

GUILHERMI MARTINS CRISPI

**USO DE REDES NEURAI PROFUNDAS NA AVALIAÇÃO DO ATAQUE DE
MOSCA MINADORA AO TOMATEIRO**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do programa de Pós-Graduação em Engenharia Agrícola, para obtenção do título de *Magister Scientiae*.

Orientador: Domingos Sárvio Magalhães
Valente

**VIÇOSA – MINAS GERAIS
2022**

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade
Federal de Viçosa - Campus Viçosa**

T

C932u
2022
Crispi, Guilhermi Martins, 1988-
Uso de redes neurais profundas na avaliação do ataque de
mosca minadora ao tomateiro / Guilhermi Martins Crispi. –
Viçosa, MG, 2022.
1 dissertação eletrônica (51 f.): il. (algumas color.).

Inclui apêndices.

Orientador: Domingos Sarvio Magalhaes Valente.

Dissertação (mestrado) - Universidade Federal de Viçosa,
Departamento de Engenharia Agrícola, 2022.

Referências bibliográficas: f. 41-47.

DOI: <https://doi.org/10.47328/ufvbbt.2022.131>

Modo de acesso: World Wide Web.

1. Pragas agrícolas. 2. Inteligência artificial. 3. Redes
neurais (Computação). 4. Aprendizado profundo (Aprendizado
do computador). I. Valente, Domingos Sarvio Magalhaes.
II. Universidade Federal de Viçosa. Departamento de Engenharia
Agrícola. Programa de Pós-Graduação em Engenharia Agrícola.
III. Título.

CDD 22. ed. 632.9

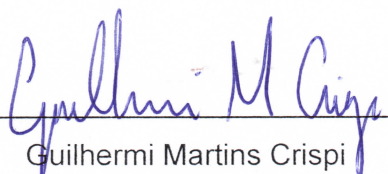
GUILHERMI MARTINS CRISPI

**REDES NEURAIAS PROFUNDAS PARA ESTIMATIVA DA SEVERIDADE DO
SINTOMA CAUSADO PELA MOSCA MINADORA NO TOMATEIRO**

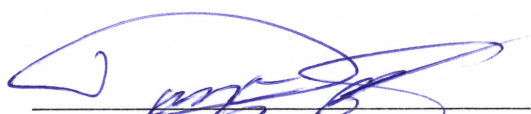
Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do programa de Pós-Graduação em Engenharia Agrícola, para obtenção do título de *Magister Scientiae*.

APROVADA: 24 de fevereiro de 2022.

Assentimento:



Guilhermi Martins Crispi
Autor



Domingos Sárvio Magalhães Valente
Orientador

AGRADECIMENTOS

A Deus pela força e resiliência para superar mais este desafio com sucesso.

Agradeço a todas as pessoas que contribuíram diretamente e indiretamente para desenvolvimento desta pesquisa.

Aos meus pais pelo esforço e apoio.

Aos professores de toda minha caminhada acadêmica.

Ao Domingos Sárvio Magalhães Valente pela orientação.

Aos amigos, Arthur Ribeiro Peron e Gabriel Feital Fusaro pela ajuda na coleta dos dados no campo.

A minha namorada, Nazle da Cruz Schuery por todo apoio e incentivo nos momentos difíceis.

À Universidade Federal de Viçosa e ao Departamento de Engenharia Agrícola pela oportunidade de realização do curso.

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão da bolsa de estudos.

RESUMO

CRISPI, Guilhermi Martins, M.Sc., Universidade Federal de Viçosa, fevereiro de 2021. **Uso de redes neurais profundas na avaliação do ataque de mosca minadora ao tomateiro.** Orientador: Domingos Sárvio Magalhães Valente.

Um dos maiores problemas da agricultura mundial é com relação às pragas e doenças. Dentre as culturas, a cultura do tomate é uma das mais susceptíveis ao ataque de pragas e doenças. No tomateiro, uma das principais pragas são as moscas minadoras de folha do gênero *Liriomyza*, especialmente da espécie *sativae*. A detecção e quantificação da severidade da infestação é importante para definir o momento de controle e definir eficácia dos sistemas de controle. A quantificação da severidade é realizado via amostragem de campo de forma manual por técnicos treinados. No entanto, a avaliação manual exige técnicos treinados, e mesmo assim poderá levar a erros de estimativa devido a subjetividade do processo de avaliação. Dessa forma, o objetivo desse trabalho foi o desenvolvimento de um modelo de inteligência artificial para detecção e estimativa automática da severidade dos sintomas do ataque da mosca-minadora em folhas do tomate. O conjunto de dados para este estudo reuniu 1932 imagens capturadas nas condições de campo contendo uma folha com o sintoma da praga em evidência. Fez-se a anotação manual das três classes presentes em todas as imagens, sendo elas, o plano de fundo, a folha do tomate e o sintoma foliar da mosca minadora. Foi comparado três arquiteturas e quatro diferentes espinhas dorsais (*backbone*) para tarefa de segmentação semântica multiclasse utilizando-se das métricas acurácia, precisão, revogação e IoU. Sequencialmente, fez-se a estimativa da severidade do sintoma na máscara predita pelo modelo utilizando-se a máscara anotada manualmente para quantificação. Comparando-se todos os modelos e espinhas dorsais utilizadas, o modelo U-Net com espinha dorsal Inceptionv3 alcançou melhor resultado de IoU médio atingindo 77,71%, seguido do modelo FPN com espinha dorsal DenseNet121, atingindo o IoU médio de 76,62%. Analisando-se o IoU separadamente para classe do sintoma em estudo, o modelo FPN com espinha dorsal DenseNet121 alcançou resultado de 61,02%, seguido do modelo LinkNet também com espinha dorsal DenseNet121 com resultado de IoU de 60,99%. Para estimativa da severidade o modelo FPN obteve melhor resultado quando comparado aos demais, sendo as espinhas dorsais ResNet34 e DenseNet121 as que apresentaram menor valor de RSME, o que confirma ainda os

valores de IoU encontrados para estes modelos. Os experimentos computacionais demonstrados nesta pesquisa se mostraram promissores, principalmente pela capacidade dos modelos em segmentar automaticamente objetos pequenos em imagens com condições desafiadoras de iluminação e fundo complexo, principalmente com a utilização de um banco de dados com desbalanceamento das classes.

Palavras-chave: Pragas agrícolas. Inteligência artificial. Redes neurais convolucionais. Segmentação semântica.

ABSTRACT

CRISPI, Guilhermi Martins, M.Sc., Universidade Federal de Viçosa, February, 2021.
Use of deep neural networks to evaluate leaf-miner flies attack on tomato plants.
Adviser: Domingos Sárvio Magalhães Valente.

One of the biggest problems of global agriculture is with regard to pests and diseases. Among crops, tomato culture is one of the most susceptible to the attack of pests and diseases. In tomato, one of the main pests are the leaf minadora flies of the genus *Liriomyza*, especially of the species *sativae*. The detection and quantification of the severity of the infestation is important to define the moment of control and define the effectiveness of the control systems. Severity quantification is performed by field sampling manually by trained technicians. However, manual evaluation requires trained technicians, and even then, can lead to estimation errors due to the subjectivity of the evaluation process. Thus, the objective of this work was the development of an artificial intelligence model for detection and automatic estimation of the severity of symptoms of the attack of the minadora fly in tomato leaves. The data set for this study gathered 1932 images captured in field conditions containing a leaf with the pest symptom in evidence. The three classes were manually annotated in all images, the background, tomato leaf and leaf symptom of the minadora fly. Three architectures and four *different backbones were compared for multiclass* semantic segmentation tasks using accuracy, precision, recall and IoU metrics. Sequentially, the severity of the symptom in the mask predicted by the model was estimated using the manually annotated mask for quantification. Comparing all models and dorsal spines used, the U-Net model with Inceptionv3 backbone achieved a better average IoU result reaching 77.71%, followed by the FPN model with Densenet121 as backbone, reaching the mean IoU of 76.62%. Analyzing the IoU separately for the class of the symptom under study, the FPN model with densenet121 backbone achieved a result of 61.02%, followed by the LinkNet model also with Densenet121 as backbone with a result of IoU of 60.99%. To estimate the severity, the FPN model obtained a better result when compared to the others, with the dorsal spines ResNet34 and DenseNet121 being the ones that presented the lowest RSME value, which also confirms the UOU values found for these models. The computational experiments demonstrated in this research were promising, mainly due to the ability of models to automatically segment small

objects in images with desharpening lighting and complex background conditions, mainly with the use of a database with class unbalance.

Keywords: Agricultural pests. Artificial intelligence. Convolutional neural networks. Semantic segmentation.

LISTA DE TABELAS

Tabela 1. Resumo da performance do modelo U-Net e espinhas dorsais no banco de dados de treinamento.....	26
Tabela 2. Resumo da performance do modelo LinkNet e espinhas dorsais no banco de dados de treinamento.....	26
Tabela 3. Resumo da performance do modelo FPN e espinhas dorsais no banco de dados de treinamento.....	27
Tabela 4. Comparação da performance do IoU médio para os modelos e espinhas dorsais no banco de dados de teste.....	27
Tabela 5. Comparação da performance do IoU separadamente para as classes do modelo U-Net e espinhas dorsais no banco de dados de teste.	28
Tabela 6. Comparação da performance do IoU separadamente para as classes do modelo LinkNet e espinhas dorsais no banco de dados de teste.....	28
Tabela 7. Comparação da performance do IoU separadamente para as classes do modelo FPN e espinhas dorsais no banco de dados de teste.....	29
Tabela 8. Tempo de treinamento dos modelos e diferentes espinhas dorsais.....	36
Tabela 9. Quantidade de parâmetros treináveis por modelo e espinha dorsais em milhões.....	38

