

UNIVERSIDADE DE PASSO FUNDO
Programa de Pós-Graduação em
Computação Aplicada

Dissertação de Mestrado

**INSECTCV: UM SISTEMA PARA
DETECÇÃO DE INSETOS EM
IMAGENS DIGITAIS**

TELMO DE CESARO JÚNIOR



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

**INSECTCV: UM SISTEMA PARA DETECÇÃO
DE INSETOS EM IMAGENS DIGITAIS**

Telmo De Cesaro Júnior

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
2020

CIP – Catalogação na Publicação

D292i De Cesaro Júnior, Telmo
InsectCV [recurso eletrônico] : um sistema para detecção de insetos em imagens digitais / Telmo De Cesaro Júnior. – 2020.
6.2 MB ; PDF.

Orientador: Prof. Dr. Rafael Rieder.
Dissertação (Mestrado em Computação Aplicada) –
Universidade de Passo Fundo, 2020.

1. Informática na agricultura. 2. Pragas agrícolas – Controle integrado. 3. Redes neurais (Computação). I. Rieder, Rafael, orientador. II. Título.

CDU: 004.8

Catálogo: Bibliotecária Juliana Langaro Silveira – CRB 10/2427


ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO

TELMO DE CESARO JÚNIOR


Aos vinte e um dias do mês de dezembro do ano de dois mil e vinte, às quatorze horas, realizou-se na Divisão UPF On-Line, prédio D1 sala 01, da Universidade de Passo Fundo (UPF), a sessão pública de defesa do Trabalho de Conclusão de Curso “InsectCV: um sistema para detecção de insetos em imagens digitais”, de autoria de Telmo de Cesaro Júnior, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA. 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, Guilherme Afonso Madalozzo, Douglas Lau e João Paulo Silva do Monte Lima. 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)

p.p. 

Prof. Dr. Guilherme Afonso Madalozzo – UPF
(Avaliador Interno)

p.p. 

Prof. Dr. Douglas Lau – EMBRAPA
(Avaliador Externo)

p.p. 

Prof. Dr. João Paulo Silva do Monte Lima – UFRPE
(Avaliador Externo)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA

AGRADECIMENTOS

Agradeço primeiramente a Deus e a minha família. Em segundo, ao orientador, Dr. Rafael Rieder, pelo apoio em todas as atividades desse trabalho, bem como, ao incentivo para as publicações. Agradeço a Universidade de Passo Fundo, pela disponibilização dos equipamentos eletrônicos do Laboratório de Realidade Virtual. A Embrapa Trigo, em especial aos colaboradores Dr. Douglas Lau e Dra. Vânia Bianchin. Ao Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense e aos colegas do Grupo de Pesquisa em Computação Aplicada, que viabilizaram as parcerias entre essas Instituições.

INSECTCV: UM SISTEMA PARA DETECÇÃO DE INSETOS EM IMAGENS DIGITAIS

RESUMO

A tarefa manual de contagem e identificação de pequenos insetos, como afídeos e parasitoides, capturados em armadilhas de campo do tipo cor é exaustiva, demorada e não escalável. Essa atividade envolve a separação dos elementos de interesse e requer a utilização de lupas e/ou microscópios. Recentes avanços em inteligência artificial, processamento de imagens e computação de alto desempenho têm viabilizado o desenvolvimento de soluções de visão computacional eficientes para monitorar pragas e identificar doenças em plantas. Nesse sentido, este trabalho apresenta o InsectCV, um sistema para a contagem e identificação automática de insetos em imagens geradas pela digitalização de amostras capturadas em armadilhas. Para o desenvolvimento dessa solução, utilizou-se um conjunto de 209 imagens em escala de cinza contendo 17.908 insetos rotulados, a rede neural convolucional profunda Mask R-CNN para a geração do modelo e a implementação de três serviços web. No treinamento do modelo foi aplicada a técnica de transferência de aprendizado e aumento de dados. Foram definidos dois novos parâmetros para ajustar a relação de falsos positivos por classe. Para a validação do modelo foram utilizadas imagens de insetos obtidas a partir de armadilhas expostas em Coxilha e Passo Fundo, RS, Brasil nas safras de trigo de 2019 e 2020. Em comparação com a contagem manual, os coeficientes de determinação foram próximos a 1 ($R^2 = 0,87$ para afídeos e $R^2 = 0,92$ para parasitoides), indicando a capacidade do modelo em identificar a flutuação dos níveis populacionais para esses insetos. Sendo assim, o InsectCV pode ser utilizado para detectar limiares de ação em sistemas de alerta para manejo integrado de pragas.

Palavras-Chave: armadilha, afídeos, convolucional, parasitoides, Mask R-CNN.

INSECTCV: A SYSTEM FOR INSECT DETECTION IN DIGITAL IMAGES

ABSTRACT

The manual task of counting and identifying small insects, such as aphids and parasitoids, captured in color-type field traps is exhausting, time-consuming, and non-scalable. This activity involves the separation of the elements of interest and requires the use of magnifiers or microscopes. Recent advances in artificial intelligence, image processing, and high-performance computing have enabled the development of efficient computer vision solutions to monitor pests and identify diseases in plants. With this in mind, this work presents InsectCV, a system for the automatic counting and identification of insects in images generated by the scanning of samples captured in traps. For the development of this solution, we used a 209 grayscale images dataset containing 17,908 labeled insects, a convolutional neural network Mask R-CNN to generate the model, and the development of three web services. During the training of the model, we applied the transfer learning technique and the data augmentation. We defined two new parameters to adjust the false-positive ratio by class. We used images of insects obtained from traps exposed in Coxilha and Passo Fundo, RS, Brazil in 2019 and 2020 wheat crops. In comparison to the manual counting, we verified coefficients determination close to 1 ($R^2 = 0.87$ for aphids and $R^2 = 0.92$ for parasitoids), proving the ability of the model to identify the fluctuation of population levels for these insects. Therefore, InsectCV can be used to detect action thresholds in alert systems for integrated pest management.

Keywords: aphids, convolutional, parasitoids, Mask R-CNN, trap.

LISTA DE FIGURAS

Figura 1.	Estrutura da Mask R-CNN reproduzida a partir de [78].	36
Figura 2.	Arquiteturas Head para Mask R-CNN [78].	37
Figura 3.	Hiper-parâmetros e a arquitetura Mask R-CNN [83].	40
Figura 4.	Estrutura Analítica do Trabalho.	43
Figura 5.	Imagem com afídeos, parasitoides, detritos e poucas conexões. Caixas verdes indicam parasitoides e amarelas representam afídeos.	45
Figura 6.	Imagem com afídeos e detritos.	45
Figura 7.	Serviço de Cadastro.	48
Figura 8.	Serviço de Detecção.	49
Figura 9.	Tabelas adicionadas no AgroDB.	50
Figura 10.	Valores atribuídos para o peso de perda <code>rpn_bbox_loss</code> e <code>rpn_class_loss</code> . Nestes gráficos, o eixo X representa a quantidade de épocas de treinamento e o Y representa o valor da perda.	53
Figura 11.	Valores atribuídos para os pesos de perda: <code>mrcnn_bbox_loss</code> , <code>mrcnn_class_loss</code> e <code>mrcnn_mask_loss</code>	54
Figura 12.	Valores de perda geral obtidos ao longo das cem épocas de treinamento.	54
Figura 13.	Exemplo com distribuição esparsa de objetos de interesse, rotulada como "Imagem Fácil".	55
Figura 14.	Resultado da detecção para a imagem fácil.	56
Figura 15.	Exemplo de imagem com distribuição intermediária de objetos de interesse, rotulada como "imagem média".	56
Figura 16.	Resultado da detecção para a imagem média.	57
Figura 17.	Exemplo com distribuição densa de objetos de interesse, rotulada como "imagem difícil".	57
Figura 18.	Resultado da detecção para a imagem difícil.	58
Figura 19.	Detecção de parasitoides: rótulos em verdes indicam a localização e classificação de positivos. Elipses em laranja indicam falsos positivos.	58
Figura 20.	Ajuste para afídeos, safra 2019 e 2020. N=200	61
Figura 21.	Relação entre a contagem manual de afídeos e o detectado pelo InsectCV.	61
Figura 22.	Ajuste para parasitoides, safra 2020. N=90	62
Figura 23.	Relação entre a contagem manual de parasitoides e o detectado pelo InsectCV.	62

Figura 24. Curva de progresso de populações de afídeos. Dados de duas safras de trigo (2019 e 2020). Dados médios de 5 armadilhas. As linhas pontilhadas representam os limiares de 20 e 50. 64

LISTA DE TABELAS

Tabela 1.	Relação de trabalhos científicos selecionados pela pesquisa	29
Tabela 2.	Trabalhos selecionados pela revisão que aplicaram Redes Neurais Con- volucionais Profundas	33
Tabela 3.	Primeiro conjunto de imagens.	44
Tabela 4.	Segundo conjunto de imagens.	44
Tabela 5.	Resultados gerados pelos experimentos iniciais.	47
Tabela 6.	Detalhes dos experimentos adicionais 6 e 7.	47
Tabela 7.	Coeficientes para afídeos em 2019.	59
Tabela 8.	Coeficientes para afídeos em 2020.	60
Tabela 9.	Coeficientes para parasitoides em 2020.	60
Tabela 10.	Métricas para parasitoides.	63

LISTA DE ABREVIATURAS

- fr. extr.. – extração de característica
- img. classif.. – classificação de imagem
- n/a. – não aplicado
- n/u. – não usado
- obj. detect.. – detecção de objeto
- preproc.. – pré-processamento
- . – não informado

LISTA DE SIGLAS

ANN – Rede Neural Artificial

DCNN – Rede Neural Convolutacional Profunda

DL – Aprendizagem Profunda

FASTER R-CNN – Faster Region-Based Convolutional Neural Network

IOU – Intersecção sobre União

IA – Inteligência Artificial

YOLO – You Only Look Once

PI – Processamento de Imagem

ML – Aprendizado de Máquina

MAP – Mean Average Precision

ROI – Região de Interesse

RELU – Funções de ativação não lineares

REST – Transferência Representacional de Estado

RPN – Region Proposal Network

R-FCN – Region-based Fully Convolutional Networks

RGB – Padrão baseado nas cores Vermelho, Verde e azul

ROC – Receiver Operator Characteristic

SVM – Máquina de Vetores de Suporte

SWCNN – Sliding-Window-based Convolutional Neural Networks

SSD – Single Shot Detector

VC – Visão Computacional

SUMÁRIO

1	INTRODUÇÃO	21
2	FUNDAMENTAÇÃO TEÓRICA	25
2.1	REDES NEURAIS CONVOLUCIONAIS PROFUNDAS	25
2.2	REVISÃO SISTEMÁTICA	28
2.2.1	Considerações sobre as implementações com DCNN	32
2.2.2	Mask R-CNN	35
2.2.2.1	Hiper-parâmetros	38
2.2.2.2	Otimizações na utilização do Hardware	41
3	MATERIAIS E MÉTODOS	43
4	RESULTADOS E DISCUSSÕES	53
4.1	RESULTADOS	53
4.1.1	Customizações	55
4.1.2	Coeficientes	59
4.1.3	Métricas	60
4.1.4	Pontos de Corte	63
4.2	DISCUSSÕES	65
5	CONCLUSÃO	67
	REFERÊNCIAS	71

1. INTRODUÇÃO

O monitoramento da flutuação populacional de pragas no campo, ou em experimentos de laboratório, permite acompanhar a variação dos níveis de infestação e embasar programas de manejo integrados de pragas [1]. Segundo Wellings e Dixon [2], a flutuação populacional de afídeos, determinada por amostragens periódicas, mostra-se sazonal e pode variar de forma significativa, mensalmente ou anualmente. Nas culturas de trigo, cevada, triticale e aveia, o aumento da população de afídeos pode causar significativa redução na produtividade, em decorrência dos seus hábitos alimentares e por serem vetores de transmissão de doenças [3]. Tendo em vista que nas regiões tritícolas do Brasil observa-se a ocorrência desse inseto-praga, o monitoramento da flutuação populacional permite acompanhar a variação dos níveis de infestação e embasar programas de manejo.

Afídeos, também conhecidos como pulgões, são insetos pertencentes a família *Aphididae*, ordem *Hemiptera* e medem aproximadamente 1,5 a 3,5 mm. Possuem corpo mole, piriforme e com antenas longas. O aparelho bucal é do tipo picador-sugador e desenvolvimento paurometabólico. São altamente prolíficos e reproduzem-se por viviparidade e partenogênese telítica (origina apenas fêmeas). Vivem sobre a planta em colônias, sendo compostas por fêmeas adultas, podendo ser alados (com asas) ou ápteros (sem asas), e por ninfas (inseto jovem) de diferentes tamanhos. Apresentam ciclo de vida curto, podendo completar uma geração em uma semana e originar até dez ninfas/fêmeas por dia. A temperatura ideal para a reprodução é de 18°C a 25°C e períodos de baixa precipitação pluvial são favoráveis a esses insetos. Podem voar por centenas de quilômetros com o auxílio do vento [4].

Como referido, afídeos ninfas ou adultos podem causar danos diretos ou indiretos às plantas. O dano direto está relacionado a sucção contínua da seiva, resultando na deformação de frutos, folhas ou flores, debilitando a planta e podendo levá-la a morte. O excremento gerado por esses insetos atrai formigas doceiras, que por sua vez podem proteger as pragas da ação de inimigos naturais. O dano indireto é devido a transmissão de viroses, visto que um único inseto é capaz de contaminar várias plantas através da sua saliva [5] [6].

As principais espécies de afídeos encontradas em trigo são: *Metopolophium dirhodum* o pulgão-da-folha (Walker, 1849), *Schizaphis graminum* o pulgão-verde-dos-cereais (Rondani, 1852), *Sitobion avenae* o pulgão-da-espiga (Fabricius, 1775), *Rhopalosiphum padi* o pulgão-da-aveia (Linnaeus, 1758), *Rhopalosiphum maidis* o pulgão-do-milho (Fitch, 1856) e o *Rhopalosiphum rufiabdominale* o pulgão-da-raiz (Sasaki, 1899) [3].

Os parasitoides são insetos que agem como inimigos naturais dos pulgões do trigo. Segundo Salvadori & Salles e Salvadori *et al.* [7, 8] de julho de 1978 até meados

de 1982, com o apoio da Organização das Nações Unidas para Alimentação e Agricultura (FAO) e da Universidade da Califórnia, a Empresa Brasileira de Pesquisa Agropecuária (Embrapa) executou um projeto para controlar biologicamente os pulgões do trigo, por meio da importação, multiplicação e liberação de parasitoides. Para tanto, foram estudadas e multiplicadas, nos laboratórios da unidade Embrapa Trigo, em Passo Fundo, 14 espécies de microimenópteros, sendo produzidos e liberados até 1992 cerca de 20 milhões de microimenópteros. Segundo Zúñiga e Salvadori & Salles [9, 7] as espécies de parasitoides introduzidas foram: *Aphelinus abdominalis*, *A. asychis*, *A. flavipes*, *A. varipes*, *Ephedrus plagiator*, *Lysiphlebus testaceipes*, *Praon gallicum*, *P. volucre*, *Aphidius colemani*, *A. ervi*, *A. pascuorum*, *A. picipes*, *A. rhopalosiphi* e *A. uzbekistanicus*.

Em relação a programas de monitoramento de pragas, a Embrapa Trigo possui uma Rede de Monitoramento de Pragas em Cereais de Inverno. Criada em 2015, com parceiros nos Estados do Rio Grande do Sul, Paraná, Minas Gerais e Mato Grosso, essa rede utiliza armadilhas e experimentos para observar migrações de populações de afídeos e parasitoides, relacionando os níveis populacionais ao dano econômico causado e gerando informações para sistemas de alerta para a tomada de decisão do manejo da lavoura [10].

Nos experimentos de campo são utilizadas armadilhas do tipo Moericke (bandejas amarelas). A cor amarela está relacionada com a teoria da radiação amarela refletiva [11]. Essas armadilhas possuem 45 cm x 30 cm x 4,5 cm de comprimento, largura e altura, respectivamente. Existem cinco orifícios laterais, de cinco milímetros de diâmetro, próximas a borda da bandeja, protegidos por uma tela fina, para evitar o extravasamento e a perda do conteúdo sólido, nos períodos de chuva. Em cada bandeja são colocados aproximadamente 2,0 litros de solução preparada com água, formol 40% e detergente¹. Este método é amplamente utilizado e de fácil implementação, permitindo capturar pulgões de uma determinada área para monitoramento [12] [13].

Nesse contexto, o processo de triagem dos elementos capturados nas bandejas requer tarefas manuais, como a coleta, a eliminação da água, a remoção dos detritos e a realocação de afídeos e de parasitoides em uma lâmina para análise visual de um especialista em laboratório. Cada armadilha pode conter centenas de insetos de diferentes espécies. Na identificação das espécies, são utilizadas chaves dicotômicas que especificam a morfologia de cada espécie [3, 14]. Em função do tamanho reduzido desses insetos, utiliza-se geralmente um microscópio estereoscópio (lupa) para localizar as características morfológicas no espécime, tais como o número de segmentos da antena, coloração do sifúnculo, ramificações nas asas e formato do corpo.

Ao avaliar atividades manuais realizadas pelos técnicos da Embrapa Trigo na identificação visual das espécies de pulgões e parasitoides, observa-se que é uma tarefa demorada, gera fadiga e a sua precisão pode ser reduzida em razão da quantidade de insetos de interesse (afídeos e parasitoides), detritos (terra, folhas, grãos, etc.) e outros insetos (mos-

¹Para cinco armadilhas é preparada uma solução com 10 litros de água, 30 ml de formol 40% e 20 ml de detergente

cas, abelhas, etc.). Amostras com muitos elementos ocasionam oclusões, sobreposições parciais, mudanças de pose e aglutinações, que também dificultam o processo de análise. processadas por Especialistas qualificados.

Os efeitos sazonais também atingem outros insetos, assim o desafio da análise de imagem em reconhecer e contar corretamente os pulgões é variável ao longo do ano. Na medida em que a temperatura do ambiente aumenta, além dos afídeos outros insetos também começam a cair na armadilha dificultando a detecção dos afídeos.

Recentes trabalhos abordam a utilização de recursos computacionais para automatizar o processo de identificação e contagem de objetos em imagens digitais, como insetos e doenças de plantas. Esses estudos também destacam que o processo manual de identificação é considerado uma atividade intensiva, lenta, tediosa, suscetível a erros e com precisão variável, o que inviabiliza a sua execução em larga escala [15], [16], [17], [18] e [19]. Em casos de grandes concentrações de objetos nas lâminas de observação, a contagem manual dos objetos de interesse é apenas estimada, com base em proporções espaciais e na experiência do especialista [17]. Além disso, com a automação é possível padronizar o processo de contagem de insetos e reduzir a quantidade de Especialistas qualificados na tarefa de análise das características morfológicas.

A classificação de pulgões da espécie *R. padi* nos experimentos de laboratório, também conhecidos como bioensaios, foi parcialmente automatizada na Embrapa Trigo a partir do trabalho de Lins *et al.* [20] [21]. Neste estudo, definiu-se um conjunto de procedimentos para a coleta de possíveis espécimes retidos nas bandejas, e criou-se uma ferramenta computacional denominada *AphidCV* [22], capaz de contar, classificar e mensurar afídeos a partir de imagens digitais. Com base nesse protocolo, o técnico filtra o conteúdo da armadilha eliminando água, detritos e outros insetos (mariposas, moscas, abelhas, etc.) de maior tamanho. Após, ele separa preliminarmente os afídeos, e os deposita em uma placa de Petri. Por fim, gera-se uma imagem da placa por meio da digitalização desta em scanner, servindo de entrada para análise do software *AphidCV*. Com a utilização do software, o tempo médio para a realização das análises foi reduzido em quatro vezes em relação a contagem manual com alto grau de confiabilidade [21].

Com isto em mente, a automatização do processo de identificação e de contagem de insetos para as amostras de campo, por intermédio da utilização de recursos computacionais, pode ser vista como alternativa para reduzir atividades manuais do Especialista e padronizar o processo, uma vez que a Embrapa Trigo digitaliza as amostras semanalmente e possui um banco de dados com aproximadamente três mil imagens. Destaca-se que as amostras de campo são mais densas em objetos, se comparadas às geradas nos bioensaios, em virtude da exposição das armadilhas ao meio externo.

Para tanto, este trabalho tem como objetivo propor um sistema computacional (InsectCV) para reconhecer afídeos e parasitoides em imagens digitais, independentemente da espécie, utilizando técnicas de visão computacional (VC) e inteligência artificial (IA).

Nesta perspectiva, a proposta considera a definição de uma abordagem convolucional, a utilização de um conjunto alvo de imagens, a definição de técnicas para pré-processamento de imagens para segmentação dos objetos de interesse, o treinamento e a validação de um modelo inteligente, bem como o desenvolvimento de serviços web para integração com o Trap System, uma plataforma de monitoramento, simulação e tomada de decisão no manejo de epidemias causadas por vírus transmitidos por insetos [23].

Por intermédio do *InsectCV*, espera-se responder a seguinte pergunta de pesquisa: um modelo inteligente, treinado com um pequeno conjunto de imagens em escalas de cinza, quantidade significativa de objetos rotulados, a aplicação das técnicas de aumento de dados, transferência de aprendizado e customizações nos hiper-parâmetros, pode ser capaz de reconhecer insetos, como afídeos e parasitoides, com precisão suficiente para embasar sistemas de alerta?

Sendo assim, este documento está estruturado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, destacando uma revisão sistemática de literatura para embasar o desenvolvimento do *InsectCV*, e uma apresentação sobre a abordagem convolucional de aprendizado profundo *Mask R-CNN*; o Capítulo 3 descreve os materiais e métodos empregados para criar a solução proposta e definir a geração do modelo inteligente; já o Capítulo 4 mostra e discute os resultados dos experimentos de treinamento e validação, para avaliar o desempenho do modelo; por fim, o Capítulo 5 apresenta as considerações finais e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Soluções baseadas em Visão Computacional (VC) e Inteligência Artificial (IA) são bem difundidas na Agricultura, contribuindo na redução de custos com a automação de tarefas repetitivas [24]. Através de IA, automatizações têm alcançado níveis satisfatórios no processo de classificação de imagens e detecção de objetos. Identificação e contagem de insetos em armadilhas, insetos em plantas e doenças em folhas de plantas [25, 26, 27], por exemplo, tem apresentado uma precisão média (mAP) ² de 90,2%, 88,5% e 83,1%, respectivamente. Estes resultados demonstram boa eficácia em comparação às técnicas manuais.

Entre as técnicas de IA aplicadas a imagens digitais, a Rede Neural Convolutiva Profunda (DCNN) é considerada uma das técnicas mais eficazes para o aprendizado de características. Nesse contexto, este Capítulo faz uma breve apresentação sobre DCNN, seguida de uma revisão sistemática de literatura, publicada recentemente no periódico *Computers and Electronics in Agriculture* [28]. Este embasamento permitiu levantar estudos sobre detecção de pequenos objetos, como insetos, e casos de sobreposição em imagens. Nessa análise, também destacam-se considerações sobre as abordagens utilizadas pelos trabalhos selecionados, e sobre o framework Mask R-CNN.

2.1 REDES NEURAS CONVOLUCIONAIS PROFUNDAS

Uma DCNN é uma abordagem computacional que emprega o uso de múltiplas camadas de redes neurais, geralmente alternadas em camadas convolucionais e de agrupamento, para processamento e classificação de imagens digitais em diferentes áreas do conhecimento [29]. Com a aplicação desse recurso na Agricultura, por exemplo, a atividade de localização de características morfológicas pertencentes a uma determinada espécie, com o auxílio de um microscópio, pode ser substituída por um processo que extrai automaticamente informações relevantes das imagens digitais e compara com um modelo de conhecimento. O uso de DCNNs permite incorporar no modelo o conhecimento do especialista com a marcação (rotulação) dos objetos de interesse nas imagens.

