e por isso não precisa ser considerada. Como os limiares de cores utilizados neste trabalho são faixas de valores em uma imagem normalizada a metodologia se mostra mais flexível com diferentes tonalidades folhas presentes nas imagens.

Na abordagem de (ESPINOZA et al., 2016), redes neurais foram utilizadas com o objetivo de identificar e contar moscas brancas em estufas com o uso de armadilhas de cola. Foram utilizados vários atributos da forma e tamanho dos objetos, como: excentricidade, alongamento, perímetro, área convexa e etc. Os resultados dentro dessas condições foram de 98,21% de acurácia na separação da mosca branca de tripses (*Frankliniella occidentalis spp.*). Novamente, neste trabalho não são considerados apenas uma fase do ciclo de vida da mosca branca mas sim, todo seu ciclo de vida. As condições de utilização dispensam uso de armadilha de cola e a metodologia proposta pode ser usada de maneira mais flexível em campo aberto (fora de estufas).

Na pesquisa de (CHEN et al., 2016), foi proposto um método para encontrar ninfas de alfídeos usando (*Convolutional Neural Networks*) CNNs. Os autores aplicaram essa metodologia em 102 imagens com diferentes níveis de infestação na folha (de 1 a 96 ninfas em uma única folha). As imagens foram previamente manualmente cortadas para que a região de interesse fosse focada. Esta abordagem obteve de resultado um *F-score* de 0,96. Esta abordagem, no entanto, foca apenas na presença ou ausência das ninfas e não sobre suas fases de desenvolvimento. O custo computacional de CNNs podem fazer a adaptação da solução para dispositivos móveis, uma tarefa pouco viável atualmente.

Neste trabalho foram usadas 240 imagens onde a presença de 6 classes poderiam estar presentes, a presença ou ausência de infestação é uma informação importante para o manejo de pragas, porém, é importante ressaltar que somente essa informação pode não ser suficiente para o manejo completo e de alto nível. A informação da dinâmica populacional da infestação pode alterar a estratégia de tratamento agrícola e por isso, essa informação é de vital importância.

Uma interessante abordagem foi usada por (LIU et al., 2016), para identificar alfídeos em campos de trigo, usando máquinas de vetor de suporte, *support vector machine* (SVM). Os autores utilizaram imagens com uma resolução de 1200 x 1800 pixels que foram repartidas em mais de 6000 imagens de 20 x 30 pixels. Este trabalho corrobora com a importância dos algoritmos de Processamento de Imagem e Visão Computacional que são capazes de entregar atributos fortes sobre os dados. Mesmo que o classificador usado não seja complexo ou convoluto, uma profunda exploração dos atributos pode entregar resultados compatíveis com classificadores mais robustos. Na presente dissertação, não somente as ninfas da mosca branca são identificadas e contadas mas a mosca adulta e a exúvia também são consideradas para análise.

Na metodologia proposta por (KARUNA et al., 2019), cinco classes de objetos foram o foco da identificação e classificação em plantas de chilli. Sendo eles: a presença de bolor cinzento nas folhas, bolor amarelo, moscas brancas adultas, curvatura anômala da folha (causada pro agente viral) e folhas saudáveis. Foram utilizadas 225 imagens sendo 45 imagens para cada classe. A metodologia utilizada foi CNNs onde as áreas afetadas eram encontradas e usadas para o treinamento dessas redes. Os resultados demonstraram o correto reconhecimento e classificação das regiões afetadas das folhas acima de 93% sendo a mosca branca a classe com o pior reconhecimento na detecção de presença. Os autores neste trabalho não abordaram a contagem ou os ínstars da mosca branca. Sendo somente o foco do trabalho um diagnóstico do que cada folha possuía no momento da análise do modelo. Comparativamente, neste trabalho a presença ou ausência de moscas brancas não é somente averiguado (identificação) mas sua distribuição nas folhas (contagem) e como essa distribuição é organizada (classificação).

No trabalho de (RUSTIA et al., 2019), uma abordagem de treinamento não supervisionada foi utilizada. Os autores utilizaram armadilhas de cola em estufas para capturar cinco classes de objetos: moscas, mosquitos, mariposas, tripes e moscas brancas. Dentro de cada armadilha todos esses objetos poderiam estar presentes. Utilizando também CNNs através do modelo "*You Only Look Once*" YOLO v3. Os objetos foram capturados das imagens e separados. No total, 12 imagens foram utilizadas para extração dos objetos. Nesta fase de extração o modelo utilizado pelos autores

conseguiu 97% de identificação dos objetos e 9% de falsos-positivos. Na fase da classificação dos objetos encontrados, um pré-modelo treinado com as 12 imagens iniciais era utilizado para subsidiar uma rede CNN não supervisionada. Em tempo real, fotografias nas estufas eram tiradas e processadas, a rede CNN treinada (utilizando treinamento supervisionado) rotulava os objetos encontrados com um grau de confiança e as novas imagens se tornavam amostras para o modelo não supervisionado. Dessa maneira, 40 novas imagens foram obtidas e seus objetos organizados em categorias pela rede não supervisionada. Os resultados, em média, foram de 0,92 de *F-score*. Os autores apenas procuravam identificar se, entre os insetos considerados, a mosca branca estava presente. Como essa metodologia utilizou armadilhas de cola, somente o último estágio de vida da mosca branca (adulta) é capaz de ser capturada e analisada. Os ínstaes da mosca branca e seu ciclo de vida não foram escopo desse trabalho, diferentemente do que é proposta nesta pesquisa.

O método de amostragem através da contagem manual em folhas ainda é um método muito recorrente no Brasil e sofre do problema da necessidade de um especialista humano (passível de viés humano e erros de execução), bem como de sua disponibilidade e limitação em função da carga de trabalho. Este método também pode levar a subjetificação dos resultados. Considerando-se que na literatura a abordagem predominante ainda não é a classificação dos ínstaes da mosca branca, este trabalho contribui para preencher esta lacuna metodológica.

Capítulo 3

Metodologia

Neste trabalho, 240 imagens de folhas de soja são consideradas para análise. Essas imagens foram disponibilizadas pela Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), cada imagem foi capturada com uma câmera compact de 10-Megapixels (Panasonic Lumix DMC-LZ10) de espectro comum.

Em cada imagem, um tubo de PVC de 10 cm de diâmetro foi aplicado sobre a folha com o objetivo de isolar a região de análise, bem como, eliminar quaisquer distorções de tamanho devido á proximidade com a câmera. Um exemplo de uma imagem original pode ser visto na Figura 3.1, assim como na Figura 3.3(S). Dentro dessas regiões de interesse é esperado que se encontre moscas brancas em quaisquer estágios de seu ciclo de vida, incluindo exúvias. É possível, também, que outros objetos como fungos, outros insetos e doenças na folha estejam presentes.

O principal objetivo deste trabalho prover uma proposta de reconhecimento e contagem de moscas brancas em seus estágios de vida (ínstares) que possa superar a contagem manual em ambientes pouco controlados (próximos ao uso de campo quanto possível) de maneira automática. Cada imagem foi manualmente marcada nas etapas de treinamento e todos seus objetos marcados e contados. Para cada imagem, um vetor de seis atributos foi produzido. Por exemplo, da Figura 3.1 é possível construir o vetor demonstrado na Tabela 3.1

A análise deste trabalho, consiste em ser capaz de identificar nas imagens: exúvias, 1º ao 4º instar e moscas brancas adultas que estejam presentes, contá-las e repassar essa informação para uma tomada de decisão informada possa ser realizada. A identificação e o controle de uma população ao longo do tempo é crucial para diferentes métodos de controle possam ser empregados com sucesso. Para atingir este objetivo, é preciso encontrar estes objetos de interesse e classificá-los em 6 diferentes classes em um tempo computacional que não atrapalhe o uso desta metodologia em situações reais no campo. Para tal, dois principais eixos temáticos são desenvolvidos: Visão Computacional e Aprendizagem de Máquina.

Tabela 3.1: Exemplo de vetor que representa os achados na Figura 3.1

<i>Exuviae</i> (1 a 19)	ínstar 1 (20 a 22)	ínstar 2 (23)	ínstar 3	ínstar 4	Adultas
19	3	1	0	0	0



Figura 3.1: Exemplo de uma imagem fonte, marcada, que produz o vetor da Tabela 3.1.

Sendo assim, a primeira abordagem é utilizar Visão Computacional para extrair os atributos das imagens e a partir daí, Aprendizagem de Máquina para classificação e análise dos dados. Na Figura 3.2 a proposta como um todo pode ser visualizada em um fluxograma. S é a imagem de entrada ou fonte (*source*) e as imagens $O1 - O5$ (*output 1 - output 5*) são as imagens de saída em cada etapa do sistema.

Primeiramente, a imagem é redimensionada para ser 20% menor em ambos altura e largura (proporcionalmente) para reduzir o montante de processamento necessário. A imagem é tratada no espaço de cores RGB (Red, Green, Blue), onde todas as cores são representadas em um intervalo numérico entre 0 e 255 para cada componente. Antes que qualquer outra operação ocorra, a imagem tem seu componente vermelho normalizado.

Este processo de normalização é utilizado para aliviar picos e vales que poderiam ocorrer devido a objetos espúrios e ruído. Esta imagem normalizada é a base para todas as operações a partir deste ponto. A Figura 3.3 contém a imagem fonte (S) todas as imagens de saída ($O1$ a $O5$) através do processo. Para reduzir o escopo de busca e análise e para fazer uso da região de interesse pré-determinada, a transformada circular de Hough foi utilizada como em (ILLINGWORTH; KITTLER, 1987) e detalhado no capítulo 2. Este é um procedimento comum em Visão Computacional para se encontrar círculos em uma imagem. Neste caso, o objetivo é encontrar a secção circular do tubo de PVC de 10 cm de diâmetro que foi previamente colocada em cada imagem. Estas operações resultam em uma região de interesse separada em uma imagem normalizada, como

pode ser visto na Figura 3.3 (O1).

