

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

UMA METODOLOGIA DE CONTAGEM E
CLASSIFICAÇÃO DE AFÍDEOS
UTILIZANDO VISÃO COMPUTACIONAL

Elison Alfeu Lins

Passo Fundo
2018

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**UMA METODOLOGIA DE CONTAGEM E CLASSIFICAÇÃO
DE AFÍDEOS UTILIZANDO VISÃO COMPUTACIONAL**

Elison Alfeu Lins

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Computação
Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Rafael Rieder

Passo Fundo
2018

CIP – Catalogação na Publicação

- L759m Lins, Alison Alfeu
Uma metodologia de contagem e classificação de
afídeos utilizando visão computacional / Alison Alfeu Lins.
– 2018.
81 f. : il. color. ; 30 cm.
- Orientador: Prof. Dr. Rafael Rieder.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2018.
1. Computação. 2. Pulgões. 3. Informática na agricultura.
4. Pragas agrícolas - Controle. I. Rieder, Rafael, orientador.
II. Título.
- CDU: 631:004
004:631

**ATA DE DEFESA DO
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**

ELISON ALFEU LINS

Aos vinte e sete dias do mês de março do ano de dois mil e dezoito, às 14 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso **“Uma metodologia de contagem e classificação de afídeos utilizando Visão Computacional”**, de autoria de Elison Alfeu Lins, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Rafael Rieder, José Maurício Cunha Fernandes e Douglas Lau. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



Prof. Dr. Rafael Rieder - UPF
Presidente da Banca Examinadora
(Orientador)



Prof. Dr. José Maurício Cunha Fernandes - UPF
(Avaliador Interno)



Dr. Douglas Lau – Embrapa Trigo
(Avaliador Externo)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA

Dedico este trabalho aos meus pais,
Lenita Ana Bristot Lins e
Edson Luiz Prestes Lins.

AGRADECIMENTOS

Em primeiro lugar, agradeço a minha família pelo incentivo durante esta jornada.

Agradeço em especial o meu orientador Dr. Rafael Rieder por ter acreditado em mim no processo de seleção. Pelo tempo dedicado, por todos os ensinamentos durante as orientações, pelos incentivos, pelas cobranças. Levarei sem dúvidas isso para minha vida profissional e pessoal.

Agradeço imensamente a Universidade de Passo Fundo pela bolsa de estudos, sem a qual não teria condições de estar realizando este projeto.

Agradeço a instituição Embrapa Trigo, parte fundamental deste trabalho. Em especial aos colaboradores: Dr. Douglas Lau, Dr. Paulo Roberto Valle da Silva Pereira, Dr. José Maurício Cunha Fernandes e Marcio Nicolau.

Aos meus colegas do PPGCA, SIN e do MOSAICO, pelo conhecimento trocado durante estes anos.

Ao amigo Ronaldo Serpa da Rosa pelo incentivo para começar esta caminhada.

A Compasso Tecnologia pelo apoio flexibilizando meus horários para que eu pudesse estudar.

Meu muitíssimo obrigado a todos!

Uma imagem vale mais que mil palavras.

Confúcio

UMA METODOLOGIA DE CONTAGEM E CLASSIFICAÇÃO DE AFÍDEOS UTILIZANDO VISÃO COMPUTACIONAL

RESUMO

Pulgões ou afídeos são insetos que atacam culturas e causam danos de forma direta, consumindo a seiva das plantas, e de forma indireta, transmitindo doenças. A contagem e a classificação desses insetos são fundamentais para mensurar e prever riscos às lavouras, e servir de base para a aplicação ou não de químicos, também para realizar testes de resistência de cultivares. Tradicionalmente, o processo de contagem é realizado de forma manual, e depende de microscópios e visão apurada do especialista, em tarefa demorada e suscetível a erros. Para tanto, este trabalho apresenta uma metodologia e um software para automatizar a contagem, a classificação e a mensuração de afídeos da espécie *Rhopalosiphum padi* utilizando métodos de processamento de imagens, visão computacional e *deep learning*. O texto apresenta também uma análise do sistema proposto, utilizando 40 amostras, cujas contagens foram feitas manualmente e comparadas com os valores obtidos no software. Como resultado, obteve-se um alto grau de correlação nas etapas de contagem e classificação ($R = 0,92579$) e de mensuração ($R = 0,9799$).

Palavras-chave: contagem, classificação, opencv, pulgões, tensorflow, visão computacional.

METHODOLOGY FOR COUNTING AND CLASSIFYING APHIDS USING COMPUTER VISION

ABSTRACT

Aphids are insects that attack crops and cause damage directly, consuming the sap of plants, and indirectly, transmitting diseases. The counting and classification of these insects are fundamental for measuring and predicting crop hazards and serving as the basis for the application or not of chemicals, also, to test plant resistance. Traditionally, the counting process is manual, and depends of microscopes and good eyesight of the specialist, in a time-consuming task susceptible to errors. With this in mind, this paper presents a methodology and a software to automate the counting and classification of aphids of specie *Rhopalosiphum padi* using image processing, computer vision and deep learning methods. The text also presents a system analysis, using 40 samples, comparing manually counts and values obtained with software. As a result, we obtain a strong positive correlation in count and classification stage ($R = 0,92579$) and measurement stage ($R = 0,9799$).

Keywords: aphids, classification, counting, computer vision, opencv, tensorflow.

LISTA DE FIGURAS

Figura 1 - Espectro eletromagnético organizado de acordo com a energia por fóton [8].	20
Figura 2 – Produtos manufaturados frequentemente verificados com técnicas de PDI [8].	21
Figura 3 – Para o computador, o espelho do carro é uma matriz de números [10].	22
Figura 4 – Exemplo de processo de aquisição de uma imagem digital [8].	23
Figura 5 – Cubo RGB.	24
Figura 6 – Efeitos da redução da resolução espacial [8].	25
Figura 7 – Exemplo de restauração de imagem [8].	27
Figura 8 – Exemplo de compressão de imagem [8].	28
Figura 9 – Exemplo de segmentação de imagem [8].	29
Figura 10 – Exemplo de extração de características de uma imagem [8].	30
Figura 11 – Exemplo de limiarização.	32
Figura 12 – Máscaras de detecção de linhas [8].	32
Figura 13 – Imagem original (a) e imagem processada com filtro laplaciano (b).	33
Figura 14 – Representação de bordas [8].	33
Figura 15 – Exemplo de bordas detectadas.	34
Figura 16 – Exemplos de aplicação de visão computacional [13].	36
Figura 17 – Comparação entre <i>Machine Learning</i> e <i>Deep Learning</i> [15].	37
Figura 18 – Exemplo de processamento de imagem usando ImageJ [19].	39
Figura 19 - Etapas de processamento [7].	42
Figura 20 – Resultado da aplicação de MSER [23].	43
Figura 21 – Desmonstração de funcionamento do processo de identificação [24].	44
Figura 22 – Processo de captura e imagens para treinamento da rede neural [25].	46
Figura 23 – Exemplo de imagem disponibilizada.	48
Figura 24 – Sequência metodológica.	48
Figura 25 – Caixa de vedação de luz.	49
Figura 26 – Comparação do método sem e com uso de uma caixa de vedação.	50
Figura 27 – Comparação em zoom, mostrando detalhes do indivíduo alado quando utilizada a caixa de vedação.	50
Figura 28 – Referencias de medida para elementos da imagem.	50
Figura 29 – Resultado pós processamento.	51
Figura 30 – Resultado da limiarização.	52

Figura 31 – Exemplo de imagem utilizada para extração de amostras	54
Figura 32 – Exemplos de imagens da classe Alados.	55
Figura 33 – Exemplos de imagens da classe Ápteros.....	55
Figura 34 – Exemplos de imagens da classe Ninfas.....	55
Figura 35 – Exemplos de imagens da classe Falsos.	56
Figura 36 - Apresentação dos resultados.....	56
Figura 37 – Tela de seleção de pasta de origem.	58
Figura 38 – Tela inicial.....	59
Figura 39 - Tela de configurações	60
Figura 40 – Detalhes da imagem processada.....	61
Figura 41 – Exemplo de resultado.....	61
Figura 42- Resultados da análise.....	62
Figura 43 – Exemplo de relatório por placa	62
Figura 44 – Valor total Manual x CropAphid.	66
Figura 45 – Contagem de “ninfas” Manual x CropAphid.	66
Figura 46 – Contagem de “ápteros” Manual x CropAphid.	67
Figura 47 – Contagem de “alados” Manual x CropAphid.	67
Figura 48 – Valor total Manual x AphidCV.....	68
Figura 49 – Contagem de “ninfas” Manual x AphidCV.	69
Figura 50 – Contagem de “ápteros” Manual x AphidCV.....	69
Figura 51 – Contagem de “alados” Manual x AphidCV.....	70
Figura 52 – Valor total CropAphid x AphidCV.....	71
Figura 53 – Contagem de “ninfas” CropAphid x AphidCV.....	71
Figura 54 – Contagem de “ápteros” CropAphid x AphidCV.....	72
Figura 55 – Contagem de “alados” CropAphid x AphidCV.	72
Figura 56 – Comparação de mensurações	73
Figura 57 – Comparação de tempo.....	74

LISTA DE TABELAS

Tabela 1 – Exemplo de implementação de limiarização em OpenCV	38
---	----

SUMÁRIO

1. INTRODUÇÃO	15
2. FUNDAMENTAÇÃO TEÓRICA.....	19
2.1. PROCESSAMENTO DIGITAL DE IMAGENS	19
2.2. IMAGENS DIGITAIS.....	22
2.3. PASSOS FUNDAMENTAIS EM PROCESSAMENTO DE IMAGENS	25
2.3.1. Realce de imagens.....	26
2.3.2. Restauração.....	26
2.3.3. Compressão	27
2.3.4. Segmentação.....	28
2.3.5. Extração de características	29
2.3.6. Reconhecimento.....	30
2.4. TÉCNICAS DE PROCESSAMENTO	31
2.4.1. Limiarização ou binarização	31
2.4.2. Detecção de linhas.....	32
2.4.3. Detecção de bordas	33
2.5. VISÃO COMPUTACIONAL.....	34
2.6. DEEP LEARNING	36
2.7. BIBLIOTECAS DE PROCESSAMENTO DE IMAGENS E DEEP LEARNING.....	37
2.7.1. OpenCV	38
2.7.2. ImageJ.....	39
2.7.3. TensorFlow	39
2.7.4. Caffe.....	40
3. TRABALHOS RELACIONADOS	41
3.1. USING DIGITAL IMAGE PROCESSING FOR COUNTING WHITEFLIES ON SOYBEAN LEAVES	41
3.2. DETECTION OF APHIDS IN WHEAT FIELDS USING A COMPUTER VISION TECHNIQUE	42
3.3. AUTOMATIC IDENTIFICATION AND COUNTING OF SMALL SIZE PESTS IN GREENHOUSE CONDITIONS WITH LOW COMPUTATIONAL COST	43
3.4. AGRICULTURE PEST CONTROL USING COMPUTER VISION TECHNIQUE.....	45
4. MATERIAIS E MÉTODOS.....	47
4.1. ÁREA DE ESTUDO	47
4.2. METODOLOGIA PROPOSTA	47

4.2.1.	Passo 1: Coleta de Imagens.....	48
4.2.2.	Passo 2: Segmentação macro	51
4.2.3.	Passo 3: Segmentação de pequenos elementos	51
4.2.4.	Passo 4: Reconhecimento de contornos	52
4.2.5.	Passo 5: Identificação de potenciais contornos	52
4.2.6.	Passo 6: Classificação	53
4.2.7.	Passo 7: Apresentação dos resultados.....	56
4.3.	FERRAMENTAS	57
5.	APHIDCV	58
6.	RESULTADOS E DISCUSSÕES	65
6.1.	MÉTODO MANUAL X CROPAPHID	65
6.2.	MÉTODO MANUAL X APHIDCV	68
6.3.	CROPAHPID X APHIDCV	70
6.4.	COMPARAÇÃO DE MENSURAÇÃO	73
6.5.	DISCUSSÃO	73
7.	CONCLUSÃO.....	77
7.1.	TRABALHOS FUTUROS	78
	REFERÊNCIAS	79

1. INTRODUÇÃO

Segundo Gallo [1], os insetos são considerados nossos mais sérios competidores pela hegemonia do planeta. Dependemos deles para a polinização de plantas utilizadas como base da nossa alimentação e também para diversas funções ecológicas, como na reciclagem de nutrientes e na base alimentar de outras espécies. Porém, o foco das preocupações com estes seres vivos são os impactos negativos quando assumem o papel de pragas e devastam plantações, gerando impactos financeiros e diminuição da produção de alimentos. A Organização Mundial para Alimentação e Agricultura (FAO) estima que as pragas sejam responsáveis por 42,1% das perdas na produção sendo que 15,6% deste total é causado por insetos e ácaros [2].

Entre as culturas afetadas pelos insetos está o trigo, o segundo cereal mais produzido no mundo, com significativo peso na economia agrícola mundial [3]. Por ser uma cultura de inverno, várias espécies de afídeos ou pulgões estão entre as principais pragas do trigo. Os pulgões são insetos que se alimentam da seiva dos cereais de inverno, como o trigo, a cevada e a aveia, desde a emergência até que os grãos estejam completamente formados.

Essa praga surgiu no Brasil na década de 1970 quando houve a expansão da cultura do trigo. Vindo da Europa e da Ásia, os pulgões chegaram aqui livres de seus inimigos naturais, o que resultou em uma explosão populacional, e tornou a cultura altamente dependente do uso de agroquímicos [4].

O pulgão pode causar danos diretos quando suga a seiva da planta ou quando transmite o vírus do nanismo amarelo da cevada (VNAC) [1]. Em qualquer uma destas situações eles são responsáveis por expressivas quedas na produção, com o enfraquecimento e morte da planta, sendo que desde meados de 1970 o VNAC vem causando uma queda de aproximadamente 20% na produção sulista de trigo [1].

Neste contexto, considerando que pequenas populações conseguem transmitir a doença e se multiplicar rapidamente, o controle deve iniciar imediatamente quando forem encontradas 10% de plantas atacadas por pulgões ou quando houver 10 pulgões por espiga [1]. Sendo assim, a contagem dos insetos é extremamente necessária para que os pesquisadores possam mensurar se existe de fato um risco as lavouras [5]. Esta análise deve ser realizada mediante avaliações semanais em locais aleatórios da lavoura de modo a obter um resultado médio da densidade de insetos [6]. Outro uso da contagem e classificação são

em testes de resistência de plantas, cujo objetivo é estimar qual espécie de cultivar promoveu um crescimento maior da população de afídeos.

Para realizar esta contagem, várias amostras de campo são coletadas para análise em laboratório sobre placas de Petri¹. Uma das metodologias tradicionalmente empregadas é o uso de uma lente de aumento e uma grade quadricular de fundo, para visualização e contagem manual (ou por amostragem) dos insetos. Devido ao tamanho dos insetos (aproximadamente um milímetro) e a quantidade, o processo é extremamente demorado considerando que várias destas amostras são coletadas por área de plantio. Em muitos casos, é necessário que exista a repetição de contagem por ensaio, tornando o trabalho penoso e desgastante à pessoa responsável. Quando existe uma grande quantidade de insetos na placa, os pesquisadores realizam a contagem de um quadrante, e estimam o total a partir deste valor.

Além da contagem, para que seja possível determinar o estágio de infestação, cada indivíduo deve ser identificado quanto as suas características corporais e classificado como: ninfa, adulto alado e adulto áptero. As ninfas são insetos em sua fase inicial de vida. Já os alados são fêmeas adultas munidas de asas e possuem a capacidade de disseminação da doença para outros ambientes. Os ápteros² são indivíduos adultos sem asas, responsáveis pela reprodução no mesmo local. Logo, o trabalho de classificação contribui para que o processo seja ainda mais demorado.

Outro problema encontrado no processo de contagem, segundo Barbedo [7], é que os humanos são extremamente sensíveis a fenômenos fisiológicos e psicológicos, como fadiga, ilusões visuais e tédio. Esses ou outros fatores associados podem influenciar nos resultados da contagem.

Neste contexto, analisando o problema dos pulgões para as lavouras de trigo, considerando que baixíssimas populações conseguem se multiplicar rapidamente e disseminar doenças, é importante saber que somente uma estratégia de controle provavelmente não seja suficiente para combater a praga.

O manejo integrado de pragas (MIP) preconiza a convivência com os insetos até que o limiar de dano econômico seja atingido. Isso significa permitir o máximo de controle natural através de inimigos naturais e oscilações climáticas, levando também em consideração a resistência das plantas [6].

¹ Recipiente cilíndrico, achatado, de vidro ou plástico, que os profissionais de laboratório utilizam para a cultura de microrganismos.

² Fêmeas sem asas encarregadas da reprodução dentro de uma mesma colônia.

A implementação prática do MIP depende do conhecimento de uma série de dados e de parâmetros em que se baseiam as decisões para evitar prejuízos por pragas [6]. Assim, uma das bases do MIP é o monitoramento das principais espécies de pragas e também de seus inimigos naturais através da coleta e análise de amostragens periódicas. A diminuição do tempo de análise pode permitir aos pesquisadores que mais amostras sejam coletadas e analisadas por região, sendo possível obter dados mais precisos da infestação.

Além disto, através de sistemas computacionais e diretivas de coleta de imagens, é possível estabelecer um padrão de análise. Um método padrão de contagem e classificação pode ser definido em um algoritmo que siga os procedimentos científicos e a expertise do pesquisador. O algoritmo pode também integrar esse conhecimento a técnicas de Inteligência Artificial que permitam ao software aprender a partir de sua base de dados, podendo aperfeiçoar seu desempenho na tarefa. O software não é suscetível a erros causados por fenômenos fisiológicos e psicológicos humanos, como fadiga e tédio – diferente do atual processo onde vários pesquisadores são responsáveis pelas análises, com tendência a variações de resultados.

O aumento do volume de dados, mais precisos e capturados de forma mais rápida, permitem que os tomadores de decisão cheguem a conclusões mais assertivas. A aplicação ou não de tratamentos depende da decisão destas pessoas, definindo inclusive quando o uso de químicos deve ser desconsiderado ou reduzido. Assim, pode-se contribuir para a redução de custos, o equilíbrio ambiental e a qualidade da cultura.

A extração de informações por meio de recursos de visão computacional pode também auxiliar no MIP e na tomada de decisão na Agricultura – em especial, a cultura do trigo. Com o aumento na velocidade de processamento e análise computacional, um número maior de amostras pode ser verificado, aumentando a confiabilidade dos resultados por meio da automação de processos. Assim, o pesquisador pode realizar outras tarefas de estudo técnico-científico sobre a cultura, os produtores podem aplicar somente a quantidade necessária de pesticidas, entre outros benefícios.

Nesse contexto, o objetivo desse trabalho é apresentar o desenvolvimento de uma aplicação para contagem, classificação e mensuração de afídeos utilizando recursos de visão computacional. O software desenvolvido, intitulado AphidCV, é destinado ao uso de pesquisadores da área agrícola, entomólogos e biólogos, e está inicialmente adaptado para identificar indivíduos da espécie *Rhopalosiphum padi* (Linnaeus, 1758).

Para tanto, esse documento está assim organizado: o Capítulo 2 apresenta fundamentação teórica sobre o estado da arte; o Capítulo 3 descreve os trabalhos

relacionados; o Capítulo 4 mostra os materiais e métodos aplicados nesse estudo; o Capítulo 5 apresenta o software desenvolvido; o Capítulo 6 mostra os resultados e discussões; e o Capítulo 7 realiza a conclusão e aponta os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Este Capítulo apresenta os principais conceitos, definições e ferramentas para Processamento de Imagens, visão computacional e *Deep Learning*, utilizados no decorrer do desenvolvimento desse trabalho.

2.1. PROCESSAMENTO DIGITAL DE IMAGENS

De acordo com Gonzalez [8], o campo de processamento digital de imagens (PDI) se refere ao processamento de imagens digitais por um computador digital. O procedimento consiste em um processo cuja entrada é obrigatoriamente uma imagem e a saída é uma imagem ou dados extraídos. Sua função principal é realçar, extrair ou melhorar a visualização de determinados elementos de uma imagem bruta, para que posteriormente seja analisada por uma pessoa, assim facilitando a identificação de padrões que antes do processamento não eram explicitamente demonstrados.

Dentro de processamento de imagens existem subdivisões cada qual possuindo objetivos distintos. Porém não existem limites claros do início e fim de cada área. Segundo Gonzalez [8], o fato de processar uma imagem para extrair apenas o valor da intensidade média poderia segundo alguns paradigmas não considerar este procedimento como PDI, partindo da definição que PDI é um processo cuja entrada e saída deve ser uma imagem. Conforme Gonzalez [8] um paradigma útil seria levar em consideração três tipos de processos computacionais: processos de nível baixo, médio e alto. Os processos de nível baixo envolvem operações primitivas, como o pré-processamento de imagens para redução de ruídos e realce de imagem, ou seja, processos para melhoria do aspecto da imagem, onde a entrada e a saída são imagens. Já o processamento de imagens de nível médio tem por finalidade a separação da imagem em regiões ou objetos, melhorando assim as partes de interesse para que estes objetos possam ser reconhecidos. Este processo computacional é caracterizado por ter uma imagem como entrada e atributos extraídos como saída, estes podem ser bordas ou contornos de objetos. O processo de nível alto realiza funções cognitivas associadas a visão, basicamente resume-se em dar sentido a um conjunto de objetos.