Kamilaris e Prenafeta-Boldú [24] apresentam uma seleção de 40 trabalhos que aplicam diferentes técnicas de aprendizado profundo na Agricultura. Destes trabalhos, 42% utilizam DCNNs. De acordo com os autores, esse recurso é superior as técnicas de processamento de imagens. No entanto, essa pesquisa apontou algumas peculiaridades, como a necessidade de um grande volume de dados (imagens) e unidade(s) de processamento

²É uma métrica utilizada para mensurar a precisão de detectores de objetos, que é obtida através da média de outras AP's, como accuracy e precision.

gráfico (GPU) para acelerar o treinamento do modelo. Considerando métodos baseados em aprendizado de máquina (ML), na maioria dos casos, quanto maior o conjunto de dados, melhor o desempenho de generalização [16]. Os autores de estudos relevantes em aprendizagem profunda (DL) recomendam conjuntos de dados com 50.000 exemplos para treinamento [30, 31].

Segundo Chen *et al.* [32], os níveis de precisão das DCNNs indicam que essa técnica representa atualmente o estado-da-arte para as tarefas de classificação de imagens [33], detecção de objetos [34] e segmentação [35], superando os métodos tradicionais, onde as características dos objetos de interesse são extraídos manualmente [36].

Na tarefa de classificação de imagens deve existir apenas um objeto no centro da imagem, portanto, não se aplica para casos de objetos conectados ou parcialmente sobrepostos. Na detecção de objetos, é possível reconhecer vários elementos em uma determinada imagem. Porém, é necessário um amplo conjunto de imagens para treinamento e teste do modelo, onde cada objeto de interesse deve ser identificado e rotulado [19]. Em termos de complexidade, essa tarefa demanda maior tempo de processamento.

A tarefa de detecção de objetos por DCNNs teve significativo aumento de desempenho com as contribuições de LeCun, Bengio e Hinton [30]. Sun *et al.* [19] sugerem a organização de DCNNs por classificadores de dois estágios baseados em *Region Proposal Network* (RPN):

- Faster Region-Based Convolutional Neural Network (Faster R-CNN) [34];
- Region-based Fully Convolutional Networks (R-FCN) [37];
- Mask R-CNN [38];

e classificadores de um estágio sem RPN:

- Single Shot Multibox Detector (SSD) [39];
- You Only Look Once (YOLO) [40];
- RetinaNet [41];

Fuentes *et al.* [27] relatam uma organização semelhante, com a definição das meta-arquiteturas Faster R-CNN, R-FCN e SSD para a detecção de objetos por DCNNs. Esses autores destacam a necessidade de combiná-las com diferentes extratores de características, como a rede neural profunda para imagens em larga escala (VGG) e a *Residual Network* (ResNet). O extrator é a principal parte em uma arquitetura profunda e a sua escolha depende de vários fatores, como o tipo e a quantidade de camadas e parâmetros. Quanto mais profunda a rede em camadas e parâmetros, maior é a complexidade, o

que demanda mais recursos computacionais. Por outro lado, redes não profundas (*Shallow networks*) podem ser ineficientes para extrair características relevantes de objetos pequenos em imagens com alta resolução.

No trabalho de Saeidi e Ahmadi [42], encontra-se uma classificação diferente à citada por Sun *et al.* [19], que define três principais abordagens para detecção de objetos usando DCNN:

- Sliding-Window-based Convolutional Neural Networks (SWCNN) [43, 44];
- Region-based Convolutional Neural Networks (R-CNN) [45];
- Single Shot Detector (SSD) [39].

Segundo Saeidi & Ahmadi [42] e Lampert *et al.* [46], a abordagem SWCNN utiliza escalas múltiplas e janelas deslizantes *Sliding window* para extrair características das imagens. O custo computacional, complexidade e quantidade de imagens para treinamento são significativamente maiores em comparação às demais abordagens.

A família R-CNN é composta por: R-CNN, Fast R-CNN e Faster R-CNN. O método R-CNN [45] utiliza algoritmos de visão computacional para localizar regiões candidatas (*selective search region proposal*) em combinação com DCNN e SVM. Embora esse método seja adequado para encontrar as posições dos objetos, ele não é capaz de realizar uma localização precisa de todo o objeto por uma caixa delimitadora (*bounding box*) [42]. Na Fast R-CNN [47] a imagem inteira e os objetos candidatos extraídos são considerados como entradas para o modelo DCNN.

A abordagem de alto desempenho Faster R-CNN [34], realizada no primeiro estágio, chamado de *Region Proposal Network* (RPN), a geração de objetos candidatos via *bounding box*. O RPN é uma rede totalmente convolucional que prediz as coordenadas do objeto na imagem, bem como, o nível de detecção [42]. No segundo estágio, que de fato é a sua essência, extrai as características destes, usando *RoIPool* e executa a classificação e regressão das *bounding box* [38]. Essa abordagem tem gerado altos níveis de precisão na tarefa de detecção de objetos.

Mask R-CNN é um framework recente, baseado no Faster R-CNN e que possui uma ramificação adicional (*branch*) para segmentar objetos através de uma máscara em paralelo com a *branch* para reconhecer as *bounding box* [38]. Esse método será detalhado na seção 2.2.2.

A partir desses fundamentos iniciais, a próxima seção descreve uma revisão sistemática que selecionou 33 estudos científicos relacionados ao objetivo deste trabalho.

2.2 REVISÃO SISTEMÁTICA

Com o objetivo de avaliar técnicas recentes de IA e VC para o reconhecimento de objetos e customizações para elevar a capacidade de detecção de pequenos objetos, bem como, possíveis tratamentos para casos de sobreposição, foi realizada uma pesquisa nas bases digitais: ACM, IEEE, IET, DBLP, ScienceDirect, Scopus, SpringerLink e Web of Science. Para isso, foram formuladas duas *string de busca*, devido a limitação da quantidade de operadores lógicos em algumas bases. Desse modo, também foi possível incluir mais resultados para aprofundar a investigação. Os termos usados na string estão vinculados ao processamento de imagens e aprendizado profundo. Também foram inclusos termos relacionados ao reconhecimento de objetos sobrepostos em imagens digitais, de acordo com os objetivos da pesquisa. As duas *string de busca* são:

(insect) AND ("image processing" OR "vision-based" OR detection OR recognition OR counting OR segmentation) AND (touching OR overlapped OR overlapping OR occlusion)

(insect) AND ("image processing" OR "vision-based" OR detection OR recognition OR counting OR segmentation) AND ("convolutional neural network" OR "deep learning" OR "artificial neural network")

Com essa pesquisa foram localizados aproximadamente trezentos trabalhos, publicados em periódicos entre 2015 e 2019 na língua inglesa. Para selecionar os trabalhos localizados, foram aplicados os seguintes critérios de elegibilidade:

- O estudo científico deve implementar técnicas para contagem, identificação ou classificação de objetos, de tamanho equivalente a afídeos, em imagens digitais (RGB ou GrayScale);
- Descartar estudos que utilizaram recursos tecnológicos para medidas espectrais, multibanda polarimétrica, multiespectral e imagens geradas por câmeras ou sensores infravermelhos;
- O estudo deve utilizar IA, VC ou processamento de imagens (PI);
- O estudo deve utilizar apenas ferramentas gratuitas (*open source*);

A Tabela 1 lista os 33 estudos que foram selecionados com base nos critérios de elegibilidade descritos acima. Foram admitidos artigos sobre a identificação de insetos e outros relacionados ao tema, como o reconhecimento de doenças em plantas, grãos, frutas, células e nematoides, para ampliar a elucidação de diferentes técnicas computacionais aplicadas.

Tabela 1. Relação de trabalhos científicos selecionados pela pesquisa

Artigo	Aprendizado de Máquina	Proc. de Imagens	Sobreposição	Pequenos objetos	Classes	Precisão média	Conj. de Imagens	Fonte
Akintayo <i>et al.</i> [15]	CNN	preproc.	-	-	2	96%	644	ovos de nematoides ³
Alemayehu <i>et al.</i> [48]	SVM	segm. ⁴ + ftr. extr.	-	-	4	93.3%	900	inseto na planta
Avila-George <i>et al.</i> [49]	ANN	preproc., ftr. extr.	-	-	3	97.2%	40	doença foliar
Bakkay <i>et al.</i> [50]	Only IP	preproc.	segm. ⁵	CLS	3	77%	100	inseto na armadilha de campo
Basati <i>et al.</i> [51]	Decision Tree	preproc., segm.	-	-	2	90.2%	16 ⁶	grãos de trigo
Bigin <i>et al.</i> [52]	SVM	ftr. extr.	-	-	15	85%	6900	inseto
Chen <i>et al.</i> [32]	CNN	preproc.	-	-	2	95.6%	68 ⁷	inseto na folha
Cheng <i>et al.</i> [53]	CNN	-	-	-	10	98.6%	550	inseto na folha
Cruz <i>et al.</i> [54]	CNN	preproc.	-	-	3	98.6%	224 ⁸	doença foliar
Ding and Taylor [16]	CNN	preproc.	-	-	2	93.1%	177 ⁹	inseto na armadilha de campo
Espinoza <i>et al.</i> [55]	ANN	IP ¹⁰	-	-	2	96%	3185 ¹¹	inseto em armadilha adesiva
Fuentes <i>et al.</i> [27]	CNN	n/u	-	-	9	83%	5000 ¹²	doença foliar
Kalamitanos <i>et al.</i> [56]	CNN	n/u	-	-	2	91.5%	542	inseto na armadilha de campo
Li <i>et al.</i> [26]	CNN	n/u	-	-	2	88.5%	1228 ¹³	inseto na planta
Li <i>et al.</i> [57]	MLE	segm.	LPSA ¹⁴	n/a	6	74.7%	700 ¹⁵	corn grains
Liu <i>et al.</i> [58]	CNN	preproc.	-	-	12	95.1%	5000	inseto na planta
Liu <i>et al.</i> [59]	CNN	n/u	-	-	16	75.4%	88670 ¹⁶	inseto na armadilha de campo
Mousavi <i>et al.</i> [60]	SVM	ftr. extr.	-	-	5	90%	100	doença foliar
Nazri <i>et al.</i> [17]	CNN	binary filtering	-	-	2	95%	687 ¹⁷	inseto em armadilha adesiva
Partel <i>et al.</i> [18]	CNN	preproc.	-	-	2	80%	800 + 8000 ¹⁸	inseto na armadilha de campo
Picon <i>et al.</i> [61]	CNN	preproc. + segm. ¹⁹	n/a	Superpixel	3	87%	8178 ²⁰	doença foliar
Priyadarshini <i>et al.</i> [62]	CNN	preproc.	-	-	2	-	-	doença de planta
Ram and Rodrigues [63]	Only IP	preproc.	FRST	-	2	88.1%	260	núcleos celulares ²¹
Ramcharan <i>et al.</i> [64]	CNN	n/u	-	-	6	93%	2756	doença de planta
Roldán-Serrato <i>et al.</i> [65]	ANN ²²	ftr. extr.	-	-	2	88%	200	inseto na folha
Shen <i>et al.</i> [66]	CNN	preproc.	-	Inception	6	88%	739 ²³	inseto na armadilha de laboratório
SivaSangari <i>et al.</i> [67]	Only IP	preproc.	ACO	-	2	n/a	-	inseto na folha
Sun <i>et al.</i> [19]	CNN	preproc.	-	optimization ²⁴	6	74.6%	2183 ²⁵	inseto na armadilha de campo
Tan <i>et al.</i> [68]	CNN	preproc.	-	-	4	97.5%	250 ²⁶	doenças na fruta
Wen <i>et al.</i> [69]	SDAE	ftr. extr.	pose estimate ²⁷	-	9	96.9%	728	inseto em armadilha adesiva
Xia <i>et al.</i> [70]	CNN	preproc.	-	-	24	89.2%	660	inseto na folha ²⁸
Yue <i>et al.</i> [71]	CNN	preproc.	-	super resolution	3	n/a	249	inseto
Zhong <i>et al.</i> [25]	CNN + SVM	ftr. extr.	-	-	6	90.1%	3000/700 ²⁹	inseto na armadilha de campo

-
- ³imagens geradas por microscópio
- ⁴utilizando k-means algorithm
- ⁵Contour estimation e region merging
- ⁶100 grãos por imagem
- ⁷a partir das imagens originais (1600x1200) outras foram geradas via recorte aleatório (256x256) e por cinco transformações de imagem
- ⁸pré-treinamento da rede no conjunto de dados PlantVillage
- ⁹72 novas imagens foram geradas por transformações geométricas simples a partir de uma amostra original
- ¹⁰obj. detect., segm. por Otsu method e ftr. extr.
- ¹¹32772 imagens foram geradas a partir de uma região recortada medindo 41x41
- ¹²mais imagens foram criadas com aumento de dados por transformações geométricas e de intensidade
- ¹³3778 imagens geradas por flipping across the horizontal-axes e adicionando salt-and-pepper noises
- ¹⁴segm. por Otsu method, determinou as coordenadas do centroide e criou uma linha de eixo
- ¹⁵este valor corresponde às imagens originais - aproximadamente 150 amostras foram segmentadas de cada imagem.
- ¹⁶imagens para treinamento duplicadas com horizontally-flipped
- ¹⁷data augmentation aplicado para incrementar o número de imagens via transformações geométricas
- ¹⁸800 para treinar YOLOv3 e 8000 para treinar a YOLOv1
- ¹⁹usando o algoritmo super pixel
- ²⁰novas imagens foram criadas com transformações geométricas e diferentes backgrounds
- ²¹imagens geradas pelo microscópio
- ²²usando Subspace Classifier e Limited Receptive Area)
- ²³após aplicação de flipping e color jittering, o tamanho do conjunto de treinamento foi aumentado em 12 vezes.
- ²⁴detector foi aprimorado por k-means anchor optimization e sub-rede de classificação residual
- ²⁵mais imagens foram criadas com aumento de dados por intensidade e transformações geométricas aleatórias
- ²⁶4000 imagens geradas após normalização e processamento
- ²⁷Optimal global thresholding, Otsu, erosion e morphological opening operations
- ²⁸4800 imagens geradas por técnicas de aumento de dados
- ²⁹10.000 imagens geradas por rotação, dimensionamento, inversão, translação e outras operações para treinar o YOLO. 17.000 imagens geradas por aumento de dados para treinar o SVM

A segunda coluna (Aprendizado de Máquina) da Tabela 1 indica que (63%) dos estudos utilizaram uma abordagem convolucional baseada em aprendizado profundo para a implementação de uma rotina de reconhecimento. Outros trabalhos (29%), aplicaram métodos de extração manual de características (*hand-crafted features*) e somente (8%) mencionaram a utilização de uma abordagem baseada exclusivamente em técnicas de processamento de imagens, como segmentação por contornos, regiões, Otsu ou k-means. Esses percentuais confirmam a tendência pela utilização de abordagens de aprendizado profundo, como DCNNs, no reconhecimento de insetos e doenças de plantas.

A terceira coluna (Processamento de Imagens) na Tabela 1 refere-se a utilização de técnicas de PI. Identificou-se que (80%) dos artigos aplicaram esses recursos para o pré-processamento de imagens, como correção de brilho, contraste, saturação, recorte, redimensionamento ou remoção de ruídos/objetos menores que os de interesse através de operações morfológicas [17]. Algoritmos de PI também foram utilizados para extrair dados relevantes nas imagens (*hand-crafted features*). A conversão para escalas de cinza foi aplicada em alguns estudos para resolver o problema da variação de luminosidade, no caso de aquisições em ambientes externos, ou para eliminar a cor do processo de extração de características, visando a redução da complexidade computacional. Nazri *et al.* [17] foi o único que mesclou PI e DCNN ao implementar uma fase de transformação usando operações de filtragem binária (*Euclidean Distance Maps*) antes de enviar os dados para a DCNN.

Com base nesse estudo, entende-se que o uso do PI para pré-processamento seja um recurso importante para a padronização das imagens de entrada, principalmente quando se considera aquisições em ambientes externos.

Conforme os dados apresentados nas colunas "Sobreposição" e "Pequenos Objetos" da Tabela 1, apenas (27%) dos trabalhos apresentaram implementações específicas para tratar sobreposições e otimizar o reconhecimento de pequenos objetos. Alguns autores afirmaram que essas situações não eram relevantes para seus estudos de caso. As colunas indicam as técnicas aplicadas para o tratamento de sobreposição, toque, conexão e aprimoramento do processo de extração de dados em pequenos objetos.

De acordo com a Tabela 1, apenas cinco estudos utilizaram mais de dez classes de objetos de interesse. Zhong *et al.* [25] criou um classificador para cada espécie de insetos, para facilitar a inclusão de novas espécies e eliminar a necessidade de retreinamento. A abordagem de Liu *et al.* [59] apresentou uma precisão média de apenas 75,5%, devido à similaridade entre algumas espécies de insetos.

A coluna "Conjunto de Imagens" na Tabela 1 indica o tamanho do conjunto de dados de treinamento (número de imagens ou amostras usadas). Em alguns trabalhos, as anotações adicionais descrevem o número de amostras geradas pela técnica de aumento de dados (*data augmentation*) ou recorte (*crop*). Nesse sentido, verificou-se que *data augmentation* foi comumente utilizado pelos autores através de transformações geométricas

e de intensidade para geração de novas imagens. Outra técnica citada foi a utilização de modelos pré-treinados no ImageNet ou PlantVillage, como parâmetros iniciais do modelo.

Alguns estudos baseados em Rede Neural Artificial (ANN) e Máquina de Vetores de Suporte (SVM) mostraram altos níveis de precisão usando pequenos conjuntos de dados quando comparados a outros [52, 49, 60], o que nos leva a considerar sobre sua capacidade de generalização em cenários reais.

A pesquisa constatou que apenas dois estudos selecionados [57] [69] combinaram técnicas de ML com PI para o tratamento de conexões ou sobreposições parciais. Outros três estudos que usaram apenas técnicas de PI para reconhecimento de objetos [50, 63, 67], foram os únicos que focaram no problema das conexões e sobreposições parciais de insetos ou células em imagens digitais. Esses autores exploraram combinações de algoritmos e parâmetros para criar uma solução personalizada para segmentar corretamente os objetos de interesse.

A Tabela 2 detalha somente os trabalhos que aplicaram abordagens baseadas em DCNN para o reconhecimento de insetos, bem como, a tarefa de análise e a configuração da rede neural.

2.2.1 Considerações sobre as implementações com DCNN

DCNNs foram empregadas para tarefas de classificação, segmentação e detecção de objetos em imagens (Tabela 2). Como referido, na tarefa de classificação de imagens, deve haver apenas um padrão de objeto no centro da imagem, não sendo recomendado quando há objetos conectados. Com a detecção de objetos, é possível reconhecer vários objetos em uma determinada imagem. No entanto, um grande conjunto de dados é necessário para treinar e testar o modelo, onde cada objeto de interesse deve ser identificado e rotulado [19].

Na Tabela 2 são listados os estudos baseados em DCNN, ordenados pela tarefa e abordagem convolucional. Verificou-se que a maioria dos estudos utilizou a detecção de objetos ou classificação de imagens. De acordo com a pesquisa, a abordagem de dois estágios com RPN foi a mais usada para detecção de objetos. Apenas dois estudos aplicaram classificadores de um estágio, em razão de recursos de hardware limitados: Sun *et al.* [19] desenvolveu software embarcado e Partel *et al.* [18] implementou uma contagem superficial.

³⁰one-stage method with 106 convolutional layers - Darknet-53 - to object detection

³¹24 convolutional layers, 4 max-pooling layers and two fully connected layers - inspired by the GoogLeNet - to image classification

³²convolution, maxpooling, reshape, fully connected, unpooling and deconvolution layers

³³Global contrast based on salient region detection to automatically find the regions of object in images

³⁴CONV and POOL layers

³⁵video processing based with three modes: early fusion, late fusion, and slow fusion

³⁶5-layer CNN

³⁷4 filters in the first convolution layer and 4 as the depth of contracting path

Tabela 2. Trabalhos seleccionados pela revisão que aplicaram Redes Neurais Convolucionais Profundas

Ref	Tarefa	Abordagem	Rede	Geração de Objetos Candidatos
[59]	obj. detect.	Faster R-CNN	ResNet101	RPN
[71]	obj. detect.	Faster R-CNN	VGG16	RPN
[66]	obj. detect.	Faster R-CNN	VGG16	RPN
[56]	obj. detect.	Faster R-CNN	Inception V2	RPN
[27]	obj. detect.	Faster R-CNN	VGG16	RPN
[26]	obj. detect.	Sliding window	ZF	RPN
[70]	obj. detect.	Sliding window	VGG19	RPN
[18]	obj. detect.	Single-stage	YOLOv3 ³⁰ + YOLOv1 ³¹	n/a
[19]	obj. detect.	Single-stage	RetinaNet	n/a
[15]	obj. detect.	Autoencoder	custom ³²	selectivity function
[58]	obj. detect.	Global contrast region ³³	AlexNet	salient region
[16]	img. classif.	Sliding window	LeNet-5	n/a
[17]	img. classif.	Sliding window	VGG16	n/a
[62]	img. classif.	Sliding window	custom ³⁴	n/a
[54]	img. classif.	Abstraction-Level Fusion ³⁵	LeNet	n/a
[64]	img. classif.	Multi-level ftr. extr.	Inception v3	n/a
[53]	img. classif.	Sliding window	ResNet101	n/a
[68]	img. classif.	Sliding window	custom ³⁶	n/a
[61]	img. classif.	FCN	Resnet-50	n/a
[32]	segm.	U-Net	CNN ³⁷	n/a

Verificou-se que para a tarefa de classificação de imagens, a abordagem mais utilizada foi a Janela Deslizante (*Sliding window*). Para a detecção de objetos, o método Faster R-CNN foi o mais aplicado. Poucos estudos aplicaram detectores de um estágio [19, 18, 25]. Também identificou-se outras técnicas, como Autoencoder, U-Net para segmentação, fusão em nível de abstração (*abstraction-level fusion*) e extrator de recursos multinível.

Customizações são geralmente aplicadas para o aprimoramento da capacidade de generalização de DCNNs, pois possibilitam adaptar o aprendizado de acordo com as peculiaridades de um determinado contexto, como tamanho do objeto, disponibilidade de imagens para cada classe, número de camadas da imagem, etc. Como exemplo, Nazri *et al.* [17] aplicaram uma redução no número de classes, de mil para duas, em razão da necessidade de identificar somente duas espécies de cigarras.

Nesse sentido, a investigação também apurou que as mudanças mais relevantes foram aplicadas para melhorar o processo de extração de informações, permitindo a identificação de pequenos objetos nas imagens. Verificamos também a preocupação dos autores na escolha das arquiteturas devido à disponibilidade de imagens para treinamento, visto que as DCNNs demandam um grande conjunto de dados de imagens. Além disso, constatou-se que a maioria dos estudos apresentou algum tipo de alteração, seja na utilização de modelos para pré-treinamento de parâmetros, alterações no número de camadas, ou no processo de extração de características, entre outras alterações.

Sobre as abordagens para detectar pequenos objetos ou em diferentes poses, destaca-se três estudos: Shen *et al.* [66] propôs uma mudança na rede de criação (inspirada em He *et al.* [72]) para gerar mapas de características em camadas com 1/8 da imagem original, tornando a rede sensível a pequenos objetos; Yue *et al.* [71] apresentou

contribuições para auxiliar a detecção de objetos usando um modelo de super-resolução baseado em uma rede residual recursiva profunda para reconstrução, onde a imagem poderia ser restaurada e visualizada de acordo com uma determinada escala, então o inseto alvo pode se adaptar ao método de detecção de objeto com base em uma escala de pixels; Wen *et al.* [69] considerou um método de identificação dependente da estimativa de pose para imagens de mariposas de campo usando arquitetura de aprendizado profundo.

A resolução das imagens de entrada é um aspecto importante na definição da arquitetura de aprendizado profundo, bem como, o número de camadas e tamanho do *kernel*. Maiores resoluções elevam o tempo computacional e podem não aumentar significativamente a precisão do modelo Shen *et al.*. No entanto, a baixa resolução torna quase impossível a extração de características relevantes para distinguir objetos de interesse. Verificou-se também que o uso de modelos pré-treinados possibilitou elevar a precisão e reduzir o tempo computacional.

Dois outros estudos apresentaram diferentes abordagens: Akintayo *et al.* [15] propôs uma abordagem de ponta-a-ponta chamada *Convolutional Selective Autoencoder* e uma função de seletividade, que é um processo intermediário entre a segmentação *pixel-wise* e *superpixel*, que aprendeu a rotular apenas as *bounding box* que continham objetos de interesse. Liu *et al.* [58] apresentou uma abordagem baseada na região de contraste global que calculou um mapa de saliência para encontrar objetos como insetos-praga.