LISTA DE FIGURAS

- Figura 1. Processo de anotação das imagens utilizando o Labelme. Fonte: Autor. ...20
- Figura 2. Imagem à esquerda e máscara de referência com as três classes anotadas, à direita. Fonte: Autor.....21
- Figura 3. Comparação entre entropia cruzada e focal loss. Fonte: (LIN et al., 2020).
.....24
- Figura 4. Representação da equação da Iou. Fonte: Autor.....25
- Figura 5. Resultados do treinamento do Modelo FPN para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16. Fonte: Autor.....29
- Figura 6. Resultados do treinamento do Modelo LinkNet para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16. Fonte: Autor.....30
- Figura 7. Resultados do treinamento do Modelo U-Net para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16. Fonte: Autor.....31
- Figura 8. Comparação visual dos resultados da segmentação semântica multiclasse do modelo FPN para o banco de dados de teste. (i) Imagem original;(ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.
.....31
- Figura 9. Comparação visual dos resultados da segmentação semântica multiclasse do modelo LinkNet para o banco de dados de teste. (i) Imagem original;(ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.
.....32
- Figura 10. Comparação visual dos resultados da segmentação semântica multiclasse do modelo U-Net para o banco de dados de teste. (i) Imagem original;(ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.
.....32
- Figura 11. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura FPN e diferentes espinhas dorsais A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte: Autor.....34
- Figura 12. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura LINKNET e diferentes

espinhas dorsais: A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte:
Autor.....35

Figura 13. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura U-NET e diferentes espinhas dorsais: A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte: Autor....35

SUMÁRIO

1 INTRODUÇÃO	12
2 REVISÃO BIBLIOGRÁFICA	14
3 MATERIAL E MÉTODOS	19
3.1 Criação do banco de dados	19
3.2 Anotação das imagens e criação das máscaras de referência	19
3.3 Extração e filtragem dos patches	21
3.4 Definição das arquiteturas e espinhas dorsais	21
3.5 Implementação dos modelos	22
3.6 Pré-processamento dos dados	23
3.7 Aumento do banco de dados	23
3.8 Métricas de avaliação dos modelos	24
3.9 Estimativa da severidade	25
4 RESULTADOS	26
4.1 Comparação dos modelos e espinhas dorsais	26
4.2 Severidade estimada pelos modelos	33
4.3 Tempo de treinamento dos modelos	35
5 DISCUSSÃO	36
6 CONCLUSÃO	40
REFERÊNCIAS	41
APÊNDICES	48
APÊNDICE A – Bibliotecas utilizadas	48
APÊNDICE B – Pré-processamento dos dados	48
Codificar Legendas	48
Normalizar imagens e expandir dimensões máscaras	48
Categorizar as máscaras	49
Augmentation e ImageDataGenerator	49
APÊNDICE C – Hiperparâmetros e condições	50
Parâmetros utilizados	50
Adicionar checkpoints, EarlyStopping e CSVLogger	50
APÊNDICE D – Arquitetura e espinhas dorsais	50
Espinhas dorsais	50
Definição das arquiteturas	51

1 INTRODUÇÃO

Os cultivos agrícolas são mundialmente atacados por uma variedade de pragas, especialmente em regiões de clima tropical e subtropical (MABVAKURE et al., 2016). O ataque de pragas agrícolas é considerado como um dos maiores responsáveis por perdas na produção agrícola mundial. Estima-se que a cada ano, as perdas na produção mundial por conta de pragas e doenças sejam da ordem de 40% (FAO, 2020). Dentre as pragas, os insetos podem ser particularmente prejudiciais, pois alimentam-se de diversas partes da planta afetando seu desenvolvimento, podendo ainda serem vetores de várias doenças (NALAM; LOUIS; SHAH, 2019). Em algumas situações as perdas podem chegar a 100% da lavoura se a detecção e o controle forem ineficazes ou inexistentes (GUSMÃO et al., 2000; PIKANÇO et al., 2007).

Dentre as culturas atacadas por pragas e doenças, a cultura do tomate é uma das mais susceptíveis (GILBERTSON; BATUMAN, 2013; LIU; WANG, 2020). As moscas minadoras do gênero *Liriomyza*, estão entre as principais pragas da cultura do tomate, possuindo o maior número de plantas hospedeiras relatadas dentro da família Agromyzida (SCHEFFER; HAWTHORNE, 2007). As larvas se alimentam abrindo galerias ou minas no parênquima foliar, por isso recebe este nome de larvas minadoras. Estima-se que as larvas de *Liriomyza sativae*, espécie mais associada ao tomate, causam até 65% de redução na fotossíntese pela planta (JOHNSON et al., 1983). Isso causa queda de produção acarretando em prejuízos econômicos para os produtores.

Nesse sentido a detecção e quantificação da severidade da infestação das pragas são ferramentas importantes do plano de amostragem que auxiliam na tomada de decisão para controle das pragas e proteção dos cultivos (ZHANG et al., 2019). A frequente inspeção visual da lavoura, visando à detecção e identificação de pragas, requer mão-de-obra especializada o que eleva o custo de produção. Na maioria dos casos, a inspeção consiste da amostragem manual e visual, ao longo do talhão utilizando-se do pano-de-batida (MOURA et al., 2014). Com base nesse levantamento, faz-se a quantificação para acompanhar o nível populacional das pragas presentes na lavoura. O problema desse método de amostragem de praga é que, muitas das vezes, quando a infestação é detectada, o dano já foi causado e o controle será mais difícil (LIN; CHANG; CHEN, 2020; WANG et al., 2013). Além disso

pode ser um processo lento dependendo do tamanho da área e sujeito a erros humanos, fazendo da amostragem manual e visual quase inviável para grandes áreas (BARBEDO, 2020; BOCK et al., 2020; DAWEI et al., 2019).

Assim, com o avanço das tecnologias e sensores e o surgimento da agricultura de precisão, a mesma entra como alternativa para diminuir custos fazendo-se aplicações mais eficientes e localizadas. Que embora ainda não estejam consolidadas, existe um aumento das pesquisas de monitoramento de insetos-praga por meio das ferramentas da agricultura de precisão. O processamento de imagens juntamente dos algoritmos de aprendizado de máquinas ajudam na identificação e mapeamento das áreas mais infestadas ajudando no Manejo Integrado de Pragas (MIP), controlando assim sua população no campo até o estágio onde não exista mais dano econômico (SENA JÚNIOR, 2002).

Como a identificação e reconhecimento das pragas podem ser feitas nas folhas das plantas, uma área em bastante evidência atualmente são as Redes Neurais Convolucionais, também conhecidas como CNNs, que performam convoluções para extrair e aprender características relevantes nas imagens e assim automaticamente identificar e detectar objetos alvo (LLORCA; YARES; MADERAZO, 2018). Além dos métodos automáticos de identificação e detecção, classificação e quantificação, existe também a segmentação que é a classificação dos pixels presentes nas imagens, bastante utilizados em diversas áreas na agricultura. Existe uma predominância dos estudos utilizando-se de imagens de armadilhas ou de imagens capturadas em ambiente controlado, porém um problema deste tipo de abordagem é que apenas se capturam insetos adultos e voadores (LIU et al., 2016). Atualmente o desafio está no desenvolvimento de métodos automáticos de identificação direta no campo das pragas e também dos sintomas causados pelas pragas nas folhas das culturas (FUENTES et al., 2017; HUANG et al., 2013; MA et al., 2014). Além de ser uma abordagem mais abrangente, existem outros desafios no que diz respeito à heterogeneidade das condições apresentadas à campo, como reflexão da luminosidade pela lâmina foliar, presença de sombras, diferenças de iluminação e o plano de fundo complexo, que podem causar confusão do modelo (BARBEDO, 2016; DAWEI et al., 2019; DIVYA; SANTHI, 2019; MUSTAFA; YAZID, 2016).

Portanto devido aos desafios apresentados, aliado aos avanços recentes das tecnologias, abriram um leque de possibilidades para a aplicação do aprendizado profundo em tarefas no campo através de imagens digitais com desempenhos mais

do que satisfatórios. Com isso, o objetivo desse trabalho foi o desenvolvimento de um modelo de inteligência artificial para detecção e estimativa automática da severidade dos sintomas do ataque da mosca-minadora em folhas do tomate.

2 REVISÃO BIBLIOGRÁFICA

O tomate (*Lycopersicon esculentum* Mill.) é uma das hortaliças mais consumidas no mundo. Além de sua importância alimentar e nutricional, possui alta concentração de licopeno, um agente oxidante que protege o organismo contra os radicais livres. Ele é consumido de diversas formas, desde o consumo in natura em saladas até mesmo em produtos industrializados como o molho ketchup e extratos (CONAB, 2019). É largamente produzido em mais de 175 países no mundo, sendo que em 2020, a produção mundial de tomate foi de 186,82 milhões de toneladas (FAO, 2021). Em 2016, data do último levantamento, a produção brasileira de tomate foi de 4.17 milhões de toneladas, ocupando uma área total de 63,98 mil hectares, sendo Goiás o maior estado produtor brasileiro, principalmente para indústria (CONAB, 2019). O tomateiro é suscetível a inúmeros patógenos e pragas que causam perda substancial da produção a cada ano.

Uma das principais pragas do tomate são as moscas minadoras, insetos da família *Agromyzidae*, são insetos cosmopolitas e hospedeiros de diversas plantas. No Brasil, as moscas-minadoras do gênero *Liriomyza* possuem maior importância, são polípagas atacando mais de 25 famílias de plantas, sendo que três espécies ocorrem naturalmente em quase todos os estados brasileiros, são elas *L. huidobrensis*, *L. sativae* e *L. trifolii* (BARROS, 2016). Os adultos são moscas de tamanho reduzido, aproximadamente 2mm de comprimento e possuem coloração preta e amarela no tórax e na face superior da cabeça. Estes insetos possuem hábito diurno e podem viver cerca de 26 dias, período onde as fêmeas fazem a postura endofítica de cerca de 500 a 700 ovos em ambas as faces das folhas do tomate. Após a eclosão, as larvas se alimentam do parênquima laminar deixando os sintomas conhecidos como minas ou galerias irregulares (CATAPAN et al., 2018). Estas minas além de prejudicarem a fotossíntese da planta, são também porta de entrada de patógenos como fungos e bactérias, o que acabam prejudicando a produção final.

A amostragem das plantas é feita periodicamente à campo por profissionais treinados durante o período vegetativo da planta, período mais crítico, sendo que o

nível de infestação pode representar de 2,5 a 65% das folhas do tomateiro (PRATISSOLI et al., 2015). O nível de ação para a tomada de decisão de controle da mosca minadora se dá quando 25% das folhas avaliadas apresentarem sintomas característicos de “minas” ou quando se constatar larvas vivas em 10% das plantas amostradas (BECKER et al., 2016; MOURA et al., 2014). A mosca minadora possui vários predadores naturais, como os parasitóides dos gêneros *Diglyphus*, *Chrysocharis* e *Halticoptera*, além de predadores de pupas, como as aranhas e formigas. Porém em sua grande maioria, o controle é feito com inseticidas do grupo dos piretróides, avermectinas e espinosinas o que aumentam o custo da produção (CATAPAN et al., 2018).