Para exemplificar os níveis de processamentos de imagens, podemos pensar em um software de processamento de imagens, cujo objetivo é identificar um texto de uma página branca. Após a captura das imagens, em níveis baixos de processamento, teremos o

tratamento que buscará melhorar o aspecto das figuras, um aumento de contraste pode ser aplicado para destacar o texto. Para este caso, um nível médio de processamento seria identificar os elementos de interesse da imagem. As letras poderão ter suas bordas destacadas ou cada caractere recortado e transformado em uma microimagem. O nível alto tentará dar sentido a imagem, transformando cada um dos pequenos objetos em letras. O resultado do processamento é a união de cada identificação isolada em um texto.

É possível também definir PDI em duas áreas de aplicação distintas. A primeira consiste em processamento de imagens destinados a interpretação humana, cujo conceito se encaixa ao nível baixo de processamento de imagens. A segunda e principal área é a resolução de problemas relacionados a percepção por máquinas, cujo objetivo é a extração de informações através de imagens.

A justificativa da utilidade e importância de PDI é o fato do olho humano ter incríveis capacidades de reconhecer padrões, porém não possuir a capacidade de processar um alto número de informações, ou ver diferentes faixas do espectro eletromagnético. Podemos tomar como exemplo um processo de análise de imagens de satélite, cujo objetivo é identificar áreas desmatadas baseando-se em imagens do ano anterior. Apesar do olho humano ser capaz de perceber as diferenças entre as duas imagens, o processo seria extremamente demorado se tratando de uma área territorial de um país.

Diferentemente dos olhos humanos, limitados a banda visual do espectro eletromagnético (EM) (Figura 1), os aparelhos de processamento de imagens são capazes de cobrir uma área maior de EM que variam de ondas gama até ondas de rádio [8]. Além disso, as máquinas processam informações com muito mais rapidez na grande maioria dos casos que envolvem processamento de imagens. Através disso é possível utilizar o processamento digital de imagens em uma ampla quantidade de aplicações.

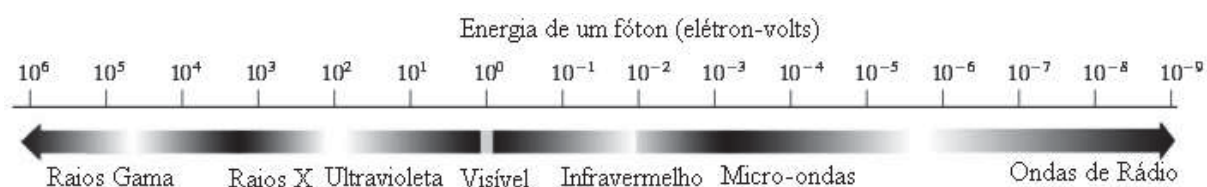


Figura 1 - Espectro eletromagnético organizado de acordo com a energia por fóton [8].

Segundo Gonzalez [8], atualmente não existem áreas de empreendimento técnico que não sejam de alguma forma impactadas pelo processamento digital de imagens. Desde 1960 até os dias atuais a área de processamento digital de imagens cresceu rapidamente em diversos setores. Procedimentos computacionais de processamento de imagens destinados a

interpretação humana, por exemplo, são usados na medicina para realçar o contraste ou para codificar os níveis de intensidade em cores para facilitar a interpretação de médicos nas imagens de radiografias. Geógrafos utilizam PDI em imagens de satélite para estudar padrões de poluição. Arqueólogos, através de procedimentos de realce e restauração, tem melhorado a qualidade de imagens borradas cujos registros eram únicos de objetos raros que foram perdidos ou danificados. Na física PDI é utilizado para realçar imagens de experimentos em áreas como plasmas de alta energia e microscopia eletrônica. No outro campo de processamento digital de imagens relacionado a percepção por máquinas, cujo objetivo é extrair informações das imagens através de processamento computacional, existem inúmeras aplicações. Nesta área destacam-se as aplicações de visão computacional industrial demonstrados na Figura 2, onde muitos dispositivos fazem inspeção de produtos utilizando PDI. Na imagem (a) um circuito impresso, na imagem (b) uma cartela de remédios, garrafas em uma linha de produção (c), bolhas de ar no plástico (d), análise de impurezas no cereal (e) e um implante intraocular (f). Também dentro desta área do PDI está o reconhecimento de textos que é utilizado no monitoramento do tráfego, análise de amostras de sangue, processamento computacional de imagens aéreas ou satélites para previsão do tempo, etc.

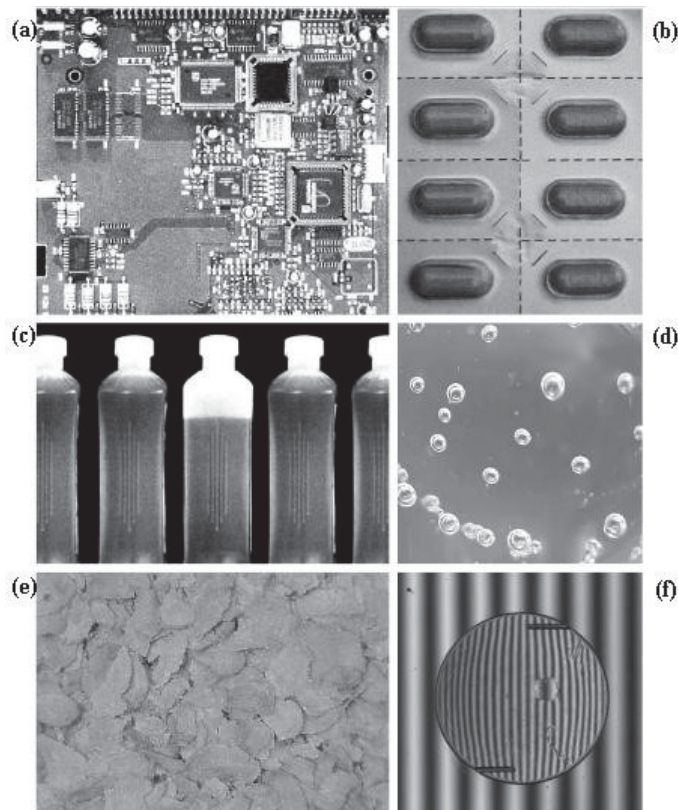


Figura 2 – Produtos manufacturados frequentemente verificados com técnicas de PDI [8].

2.2. IMAGENS DIGITAIS

A luz é um tipo particular de radiação eletromagnética que pode ser percebida pelo olho humano. Como Newton demonstrou em seus experimentos, um feixe de luz solar após atravessar um prisma, consiste em um espectro contínuo de cores. A diferenciação destas cores é baseada nos diferentes comprimentos de onda. Estas diferenças de onda são capturadas pelos olhos humanos e interpretadas pelo cérebro em cores [9]. Já as imagens digitais, são compreendidas pelo computador, através de uma matriz de duas dimensões compostas por números. A Figura 3 demonstra o espelho do carro como o computador vê, para os humanos isto é pouca informação, mas a matriz de números apresentada é um exemplo de dado que o computador interpreta como imagem [10].

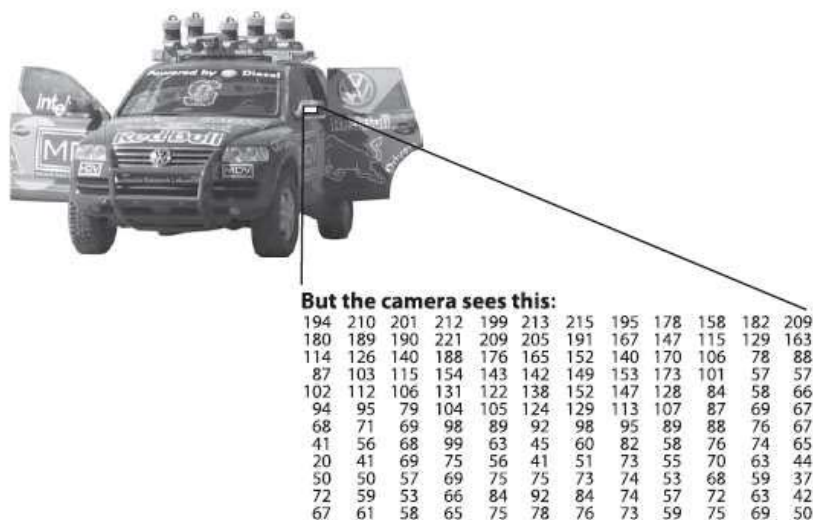


Figura 3 – Para o computador, o espelho do carro é uma matriz de números [10].

Cada elemento desta matriz é chamado de elemento de imagem, elemento pictórico, pel ou pixel como é mais conhecido [8]. Conforme Gonzalez [8], expressam-se imagens digitais como funções bidimensionais na forma $f(x,y)$, o valor ou a amplitude de f nas coordenadas espaciais (x,y) é uma quantidade escalar positiva cujo significado físico é determinado pela origem da imagem. A função $f(x,y)$, pode ser caracterizada pela quantidade de iluminação da fonte que incide na cena que está sendo visualizada, e a quantidade de iluminação refletida pelos objetos da cena. Estas características são respectivamente chamadas de iluminância e refletância. A função $f(x,y)$ representa o produto da interação entre iluminância $i(x,y)$ e refletância $r(x,y)$

As imagens digitais são capturadas através de sensores, que possuem a função de transformar a energia de iluminação em uma imagem digital. Segundo Marques Filho e Vieira

Neto [9], a grande maioria dos dispositivos de aquisição de imagens utilizam sensores CCD, estes geralmente são encontrados em câmeras digitais ou outros instrumentos que utilizam sensores de luz. O funcionamento consiste de uma matriz de células semicondutoras fotossensíveis, que atuam como capacitores, armazenando eletricidade proporcional a energia luminosa incidente. O sinal elétrico produzido é condicionado por circuitos eletrônicos, produzindo como saída um Sinal Composto de Vídeo (SCV) monocromático. A forma principal de funcionamento dos sensores pode ser vista na Figura 4. Esta figura demonstra a energia de uma fonte luminosa (a) sendo refletida em um elemento da cena (b), e capturada por um sensor (c). E a conversão de sinais analógicos em uma imagem digital (d)(e).

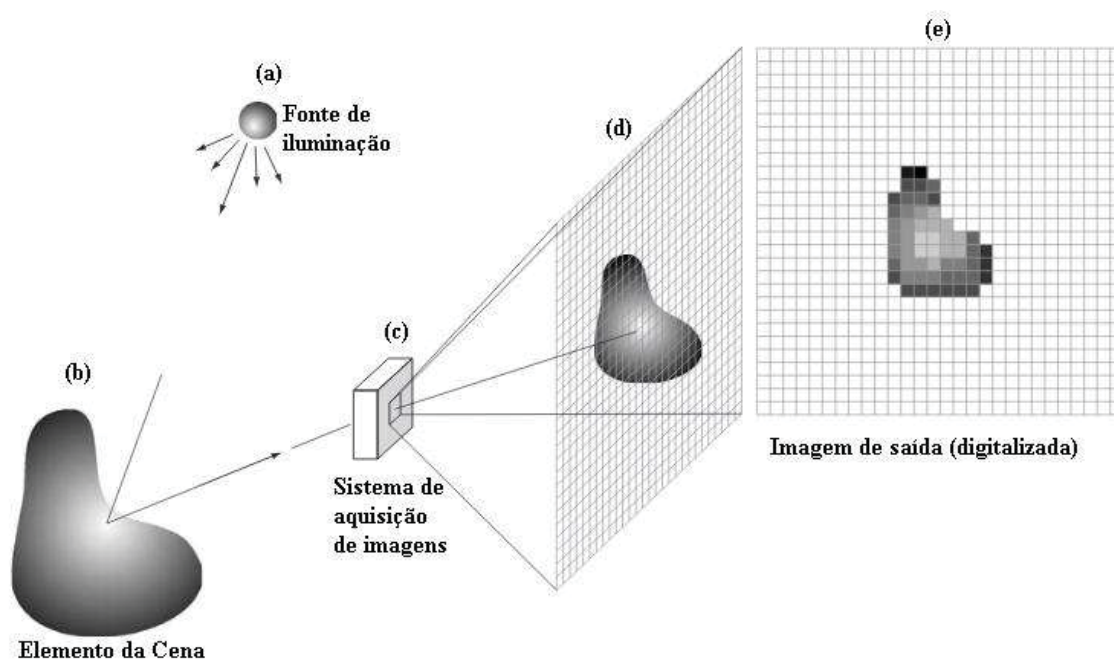


Figura 4 – Exemplo de processo de aquisição de uma imagem digital [8].

Diferentemente das imagens em escala de cinza, as imagens coloridas possuem informações em intervalos ou bandas distintas de frequência, assim é necessário uma função $f(x,y)$ para cada banda. É por exemplo o caso de imagens coloridas no padrão RGB, que são formadas pela informação de cores primárias, como vermelho (R - *Red*), verde (G - *Green*) e azul (B - *Blue*) [9]. Existem diversos padrões de cores, estes, permitem a especificação das cores em um formato padronizado. Um padrão de cor consiste em uma representação tridimensional onde cada cor é representada por um ponto no sistema de coordenadas 3-D [9]. Os modelos mais utilizados em processamento de imagens são: RGB, CMY, CMYK e HSI.

O modelo RGB de cores é caracterizado pelos seus componentes espectrais primários de vermelho, verde e azul. Este modelo baseia-se em um sistema de coordenadas cartesianas, onde o subespaço de interesse é o cubo RGB apresentado na Figura 5, no qual os

valores RGB primários estão em três vértices e as cores secundárias ciano, magenta e amarelo estão em outros três vértices. O preto está na origem e o branco no vértice mais distante [8]. Este modelo é usado principalmente em monitores e câmeras fotográficas.

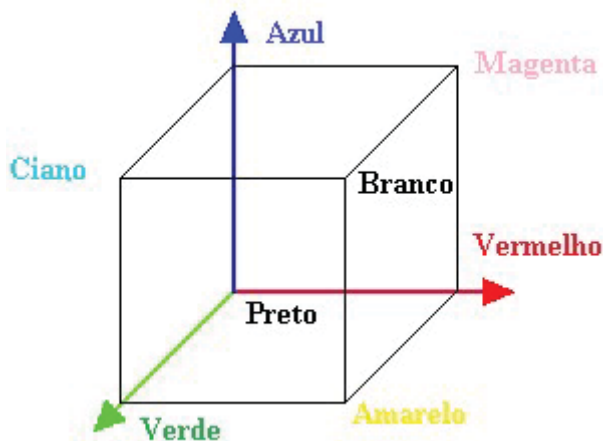


Figura 5 – Cubo RGB.

O padrão CMY é composto pelas cores ciano (C - *Cyan*), magenta (M - *Magenta*) e amarelo (Y - *Yellow*). Estas cores são as cores secundárias de luz ou cores primárias de pigmentos. Ou seja, quando uma superfície coberta com pigmento ciano é iluminada com luz branca, nenhuma luz vermelha é refletida da superfície [8]. A maioria dos dispositivos que opera sob o princípio da deposição de pigmentos coloridos em papel como, por exemplo, impressoras, opera sobre este modelo [9]. Na prática, a combinação das cores deste modelo, produz um preto turvo, desta forma, para produzir um preto verdadeiro, foi necessária a adição de uma quarta cor, o preto [8]. Este modelo chama-se CMYK, onde K é a cor preta, e é conhecido na indústria gráfica como “impressão a quatro cores”.

Os modelos apresentados acima, são recomendados para implementações em hardware. Além disso, o modelo RGB se adapta bem ao fato do olho humano ser bastante perceptivo às cores primárias [8]. Porém quando é necessário utilizar um modelo de percepção de cor pelo olho humano, utiliza-se HSI. Basicamente, não se faz referência a cor de um objeto pela porcentagem de cada uma das cores primárias, tampouco se pensa em imagens coloridas como a união de três cores. Quando um humano vê uma imagem ele descreve ela em termos de matriz (H - *Hue*), saturação (S - *Saturation*) e intensidade ou brilho (I - *Intensity*) [8]. Onde matriz é um atributo que descreve a cor pura, saturação é o grau de diluição da cor pura em luz branca e brilho incorpora a noção acromática de intensidade e é um dos principais fatores de descrição da sensação de cores [8]. As características apresentadas, tornam o modelo HSI ideal para utilização em processamento de imagens com base nas descrições de cores que são naturais e intuitivas para os seres humanos. Este modelo

é mais indicado para uso em sistemas de visão artificial, como por exemplo, uma máquina de colheita automática que precisa identificar qual são os frutos maduros [9].

Ao digitalizar uma imagem, ela assume um tamanho adimensional em pixels, ou seja, a quantidade de pixels verticais e horizontais. Porém, pode-se conhecer uma medida de qualidade da amostragem, conhecendo-se a razão entre o número de pixels e o tamanho real da imagem. Isso é definido por resolução [9]. Em geral é medida em *dots per inch* (pontos por polegada ou dpi), mas também pode ser medida por pontos por centímetro ou dpc, ou qualquer unidade equivalente. Pontos por unidade de distância é uma medida de resolução de imagem utilizada em editoras e indústrias gráficas. Para dar ideia da qualidade, os jornais são impressos em 75 dpi, enquanto em revistas utiliza-se 133 dpi [8]. Portanto, para que as medidas de resolução espacial tenham significância, elas devem ser expressas com relação a unidades espaciais. Ou seja, apenas o tamanho da imagem não diz nada, dizer que uma imagem tem uma resolução de 1024 x 1024 pixels não fará sentido se as dimensões espaciais da imagem não forem especificadas. Na Figura 6 é possível perceber o conceito comparando uma imagem de 1250 dpi (a) e a mesma com 72 dpi ampliada (b).

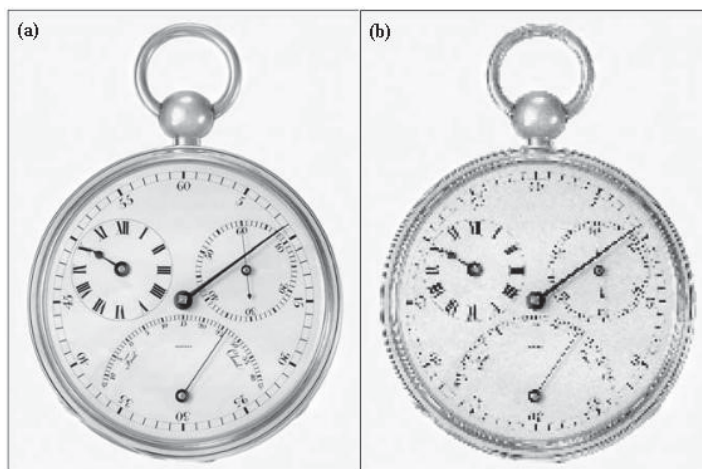


Figura 6 – Efeitos da redução da resolução espacial [8]

2.3. PASSOS FUNDAMENTAIS EM PROCESSAMENTO DE IMAGENS

O processamento de imagens é constituído por várias etapas, onde cada uma delas se concentra em realizar alguma tarefa visando a próxima etapa ou o resultado final. Existem etapas onde estamos preocupados em aproximar a imagem digital da imagem original, ou melhorar alguns atributos para que melhor apresentem as características interessantes. Outras etapas preocupam-se em extrair as informações contidas nas imagens [11]. As etapas mais importantes relacionadas ao tema são: aquisição de imagens, realce, restauração, compressão,

segmentação, extração de características e reconhecimento [8] [9]. Este tópico tem como objetivo especificar cada uma das etapas descritas exceto a etapa de aquisição que foi tratada no tópico anterior.

2.3.1. Realce de imagens

O realce de imagens é o processo de manipular uma imagem de forma que o resultado seja mais adequado do que o original para uma aplicação específica, seja isto, uma análise humana ou um próximo processo de processamento de imagens [8]. As técnicas de realce são aplicadas de acordo com o problema especificado, ou seja, uma técnica que realce uma chapa de raio X pode não ser adequada para realçar uma imagem de satélite. O realce também não possui nenhum padrão de resultado, pois o que realmente importa é a avaliação do usuário quanto a imagem de saída.

2.3.2. Restauração

Semelhante ao realce, o processo de restauração tem por objetivo, melhorar uma imagem em algum sentido predefinido [8], geralmente, este processo é aplicado para remover os ruídos das imagens. Apesar de haver sobreposição, o realce é considerado um processo subjetivo, enquanto a restauração é um processo objetivo, ou seja, enquanto a etapa de realce torna uma imagem melhor do ponto de vista humano, a restauração torna a imagens melhor do ponto de vista matemático e computacional.

Basicamente a restauração procura recuperar uma imagem corrompida por ruídos com base no conhecimento de degradação [8], ou seja, quando é possível identificar a função que representa a deformação ou construir um modelo matemático, é possível aplicar uma função inversa que melhora o aspecto da imagem [9]. Correção de foco, imagens borradas, são exemplos de casos onde a restauração é aplicada, porém é um processo de alta complexidade que existe um alto custo computacional [9].

Na Figura 7 é demonstrado um exemplo de restauração. Na imagem (a) um circuito impresso com ruído sal e pimenta³. Na imagem (b) é aplicado um filtro de mediana, e na imagem (c) a aplicação de mais um filtro de mediana que torna a imagem ainda mais nítida que a (b).

³ Nomenclatura de ruído que se caracterizam por pontos brancos (sal) e pretos (pimenta) na imagem.