Em relação às técnicas de tratamento de conexões ou sobreposições, verificou-se que nenhum método específico de DCNN foi capaz de solucionar definitivamente esse problema. Partel *et al.* [18], Nazri *et al.* [17] e Liu *et al.* [59] não consideram casos de conexões ou sobreposição de objetos, devido ao tipo de armadilha. Porém, analisando alguns estudos baseados em processamento de imagens [50, 67, 63], conclui-se que a técnica proposta por Bakkay *et al.* [50], que descreve a segmentação baseada em contornos e regiões, apresentou resultados promissores.

Considerando as discussões sobre o problema de conexão/sobreposição, é importante ressaltar que os dados extraídos de imagens digitais bidimensionais, em RGB ou GrayScale, podem ser insuficientes para que DCNNs ou algoritmos de PI resolvam situações de sobreposições parciais ou totais. Portanto, pode ser necessário obter informações usando outros recursos, como movimento de objetos [73], análise espectral, medidas polarimétricas multibanda [74] e dados espaço-temporais [75]. Porém, essas alternativas requerem equipamentos não convencionais, aumentando os custos da solução.

Com base na necessidade de automação do processo manual de separação de insetos realizado em centros de pesquisa (como a Embrapa Trigo), na disponibilidade de imagens provenientes da digitalização das amostras e na fundamentação teórica apresentada anteriormente, DCNN se configura como uma alternativa adequada para detecção de insetos. As imagens contém centenas de objetos de interesse, detritos e outros insetos

alados, como moscas e cigarras. Não são raros os casos de sobreposição ou conexão de objetos nas imagens.

Pelo fato do Mask R-CNN ter sido pouco explorado, optou-se pela avaliação desse algoritmo, na tarefa de detecção de objetos como afídeos e parasitoides em imagens digitais. Além da localização e classificação, a segmentação de instâncias poderá auxiliar na solução do problemas das conexões e sobreposições, de forma mais eficiente que a Faster R-CNN, uma vez que poderá ser delimitada a área de cada objeto reconhecido. Além disso, a segmentação possibilitará o cálculo da área dos objetos nas imagens.

Conforme recentes publicações, Mask R-CNN tem sido empregada para segmentar objetos em imagens para diferentes finalidades. Em [76] foram utilizadas 1280 imagens para treinar e outras 320 para validar um modelo inteligente com o objetivo de classificar e segmentar defeitos na superfície de placas de madeira. A dimensão da imagem foi de 512 x 512 pixels. A precisão média (mAP) do modelo foi superior a 97%. No artigo [77] foi relatada a utilização de Mask R-CNN, em combinação com um algoritmo de agrupamento (DBSCAN), para segmentar folhas de mudas de poplar (bugreiro). Nesse trabalho, o conjunto de imagens para treinamento foi de 1200 e 400 para teste. A resolução das imagens foi de 1920 x 1080 pixels. Cada imagem tinha aproximadamente 15,4 folhas. A precisão média obtida foi de 0,91 utilizando um limiar de 0,7.

Com base na revisão sistemática apresentada anteriormente, não foi identificado nos trabalhos selecionados uma solução computacional definitiva para tratar conexões e sobreposições. No entanto, não é possível descartar a existência de métodos mais recentes ou customizações aplicadas nas abordagens da família R-CNN capazes de resolver esse problema. Na próxima subseção serão apresentados os detalhes do Mask R-CNN.

2.2.2 Mask R-CNN

Proposto por He *et al.* [38] em 2017, Mask R-CNN é um *framework*, de código aberto³⁸, que estende as funcionalidades do Faster R-CNN. Segundo os autores, possui uma estrutura conceitualmente simples e flexível para a segmentação de instâncias de objetos, como poses humanas, podendo ser customizado para outros tipos de objetos. Além da predição da classe e geração das *bounding boxes*, definiu uma máscara de segmentação de alta qualidade para cada instância, em paralelo. Testes realizados com o conjunto de dados COCO 2016 (<https://cocodataset.org>) demonstram que Mask R-CNN supera os métodos vencedores do Desafio COCO 2016 nas tarefas de segmentação de instâncias e detecção de objetos por *bounding box*.

Segundo os autores, a segmentação de instâncias é uma tarefa desafiadora, pois requer a correta detecção de todos os objetos em uma imagem, enquanto precisa também

³⁸Disponível em <https://github.com/facebookresearch/Detectron>

segmentar cada instância. Esse processo combina elementos clássicos da tarefa de detecção de objetos de VC, onde o objetivo é classificar objetos individuais e localizar cada um usando uma *bounding box* e segmentação semântica, cuja finalidade é detectar cada pixel dentro de um conjunto de categorias pré-definidas sem diferenciar instâncias de objetos.

Mask R-CNN possui uma *branch* adicional, em comparação ao Faster R-CNN, para a predição de máscaras de segmentação em cada região de interesse (RoI). Isso ocorre em paralelo com a *branch* que realiza a classificação e regressão das *bounding box*. A Figura 1 ilustra os passos para a geração das três saídas (classes, caixas e máscaras). Essa *branch* é uma pequena rede FCN³⁹, aplicada a cada RoI, prevendo máscaras de segmentação a cada pixel. Os autores destacam que essa *branch* adicional não aumenta significativamente a complexidade computacional. Além disso, ressaltam a facilidade para aplicar padrões de arquiteturas flexíveis.

Inicialmente a imagem de entrada é enviada para uma rede convolucional (*conv net*), onde são aplicados filtros para a geração de mapas de características (*feature maps*) (Figura 1). Esses filtros são espacialmente menores quando comparados a imagem. Múltiplas convoluções são realizadas em paralelo com a aplicação de funções de ativação não lineares (RELU). Os mapas representam bordas detectadas, diferentes formas, curvas ou cores.

Em seguida, os *feature maps* são encaminhados para a RPN que fará a geração de diferentes regiões de interesse (*anchor boxes*), baseados na presença de qualquer objeto que possa ser detectado.

No próximo passo, os *anchor boxes* são encaminhados para o estágio de *RoI-Align* que converte as RoI's para o tamanho requerido no restante do processo. Esse é um ponto chave do Mask R-CNN, que mantém a orientação espacial. A saída é enviada para as camadas *Fully connected* que irão identificar a classe do objeto e especificar a sua localização através das *bounding boxes*. Em paralelo, a saída da *RoIAlign* também é encaminhada para as *conv net* onde serão geradas as máscaras por pixel dos objetos.

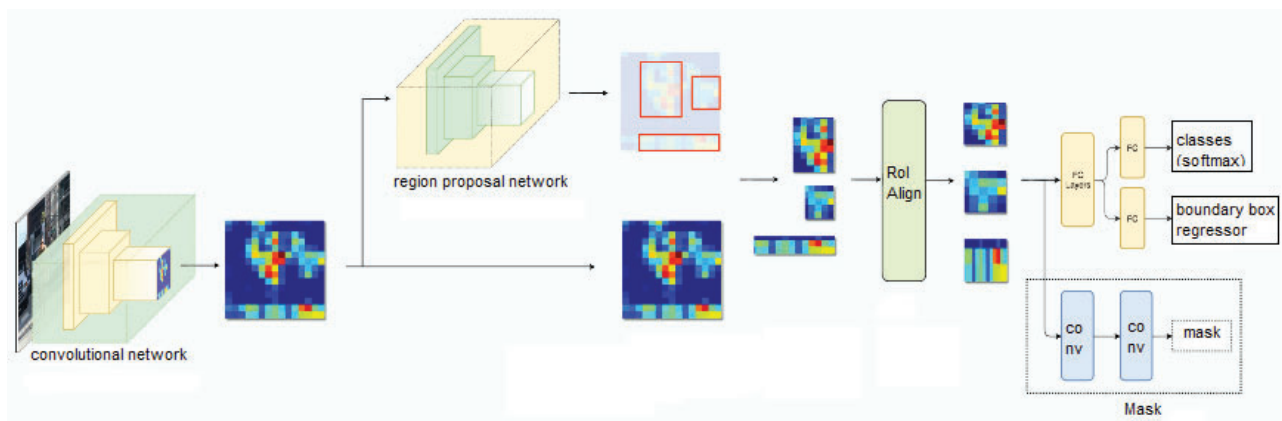


Figura 1. Estrutura da Mask R-CNN reproduzida a partir de [78].

³⁹Esse tipo de rede é semelhante ao classificador de dois estágios citado por Sun *et al.* [19] e implementado por Picon *et al.* [61]

Considerando que o Faster R-CNN não foi projetado para o alinhamento pixel-a-pixel entre as redes de entrada e saída, pois o *RoIPool* realiza apenas uma quantização espacial superficial para a extração de características, no Mask R-CNN, os autores propuseram uma camada simples, livre de quantização, chamada *RoIAlign*, que preserva fielmente as localizações espaciais. Além disso, com base nos experimentos realizados, identificou-se a necessidade de desacoplar o processo de geração de máscaras da predição das classes, através da criação independente de máscaras binárias para cada RoI, sem competição entre as classes e contando com a rede RoI para prever a categoria.

Durante o treinamento, para cada RoI é aplicada a seguinte fórmula para calcular a taxa de perda: $L = L_{cls} + L_{box} + L_{mask}$. Sendo que L_{cls} refere-se ao valor da perda no reconhecimento das classes. L_{box} corresponde ao valor da perda no reconhecimento das *bounding boxes* e L_{mask} é o valor de perda na predição das máscaras.

Para demonstrar a capacidade de generalização da proposta, os autores avaliaram diferentes composições de arquitetura de rede, como: (i) arquitetura convolucional *backbone*, também conhecida como *network-depth-features* para a extração de características sobre uma imagem inteira; (ii) rede *head* para reconhecimento das *bounding boxes* (classificação e regressão) e predição das máscaras para cada RoI.

A Figura 2, criada por [38], ilustra duas possíveis arquiteturas *head* para os *backbone* ResNet C4 [72] e FPN [79], as quais receberam a *branch* para a geração das máscaras. Os números indicam a resolução espacial e a quantidade de canais. As setas indicam camadas convolucionais (*conv*), deconvolucionais (*deconv*) ou totalmente convolucionais (*fc*). Camadas *conv* preservam a dimensão espacial, enquanto que as *deconv* incrementam. Na arquitetura apresentada no lado esquerdo a predição das máscaras é incluída após o quinto estágio convolucional (*res5*). Na outra arquitetura, o (x4) indica que a máscara é gerada após quatro camadas convolucionais consecutivas.

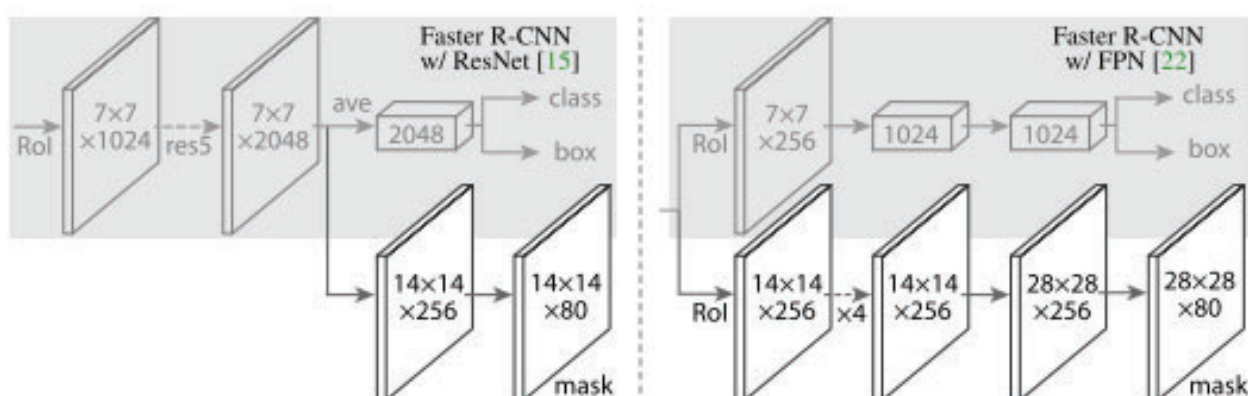


Figura 2. Arquiteturas Head para Mask R-CNN [78].

Segundo os autores, utilizando o *backbone* ResNet-FPN [72, 79] para a extração de características alcançou-se excelentes ganhos em termos de precisão e velocidade.

Para chegar a esse entendimento, também foram avaliados as redes ResNet [72] e ResNeXt [80] com profundidade de 50 ou 101 camadas. Sendo assim, as variações de rede que apresentaram os melhores resultados foram ResNet-101-FPN e ResNeXt-101-FPN.

Com relação a sobreposição de objetos, Mask R-CNN alcança bons resultados nessas situações, consideradas desafiadoras [38]. Em comparação com FCIS +++ [81], percebeu-se que esse último exibe artefatos entre instâncias sobrepostas, em função da dificuldade em segmentar essas instâncias. Entretanto, Mask R-CNN não mostra tais artefatos.

2.2.2.1 Hiper-parâmetros

O conjunto de hiper-parâmetros definidos nos trabalhos de Fast/Faster R-CNN [47, 34, 79] para detecção de objetos também são eficientes para segmentação de instâncias [38]. Os principais hiper-parâmetros são:

- Backbone;
- TRAIN_ROIS_PER_IMAGE;
- MAX_GT_INSTANCES;
- DETECTION_MIN_CONFIDENCE;
- IMAGE_MIN_DIM e IMAGE_MAX_DIM;
- DETECTION_MAX_INSTANCES;
- RPN_ANCHOR_SCALES;
- Pesos de perda: rpn_class_loss;
- Pesos de perda: rpn_bbox_loss;
- Pesos de perda: mrcnn_class_loss;
- Pesos de perda: mrcnn_bbox_loss;
- Pesos de perda: mrcnn_mask_loss.

Como referido, o *backbone* é a arquitetura da *conv net* utilizada na primeira etapa da Mask R-CNN. As opções disponíveis são: ResNet50, ResNet101 e ResNext101. A escolha deve ser baseada no equilíbrio entre tempo de treinamento e precisão. Nesse sentido, ResNet50 é relativamente mais rápida que as demais, além disso, pode ser pré-treinada com o conjunto de dados COCO, o que reduz consideravelmente o tempo para o

treinamento [82]. ResNet101 e ResNext101 necessitam de mais tempo de treinamento, em razão do número de camadas, porém, elas tendem a ser mais precisas se não houver pesos pré-treinados e parâmetros básicos como taxa de aprendizagem *learning rate* e número de épocas *epochs* bem ajustados [83]. Na Figura 3 é ilustrada a relação entre os hiper-parâmetros com a arquitetura Mask R-CNN.

O TRAIN_ROIS_PER_IMAGE é o número máximo de RoI's que a RPN gerará para cada imagem, as quais serão processadas no segundo estágio. Se o número de instâncias for limitado, o tempo de treinamento será reduzido [83].

O hiper-parâmetro MAX_GT_INSTANCES corresponde ao número máximo de instâncias que podem ser detectadas em uma imagem. Se o número for limitado, é possível reduzir os casos de falsos positivos e o tempo de treinamento [83]. O DETECTION_MAX_INSTANCES limita o número máximo de objetos que podem ser detectados no processo de inferência.

O hiper-parâmetro DETECTION_MIN_CONFIDENCE define o nível mínimo de confiança para uma determinada instância. A redução desse parâmetro aumentará o número de instâncias detectadas pelo modelo. Isso pode ser relevante, se for necessário detectar qualquer objeto e falsos positivos não sejam um problema. No entanto, se a precisão na detecção for importante, o incremento desse parâmetro reduzirá o número de falsos positivos, garantindo que o modelo faça predição apenas de instâncias com alta confiança.

Os parâmetros IMAGE_MIN_DIM e IMAGE_MAX_DIM controlam o tamanho da imagem de entrada. O valor padrão para o redimensionamento, pelo modo *square* é 1024x1024. Imagens menores (512x512) podem reduzir a alocação de memória e o tempo de treinamento. Bobba [83] recomenda o treinamento dos primeiros modelos com imagens menores, permitindo a atualização mais rápida dos pesos e utilizar imagens maiores na fase final para ajustar os parâmetros do modelo final.

De acordo com a dimensão das imagens e escala dos objetos, pode ser necessário ajustar o hiper-parâmetro RPN_ANCHOR_SCALES, com base nos pesos de perda gerados no processo de treinamento da rede.

Como referido, Mask R-CNN aplica uma função de perda *loss function*, que faz um somatório dos pesos *Weights* de diferentes perdas em cada um dos estágios do modelo. Os hiper-parâmetros de peso de perda *loss weight* correspondem ao valor que o modelo deve atribuir a cada uma das etapas.

O *rpn_class_loss* corresponde a perda atribuída na classificação inadequada de *anchor boxes* (presença ou falta de qualquer objeto) pela RPN. Segundo Bobba [83], esse parâmetro deve ser incrementado quando vários objetos não estão sendo detectados pelo modelo no final da saída.

O rpn_bbox_loss é relacionado a acurácia de localização pela RPN. Esse peso deve ser ajustado, quando o objeto é detectado e a *bounding box* não está bem posicionada [83].

O $mrcnn_class_loss$ é vinculado a perda atribuída para a classificação inadequada de objetos que estão presentes na região proposta *region proposal*. Deve ser incrementado no caso de objetos que estejam sendo detectados na imagem, mas classificados incorretamente [83].

O $mrcnn_bbox_loss$ é a perda atribuída na localização de uma *bounding box* na identificação da classe. Deve ser incrementado se a classificação do objeto for correta e a sua localização não for precisa [83].

O parâmetro $mrcnn_mask_loss$ corresponde as máscaras criadas na identificação dos objetos. Se a identificação no nível de pixel for importante, esse peso deve ser incrementado [83].

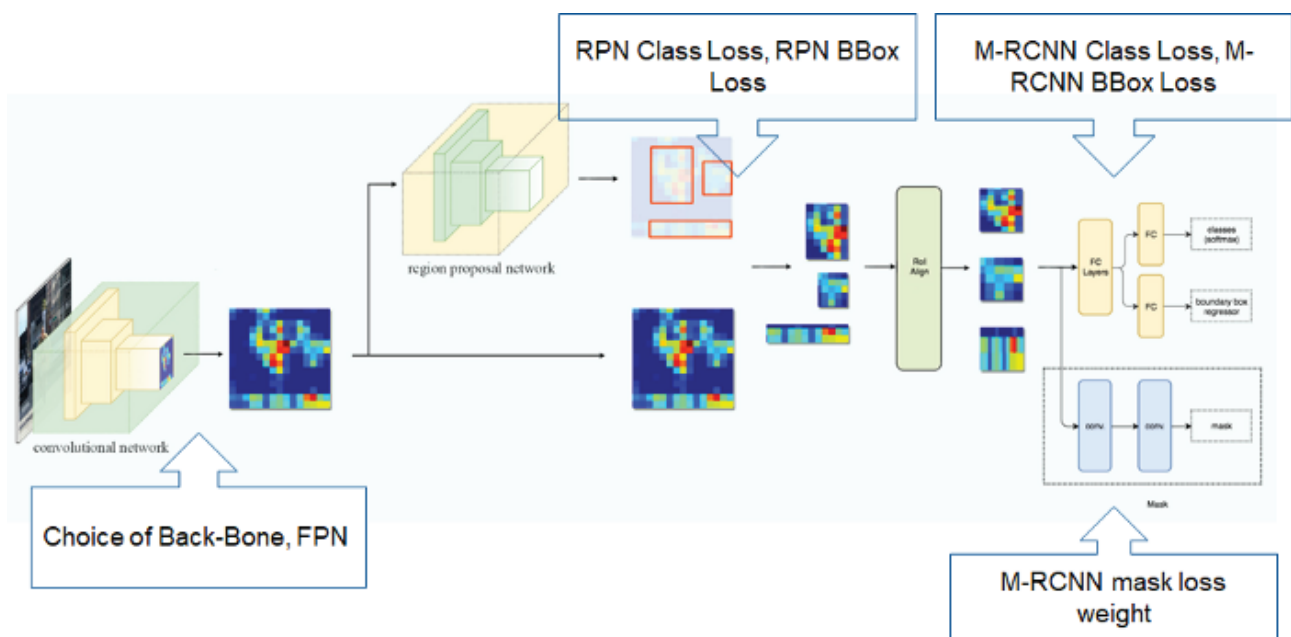


Figura 3. Hiper-parâmetros e a arquitetura Mask R-CNN [83].

Com base nas contribuições de Bobba [83], a estratégia mais adequada para o treinamento do modelo é iniciar com o valor padrão 1 para cada parâmetro de pesos de perda. Além disso, deve-se avaliar o desempenho do modelo no conjunto de validação, visualizando a detecção em diferentes imagens, considerando o número de objetos detectados, a precisão dos objetos classificados, bem como, a localização das *bounding boxes* e das máscaras. Então, os valores dos parâmetros correspondentes deverão ser alterados baseados no desempenho do modelo.

2.2.2.2 Otimizações na utilização do Hardware

Com base nas contribuições de Abdulla [82], Mask R-CNN foi projetado visando a precisão e não a redução no consumo de memória. Portanto, recomenda-se utilizar uma GPU com 12 GB de memória RAM ou mais. No entanto, é possível treinar um modelo com GPUs menores, escolhendo configurações e compensações adequadas. A implementação de Mask R-CNN desenvolvida por Abdulla [82] disponibiliza apenas os *backbones*: ResNet101 (padrão) e o ResNet50. A necessidade de alocação de memória no ResNet50 é significativamente menor em relação ao *backbone* padrão.

A estratégia de treinar menos camadas também pode reduzir o consumo de memória, evitando interrupções no treinamento, em virtude de falha na alocação de memória (*Out of Memory error*). A utilização de modelos pré-treinados (como o COCO) possibilita iniciar as primeiras camadas com pesos não aleatórios. Isso permitirá a extração de características de baixo nível de forma mais rápida, especialmente se os objetos do conjunto de imagens forem semelhantes as do modelo pré-treinado.

A utilização de pequenas imagens reduz a necessidade de memória e o tempo para treinamento do modelo. Por padrão, as imagens de entrada são redimensionadas através do modo *square* para 1024x1024. A referida implementação disponibiliza outros modos, como *crop* que realiza recortes aleatórios de determinado tamanho. O treinamento com imagens recortadas é mais eficiente em relação a utilização de imagens inteiras, bem como, evita possíveis distorções geradas pelo redimensionamento.

Outro parâmetro que afeta diretamente o consumo de memória é o *batch size*, que define a quantidade de imagens processadas por GPU em um dado momento. A quantidade de RoI's (Train_ROIs_Per_Image) no treinamento no segundo estágio é outro fator importante relacionado ao consumo de memória, bem como, a quantidade máxima de instâncias (Max_GT_Instances).

A utilização de imagens em escalas de cinza *GrayScale* pode ser uma alternativa para reduzir o consumo de memória, principalmente quando utiliza-se imagens de alta resolução. Nesse caso, não é possível utilizar os pesos de modelos pré-treinados nas camadas conv1, pois, o COCO e ImageNet trabalham com imagens no formato RGB. Sendo assim, essas camadas precisam ser obrigatoriamente retreinadas.

Com base no detalhamento da arquitetura e dos principais hiper-parâmetros, o próximo Capítulo apresenta materiais e métodos aplicados para a realização de sete experimentos utilizando Mask R-CNN. Eles consideram diferentes configurações de resolução de imagem, aumento de dados e parâmetros, para a obtenção de um modelo capaz de reconhecer insetos, como parasitoides e afídeos. Além disso, são descritos os recursos utilizados para a implementação dos serviços web para armazenamento e preparação das imagens, definição do conjunto de imagens para treinamento e integração desse sistema no ambiente web Trap System.

3. MATERIAIS E MÉTODOS

Nesse capítulo são detalhados os procedimentos e recursos utilizados para o desenvolvimento dos componentes do sistema *InsectCV* e validação do modelo inteligente. A Figura 4 sintetiza as etapas de projeto e desenvolvimento do *InsectCV*.