Além disso, atrelado ao aumento do custo de produção pelo uso dos defensivos, tem-se ainda o aumento dos custos com a amostragem das pragas à campo, que é um processo manual e lento e deve ser frequente a fim de se ter um controle satisfatório, quase inviável para grandes áreas (BOCK et al., 2020; DAWEI et al., 2019). Entretanto, quando ausente ou até mesmo quando realizada uma amostragem menos frequente e a infestação for detectada, o dano já foi causado. Sabe-se que quanto mais precoce for identificado, menores serão os custos com a aplicação de defensivos agrícolas (LIN; CHANG; CHEN, 2020; WANG et al., 2013). Assim, a agricultura de precisão entra como alternativa para diminuir custos fazendo-se aplicações mais eficientes e localizadas.

Com o avanço das tecnologias, desenvolvimento de sensores para obtenção de dados no campo e o surgimento da agricultura de precisão, a identificação precoce de pragas utilizando-se sistemas inteligentes é uma alternativa nos campos de produção. Com o uso dessas tecnologias pode-se reduzir o tempo entre a identificação e tomada de decisão pelo produtor (MUPPALA; GURUVIAH, 2020; WANG et al., 2013). Um número crescente de soluções, que combina o uso de imagens digitais com técnicas de aprendizado, tem sido desenvolvido para problemas relacionados à identificação de pragas (LIU; LEE; CHAHL, 2017; LIU; WANG, 2020; WANG et al., 2018). A maioria dos estudos recentes utilizam imagens digitais capturadas por sensores RGB (abreviatura do inglês para as bandas do vermelho, verde e azul) devido a seu baixo custo e portabilidade (BARBEDO, 2019; BOCK et al., 2020; KARISTO et al., 2018; SHIJIE et al., 2017; ZHENG et al., 2020). O equipamento utilizado para captura das imagens pode também impactar as características registradas na imagem assim como também, a resolução e compressão do arquivo

(BARBEDO, 2019). Existe uma preferência pelo uso de câmeras digitais DSLR, que permite uma melhor performance óptica. Porém, os avanços na tecnologia permitem agora a captura de fotos em ambientes com pouca iluminação com qualidade suficiente utilizando o aparelho celular (BARBEDO, 2016).

Alguns métodos de identificação automática de pragas baseiam-se no sintoma do dano causado na planta (FUENTES et al., 2017; HUANG et al., 2013; MA et al., 2014). Porém, existe uma predominância na detecção e identificação direta das pragas nas plantas. Atualmente, o desafio está na identificação das espécies em estágio inicial de desenvolvimento, por conta dos mais diversos estádios morfológicos que as pragas apresentam (LIU et al., 2016).

Outra abordagem bastante utilizada é a detecção e identificação de pragas utilizando-se imagens de armadilhas (CHEN et al., 2018; MUPPALA; GURUVIAH, 2020). Neste método obtém-se uma maior acurácia, por se tratar de imagens com o plano de fundo mais homogêneo quando comparadas as imagens capturadas no campo (BARBEDO, 2020). Entretanto, as armadilhas geralmente capturam apenas os insetos adultos voadores, portanto a detecção e identificação das pragas com imagens das folhas capturadas no campo é uma abordagem mais abrangente (LIU et al., 2016). Em estudos utilizando imagens de folhas capturadas nas condições de campo, existem outros desafios no que diz a respeito à heterogeneidade das mesmas (MAHARLOOEI et al., 2017), que varia com a arquitetura da planta e disposição das folhas, o que causa oclusões de algumas partes, a incidência e reflexão dos raios solares com a formação de sombra além do fundo complexo, especialmente quando a cor verde está presente no fundo da imagem (BARBEDO, 2016; DAWEI et al., 2019; DIVYA; SANTHI, 2019; MUSTAFA; YAZID, 2016). Como alternativa à captura no campo, tem sido usual a adoção de um fundo padrão de uma só cor ou a captura das imagens em laboratório com luz artificial, mas com um custo agregado operacional o que compromete a viabilidade de todo o processo (BARBEDO, 2016; KARISTO et al., 2018).

Porém de uma forma geral, os métodos automatizados de segmentação possuem as seguintes etapas, primeiro tem-se a aquisição das imagens das folhas com o sintoma, depois tem-se a segmentação da folha e sintoma, com a extração de características do sintoma e por último, a classificação das características extraídas. A segmentação é o processo que separa a região de interesse das demais regiões da imagem. Algumas técnicas de segmentação pelo tradicional aprendizado de máquina

(*Machine Learning*), são a limiarização de histograma de intensidade (MA et al., 2014; SRIWASTWA et al., 2018), que envolve a determinação de um valor limiar, o qual segrega a região de interesse do fundo e também o *clustering*, outra técnica que utiliza algoritmos de agrupamento de cores para segregar regiões homogêneas na imagem (CHANG et al., 2017; SRIWASTWA et al., 2018; UMA RANI; AMSINI, 2016). Tem-se ainda técnicas morfológicas, envolvendo a detecção das bordas dos objetos utilizando algoritmos como *canny edge* e *watershed* (SRIWASTWA et al., 2018; XIA et al., 2015)(SRIWASTWA et al., 2018; XIA et al., 2015).

Uma subárea do aprendizado de máquinas que está sendo utilizada em diversas aplicações é aprendizado profundo (*Deep Learning*), onde usa-se arquiteturas organizadas em camadas, onde cada camada utiliza os dados de saída da camada anterior como dados de entrada. Dessa forma, uma camada posterior na rede é resultado de transformações lineares e não lineares da camada anteriores (LECUN; BENGIO; HINTON, 2015). Bastante utilizado para a aplicações em processamento de imagem e visão computacional, sendo as CNN ou Redes Neurais Convolucionais as mais usuais (CHOWDHURY et al., 2021; LIU et al., 2019) . No caso da de segmentação de imagens digitais utilizando-se o aprendizado profundo de máquinas, existem diversas aplicações em diversos áreas na agricultura como detecção de pragas e doenças (CHEN et al., 2021; CHENG et al., 2017; CHOWDHURY et al., 2021; FUENTES et al., 2018; HIDAYATULOH; NURSALMAN; NUGRAHA, 2018), detecção e classificação de daninhas (DAWEI et al., 2019; KAZMI et al., 2015; MILIOTO; LOTTES; STACHNISS, 2018; MUPPALA; GURUVIAH, 2020) .

Existem diversas arquiteturas de CNNs para segmentação desenvolvidos para diversos fins, porém existe hoje um outro grupo de CNNs, que são chamadas de Redes Neurais Totalmente Convolucionais (*Fully connected neural networks - FCNNs*) que se assemelham à uma CNN tradicional, a única diferença é que não possuem a camada densa de *layers*, ao invés disso, possuem convoluções linha a linha o que performa o mesmo que a camada densa. As arquiteturas com maior evidência atualmente para segmentação semântica são a U-Net, criada originalmente para segmentação de imagens de microscopia biomédica (WENG; ZHU, 2021; RONNEBERGER; FISCHER; BROX, 2015), a FPN, desenvolvida primariamente para detecção de objetos (LIN et al., 2017) e segmentação de imagens para planejamento urbano e paisagens naturais (SEFERBEKOV et al., 2018) e a LinkNet, desenvolvida

para segmentação de imagens de paisagens urbanas (CHAURASIA; CULURCIELLO, 2018).

Todos as três arquiteturas utilizaram da técnica conhecida como aprendizado por transferência, a ideia é aplicar pesos das chamadas espinhas dorsais pré-treinadas no bloco de *encoder* dos modelos, o que traz ganhos significativos de desempenho (MOHANTY; HUGHES; SALATHÉ, 2016; SEFERBEKOV et al., 2018), sendo estes capazes de generalizar imagens fora do conjunto de dados do *ImageNet*. As espinhas dorsais utilizadas nos três modelos foram VGG16, ResNet34, Inceptionv3 e DenseNet121. Todas as espinhas dorsais foram treinadas usando o conjunto de dados *2012 ILSVRC ImageNet*. O *ImageNet* (<https://image-net.org>) é formalmente um projeto destinado a anotar e legendar manualmente quase 22.000 categorias de objetos separados para diversos fins em mais de 14 milhões de imagens. Estas redes pré-treinadas são incluídas na biblioteca do Keras.

Os algoritmos de aprendizado profundo utilizam a abordagem de descida gradiente estocástica para otimizar o aprendizado de uma determinada tarefa. Para isso, precisamos garantir que nossa representação matemática dos objetivos, também conhecida como funções de perda, seja capaz de cobrir até mesmo os casos extremos (JADON, 2020). A função perda examinada cada pixel individualmente comparando-se as previsões para cada classe com a máscara anotada manualmente. A perda focal é uma entropia cruzada categórica, porém adicionado um fator de ponderação γ na equação, ela reduz o peso de classes mais fáceis de serem preditas e permite ao modelo focar o treinamento em classes mais difíceis de aprendizado. Recomendado para trabalhos utilizando um banco de dados com desequilíbrio entre as classes (JADON, 2020; LIN et al., 2020). O coeficiente de Dados (*Dice*) é bem similar à Interseção-sobre-União (IoU), frequentemente utilizado como métrica para avaliar semelhança entre duas imagens, porém vem sendo bastante utilizado como função de perda de Dados (*Dice loss*).

Na avaliação do desempenho dos modelos, existem diversas métricas como a acurácia, precisão, revogação e ainda a Interseção-sobre-União (IoU). Ambas as métricas variam de 0 a 1, sendo que o valor 0 (zero) significa ausência de sobreposição das classes entre a máscara de referência e a predita, já o valor 1 (um), representa uma sobreposição completa, sendo este o resultado desejado. Porém algumas são mais recomendadas para determinadas tarefas, a IoU também conhecida por coeficiente de similaridade de Jaccard, é justamente a interseção da

máscara de referência com a predita pelo modelo sobre a união da máscara de referência com a predição do modelo (JADON, 2020). É uma métrica amplamente usada para calcular a semelhança entre duas imagens na tarefa de segmentação com banco de dados desbalanceado (JADON, 2020). Já a acurácia, precisão e revogação, já não são muito recomendadas, pois, apresentam um falso resultado, podem apresentar um ótimo resultado médio geral, porém quando se avalia a predição feita pelo modelo, não se tem predições para as classes menos amostradas, as classes mais amostradas acabam sombreando o resultado.