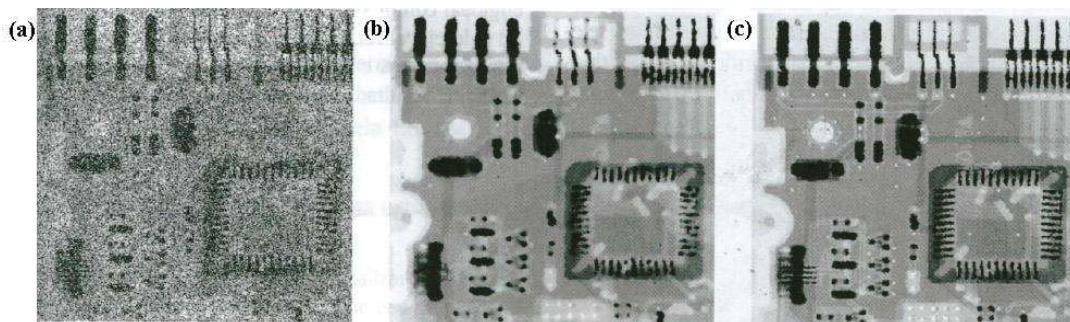


Figura 7 – Exemplo de restauração de imagem [8].

Uma das principais aplicações desta etapa de processamento é a tomografia computadorizada por raios X, tomografia por emissão de fóton único (SPECT), tomografia por emissão de pósitrons (PET), ressonância magnética (MRI) e algumas modalidades de aquisição por ultrassom [8].

2.3.3. Compressão

Um dos grandes desafios do processamento de imagens é contornar o problema da grande quantidade de dados necessários para armazenar e transmitir imagens digitais [9]. Para tal, como o nome sugere, a compressão é a arte e a ciência de reduzir o volume de dados necessários para representar uma imagem [8]. As imagens podem conter informações irrelevantes ou repetidas, são nelas que os modelos matemáticos de compressão atuam diminuindo a quantidade de dados, sem que a informação seja prejudicada.

Quanto aos dados irrelevantes, a compressão procura eliminar de uma imagem as informações ignoradas pelo sistema de visão humano, que tem como característica o descarte das pequenas variações de intensidades. Este processo chama-se compressão com perda, pois elimina alguns dados da imagem, destaca-se neste cenário o JPEG [9]. Outra forma de processamento, a compressão sem perda, consiste em comprimir uma imagem, utilizando algoritmos que eliminam as informações redundantes que posteriormente na descompressão podem ser reconstruídas.

Na Figura 8 é demonstrado um exemplo de compressão. A imagem (a) é comprimida em uma taxa de 25:1 (b). Na imagem (d) é aplicada uma taxa de compressão de 52:1 (e). As imagens (c) e (f) demonstram a área ampliada contendo as diferenças entre as duas taxas de compressão.

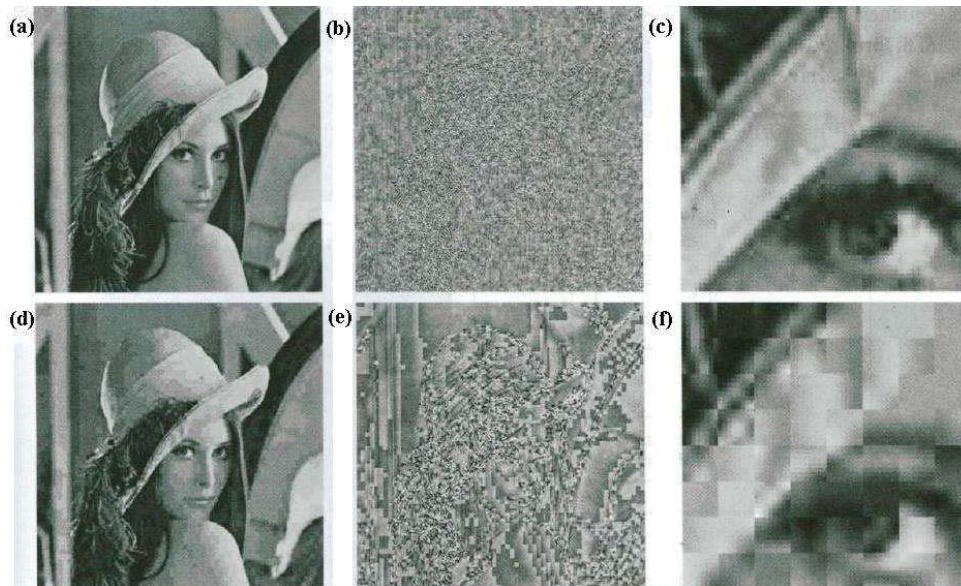


Figura 8 – Exemplo de compressão de imagem [8].

A compressão, atualmente é aplicada em várias áreas do dia a dia, por exemplo, em imagens de sites para que seja diminuído o tempo de carregamento das páginas, também em vídeos para que possuam seus tamanhos equivalentes a mídia proposta para gravação.

2.3.4. Segmentação

O processo de segmentação consiste em dividir uma imagem em suas unidades significativas que a compõem [9]. Estas divisões são baseadas nos objetos de interesse, onde o nível de detalhe em que a subdivisão é aplicada depende do problema a ser resolvido [8]. A segmentação é considerada uma das tarefas mais difíceis em PDI, pois a precisão desta etapa garante o sucesso ou o fracasso dos procedimentos de análise computacional [8]. Para potencializar este processo, é possível utilizar sensores que realcem objetos de interesse, como por exemplo, um sensor que realce a luz vermelha [8].

Na Figura 9 é demonstrado um exemplo de aplicação de segmentação baseada na estimativa do limiar que será abordado no tópico 2.4.1. Na imagem (a) a impressão digital está ruidosa, após a segmentação, nota-se na imagem (b) a existência apenas dos pontos de interesse da imagem.

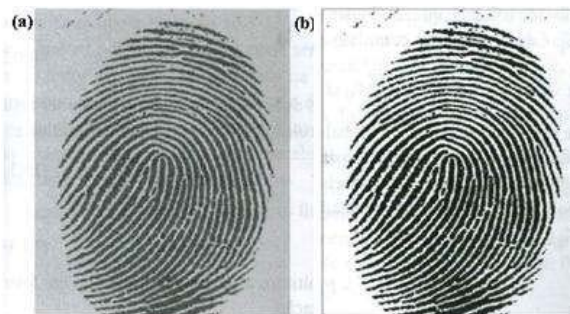


Figura 9 – Exemplo de segmentação de imagem [8].

Os algoritmos de segmentação baseiam-se em duas propriedades básicas de valores de intensidade: descontinuidade e similaridade. Onde a descontinuidade procura dividir uma imagem com base nas mudanças bruscas de intensidade, como bordas [8]. Já a similaridade baseia-se na divisão de uma imagem em regiões semelhantes de acordo com critérios predefinidos [8].

2.3.5. Extração de características

Esta etapa tem por objetivo coletar características das imagens resultantes da segmentação [9]. Após o processo de segmentação separar a imagem em regiões é possível representar ou descrever com mais facilidade o conteúdo presente na imagem. Neste momento a etapa de extração de características, também conhecido como representação ou descrição, propõe-se a representar uma determinada região utilizando, por exemplo, algoritmos como o seguidor de contorno.

Determinar uma região envolve duas opções, uma delas é a representação de acordo com as características externas, que se baseia na fronteira de pixels que separam o objeto de interesse do restante. Uma região externa pode ser representada por sua fronteira e pode ser descrita por suas características de extensão [8]. A representação externa é escolhida, quando o elemento a ser extraído é definido de acordo com sua forma.

Na Figura 10 é demonstrado um exemplo de extração de características. Na imagem (a) está a imagem original capturada na banda infravermelha. Após ela é submetida ao processamento de limiarização (b). Na imagem (c) é visualizado o resultado da busca por maior componente conexo de (b). Por fim, na imagem (d) o resultado, apenas o esqueleto do de (c), que é o curso do rio.

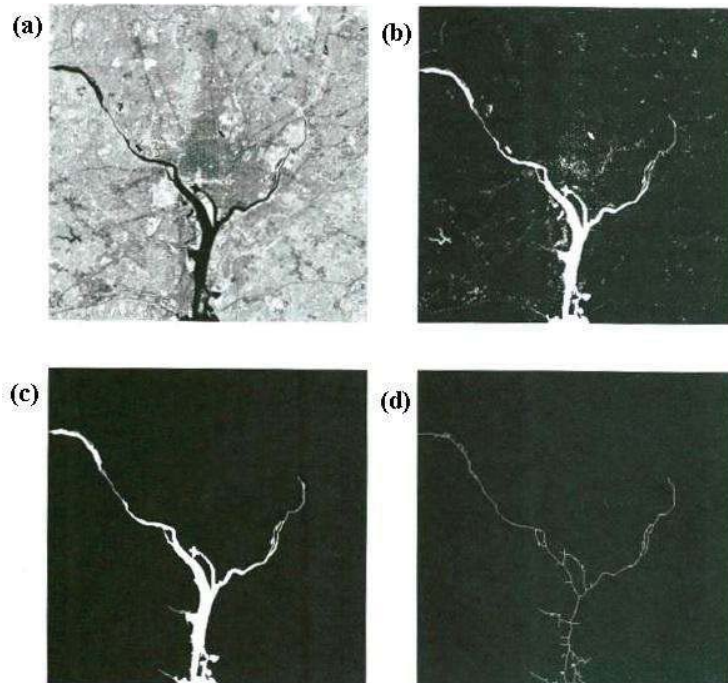


Figura 10 – Exemplo de extração de características de uma imagem [8].

Outra representação baseia-se nas características internas (pixels que compõem o preenchimento do objeto). Esta representação é utilizada quando é necessário descrever algo de acordo com as cores [8]. É comum a utilização das duas representações juntas, por exemplo, quando é necessário identificar um objeto e classificá-lo conforme sua cor.

2.3.6. Reconhecimento

Reconhecimento é o processo de atribuição de um rótulo a um objeto ou padrão da imagem, baseado em suas características, traduzidas por seus descritores [9]. Uma característica desta etapa é que apesar da entrada ser uma imagem, a saída são os dados extraídos da imagem. Por exemplo, em uma imagem cujo objetivo é descobrir quantas garrafas apresentam defeito em uma linha de produção, teremos como saída da etapa de reconhecimento um número, que é a quantidade de produtos defeituosos.

Conceitualmente dentro das metodologias de reconhecimento existem duas áreas principais: decisões teórica e estrutural. A categoria de decisões teórica utiliza descritores quantitativos como comprimento, textura e área. Já a estrutural utiliza descritores relacionais, estes se aplicam bem às fronteiras ou às regiões e seu objetivo é reescrever os padrões repetidos em uma região.

Nesta etapa são aplicados ainda de forma precoce os conceitos de visão computacional e inteligência artificial. Porém as soluções de reconhecimento atuais baseiam-se em abordagens heurísticas [8]. Nesta etapa também estão inseridas as técnicas de aprendizado de máquina (*machine learning*), que utiliza algoritmos de treinamento para capturar padrões das imagens e reconhecê-los em outra imagem.

2.4. TÉCNICAS DE PROCESSAMENTO

Neste tópico, serão descritos alguns métodos utilizados no processamento de imagens digitais, mas especificamente para a etapa de segmentação. Se tratando de detecção de elementos é imprescindível o uso desta etapa e de algumas das técnicas que serão descritas.

2.4.1. Limiarização ou binarização

A limiarização é considerada uma das mais importantes abordagens para segmentação de imagens [8]. Tem como princípio separar regiões de uma imagem quando esta apresenta duas classes como, por exemplo, o fundo e um objeto [9]. Devido ao produto do processo ser uma imagem contendo valores matriciais de 1 ou 0, ele também é conhecido como binarização. A função baseia-se em definir um valor de corte (T) para uma imagem que ao processar, considerará todos os valores de pixels acima ou iguais a T como um, e os valores menores como zero. Matematicamente a operação de limiarização é definida pela Equação (1) [8].

$$\begin{aligned} g(x, y) &= 1 \text{ se } f(x, y) \geq T \\ &= 0 \text{ se } f(x, y) < T \end{aligned} \quad (1)$$

Na Figura 11 é dado um exemplo de limiarização com a imagem original (a) e uma imagem processada com $T = 160$ (b) e outra com $T = 215$ (c). É possível notar que ao aumentar T mais pontos claros da imagem tendem a ficar com valor um (preto).



Figura 11 – Exemplo de limiarização.

2.4.2. Detecção de linhas

A detecção de linhas, como o nome sugere, consiste em detectar linhas em várias direções ou em direções específicas em uma imagem digital. Este processo também pode ser chamado de detecção de descontinuidades. Porém para compreender como as linhas são detectadas, é importante compreender o filtro laplaciano, pois a aplicação deste filtro faz parte da etapa de detecção de linhas. O filtro laplaciano baseia-se em converter a imagem, para que os pixels que contenham linhas possuam valores positivos, enquanto o restante possui negativo [8]. A partir deste momento é possível aplicar as máscaras de detecção de linhas demonstradas na Figura 12.

-1	-1	-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	2	-1	-1	2
2	2	2	-1	2	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+ 45°			Vertical			-45°							

Figura 12 – Máscaras de detecção de linhas [8].

Nota-se que para cada direção existe uma máscara de detecção que são números positivos indicando presença de linhas. Em uma aplicação também é possível especificar as direções específicas quando é necessário encontrar um padrão de linha.

Na Figura 13 é exibida a imagem original (a) e a imagem processada utilizando um filtro laplaciano (b), nota-se que as linhas da imagem estão destacadas.



Figura 13 – Imagem original (a) e imagem processada com filtro laplaciano (b).

2.4.3. Detecção de bordas

A detecção de bordas é o método utilizado para segmentar imagens com base nas variações abruptas de intensidade [8]. As detecções são baseadas em modelos de borda conforme Figura 14 são classificados de acordo com o perfil de intensidade. A imagem (a) mostra a seção de uma borda vertical em degrau e o perfil horizontal da borda. A (b) demonstra um perfil de rampa. Um terceiro modelo de borda é chamado de borda em forma de telhado ou *roof edge* (c).

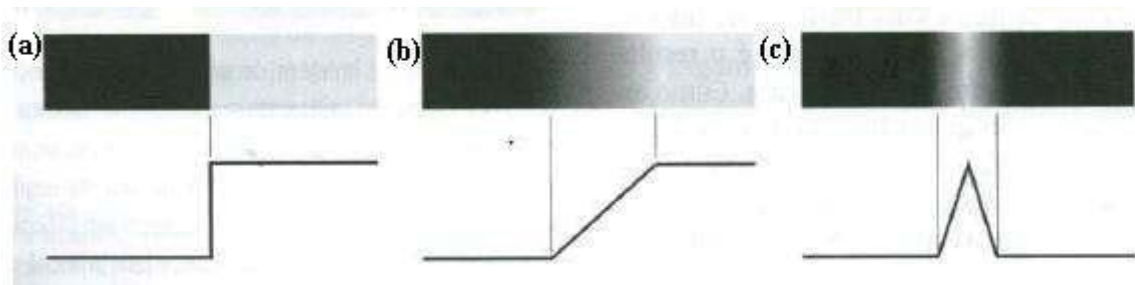


Figura 14 – Representação de bordas [8].

A imagem (a) demonstra as bordas degrau, este tipo de borda é comum quando existe uma variação de cor ou uma borda não suavizada. As bordas rampa (b) tendem a serem desfocadas e ruidosas. As bordas telhado (c) são modelos de linhas através de uma região. A Figura 15 demonstra um exemplo dos tipos de bordas em uma imagem.

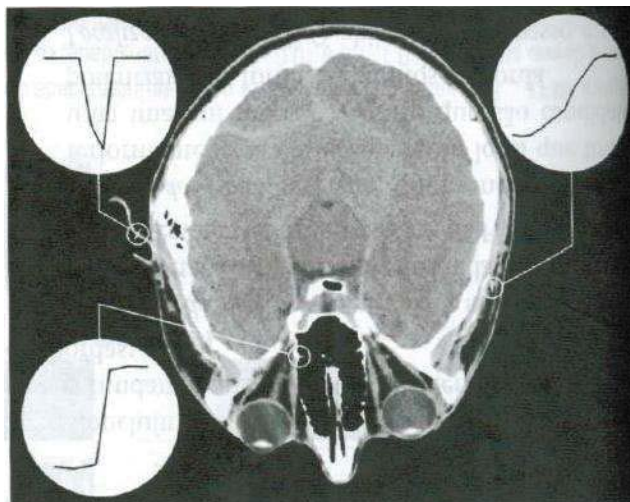


Figura 15 – Exemplo de bordas detectadas.

É importante observar que para a detecção de borda, são necessários três passos fundamentais. A suavização da imagem para redução dos ruídos cujo objetivo é melhorar os aspectos da imagem. A detecção de pontos de borda que procura todos os potenciais candidatos a borda. Por fim a localização da borda, que seleciona entre os possíveis pontos de borda os que realmente pertencem ao conjunto de pixels que formam uma borda [8].

2.5. VISÃO COMPUTACIONAL

Segundo Bradaski [10], visão computacional é a transformação de dados a partir de uma câmera fotográfica ou de vídeo em uma decisão ou em uma nova representação. A decisão pode ser “há uma pessoa nesta sala” ou “existem cinco carros nesta imagem”, e a nova representação pode transformar a imagem de entrada em uma imagem com contornos sobre os objetos alvos [10]. Em uma definição simples e clara, a visão computacional tem como objetivo fazer com que o computador “veja” como o olho humano vê.

A visão computacional é uma especialização de Processamento de Imagens. Ou seja, Processamento de Imagens é um processo onde a entrada do sistema é uma imagem e a saída é um conjunto de valores numéricos que podem ou não ser demonstrados em uma nova imagem. Já visão computacional procura emular a visão humana, assim, possui como entrada uma imagem, porém a saída é uma interpretação desta imagem [8]. Por exemplo, ao utilizar como entrada uma imagem de um ambiente, um sistema de visão computacional consegue distinguir determinados objetos, diferente de Processamento de Imagens que neste caso aplicaria técnicas para aprimorar alguns aspectos desta imagem auxiliando assim a visão humana a distinguir os objetos.

A tarefa de tornar o computador um sistema visual parece simples, pois para nosso cérebro basta cruzar informações feitas a partir de anos de vida para identificar algo. Em um sistema de visão computacional, o computador recebe apenas uma matriz de números, e entre estes números, além das informações reais que nos interessam, pode-se ter ruídos ou mesmo o objeto alvo desfocado. Isto torna a tarefa de encontrar objetos neste universo “barulhento” algo extremamente difícil.

Como neste universo não existe nenhum modelo genérico para detecção e percepção visual, a visão computacional utiliza um conjunto de algoritmos que são responsáveis por realizar subtarefas bastante limitadas. Estes algoritmos são divididos em grupos como, por exemplo, algoritmos de segmentação. Geralmente os problemas em visão computacional são resolvidos encaixando os conjuntos de algoritmos [12] obtendo assim um resultado desejado.

Para Wangenheim [12], compreender as limitações da visão computacional e conhecer formas de tirar o máximo de proveito do que atualmente há disponível, é um aspecto muito importante a ser considerado por quem deseja trabalhar em problemas cuja solução possivelmente envolva o uso da ciência.

Cientistas e escritores de ficção científica são fascinados pela possibilidade de construir máquinas inteligentes com capacidade de compreender o mundo visual. Tais máquinas poderiam, por exemplo, realizar tarefas de reconhecimento auxiliando diversas áreas. Na Figura 16 são apresentados alguns exemplos que demonstram a aplicabilidade da visão computacional. A visão computacional, além de tarefas de reconhecimento, é capaz de reconstruir objetos ou espaços (a) e (b). Na imagem (c) é demonstrado um sistema de rastreamento de caminhada, enquanto na imagem (d) a visão computacional é aplicada para reconhecer as faces humanas presentes na imagem [13].

Outro exemplo de uso da visão computacional neste caso para fins industriais são os sistemas de inspeção visual. Nesta área são construídos softwares que buscam características específicas em produtos gerados em linhas de produção [12], possibilitando que o ambiente seja mais produtivo.

Outras áreas, como a Medicina, utilizam visão computacional para a identificação de doenças em diagnósticos por imagens; a Robótica, para construir os sistemas de visão de robôs, como navegação e identificação de sinalizações, veículos, seres vivos, etc. [14].

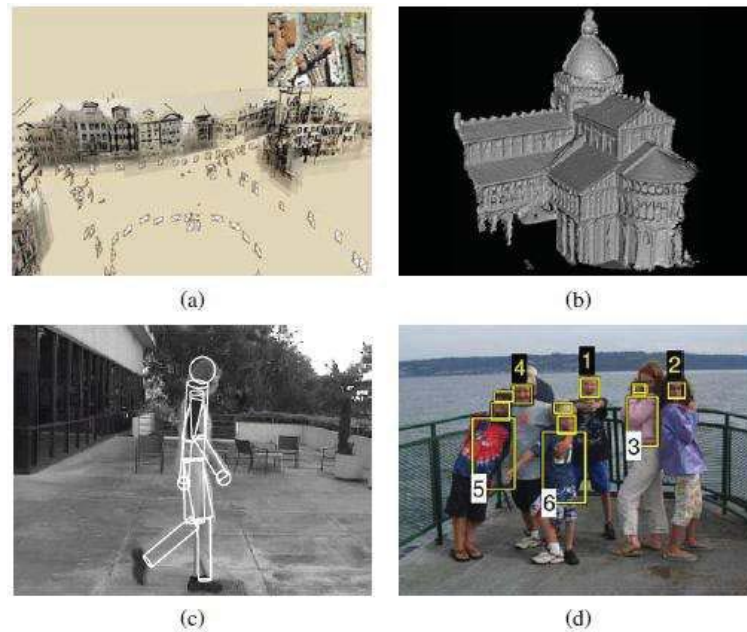


Figura 16 – Exemplos de aplicação de visão computacional [13].