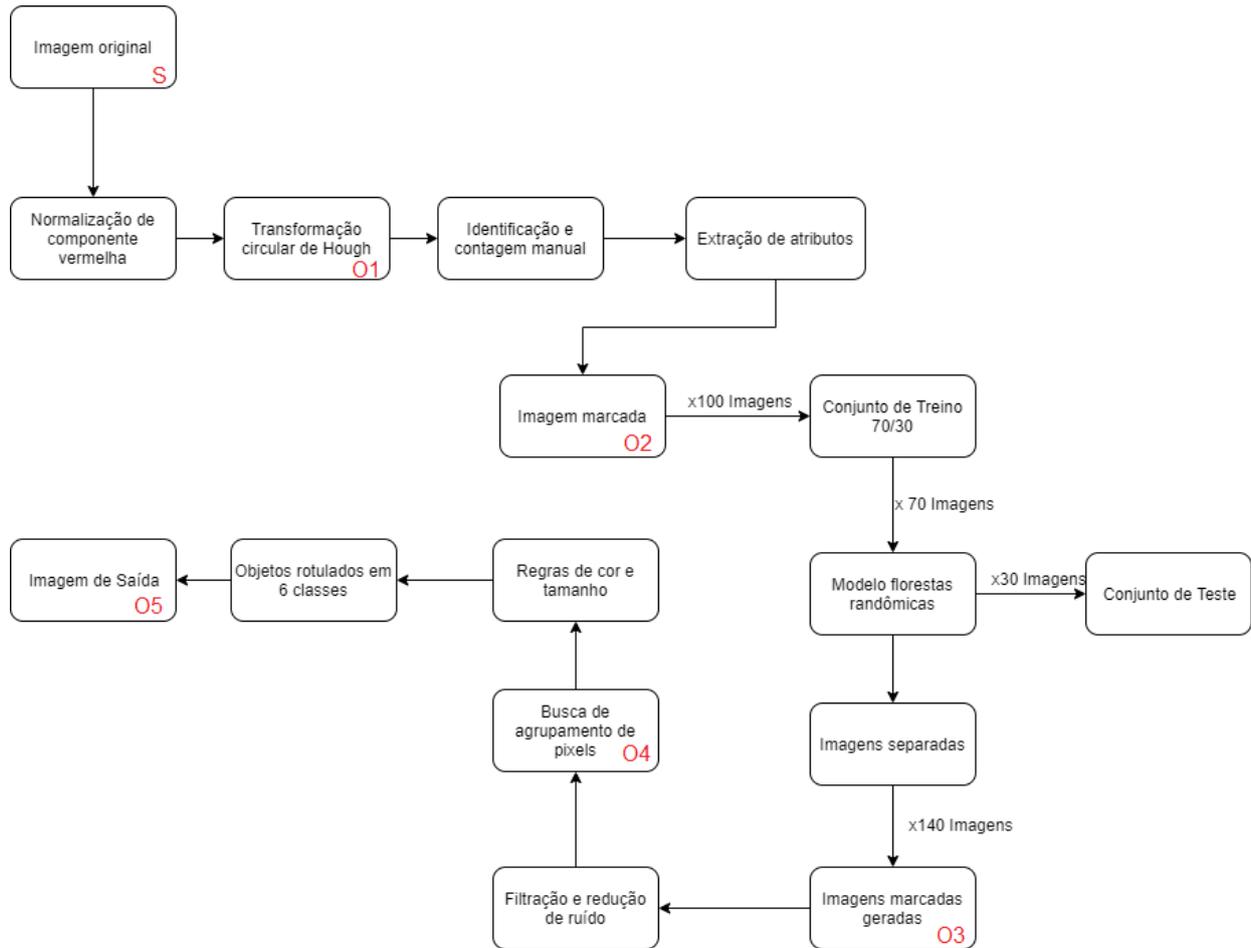


Figura 3.2: Fluxograma da metodologia proposta.

3.1 Seleção de atributos e treinamento

Na imagem normalizada, o próximo objetivo é ser possível discernir entre pixels ruidosos ou de fundo, de pixels que formam objetos de interesse. Nesta etapa é utilizada a imagem fonte para servir como um guia visual para a marcação destes pixels (representados pela cor azul) de todos os pixels que pertencem a algum objeto de interesse (de qualquer uma das classes) Figura 3.3 (O2). Neste ponto, a imagem estará dividida entre pixels marcados (supervisionados) e pixels comuns (não marcados). De cada pixel, os seguintes atributos são extraídos para criarem cada uma das amostras: componente vermelho normalizado (inteiro), componente azul normalizado (inteiro), componente verde normalizado (inteiro), número de vizinhos marcados (inteiro), posição X (inteiro), posição Y (inteiro), alongamento no eixo X (inteiro), alongamento no eixo Y (inteiro) e marcado (nominal: sim/não). O conjunto de treinamento são todas estas amostras extraídas dos pixels das imagens.

Como a imagem está normalizada, as características mais contrastantes tendem a permanecer e algumas nuances menores podem ser perdidas. Pixels similares que formam formas fechadas,

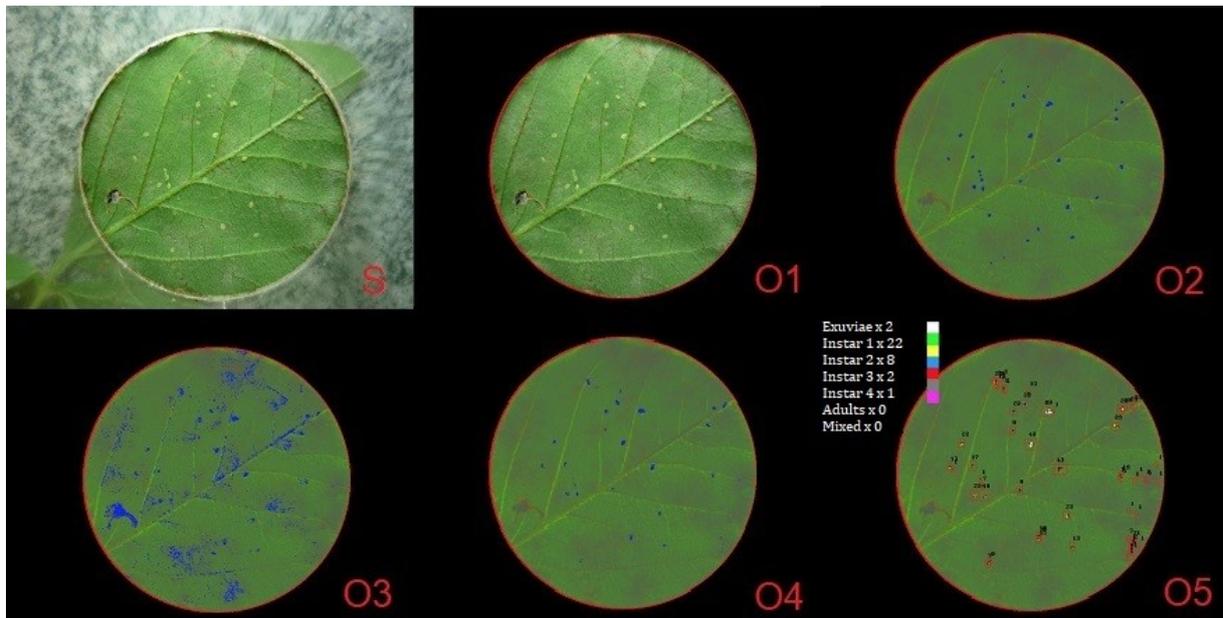


Figura 3.3: Imagem exemplificando cada saída de cada etapa da metodologia. S é a imagem fonte, O1 - O5 são as saídas das etapas marcadas na Figura 3.3. As imagens individuais com mais detalhes podem ser visualizadas no Anexo 1.

tem a tendência de se agrupar e sendo assim, não desviar demais em atributos como posição, cor e número de vizinhos similares. O atributo (número de vizinhos) é a soma de todos os vizinhos que não desviem mais de 20% em cor do pixel em análise e que estejam marcados. A posição em X e Y são a coordenadas nestes eixos (sendo o canto superior esquerdo da imagem o ponto de origem). O alongamento em X e Y são os números de pixels similares (que também foram marcados) ao longo daquele eixo, usando o pixel em análise como origem, até que um pixel diferente (desviado mais de 20% ou não marcado) seja encontrado. O atributo (marcado) representa se aquele pixel foi (marcado) como sendo pertencente a um objeto de interesse, “sim” caso isso tenha acontecido e “não” caso contrário. Um exemplo de uma amostra de pixel(Tabela 3.2):

Tabela 3.2: Vetor de exemplo da amostra retirada de cada pixel

Vermelho	Verde	Azul	pos X	pos Y	Vizinhos	Along. X	Along. Y	É objeto
78	120	55	8	101	3	7	6	No

Das 240 imagens, 100 foram usadas para treinamento. Cada imagem com aproximadamente 181 mil pixels usados como amostras. Dessas 100 imagens, 70 foram usadas para montar o conjunto de treinamento e os restantes 30 para o conjunto de testes. As 140 imagens restantes não foram usadas nesta etapa e por isso não tiveram contato com o modelo na fase de treinamento.

A natureza dos dados usados neste trabalho são imagens. Assim sendo, os dados que compõem o espaço visual representam todo o escopo da informação utilizada. Dados visuais são facilmente afetados por ruído: variações de luz ambiente, sombras, oclusão (parcial ou total), foco da câmera, distorções de tamanho pela distância da câmera e etc. Em situações reais, o operador do equipamento fotográfico pode não controlar perfeitamente as condições nas quais as imagens foram obtidas.

Como mencionada anteriormente, a normalização da imagem ajuda a aliviar alguns destes problemas (variação da luz, sombra e etc). O tubo de PVC de tamanho determinado ajuda a aliviar os problemas de escala. Aprendizagem de Máquina possui diversos modelos que podem ser utilizado em tarefas de classificação, e performar de maneira acurada mesmo em ambientes ruidosos. Como essa metodologia tem como um dos seus objetivos ser portada para dispositivos móveis no futuro, para que seja usado em campo, também foram considerados modelos que fossem computacionalmente leves. Para aferição de qual algoritmo de Aprendizagem de Máquina melhor se adaptaria ao dados da pesquisa, um pré-modelo com 30 imagens aleatoriamente selecionadas foi utilizado. Dessas 30 imagens, 21 foram usadas para o conjunto de treinamento e 9 imagens para o conjunto de testes. Essas condições foram repetidas para todos os algoritmos testados. Os algoritmos testados são: árvore de decisão (c4.5) (QUINLAN, 1993), naive bayes (RISH, 2001), funções de base radial (usadas em redes neurais) com kernel gaussiano (BROOMHEAD; LOWE, 1988) e florestas randômicas (HO, 1995). Na Tabela 1, os resultados de testes preliminares usando o software WEKA (versão 3.8.2) (FRANK; MARK; IAN, 2016), em suas configurações padrões no pré-modelo podem ser visualizados. Figura 3.4a são as funções de base radial, Figura 3.4b é naive bayes, 3.4c são floresta randômicas e 3.4d árvore de decisão.