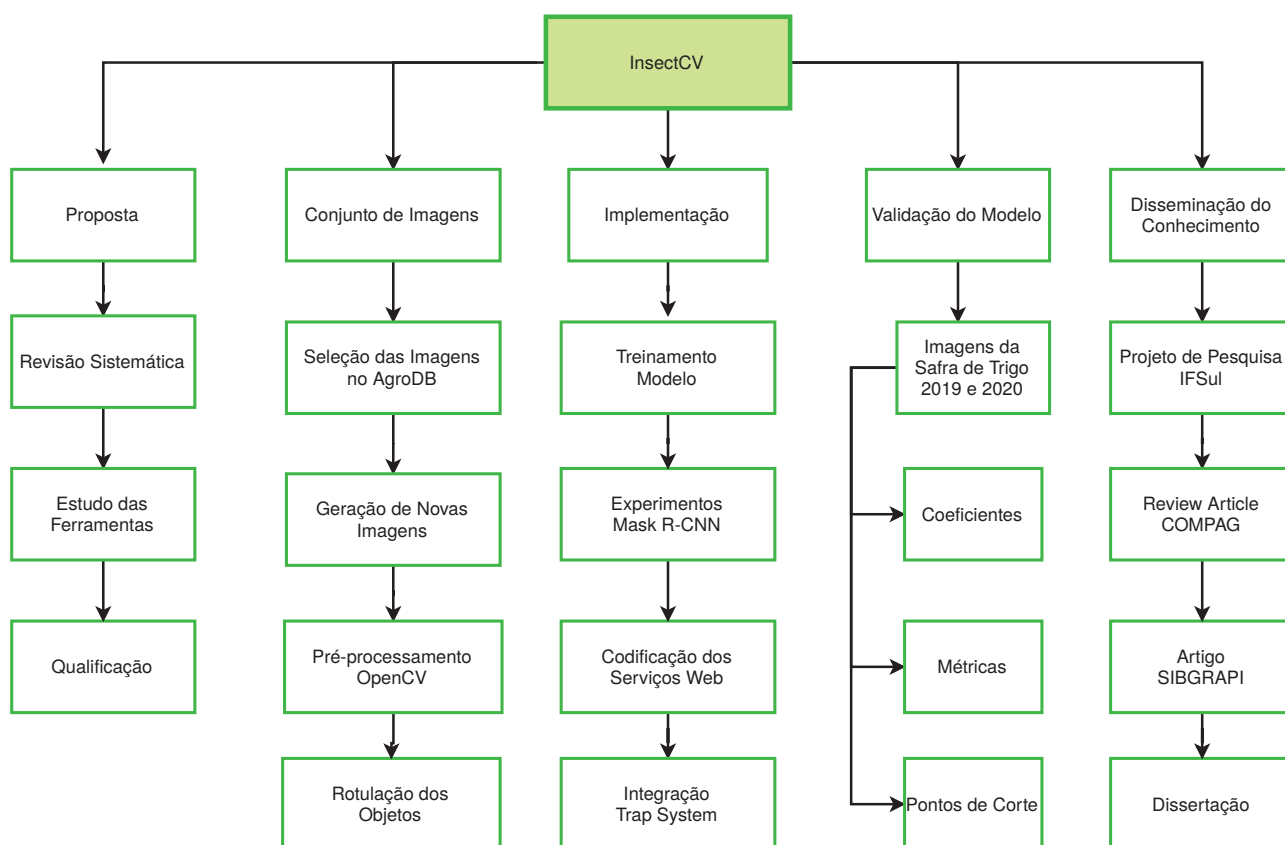


Figura 4. Estrutura Analítica do Trabalho.

Para formar os conjuntos de imagens, foram geradas em laboratório 119 novas imagens contendo parasitoides (*Hymenoptera: Aphelinidae e Braconidae, Aphidiinae*) e afídeos alados (*Hemiptera, Aphididae*). O processo de aquisição seguiu os procedimentos definidos por Lins *et al.* [21]. Além dessas, foram selecionadas outras 90 imagens contendo insetos de interesse e detritos, armazenadas no banco de dados AgroDB - um banco de dados genérico que funciona como um repositório de dados necessários para simulações de culturas agrícolas [84]. A partir das 209 imagens, foram rotulados 17.908 objetos. O detalhamento desses conjuntos é apresentado na Tabela 3 e na Tabela 4.

Para o treinamento e teste do modelo, foram definidos dois conjuntos de imagens. O primeiro com 167 imagens, 14.809 insetos rotulados, sendo 12.354 para treinamento e 2.455 para teste do modelo. Para a formação do segundo conjunto, foram utilizadas as imagens do primeiro e incluídas novas imagens, chegando-se nas 209, como referido.

Em cada conjunto, as imagens estão organizadas nos seguintes subconjuntos: (i) afídeos alados; (ii) parasitoides; (iii) afídeos alados e parasitoides, (iv) afídeos alados, parasitoides e detritos; (v) afídeos alados e detritos. Com o auxílio da ferramenta gráfica Labellmg ⁴⁰ foram rotulados todos os objetos de interesse, classificando-os em duas classes: afídeo ou parasitoide. Essa tarefa foi supervisionada por um especialista da Embrapa.

Na Tabela 3 é apresentado o detalhamento do primeiro conjunto de imagens. Apesar do pequeno conjunto para a etapa de aprendizado profundo, foi rotulada uma quantidade significativa de objetos. Apenas os subconjuntos (iv) e (v) contém imagens geradas pelo processo de triagem, de 2018 a 2020, em armadilhas instaladas nos municípios de Passo Fundo, Coxilha e Cruz Alta, RS, Brasil. Os demais foram criados artificialmente com a inserção manual dos espécimes na lâmina sem a presença de detritos.

Tabela 3. Primeiro conjunto de imagens.

Conj.	Afídeos	Parasit.	Insetos / Imagens Train.	Insetos / Imagens Test.	Total
I	4934	0	4688/ 44	246/ 2	4934/ 46
II	0	6137	5942/ 43	195/ 2	6137/ 45
III	1745	1611	1506/ 13	1850/ 15	3356/ 28
IV	77	15	10/ 1	82/ 5	92/ 6
V	290	0	208/ 33	82/ 9	290/ 42
Total	7046	7763	12354/ 134	2455/ 33	14809/ 167

A Tabela 4 lista o segundo conjunto, que possui 209 imagens. Destaca-se que nesse novo conjunto os cinco subgrupos foram balanceados com a inclusão de novas imagens, para alcançar a proporção de 80% de imagens para treinamento e 20% para teste. As imagens de teste foram escolhidas conforme a quantidade de objetos, de forma que houvesse a mesma proporção entre imagens com diferentes densidades de insetos, consideradas de fácil reconhecimento (menos densas, Figura 5) e de médio reconhecimento (mais densas, Figura 6). Sendo assim, a quantidade de insetos rotulados passou para 17.908, sendo 14.669 para treinamento e 3.239 para teste.

Tabela 4. Segundo conjunto de imagens.

Conj.	Afídeos	Parasit.	Insetos / Imagens Train.	Insetos / Imagens Test.	Total
1	6101	0	5035/ 36	1066/ 9	6101/ 45
2	0	6311	5154/ 36	1157/ 9	6311/ 45
3	1950	1646	2884/ 23	712/ 6	3596/ 29
4	1093	168	1042/ 34	219/ 8	1261/ 42
5	639	0	554/ 38	85/ 10	639/ 48
Total	9783	8125	14669/ 167	3239/ 42	17908/ 209

Para a geração do modelo inteligente foi utilizada a implementação de Mask R-CNN de Abdulla [82], bem como, a linguagem de programação Python e as bibliotecas OpenCV ⁴¹, TensorFlow ⁴² e Keras ⁴³. Em termos de recursos de hardware, utilizou-se um computador com processador Intel Core I7-6950X, 32 GB de RAM e a unidade de

⁴⁰Disponível em: <https://github.com/tzutalin/labelImg>

⁴¹Disponível em: <https://opencv.org/>

⁴²Disponível em: <https://tensorflow.org>

⁴³Disponível em: <https://keras.io>

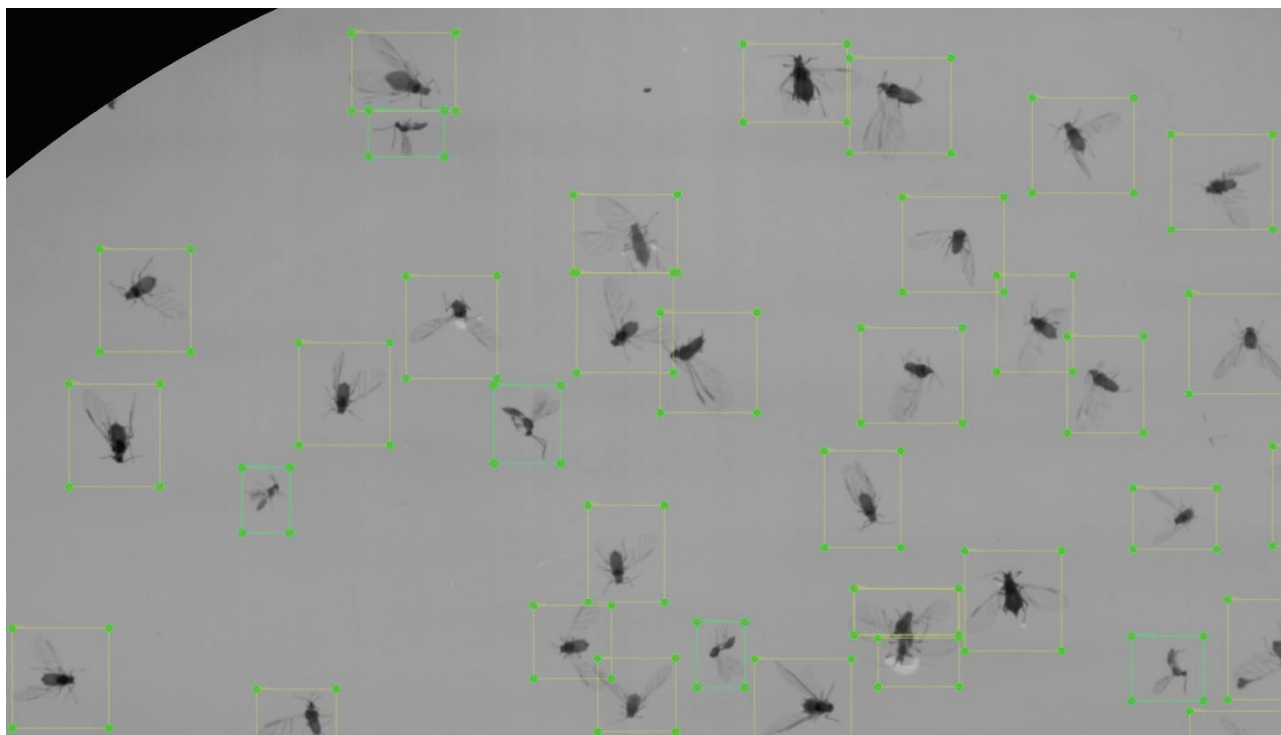


Figura 5. Imagem com afídeos, parasitoides, detritos e poucas conexões. Caixas verdes indicam parasitoides e amarelas representam afídeos.

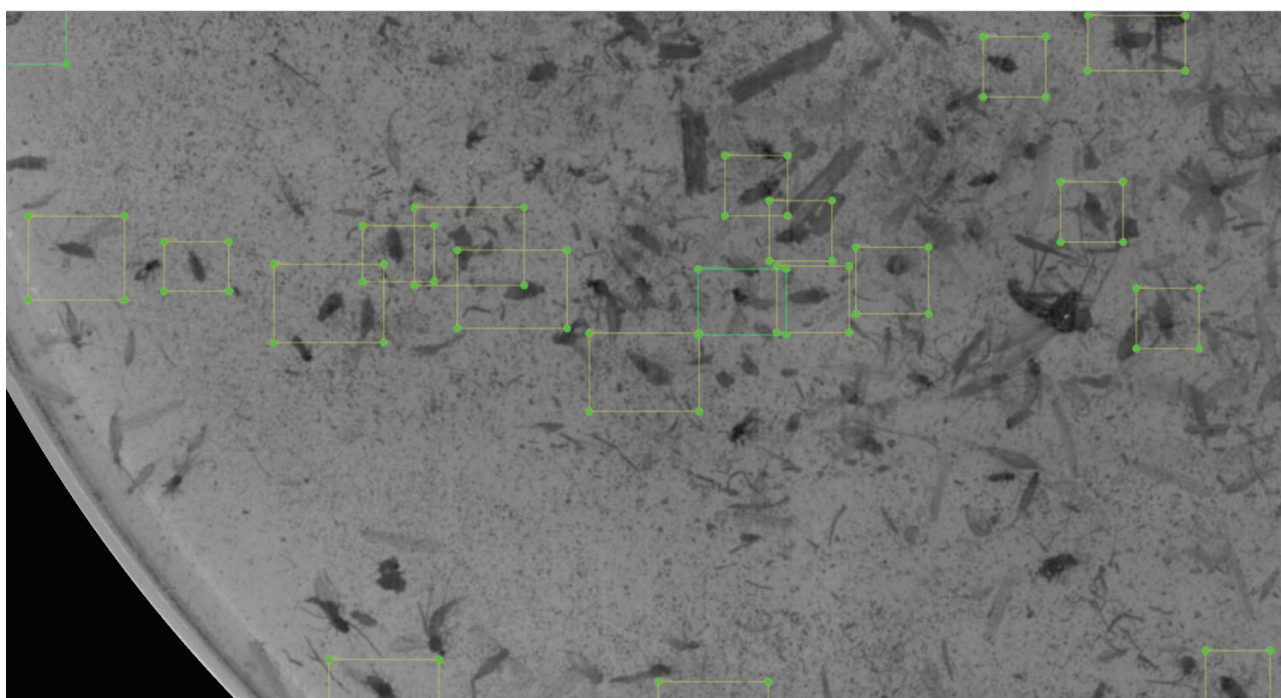


Figura 6. Imagem com afídeos e detritos.

processamento gráfico (GPU) GeForce GTX Titan X com 12 GB de VRAM do Laboratório de Realidade Virtual (LabRV) da UPF.

Através da implementação de Abdulla [82] é possível iniciar o treinamento de um modelo inteligente partindo de pesos aleatórios *starch*, que implicará na alteração dos pe-

tos de todas as camadas da rede, ou pelo reaproveitamento de pesos *transfer learning*, onde são utilizados pesos de um modelo externo. Nesse sentido, ao aplicar a técnica *transfer learning* é possível treinar apenas um subconjunto de camadas da rede, reduzindo o tempo de processamento e demandando menos recursos de hardware. Prevendo a utilização de imagens de entrada em escalas de cinza, foram ajustados os seguintes subconjuntos de camadas:

- **heads**: conv1.*, mrcnn_.*, rpn_.* e fpn_.*;
- **5+** : conv1.*, res5.*, bn5.*, mrcnn_.*, rpn_.* e fpn_.*,
- **4+** : conv1.*, res4.*, bn4.*, res5.*, bn5.*, mrcnn_.*, rpn_.* e fpn_.*;
- **3+** : conv1.*, res3.*, bn3.*, res4.*, bn4.*, res5.*, bn5.*, mrcnn_.*, rpn_.*, fpn_.*;
- **all** : ".*"

Em virtude da dimensão original das imagens de entrada 6156x6156 pixels, não foi possível processá-las por inteiro, com base nos recursos de hardware disponíveis. Sendo assim, optou-se pelo redimensionamento (modo *square*) das imagens para 1024x1024 e 2048x2048, bem como, a aplicação da técnica de recorte aleatório (modo *crop*) de amostras na dimensão de 2048x2048 pixels. Esses modos estão disponíveis na implementação de Abdulla [82], sendo ativados por parâmetros.

O extrator de características utilizado foi o ResNet50. O *Batch Size* foi fixado em 1. Foram aplicados três conjuntos de valores para o hiper-parâmetro *RPN_ANCHOR_SCALES*, de acordo com o tamanho da imagem. O *DETECTION_MAX_INSTANCES* foi definido em 500, considerando a possibilidade de imagens com centenas de objetos de interesse. O *MAX_GT_INSTANCES* foi definido em 100, em razão da limitação do hardware. Os demais parâmetros presentes na implementação de Abdulla [82] não foram alterados.

Inicialmente foram realizados cinco experimentos com o conjunto de 167 imagens. Cada experimento executou de 40 a 140 épocas (treinamento), onde cada época possui 623 passos. Em razão do número restrito de imagens, aplicou-se duas técnicas compensatórias para evitar o problema de *over-fitting*. A primeira foi inicializar todas as camadas da rede com os pesos do modelo MS COCO (*transfer learning*), exceto as camadas: *mrcnn_class_logits*, *mrcnn_bbox_fc*, *mrcnn_bbox*, *mrcnn_mask* e *conv1*, que foram inicializadas com pesos aleatórios. O subconjunto de camadas retreinadas foi o 4+. A segunda técnica foi a aplicação nos experimentos 2, 4 e 5 das transformações geométricas randômicas de rotação (-90, +90), inversão horizontal (0.5) e vertical (0.5), para triplicar a quantidade de imagens *data augmentation*.

O objetivo desses experimentos foi de avaliar a configuração que alcançasse a maior precisão média (mAP) com as imagens de teste. Eles estão detalhados na Tabela 5.

A coluna "Modo" indica a utilização da técnica de redimensionamento da imagem original (*square*) ou recorte aleatório (*crop*).

Tabela 5. Resultados gerados pelos experimentos iniciais.

Exp.	Modo	Conj. Imagens	Resolução	Aumento	Tempo	Prec. train.	Prec. teste
1	square	1	1024x1024	Não	1d 4h 47m	87,0%	49.8%
2	square	1	1024x1024	Sim	1d 15h 39m	71,1%	53.6%
3	square	1	2048x2048	Não	3d 19h 16m	84,5%	59.4%
4	square	1	2048x2048	Sim	16d 1h 15m	85,2%	60.4%
5	crop	1	2048x2048	Sim	15d 4h 15m	70,8%	68,3%

Conforme os dados apresentados na Tabela 5, o quinto experimento alcançou a maior precisão média (68,3%), após a execução de 40 épocas. Sendo assim, com base no melhor modelo desse experimento, realizou-se outros dois adicionais com o segundo conjunto de imagens, envolvendo o grupo de camadas **3+** e ajuste no hiper-parâmetro RPN_ANCHOR_SCALES, para tentar reduzir os valores de perda e incrementar a precisão na classificação e localização dos insetos de interesse. Os detalhes são apresentados na Tabela 6.

Os experimentos 6 e 7 executaram mais 60 épocas, totalizando 100 épocas. A partir da época 80 o hiper-parâmetro RPN_ANCHOR_SCALES foi incrementado de (2,4,8,16,32) para (4,8,16,32) e (8,16,32,64). Também foi definido o número máximo de instâncias por imagem através do hiper-parâmetro MAX_GT_INSTANCES em 466. Esse valor corresponde a maior quantidade de rotulações realizadas em uma única imagem do conjunto de treinamento. Para reduzir o custo computacional, optou-se pela diminuição do número de regiões de interesse no segundo estágio, de 200 para 100 no hiper-parâmetro TRAIN_ROIS_PER_IMAGE.

Tabela 6. Detalhes dos experimentos adicionais 6 e 7.

Exp.	Modo	Conj. Imagens	RPN_ANCHOR_SCALES	Resolução	Aumento	Tempo
6	crop	2	4,8,16,32,64	2048x2048	Sim	-
7	crop	2	8,16,32,64,128	2048x2048	Sim	35d 8h 40m 5s

Para a disponibilização do modelo de inteligência foram desenvolvidos três serviços web através do framework Django REST ⁴⁴ e a linguagem de programação Python para cadastro de novas imagens, exclusão e detecção de insetos.

Esses serviços foram integrados no site Trap System, com a utilização da biblioteca *libcurl* ⁴⁵ e a linguagem de programação PHP. O Trap System é uma plataforma utilizada pela Embrapa Trigo para integração, monitoramento, simulação e tomada de decisão no manejo de epidemias causadas por vírus transmitidos por insetos ⁴⁶.

O serviço de cadastro (Figura 7), realiza a conversão da imagem original colorida (RGB) para o formato em escalas de cinza (GrayScale) e a aplicação do filtro *GaussianBlur* para a eliminação de ruídos. A imagem original tem um formato retangular, com dimensão

⁴⁴Disponível em: <https://www.django-rest-framework.org/>

⁴⁵Disponível em: <https://curl.haxx.se/libcurl/>

⁴⁶Disponível em: <http://gpca.passofundo.ifsul.edu.br/traps/>

de 10200 pixels de largura e 14040 pixels de altura, e no centro dessa imagem está a placa de Petri, com um diâmetro de 6156 pixels. Portanto, foi necessário desenvolver uma rotina para identificar e recortar um círculo contendo a área da placa, através do algoritmo *Hough-Circles*. Em seguida é gerado uma nova imagem com dimensão de 6156 pixels de altura e 6156 pixels de largura e adicionado o recorte no centro. Dessa forma é gerada a imagem final, com resolução horizontal e vertical de 96 dpi e intensidade de 8 bits. Por fim, a imagem original e final, são armazenadas em um sistema de arquivos e as respectivas localizações gravadas no AgroDB.

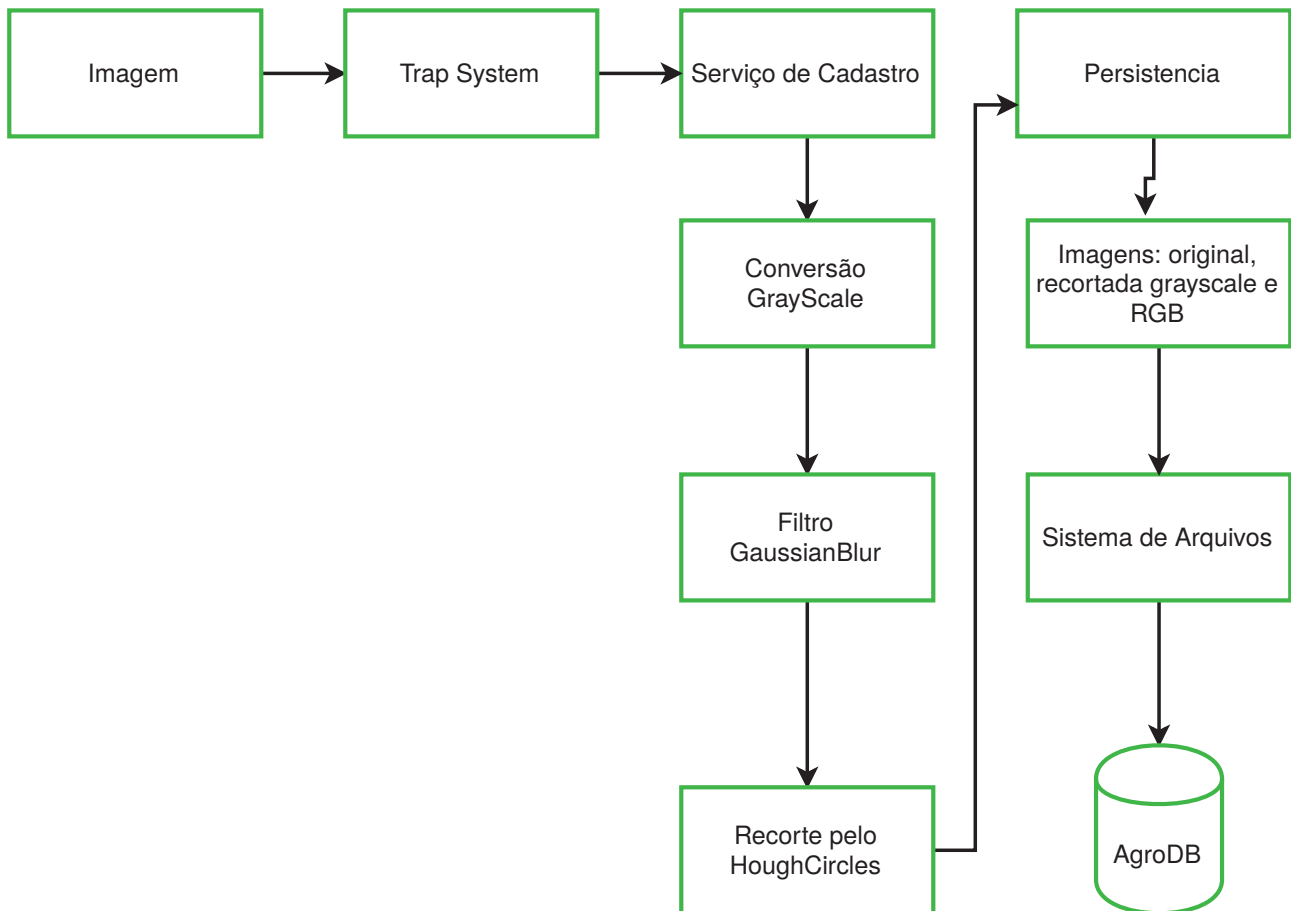


Figura 7. Serviço de Cadastro.