3 MATERIAL E MÉTODOS

3.1 Criação do banco de dados

O banco de dados de imagens deste estudo foi composto por 90 imagens de folhas de tomate com sintomas do ataque da mosca minadora (*Liriomyza spp.*), com dimensões de 2268x4032 pixels. As imagens foram obtidas em visita à duas propriedades distintas na cidade de Coimbra, localizado próximo à cidade de Viçosa, Minas Gerais. Para captura das imagens, utilizou-se um aparelho *smartphone* Huawei P20 Lite com resolução de 13MP. Em cada imagem fotografada, buscou-se evidenciar uma folha centralizando-a na imagem, a fim de evidenciar o sintoma em estudo. Todas as imagens foram capturadas no campo durante o dia, possuindo variações de iluminação e plano de fundo complexo, a fim de generalizar as condições de amostragem no campo.

Este banco de dados, foi previamente dividido entre treinamento, teste e validação para evitar a contaminação. Fez-se a divisão em treinamento e teste e posteriormente subdividiu-se o banco de dados de treinamento em treinamento e validação.

3.2 Anotação das imagens e criação das máscaras de referência

O processo de anotação é um processo importante na implementação dos modelos de aprendizado profundo pois é nesta etapa que é identificado e rotulado as classes para cada pixel amostrado. Para que haja um treinamento e classificação satisfatória, cada imagem deve ser anotada manualmente amostrando-se todas as classes presentes, no caso deste estudo, foram anotados a folha e o sintoma, sendo

o plano de fundo, segmentado automaticamente após a demarcação do alvo de interesse. A anotação das imagens é um processo totalmente supervisionado e demorado, é feita utilizando-se a ferramenta gráfica Labelme (<https://github.com/wkentaro/labelme>). Neste processo, definiu-se polígonos sobre cada região de interesse presente nas imagens (Figura 1) amostrando cada classe em questão para extração das informações dos pixels que compõem as classes (DOS SANTOS et al., 2019; FRÖHLICH; RODNER; DENZLER, 2010).

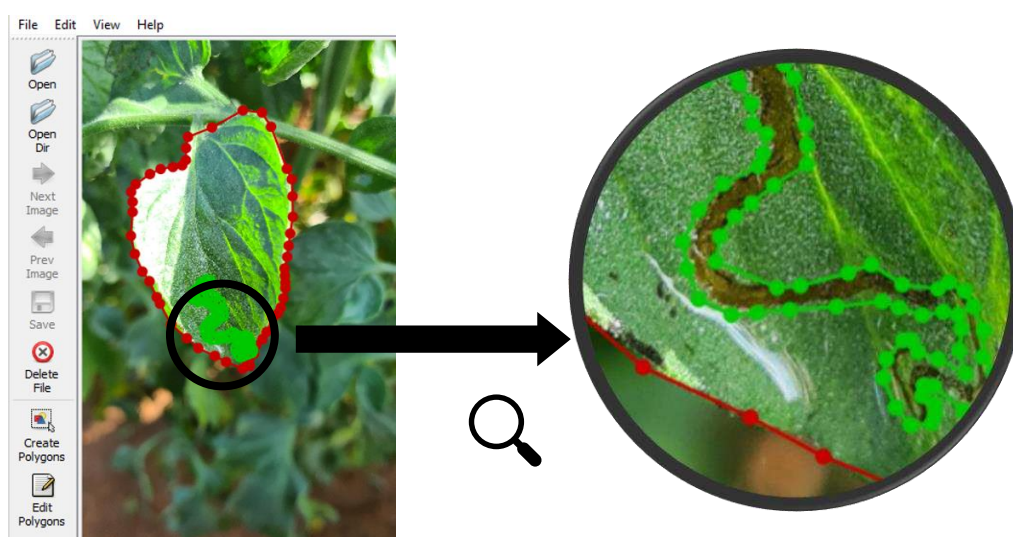


Figura 1. Processo de anotação das imagens utilizando o Labelme. Fonte: Autor.

Para cada imagem anotada, criou-se um arquivo no formato JSON com todas as informações e classes contidas na imagem. Em seguida, converteu-se cada arquivo JSON em uma máscara binária contendo as três classes no formato PNG. Estas máscaras foram utilizadas como referência, ou seja, utilizadas como dado de entrada dos modelos, juntamente das respectivas imagens (Figura 2).



Figura 2. Imagem à esquerda e máscara de referência com as três classes anotadas, à direita. Fonte: Autor.

3.3 Extração e filtragem dos patches

Nesta etapa, foi feita extração de patches de dimensões 256x256 pixels nas 90 imagens e as respectivas máscaras, totalizando 120 patches por imagem e máscara. Este processo é uma estratégia importante usualmente aplicada a uma rede com imagens de resoluções mais altas (RONNEBERGER; FISCHER; BROX, 2015), caso contrário a resolução seria limitada pela memória de processamento gráfico (GPU).

Outro ponto comum em trabalhos de segmentação semântica é o desequilíbrio das classes amostradas, que tem efeito prejudicial no desempenho de das redes neurais treinadas (CHAN et al., 2020). Portanto, após a extração dos patches das imagens e máscaras, foi feito ainda uma filtragem dos patches afim de fazer um balanceamento inicial das classes amostradas nos patches extraídos, fez-se então a exclusão de patches contendo mais de 95% de pixels correspondente à classe do plano de fundo.

3.4 Definição das arquiteturas e espinhas dorsais

Para o presente trabalho de segmentação semântica multiclasse, foram selecionadas três arquiteturas de redes neurais convolucionais, mais especificamente redes neurais totalmente convolucionais de acordo com sua relevância em trabalhos

atuais de segmentação. Uma rede totalmente convolucional (*Fully Convolutional Network - FCN*), é uma adaptação de redes neurais convolucionais já existentes que são utilizadas para classificação de imagens, de forma a proporcionar segmentações mais precisas com conjuntos de treinos pequenos (IGLOVIKOV; SHVETS, 2018). Foram definidos a U-Net, a FPN (GONÇALVES et al., 2021; SZEGEDY et al., 2016) e o LinkNet. Todos as três arquiteturas utilizaram da técnica conhecida como aprendizado por transferência, a ideia é aplicar pesos das chamadas espinhas dorsais pré-treinadas no bloco de *encoder* dos modelos. As espinhas dorsais utilizadas nos três modelos foram VGG16, ResNet34, Inceptionv3 e DenseNet121.

3.5 Implementação dos modelos

Essa pesquisa foi inteiramente implementada utilizando-se as plataformas de serviços em nuvem do Google, para processamento e treinamento utilizou-se o *Google Colaboratory* ou *Colab* e para armazenamento, o *Google Drive*. O *Colab* permite a escrita e execução de códigos em linguagem Python e instalação das bibliotecas externas. A configuração da máquina virtual utilizada foi: processador Intel(R) Xeon(R) de 2.2GHz, placa de vídeo Nvidia Tesla P100 com 16Gb de memória dedicada e 32Gb de memória RAM.

Os modelos foram implementados utilizando algoritmos desenvolvidos em linguagem Python (<http://www.python.org>), versão 3.7 e treinados usando a biblioteca Keras (<https://keras.io/>) com o Tensorflow (<https://www.tensorflow.org>), como estrutura (*back-end*) ambas versões 2.6.0, assim como a biblioteca *segmentation_models* (https://github.com/qubvel/segmentation_models).

Considerando os atuais desafios de trabalhos de segmentação semântica presentes na literatura, da utilização de imagens capturadas no campo com as mais diversas variações na intensidade e ângulo de incidência da luminosidade, presença de sombra, angulação da câmera na captura, além do plano de fundo complexo e confuso (BARBEDO, 2016, 2019). E evidenciando ainda os avanços tecnológicos atuais, fez-se ainda a utilização de um aparelho *smartphone* para captura das imagens no campo, sendo considerada ainda uma metodologia não destrutiva, por não se removerem as folhas para amostragem. Metodologia que ainda acelera todo o processo de captura das imagens no campo.

Os hiper parâmetros foram padronizados para todos os modelos, definiu-se como 3 (três) o número de classes, sendo elas, plano de fundo, folha e o sintoma em estudo. Fez-se o treinamento dos modelos utilizando-se os seguintes hiper parâmetros: 30 (trinta) de épocas (*epochs*), utilizando-se do IoU do banco de dados de validação como parada precoce (*early stopping*) e com paciência (*patience*) de 5 (cinco) épocas. Definiu-se ainda o tamanho de lote (*batch size*) de 16 (dezesesseis) e a *Softmax* como a função de ativação em todos os neurônios de saída da última camada. Adam, foi o algoritmo otimizador dos parâmetros das redes e por fim, definiu-se a taxa de aprendizado de 0,001.

3.6 Pré-processamento dos dados

Foi feita a recodificação das legendas nas máscaras de referência, ajustando os dados de acordo com as necessidades da métrica de avaliação escolhida para o modelo, atribuindo-se valores de pixel 0 para classe fundo, 1 para classe folha e 2 para classe do sintoma da mosca minadora. Posteriormente, fez-se a expansão da dimensão das máscaras seguida da categorização, o também chamado *one-hot encoding* separando-se cada classe em uma respectiva categoria, ou seja, binarizá-las em bandas separadas.

Ainda nesta etapa, foi feita a expansão da dimensão das imagens para que tenha as dimensões corretas para entrada no modelo. Fez-se a normalização das imagens ou escala das imagens, que ajusta o valor dos pixels para uma escala uniforme, no caso, entre 0 e 1. Colocar as características numa escala similar pode ajudar o gradiente descendente a convergir mais rapidamente para seu valor mínimo.

3.7 Aumento do banco de dados

Para aumentar e diversificar o banco de dados, aumentando a capacidade de generalização e ainda para impedir o super ajuste (*overfitting*) do modelo, todas as imagens e máscaras de referência do banco de dados de treinamento passaram pelo processo de aumento de dados (*data augmentation*) com diversas transformações (BARBEDO, 2020; PEREZ; WANG, 2017; TAKAHASHI; MATSUBARA; UEHARA, 2020). Para isso, utilizou-se a função *ImageDataGenerator* da biblioteca Keras com as seguintes transformações: rotação de 90 graus, faixa de deslocamento de largura,

faixa de deslocamento de altura, faixa de cisalhamento, faixa de zoom e reorientação horizontal e vertical. Ainda, para fazer predição dos pixels na região das bordas das imagens de entrada, as bordas foram espelhadas.