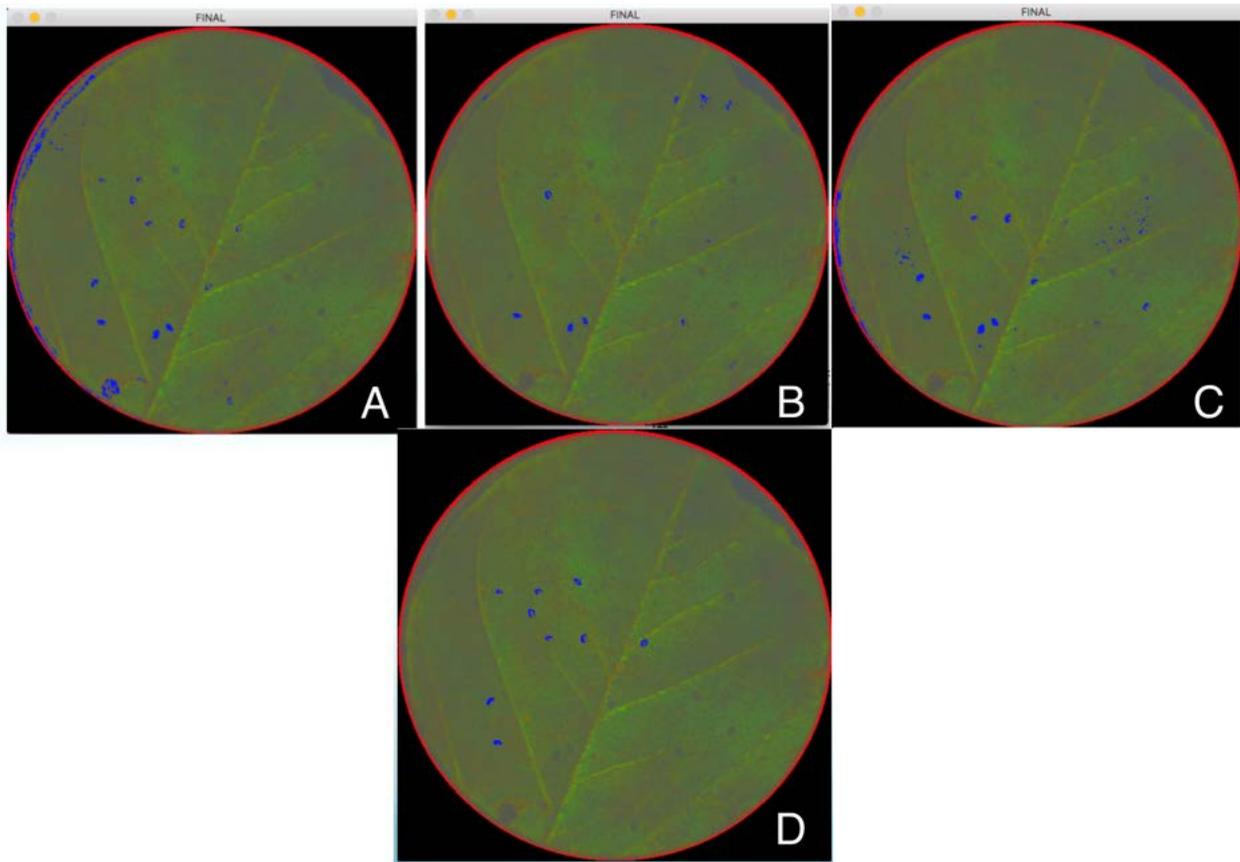


Figura 3.4: Imagens resultantes dos algoritmos testados no pré-modelo (30 imagens).

O algoritmo que obteve o melhor resultado neste pré-modelo e que teve melhor sinergia com os dados apresentados foi florestas randômicas. Apartir daqui, esse foi o algoritmo selecionado para a construção do modelo final. Em sua publicação original deste algoritmo Ho (1995) demonstrou

Tabela 3.3: Resultados dos testes no pré-modelo (30 imagens) no algoritmos testados

Modelo	Acurácia por pixel	Precisão	<i>Recall</i>	<i>F-measure</i>
<i>Naive Bayes</i> (Figura 3.4a)	78,75%	78,91%	78,88%	78,71%
Árvore de decisão (C4.5)(Figura 3.4b)	81,39%	81,40%	81,40%	81,40%
Florestas Randômicas (Figura 3.4c)	88,95%	85,18%	85,88%	85,88%
Funções de base radial (Figura 3.4d)	78,34%	78,51%	78,30%	78,40%

a metodologia aplicada em um problema de natureza similar ao desenvolvido neste trabalho, com dígitos escritos manualmente utilizando imagens (pixels) como amostras. Como mencionado anteriormente, o algoritmo do florestas randômicas é treinado em subconjuntos dos dados utilizando subconjuntos dos atributos, quanto maior o número de atributos (dimensionalidade das amostras) mais este modelo se destaca em relação aos outros testados, sendo o maior o número de divisões possíveis desse subconjunto dos atributos. Como as amostras de pixels apresentam não ser linearmente separáveis, os modelos de RBF e *Naive Bayes* obtiveram resultados próximos entre si. No entanto como esses modelos são mais suscetíveis a ruídos aleatórios (presentes em imagens reais), o florestas randômicas obteve um resultado superior.

O algoritmo de árvore de decisão foi o que obteve o resultado no pré-modelo mais próximo do florestas randômicas. Porém como o florestas randômicas possui a qualidade de se adaptar melhor a atributos mais fortemente correlacionados que o algoritmo de árvore de decisão, tendo em vista a parte aleatória do algoritmo, o floresta randômicas obteve o melhor aproveitamento do treinamento no pré-modelo.

Para propriamente utilizar o modelo de florestas randômicas, 240 imagens foram convertidas em dados de atributos, 100 foram aleatoriamente selecionadas. Dessas, 70 foram aleatoriamente selecionadas para compor o conjunto de treinamento e os 30 restantes para compor o conjunto de testes. O conjunto de testes teve seu último atributo (marcado) apagado. As 140 imagens restantes não foram usadas nessa etapa e foram usados para o conjunto de validação. O modelo tem o objetivo de classificar cada pixel em duas classes: “sim” para pixels que pertencem a algum objeto de interesse e “não” para qualquer outro pixel. Uma imagem manualmente supervisionada pode ser vista na Figura 3.3 (O2).

3.2 Balanceamento e algoritmo SMOTE (CHAWLA et al., 2002)

Como esperado, converter uma imagem supervisionada em dados irá levar a grande maioria dos pixels a serem classificados como “não” e apenas uma pequena quantidade de pixels manualmente supervisionados e marcados a serem classificados como “sim”. Isso acontece devido a grande maioria dos pixels na imagem não pertencer a nenhum objeto de interesse e o somente uma minoria pertencer. Na soma de 100 imagens, 10.797.224 pixels foram classificados como “não” e apenas 24.985 classificados como “sim”. Em comparação com o total, a classe “sim” está sub-representada. Se o modelo erradamente predissesse todos os (24.985) pixels marcados, este modelo apresentaria um erro de apenas 0,002%. Para que acurácia do modelo fosse útil, o conjunto de treinamento foi balanceado em 3 diferentes abordagens:

1) Do total de amostras da classe “não”, 24.985 amostras foram aleatoriamente selecionadas para igualar com o mesmo número de amostras da outra classe. De maneira que nenhuma amostra artificial foi gerada.

2) Gerar dados artificiais usando o algoritmo (SMOTE) (Chawla et al, 2002) para super amostrar a classe “sim” para que essa igualasse o número de amostras da outra classe. Gerando assim, 10.772.239 amostras artificiais.

3) Gerando menos dados artificiais que a abordagem 2, aleatoriamente selecionando 30.000 amostras da classe “não” e gerando 5015 amostras artificiais da classe “sim”. Fazendo com que ambas possuíssem 30.000 amostras.

Utilizando o modelo proposto nestas 3 diferentes abordagens aplicada no software WEKA (FRANK; MARK; IAN, 2016), a melhor acurácia atingida foi da abordagem 1. O conjunto de treinamento foi reconstruído 10 vezes, cada vez escolhendo 24.985 amostras aleatórias da classe “não” do total e colocado para preencher o último atributo (“marcado”) no conjunto de teste. A acurácia média for 95,85% de pixels corretamente classificados, como pertencentes ou não a um objeto de interesse. Uma imagem com pixels marcados pelo sistema pode ser vista na Figura 3.3 (O3).

3.3 Extração de objetos e filtragem de ruídos

Com o modelo já treinado e capaz de discernir entre pixels que pertencem ou não a objetos de interesse, o próximo passo é encontrar e isolar o conjunto desses pixels que foram marcados como partes de um objeto. O modelo treinado agora é utilizado nas 140 imagens restantes e previamente não vistas pelo modelo. São esperados falso-positivos nos quais pixels são erroneamente marcados como partes de objetos. Pixels que pertencem a formas específicas tem a tendência de se agruparem. Neste passo, são exploradas estas propriedades e utilizados procedimentos de redução e filtragem de ruídos, para minimizar pixels ruidosos isolados que não pertencem a nenhum objeto. Os seguintes procedimentos foram utilizados:

1) Se um pixel marcado não possuir no mínimo três vizinhos também marcados, em qualquer uma dos oito possível pixels vizinhos, este será eliminado da análise.

2) Se um pixel não marcado possuir cinco ou mais vizinhos marcados, este também será marcado.

Para que se diferencie pixels de interesse de pixels do plano de fundo, majoritariamente da cor verde para folhas saudáveis ou algum tom de verde mais próximo do amarelo (para folhas doentes), foi utilizado a média $((R + G + B) / 3)$ dos três componentes do pixel em análise chamado de *mediaPixel*. Para que se encontre o melhor valor máximo de verde permitido em um determinado pixel antes que este seja considerado próximo de mais do fundo, representado por *tetoVerde*. Foi utilizado a equação 1. Foram realizadas avaliações práticas nas imagens e verificou-se que esta precisão nos valores foi a que obteve os melhores resultados.

$$tetoVerde = (0,01905 \times mediaVerde) + 0,32101 \quad (3.1)$$

Note que, a variável *mediaVerde* representa a soma de todos os componentes verdes de todos os pixels, dividido pelo total de pixels, ou seja a média do componente verde da imagem inteira.

3) Se um pixel possui seus componentes azul e verde da seguinte maneira:

$$componentAzul < mediaPixel - (mediaPixel \times 0,45) \quad (3.2)$$

$$componenteVerde < tetoVerde \times mediaPixel \quad (3.3)$$

Se ao menos uma dessas inequações 3.2 e 3.3 ou ambas forem falsas, o pixel em análise é considerado como sendo de fundo e excluído da análise. Se ambas forem verdadeiras para o pixel em análise, este pixel pode permanecer.

Depois desses três procedimentos a maioria do ruído é eliminado, como pode ser visto na Figura 3.3 (O4).

A próxima etapa é encontrar esses grupos de pixels marcados e classificá-los. A abordagem utilizada é a janela de tamanho dinâmico. A imagem é iterativamente escaneada, do ponto de origem até o a posição máxima de altura e largura. Se um pixel azul (marcado) é encontrado, então o algoritmo tenta encontrar uma janela com um tamanho que englobe aquele grupo de pixels. Este algoritmo é mostrado no Algoritmo 3.

O primeiro pixel encontrado é chamado de pixel “pivô”. Da posição do pixel pivô, o algoritmo vai para baixo verticalmente até encontrar o primeiro pixel não azul. Se o próximo pixel a este anteriormente encontrado também não for azul (para aliviar problemas de continuidade nas formas) a posição inferior da janela é encontrada. De volta á posição do pivô, o mesmo procedimento é realizado para a direita e para a esquerda, incrementando e decrementando horizontalmente para que se encontre a posição superior direita. O pixel pivô é considerado a posição superior esquerda da janela. Tendo os quatro pontos que delimitam essa janela e adicionando cinco pixels extra de margem, a janela de tamanho dinâmico é formada e todos os pixels anteriormente azuis são re-coloridos para que não sejam contados novamente no escaneamento iterativo que está acontecendo.