2.6. DEEP LEARNING

Deep Learning é uma técnica de aprendizagem de máquina que ensina os computadores a fazer coisas naturalmente como os humanos fazem, como por exemplo, aprender [15]. Nos últimos anos, as técnicas de *Deep Learning* têm revolucionado diversas áreas da computação, em especial a visão computacional. Isso ocorreu devido a disponibilidade de bancos de dados com milhões de imagens e o aumento do desempenho dos computadores que permitiram reduzir o tempo necessário para realizar o processamento [16].

Em *Deep Learning* os computadores aprendem a executar as tarefas de classificação diretamente em imagens, textos ou sons. Para isso, os modelos são treinados utilizando grandes massas de dados, como por exemplo, uma coleção de milhões de imagens, várias horas de filmagens, várias horas de gravação de voz, etc. Os modelos de aprendizagem podem atingir precisão equivalente a humana, porém, as vezes excedem este nível [15].

A grande maioria dos métodos de aprendizagem baseia-se em arquiteturas de rede neural, sendo muitas vezes conhecidos com redes neurais profundas (*deep neural networks*). Um dos mais populares tipos de rede neural profunda é conhecido como *convolutional neural networks* (CNN). Os CNNs eliminam a necessidade de extração manual das características das imagens usadas na classificação [15].

A tecnologia é utilizada em inúmeras áreas. Em carros autônomos é utilizada para aprender automaticamente a detectar objetos como placas de trânsito, pedestres, etc. Na

medicina é utilizada, por exemplo, para detectar células cancerígenas. Nos eletrônicos, é usado para fazer traduções, reconhecimento de comandos por voz, etc.

Basicamente, *Deep Learning* é uma especialização de *Machine Learning*. Ou seja, *Machine Learning* utiliza características extraídas manualmente das imagens, enquanto *Deep Learning* extrai automaticamente essas características. Este fato oferece vantagem às técnicas de *Deep Learning*, já que conseguem manter e melhorar o aprendizado a medida que o tamanho dos dados aumenta.

A Figura 17 demonstra em alto nível a arquitetura de um sistema de classificação utilizando *Machine Learning* (cima) em comparação com *Deep Learning* (baixo). É possível notar que enquanto *Machine Learning* analisa algumas características extraídas manualmente, *Deep Learning* analisa um conjunto destas para aprender a identificar um objeto.

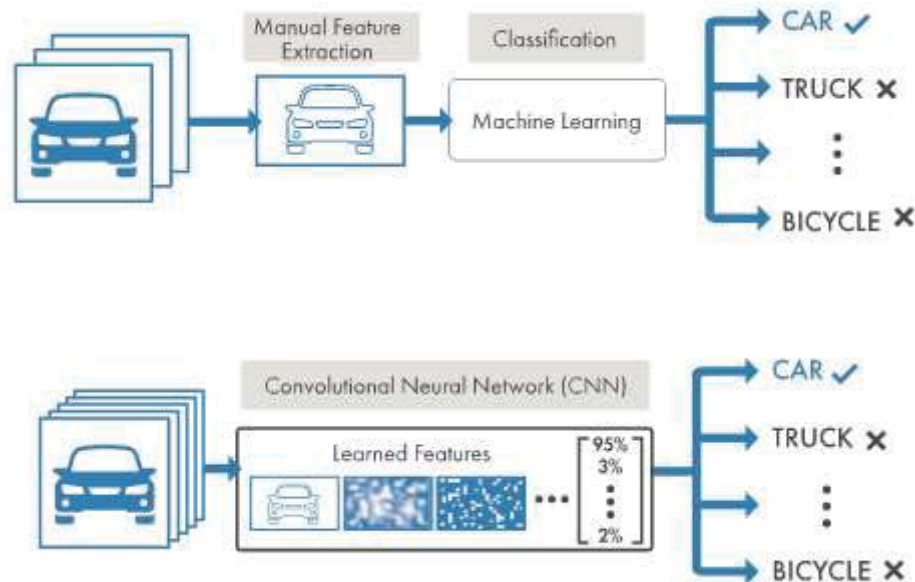


Figura 17 – Comparação entre *Machine Learning* e *Deep Learning* [15]

2.7. BIBLIOTECAS DE PROCESSAMENTO DE IMAGENS E DEEP LEARNING

Neste tópico, serão descritos alguns dos principais *frameworks* de processamento de imagens e visão computacional. Um *framework* é conceitualmente definido por um conjunto de ferramentas ou uma biblioteca munida de funções que se propõem acelerar o desenvolvimento de softwares através de funções prontas para o uso. Como muitas tarefas de PDI são comuns, por exemplo, a limiarização estas bibliotecas já implementam a função

deixando-a pronta para o uso. A seção também apresenta bibliotecas utilizadas para o aprendizado de máquina utilizando o conceito de *deep learning*.

2.7.1. OpenCV

OpenCV (*Open Source Computer Vision*) é uma biblioteca de visão computacional de código aberto que contém mais de 500 algoritmos para análise de vídeo e imagens [17]. A biblioteca foi desenvolvida pela Intel e um time liderado por Gary Bradski em 1999 e atualmente é utilizada em larga escala como ferramenta de desenvolvimento e pesquisas na área de visão computacional. Foi desenvolvida em C e C++ e é multiplataforma, ou seja, pode ser executada em Linux, Windows e Mac OS X [10].

O objetivo do OpenCV é prover um simples uso da visão computacional que ajude as pessoas a construir sofisticados sistemas de visão de uma forma rápida [10]. Além disso, tem suporte para aplicação de aprendizado de máquina através da *Machine Learning Library* (MLL) [10].

Na Tabela 1 é demonstrado um exemplo de código desenvolvido na linguagem Java que realiza o procedimento de limiarização em OpenCV. Na linha quatro é executado o algoritmo que já contém toda a lógica responsável pela limiarização. Caso esta técnica de processamento fosse construída exigiria bastante tempo de desenvolvimento. Através da biblioteca basta apenas invocar o método.

Tabela 1 – Exemplo de implementação de limiarização em OpenCV

1.	<code>public void run() {</code>
2.	<code> int tValue = 100;</code> <code> //leitura da imagem</code>
3.	<code> Mat source = Imgcodecs.imread("in.jpg",</code> <code> Imgcodecs.CV_LOAD_IMAGE_GRAYSCALE);</code> <code> //método que realiza a limiarização</code>
4.	<code> Imgproc.threshold(source, source, tValue, 255,</code> <code> Imgproc.THRESH_BINARY_INV);</code> <code> //método que salva a imagem após as alterações</code>
5.	<code> Imgcodecs.imwrite("out.jpg", source);</code>
6.	<code>}</code>

2.7.2. ImageJ

ImageJ é um software de processamento de imagens *open source*⁴ inspirado no NIH Image⁵. Ele pode rodar em qualquer computador que possua a versão 1.6 ou superior do Java. É disponibilizado para os sistemas operacionais: Windows, Linux e Mac OS X [18].

Dentre as características do ImageJ, além ser gratuito, destacam-se a simplicidade de uso da ferramenta e a capacidade de alterações que podem ser feitas no software baseadas nas necessidades do usuário [19]. Estas alterações são realizadas por *plugins* ou macros disponíveis na internet. O ImageJ é sustentado por uma comunidade que além de melhorar constantemente a ferramenta, dissemina o conhecimento relacionado a ferramenta [19].

Na Figura 18 é demonstrado um exemplo de uso do ImageJ com o *plugin*: *Contrast-Limited Adaptive Histogram Equalization (CLAHE)* [19]. Este *plugin* recebe como entrada a imagem original (a), após o processamento resulta na imagem (b) com o contraste melhorado, este procedimento permite a melhor visualização dos componentes da imagem.

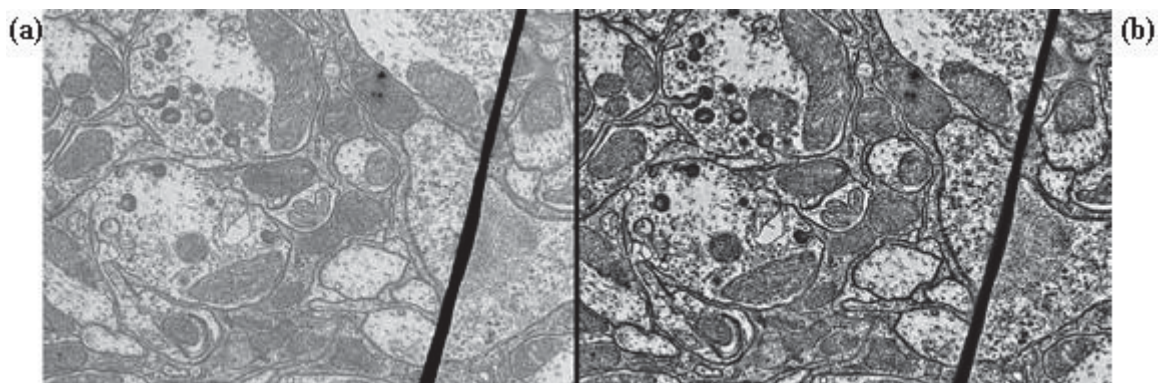


Figura 18 – Exemplo de processamento de imagem usando ImageJ [19].

2.7.3. TensorFlow

TensorFlow é uma biblioteca aberta para aprendizado de máquina. Foi originalmente desenvolvida por pesquisadores e engenheiros do Google Brain Team juntamente com o Google's Machine Intelligence. A solução é flexível e pode ser usada para expressar uma variedade expressiva de algoritmos, incluindo algoritmos de treinamento e inferência para modelos de redes neurais profundas.

⁴ Open Source é um termo em inglês que significa código aberto.

⁵ NIH Image foi originalmente escrito por Wayne Rasband em Pascal para a Macintosh. É considerado o precursor do ImageJ.

Essa biblioteca tem sido utilizada para realizar pesquisas e para implantar sistemas de aprendizagem de máquinas, incluindo reconhecimento de fala, visão computacional, robótica, recuperação de informações, processamento de linguagem natural, extração de informações geográficas e descoberta computacional de medicamentos [20].

Vários serviços do Google utilizam o TensorFlow em produção. Por ser um projeto de código aberto, tornou-se amplamente utilizado em pesquisas relacionadas com aprendizado de máquina [21].

2.7.4. Caffe

Caffe é um *framework* de *Deep Learning* desenvolvido por *Berkeley AI Research* (BAIR), em conjunto com uma comunidade aberta de desenvolvedores, da qual o Facebook faz parte [22].

O software foi desenvolvido em C++ com interface em Python com a filosofia de ser expressivo, rápido, modular e servir a comunidade de pesquisadores. Suporta vários tipos de redes neurais de *Deep Learning*, como a CNN e a rede neural recorrente (RNN) [22].

3. TRABALHOS RELACIONADOS

Existem diversas aplicações na Agricultura que envolvem ou necessitam do uso de processamento de imagens. Dentre elas, uma parcela tem por objetivo auxiliar na detecção de pragas e na tomada de decisão. Por meio de dados estatísticos, o agricultor pode, por exemplo, saber se é realmente necessário aplicar um determinado pesticida, ou qual a quantidade ideal. Além de benefícios financeiros, soluções dessa natureza também podem contribuir na redução de agentes químicos e preservação do meio ambiente.

Dentro desse contexto, esse capítulo apresenta alguns trabalhos relacionados ao auxílio à tomada de decisão na Agricultura utilizando processamento de imagens, em específico, abordagens destinadas a contagem e caracterização de pragas. Os trabalhos desta seção apresentam abordagens similares a proposta deste documento.

3.1. USING DIGITAL IMAGE PROCESSING FOR COUNTING WHITEFLIES ON SOYBEAN LEAVES

O trabalho de Barbedo [7] tem por finalidade a contagem de aleirodídeos⁶ nas folhas de soja utilizando processamento de imagens. Os seres humanos são normalmente muito mais lentos do que as máquinas na realização de tarefas simples de contagem. Também são muito mais suscetíveis a erros causados por ilusões visuais, fadiga, tédio, entre outros.

Baseando-se nisso e na inexistência de um sistema que não dependesse de um hardware complexo, Barbedo [7] inicialmente aborda a identificação das características em cada fase do ciclo de vida dos insetos, baseando-se na coloração dos indivíduos em cada etapa de crescimento. Numa etapa seguinte, propõe um algoritmo para contagem de unidades de aleirodídeos. Para desenvolver a aplicação o autor utilizou o software Matlab.

A estratégia de detecção inicia com a imagem contendo os insetos conforme Figura 19 (a). Na imagem (b) os objetos de interesse são segmentados, porém ainda existem várias impurezas. A imagem (c) demonstra mais um processo onde são eliminadas as partes não interessantes da imagem. Já na imagem (d) aparecem todos os objetos que tem até 10% do tamanho do maior objeto. Na imagem (e), o resultado do processamento. Nota-se que nem todos os insetos foram detectados, segundo o autor isso acontece pois alguns aleirodídeos são quase transparentes e acabam sumindo na etapa de segmentação por binarização.

⁶ Aleirodídeos são fitófagos e normalmente se alimentam da parte inferior das folhas das plantas.

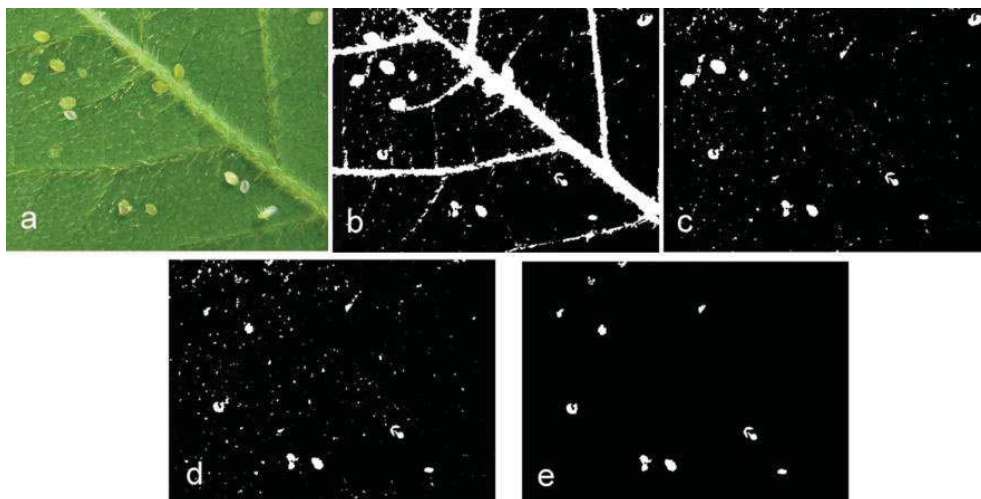


Figura 19 - Etapas de processamento [7].

Como resultado, Barbedo [7] obteve um significativo número de acertos quando os insetos estão em uma fase mais próxima a adulta, a taxa de acerto foi de 95%. Já para as fases iniciais, em virtude das características semelhantes nos primeiros estágios de vida, a eficiência do algoritmo caiu tendo como taxa de acerto 68%. Segundo o autor, pode-se considerar como bons os resultados alcançados, passíveis ainda de correção humana em alguns casos. Para diminuir esta interação, futuramente o autor pretende aplicar técnicas de aprendizado de máquina ao software.

3.2. DETECTION OF APHIDS IN WHEAT FIELDS USING A COMPUTER VISION TECHNIQUE

Conforme Tao Liu *et al* [23], o monitoramento das populações de pulgões e a identificação de espécies, fornece dados importantes relacionados com a dinâmica das populações de pragas e manejo integrado. Dentro desse contexto, os autores desenvolveram um software para contagem e identificação de pulgões, na cultura do trigo, que não necessita de armadilhas de coleta, ou de um fundo explicitamente definido. As imagens para processamento são simplesmente capturadas e processadas diretamente nas plantas infestadas conforme a Figura 20.

Para identificar os insetos, os autores utilizam técnicas de *support vector machine* (SVM) para mineração de dados. Para formar a base de conhecimento, a solução utilizou 2000 imagens contendo exemplos verdadeiros e 5000 imagens de exemplos negativos.

A Figura 20 demonstra um exemplo da etapa de pré-processamento, onde as imagens são submetidas ao processamento por MSER. As imagens (a) e (c) são os originais

usadas como entrada, sendo (a) como um caso com poucos pulgões e (c) com muitos. As imagens (b) e (d) contém os resultados do processamento.

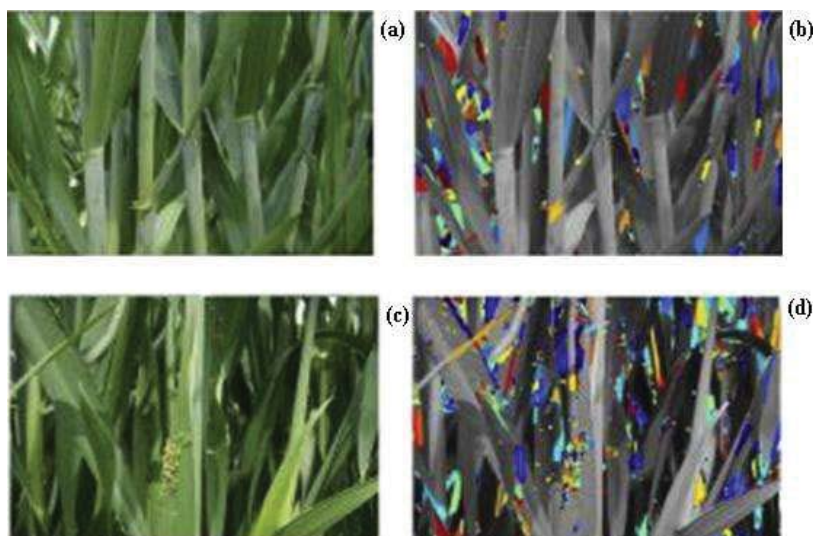


Figura 20 – Resultado da aplicação de MSER [23].

Na etapa de processamento e pré-processamento foram utilizadas técnicas de detecção de *Maximally Stable Extremal Region* (MSER) e *Histograms of oriented gradient* (HOG). A fusão dos três métodos de processamento de imagens: SVM, MSER e HOG. Ocasionalmente resultados satisfatórios, porém os autores relatam um alto valor de falsos negativos quando os insetos possuem cor verde e a quantidade é acima de 80 indivíduos, o valor da taxa foi de 14%. Segundo Tao Liu *et al.* [23], este resultado é atribuído às semelhanças físicas dos insetos com algumas partes do caule, das folhas ou do fundo da planta.

Como conclusão, os autores comentam que foi possível criar um software para contagem de pulgões com diferentes tamanhos e cores. Como média de acerto foi obtido o valor de 86,81% e 8,91% de erro. Isto demonstrou que o software pode ser uma ferramenta para monitoramento e investigação de populações de pulgões do trigo.

3.3. AUTOMATIC IDENTIFICATION AND COUNTING OF SMALL SIZE PESTS IN GREENHOUSE CONDITIONS WITH LOW COMPUTATIONAL COST

Conforme Chunlei Xia *et al.* [24], contar o número de insetos visualmente em armadilhas adesivas, especialmente se tratando de pequenas pragas de estufas, tem resultados pouco confiáveis e eficientes. Diante disso, implementar um modo automático de identificação de pragas é vital à agricultura moderna. Estas soluções são capazes de aumentar a produtividade e diminuir o uso de pesticidas e agentes poluentes.

Neste trabalho, os autores propuseram um procedimento de processamento de imagens para identificação de três espécies de pragas (Figura 21 (b)) de estufas utilizando imagens de baixa resolução. O processo inicia com a captura de insetos utilizando armadilhas adesivas (Figura 21 (a)), que são digitalizadas e, posteriormente, processadas.

Na etapa de segmentação, Chunlei Xia *et al.* [24] aplicam o algoritmo de segmentação de *watersheds*⁷ controlado para extrair as possíveis áreas da imagem consideradas como insetos. As microimagens candidatas são medidas de acordo com as medidas da sua área e consideradas válidas se satisfazem um determinado intervalo definido através de um tamanho padrão aceitável. Posteriormente, para cada inseto detectado é feita uma conversão de imagem RGB para HSV, e aplicado um cálculo de média. Por último, calcula-se a distância de Mahalanobis⁸ para extrair dados das imagens e compará-los como os dados obtidos de fotos reais das espécies. Assim, os insetos são selecionados, classificados e identificados conforme demonstrado na Figura 21 (c) e (d).

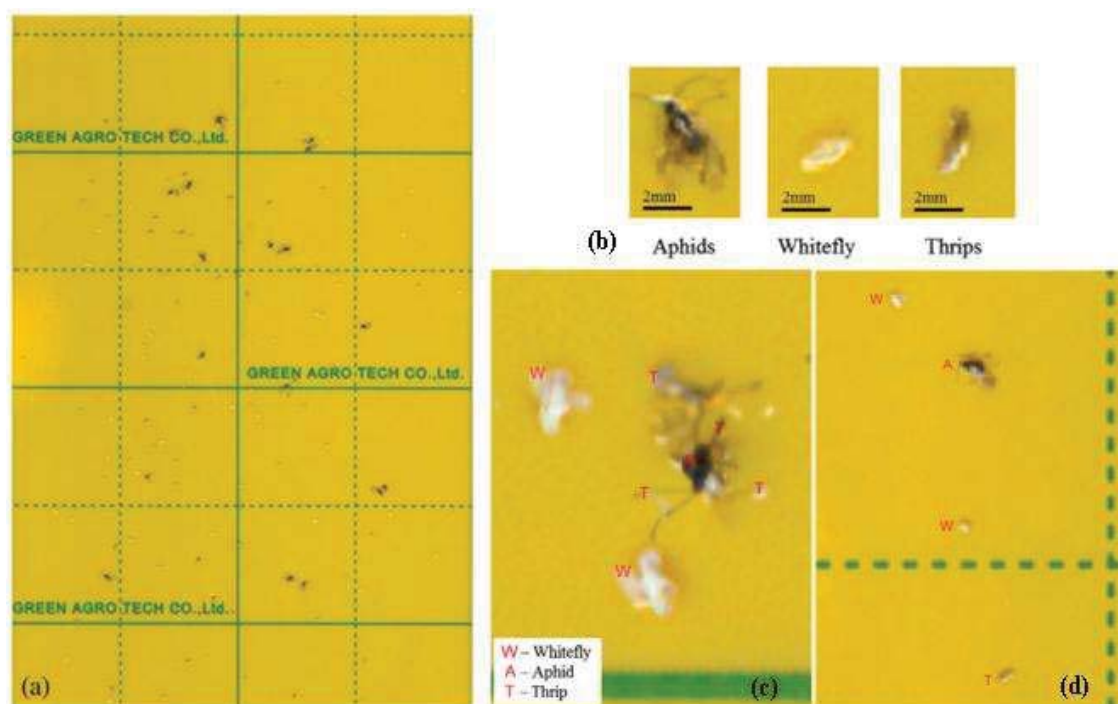


Figura 21 – Demonstração de funcionamento do processo de identificação [24].