3.8 Métricas de avaliação dos modelos

Foi utilizado como função de perda o chamado perda total (*total loss*), que é a soma da perda focal (*focal loss*) com perda de Dice (*Dice loss*). A perda focal é uma entropia cruzada categórica, porém adicionado um fator de ponderação γ na equação. Recomendado para trabalhos utilizando um banco de dados com desequilíbrio entre as classes (JADON, 2020; LIN et al., 2020). O coeficiente de Dice é bem similar à Interseção-sobre-União (IoU), frequentemente utilizado como métrica para avaliar semelhança entre duas imagens, porém aqui foi aplicado como função de perda.

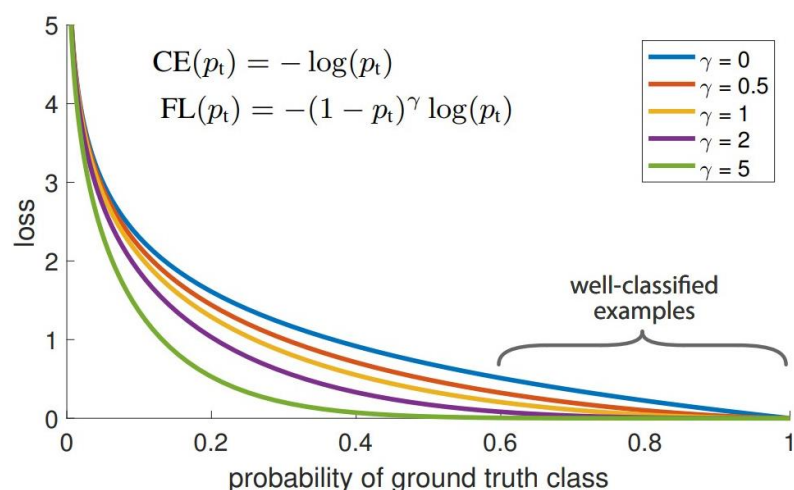


Figura 3. Comparação entre entropia cruzada e focal loss. Fonte: (LIN et al., 2020).

Para avaliação do desempenho dos modelos empregados nesta pesquisa, utilizou-se como métrica a acurácia, precisão, revogação e ainda a Interseção-sobre-União (IoU), também conhecida por coeficiente de similaridade de Jaccard, que é justamente a interseção da máscara de referência com a predita pelo modelo sobre a união da máscara de referência com a predição do modelo (Figura 5). É uma métrica amplamente usada para calcular a semelhança entre duas imagens (JADON, 2020).

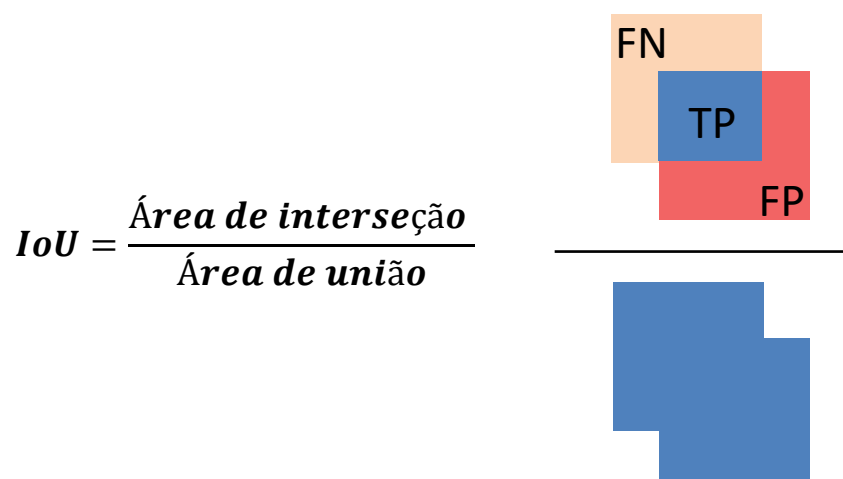


Figura 4. Representação da equação da Iou. Fonte: Autor.

3.9 Estimativa da severidade

A estimativa da severidade do ataque de pragas nas plantas, tradicionalmente é dado pela proporção do tecido atacado da planta em relação a área total da folha (BOCK et al., 2008, 2010, 2020; MADDEN; HUGHES; VAN DEN BOSCH, 2007). A imprecisão pode prejudicar o processo de controle da praga, desperdiçando recursos, podendo afetar a lucratividade do produtor. Medidas ou estimativas precisas são importantes para garantir que os dados sejam analisados corretamente. Além disso, os dados podem ser usados como um limite de decisão de acordo com dano de nível econômico, portanto, a necessidade de pulverização ou não. A severidade foi estimada pela equação (1), relação entre a quantidade de pixels contidas na classe lesão e na área total da folha (CHEN et al., 2021; GONÇALVES et al., 2021).

$$\textit{Severidade} = \frac{S}{S + L} * 100 \quad (1)$$

Onde,

L – Área total dos pixels contidos na classe folha;

S – Área dos pixels contidos na classe sintoma.

4 RESULTADOS

4.1 Comparação dos modelos e espinhas dorsais

Neste estudo, investigou-se a performance de diferentes modelos CNN com diferentes espinhas dorsais, todos os modelos CNN concluíram o processo de convergência em 6 épocas, com exceção do modelo FPN com a espinha dorsal VGG16, que convergiu com 7 épocas.

As Tabelas 1, 2 e 3 apresentam o desempenho dos modelos empregados nesta pesquisa de segmentação semântica utilizando-se do aprendizado por transferência. De uma forma geral, os modelos desempenharam bem a tarefa de segmentação no banco de dados de teste, sendo o modelo com melhor acurácia, precisão e revogação foi o FPN com espinha dorsal DenseNet121, com 91,56%, 88,63% e 87,71%, respectivamente. O segundo melhor modelo com desempenho em acurácia, precisão e revogação foi o U-Net com espinha dorsal Inceptionv3, com 91,58%, 87,84% e 87,59%, respectivamente. O modelo com pior resultado foi o LinkNet com a espinha dorsal VGG16, que obteve 81,23%, 73,94% e 67,87% de acurácia, precisão e revogação, respectivamente.

Tabela 1. Resumo da performance do modelo U-Net e espinhas dorsais no banco de dados de treinamento.

Espinha dorsal	Modelo de aprendizado profundo		
	U-Net		
	Acurácia de Teste (%)	Precisão Média (%)	Revogação Média (%)
VGG16	83.90	80.63	78.31
ResNet34	90.00	86.63	85.81
Inceptionv3	91.58	87.84	87.59
DenseNet121	91.25	87.83	87.74

Tabela 1. Resumo da performance do modelo U-Net e espinhas dorsais no banco de dados de treinamento.

Tabela 2. Resumo da performance do modelo LinkNet e espinhas dorsais no banco de dados de treinamento.

Espinha dorsal	Modelo de aprendizado profundo		
	LinkNet		

	Acurácia de Teste (%)	Precisão Média (%)	Revogação Média (%)
VGG16	81.23	73.94	67.87
ResNet34	88.66	84.62	84.77
Inceptionv3	91.06	87.09	87.39
DenseNet121	90.72	87.00	86.79

Tabela 3. Resumo da performance do modelo FPN e espinhas dorsais no banco de dados de treinamento.

Modelo de aprendizado profundo			
Espinha dorsal	FPN		
	Acurácia de Teste (%)	Precisão Média (%)	Revogação Média (%)
VGG16	89.27	85.58	84.31
ResNet34	90.53	87.29	86.27
Inceptionv3	91.10	87.98	86.82
DenseNet121	91.56	88.63	87.71

Para o IoU médio (Tabela 4), o modelo U-Net com espinha dorsal Inceptionv3 alcançou melhor resultado com 77,71%, seguido do modelo FPN com espinha dorsal DenseNet121 alcançou o IoU médio de 76,62, sendo estes os melhores resultado encontrado entre os modelos comparados. Para arquitetura LinkNet, o modelo com melhor resultado foi com a espinha dorsal DenseNet121, com 75,99%, já o modelo com pior resultado foi com espinha dorsal VGG16, obteve o IoU médio de 53,03%.

Tabela 4. Comparação da performance do IoU médio para os modelos e espinhas dorsais no banco de dados de teste.

Modelos de aprendizado profundo			
Espinha dorsal	U-Net	LinkNet	FPN
	IoU Médio (%)	IoU Médio (%)	IoU Médio (%)
VGG16	61.76	53.03	73.12
ResNet34	72.73	73.21	74.61
Inceptionv3	77.71	75.67	75.12
DenseNet121	76.33	75.99	76.62

Analisando-se o IoU separadamente para cada classe (Tabelas 5, 6 e 7), plano de fundo (B), área da folha (L) e sintoma (S), podemos entender melhor cada classe. O modelo FPN com espinha dorsal DenseNet121 obteve o melhor resultado para as classes estudadas, com 82,76%, 86,07% e 61,02% respectivamente. Já o modelo LinkNet com espinha dorsal VGG16 obteve resultado de 64,89% para a classe plano de fundo, 69,19% para a classe folha e apenas 25,02% para a classe do sintoma, sendo este o modelo com pior resultado de segmentação.

Tabela 5. Comparação da performance do IoU separadamente para as classes do modelo U-Net e espinhas dorsais no banco de dados de teste.

Espinha dorsal	Modelo de aprendizado profundo		
	U-Net		
	IoU Classe 1 (%)	IoU Classe 2 (%)	IoU Classe 3 (%)
VGG16	67.11	73.17	45.00
ResNet34	78.21	81.75	58.24
Inceptionv3	86.40	87.43	59.30
DenseNet121	84.52	86.68	57.79

Classe 1 – Plano de fundo; Classe 2 – Área da folha; Classe 3 – Sintoma da mosca minadora.

Tabela 6. Comparação da performance do IoU separadamente para as classes do modelo LinkNet e espinhas dorsais no banco de dados de teste.

Espinha dorsal	Modelo de aprendizado profundo		
	LinkNet		
	IoU Classe 1 (%)	IoU Classe 2 (%)	IoU Classe 3 (%)
VGG16	64.89	69.19	25.02
ResNet34	78.45	82.14	59.03
Inceptionv3	82.15	84.41	60.45
DenseNet121	81.93	85.04	60.99

Classe 1 – Plano de fundo; Classe 2 – Área da folha; Classe 3 – Sintoma da mosca minadora.

Tabela 7. Comparação da performance do IoU separadamente para as classes do modelo FPN e espinhas dorsais no banco de dados de teste.