Algoritmo 3 Janela de tamanho dinâmico para englobar os grupos de pixels

```
for int i = 0 ;i < image.linhas;i++ do
  for int j=0 ;j < image.colunas;j++ do
    int b = imagem.ComponenteAzul(i,j);
    int g = imagem.ComponenteVerde(i,j);
    int r = imagem.ComponenteVermelha(i,j) ;
    if b==255 and g==0 and r==0 then
      Ponto pivô = Ponto(i,j);
      int x=0,y=0;
      x = i;
      y = j;
    end if
    while imagem.ComponentAzul(x,y) == 255 and imagem.ComponenteVerde(x,y) ==0
and imagem.ComponenteVermelho(x,y) ==0 and x >= 0 and x< imagem.linhas) do
      x = x + 1
    end while
    Ponto baixo = Ponto(x,y);
    x = i;
    y = j;
    while imagem.ComponenteAzul(x,y) == 255 and imagem.ComponenteVerde(x,y) ==0
and imagem.ComponenteVermelha(x,y) ==0 and y >= 0 and y <imagem.colunas) do
      y = y + 1
    end while
    Ponto direita = Ponto(x,y);
    x = i;
    y = j;
    while Imagem.ComponenteAzul(x,y) == 255 and imagem.ComponenteVerde(x,y) ==0
and imagem.ComponenteVermelha(x,y) ==0 y >=0 and y< imagem.colunas) do
      y = y - 1
    end while
    Ponto esquerda = Ponto(x,y);
    if esquerda.x-5 > 0 and baixo.x+5 < imagem.linhas) then
      janela = Imagem(Retângulo( Ponto(esquerda.y-(baixo.x-pivô.x),esquerda.x-5),Ponto
      (direita.y+(baixo.x-pivô.x),baixo.x+5) ) );
    end if
  end for
end for
```

Tabela 3.4: Regras de RGB e tamanhos para classificação dos objetos

Classes:	<i>Exúvia</i>	ínstar 1	ínstar 2	ínstar 3	ínstar 4	Adultas	Indefinido
Pixels Verdes	<5	>0	>5	>5	>5	<3	>8
Pixels Amarelos	<3	<3	<3	<3	<3	>5	<5
Pixels Claros	<8	<5	<5	<5	<5	>8	>8
Área (Pixels)	27 - 70	5 - 17	18 -27	28 - 34	35 - 40	40-55	23-95

Algumas dessas janelas são excluídas neste processo. Se uma janela contém menos que três pixels marcados ou mais que 115 pixels, imperfeições ou buracos nas folhas são eliminados neste passo, resultando em janelas vazias. Com todos os objetos extraídos na imagem, cada janela é individualmente armazenada, analisada com as regras de classificação para que se encaixe nas seis diferentes classes finais. Dos testes experimentais foi derivada a tabela de regras de classificação (Tabela 3.4). Na Figura 3.5 até a Figura 3.10, cinco imagens iniciais (A, B, C, D, E) e suas respectivas saídas do sistema (A1, B1, C1, D1, E1) ao lado direito. Os pixels que são considerados claros na Tabela 3.4 são aqueles que possuem os valores de todas as suas componentes maiores que 210. Os objetos encontrados que possuem características de mais de uma classes (geralmente quando objetos de classes diferentes estão sobrepostos) são classificados como "Indefinidos", não sendo estes objetos considerados para fins de análise ou resultados.

Ainda que os resultados da classificação possam ser sólidos, alguns problemas de classificação ocorreram em situações específicas. Se a folha contiver uma doença ou problema de saúde que altere a reflexão da luz e essa região fique mais clara, o sistema pode interpretar parte dessa região como parte de algum objeto e classificá-lo incorretamente. Na Figura 3.5 isto pode ser visualizado. Neste caso existem apenas quatro exúvias na folha e um ínstar 2, devido ao ínstar 2 estar localizado em uma região mais clara que o normal, foi erroneamente classificado como exúvia.

Na Figura 3.8, duas exúvias foram erroneamente classificados como ínstar 3 e 4 podem ser vistos no segmento inferior da imagem, isto ocorre quando, por alguma, razão (geralmente quanto mais antigas) as exúvias possuem um tamanho bem inferior ao esperado para essa classe. Se as exúvias forem muito pequenas e estiverem dispostas sobre uma área mais escura que o resto da folha ou em uma sombra, o sistema pode classificá-la de maneira incorreta. Se uma exúvia possuir o tamanho esperado e estiver disposta em uma região amarelada (devido a algum problema de saúde na folha) o sistema pode classificá-la como adulta. Isto ocorre porque uma mosca adulta pode aparecer com um tamanho menor, se por a caso esteja em uma pose relativa a câmera que sua perspectiva bidimensional se torna oval. Neste caso o sistema esperaria encontrar pixels amarelos (presentes na cabeça da adulta) para classificá-la dessa maneira, no entanto, se a exúvia estiver numa região que disponha dessa faixa de cor, a classificação errada pode ocorrer.

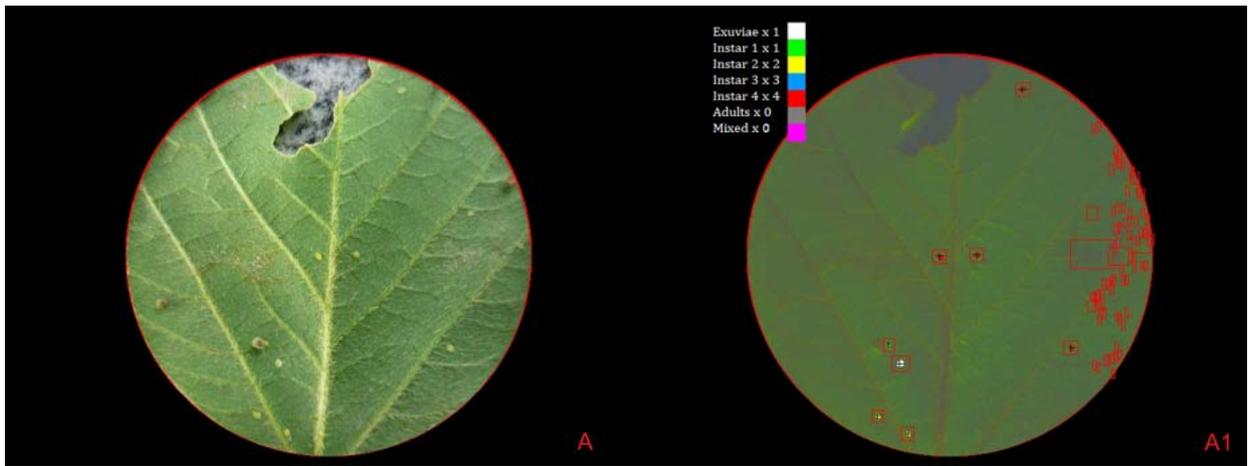


Figura 3.5: A é a imagem original e A1 a saída do sistema: 1 exúvia, 1 primeiro ínstar, 2 segundo ínstar, 3 terceiros ínstar e 4 quarto ínstar.

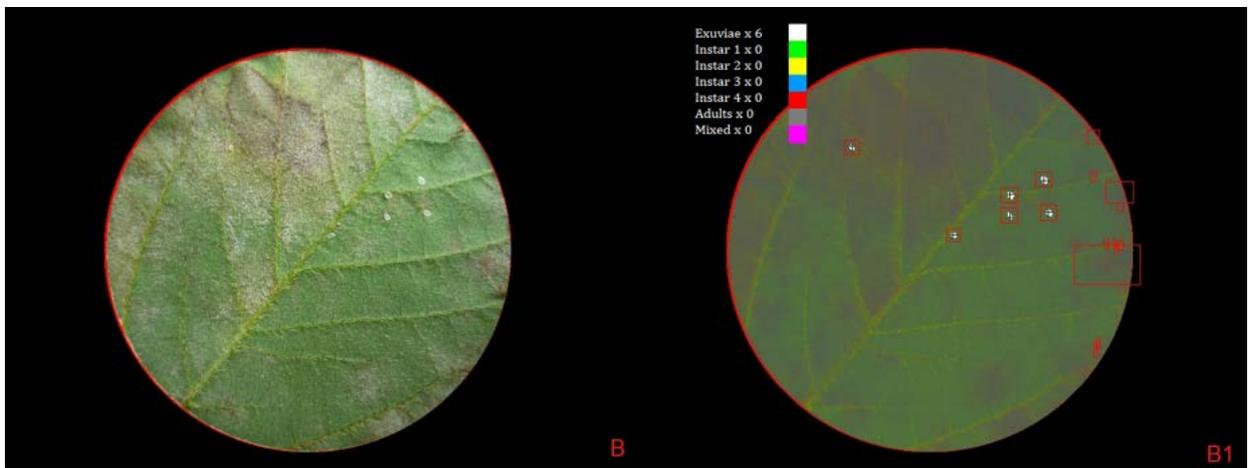


Figura 3.6: B é a imagem original e B1 a saída do sistema: 6 exúvias.

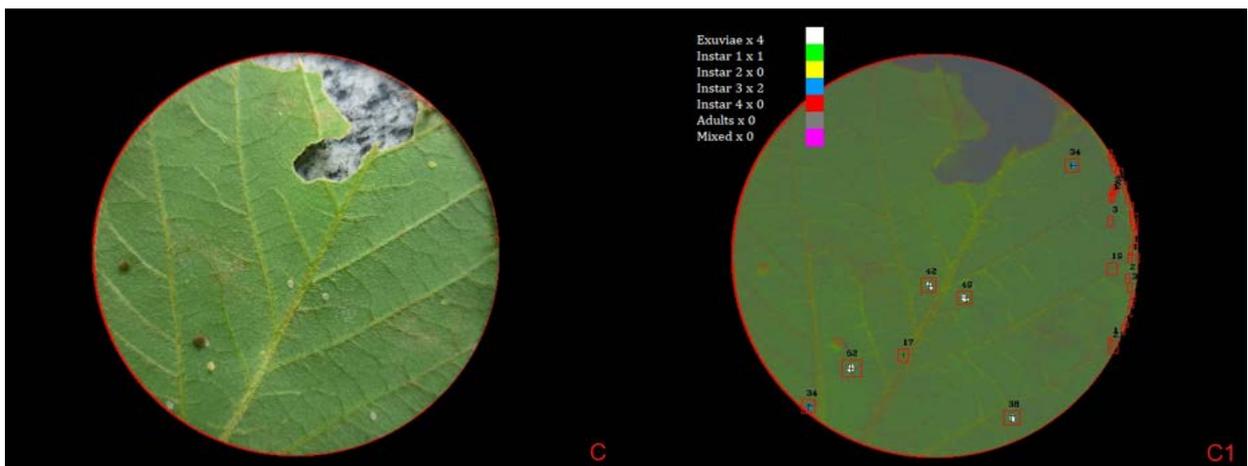


Figura 3.7: C é a imagem original e C1 é a saída do sistema: 4 exúvias, 1 primeiro ínstar e 2 segundo ínstar.