O serviço de detecção (Figura 8) recebe o código de uma determinada imagem via Trap System, realiza a inferência através do modelo de inteligência, aplica limiares por classe e gera uma nova imagem demonstrativa com a localização das *bounding-boxes* e a classificação dos objetos. Na sequência, retorna uma mensagem no formato *JSON* contendo a imagem em formato Base64 e a contagem dos insetos de interesse. Por fim, o Trap System armazena o resultado da detecção nas tabelas do AgroDB.

O serviço de exclusão remove uma determinada imagem. Esses serviços estão hospedados em um servidor no Instituto Federal Sul-rio-grandense (IFSul), Campus Passo Fundo. Esse servidor tem aproximadamente 14 GB de memória RAM e 8 vCPUs. A implementação no formato de serviços permitirá que outros sistemas possam acessar essas

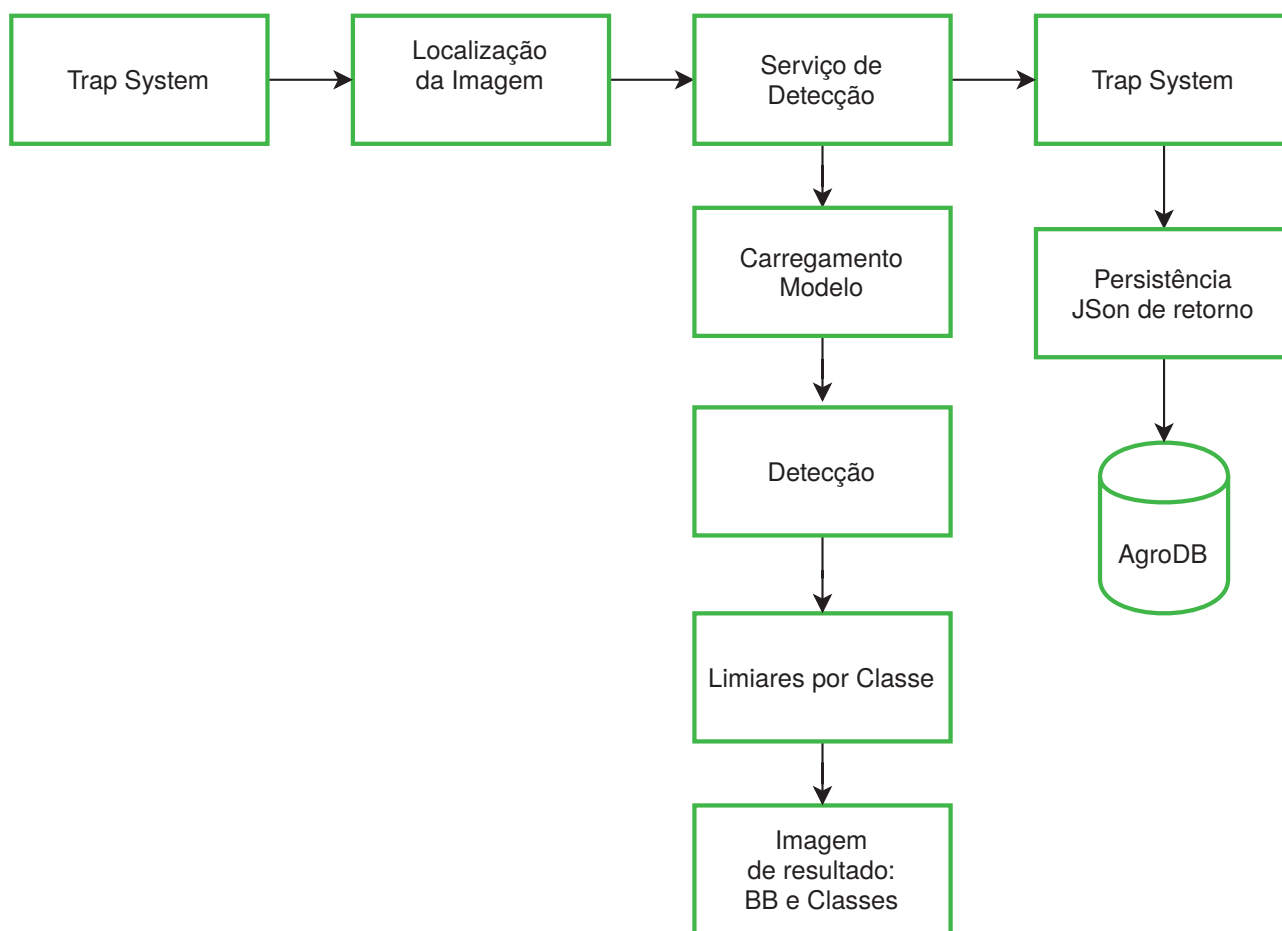


Figura 8. Serviço de Detecção.

funcionalidades. Além desses serviços, foi desenvolvida uma rotina para processar as imagens já inseridas no AgroDB.

O Modelo Entidade-Relacionamento (MER) apresentado na Figura 9 ilustra as quatro tabelas adicionadas no AgroDB: *detect_trap_image* e *insect_detect_trap_image* mantêm os dados gerados pelo serviço de detecção, e as tabelas *model_detection* e *detection_inference_params* guardam a referência para os modelos e os valores para os seus respectivos parâmetros. Por intermédio dessas tabelas, é possível avaliar a precisão de diferentes modelos e compará-los com o processo manual de identificação de insetos.

Para a validação do modelo foram utilizadas 580 imagens, contendo insetos coletados em armadilhas tipo Moericke instaladas em campos da Embrapa Trigo em Passo Fundo e Coxilha, RS, durante os períodos da safra de 2019 e 2020. Essas imagens foram organizadas em quinze séries: dez séries de imagens para analisar a identificação dos afídeos e cinco séries para validar o modelo na tarefa de identificação de parasitoides. Nessa região subtropical, em função das baixas temperaturas de inverno, as populações tendem a decrescer atingindo os níveis mais baixos ao longo do ano. A seguir, com o aumento da temperatura e fotoperíodo, a medida que se aproxima a primavera, as populações de afídeos alados capturados em armadilhas aumentam até o atingimento do pico.

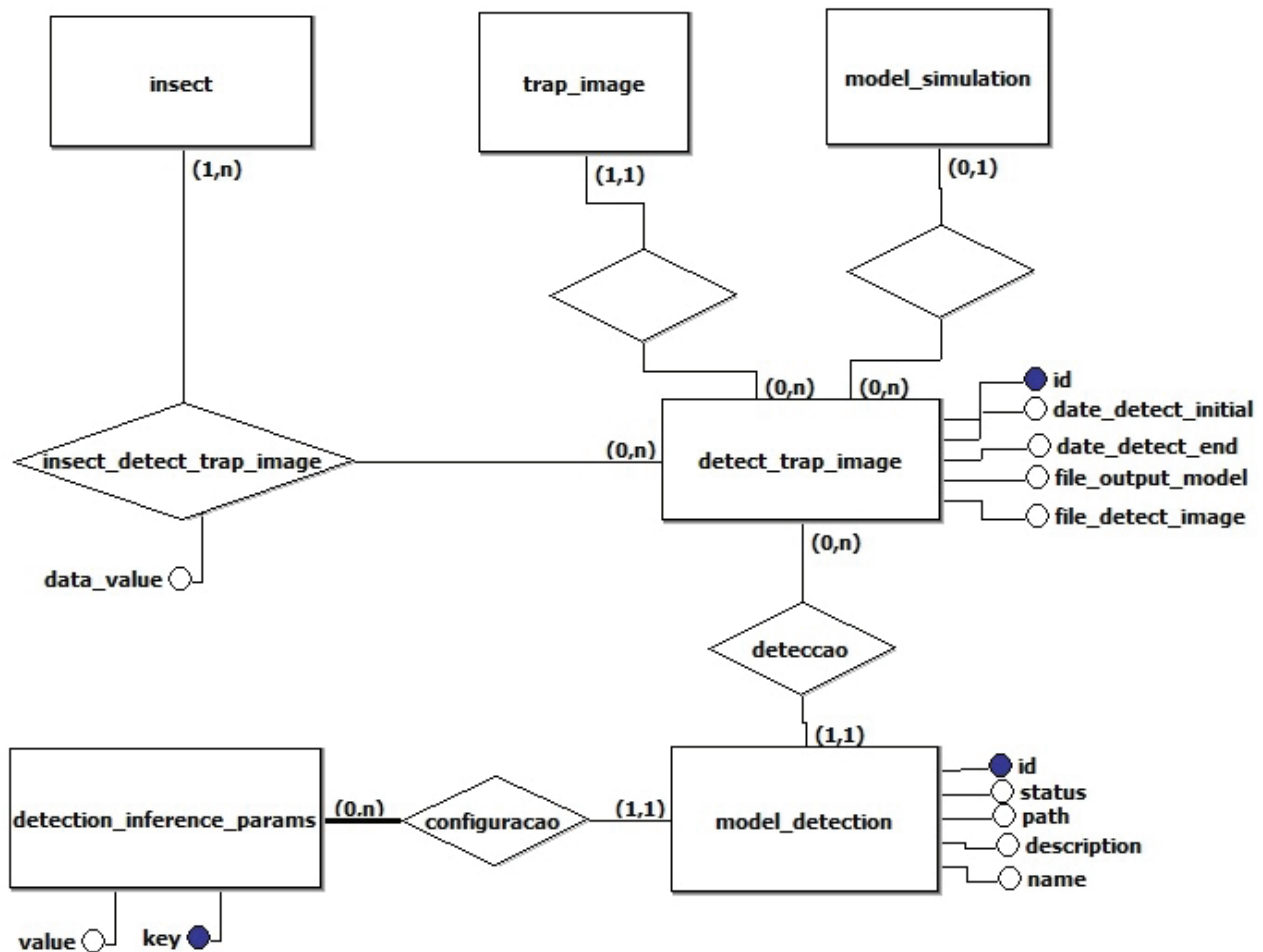


Figura 9. Tabelas adicionadas no AgroDB.

Com os dados obtidas pela inferências das 580 imagens através do serviço de detecção, foram aplicadas três abordagens para analisar o desempenho do modelo: coeficientes, *precision/recall/F1 score* e pontos de corte. Além disso, foram desenvolvidas customizações com a inclusão de hiper-parâmetros e variação dos limiares de confiança.

Para a geração dos coeficientes, foram comparados os somatórios semanais obtidos pela contagem manual, realizada pelo Especialista da Embrapa, com os dados gerados pelo InsectCV. Nesse sentido, adotou-se o coeficiente de determinação (R^2), que indica quanto o modelo é capaz de explicar os dados coletados, o coeficiente angular e de intercepto, os quais definem a relação linear entre duas variáveis, e podem ser usados para estimar uma taxa média de mudança. Sendo assim, interessa que o intercepto seja zero (0) e que o coeficiente angular seja um (1), indicando que o modelo não falha ao estimar o número de insetos de interesse na imagem.

Nesse sentido, foi desenvolvido uma página web no Trap System para a renderização de gráficos de linha e dispersão, através da biblioteca *Google Charts*⁴⁷, onde é correlacionado a contagem pela leitura manual com a do InsectCV, para um determinado

⁴⁷Disponível em: <https://developers.google.com/chart>

grupo de imagens. Nessa página, foram implementados filtros de período, espécie do inseto, armadilha(s) e modelo de detecção. Além dos gráficos, são impressos os coeficientes de determinação e angular, e distorção do intercepto, conforme exemplo ilustrado na Figura 21.

Sobre as métricas, considerou-se a *precision* para indicar o nível de acerto do modelo para os objetos classificados como parasitoides. Já o *recall* indica a frequência que o modelo encontra exemplos de determinada classe. Por fim, usou-se também o *F1 score*, razão entre *recall* e *precision*, que gera um percentual combinado. Em razão da complexidade para contabilizar manualmente falsos positivos e negativos nas imagens, essas métricas foram aplicadas somente para a identificação de parasitoides. Essa contagem foi realizada com a ajuda do Especialista da Embrapa.

Por fim, a avaliação da assertividade para detecção de níveis de manejo é um aspecto importante a ser analisado, conforme especialistas da Embrapa. Nesse sentido, foram avaliados diferentes pontos de corte, entre 10 e 50, para as contagens de afídeos alados via InsectCV. Esses pontos de corte podem representar o início ou o fim dos períodos de infestação, e são suficientes para servir como indicadores e embasar sistemas de alerta.

4. RESULTADOS E DISCUSSÕES

Este Capítulo detalha a escolha do modelo inteligente a partir dos experimentos 6 e 7 relatados no Capítulo 3, as customizações aplicadas e a validação para aferir o desempenho do modelo selecionado. Nas discussões são analisados os fatores que influenciaram os resultados obtidos.

4.1 RESULTADOS

Com base nos valores de perda geral gerados nos experimentos 6 e 7, observados com a ferramenta *TensorBoard*⁴⁸, verificou-se que o peso de perda `rpn_bbox_loss` recebeu valores elevados, significativamente maiores que os demais. Esses valores são apresentados na Figura 10 e na Figura 11. Sendo assim, após a 80^a época, o hiper-parâmetro `RPN_ANCHOR_SCALES` foi incrementado, o que resultou na redução do `rpn_bbox_loss` e, conseqüentemente, no valor de perda geral. Sendo assim, o experimento 7, que recebeu os valores (8,16,32,64,128) para o hiper-parâmetro supracitado, nas últimas vinte épocas de treinamento, alcançou o menor valor de perda geral, próximo a 1,4 conforme dados apresentados na Figura 12. Segundo Abdulla [82] o valor de referência para o valor de perda é 0,5.

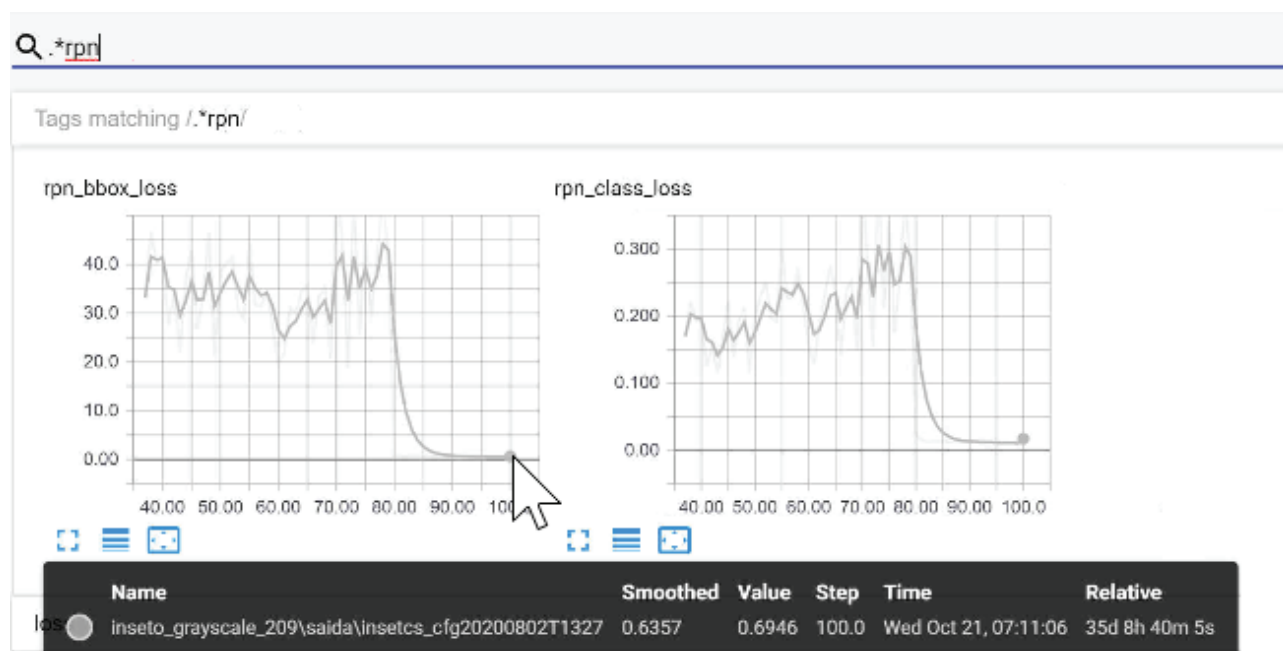


Figura 10. Valores atribuídos para o peso de perda `rpn_bbox_loss` e `rpn_class_loss`. Nestes gráficos, o eixo X representa a quantidade de épocas de treinamento e o Y representa o valor da perda.

⁴⁸Inclusa no TensorFlow

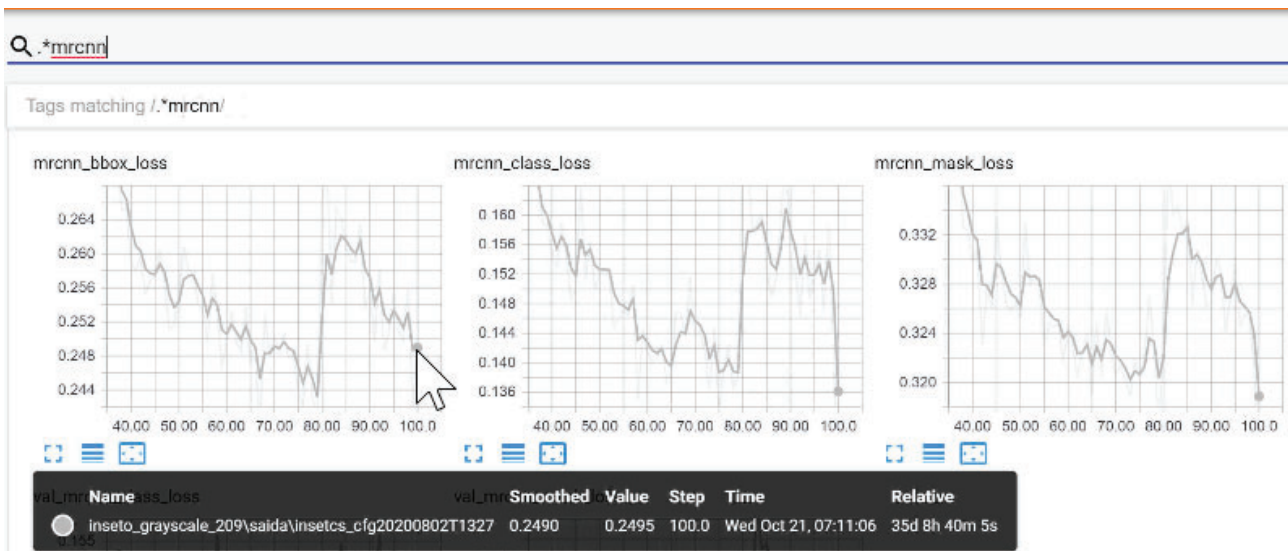


Figura 11. Valores atribuídos para os pesos de perda: mrcnn_bbox_loss, mrcnn_class_loss e mrcnn_mask_loss.

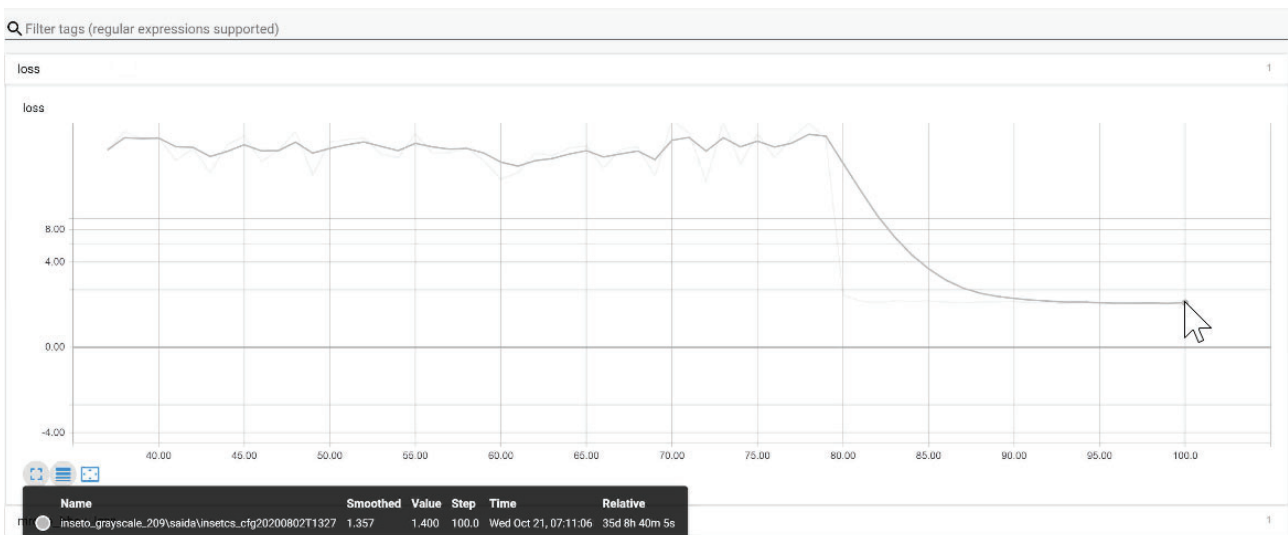


Figura 12. Valores de perda geral obtidos ao longo das cem épocas de treinamento.

Para a seleção do melhor modelo do experimento 7, não foi possível calcular a precisão média (mAP) através da rotina disponível na implementação de Abdulla [82] em função de limitação de hardware em termos de memória VRAM da GPU e espaço disponível em disco para a realização de *swap*. Portanto, foi realizado um procedimento com dez imagens contendo altas concentrações de afídeos, parasitoides e detritos. Essas imagens foram inferidas pelos últimos vinte modelos, e em seguida foi contabilizada manualmente a quantidade de falsos negativos e falsos positivos para cada modelo. Dessa forma, constatou-se que o modelo gerado na 97^a época de treinamento apresentou a maior precisão.

4.1.1 Customizações

Pelo fato das armadilhas estarem localizadas em ambientes externos, houve grande variação de detritos nas amostras. Nos períodos de pico da infestação de afídeos, identificou-se, principalmente nas armadilhas 1 de Coxilha, altas concentrações de insetos, ocupando todo o espaço disponível na placa. A Figura 13, a Figura 15 e a Figura 17 ilustram três situações identificadas nas imagens utilizadas para a validação.



Figura 13. Exemplo com distribuição esparsa de objetos de interesse, rotulada como "Imagem Fácil".

A Figura 13 é um recorte de uma imagem que representa uma situação de fácil detecção, pois a quantidade de afídeos é inferior a 50 espécimes, existem poucos casos de conexões e detritos dispersos. Nessa imagem existem apenas dois afídeos.

A Figura 14 ilustra o resultado do processo de detecção, onde os dois afídeos foram localizados e identificados corretamente, conforme indicação das *bounding boxes* vermelhas. Para a detecção de parasitoides, identificou-se dois fatores adicionais em relação aos afídeos, que reduziram a precisão do modelo. Conforme exemplo de detecção apresentado na Figura 19, os parasitoides são semelhantes a outros insetos, como moscas e tripes, e também podem apresentar variação no ângulo da cauda, unindo-a com o seu corpo. Nessa imagem, os elementos em destaque indicam casos de falsos negativos.

A Figura 15 representa um recorte de imagem, de média complexidade para detecção, com quantidade significativa de afídeos, ocorrência de conexões e a presença de detritos. A Figura 16 ilustra o resultado do processo de detecção, onde apenas os insetos com menos conexões foram identificados, totalizando 39 afídeos identificados corretamente.