Espinha dorsal	Modelo de aprendizado profundo		
	FPN		
	IoU Classe 1 (%)	IoU Classe 2 (%)	IoU Classe 3 (%)
VGG16	76.84	82.08	60.43
ResNet34	81.09	84.11	58.62
Inceptionv3	81.82	84.91	58.62
DenseNet121	82.76	86.07	61.02

Classe 1 – Plano de fundo; Classe 2 – Área da folha; Classe 3 – Sintoma da mosca minadora.

O IoU de treinamento e validação para cada época foi comparado graficamente para avaliar a qualidade do treinamento de cada modelo, como pode ser visto nas Figuras 5, 6 e 7. Ressaltando que ambos os modelos convergiram com 6 épocas, exceto o modelo FPN com espinha dorsal VGG16, que convergiu com 7 épocas.

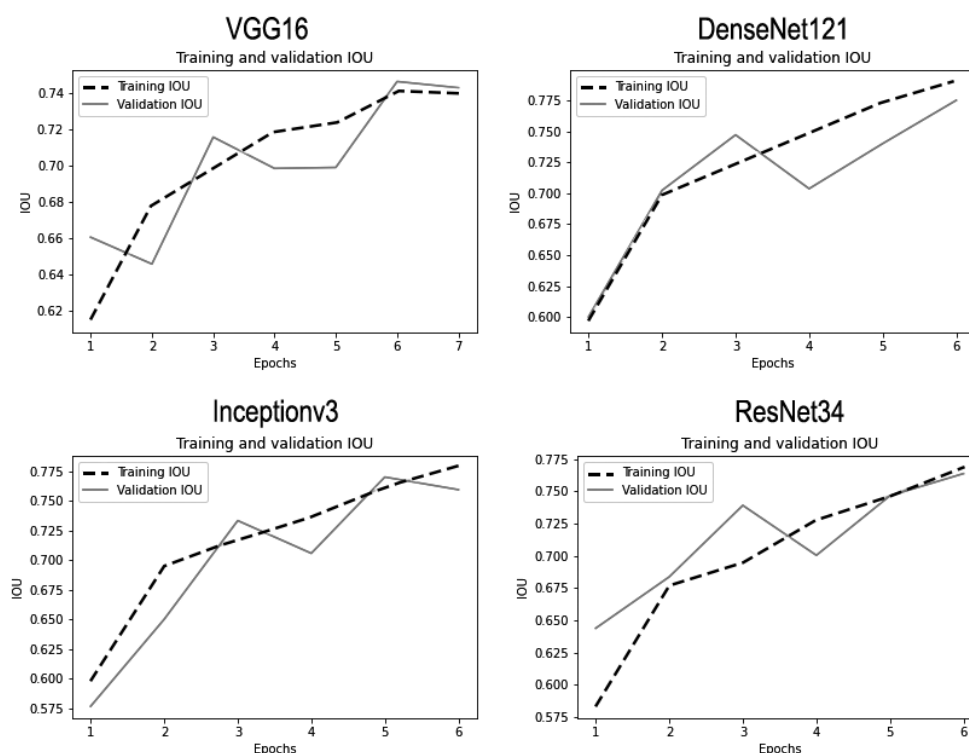


Figura 5. Resultados do treinamento do Modelo FPN para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16.

Fonte: Autor.

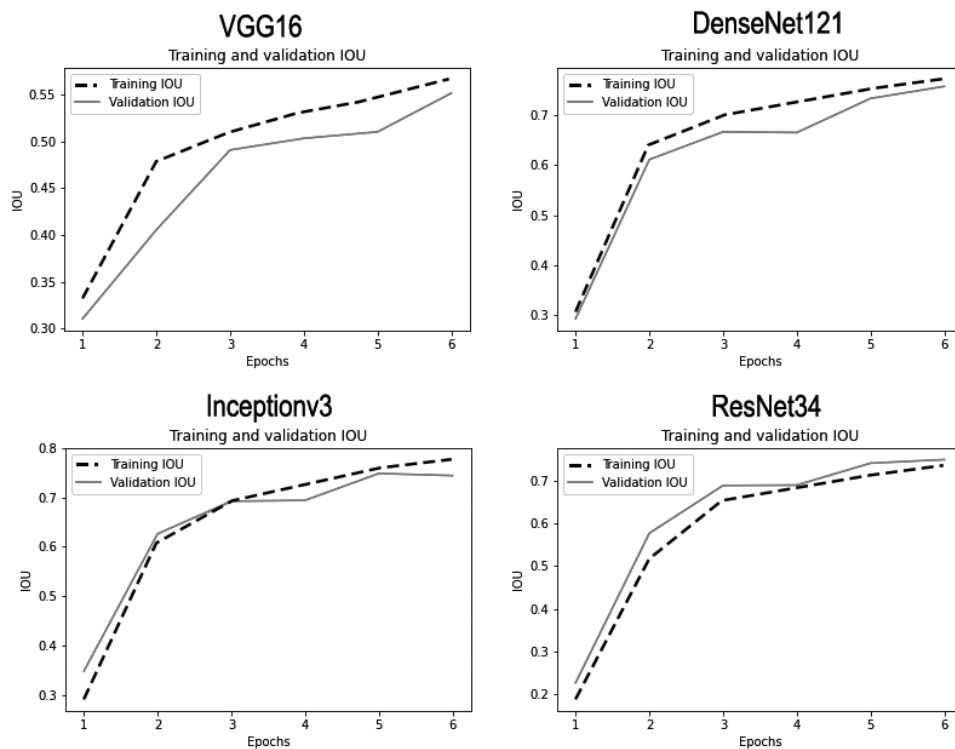


Figura 6. Resultados do treinamento do Modelo LinkNet para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16. Fonte: Autor.

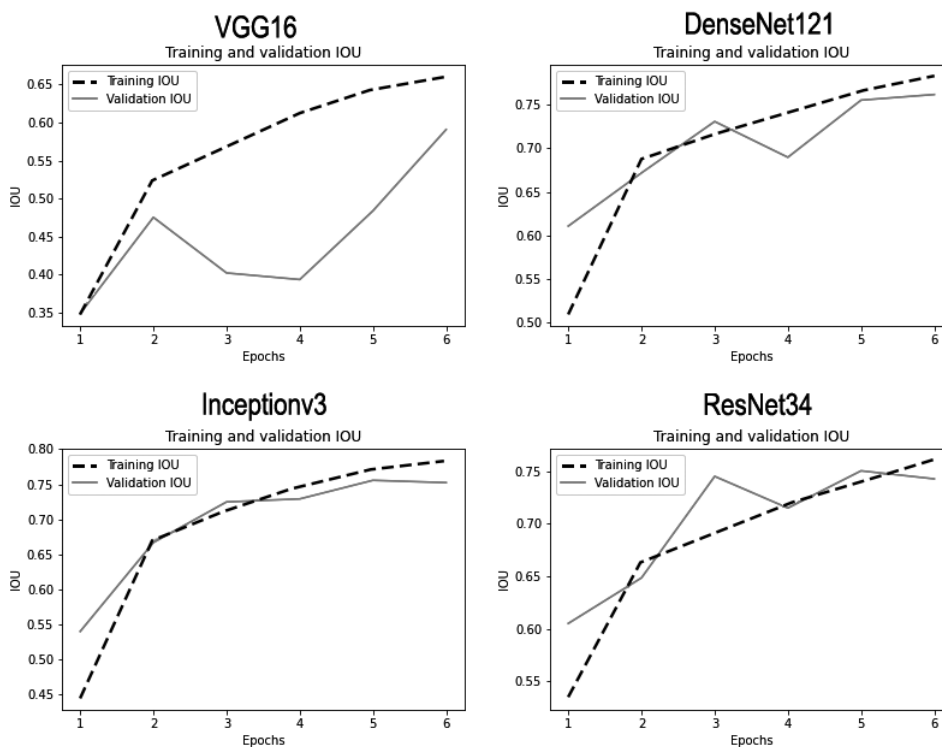


Figura 7. Resultados do treinamento do Modelo U-Net para as espinhas dorsais DenseNet121, Inceptionv3, ResNet34 e VGG16.

Fonte: Autor.

Como espinha dorsal dos modelos de segmentação semântica utilizados nesta pesquisa, a DenseNet121 alcançou melhores resultados porque pode segmentar melhor os pixels referentes a classe do sintoma. As espinhas dorsais que apresentaram pior resultado foram VGG16 seguido da ResNet34.

Nas Figuras 8, 9 e 10 estão dispostos os resultados da segmentação semântica multiclasse realizada pelos diferentes modelos CNNs e espinhas dorsais para o banco de dados de teste. Nesta comparação visual com a máscara de referência, a qual foi anotada manualmente, pode-se notar uma melhor segmentação das imagens com o modelo FPN, o que certifica os resultados dos valores de IoU mais altos obtidos para este modelo. Quanto às espinhas dorsais testadas, pode-se ainda notar uma melhor segmentação feita pelas espinhas dorsais DenseNet121 e Inceptionv3 e uma maior dificuldade de segmentação das classes pela espinha dorsal VGG16 em todos os três modelos utilizados, principalmente quando analisando-se a classe sintoma.

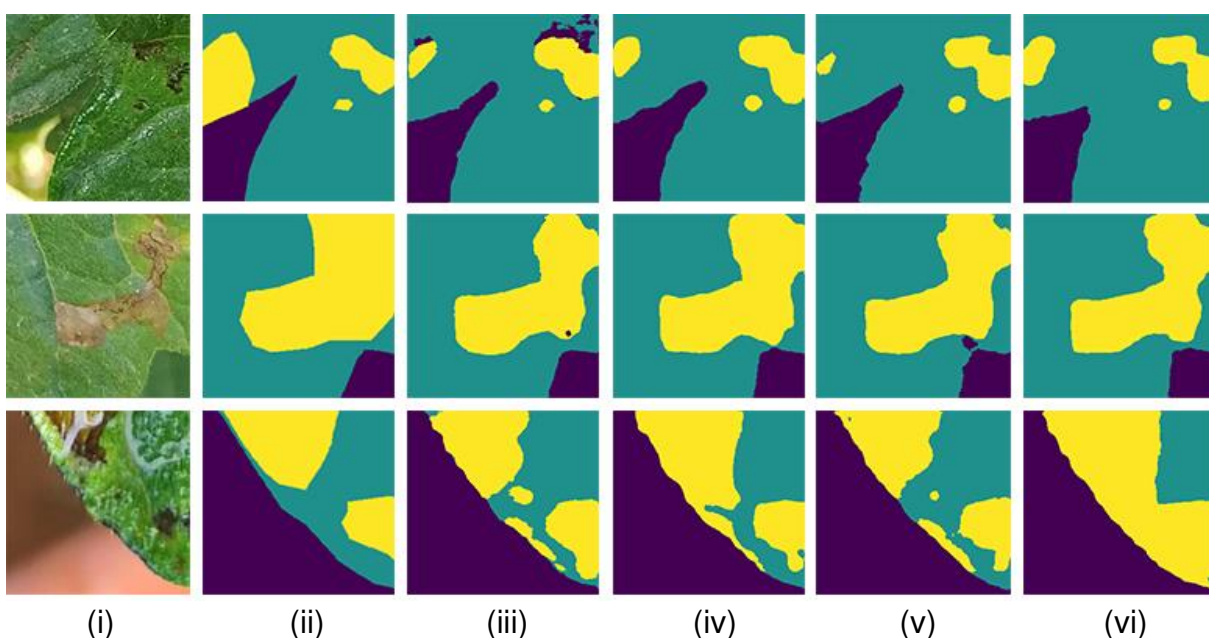


Figura 8. Comparação visual dos resultados da segmentação semântica multiclasse do modelo FPN para o banco de dados de teste. (i) Imagem original; (ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.