Segundo Chunlei Xia *et al.* [24], em comparação com a resultados de contagens feitas por humanos, foi proposto um método de alto desempenho na identificação de espécies e detecção de pequenos insetos utilizando imagens de baixa resolução. De acordo com os

⁷ Segmentação cujo objetivo é encontrar linhas nas imagens. Baseia-se em uma analogia de “inundação” [33].

⁸ Medida de distância que procura buscar similaridade entre uma amostra desconhecida e uma conhecida.

autores, é uma solução viável para uso em larga escala em dispositivos portáteis, pois requer pouco consumo de energia e pouco espaço para armazenamento de dados.

3.4. AGRICULTURE PEST CONTROL USING COMPUTER VISION TECHNIQUE

Conforme Wadhai *et al.* [25], a utilização excessiva de pesticidas é extremamente prejudicial ao meio ambiente, e a identificação precoce de pestes é necessária para que seja possível diminuir o uso destes tratamentos nas plantações. Para tal, o trabalho propõe um sistema de detecção de pragas em estufas utilizando processamento de imagens e redes neurais exclusivamente aplicável aos tipos de pragas: *Trialeurodes vaporariorum* e pulgões.

O processo de identificação inicia com a coleta de imagens através de 20 câmeras distribuídas pela estufa conforme Figura 22 (a). Estas câmeras aplicam um zoom de 20X sobre as folhas para melhorar o foco nos insetos (Figura 22 (b)). Na etapa de pré-processamento, as imagens são redimensionadas e recebem diferentes técnicas de processamento para cada espécie de praga em análise. Para os *Trialeurodes vaporariorum*, por exemplo, a imagem RGB é convertida em escala de cinza. No caso dos pulgões, a imagem RGB é convertida para os padrões HSV e YCBCR.

O procedimento seguinte é a extração de características, eliminando conteúdo sem importância da imagem. Além disso, extrai-se a média, o desvio padrão e o contraste, usados como entrada em uma rede neural artificial (RNA). Essa RNA é uma solução computacional desenvolvida pelos autores, que recebeu treinamento utilizando bases de referência com exemplos de imagens afetadas e imagens não afetadas pela praga, como mostra a Figura 22 (c) e (d).

Por fim, na etapa de identificação, o software apresenta a saída da RNA, indicando qual espécie está agindo na planta.

Na conclusão do trabalho, os autores relatam que a detecção de pestes deve ser feita em estágios iniciais de infestação, quando isso é realizado existe uma redução no uso de pesticidas em até 80%. Quanto ao método, mostrou-se confiável e eficiente para detecção rápida de pestes, também se mostrou rentável e simples.

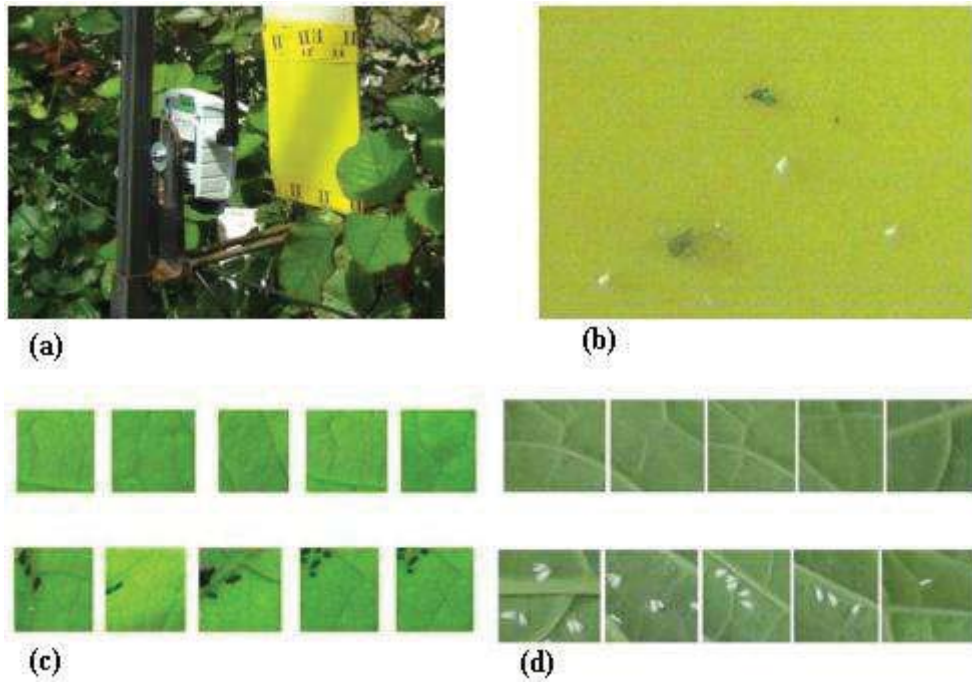


Figura 22 – Processo de captura e imagens para treinamento da rede neural [25].

4. MATERIAIS E MÉTODOS

Esta seção demonstra os recursos e os passos empregados para concepção e implementação do software proposto.

4.1. ÁREA DE ESTUDO

O desenvolvimento deste trabalho é apoiado pela Embrapa Trigo. A instituição situada em Passo Fundo é uma das 47 unidades descentralizadas da Empresa Brasileira de Pesquisa Agropecuária (Embrapa) [26]. Ela é responsável pelo fornecimento dos materiais necessários para obtenção das imagens utilizadas durante o desenvolvimento do sistema.

Para a definição do método de contagem e classificação, a etapa de aquisição de imagens é primordial, uma vez que é preciso estabelecer uma configuração padronizada para aquisição das amostras e redução de vieses. Este passo, com exceção da digitalização, é baseado na metodologia manual de contagem utilizado na Embrapa Trigo de Passo Fundo.

O processo inicia com a coleta de amostras por pesquisadores em campo, colocando os pulgões em tubos de ensaio contendo solução aquosa. Em seguida, acontece a transferência destas amostras para placas de Petri, circulares e transparentes, com 14 cm de diâmetro. Na sequência, ocorre a digitalização das placas com os materiais coletados, conforme Figura 23.

A digitalização é feita em scanner comum de mesa, na resolução de 1200 dpi, dimensões de 7000x7000 pixels e imagem colorida. A metodologia foi desenvolvida exclusivamente para amostras da espécie *R. padi*, utilizada nesse trabalho.

4.2. METODOLOGIA PROPOSTA

Como citado no Capítulo 2, o processamento de imagens digitais é composto por alguns passos fundamentais. A Figura 24 apresenta um fluxograma que detalham os passos a executados na metodologia de contagem e classificação dos afídeos. As subseções seguintes descrevem cada um desses passos.

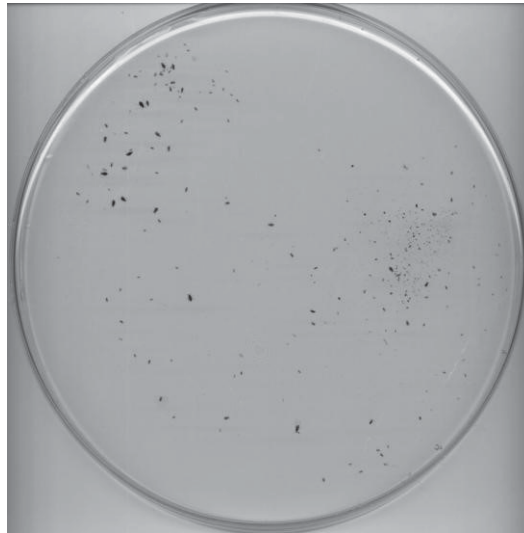


Figura 23 – Exemplo de imagem disponibilizada.



Figura 24 – Sequência metodológica.

4.2.1. Passo 1: Coleta de Imagens

Apesar de existir um método padrão de coleta e armazenamento das amostras de pulgões em laboratório, é necessário também definir diretrizes de coleta das imagens digitais. Estas regras permitem que o método seja destinado a entrada do software proposto, não contemplando todas as diferentes formas de coleta de imagens. Por exemplo, desenvolver um

método de processamento de imagens baseado em imagens obtidas por câmeras em preto e branco é diferente do desenvolvimento com imagens oriundas de câmeras multiespectrais.

Os materiais necessários para a padronização da coleta de imagens para essa proposta são um scanner e placas de Petri conforme citados na Seção 0. Nenhuma marca ou modelo específico de scanner foi definido para esse trabalho. Porém, recomenda-se utilizar um scanner de mesa para todas as coletas, uma vez que ele permite a disposição da placa de Petri sobre a área de captura. Outro requisito básico é a área de captura que deve ser equivalente ao tamanho de uma placa de Petri. Quanto a resolução do scanner, deve-se manter um padrão para todas as coletas. Para esse trabalho, optou-se pela captura de alta qualidade, em 1200 dpi.

A placa de Petri tem aproximadamente 1cm de altura. Com isso, ao fechar a tampa do scanner, pode ocorrer a entrada de luz externa que afeta a qualidade da imagem. Como a etapa de classificação depende de pequenos detalhes presentes na imagem foi necessário resolver este problema utilizando um caixa de vedação de luz. A Figura 25 demonstra o equipamento desenvolvido em impressora 3D nas dimensões de 15cmX15cm que veda totalmente a entrada de luz externa na placa.



Figura 25 – Caixa de vedação de luz.

A Figura 26 demonstra um comparativo, onde (a) foi digitalizada sem o uso da caixa de vedação, resultando em uma imagem muito mais escura que (b), digitalizada com o uso da caixa. A Figura 27 apresenta dois recortes de imagens aumentadas sendo (a) sem o uso da caixa de vedação e (b) utilizando. Nota-se também que detalhes das asas dos pulgões ficaram mais visíveis com o uso do equipamento.

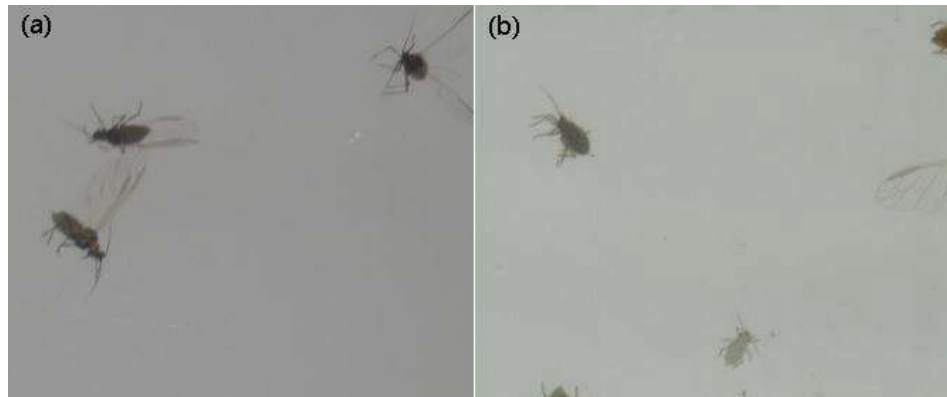


Figura 26 – Comparação do método sem e com uso de uma caixa de vedação.

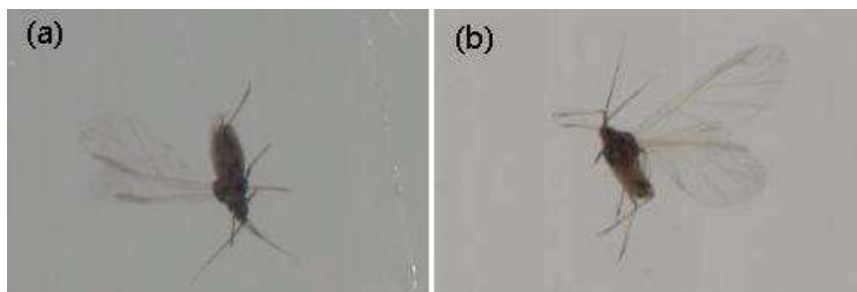


Figura 27 – Comparação em zoom, mostrando detalhes do indivíduo alado quando utilizada a caixa de vedação.

Também na caixa de vedação foram feitas pequenas marcações circulares em todos os cantos, como mostra a Figura 28. Os círculos possuem 0,5cm de diâmetro e foram utilizados como referenciais para cálculos de tamanho dos objetos presentes na imagem, úteis na conversão da medida de pixels para milímetros. Mais detalhes podem ser vistos no Passo 5: Identificação de potenciais contornos.



Figura 28 – Referencias de medida para elementos da imagem.

4.2.2. Passo 2: Segmentação macro

Para o processamento das imagens, adotou-se alguns métodos que visam extrair das imagens os objetos de interesse e eliminar possíveis sujeiras como exoesqueletos, utilizando recursos da OpenCV.

O processo tem início com a segmentação macro. Ao receber uma imagem digitalizada, sabe-se que os dados fora do perímetro da placa de Petri não interessam. Sendo assim, é necessário aplicar um método de exclusão dessas áreas externas a placa. A Figura 29 representa esta situação, onde a parte em preto foi considerada desnecessária (direita) em relação à imagem de entrada (esquerda).

Para tanto, utilizou-se a transformada de Hough para detecção de círculos [27], considerando o formato circular da placa de Petri. No OpenCV o recurso é utilizado através da função: `HoughCircles`.



Figura 29 – Resultado pós processamento.

4.2.3. Passo 3: Segmentação de pequenos elementos

Esta etapa também realiza segmentação, porém com objetivo de excluir exúvias (exoesqueletos eliminados pelos insetos a medida que crescem). Restos de vegetais que, na maioria dos casos, têm tonalidade de cor mais clara que os insetos, acabam sendo removidos também nesta etapa, por um algoritmo de limiarização aplicado à região de interesse.

A limiarização ou *thresholding* é considerada uma etapa básica e fundamental para segmentação de imagens [8]. Tem como princípio separar regiões de uma imagem que apresentam duas classes como, por exemplo, o fundo e um objeto. A função se baseia em definir um valor de corte (T) para uma imagem que, ao ser processada, considerará todos os valores de pixels acima ou iguais a T como valor um, e os valores menores como zero.

Por meio de calibração, definiu-se a constante de intensidade $T = 110$ para essa abordagem. A Figura 30 apresenta o resultado (b) desta etapa em uma área ampliada de uma imagem de entrada (a). Assim, eliminou-se possíveis falsos positivos, já que exúvias tem um grau alto de semelhança com os insetos.

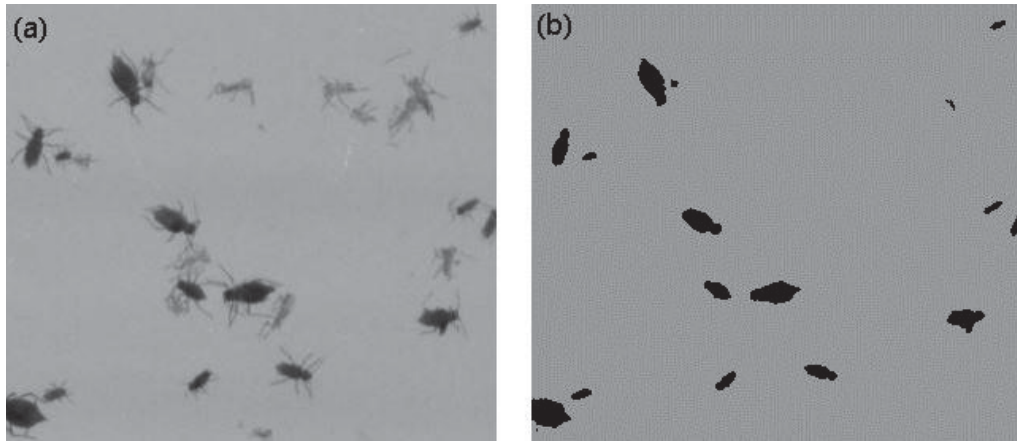


Figura 30 – Resultado da limiarização.

4.2.4. Passo 4: Reconhecimento de contornos

Na quarta etapa do processo, são reconhecidos os contornos dos objetos encontrados na imagem por meio de técnicas de detecção de bordas. Para tanto, adotou-se a função `findContours` presente na biblioteca `OpenCv`, que utiliza código de Freeman para armazenar os contornos [28]. Basicamente, todos os objetos que resultaram do Passo 3 são encontrados na imagem.

4.2.5. Passo 5: Identificação de potenciais contornos

Após identificar cada contorno, avalia-se se cada região de interesse destacada pode ser considerada um inseto. Para tal, as dimensões em milímetros de cada contorno são comparadas com valores definidos que consideram um intervalo que contempla desde uma ninfa (menor área possível) até um pulgão alado (maior área possível).

Caso o contorno seja considerado um potencial inseto, ele será extraído da imagem para que siga para o procedimento de classificação. Esta extração é feita por meio de um recorte retangular do contorno. Cada recorte é enviado separadamente para o processo de classificação.

Também nesta etapa realiza-se a mensuração do contorno, ou seja, são extraídos os valores de comprimento, largura, área e perímetro. Todos estes valores estão em pixels, sendo assim necessário fazer a conversão para milímetros. Para realizar a conversão são utilizados os referenciais de medida citados no Passo 1: Coleta de Imagens, que possuem valores em milímetros conhecidos.

Basicamente, calcula-se a proporção do valor do diâmetro do círculo de referência em pixels para seu valor conhecido, em milímetros. A Equação (2) demonstra o cálculo onde d é o diâmetro do círculo encontrado em pixels e y o tamanho do referencial conhecido em milímetros.

$$x = \frac{d}{y} \quad (2)$$

4.2.6. Passo 6: Classificação

Na etapa de classificação o objetivo é rotular o inseto segundo suas características em: ninfa, adulto alado e adulto áptero. As ninfas são insetos na fase inicial de vida, possuem um tamanho menor comparado aos adultos, e têm morfologia menos arredondada. Os adultos ápteros são insetos na fase adulta, identificados pelo tamanho maior, e morfologia mais arredondada. Já os adultos alados diferem-se dos ápteros por possuírem asas. Através desta etapa é possível obter o número de insetos em cada fase de vida.

Assim cada área reconhecida no Passo 5 é avaliada por meio de técnicas de aprendizado de máquina. Para tanto, faz-se a comparação de características da imagem corrente com um banco de dados de positivos e falsos positivos para classificar o objeto em análise.

Para gerar este banco de dados, uma coleção de imagens previamente classificada por especialistas foi submetida ao treinamento com recursos de Inteligência Artificial utilizando a biblioteca TensorFlow 1.4. Esta biblioteca utiliza algoritmos de *deep learning* para detectar e decifrar padrões e correlações, análogo (mas não igual) à forma como humanos aprendem e raciocinam. Após esse processamento, gera-se como resultado uma base de conhecimento, útil para analisar novas imagens.

Para realizar o treinamento como o TensorFlow é necessário ter um classificador. Este contém os métodos utilizados para extrair as características das imagens, como por

exemplo, extração de características com uma rede neural convolucional, camadas totalmente conectadas, softmax, etc. Muitos classificadores são disponibilizados pela comunidade de forma gratuita contendo modelos que funcionam muito bem para várias categorias de imagens, além de já estarem prontos evitando assim horas de desenvolvimento para tal. Portanto nesta etapa do projeto, optou-se por utilizar um classificador pronto chamado o Inception V3 criado pela ImageNet [29].

Para obter o conjunto de imagens para o treinamento, desenvolveu-se um software auxiliar, denominado CropAphid, desenvolvido em linguagem Java. A partir desta ferramenta, fez-se recortes de imagens de entrada de potenciais afídeos, e criou-se um banco de dados primário, tomando por base as etapas de processamento de imagens apresentadas anteriormente. Para gerar os recortes foram utilizadas aproximadamente 100 amostras similares a Figura 31.



Figura 31 – Exemplo de imagem utilizada para extração de amostras

Destas imagens, obteve-se aproximadamente 50000 recortes de 100x100 pixels. Estes recortes foram submetidos a classificação por entomólogos da Embrapa Trigo, considerando quatro classes: ninfas, ápteros, alados e falsos. Para a aprendizagem de máquina, foram utilizadas 2475 imagens da classe alados (Figura 32), 4173 da classe ápteros (Figura 33), 15896 da classe ninfas (Figura 34) e 4835 da classe falsos (Figura 35). Após treinamento da solução, obteve-se uma inteligência de máquina com precisão de 98,10%.