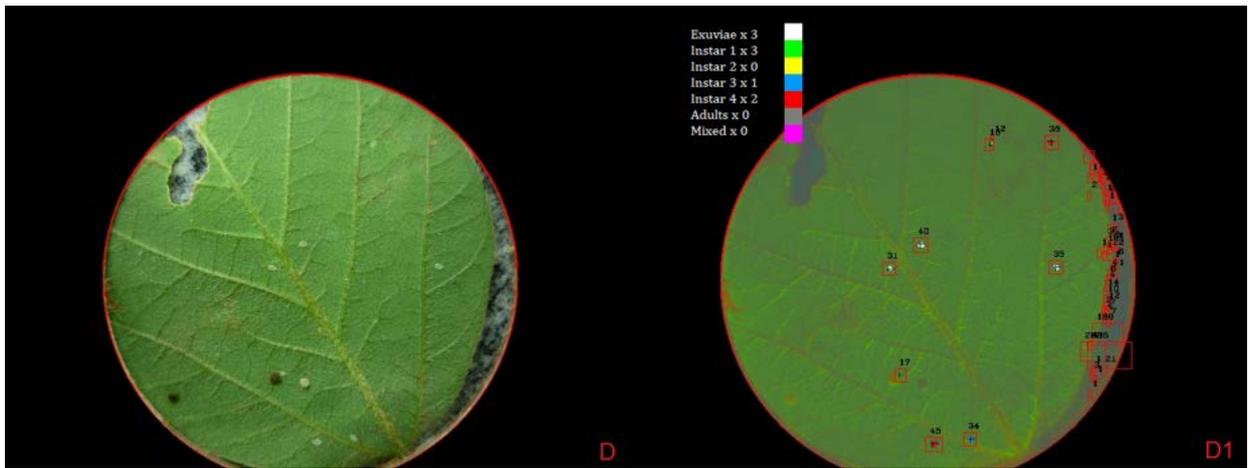


Figura 3.8: D é a imagem original e D1 é a saída do sistema: 3 exúvias, 3 primeiro ínstar, 1 terceiro ínstar, 2 quarto ínstars

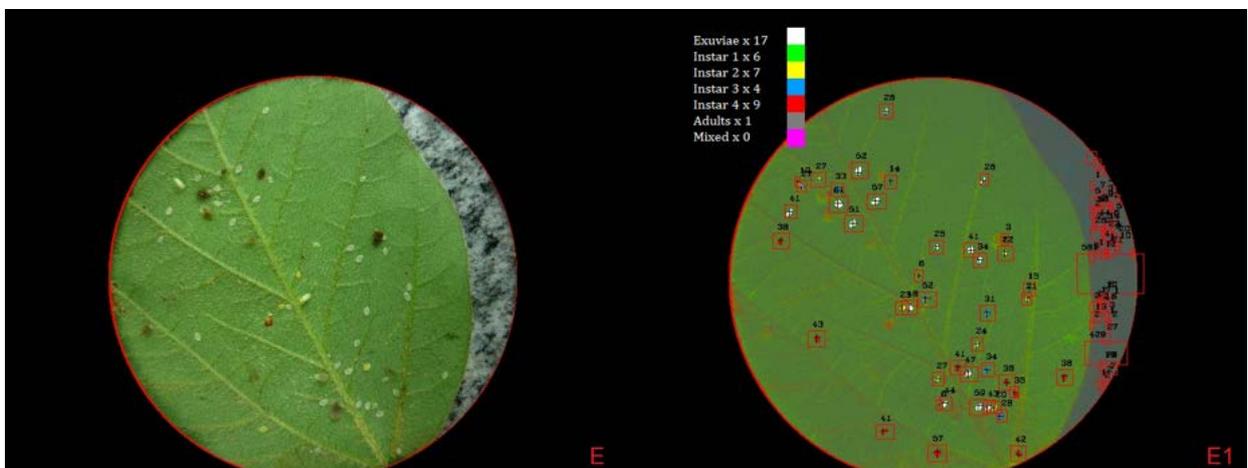


Figura 3.9: E é a imagem original e E1 é a saída do sistema: 17 exúvias, 1 primeira ínstar, 3 segundo ínstars, 3 quarto ínstars and 1 adulta

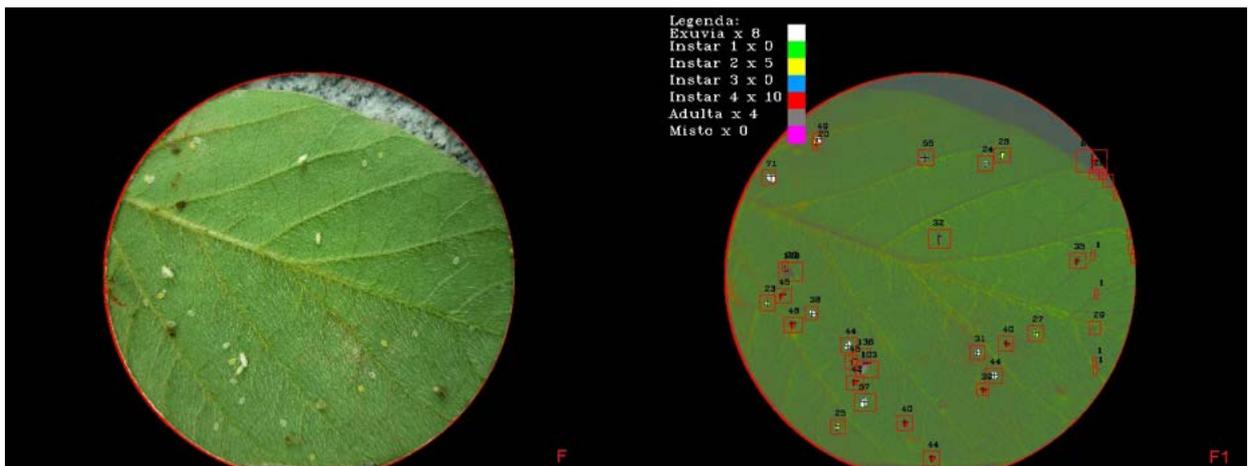


Figura 3.10: F é a imagem original e F1 é saída do sistema: 8 exúvias, 9 primeiro ínstar, 5 segundo ínstar, 10 quarto ínstar e 4 adultas.

Capítulo 4

Resultados

Depois que as regras de classificação de cor e tamanho e cores foram aplicadas, cada objeto foi re-colorido com a legenda para representar a seis classes, como visto na Figura 3.3 (O5). Note que, janelas vazias não são consideradas, pois foram eliminadas pelas regras de tamanho. A resposta do sistema é um vetor nos mesmos padrões dos que foram manualmente contados. Com os 140 vetores humanamente marcados e os 140 vetores produzidos pelo sistema, a diferença entre eles é então usada para produzir os resultados da classificação. As 140 imagens que não tinham sido usadas até então serão usadas para a validação do modelo treinado.

Uma validação cruzada (*cross-validation*) (KOHAVI, 1995) foi realizada para avaliar os resultados. Assim sendo, 10% das 140 imagens (14) foram aleatoriamente selecionadas 10 vezes para 10 testes. Uma mesma imagem poderia ser selecionada mais de uma vez para testes diferentes. Para cada teste uma matriz de confusão foi construída com os vetores produzidos pelo sistema e uma matriz dos vetores produzidos por humanos das mesmas imagens selecionadas. A média dos 10 testes é então considerado o resultado final. A acurácia média do sistema foi de 86,23% com uma precisão de 33,95%, um *recall* de 93,12% e um *F-score* de 49,08%. Na Tabela 4.1 estão os resultados de cada turno de teste (*fold*). Na Tabela 4.2 estão os resultados por classe de objeto encontrados nessas 140 imagens de validação e comparados com que o sistema proposto encontrou. No Anexo 1 é possível visualizar algumas imagens antes e depois da aplicação do sistema.

Analisando-se os resultados da métrica precisão e *recall* nos resultados dos testes da validação cruzada (Tabela 4.1) é possível notar que o modelo apresentou melhores resultados no *recall* do que na precisão. A precisão é dada pelo número de objetos verdadeiramente positivos dividido pelo número de objetos encontrados. Uma precisão alta pode significar que muitos objetos não são encontrados, porém os que são encontrados são classificados corretamente. Quanto maior o *recall* significa que mais objetos foram encontrados, e por isso, a classificação correta tem a tendência de diminuir. A métrica mais importante varia de acordo com a finalidade e contexto do sistema. Em um sistema de reconhecimento biométrico, por exemplo, em um condomínio, a precisão é muito mais interessante neste contexto que o *recall*. No contexto deste trabalho, para fins de manejo agrícola, é mais interessante que mais objetos sejam encontrados com menor precisão entre eles do que o contrário. Basta um exemplo de uma determinada classe para saber que essa

classe se encontra presente naquele ambiente. No entanto, se em uma determinada situação com poucos exemplares de uma determinada classe e o sistema com alta precisão, falha em encontrar qualquer exemplar daquela classe (mesmo que todos de outras classes que foram encontrados sejam corretamente classificados) aquela que seria considerada como ausente nesta situação. A presença ou ausência das classes nas folhas, é uma informação de alta relevância para seu manejo e por isso o *recall* tem uma significância prática maior que uma precisão mais alta neste contexto.

Com relação aos resultados da classificação por classe (Tabela 4.2) é possível notar que dos 2360 objetos reais presentes, 1826 foram encontrados, sendo destes, 63 falso positivos. Levando-se em consideração que nas imagens de validação vários tipos de texturas e colorações (advindo de fungos e outras doenças folhais) estavam presentes, o achados falso-positivos representam apenas 3,5% dos objetos encontrados. As características morfológicas das moscas na fase adulta, por serem mais distoantes dos outros ínstares, como a forma mais alongada e a coloração branca marcante contribuíram para essa classe ser a melhor classificada. Este resultado das moscas adultas é compatível com outras metodologias (citadas no Capítulo 2) que focaram em apenas nesta classe.

A exúvia, por também possuir uma coloração específica, obteve resultados próximos ao da mosca adulta. O maior problema da classificação desta classe é que, com o passar do tempo, o tamanho dos objetos dessa classe diminui (devido a evaporação da água), o que resulta em objetos com uma dispersão de valores de tamanho maiores que as outras classes. A classe que obteve a menor precisão é o ínstar 1. Tendo em vista que essa classe é a que possui o menor tamanho, sua coloração é mais confundível com o plano de fundo e seu tamanho é pouco representativo se comparado com algum possível ruído presente na imagem. Os demais ínstares apresentaram resultados próximos entre si. Na Tabela 4.3 é possível visualizar a matriz de confusão do objetos encontrados. Os objetos que o sistema falhou em encontrar (principal problema para o ínstar 1 por exemplo) não aparecem nessa matriz mas sim na Tabela 4.2.

Comparando-se os resultados obtidos com os da literatura mencionado no Capítulo 2, a metodologia apresentada possui resultados competitivos com outras metodologias. No trabalho de (BARBEDO, 2014) as classificação por ínstares da mosca branca atingiu resultados inferiores nos primeiros ínstares (ínstar 1 a 3) e exúvias. Apresentou resultados compatíveis para as adultas. Em relação ao trabalho de (CHO et al., 2007), onde somente as adultas eram consideradas, a metodologia apresentada neste trabalho superou em precisão na classe da mosca adulta. Como no trabalho de (ESPINOZA et al., 2016), o foco foi a separação dos objetos alvos de outros objetos, a metodologia neste trabalho consegue ser comparada sob a ótica de que apenas 3,5% dos objetos encontrados eram falso-positivos.

Nos trabalhos que utilizaram CNNs a presença ou ausência dos objetos encontrados foi o ponto principal de análise. Como nesta metodologia, a presença de pixels que formam os objetos das classes denotam a sua presença (mesmo que a captura dos objetos possa ser parcial) a metodologia apresentado obteve 95% de acerto para essa separação. No trabalho de (CHEN et al., 2018), o número imagens utilizadas foi de apenas 102 imagens e por isso o número de objetos esperados total é consideravelmente menor. Mesmo assim, na presença ou ausência de ninfas de alvídeos obteve um *F-score* de 0,96 em média. Sob essa ótica, o trabalho de (KARUNA et al., 2019)

obteve 93% de acerto e o trabalho de (RUSTIA et al., 2019) 97%. Ambos os resultados próximos da metodologia apresentada neste trabalho.