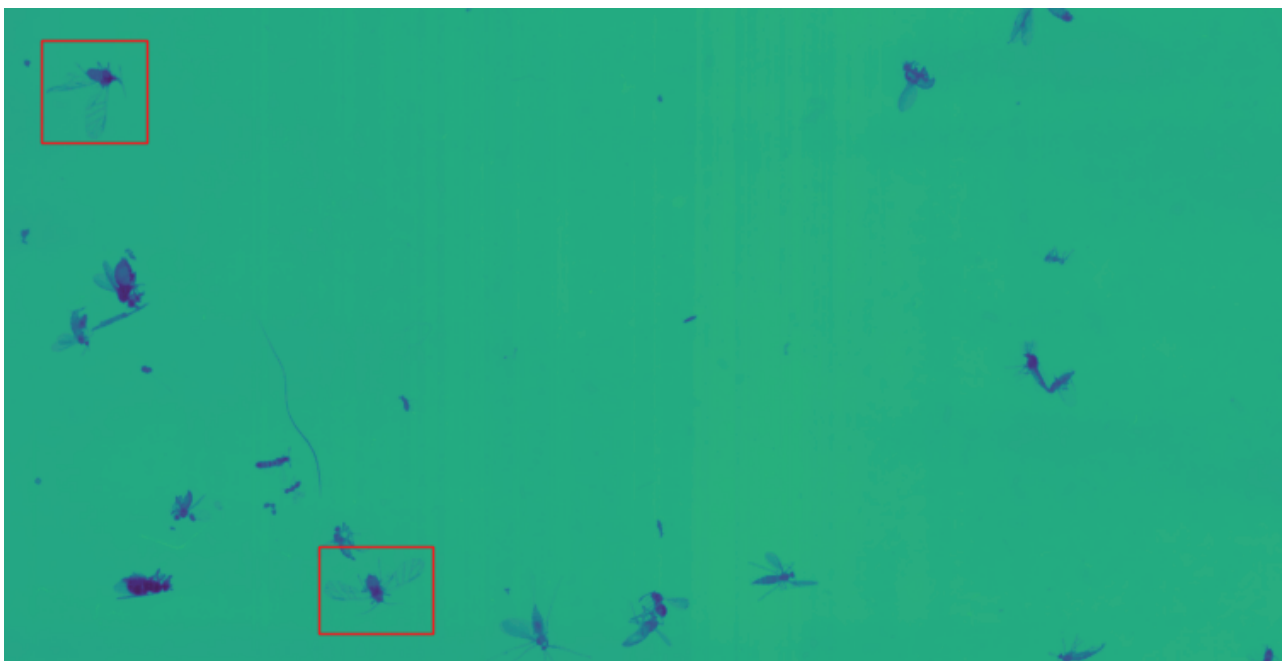


Figura 14. Resultado da detecção para a imagem fácil.



Figura 15. Exemplo de imagem com distribuição intermediária de objetos de interesse, rotulada como "imagem média".

A terceira situação é ilustrada na Figura 17, onde a grande quantidade de outros insetos e detritos dificultam o processo de detecção. A Figura 18 ilustra o resultado do processo de detecção, para a imagem difícil, onde apenas três afídeos foram localizados.

Destaca-se a possibilidade da existência de uma quarta situação, com uma quantidade significativa de insetos de interesse, detritos e outros insetos. No entanto, como essa situação foi identificada em algumas imagens na safra de 2020, entre as semanas 41 e 44, não será considerada nesse trabalho, podendo ser abordada em trabalhos futuros.

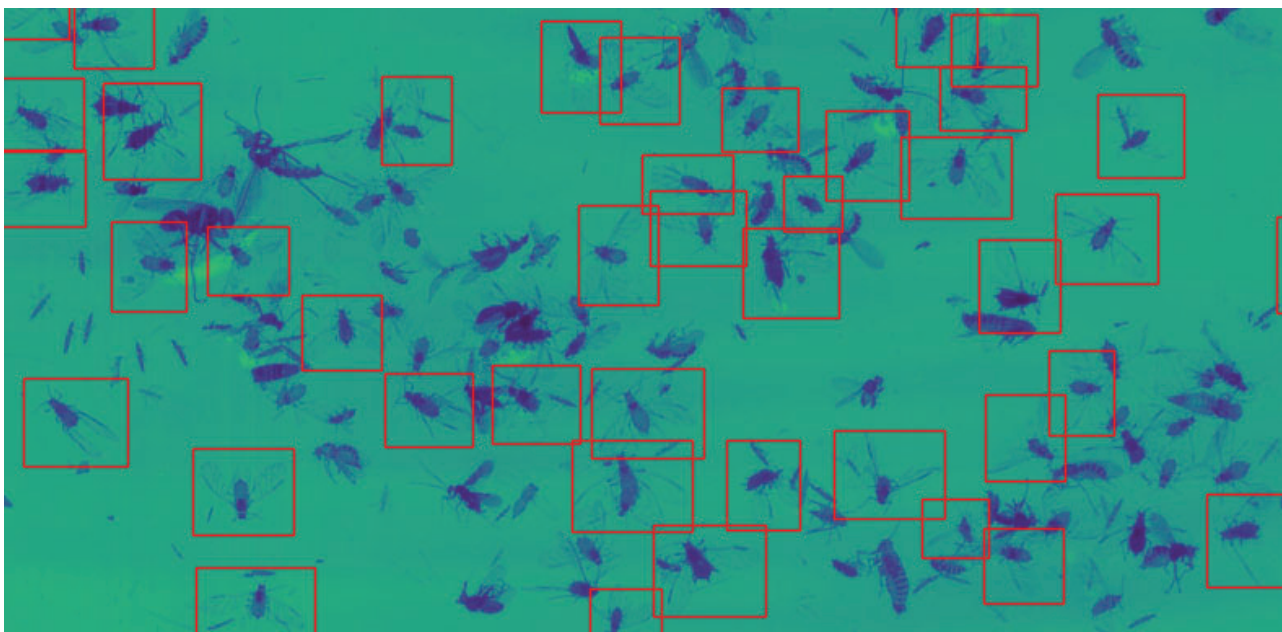


Figura 16. Resultado da detecção para a imagem média.

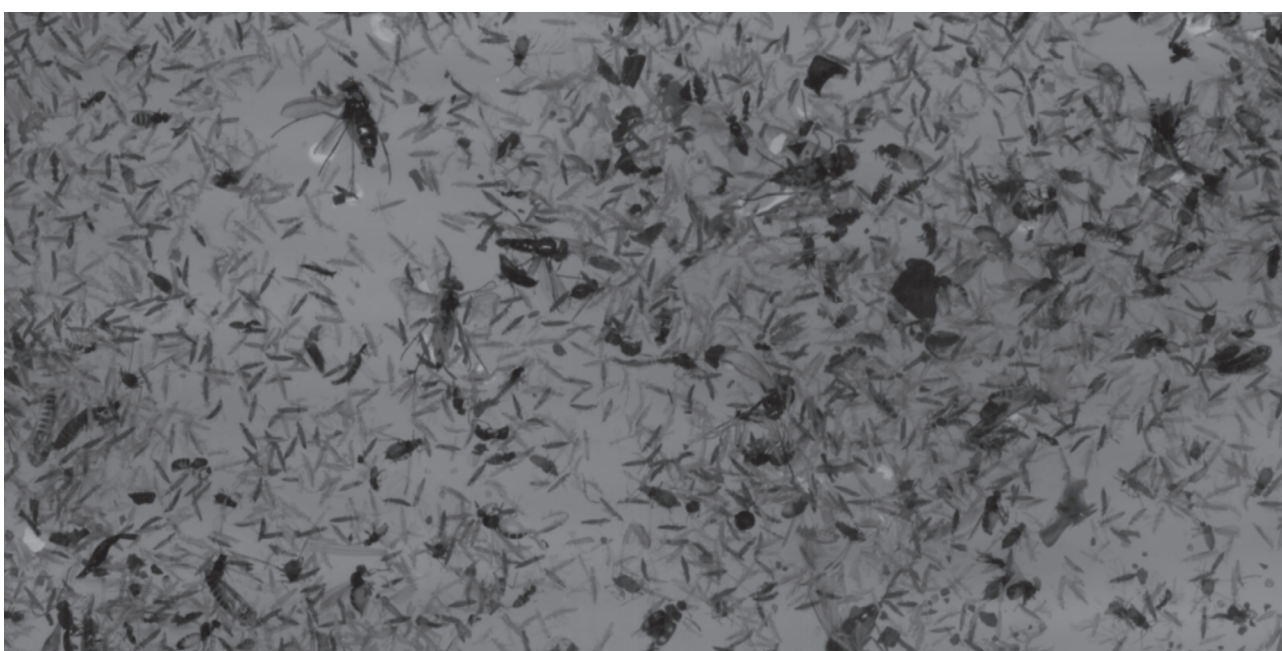


Figura 17. Exemplo com distribuição densa de objetos de interesse, rotulada como "imagem difícil".

Em razão das três situações identificadas nas imagens utilizadas para a validação, e da complexidade adicional na detecção de parasitoides, foram criados dois hiperparâmetros para definir o limiar de aceitação (confiança) por classe. Dessa forma, foi possível estabelecer diferentes valores para as classes de parasitoides e afídeos. Através de testes empíricos, realizados com apoio do especialista, chegou-se aos limiares de aceitação de 0,985 e 0,976 para as respectivas classes. Desse modo, para que um objeto seja considerado um parasitoide, o nível de confiança obtido pelo modelo deve ser maior do que

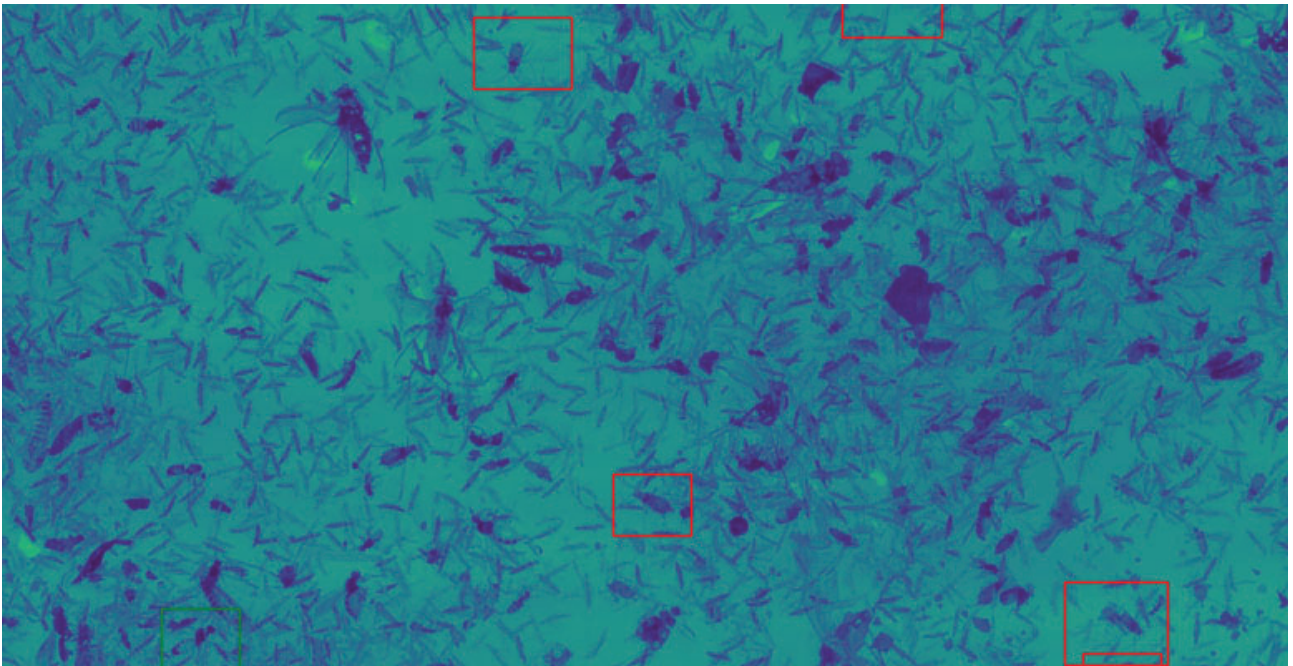


Figura 18. Resultado da detecção para a imagem difícil.

98,5%. Enquanto que para afídeos, 97,6%. Ressalta-se que limiares elevados ocasionam falsos negativos e a sua redução incrementa a probabilidade de falsos positivos.

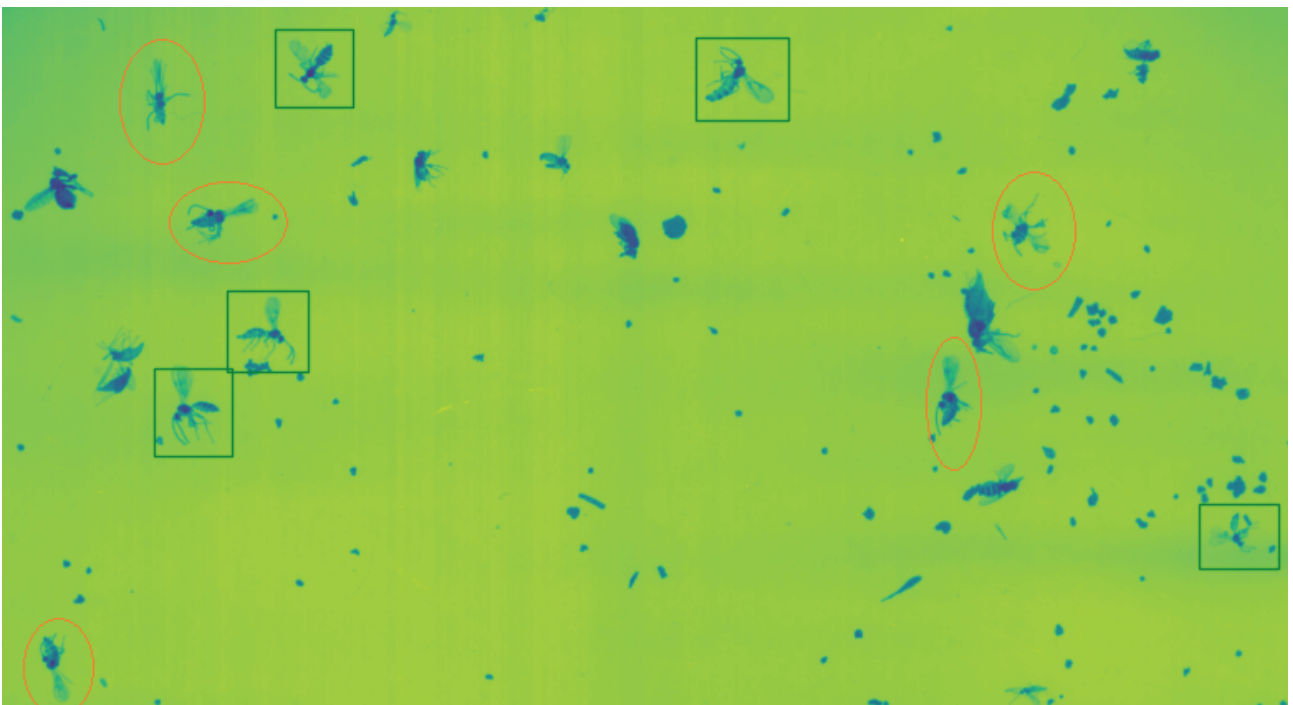


Figura 19. Detecção de parasitoides: rótulos em verdes indicam a localização e classificação de positivos. Elipses em laranja indicam falsos positivos.

4.1.2 Coeficientes

Nessa subseção são apresentados os coeficientes angulares, o intercepto e os coeficientes de determinação R^2 , gerados pela comparação com a identificação manual. O objetivo é estimar o nível do ajuste do modelo na detecção de afídeos e parasitoides.

Na Tabela 7 e na Tabela 8 são listadas dez séries de validação, referentes as safras de 2019 e 2020. Na Tabela 9, são listadas cinco séries, referente a safra de 2020 (01/07/2020 a 30/10/2020). Optou-se pela utilização de imagens de safras diferentes, pois em 2019 não foi identificada, pela leitura manual, quantidade significativa de parasitoides. Segundo a Embrapa Trigo, em 2019 houve menor proliferação de outros insetos em comparação com 2020, devido aos períodos de estiagem ocorridos em 2020.

Conforme os dados da coluna "Pico", na Tabela 7, que indica a quantidade máxima de insetos encontrados em uma determinada semana pela identificação manual, é possível verificar que, conforme o aumento do pico, a precisão do modelo reduz. Nesse sentido, para a armadilha 4 em 2019, o modelo alcançou o melhor ajuste, com um coeficiente angular de 0,67 e o coeficiente $R^2 = 0,99$. No entanto, na armadilha 3, onde foram registrados 630 afídeos no pico da infestação, o coeficiente angular foi de apenas 0,39 e o coeficiente $R^2 = 0,91$. Ao contabilizar as cinco séries, obteve-se coeficiente $R^2 = 0,98$, e o coeficiente angular de 0,50. Na segunda e terceira coluna, o valor X representa a identificação manual.

Nas cinco séries de 2020 de validação para afídeos, conforme dados apresentados na Tabela 8, observa-se redução nos coeficientes angulares e R^2 , indicando menor capacidade de detecção do modelo em relação a safra passada. Portanto, além do pico de infestação, detritos e outros insetos contribuíram para essa diminuição. O gráfico de linha apresentado na Figura 20 representa a relação entre a contagem manual (linha azul) e os dados gerados pelo InsectCV (linha vermelha) para afídeos. É possível constatar que o modelo identificou corretamente as elevações dos níveis populacionais nas duas safras. No gráfico de dispersão apresentado na Figura 21 para afídeos, considerando as dez séries, o coeficiente angular foi 0,39 e coeficiente $R^2 = 0,87$.

Tabela 7. Coeficientes para afídeos em 2019.

Armadilha	Coeficiente angular	Coeficiente angular (VL)	Intercepto	Intercepto (VL)	R^2	R^2 (VL)	Pico
4	0.67 * X	0.92 * X	+ 1.22	- 0.84	0.99	0.99	144
0	0.67 * X	0.93 * X	+ 3.00	- 1.34	0.99	0.99	271
2	0.61 * X	0.98 * X	+ 2.01	- 3.26	0.98	1.00	276
1	0.40 * X	0.81 * X	+ 10.77	+ 1.58	0.90	0.99	614
3	0.39 * X	0.75 * X	+ 10.71	+ 5.63	0.91	0.98	630
Todas	0.50 * X	0.86 * X	+ 25.61	- 0.76	0.98	1.0	1735

Conforme os dados apresentados na Tabela 9, a precisão na detecção de parasitoides não esteve relacionada ao pico de infestação. Para essa classe, ao contabilizar as cinco séries, o coeficiente angular foi de 0,45 e o coeficiente $R^2 = 0,92$, conforme gráfico de dispersão apresentado na Figura 23.

Tabela 8. Coeficientes para afídeos em 2020.

Armadilha	Coeficiente angular	Coeficiente angular (VL)	Intercepto	Intercepto (VL)	R^2	R^2 (VL)	Pico
0	0.54 * X	0.68 * X	+2.99	+2.13	0.91	0.83	79
1	0.25 * X	0.64 * X	+11.27	+6.17	0.65	0.95	324
3	0.26 * X	0.56 * X	+9.27	+6.27	0.77	0.82	484
4	0.28 * X	0.59 * X	+8.95	+5.12	0.77	0.92	604
2	0.30 * X	0.59 * X	+11.47	+4.03	0.87	0.91	686
Todas	0.30 * X	0.60 * X	+37.96	+22.11	0.85	0.92	1954

Tabela 9. Coeficientes para parasitoides em 2020.

Armadilha	Coeficiente angular	Coeficiente angular (VL)	Intercepto	Intercepto (VL)	R^2	R^2 (VL)	Pico
0	0.53 * X	0.73 * X	+ 0.12	+ 0.44	0.89	0.93	24
1	0.49 * X	0.63 * X	+ 0.64	+ 0.60	0.80	0.85	14
4	0.32 * X	0.67 * X	+ 0.67	- 0.23	0.64	0.92	29
3	0.41 * X	0.72 * X	+ 0.21	- 0.03	0.92	0.85	30
2	0.40 * X	0.77 * X	+ 0.13	-0.03	0.75	0.92	38
Todas	0.45 * X	0.72 * X	+ 1.03	+ 0.61	0.92	0.96	129

A partir desses resultados, identificou-se uma quantidade significativa de falsos negativos nas imagens médias, conforme exemplo na Figura 16. Com o intuito de reduzir essa característica, e oferecer outra opção para análise do especialista, foi implementado um novo parâmetro para que o usuário da aplicação pudesse indicar, em cada requisição ao serviço de detecção, o tipo da imagem: (i) imagem padrão; (ii) alta população de afídeos; (iii) alta população de parasitoides; e (iv) alta população de afídeos e parasitoides. Dessa forma, na opção (ii), o limiar é reduzido para 75%. Na opção (iii), o limiar é definido em 70%. E na opção (iv), ambos limiares são reduzidos. Com esse novo recurso, as quinze séries de validação foram reprocessadas, aplicando os limiares de 75% e 70% nas imagens das semanas de pico de infestação de afídeos ou parasitoides, respectivamente.

Os coeficientes gerados após o reprocessamento das séries também estão listados na Tabela 7, na Tabela 8 e Tabela 9, nas colunas com o sufixo (VL), representando a variação de limiar. Nota-se valores mais próximos de 1 para o coeficiente angular, o que representa um ajuste mais preciso para detectar variações em altas populações de afídeos.

O gráfico de linha apresentado na Figura 22 mostra a relação entre a contagem manual (linha azul) e os dados gerados pelo InsectCV (linha vermelha) para parasitoides. É possível constatar que o modelo identificou corretamente as elevações dos níveis populacionais ocorridas entre as semanas 28 e 34, o declínio verificado entre a semana 35 e 38, bem como, a pequena elevação na semana 41. No entanto, em razão da quantidade de parasitoides encontradas na safra 2020 nas armadilhas de Coxilha, a análise do ajuste do modelo foi prejudicada.

4.1.3 Métricas

Os resultados da aplicação das métricas *precision*, *recall* e *F1 score* são apresentados na Tabela 10. Ao analisar os percentuais obtidos, pode-se verificar que, para objetos classificados como parasitoide, o modelo acertou em 85% dos casos (coluna Precision).