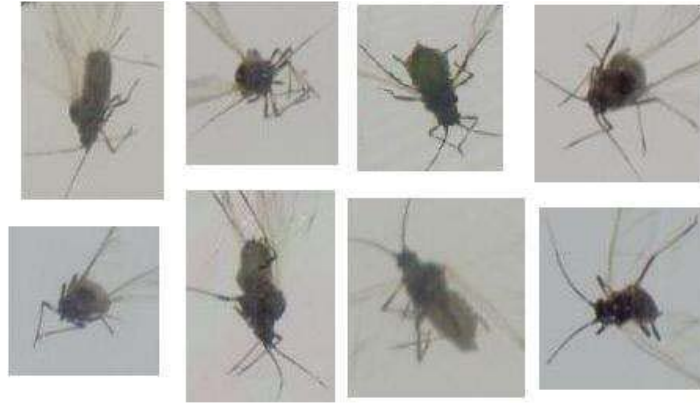


Figura 32 – Exemplos de imagens da classe Alados.



Figura 33 – Exemplos de imagens da classe Ápteros.



Figura 34 – Exemplos de imagens da classe Ninfas.

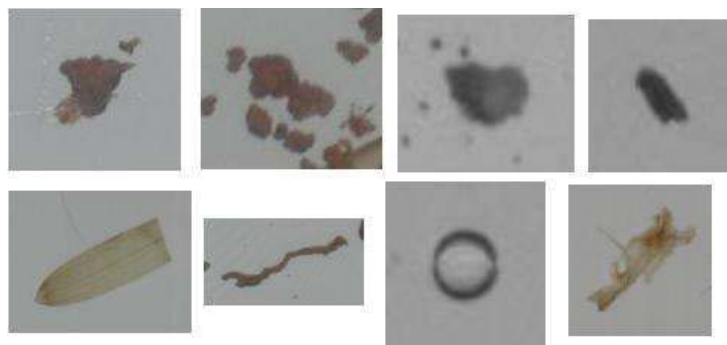


Figura 35 – Exemplos de imagens da classe Falsos.

4.2.7. Passo 7: Apresentação dos resultados

Os resultados são apresentados em uma interface gráfica construída em JavaFX conforme Figura 36. Basicamente as imagens pós processamento são apresentadas na tela para que o usuário possa analisar o resultado. Também é gerado um JSON⁹ contendo os dados da análise, este poderá ser utilizado futuramente para inserir os resultados em outro sistema, por exemplo, um software estatístico. Outra forma de apresentação é um arquivo CSV¹⁰, este contém dados de fácil visualização humana.

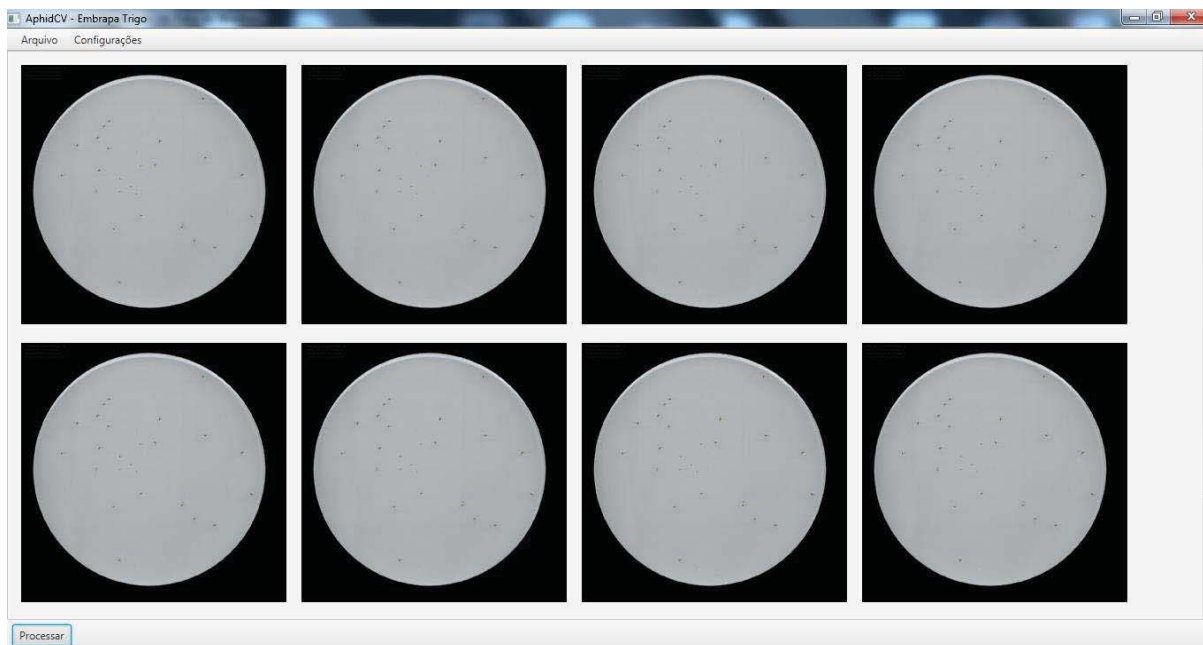


Figura 36 - Apresentação dos resultados.

⁹ Notação de Objetos JavaScript. Formato texto e completamente independente de linguagem, utilizado para troca de dados entre sistemas diferentes.

¹⁰ Arquivo de valores separados por vírgula, muito utilizado em planilhas, por exemplo, Excel.

4.3. FERRAMENTAS

Para desenvolver o software que implementa a metodologia proposta, definiu-se o uso da linguagem Java e das bibliotecas OpenCV e TensorFlow.

Java é uma das linguagens de programação mais utilizadas no mundo. É baseada no paradigma de orientação a objetos, possui características como segurança, portabilidade, robustez, multithreaded¹¹, e de arquitetura neutra, interpretada, distribuída e dinâmica [30].

As bibliotecas OpenCV e TensorFlow já foram previamente apresentadas, na Seção 2.7.1 e na Seção 2.7.3, respectivamente.

O próximo Capítulo apresenta o software desenvolvido, intitulado AphidCV, adaptado para identificar indivíduos da espécie *R. padi*.

¹¹ Permite ao desenvolvedor escrever programas que façam muitas coisas ao mesmo tempo.

5. APHIDCV

A definição da metodologia e das ferramentas mencionadas no Capítulo 4 permitiram a concepção e o desenvolvimento da aplicação proposta por este trabalho, denominada AphidCV.

O sistema foi projetado para a plataforma Windows, de fácil utilização dos usuários finais, sem a necessidade de conhecimentos avançados em informática ou a realização de configurações complexas.

A aplicação tem como objetivo ler as imagens de um diretório e processá-las de acordo com os passos citados na Seção 4.2. Após o processamento, o software apresenta os resultados através das imagens processadas e dados no formato CSV e JSON. Além disso, realiza recortes de pulgões e calcula medidas, a fim de coletar dados para melhorar futuramente o sistema de reconhecimento.

A instalação é realizada por meio do arquivo executável AphidCV.exe. Ao executar, basta selecionar a pasta onde o sistema deve ser instalado. Ao fim, um ícone é criado na área de trabalho.

Ao executar o software através deste ícone é exibida uma janela para selecionar a pasta de origem das imagens conforme a Figura 37.

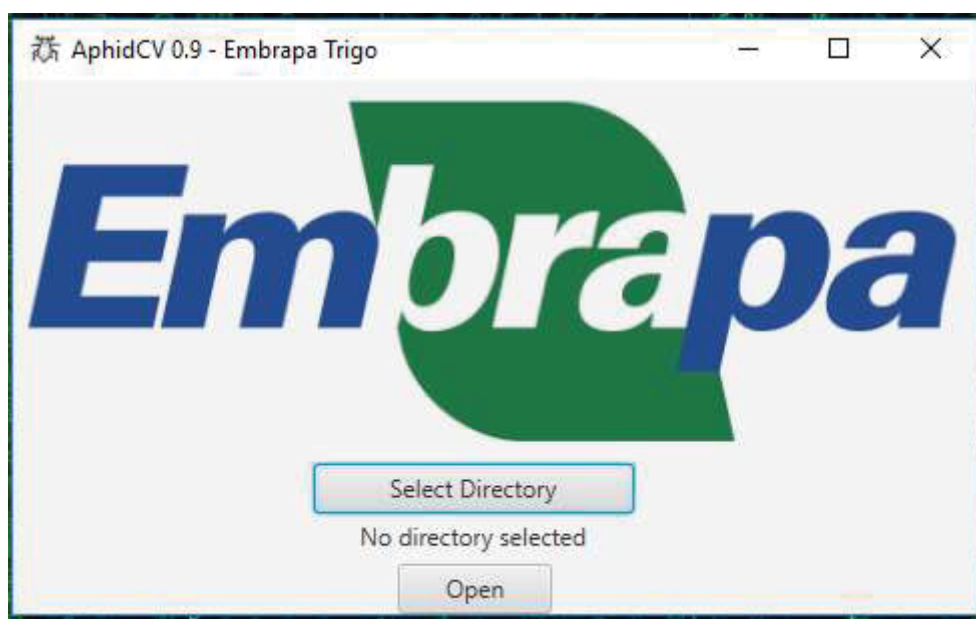


Figura 37 – Tela de seleção de pasta de origem.

Ao clicar em “Select Directory” o usuário escolhe a pasta onde estão salvas as imagens que devem ser processadas. A opção “Open” mostra todas as imagens contidas no diretório de origem, conforme Figura 38.

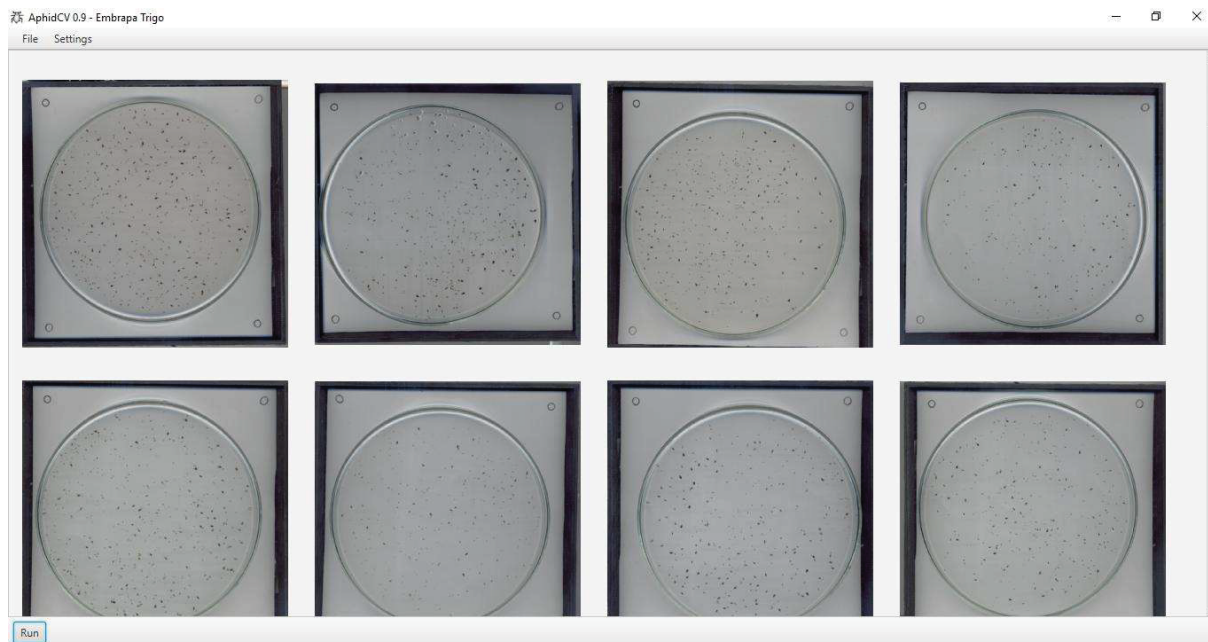


Figura 38 – Tela inicial.

Na barra superior existe um menu onde em “File” é possível obter as informações sobre os autores do software. Em “Settings” é possível realizar ajustes nas configurações do sistema caso seja necessário. O software já oferece uma configuração padrão de valores para análise da espécie *R. padi*, estes padrões foram obtidos por meio de calibrações realizadas durante o desenvolvimento do sistema. A Figura 39 demonstra os parâmetros que podem ser alterados no sistema, sendo:

- KValue: a limiar utilizada para a binarização das imagens;
- cropRect: o tamanho em pixels das micro imagens recortadas;
- minSize: tamanho mínimo em milímetros de um objeto considerável;
- maxSize: tamanho máximo em milímetros de um objeto considerável;
- maxLengthNymph: comprimento máximo de uma ninfa em milímetros. Este valor é utilizado para auxiliar o classificador a diferenciar ninfas de ápteros.
- referenceSize: tamanho em milímetros dos referenciais circulares;
- printTextToImage: escreve os resultados na imagem processada.

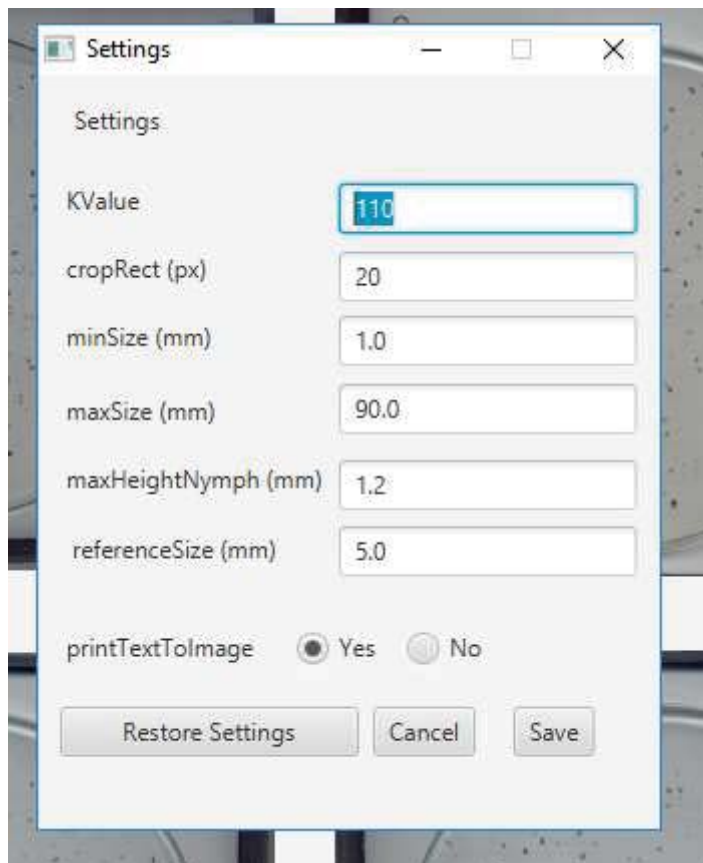


Figura 39 - Tela de configurações

Ao clicar em “Retore Settings”, todas as configurações padrões são restabelecidas.

Para iniciar o processamento das imagens, basta o usuário clicar no botão “Run”, localizado no canto esquerdo inferior da tela inicial. Ao iniciar o processo, uma barra de progressão apresenta o andamento da análise.

Ao final do processamento, o sistema apresenta as imagens analisadas em uma janela semelhante à tela inicial, conforme Figura 36, permitindo ampliar a visualização de cada imagem para exibição de detalhes.

Ao abrir uma imagem processada, pode-se ter acesso às interpretações do software quanto a classificação e medidas dos insetos individualmente. Os insetos, cujo contorno é roxo, foram classificados como “alados” (Figura 40 a); vermelhos como “ápteros” (Figura 40 b); e amarelos como “ninfas” (Figura 40 c). Ao lado de cada indivíduo identificado são apresentados os valores de área (A) em milímetros, perímetro em milímetros (P), comprimento em milímetros (L) e probabilidade de acerto (valor percentual).

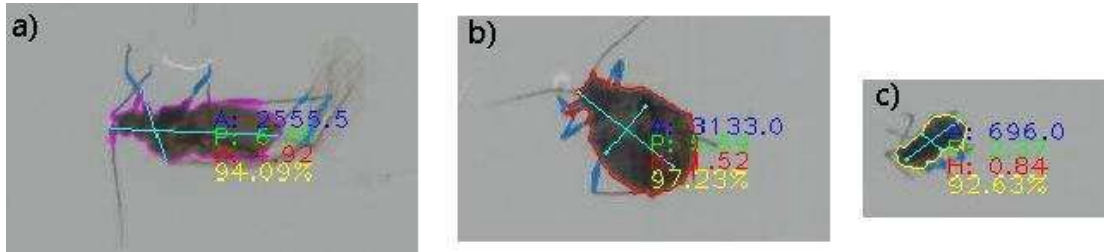


Figura 40 – Detalhes da imagem processada.

Demais resultados podem ser encontrados em “Caminho da Pasta de Origem/result_data_da_análise, conforme Figura 41.

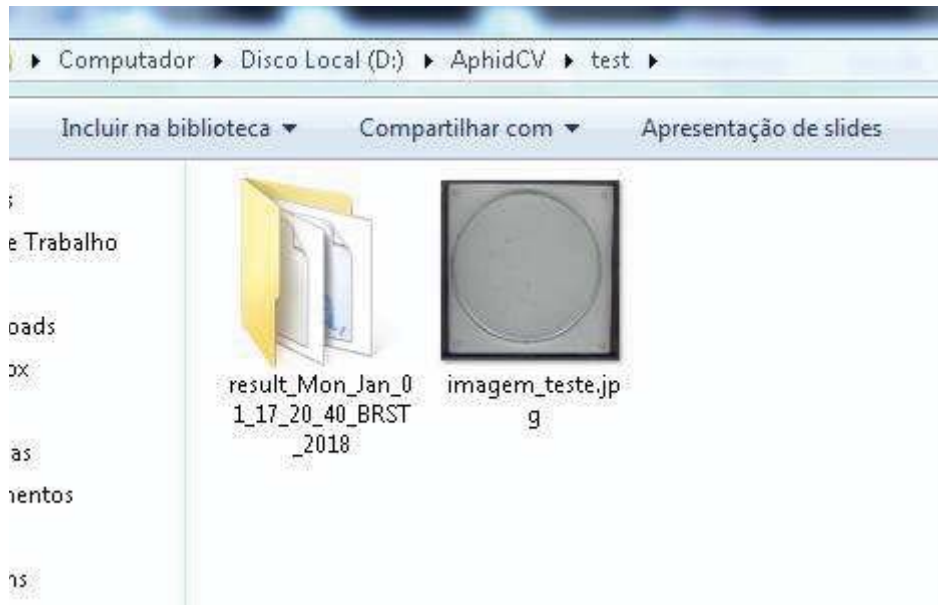


Figura 41 – Exemplo de resultado.

Ao acessar este diretório, alguns arquivos são apresentados (Figura 42):

- Pasta csv: contém um arquivo .csv para cada imagem analisada, cujo conteúdo são os dados (Specie, Type, Probability (%), Length (mm), Weight, Perimeter (mm), Area) de cada inseto encontrado;
- Pasta doubtful: contém imagens recortadas da análise corrente cuja probabilidade de acerto no reconhecimento foi menor que 80%. Estes dados serão utilizados no decorrer do tempo para treinar novamente a base e melhorar sua precisão;

- Amostra Teste AphidCV.csv: csv cujo conteúdo são os dados (ImageLabel, QuantityNymphs, QuantityWinged, QuantityWingless, QuantityTotal, WeightTotal) por imagem analisada;
- Amostra Teste AphidCV.json: arquivo com dados de todas as imagens, incluindo os detalhes de cada inseto. Utilizado para alimentar softwares terceiros;
- imagem_teste.jpg: resultado da imagem processada;
- log.txt: log dos eventos ocorridos na análise.

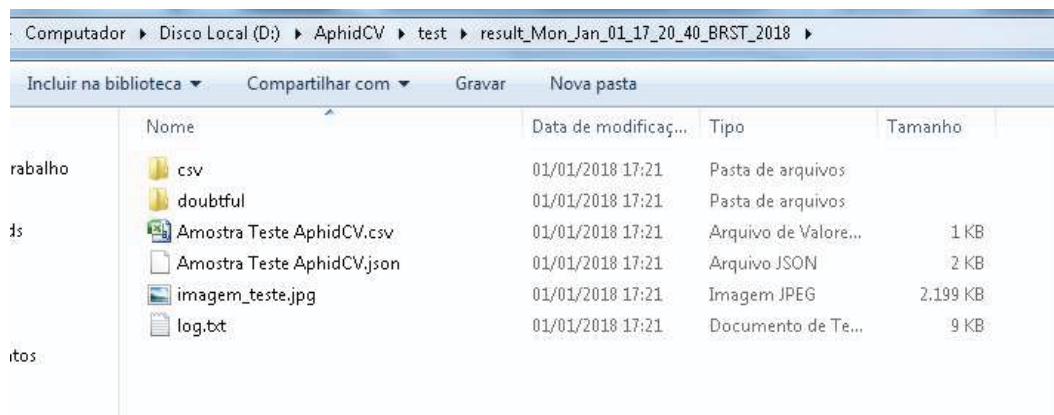


Figura 42- Resultados da análise.

A Figura 43 apresenta um exemplo de relatório no formato CSV de uma única imagem. O objetivo é detalhar cada inseto encontrado com sua espécie, classe, probabilidade e as mensurações. O software gera este arquivo CSV para cada uma das imagens da coleção de amostras utilizadas na análise.