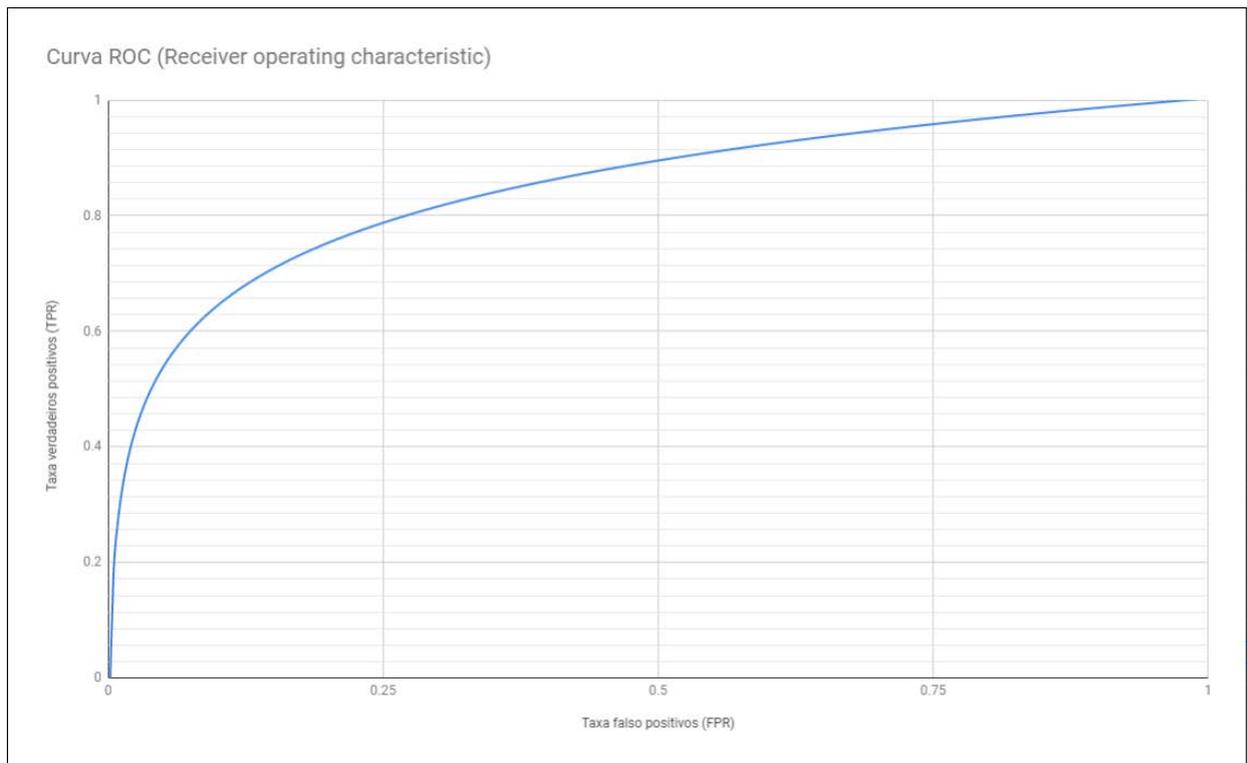


Figura 4.1: Curva ROC do modelo nos dados utilizados (verdadeiros positivos x falso positivos)

Tabela 4.1: Validação cruzada nas 140 imagens de validação.

Test N°	Acurácia média dos objetos (%)	Precisão (%)	<i>Recall</i> (%)	<i>F-score</i> (%)	Imagens
1	83,21	34,16	91,54	49,75	110 118 119 153 155 157 179 188 198 215 220 223 229 230
2	80,50	32,34	94,08	48,13	101 106 122 148 151 194 199 200 211 225 227 231 234 238
3	92,54	29,95	91,07	45,07	120 124 126 132 135 143 146 155 174 190 197 202 227 234
4	85,16	25,95	93,11	40,59	104 116 119 126 133 136 139 143 163 173 180 197 199 208
5	83,66	25,66	89,71	39,91	101 128 133 135 144 150 151 154 160 189 209 221 235 239
6	90,86	60,59	98,03	74,89	107 115 120 125 138 142 161 181 182 207 209 219 226 230
7	86,39	24,95	91,80	39,24	101 106 135 139 143 150 154 160 186 188 200 202 220 226
8	91,77	44,00	94,83	60,11	102 105 109 113 118 129 136 166 169 172 177 188 204 211
9	85,09	30,80	94,53	46,47	105 109 114 125 131 141 143 147 156 164 171 190 204 219
10	83,14	31,17	92,54	46,63	104 120 126 139 150 177 183 185 190 206 210 221 229 236
Média	86,23	33,95	93,12	49,08	

Tabela 4.2: Resultados por classe dos 2360 objetos encontrados nas 140 imagens de validação

Classes	Estimado	Real	Falso positivos	Acurácia	Precisão	<i>Recall</i>	<i>F-measure</i>
Exúvia	395	566	3	76,10%	88,97%	69,25%	81,15%
Ínstar 1	1243	1624	30	74,87%	57,73%	74,84%	84,71%
Ínstar 2	94	105	16	75,55%	98,44%	74,28%	78,39%
Ínstar 3	27	25	6	75,97%	99,77%	84,00%	80,76%
Ínstar 4	33	33	7	75,93%	99,60%	78,78%	78,78%
Adultas	7	7	1	76,18%	99,94%	85,71%	85,71%

Tabela 4.3: Matriz de confusão dos objetos encontrados

Classes:	<i>Exúvia</i>	Instar 1	instar 2	Instar 3	Instar 4	Adultas
Exúvia	388	0	0	4	10	1
Instar 1	0	1213	25	0	0	0
Instar 2	4	5	78	4	0	0
Instar 3	3	0	2	21	0	0
Instar 4	3	0	0	2	26	1
Adultas	1	0	0	0	0	6

Capítulo 5

Considerações finais

Este trabalho cumpriu o objetivo geral de prover uma metodologia para detecção de moscas brancas em folhas de soja em seis diferentes fases de seu desenvolvimento utilizando florestas randômicas e Visão Computacional. Na fase de pré-modelo (antes do ajuste dos hiper-parâmetros), o algoritmo de florestas randômicas obteve os melhores resultados entre os algoritmos testados. Utilizando a metodologia proposta em imagens reais tiradas do campo, o modelo foi capaz de separar pixels ruidosos e objetos espúrios de pixels de interesse. O objetivo geral foi alcançado através da resolução dos objetivos específicos:

- os pixels foram separados pela utilização da modelo de florestas randômicas;
- o Teste de *hold-out* foi realizado nas 100 imagens particionando o dados de treinamento: 70 imagens para treinamento e 30 para teste;
- foi realizada a filtragem de pixels ruidosos para diminuir o erro proveniente do modelo de Aprendizagem de Máquina: Atráves das regras de vizinhança e de determinação da análise de similaridade de cada pixel com o plano de fundo da imagem, eliminando a maior parte do ruído;
- utilizando o algoritmo de janela dinâmica os grupos de pixels foram capturados individualmente para análise e as regras de cor e tamanho para classifica-los;
- utilizando-se 10% das imagens das 140 imagens de validação (14) os vetores da resposta do sistema foram comparados com os vetores da aferição manual dez vezes com o teste de validação-cruzada (*cross-validation 10 fold*).

O problema de pesquisa pôde ser respondido através das confirmações das hipóteses abaixo:

- hipótese 1: por meio da Visão Computacional e Aprendizagem de Máquina é possível identificar a mosca branca. O teste de "hold-out" utilizando florestas randômicas nos dados das 100 imagens consideradas, o modelo conseguiu 95% de acerto na separação de pixels pertencentes ou não a objetos de interesse;

- hipótese 2: por meio da Visão Computacional e Aprendizagem de Máquina é possível classificar as fases de vida da mosca branca (ínstares 1 a 4 e exúvias). O teste de validação-cruzada nas 140 imagens de validação conseguiu corretamente classificar as seis classes procuradas (objetos inteiros) corretamente para 86,23% dos objetos, em média.

A metodologia como foi proposta, foi testada e se mostrou funcional para ser usada nos mais variáveis tons de folhas, sombras e texturas. Contando com os problemas encontrados, como objetos de tamanhos anormais e dispostas em texturas anormais (doenças de folhas) o modelo foi capaz de classificar os objetos por instar. Com a menor classificação de 78,39% para o instar 2 e a melhor 85,71% para moscas adultas.

Como pesquisa futura, mais imagens podem ser consideradas no uso do modelo, a adaptação deste trabalho pra identificação de outras pragas, a pesquisa em diferentes atributos que não foram utilizados neste trabalho e o desenvolvimento de uma aplicação móvel que possa levar essa metodologia até a utilização pelo agricultor no campo.

REFERÊNCIAS BIBLIOGRÁFICAS

BAIG, M. M., DUBEY, A. K., RAMMAMURTHY, V. V., (2015), "Biology and morphology of life stages of three species of whiteflies (Hemiptera: Aleyrodidae) from India". *The Pan-Pacific Entomologist*, V.91, n.2, pp.690-698.

BARBEDO, JAYME GARCIA ARNAL, (2014), "Using digital image processing for counting whiteflies on soybean leaves", *Journal of Asia-Pacific Entomology*, V.17 , pp.685–694.

BISHOP, C. M., (2006), "Pattern Recognition and Machine Learning", Springer Editora, ISBN 978-0-387-31073-2

BREIMAN, L., (2001), "Random Forests." *Machine Learning*, V.45, n.1, pp.5–32.

BROOMHEAD, D. S., LOWE, DAVID., (1988), "Radial basis functions, multi-variable functional interpolation and adaptive networks (Technical report)", *RSRE*, V.4148.

CHAWLA, N. V., BOWYER, K. W, HALL, L. O, KEGELMEYER, W. P., (2002), "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, n.16, pp.321–357.

CHEN, J., FAN, Y., WANG, T., ZHANG, C., QIU, Z., HE, Y., (2018), "Automatic Segmentation and Counting of Aphid Nymphs on Leaves Using Convolutional Neural Networks.", *Agronomy* 2018, V.8, n.129.

CHI, HSIN., (1990), "Timing of Control Based on the Stage Structure of Pest Populations: A Simulation Approach", *Journal of Economic Entomology*, 83, V.4, n.1, pp.1143–1150.

CHO, J., CHOI, J., QIAO, M., JI, C., KIM, H., UHM, K., CHON, T, (2007) "Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis", *International Journal of Mathematics and Computers in Simulation*, V.1, n.1, pp.46-53.

COSTA, T. L., SARMENTO, R. A., ARAÚJO, T. A., PEREIRA, P. S., SILVA, R. S., LOPES, M. C., PIKANÇO, M. C. (2019). "Economic injury levels and sequential sampling plans for *Bemisia tabaci* (Hemiptera: Aleyrodidae) biotype B on open-field melon crops". *Crop Protection*, N.104887.

DANA, H., et al., (1982), "Computer Vision", *Prentice Hall* Editora, ISBN 978-0-13-165316-0.

DE BARRO, PAUL J., LIU, SHU-SHENG., BOYKIN, LAURA M., DINSDALE, ADAM B., (2015), "*Bemisia tabaci*: a statement of species status", *Annual Review of Entomology*, V.56, pp.1–19

EHLER, LESTER E., "Perspective Integrated pest management (IPM): definition, historical development and implementation and the other IPM", (2006), *Pest Manag Sci*, V.62, pp.787-789.

ESPINOZA, K., VALERA, DIEGO L., TORRES, JOSÉ A., LÓPEZ, A., MOLINA-AIZ, FRANCISCO D. "Combination of image processing and artificial neural networks as a novel approach for the identification of Bemisia tabaci and Frankliniella occidentalis on sticky traps in greenhouse agriculture", *Computers and Electronics in Agriculture*, 127, pp.495–505.