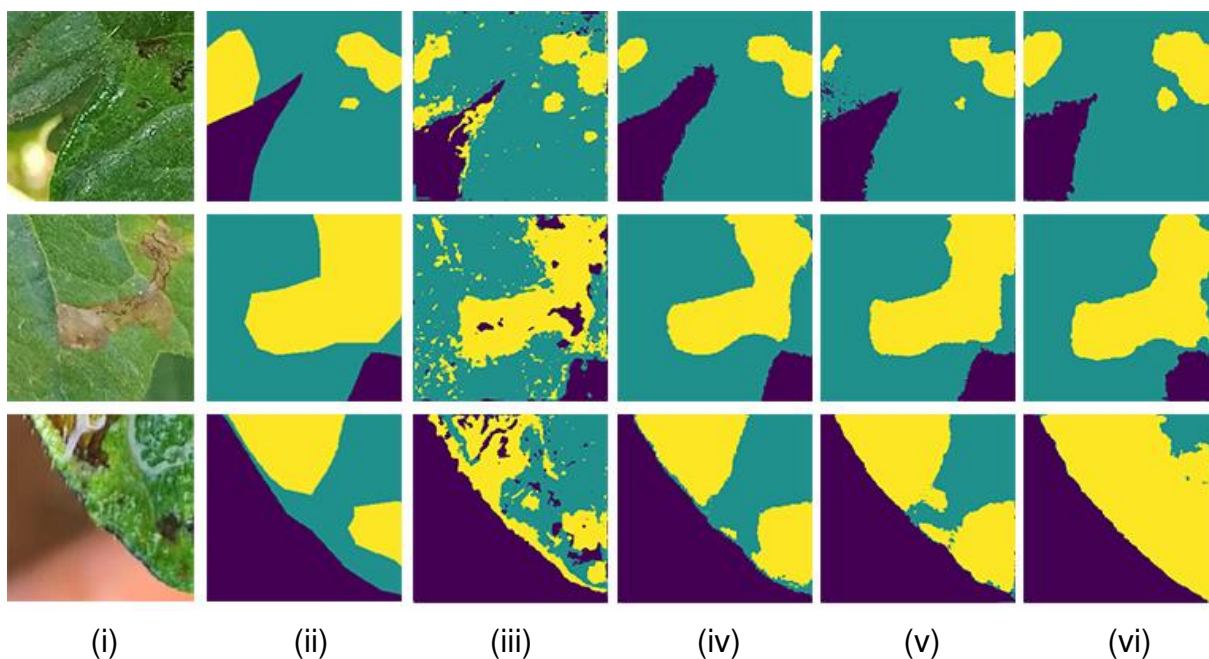


Figura 9. Comparação visual dos resultados da segmentação semântica multiclasse do modelo LinkNet para o banco de dados de teste. (i) Imagem original; (ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.

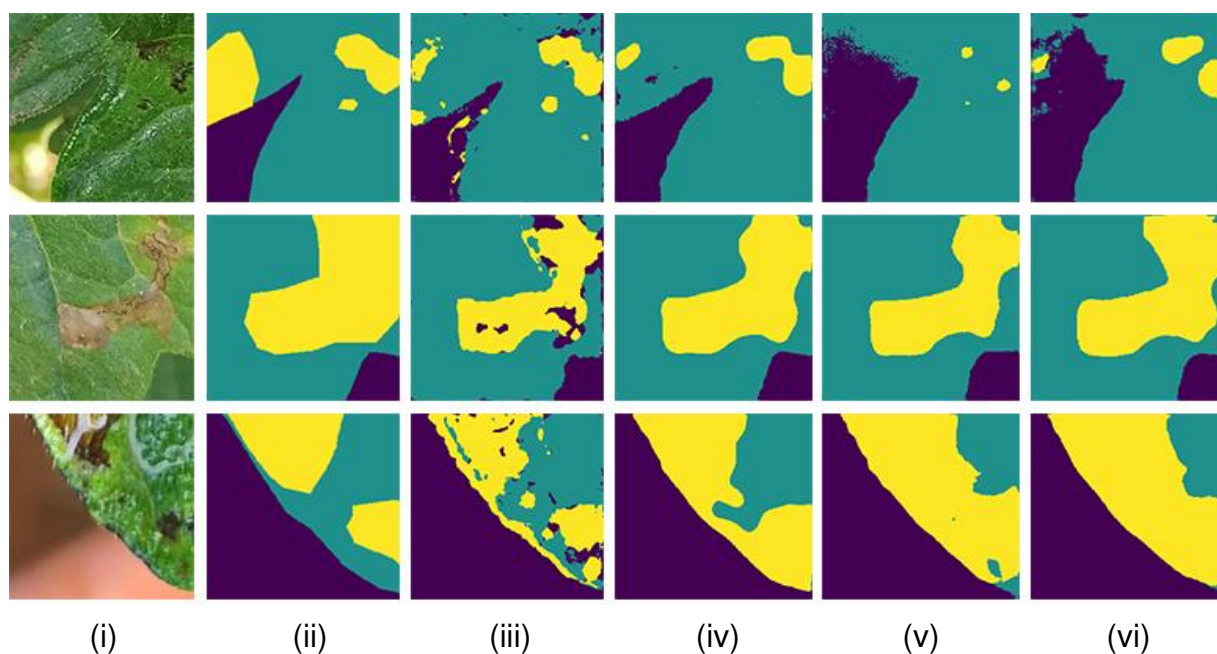


Figura 10. Comparação visual dos resultados da segmentação semântica multiclasse do modelo U-Net para o banco de dados de teste. (i) Imagem original; (ii) Anotação de referência; (iii) VGG16; (iv) DenseNet; (v) Inceptionv3; (vi) ResNet34. Fonte: Autor.

4.2 Severidade estimada pelos modelos

Para a avaliação da severidade do sintoma foliar causado pela mosca minadora na cultura do tomate, foram utilizadas as máscaras anotadas manualmente com as máscaras preditas em nível de pixel pelos modelos de segmentação U-Net, FPN e LinkNet. Fez-se a relação entre área total compreendida pelos pixels da classe folha e classe sintoma, como apresentado anteriormente na Equação 1. As Figuras 11, 12 e 13 apresentam os gráficos de dispersão de primeira ordem da estimativa da severidade pelos modelos treinados. Como resultado, os modelos FPN apresentaram melhores estimativa da severidade do sintoma foliar quando comparado à máscara de referência utilizada como dado de entrada para treinamento dos modelos, apresentando menores valores de RMSE.

Os modelos U-Net obtiveram o segundo melhor resultado dentre os modelos treinados, apresentando valores de RMSE menores quando comparado ao LinkNet. Os modelos LinkNet superestimaram a severidade do sintoma foliar, classificando erroneamente as classes, apresentando maiores valores de RMSE. Comparando-se as espinhas dorsais utilizadas em todas as arquiteturas, para a arquitetura FPN a espinha dorsal ResNet34 apresentou menor valor RMSE, porém quando comparando-se os valores de IoU para classe do sintoma, ocupou apenas a terceira melhor predição.

Para a arquitetura LinkNet, a espinha dorsal Inceptionv3 obteve melhor resultado RMSE, sendo este resultado maior que de todas espinhas dorsais dos melhores modelos treinados, utilizando a arquitetura FPN. Quando comparado o resultado RMSE dos modelos LinkNet com o resultado IoU apresentado para classe sintoma, verificou-se que a espinha dorsal Inceptionv3 ocupa o segundo melhor lugar, perdendo-se no desempenho das predições apenas para a espinha dorsal DenseNet121. Quanto a arquitetura U-Net, a espinha dorsal Inceptionv3 e ResNet34 obtiveram melhores resultados RSME, o que se confirma quando comparado aos resultados de IoU para a classe sintoma em ambos os treinamentos.