	A	B	C	D	E	F	G	H
1	Specie	Type	Probability(%)	Lenght(mm)	Weight	Perimeter(mm)	Area	
2	Rhopalosiphum padi	Nymph	76.49	1.35	1	5.25	1283.5	
3	Rhopalosiphum padi	Nymph	99.98	0.71	1	2.11	282.5	
4	Rhopalosiphum padi	Nymph	98.58	0.65	1	2.55	340.0	
5	Rhopalosiphum padi	Nymph	81.88	0.54	1	1.86	186.5	
6	Rhopalosiphum padi	Nymph	99.65	0.8	1	2.45	469.0	
7	Rhopalosiphum padi	Nymph	84.32	0.68	1	1.77	295.5	
8	Rhopalosiphum padi	Nymph	73.75	1.12	1	3.63	995.0	
9	Rhopalosiphum padi	Nymph	50.75	0.48	1	1.13	112.5	
10	Rhopalosiphum padi	Nymph	64.59	1.4	1	4.37	1595.0	
11	Rhopalosiphum padi	Nymph	62.36	1.43	1	4.84	1583.5	
12	Rhopalosiphum padi	Nymph	94.46	0.91	1	2.99	586.5	
13	Rhopalosiphum padi	Nymph	88.7	1.09	1	4.58	972.0	
14	Rhopalosiphum padi	Nymph	98.07	0.83	1	2.63	481.0	
15	Rhopalosiphum padi	Nymph	67.56	0.97	1	2.43	633.0	

Figura 43 – Exemplo de relatório por placa

Para realizar a validação do sistema desenvolvido foram coletadas 40 amostras, estas, foram submetidas a contagem manual, ou seja, o método tradicional de contagem empregado hoje na Embrapa Trigo. Os resultados foram armazenados e posteriormente as 40 amostras foram digitalizadas para realizar a contagem através do AphidCV e do CropAphid. Os detalhes dos resultados dessa comparação serão apresentados a seguir, no capítulo 6.

6. RESULTADOS E DISCUSSÕES

Ao finalizar o desenvolvimento do software proposto, foi realizada a etapa de validação, comparando os dados obtidos através da contagem manual e AphidCV. Também foi realizada uma análise utilizando o software auxiliar CropAphid, cujo objetivo foi contar e classificar manualmente os recortes feitos pelo software.

O teste do software foi realizado utilizando 40 amostras e considerou a contagem, classificação e tempo de análise. Inicialmente as amostras foram contadas e classificadas manualmente por um laboratorista da Embrapa Trigo. Após, foram digitalizadas e submetidas a contagem e classificação através do CropAphid. Por último a coleção de imagens foi processada no AphidCV.

Para a análise estatística, foi utilizado primeiramente o teste Shapiro-Wilk para avaliar a normalidade dos dados. Todos os conjuntos foram considerados não normais com $n=40$ com nível de significância de 5%. Sendo os dados não normais, utilizou-se para a comparação dos resultados o coeficiente de correlação de Spearman.

A sequência de comparações foi assim definida: método manual X CropAphid, método manual x AphidCV, e CropAphid x AphidCV. Todos os testes consideraram a contagem total, afim de avaliar a eficiência do método de contagem. Já a contagem de ninfas, ápteros e alados foi realizada para comparar a eficiência do método de classificação.

Os tópicos a seguir apresentam os detalhes sobre cada comparação.

6.1. MÉTODO MANUAL X CROPAPHID

Na comparação entre a contagem total o valor de R foi de 0,93488. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 44 apresenta o gráfico de dispersão.

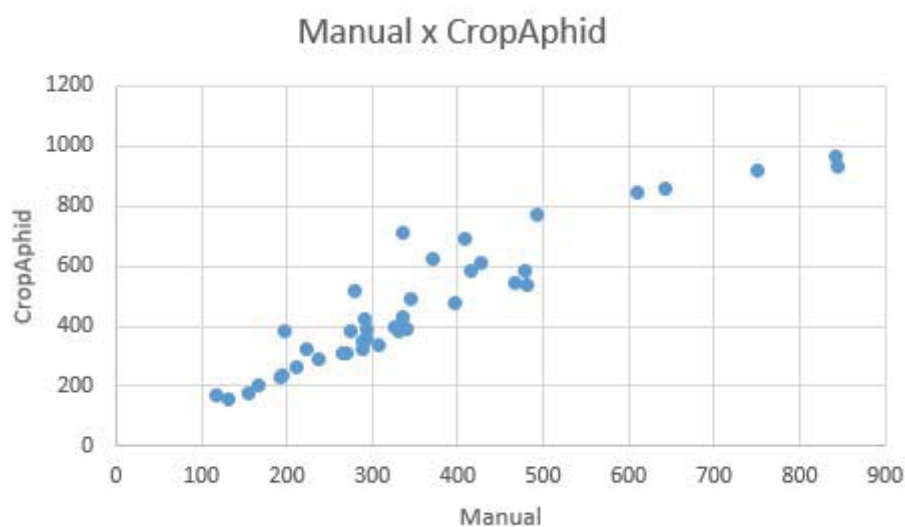


Figura 44 – Valor total Manual x CropAphid.

Na comparação entre a contagem de “ninfas” o valor de R foi de 0,90164. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 45 apresenta o gráfico de dispersão.

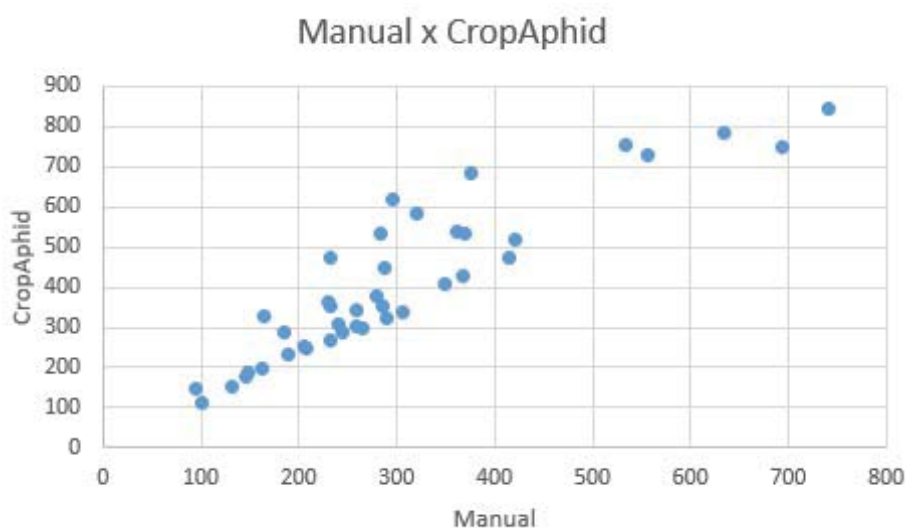


Figura 45 – Contagem de “ninfas” Manual x CropAphid.

Na comparação entre a contagem de “ápteros” o valor de R foi de 0,85097. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 46 apresenta o gráfico de dispersão

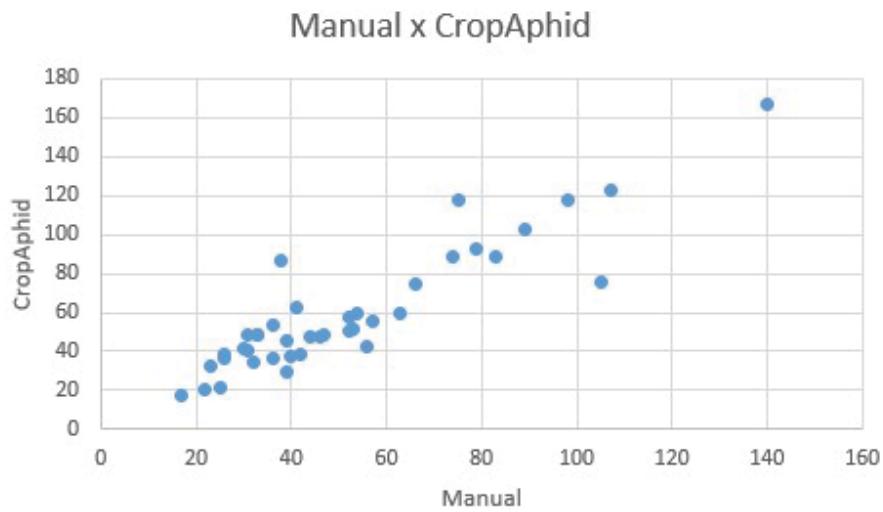


Figura 46 – Contagem de “ápteros” Manual x CropAphid.

Na comparação entre a contagem de “alados” o valor de R foi de 0,93184. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 47 apresenta o gráfico de dispersão

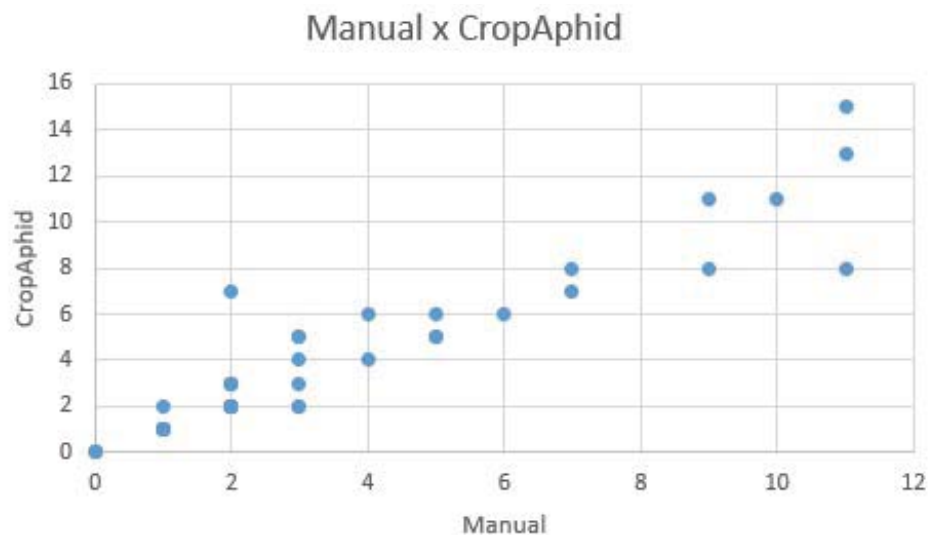


Figura 47 – Contagem de “alados” Manual x CropAphid.

O tempo necessário para realizar exclusivamente a contagem manualmente, por especialistas, foi de aproximadamente 40 horas ou 60 minutos por amostra. Já utilizando o

CropAphid, considerando o tempo de preparação e digitalização das placas, foi de aproximadamente 60 horas ou 90 minutos por amostra.

Em relação a diferença de tempo, cabe destacar que o CropAphid tende a ser mais lento que o método manual, visto que, além de realizar a digitalização e processamento da imagem, o laboratorista precisa fazer a contagem e classificação das imagens segmentadas. Esse processo se assemelha ao método manual, sendo que a única diferença é que os pulgões não estão sendo visualizados por lupas, e sim em imagens digitais. Além disso, o CropAphid captura algumas ninfas menores que um milímetro que não são vistas no método manual. Assim, o aumento de unidades também contribui no acréscimo de tempo para contagem e classificação.

6.2. MÉTODO MANUAL X APHIDCV

Na comparação entre a contagem total o valor de R foi de 0,92579. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 48 apresenta o gráfico de dispersão.

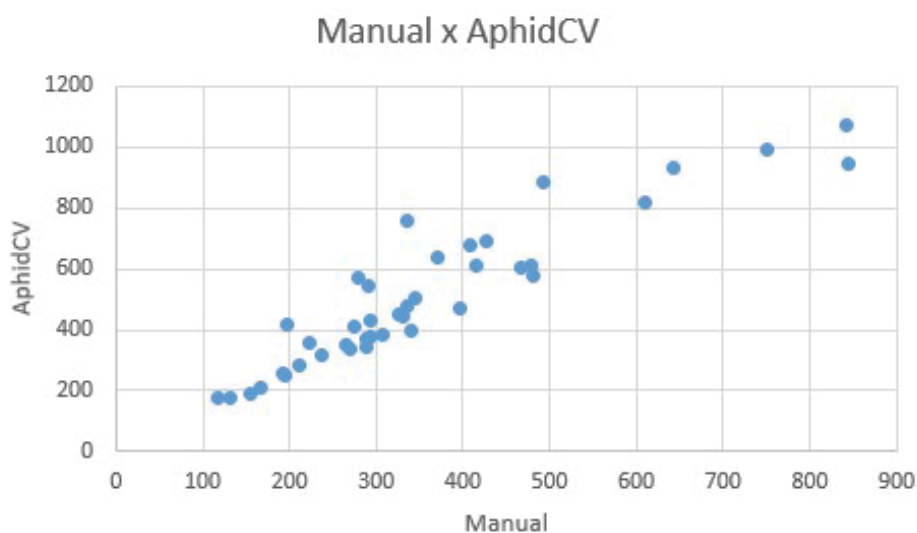


Figura 48 – Valor total Manual x AphidCV.

Na comparação entre a contagem de “ninfas” o valor de R foi de 0,89601. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 49 apresenta o gráfico de dispersão.

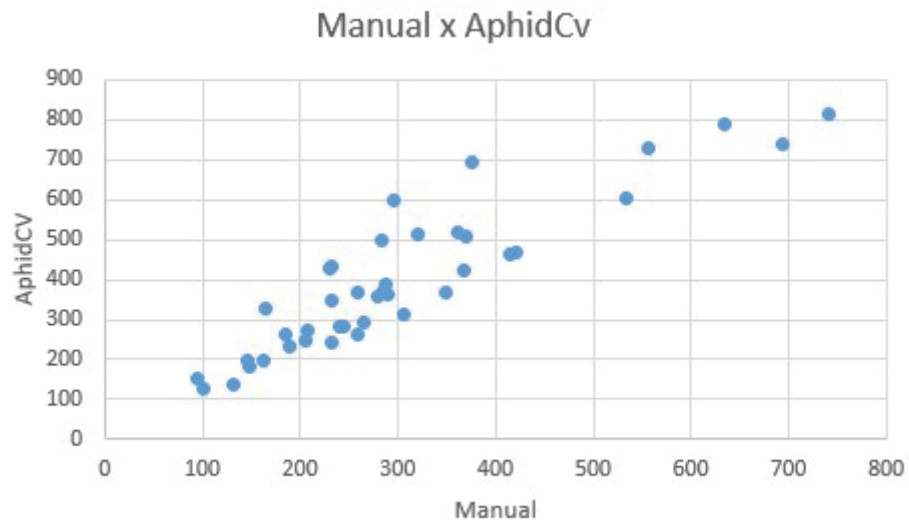


Figura 49 – Contagem de “ninfas” Manual x AphidCV.

Na comparação entre a contagem de “ápteros” o valor de R foi de 0,86138. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 50 apresenta o gráfico de dispersão.

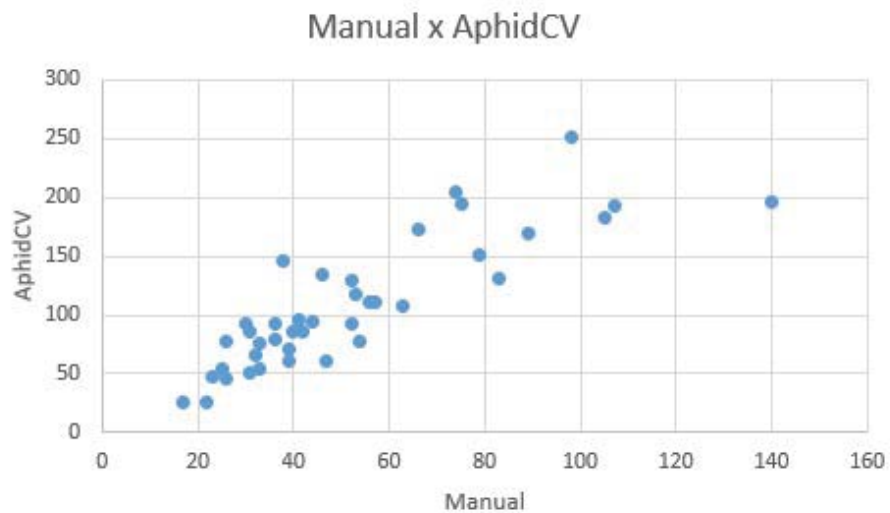


Figura 50 – Contagem de “ápteros” Manual x AphidCV.

Na comparação entre a contagem de “alados” o valor de R foi de 0,63149. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 51 apresenta o gráfico de dispersão.

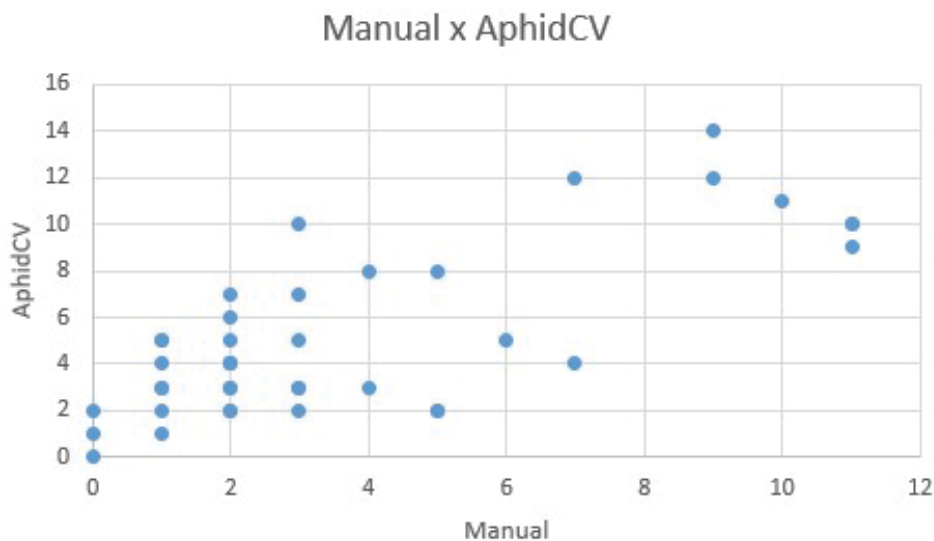


Figura 51 – Contagem de “alados” Manual x AphidCV.

O tempo necessário para realizar exclusivamente a contagem manualmente, por especialistas, foi de aproximadamente 40 horas ou 60 minutos por amostra. Já utilizando o AphidCV, considerando o tempo de preparação e digitalização das placas, foi de aproximadamente 10 horas ou 15 minutos por amostra.

6.3. CROPAHPID X APHIDCV

Na comparação entre a contagem total o valor de R foi de 0,99249. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 48 apresenta o gráfico de dispersão.

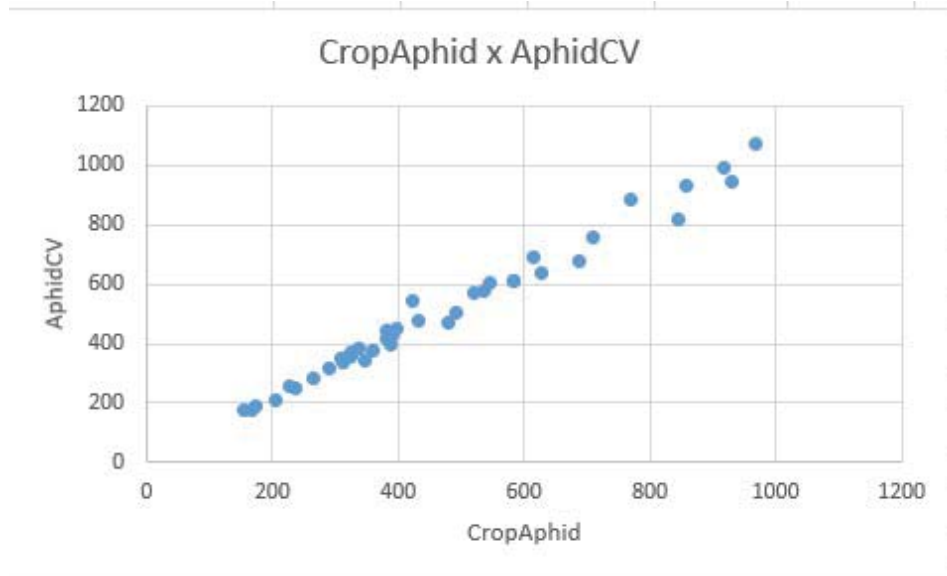


Figura 52 – Valor total CropAphid x AphidCV.

Na comparação entre a contagem de “ninfas” o valor de R foi de 0,98555. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 53 apresenta o gráfico de dispersão.

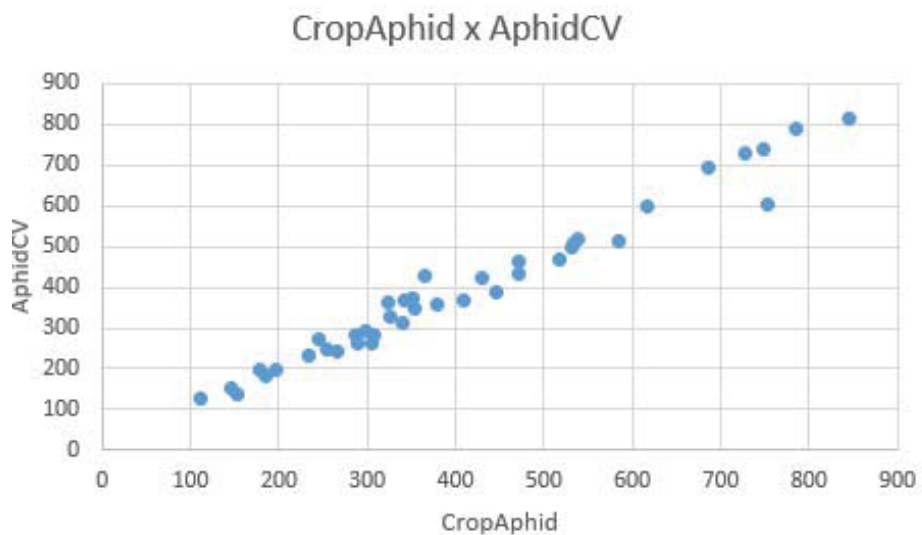


Figura 53 – Contagem de “ninfas” CropAphid x AphidCV.