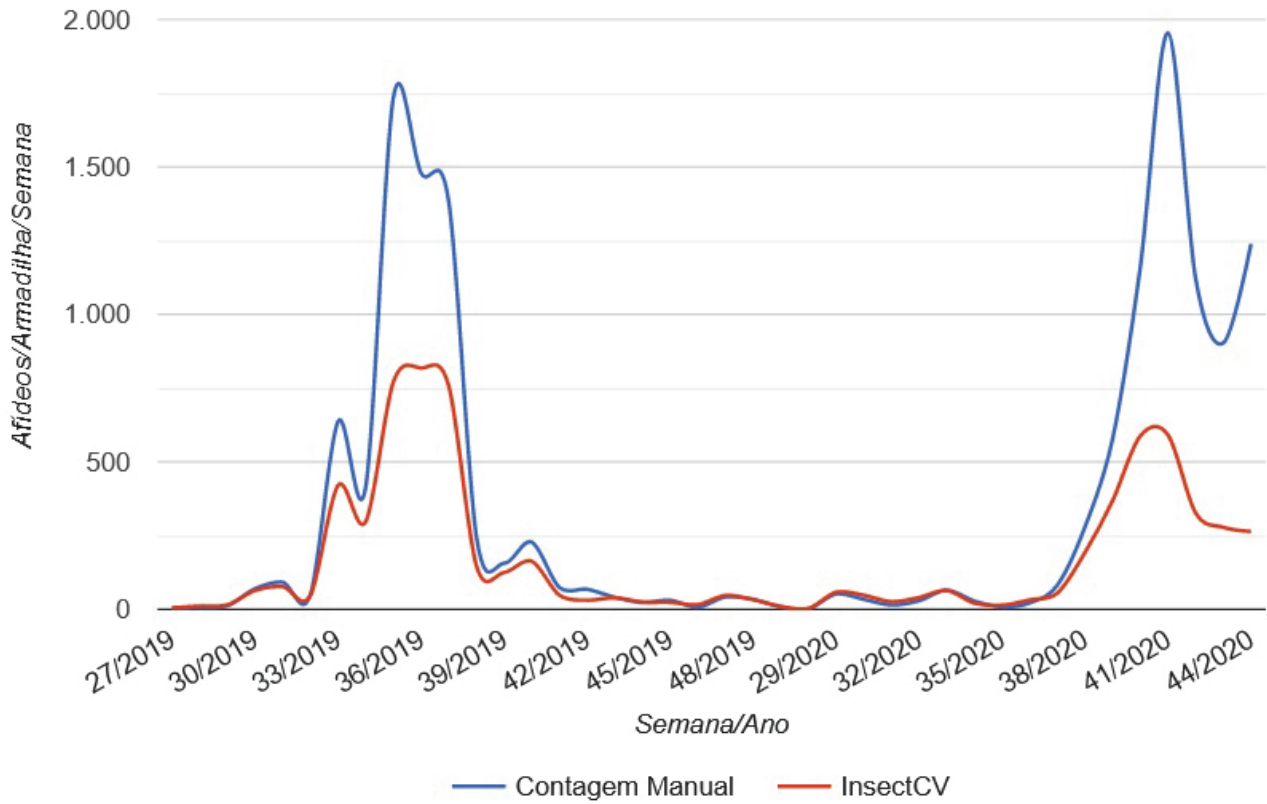


Figura 20. Ajuste para afídeos, safra 2019 e 2020. N=200

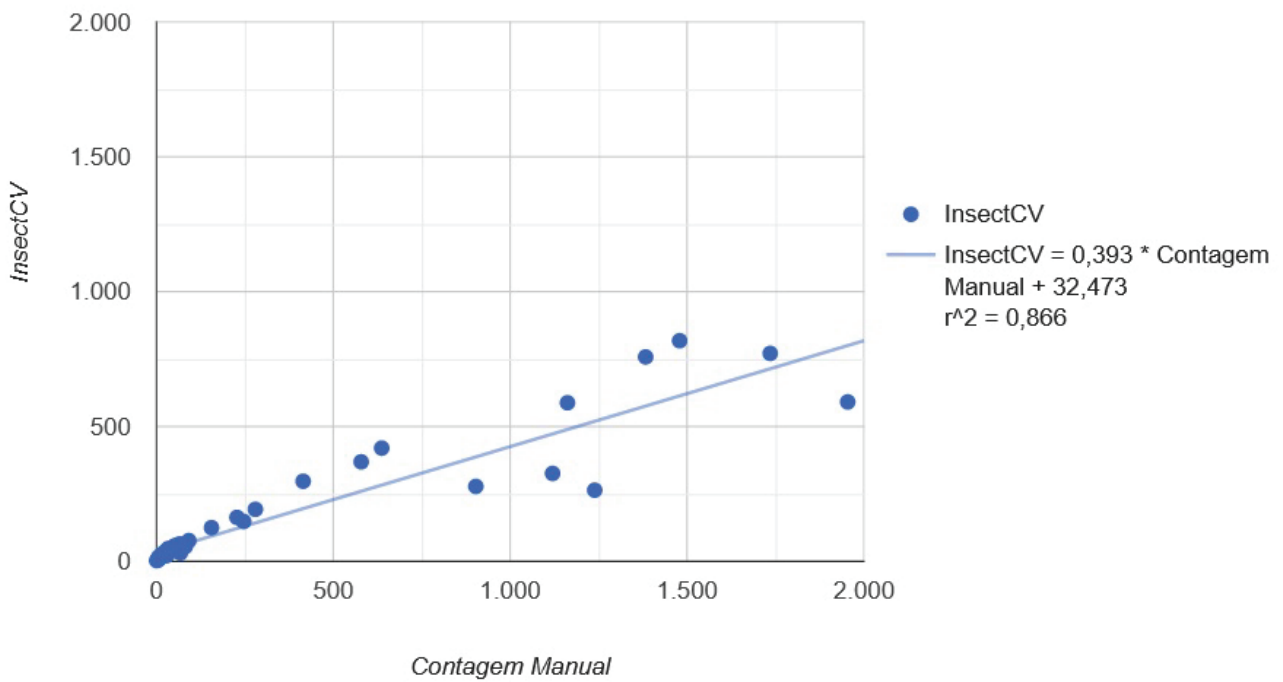


Figura 21. Relação entre a contagem manual de afídeos e o detectado pelo InsectCV.

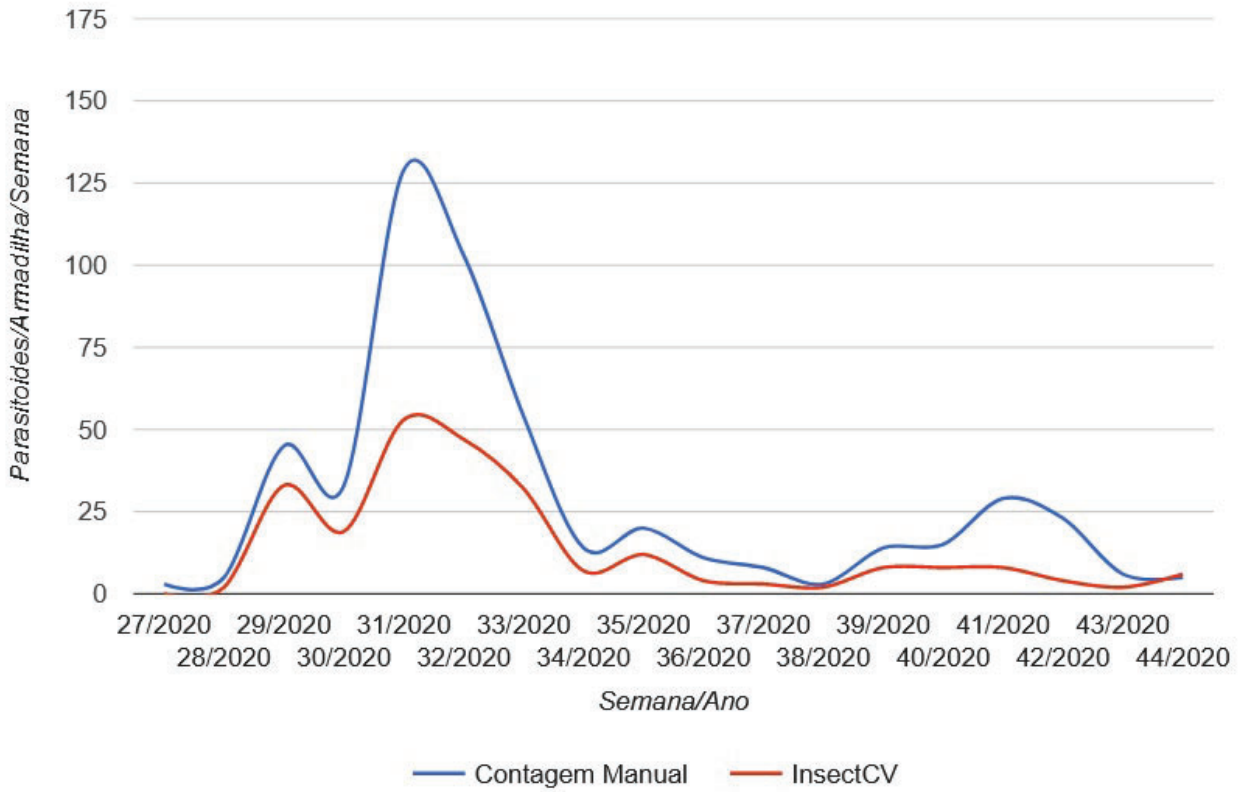


Figura 22. Ajuste para parasitoides, safra 2020. N=90

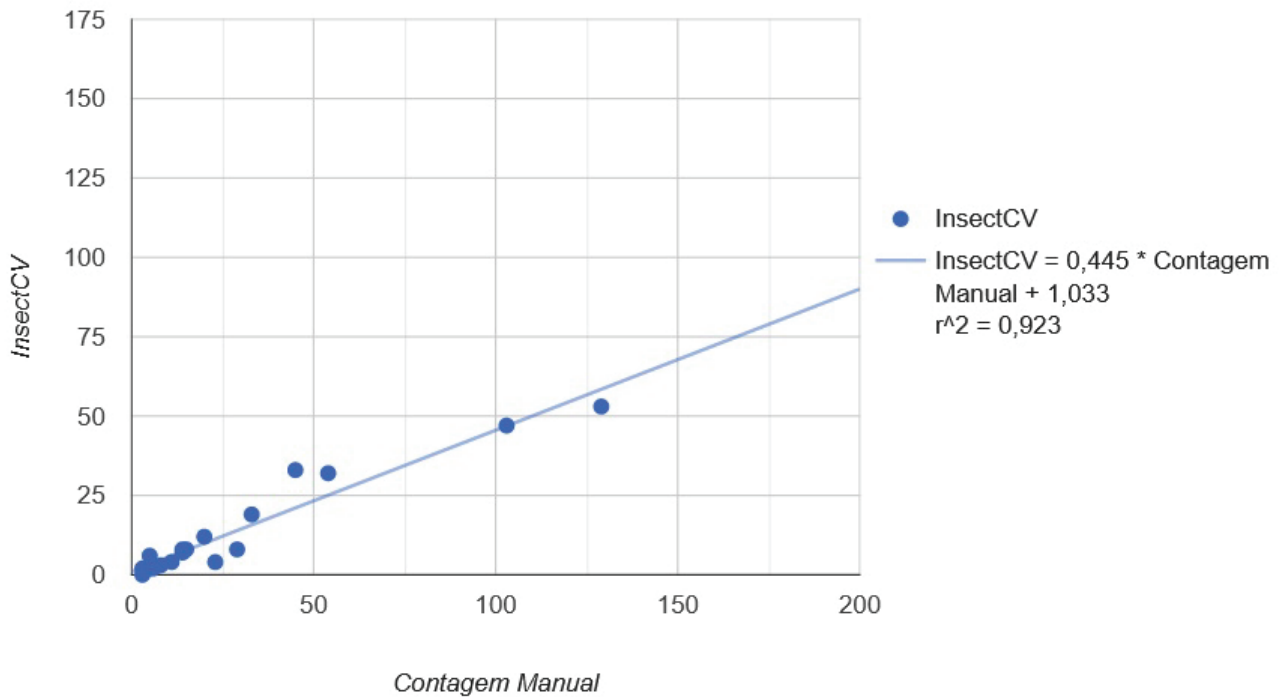


Figura 23. Relação entre a contagem manual de parasitoides e o detectado pelo InsectCV.

Quando um inseto é realmente um parasitoide, a frequência em que o modelo classificou corretamente foi de 41%. Nesse sentido, a estratégia de utilizar limiares altos possibilitou a diminuição de falsos positivos. Porém, 60% dos parasitoides não foram identificados. Ao combinar as métricas de *precision* e *recall*, o valor obtido para o *F1 score* foi de 0,55 enquanto que um modelo perfeito chegaria a um score de no máximo 1.

Tabela 10. Métricas para parasitoides.

Armadilha	Verdadeiros Positivos (TP)	Falsos Positivos (FP)	Falsos Negativos (FN)	Precision	Recall	F1 score
1	32	9	29	0,78	0,52	0,63
2	59	8	103	0,88	0,36	0,52
3	46	6	63	0,88	0,42	0,57
4	36	7	57	0,84	0,39	0,53
Todas	173	30	252	0,85	0,41	0,55

4.1.4 Pontos de Corte

Como uma das principais aplicações práticas da contagem de afídeos em armadilhas é na automação de sistemas de alerta, a avaliação da assertividade da detecção do modelo em diferentes níveis é relevante. O principal ponto de corte é de 20 pulgões por armadilha, o que representa cerca de 10% de plantas com infestação de pulgões⁴⁹. Nesse sentido, também foram considerados níveis de variação de 10 a 50 pulgões, uma vez que a população de insetos em armadilhas varia em função de vários fatores. Entre eles, os principais são a localização e a paisagem na qual a armadilha está localizada.

Em termos de dinâmica populacional de afídeos na safra de 2019, nas cinco séries de validação, conforme as linhas azul e laranja na Figura 24, o pico foi atingido na semana 35 (S35), sendo este de 347 afídeos/armadilha/semana para a contagem manual e de 154 afídeos/armadilha/semana para o InsectCV. Considerando limiares intermediários para a tomada de decisão do manejo até o atingimento do pico, o limiar de 10 afídeos/armadilha/semana foi ultrapassado na S30 para contagem manual (14 afídeos) e da mesma forma para o InsectCV (12,8 afídeos). A partir desse ponto, o crescimento da captura de afídeos aumentou rapidamente e o limiar de 20 afídeos afídeo/armadilha/semana foi atingido na S33 (127 afídeos na contagem manual e 84 para o InsectCV) superando, portanto, também o limiar de 50 afídeos.

Na safra de trigo de 2020 (semanas 27 a 44), considerando o conjunto das 5 séries em análise, conforme as linhas amarela e cinza na Figura 24, o pico foi detectado na S41 por ambos os métodos, sendo este de 391 afídeos/armadilha/semana para a contagem manual e de 136 afídeos/armadilha/semana para o InsectCV. Considerando limiares intermediários para a tomada de decisão do manejo até o atingimento do pico, o limiar de 10 afídeos/armadilha/semana foi ultrapassado na S29 para contagem manual (10 afídeos)

⁴⁹Nível utilizado pela Embrapa Trigo para as culturas de trigo

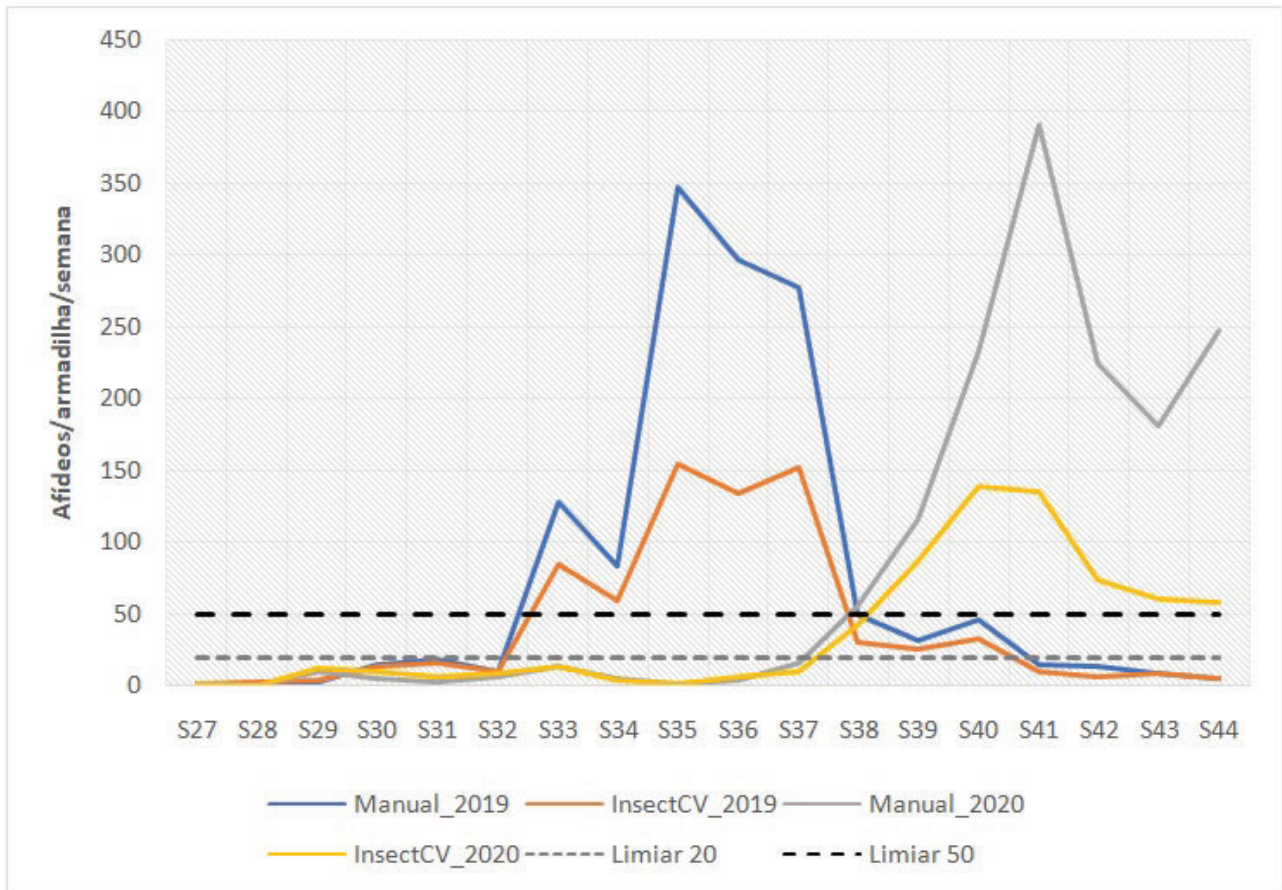


Figura 24. Curva de progresso de populações de afídeos. Dados de duas safras de trigo (2019 e 2020). Dados médios de 5 armadilhas. As linhas pontilhadas representam os limiares de 20 e 50.

e da mesma forma para o InsectCV (13 afídeos). A partir desse ponto, houve queda na captura de alados, com alcance novamente desse patamar na S33 (13 afídeos na contagem manual e 14 por InsectCV). Novamente, houve queda e então a curva retomou com crescimento sustentado com limiar de 10 afídeos na S37 (16,4 afídeos na contagem manual e 11 por InsectCV), e limiar de 20 afídeos na S38 (manual: 56 afídeos; InsectCV: 43 afídeos). Apenas para o limiar de 50 afídeos por armadilha houve um atraso de uma semana entre os métodos de detecção e contagem (S38 para manual e S39 para InsectCV).

Portanto, verificou-se que os desvios entre a leitura manual e InsectCV aumentam com a elevação da densidade populacional. Apesar desses desvios, a forma da curva de captura de afídeos alados ao longo do tempo foi a mesma em duas safras de trigo analisadas. Considerando o baixo desvio nas fases iniciais da curva de crescimento populacional e a manutenção da forma dessa curva, o uso do InsectCV permitiu detectar os pontos críticos de tomada de decisão para manejo e o pico da epidemia com a mesma precisão da contagem manual, ainda que nas semanas de elevada contagem de insetos o programa tenha subestimado a população.

4.2 DISCUSSÕES

Com base nos resultados obtidos na validação do modelo, é possível afirmar que através de um pequeno conjunto de imagens (209), quantidade significativa de objetos rotulados (17908 objetos) e a aplicação de customizações, foi possível gerar um modelo através da algoritmo Mask R-CNN capaz de identificar variações nos níveis populacionais de afídeos e parasitoides, com base nos padrões de imagens encontrados. Para parasitoides, a precisão do modelo foi levemente menor devido a variação da pose e posicionamento da cauda em relação ao restante do corpo, conforme recorte de imagem apresentado na figura 19. Destaca-se também que em função da baixa quantidade de parasitoides nas imagens de validação, referentes a safra de 2020, não foi possível realizar uma estimativa aprofundada através dos coeficientes, pois, os picos de infestação foram menores. No entanto, os valores encontrados para as métricas de *precision* e *recall* indicam a necessidade de equilibrar essa relação, com ajustes dos limiares.

Em termos da capacidade de detecção de insetos conectados, exemplificado na Figura 16, é possível constatar que o modelo foi capaz de detectar afídeos conectados. No entanto, em casos de densas aglomerações, envolvendo vários insetos, o modelo não conseguiu reconhecer a maioria dos insetos dessas regiões. De modo geral, considerando os três tipos de imagens utilizados para a validação, o modelo obteve precisão satisfatória, com base nos coeficientes apresentados, sobretudo na safra de 2019.

Em comparação com os trabalhos listados na Tabela 2, a implementação do InsectCV tem semelhança com os trabalhos: [59], [66] e [56], que aplicaram Faster R-CNN para o reconhecimento de insetos. No entanto, nesses estudos não houve a presença significativa de detritos e/ou insetos semelhantes, o que reduz a precisão do modelo. Em termos de conjunto de dados para treinamento, o número de imagens utilizadas no InsectCV foi menor em relação a esses três trabalhos. A dimensão das imagens para treinamento e inferência foi maior em relação aos trabalhos listados nessa Tabela.

A redução da precisão na contagem de afídeos em densas aglomerações pode ser amenizada com adoção de procedimentos de manejo como: (i) divisão da amostra em duas ou mais imagens; (ii) realocação ou elevação das armadilhas e; (iii) utilização de uma placa de Petri maior. Essas alterações podem contribuir para a redução de conexões e a quantidade de detritos, elevando a precisão do modelo.

Em função dos recursos de hardware disponíveis para a realização dos experimentos (Intel Core I7-6950X, 32 GB de RAM e GPU GeForce GTX Titan X com 12 GB de VRAM) e do tamanho da imagem (6156x6156), não foi possível ampliar a quantidade de camadas treinadas, devido a necessidade de mais memória VRAM.

O processo de inferência de imagens para validação do modelo também foi afetado pela falta memória RAM e de GPU no servidor do IFSul (Intel Xeon E31240, 8 vCPUs, 14 GB de RAM). Nesse caso, foi necessário redimensionar a resolução da imagem para

5568 x 5568 pixels, o que pode ter reduzido a quantidade de informações analisadas pelo modelo. O tempo necessário para realizar a inferência de cada imagem, através do serviço de detecção, variou de 3 a 6 minutos.

O tempo para execução de cada época de treinamento foi de aproximadamente 9 horas, totalizando 35 dias para a conclusão do experimento 7. Essa demora dificultou a realização de experimentos adicionais. Em termos de alternativas para a redução do tempo de treinamento, destaca-se duas possibilidades: a primeira é utilizar a estratégia indicada por Bobba [83], que consiste na utilização de imagens menores (1024x1024) para a atualização mais rápida dos pesos e aplicar imagens maiores (6156x6156) na fase final para ajustar os parâmetros do modelo final. No entanto, isso requer o recorte das imagens e o reposicionamento dos rótulos.

A segunda alternativa é relacionada ao fato da máscara gerada pelo Mask R-CNN não ser utilizada para a classificação e contagem. Nesse contexto, a aplicação de abordagens convolucionais de um estágio, como o YOLO e SSD, podem ser mais eficientes na utilização de memória, em função do menor número de camadas. Conseqüentemente, necessitaria de menos tempo para a execução dos experimentos.

Através da utilização do recurso que permite adequar os limiares, com base no tipo da imagem, o coeficiente angular (VL), Intercepto (VL) e coeficiente de determinação R^2 (VL) indicaram aumento significativo na precisão do modelo para afídeos e parasitoides em situações de alta densidade populacional. Nesse sentido, é possível aprimorar esse recurso, com a utilização de uma imagem colorida para representar o resultado da detecção, de forma que o usuário tenha mais informações para determinar o limiar adequado.

Portanto, considerando os coeficientes destacados em **negrito** na Tabela 7 e na Tabela 8, e examinando a análise de pontos de corte, nota-se que o modelo identificou corretamente as variações dos níveis populacionais, com base nas 580 imagens utilizadas para a validação. No entanto, em épocas de maior proliferação de insetos e baixos níveis de precipitação pluviométrica, a precisão do modelo tende a ser menor em função da presença de outros insetos e conexões entre os afídeos. Nesse sentido, ressalta-se novamente a necessidade de avaliar os procedimentos de manejo.

5. CONCLUSÃO

Neste trabalho foi apresentado um sistema para a identificação de afídeos e parasitoides em imagens digitais, geradas pela digitalização das amostras capturadas pelas armadilhas, nas estações experimentais da Embrapa Trigo. Para o treinamento do modelo inteligente utilizou-se uma rede neural convolucional profunda, de dois estágios, chamada Mask R-CNN e um conjunto de dados com 209 imagens em escala de cinza, contendo 17908 objetos rotulados.

Em termos de customizações do modelo, foram criados dois hiper-parâmetros para definir os limiares de aceitação por classe, devido a complexidade adicional no reconhecimento de parasitoides. Sete experimentos foram realizados, com variações no tamanho da imagem, transferência de aprendizado e parametrizações. Após cem épocas de treinamento, o valor de perda foi reduzido para 1,4. Além do modelo, foram desenvolvidos três serviços web, no padrão REST, para o cadastro, remoção e detecção de insetos, integrados a plataforma Trap System. No serviço de cadastro foi desenvolvido uma rotina para o pré-processamento das imagens, através da biblioteca OpenCV. Também foi criado uma rotina para processar as imagens já inseridas no AgroDB. Com a implementação no formato de serviços, será possível que outros sistemas acessem as funcionalidades desenvolvidas nesse sistema.

Para a validação do modelo, foram utilizadas 580 imagens referentes aos períodos de safras de trigo de 2019 e 2020 das estações de Passo Fundo e Coxilha. Em comparação com a contagem manual, os coeficientes de determinação obtidos ($R^2 = 0,87$ para afídeos e $R^2 = 0,92$ para parasitoides), bem como, a análise de pontos de corte, comprovaram a capacidade do modelo em identificar a flutuação dos níveis populacionais para esses insetos.