FORGY, E. W., (1965), "Cluster analysis of multivariate data: efficiency versus interpretability of classifications", *Biometrics*, V.21, N.3, pp 768-769.

FRANK, E., MARK, A., IAN, H., (2016), "The WEKA Workbench.", Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", *Morgan Kaufmann*, V.4, Disponível em: <<http://prdownloads.sourceforge.net/weka/weka-3-8-3jre-x64.exe>> Acesso em: 9 de Novembro de 2019.

HO, TIN KAM, (1995), "Random Decision Forests." *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, pp.278–282.

HOROWITZ, A. R., ISHAAYA I., (2016), "Advances in Insect Control and Resistance Management", *Springer International Publishing Switzerland*, N.11, pp 205-218.

ILLINGWORTH, J., KITTLER, J., (1987), "The Adaptive Hough Transform," *PAMI-9* ,5, pp.690-698.

KARUNA, M., et al., (2019), "Early Detection of Chili Plant Leaf Diseases using Machine Learning", *International Journal of Engineering Science and Computing*, V.9, N.5, pp 22328-22335.

KOHAVI, RON, (1995), "A study of cross-validation and bootstrap for accuracy estimation and model selection", *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, CA, Morgan Kaufmann, V.2, n.12, pp.1137–1143.

LIU, T., CHEN, W., WU, W., SUN, C., GUO, W., ZHU, X., (2016), "Detection of aphids in wheat fields using a computer vision technique", *Biosystems Engineering*, V.141, pp.82-93.

MORALES, F. J., ANDERSON, P. K., (2011), "The emergence and dissemination of whitefly-transmitted geminiviruses in Latin America", *Archives of Virology*, V.146, n.3, pp.415–441.

QUINLAN, J. R., (1993), "C4.5: Programs for Machine Learning.", *Morgan Kaufmann Publishers*.

RISH, IRINA, (2001), "An empirical study of the naive Bayes classifier.", *IJCAI Workshop on Empirical Methods in AI*.

RUSTIA, DAN J. A. R., et al., (2019), "An Online Unsupervised Deep Learning Approach for an Automated Pest Insect Monitoring System", *ASABE 2019 Annual International Meeting*, N.477, pp 1-10.

SAVARY, SERGE, et al., (2019), "The global burden of pathogens and pests on major food crops.", *Nature Ecology Evolution*, V.3, pp.430–439.

SCOTT, MAXWELL J., et al., (2018), "Agricultural production: assessment of the potential use of Cas9-mediated gene drive systems for agricultural pest control", *Journal of Responsible Innovation*, V.5, S98-S120.

SHALEV-SHWARTZ, SHAI; BEN-DAVID, SHAI, (2014), *Understanding Machine Learning*, Cambridge University Press Editora.

SPARKS, T. C., NAUEN R., (2015), "IRAC: Mode of action classification and insecticide resistance management.", *Pesticide Biochemistry and Physiology*, V.121, pp.122–128.

STUART, A.; ORD, K., (1994), "*Kendall's Advanced Theory of Statistics: Volume I—Distribution Theory*", 5 edição, Griffin Editora, ISBN 0 85264 285 7, 604 p.

VANRELL, MARIA et al., (2001), "*Colour normalisation based on background information*", *Image Processing*, V.1, pp 874-877, doi:10.1109/icip.2001.959185.

ANEXOS

I. ANEXO 1



Figura I.1: Imagem fonte (*Source*).

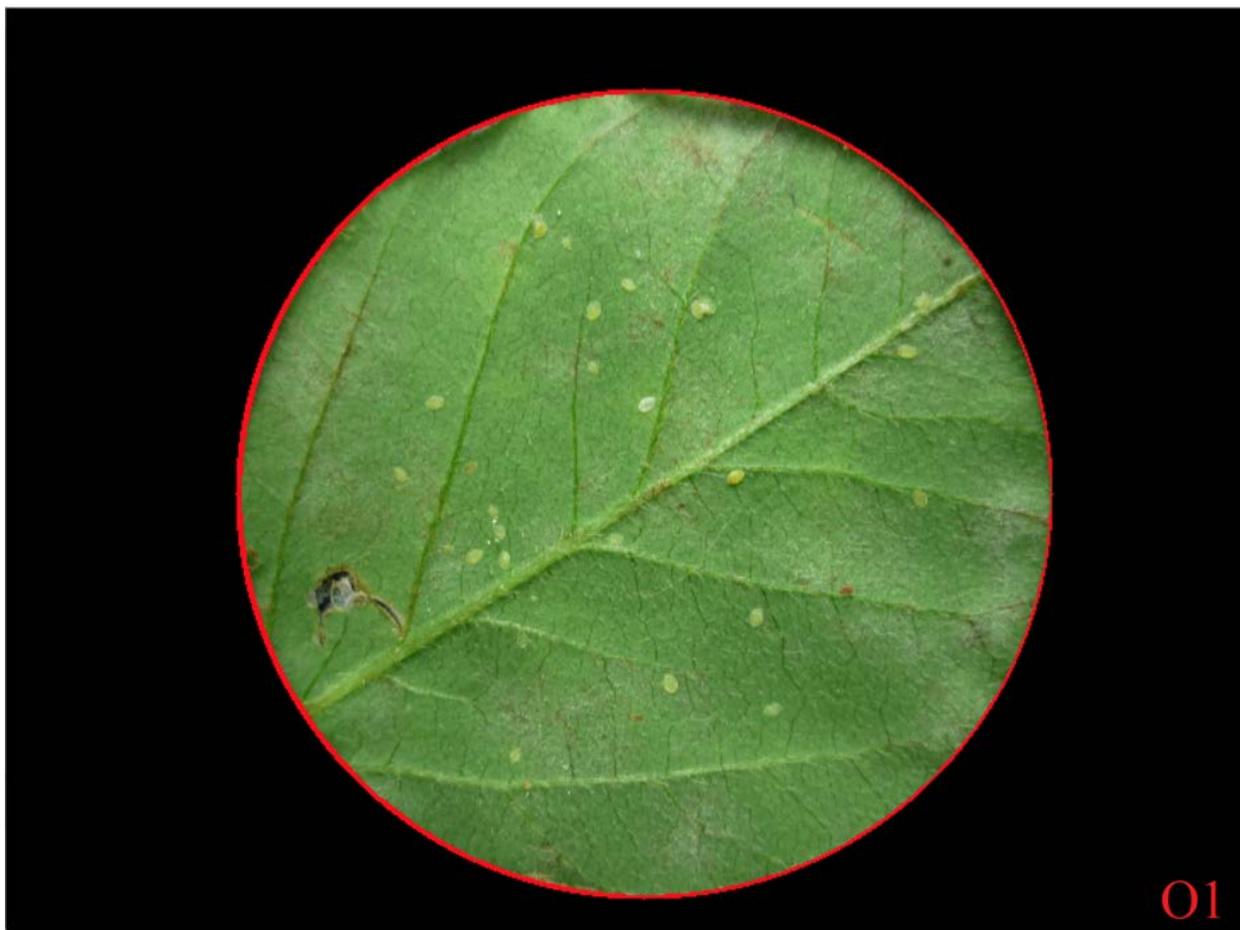


Figura I.2: Imagem fonte após o uso da transformada circular de Hough (*Output 1*).

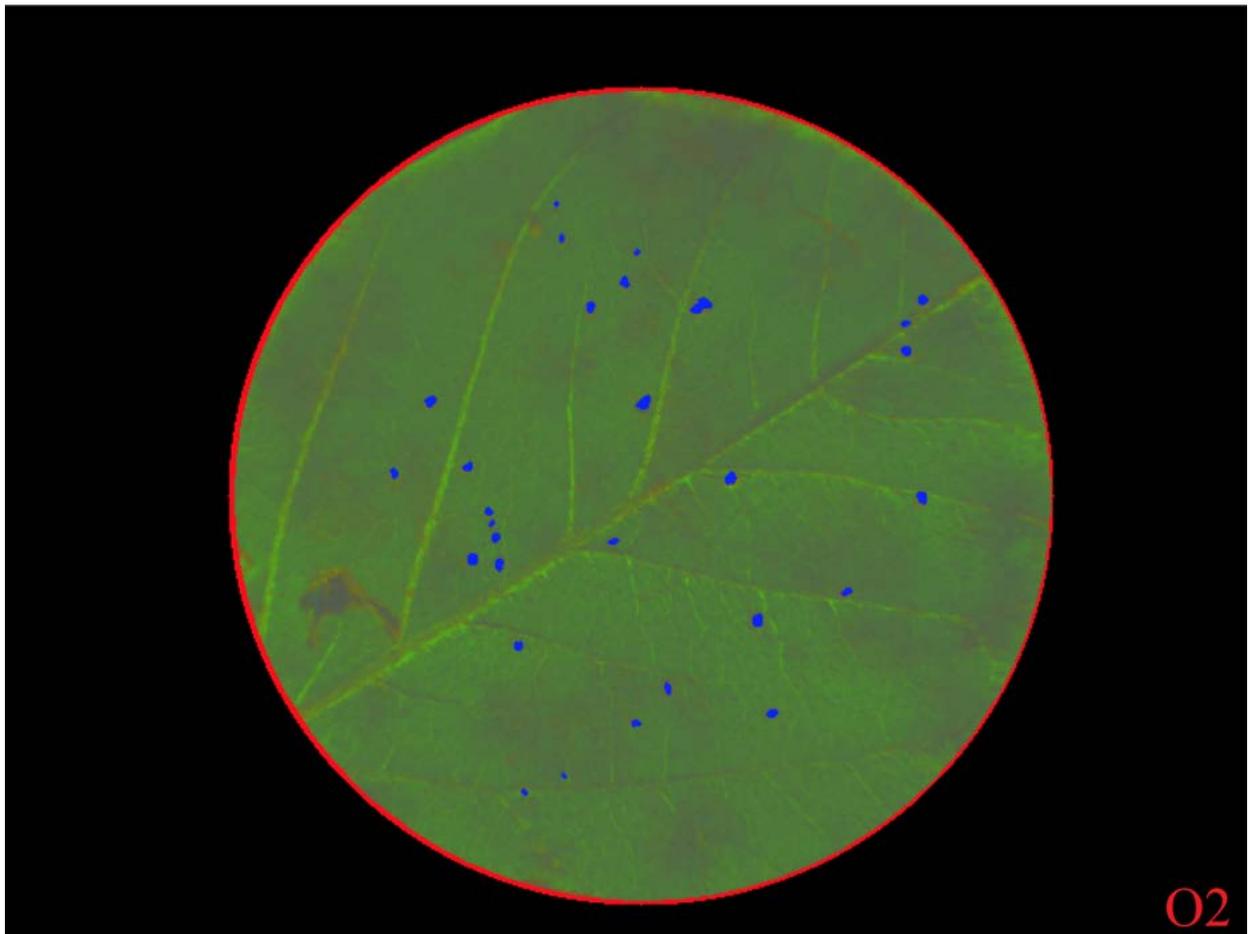


Figura I.3: Imagem fonte após o preenchimento humanamente supervisionado (*Output 2*).