Na comparação entre a contagem de “ápteros” o valor de R foi de 0,84755. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 54 apresenta o gráfico de dispersão.

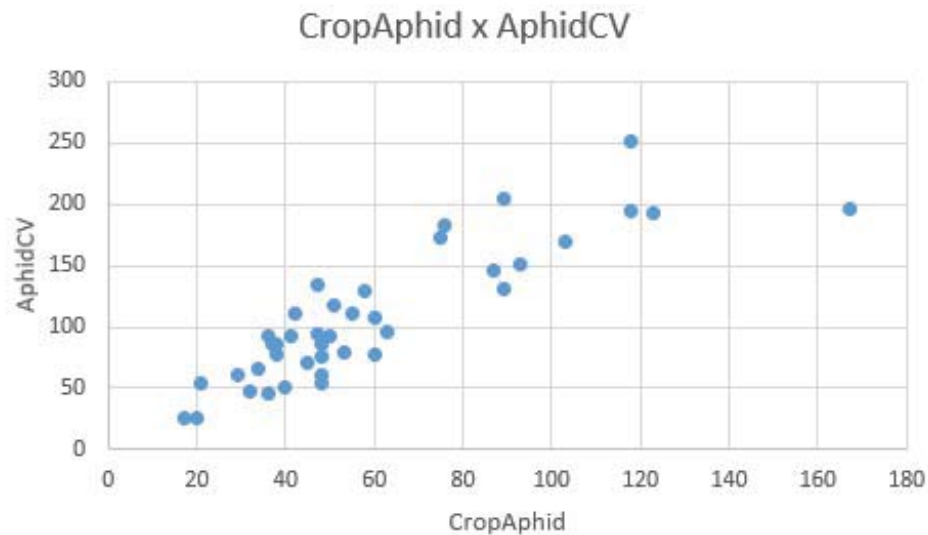


Figura 54 – Contagem de “ápteros” CropAphid x AphidCV.

Na comparação entre a contagem de “alados” o valor de R foi de 0,69146. Por padrões normais, a associação entre as duas variáveis é considerada estatisticamente significativa. A Figura 55 apresenta o gráfico de dispersão.

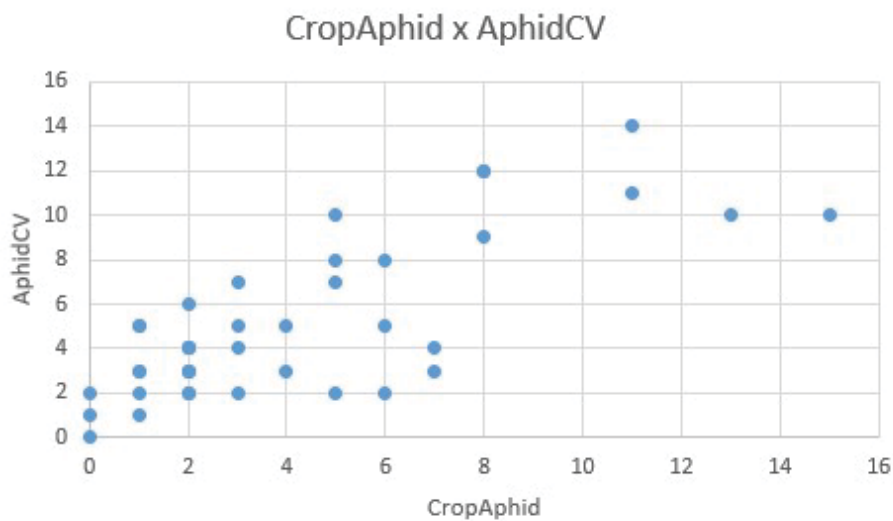


Figura 55 – Contagem de “alados” CropAphid x AphidCV.

6.4. COMPARAÇÃO DE MENSURAÇÃO

As medidas de tamanho dos insetos foram coletadas aleatoriamente de um inseto por placa, totalizando 40 mensurações. As coletas manuais foram feitas através do editor de imagens Gimp¹² [31]. Como todas as imagens utilizadas possuem um referencial conhecido, foi possível realizar através da ferramenta a conversão para uma medida em milímetros.

Para comparação dos resultados, foi utilizado o teste Shapiro-Wilk para avaliar a normalidade dos dados. Os dados foram considerados normais, para tal, foi utilizado a Correlação de Pearson.

O valor de R foi de 0,97990, indicando uma alta correlação entre as medidas manuais e medidas obtidas através do AphidCV. A Figura 56 apresenta o gráfico de dispersão das mensurações.

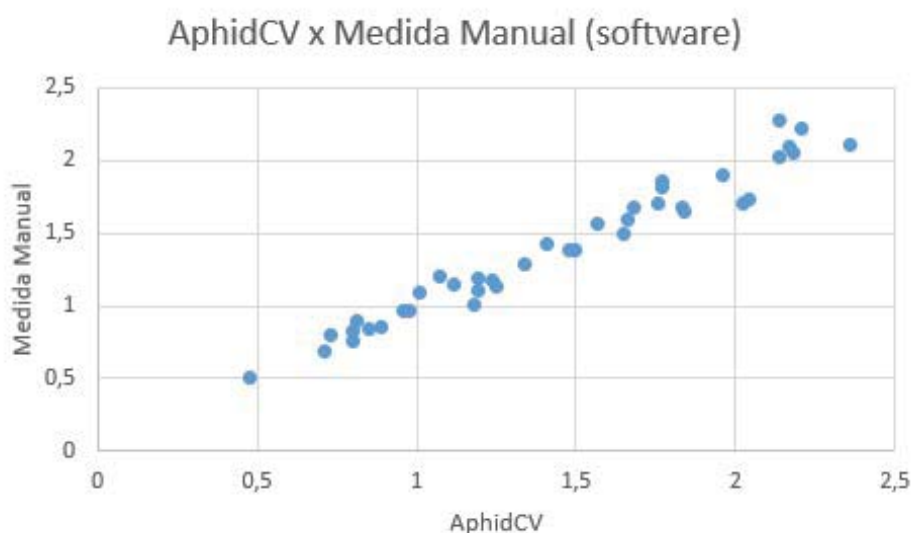


Figura 56 – Comparação de mensurações

6.5. DISCUSSÃO

Após análise dos resultados, verificou-se que o método implementado no software AphidCV mostra-se eficiente, principalmente quando são comparados os valores de contagens totais. Já a etapa de classificação mostrou-se eficiente para “ninfas” e “ápteros”, porém para a classe dos “alados”, apesar de ser considerada estatisticamente significativa, apresentou um

¹² Software de código aberto voltado para edição de imagens

valor para R baixo em relação aos demais, demonstrando que esta classe precisa de mais elementos em sua base de inteligência de máquina.

A comparação que chama a atenção é CropAphid x AphidCV. O CropAphid é software que não utiliza inteligência de máquina para realizar a classificação, ou seja, isso é feito por um humano. É possível notar o valor de R próximo a um no valor total e consideravelmente bom nas fases de vida demonstra um possível erro na contagem manual.

A provável causa são as “ninfas” extremamente pequenas (menores que 1mm) que em alguns casos não são visualizadas no método manual, porém, são vistas pelos dois softwares. Isso pode ser comprovado analisando os totais de “ninfas”, considerando todas as placas, no Manual: 12095, no CropAphid: 16329 e no AphidCV 15784. Para melhor averiguação desde dado, deveriam ser realizadas algumas comparações entre contagens manuais realizadas por diferentes pessoas.

O maior ganho da utilização do software, sem dúvida, está no tempo gasto para realizar as análises. A Figura 57 apresenta o gráfico com a comparação do tempo médio de análise por placa entre os três métodos. Nota-se que o AphidCV é aproximadamente quatro vezes mais rápido que a metodologia utilizada hoje. Cabe lembrar que o tempo apresentado para o AphidCV e CropAphid, considera a preparação e digitalização da placa.

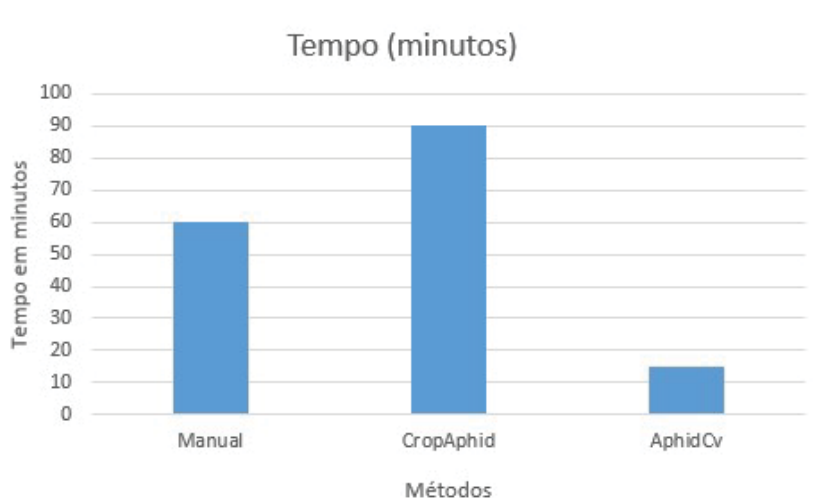


Figura 57 – Comparação de tempo

Além disso, o AphidCV após ter as imagens digitalizadas e ter a execução iniciada, não depende mais de um humano, ou seja, pode ser executado fora do horário de trabalho. O software também não é suscetível ao cansaço ou fadiga, permitindo que as análises possam ser feitas nas 24 horas do dia, com a mesma eficiência para todas as imagens.

Por meio desta economia de tempo, é possível analisar quatro vezes mais placas do que é feito atualmente, baseando-se no tempo do método manual. Isso pode aumentar consideravelmente a confiabilidade dos dados.

Quanto as mensurações realizadas pelo software, houve uma alta correlação dos valores de medições feitos manualmente e pelo AphidCV. Atualmente os insetos são raramente medidos devido ao tempo necessário para tal tarefa. Por se tratar de um dado importante para os pesquisadores, o software agrega esta informação, obtendo-a de forma extremamente rápida, automatizada e consideravelmente confiável.

Por fim, os resultados obtidos neste primeiro experimento demonstraram que o software apresenta um alto grau de confiabilidade quando avaliada a contagem dos insetos. Com relação a classificação também pode-se considerar os resultados bons, porém com uma baixa confiabilidade quanto se trata de classe “alados”. Acredita-se que com o aprimoramento do software na classificação desta categoria, o mesmo pode se tornar extremamente confiável. Com sua evolução, também pode-se atingir níveis de acerto equivalentes ou maiores que os humanos, além de ser mais rápido e apresentar informações como, por exemplo, o tamanho de cada inseto.

7. CONCLUSÃO

Este trabalho apresentou o embasamento e a definição de uma metodologia para automatizar a contagem e a classificação de afídeos da espécie *R. padi* utilizando métodos de processamento de imagens, visão computacional e *deep learning*. Para tanto, foi gerado o software AphidCV, responsável por aplicar a metodologia proposta.

Ao final do projeto, considerando a alta correlação obtida nos testes com o software, pode-se concluir que é possível utilizá-lo de forma prática na contagem, classificação e mensuração de afídeos da espécie *R. padi*.

Em comparação com o método manual utilizado em laboratório, pode-se destacar que o método automatizado diminui o tempo de contagem por placa, faz a classificação a partir de uma única contagem, agrega dados que antes não eram contabilizados em todas as amostras, como a morfometria de cada inseto. Além disso, ao final de cada análise, o software produz um relatório completo dos resultados, e sua modularidade permite integração com novos sistemas e sistemas de terceiros.

O software ainda necessita de alguns ajustes finos para corrigir a baixa correlação na classificação de algumas fases de vida, devido a quantidade reduzida de amostras utilizada nos treinamentos da base de inteligência artificial (em especial, para a classe de “alados”). Porém, para tanto, basta realizar periodicamente novos treinamentos utilizando novas imagens geradas pelo uso do próprio AphidCV.

A metodologia proposta e o software desenvolvido para representá-la nos leva a crer que a visão computacional pode contribuir exponencialmente para a área agrícola, em especial para a entomologia, cuja maioria das tarefas maçantes se resume em contar e classificar insetos por um especialista. Por meio do sistema, é possível aumentar o número de amostras analisadas, dar rapidez às análises, e oferecer resultados analíticos que tendem a uma maior confiabilidade.

Além de ser mais um recurso tecnológico para a Agricultura, a criação dessa ferramenta também possibilita que novas versões do modelo sejam projetadas e concebidas. Pode-se avançar, por exemplo, para uma solução que ajude tanto pesquisadores, como produtores, na tomada de decisão sobre o uso mais assertivo de recursos agrícolas e, conseqüentemente, contribuir para a proteção do meio ambiente.

O método proposto foi submetido à alguns eventos, como por exemplo, Sibgrapi 2017 na forma de *short paper*, mostra científica de pós-graduação da Embrapa Trigo 2017,

onde obteve a premiação de segundo lugar na categoria de bolsistas PIBIC-CNPq e no *International Plant Resistance to Insects Symposium 2018* ocorrido na Inglaterra.

7.1. TRABALHOS FUTUROS

Em relação a melhorias e trabalhos futuros, pode-se citar:

- A obtenção da medida de peso de cada inseto, por meio de equações matemáticas que estimam essa medida a partir de outras medidas já conhecidas;
- Adaptar as etapas de classificação e processamento de imagens para utilizar GPU e assim diminuir o tempo de análise.
- A adaptação da etapa de classificação para reconhecimento de pulgões parasitados e parasitóides;
- A realização de testes para ajustar e homologar o modelo proposto em amostras não controladas;
- A adaptação da metodologia para outras espécies de pulgões ou outros insetos;
- O desenvolvimento de uma solução embarcada para coleta de imagens em campo, utilizando dispositivos móveis, *appliances*, armadilhas, entre outros;
- Desenvolvimento de uma metodologia para que os usuários do software possam colaborar através de imagens para a melhoria do sistema de classificação.

REFERÊNCIAS

- [1] GALLO, D. *Entomologia Agrícola*. Piracicaba: FEALQ, v. 10, 2002. ISBN 85-7133-011-5.
- [2] FABRETTI KREYCI, P.; MENTEN, J. O. Seres indesejáveis. *Revista Cultivar*, n. 167, p. 03-10, Abril 2013.
- [3] ABITRIGO. *Abitrigo*. Disponível em: <<http://www.abitrigo.com.br/trigo-na-historia.php>>. Acesso em: 04 Agosto 2016.
- [4] GASSEN, D. N. Inimigos do pulgão. *Cultivar Grandes Culturas*, n. 10, novembro 1999.
- [5] MARTIN, A. E. A. Identification and counting of pests using extended region grow algorithm. *2nd Electronics and Communication Systems (ICECS)*. [S.l.]: IEEE. 2015. p. p. 1229-1234.
- [6] SALVADORI, J. R.; LESCHE, G. E. *Manejo Integrado dos Pulgões do Trigo*. 1º. ed. Passo Fundo: Embrapa Trigo, 2001. ISBN 1516-5582.
- [7] BARBEDO, J. Using digital image processing for counting whiteflies on soybean leaves. *Journal of Asia-Pacific Entomology*, v. 17, n. 4, p. 685-694, 2014. ISSN 12268615.
- [8] GONZALEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. 3º Edição. ed. São Paulo: Pearson Education, Inc., 2010.
- [9] MARQUES FILHO, O.; VIEIRA NETO, H. *Processamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. ISBN 8574520098.
- [10] BRADASKI, G.; A., K. *Learning OpenCV Computer Vision with the OpenCV Library*. 1º. ed. Sebastopol: O'Reilly, 2008. ISBN 978-0-596-51613-0.
- [11] SCURI, A. E. *Fundamentos da Imagem Digital*. Rio de Janeiro: [s.n.], 2002.
- [12] WANGENHEIN A. V., C. E. . R. R. Seminário Introdução à Visão Computacional. *Seminário Introdução à Visão Computacional*, 2001. Disponível em: <<http://www.inf.ufsc.br/~visao/>>. Acesso em: 06 jun. 2016.
- [13] SZELISKI, R. *Computer Vision: Algorithms and Applications*. 1º Edição. ed. New York: Springer, 2011. ISBN 978-1-84882-934-3 DOI 10.1007/978-1-84882-935-0.
- [14] PAULA, M. B. Visão Computacional Para Veículos Inteligentes Usando Câmeras Embarcadas, Porto Alegre, 2015. Disponível em: <http://hdl.handle.net/10183/122511>.
- [15] MATHWORKS. What Is Deep Learning? | How It Works, Techniques & Applications. *mathworks*. Disponível em: <<https://www.mathworks.com/discovery/deep-learning.html>>. Acesso em: 31 Dezembro 2017.
- [16] PONTI, M. A.; PARANHOS DA COSTA, G. B. Como funciona o Deep Learning. *SBC*, n. 1º, 2017. ISSN 978-85-7669-400-7.
- [17] LAGANIÈRE, R. *OpenCV 2 Computer Vision Application Programming Cookbook*. Birmingham: Published by Packt Publishing Ltd., 2011. ISBN 1180511.
- [18] COMMUNITY, I. ImageJ. *ImageJ*. Disponível em: <<http://imagej.net/ImageJ>>. Acesso em: 07 jun. 2016.
- [19] SCHINDELIN, J. . E. A. The ImageJ ecosystem: An open platform for biomedical image analysis. *Molecular Reproduction and Development*, v. 82, n. 7-8, p. 518-529, 2015. DOI: 10.1002/mrd.22489.
- [20] ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous

- Distributed Systems, 2016. Disponível em: <<http://arxiv.org/abs/1603.04467>>. DOI 10.1038/n.3331.
- [21] TENSORFLOW, 2017. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 12 Novembro 2017.
- [22] CAFFE | Deep Learning Framework. *Caffe*. Disponível em: <<http://caffe.berkeleyvision.org/>>. Acesso em: 31 dez. 2017.
- [23] LIU, T. et al. Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering*, v. 141, p. 82-93, 2016. ISSN 15375110.
- [24] XIA, C. E. A. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. *Ecological Informatics*, v. 29, p. 139-146, 2015.
- [25] WADHAI, M. Agriculture Pest Control Using Computer Vision Technique. *International Journal of Advanced Research*, v. 3, n. 8, p. 309-314, 2015. ISSN 2320-5407.
- [26] EMBRAPA. Embrapa Trigo: Apresentação. *Embrapa Trigo*. Disponível em: <<https://www.embrapa.br/trigo/apresentacao>>. Acesso em: 30 Julho 2016.
- [27] HOUGH, P.; POWELL, B. W. A method for faster analysis of bubble chamber photographs. *Il Nuovo Cimento*, v. 18, n. 6, p. 1184-1191, 1960.
- [28] FREEMAN, H. On the classification of line-drawing data. *Models for the Perception of Speech and Visual Form*, p. 408-412, 1967.
- [29] TENSORFLOW. Image Recognition. *TensorFlow*, 2018. Disponível em: <https://www.tensorflow.org/tutorials/image_recognition>. Acesso em: 10 Março 2018.
- [30] SCHILDT, H. *Java. A Referência Completa*. 1º. ed. Rio de Janeiro: Alta Books, 2014. ISBN 8576087553.
- [31] TEAM, G. GIMP - GNU Image Manipulation Program. *GIMP - GNU Image Manipulation Program*. Disponível em: <<https://www.gimp.org/>>. Acesso em: 20 fev. 2018.
- [32] COMBATALADE, D. C. *Basics of Heart Rate Variability Applied to Psychophysiology*. 2010. 31p.
- [33] BOWMAN, D. A. et al. *3D User Interfaces: theory and practice*. Boston: Addison-Wesley, 2004. 478 p. ISBN 0-201-75867-9.
- [34] BOWMAN, D. A.; GABBARD, J. L.; HIX, D. A Survey of Usability Evaluation in Virtual Environments: classification and comparison of methods. *Presence: Teleoperators and Virtual Environments*, v. 11, n. 4, p. 404-424, 2002.
- [35] BORTH, M. R. E. A. A Visão Computacional no Agronegócio: Aplicações e Direcionamentos. *2º Seminário Internacional de Integração e Desenvolvimento Regional*, Ponta Porã, 07, 08 e 09 out. 2014. Disponível em: http://www.gpec.ucdb.br/pistori/publicacoes/borth_eaeco2014.pdf.
- [36] GASQUES, J. G. Desempenho e crescimento do agronegócio no Brasil. *Instituto de Pesquisa Econômica Aplicada*, Brasília, fev. 2004..
- [37] DA SILVA, D. S. Atributos de Ponto de Interesse e Casamento de Modelos para Contagem de Insetos-Praga em Cultura de Soja, Campo Grande, Março 2016.
- [38] MACHADO, C. C.; SILVA DOS SANTOS, R. S. Pulgões do trigo e ação de parasitoides em Augusto Pestana, noroeste do estado do Rio Grande do Sul, Brasil. *Revista Brasileira de Agroecologia*, Pelotas, p. 179-186, out. 2012. ISSN 1980-9735.
- [39] LIU, T. E. A. Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering*, v. 141, p. 82-93, 2016. ISSN 15375110.

[40] OPENCV. DOCUMENTATION OpenCV. *OpenCV*. Disponível em: <<http://opencv.org/documentation.html>>. Acesso em: 10 Agosto 2016.