Nesse sentido, a principal contribuição desse trabalho é relacionada a aplicabilidade do InsectCV, que poder ser utilizá-lo como fonte de dados para embasar a tomada de decisão no manejo integrado de doenças, que geralmente consideram curvas ascendentes entre 20 a 50 afídeos. A utilização desse sistema poderá reduzir o tempo de identificação, tornando o processo escalável.

Com esses resultados, foi possível chegar a uma resposta afirmativa para a pergunta de pesquisa: sim, o modelo inteligente, baseado em Mask R-CNN, treinado com um pequeno conjunto de imagens e transferência de aprendizado, quantidade significativa de objetos rotulados, e customizações foi capaz de reconhecer insetos com precisão suficiente para embasar sistemas de alerta.

Com relação as alternativas para aumentar a precisão do modelo, conclui-se que é necessário adicionar novas imagens no conjunto de imagens para treinamento, contendo

parasitoides em diferentes poses e posicionamento de cauda, bem como, executar mais épocas para a redução do valor de perda. Em termos de triagem das armadilhas e geração das imagens, é necessário revisar os procedimentos de campo para reduzir a geração de imagens muito densas, que representam um ambiente altamente confuso para análise. Por fim, com a disponibilidade de uma GPU para a realização da inferência será possível processar imagens sem a necessidade de redimensioná-las.

Em relação ao aprimoramento do sistema, é necessário ajustar com maior precisão o limiar de aceitação para parasitoides e afídeos, que através de testes empíricos, realizados com apoio do Especialista foi definido em 0,985 e 0,976, respectivamente. Nesse sentido, pretende-se estudar uma alternativa para automatizar a geração das contagens de falsos positivos e falsos negativos, facilitando a geração das métricas. Dessa forma será possível avaliar diferentes valores de limiares através de uma curva precision-recall ou de uma curva ROC [85]. Além disso, seria relevante avaliar a utilização de imagens coloridas, caso seja possível incrementar a capacidade de processamento gráfico com novas GPU's. Além disso, considera-se importante testar outros algoritmos, para avaliar a capacidade de reconhecimento de insetos conectados e/ou sobrepostos do Mask R-CNN através da métrica de Intersecção sobre União (IOU).

Em termos de trabalhos futuros para o desenvolvimento de novas funcionalidades, considera-se a possibilidade de contabilizar a quantidade de material presente na amostra, através do somatório da área dos objetos na imagem. Esse cálculo poderia ser realizado com os dados da máscara, que não foi explorado nesse trabalho, permitindo mensurar os níveis de atividade biológica no entorno da armadilha. Dessa forma, também seria possível correlacionar a redução da precisão da identificação de insetos de interesse com o aumento da quantidade de resíduos e outros insetos.

Por fim, sobre a disseminação do conhecimento, destaca-se que esse trabalho gerou duas importantes publicações: a primeira, relacionada a fundamentação teórica, através do artigo de revisão *Automatic identification of insects from digital images: A survey* publicado no periódico *Computers and Electronics in Agriculture* [28] em 2020.

A segunda, foi a apresentação do artigo *Uma Implementação Baseada em Mask R-CNN para Detecção de Insetos em Imagens Digitais* no evento *SIBGRAPI 2020 – 33rd Conference on Graphics, Patterns and Images* na trilha *Work in Progress* [86], que detalhou os resultados gerados pelos cinco primeiros experimentos.

Além dessas duas publicações, destaca-se a aprovação do projeto de pesquisa: *Estudo sobre Abordagens Convolucionais para Desenvolver uma Ferramenta Computacional que Reconheça Insetos em Imagens Digitais* em edital de fomento à pesquisa do IFSul. Esse projeto tem como objetivo principal aprofundar os estudos sobre abordagens convolucionais de um estágio, visando a redução do tempo de treinamento e a demanda por recursos de hardware.

Ressalta-se que esse estudo de caso pode ser utilizado como referência e/ou base de comparação para futuros trabalhos de reconhecimento de insetos ou objetos semelhantes em imagens digitais, sobretudo na utilização de técnicas para identificação de pequenos objetos conectados ou sobrepostos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WRIGHT, L. C.; CONE, W. W. Population Dynamics of *Brachycorynella asparagi* (Homoptera: Aphididae) on Undisturbed Asparagus in Washington State. *Environmental Entomology*, v. 17, n. 5, p. 878–886, 10 1988. ISSN 0046-225X.
- [2] WELLINGS, P. W.; DIXON, A. F. G. The role of weather and natural enemies in determining aphid outbreaks. In: BARBOSA, P.; SCHULTZ, J. C. (Ed.). *Insect Outbreaks*. San Diego, USA: Academic Press, 1987. cap. 13, p. 313–346. ISBN 0120781484. Disponível em: <<https://ro.uow.edu.au/vc/8/>>.
- [3] PEREIRA, P. R. V. d. S.; SALVADORI, J. R.; LAU, D. *Identificação de adultos ápteros e alados das principais espécies de afídeos (Hemiptera: Aphididae) associadas a cereais de inverno no Brasil*. Passo Fundo, RS, Brasil, 2009. Disponível em: <http://www.cnpt.embrapa.br/biblio/co/p_co258.htm>.
- [4] SALVADORI, J. *Controle biológico de pulgões de trigo: o sucesso que perdura*. Passo Fundo, 1999. 2 p. Disponível em: <http://www.cnpt.embrapa.br/biblio/p_co27.htm>.
- [5] SOUSA-SILVA C.R.; ILHARCO, F. A. *Afídeos do Brasil e suas plantas hospedeiras (lista preliminar)*. São Carlos, SP: EDUFSCar, 1995. ISBN 1000209006482.
- [6] RESENDE, A. L. S. et al. *Amostragem de pulgões alados utilizando bandeja d'água e placa adesiva*. [S.l.], 2007. Disponível em: <<http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/625925>>.
- [7] SALVADORI, J.; SALLES, L. d. Controle biológico dos pulgões do trigo. In: MANOLE (Ed.). *Controle biológico no Brasil: parasitóides e predadores*. 1. ed. São Carlos, SP: [s.n.], 2002. cap. 26, p. 427–447. ISBN 85-204-1554-7.
- [8] SALVADORI, J.; PEREIRA, P.; SILVA, M. *Manejo de pulgões*. [S.l.], 2005. v. 75, 32-34 p.
- [9] ZÚÑIGA, E. S. *Controle biológico dos afídeos do trigo (Homoptera: Aphididae) por meio de parasitóides no planalto médio do Rio Grande do Sul*. 319 p. Tese (Doutorado) — Doutorado em Entomologia, Curitiba, 1982.
- [10] TRIGO, E. *Rede de Monitoramento de Pragas em Cereais de Inverno*. 2015. Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/43111123/treinamento-para-aprimorar-tecnicas-de-identificacao-de-insetos-em-cereais-de-inverno>>.
- [11] KENNEDY, J. S.; BOOTH, C. O.; KERSHAW, W. J. S. Host finding by aphids in the field. *Annals of Applied Biology*, v. 49, n. 1, p. 1–21, 1961.

- [12] HEATHCOTE, G. D. The comparison of yellow cylindrical, flat and water traps, and of johnson suction traps, for sampling aphids. *Annals of Applied Biology*, v. 45, n. 1, p. 133–139, 1957. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1744-7348.1957.tb00449.x>>.
- [13] MORRIS, R. First experiences with water traps. *Methodology*, 2018.
- [14] PEREIRA P. R. V. DA S.; SALVADORI, J. R. *Identificação dos principais parasitóides (Hymenoptera: Aphelinidae e Braconidae, Aphidiinae) envolvidos no controle biológico de pulgões (Hemiptera: Aphididae) em trigo no sul do Brasil*. Passo Fundo, RS, Brasil, 2005. Disponível em: <<https://www.embrapa.br/busca-de-publicacoes/-/publicacao/850114/identificacao-dos-principais-parasitoides-hymenoptera-aphelinidae-e-braconidae-aphidiinae-envolvidos-no-controle-biologico-de-pulgoes-hemiptera-aphididae-em-trigo-no-sul-do-brasil>>.
- [15] AKINTAYO, A. et al. A deep learning framework to discern and count microscopic nematode eggs. *Scientific Reports*, v. 8, n. 1, 2018. Disponível em: <<https://doi.org/10.1038/s41598-018-27272-w>>.
- [16] DING, W.; TAYLOR, G. Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture*, Elsevier, v. 123, p. 17–28, apr 2016. ISSN 0168-1699. Disponível em: <<https://doi.org/10.1016/J.COMPAG.2016.02.003>>.
- [17] NAZRI, A.; MAZLAN, N.; MUHARAM, F. Penyek: Automated brown planthopper detection from imperfect sticky pad images using deep convolutional neural network. *PLoS ONE*, Public Library of Science, v. 13, n. 12, 2018. Disponível em: <<https://doi.org/10.1371/JOURNAL.PONE.0208501>>.
- [18] PARTEL, V. et al. Automated vision-based system for monitoring asian citrus psyllid in orchards utilizing artificial intelligence. *Computers and Electronics in Agriculture*, v. 162, p. 328–336, jul 2019. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2019.04.022>>.
- [19] SUN, Y. et al. Automatic in-trap pest detection using learning for pheromone-based *Dendroctonus valens* monitoring. *Biosystems Engineering*, v. 176, p. 140–150, dec 2018. ISSN 15375110. Disponível em: <<https://doi.org/10.1016/j.biosystemseng.2018.10.012>>.
- [20] LINS, E. A. *Uma metodologia de contagem e classificação de afídeos utilizando visão computacional*. Dissertação (Mestrado) — Programa de Pós-Graduação em Computação Aplicada, 2018. Instituto de Ciências Exatas e Geociências – ICEG. Disponível em: <<http://tede.upf.br/jspui/handle/tede/1505>>.
- [21] LINS, E. A. et al. A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture*, v. 169, p. 105200, 2020. ISSN 0168-1699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0168169919306039>>.

- [22] LINS, E. A. et al. *AphidCV*. 2019. Patente: Programa de Computador. Número do registro: BR512019000884-7, Instituto Nacional da Propriedade Industrial. Acesso em: Mai. 09, 2019.
- [23] LAZZARETTI, A. T. et al. Trapsystem - uma aplicação para gerenciamento de dados coletados a partir de armadilhas de insetos. In: *REUNIÃO DA COMISSÃO BRASILEIRA DE PESQUISA DE TRIGO E TRITICALE*. [S.l.]: Comissão Brasileira de Pesquisa de Trigo e Triticale, 2016.
- [24] KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, v. 147, p. 70–90, 2018. ISSN 0168-1699. Disponível em: <<https://doi.org/10.1016/j.compag.2018.02.016>>.
- [25] ZHONG, Y. et al. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 5, p. 1489, 2018. ISSN 1424-8220. Disponível em: <<https://doi.org/10.3390/s18051489>>.
- [26] LI, W. et al. Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline. *Scientific Reports*, v. 9, n. 1, 2019. Disponível em: <<https://doi.org/10.1038/s41598-019-43171-0>>.
- [27] FUENTES, A. et al. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors (Switzerland)*, v. 17, n. 9, 2017. Disponível em: <<https://doi.org/10.3390/s17092022>>.
- [28] DE CESARO JÚNIOR, T.; RIEDER, R. Automatic identification of insects from digital images: A survey. *Computers and Electronics in Agriculture*, v. 178, p. 105784, 2020. ISSN 0168-1699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0168169920311224>>.
- [29] GULLI, A.; PAL, S. *Deep learning with Keras*. [S.l.]: Packt Publishing Ltd, 2017.
- [30] LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. ISSN 1476-4687. Disponível em: <<https://doi.org/10.1038/nature14539>>.
- [31] Lecun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219. Disponível em: <<https://doi.org/10.1109/5.726791>>.
- [32] CHEN, J. et al. Automatic segmentation and counting of aphid nymphs on leaves using convolutional neural networks. *Agronomy*, v. 8, n. 8, 2018. Disponível em: <<https://doi.org/10.3390/agronomy8080129>>.
- [33] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, v. 25, 01 2012.

- [34] Ren, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 6, p. 1137–1149, 2016. Disponível em: <<https://doi.org/10.1109/TPAMI.2016.2577031>>.
- [35] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2015. p. 3431–3440. Disponível em: <<https://ieeexplore.ieee.org/document/7298965>>.
- [36] FISCHER, P.; DOSOVITSKIY, A.; BROX, T. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014. Withdrawn. Disponível em: <<http://arxiv.org/abs/1405.5769>>.
- [37] DAI, J. et al. R-fcn: Object detection via region-based fully convolutional networks. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2016. (NIPS'16), p. 379–387. ISBN 9781510838819. Disponível em: <<https://dl.acm.org/doi/10.5555/3157096.3157139>>.
- [38] He, K. et al. Mask r-cnn. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. [s.n.], 2017. p. 2980–2988. Disponível em: <<https://doi.org/10.1109/ICCV.2017.322>>.
- [39] LIU, W. et al. Ssd: Single shot multibox detector. In: LEIBE, B. et al. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 21–37. ISBN 978-3-319-46448-0. Disponível em: <https://doi.org/10.1007/978-3-319-46448-0_2>.
- [40] Redmon, J. et al. You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2016. p. 779–788. Disponível em: <<https://doi.org/10.1109/CVPR.2016.91>>.
- [41] Lin, T. et al. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 42, n. 2, p. 318–327, 2018. Disponível em: <<https://doi.org/10.1109/TPAMI.2018.2858826>>.
- [42] SAEIDI, M.; AHMADI, A. Deep Learning Based on Parallel CNNs for Pedestrian Detection. *International Journal of Information & Communication Technology Research*, v. 10, n. 4, 2018. Disponível em: <<http://ijict.itrc.ac.ir/article-1-410-en.html>>.
- [43] SZEGEDY, C.; TOSHEV, A.; ERHAN, D. Deep neural networks for object detection. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13, v. 2), p. 2553–2561.
- [44] SERMANET, P. et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2014.

- [45] Girshick, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: [s.n.], 2014. p. 580–587. ISBN 978-1-4799-5118-5.
- [46] Lampert, C. H.; Blaschko, M. B.; Hofmann, T. Beyond sliding windows: Object localization by efficient subwindow search. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. [s.n.], 2008. p. 1–8. Disponível em: <<https://ieeexplore.ieee.org/document/4587586>>.
- [47] Girshick, R. Fast r-cnn. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [s.n.], 2015. p. 1440–1448. Disponível em: <<https://doi.org/10.1109/ICCV.2015.169>>.
- [48] ALEMAYEHU, D. M.; MENGISTU, A. D.; GEBEYEHU, S. Computer vision for ethiopian agricultural crop pest identification. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 3, n. 1, p. 209–214, 2016. Disponível em: <<http://ijeecs.iaescore.com/index.php/IJECS/article/view/4153>>.
- [49] AVILA-GEORGE, H. et al. Using artificial neural networks for detecting damage on tobacco leaves caused by blue mold. *International Journal of Advanced Computer Science and Applications*, v. 9, p. 579–583, 09 2018. Disponível em: <<https://doi.org/10.14569/IJACSA.2018.090873>>.
- [50] BAKKAY, M. C. et al. Automatic detection of individual and touching moths from trap images by combining contour-based and region-based segmentation. *IET Computer Vision*, v. 12, n. 2, p. 138–145, 2018. ISSN 1751-9632 VO - 12. Disponível em: <<https://doi.org/10.1049/iet-cvi.2017.0086>>.
- [51] BASATI, Z.; RASEKH, M.; ABBASPOUR-GILANDEH, Y. Using different classification models in wheat grading utilizing visual features. *International Agrophysics*, v. 32, n. 2, p. 225–235, 2018. Disponível em: <<https://doi.org/10.1515/intag-2017-0008>>.
- [52] BISGIN, H. et al. Comparing SVM and ANN based Machine Learning Methods for Species Identification of Food Contaminating Beetles. *Scientific Reports*, v. 8, n. 1, 2018. Disponível em: <<https://doi.org/10.1038/s41598-018-24926-7>>.
- [53] CHENG, X. et al. Pest identification via deep residual learning in complex background. *Computers and Electronics in Agriculture*, v. 141, p. 351–356, sep 2017. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2017.08.005>>.
- [54] CRUZ, A. C. et al. X-fido: An effective application for detecting olive quick decline syndrome with deep learning and data fusion. *Frontiers in Plant Science*, v. 8, 2017. Disponível em: <<https://doi.org/10.3389/fpls.2017.01741>>.

- [55] ESPINOZA, K. et al. Combination of image processing and artificial neural networks as a novel approach for the identification of *bemisia tabaci* and *frankliniella occidentalis* on sticky traps in greenhouse agriculture. *Computers and Electronics in Agriculture*, v. 127, p. 495–505, sep 2016. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2016.07.008>>.
- [56] KALAMATIANOS, R. et al. Dirt: The dacus image recognition toolkit. *Journal of Imaging*, v. 4, n. 11, 2018. Disponível em: <<https://doi.org/10.3390/jimaging4110129>>.
- [57] LI, X. et al. Corn classification system based on computer vision. *Symmetry*, v. 11, n. 4, 2019. Disponível em: <<https://doi.org/10.3390/sym11040591>>.
- [58] LIU, Z. et al. Localization and Classification of Paddy Field Pests using a Saliency Map and Deep Convolutional Neural Network. *Scientific Reports*, v. 6, n. 1, p. 20410, apr 2016. ISSN 2045-2322. Disponível em: <<https://doi.org/10.1038/srep20410>>.
- [59] LIU, L. et al. Pestnet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access*, v. 7, p. 45301–45312, 2019. ISSN 2169-3536. Disponível em: <<https://doi.org/10.1109/ACCESS.2019.2909522>>.
- [60] MOUSAVI, S. A. et al. Enhancing the diagnosis of corn pests using gabor wavelet features and svm classification. *Journal of Scientific and Industrial Research*, v. 75, n. 6, p. 349–354, 2016. Disponível em: <<http://nopr.niscair.res.in/handle/123456789/34349>>.
- [61] PICON, A. et al. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Computers and Electronics in Agriculture*, 2018. ISSN 0168-1699. Disponível em: <<https://doi.org/10.1016/j.compag.2018.04.002>>.
- [62] PRIYADHARSHINI, M. K.; SIVAKAMI, R.; JANANI, M. Sooty mould mango disease identification using deep learning. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, v. 8, n. 5s, p. 402–405, 2019. ISSN 2278-3075. Disponível em: <<https://www.ijitee.org/wp-content/uploads/papers/v8i5s/ES3452018319.pdf>>.
- [63] RAM, S.; RODRÍGUEZ, J. J. Size-Invariant Detection of Cell Nuclei in Microscopy Images. *IEEE Transactions on Medical Imaging*, v. 35, n. 7, p. 1753–1764, 2016. Disponível em: <<https://doi.org/10.1109/TMI.2016.2527740>>.
- [64] RAMCHARAN, A. et al. Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, v. 8, 2017. Disponível em: <<https://doi.org/10.3389/fpls.2017.01852>>.
- [65] ROLDÁN-SERRATO, K. L.; ESCALANTE-ESTRADA, J.; RODRÍGUEZ-GONZÁLEZ, M. Automatic pest detection on bean and potato crops by applying neural classifiers. *Engineering in Agriculture, Environment and Food*, v. 11, n. 4, p. 245–255, oct 2018. ISSN 18818366. Disponível em: <<https://doi.org/10.1016/j.eaef.2018.08.003>>.

- [66] SHEN, Y. et al. Detection of stored-grain insects using deep learning. *Computers and Electronics in Agriculture*, v. 145, p. 319–325, feb 2018. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2017.11.039>>.
- [67] Siva Sangari, A.; SARASWADY, D.; SASIKUMAR, G. Performance evaluation of optimal parameters for pest image segmentation using FCM and ACO. *Indian Journal of Science and Technology*, v. 9, n. 39, 2016. Disponível em: <<https://doi.org/10.17485/ijst/2016/v9i39/100774>>.
- [68] TAN, W.; ZHAO, C.; WU, H. Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimedia Tools and Applications*, v. 75, n. 24, p. 16741–16761, 2016. Disponível em: <<https://doi.org/10.1007/s11042-015-2940-7>>.
- [69] WEN, C. et al. Pose estimation-dependent identification method for field moth images using deep learning architecture. *Biosystems Engineering*, v. 136, p. 117–128, aug 2015. ISSN 15375110. Disponível em: <<https://doi.org/10.1016/j.biosystemseng.2015.06.002>>.
- [70] XIA, D. et al. Insect detection and classification based on an improved convolutional neural network. *Sensors*, v. 18, n. 12, p. 4169, nov 2018. ISSN 1424-8220. Disponível em: <<https://doi.org/10.3390/s18124169>>.
- [71] YUE, Y. et al. Deep recursive super resolution network with laplacian pyramid for better agricultural pest surveillance and detection. *Computers and Electronics in Agriculture*, v. 150, p. 26–32, jul 2018. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2018.04.004>>.
- [72] He, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [s.n.], 2016. p. 770–778. ISSN 1063-6919. Disponível em: <<https://doi.org/10.1109/CVPR.2016.90>>.
- [73] YAO, Q. et al. Segmentation of touching insects based on optical flow and NCuts. *Elsevier*, 2013. Disponível em: <<https://doi.org/10.1016/j.biosystemseng.2012.11.008>>.
- [74] ZHAO, Y. et al. Fov expansion of bioinspired multiband polarimetric imagers with convolutional neural networks. *IEEE Photonics Journal*, v. 10, n. 1, p. 1–14, feb 2018. ISSN 1943-0655. Disponível em: <<https://doi.org/10.1109/JPHOT.2017.2783039>>.
- [75] ZHANG, Z. et al. Animal detection from highly cluttered natural scenes using spatio-temporal object region proposals and patch verification. *IEEE Transactions on Multimedia*, v. 18, p. 1–1, 10 2016. Disponível em: <<https://doi.org/10.1109/TMM.2016.2594138>>.
- [76] HU, K. et al. Defect identification method for poplar veneer based on progressive growing generated adversarial network and mask r-cnn model. *BioResources*, v. 15, n. 2, p. 3041–3052, 2020. ISSN 1930-2126. Disponível em: <<https://ojs.cnr.ncsu.edu/index.php/BioRes/issue/view/92>>.

- [77] LIU, X.; HU, C.; LI, P. Automatic segmentation of overlapped poplar seedling leaves combining mask r-cnn and dbscan. *Computers and Electronics in Agriculture*, v. 178, p. 105753, 2020. ISSN 0168-1699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0168169920311777>>.
- [78] HUI, J. *Image segmentation with Mask R-CNN*. 2018. Medium. Disponível em: <<https://jonathan-hui.medium.com/image-segmentation-with-mask-r-cnn-ebe6d793272>>.
- [79] Lin, T. et al. Feature pyramid networks for object detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 936–944.
- [80] Xie, S. et al. Aggregated residual transformations for deep neural networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 5987–5995.
- [81] Li, Y. et al. Fully convolutional instance-aware semantic segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 4438–4446.
- [82] ABDULLA, W. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. Github, 2017. Disponível em: <https://github.com/matterport/Mask_RCNN>.
- [83] BOBBA, R. *Taming the Hyper-Parameters of Mask RCNN*. 2019. Analytics Vidhya - Community of Analytics and Data Science professionals. Disponível em: <<https://medium.com/analytics-vidhya/taming-the-hyper-parameters-of-mask-rcnn-3742cb3f0e1b>>.
- [84] LAZZARETTI, A. T. et al. Agrodb—integration of database management systems with crop models. In: *Proceedings of the 8th International Congress on Environmental Modelling and Software*. [S.l.]: iEMSs Society, 2016. p. 194–201.
- [85] DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning, ACM*. [S.l.: s.n.], 2006. v. 06, p. 8.
- [86] DE CESARO JÚNIOR, T.; RIEDER, R. Uma implementação baseada em mask r-cnn para detecção de insetos em imagens digitais. In: *XX-XIII Conference on Graphics, Patterns and Images (SIBGRAPI '20) - Workshop of Works in Progress*. [s.n.], 2020. p. 139–142. Disponível em: <https://sol.sbc.org.br/index.php/sibgrapi_estendido/article/view/12996/12850>.



UPF

UNIVERSIDADE
DE PASSO FUNDO

UPF Campus I - BR 285, São José
Passo Fundo - RS - CEP: 99052-900
(54) 3316 7000 - www.upf.br