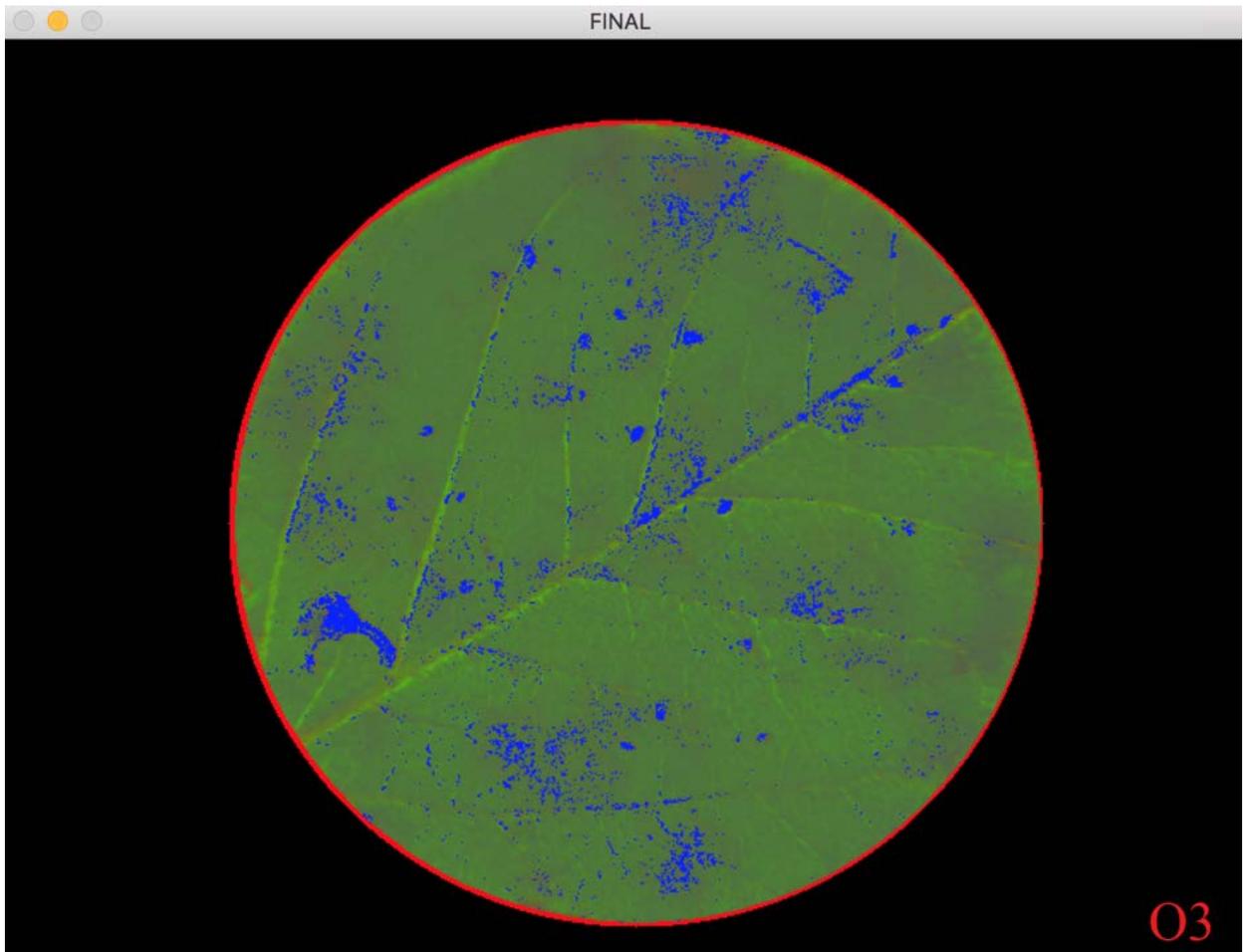


Figura I.4: Imagem fonte após o preenchimento automático (*Output 3*).

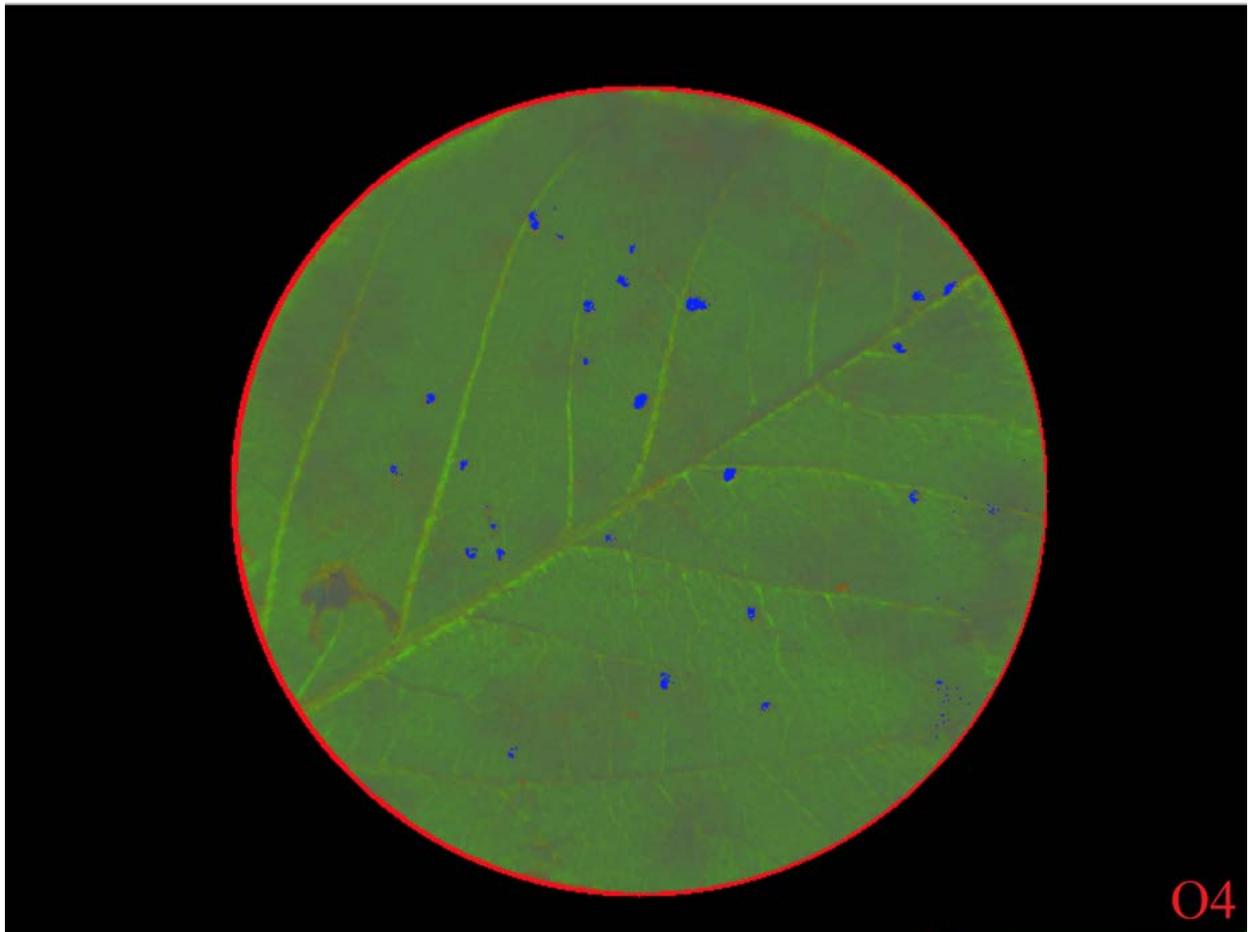


Figura I.5: Imagem anterior após a filtragem de ruído (*Output 4*).

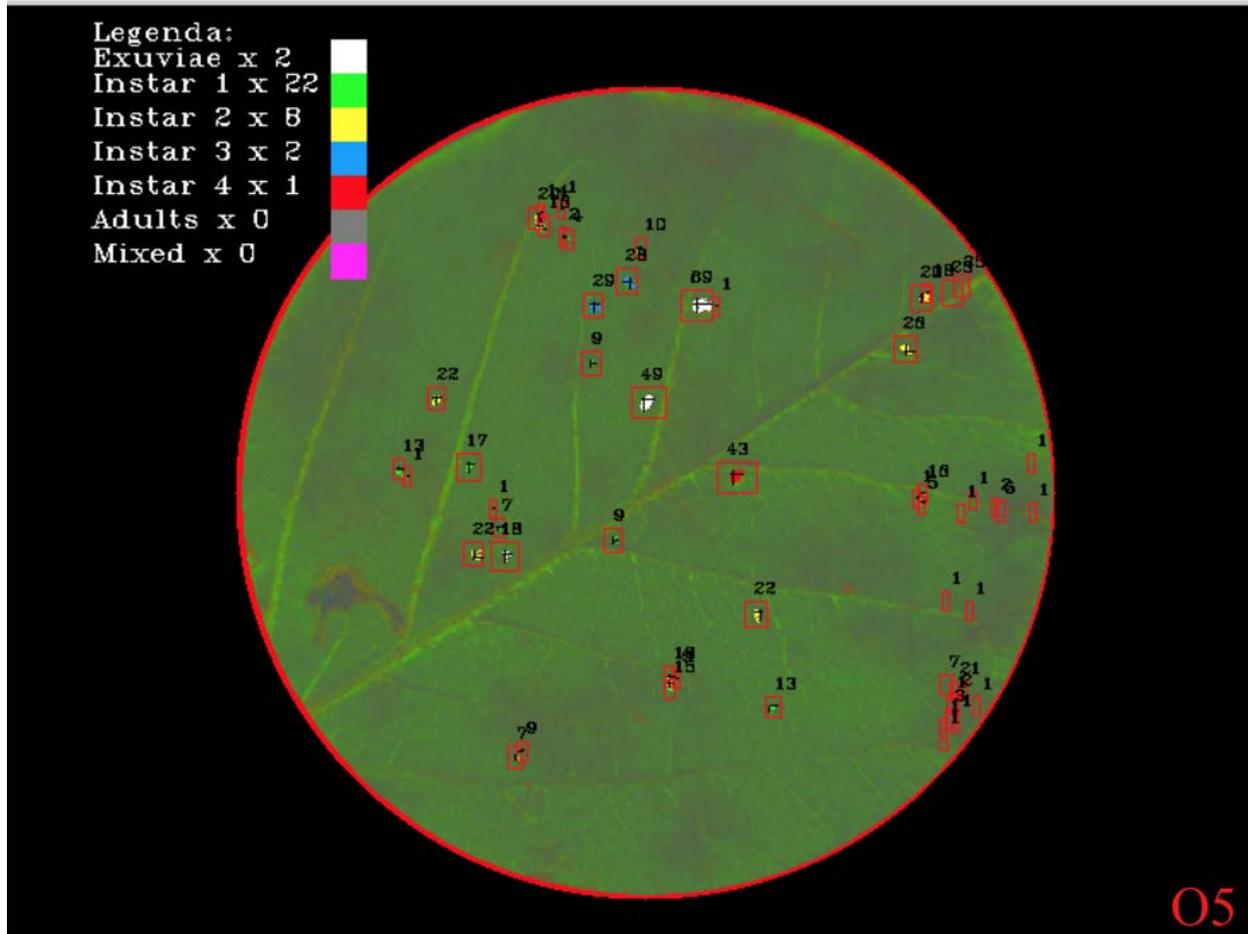


Figura I.6: Imagem anterior após a classificação dos objetos (*Output 5*).