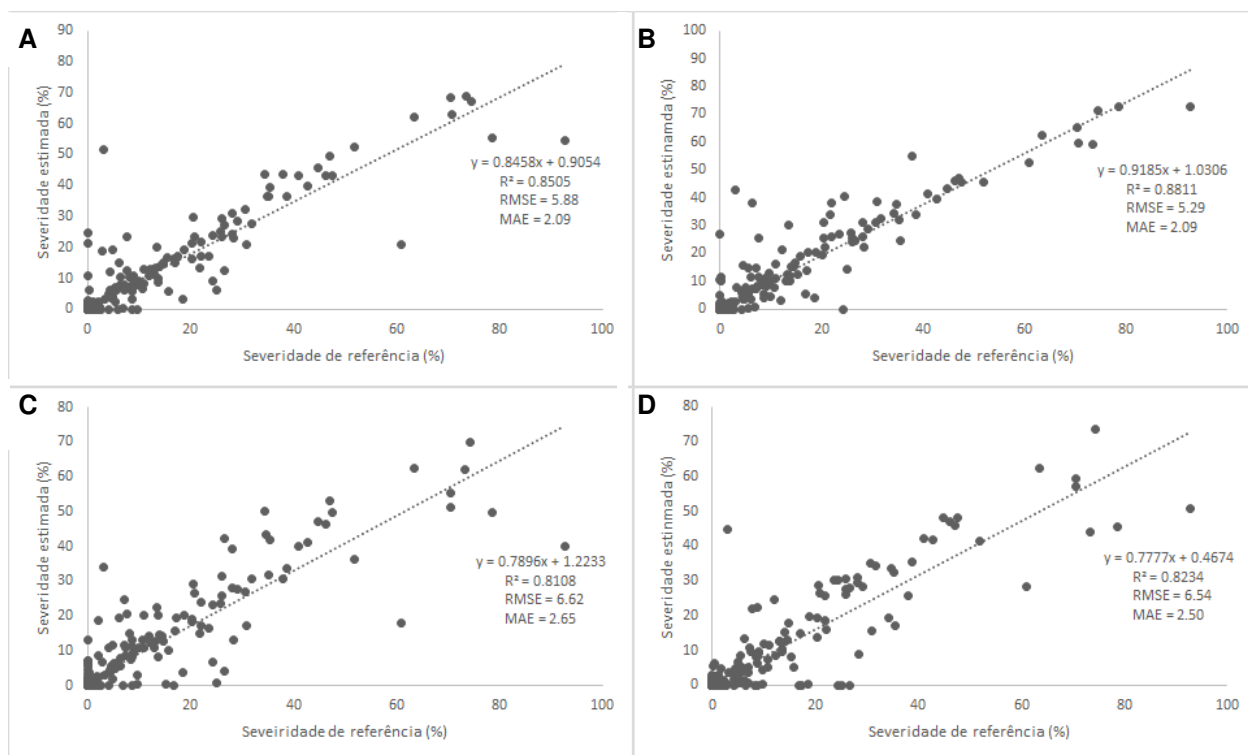


Figura 11. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura FPN e diferentes espinhas dorsais A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte: Autor.

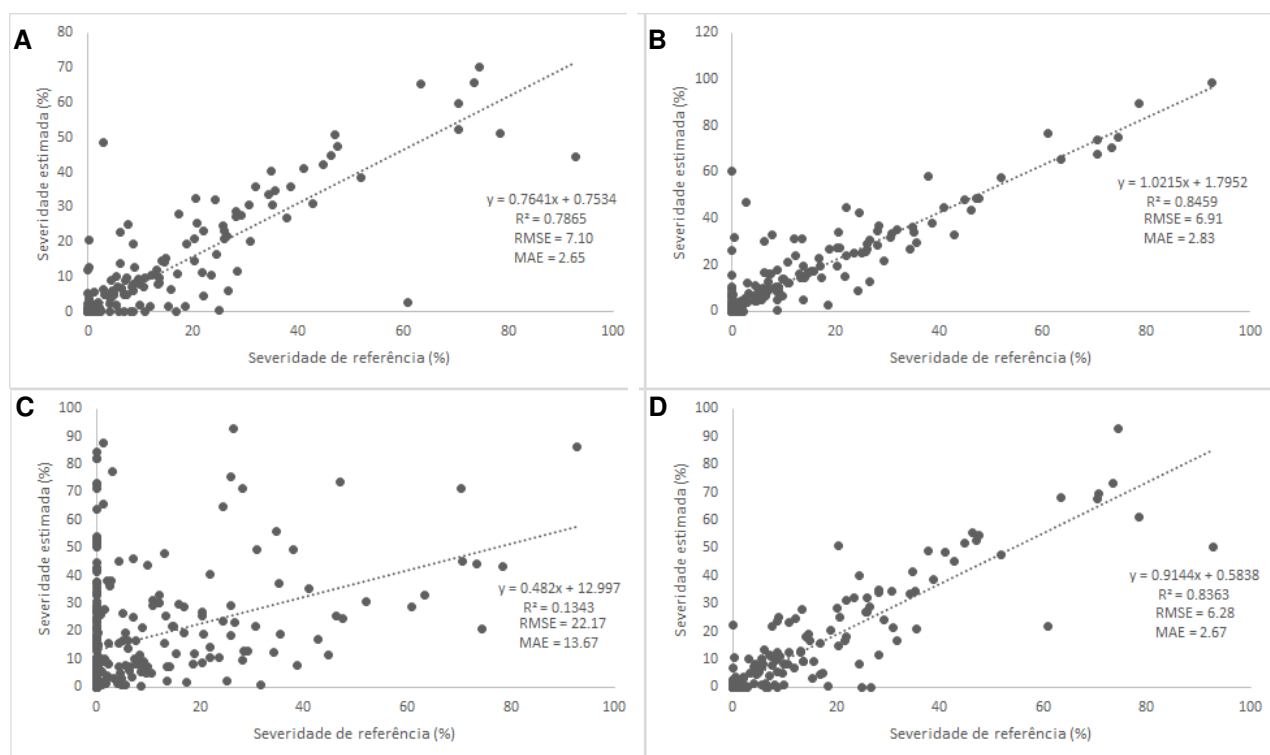


Figura 12. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura LINKNET e diferentes espinhas dorsais: A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte: Autor.

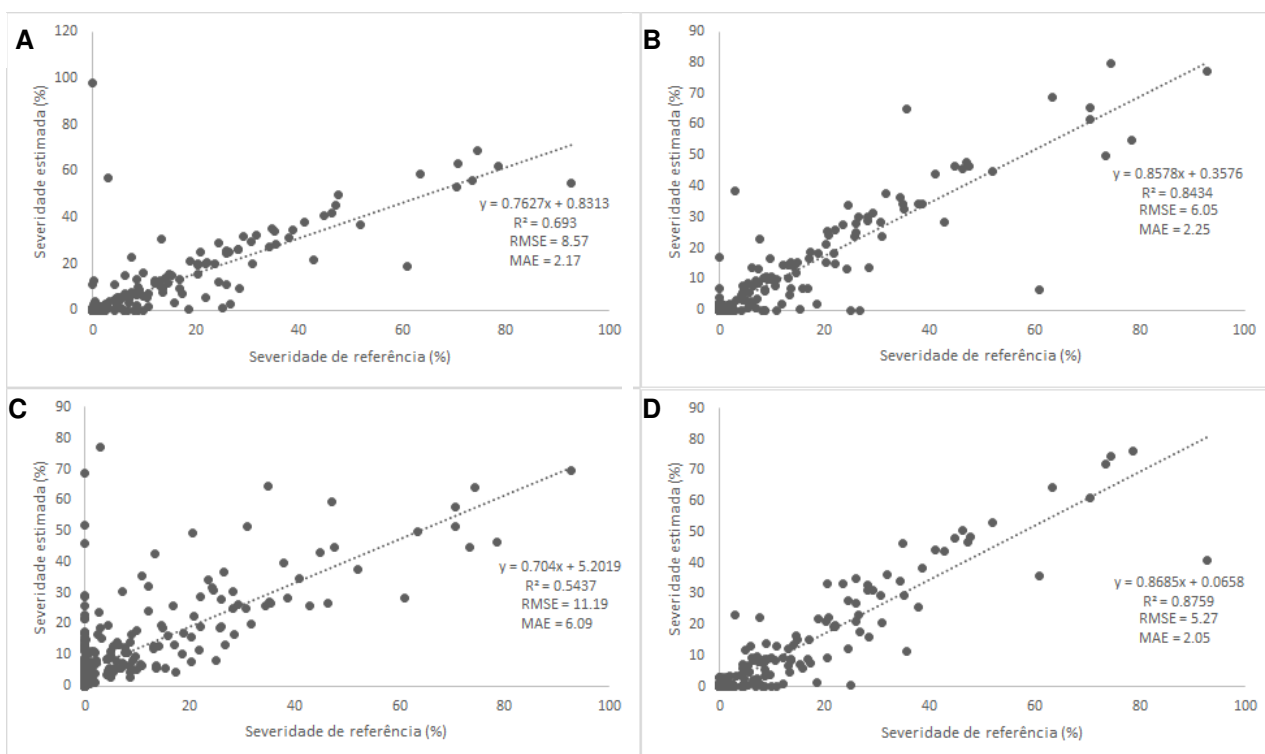


Figura 13. Gráfico de dispersão da linha de regressão de primeira ordem para relação entre severidade de referência obtida das máscaras de referência (anotadas manualmente) e a severidade estimada pela arquitetura U-NET e diferentes espinhas dorsais: A- DenseNet121; B- ResNet34; C- VGG16; D-Inceptionv3. Fonte: Autor.

4.3 Tempo de treinamento dos modelos

A Tabela 8 demonstra os tempos de treinamento para os três modelos com quatro diferentes espinhas dorsais. Pode-se observar que o tempo de treinamento do modelo FPN para o conjunto de dados de teste foi o mais longo nas quatro diferentes espinhas dorsais. Em contraste, o modelo LinkNet exigiu o menor tempo de treinamento, sendo ele com a espinha dorsal DenseNet121, o segundo modelo em performance para a classe em questão. O tempo de treinamento do modelo FPN com

espinha dorsal DenseNet121, que proporcionou a melhor performance para classe do sintoma, foi o segundo maior tempo de todos os modelos estudados.

Tabela 8. Tempo de treinamento dos modelos e diferentes espinhas dorsais

Espinhas dorsais	Modelos de aprendizado profundo		
	U-Net	LinkNet	FPN
VGG16	11m 09s	13m 06s	15m 36s
ResNet34	12m 54s	12m 53s	13m 20s
Inceptionv3	13m 08s	13m 04s	13m 29s
DenseNet121	13m 18s	10m 52s	13m 40s

5 DISCUSSÃO

Os avanços na área do aprendizado profundo de máquinas estão no desenvolvimento de sistemas automáticos mais avançados que sejam menos supervisionados ou até mesmo sem nenhuma supervisão, mais rápidos e precisos na execução de uma determinada tarefa. Diferentemente do tradicional aprendizado de máquina, os modelos de aprendizado profundo não requerem ajuste de parâmetros, nem mesmo conhecimento prévio do tipo de dado a ser trabalhado, para seleção dos extratores de características adequados que vão melhor representar as características desejáveis do alvo em estudo, o que já torna o processo mais rápido, menos supervisionado.

Na criação do banco de dados desta pesquisa, buscou-se evidenciar os desafios atuais em trabalhos de segmentação semântica, ao capturar-se as imagens diretamente no campo com plano de fundo confuso e iluminação variável, sendo assim, um método não destrutivo, não havendo necessidade de remoção da folha. Existem outros fatores decorrentes da variação da luminosidade e arquitetura da copa, que são luz especular, que é resultante do ângulo de incidência da luz na planta e ângulo de captura da câmera, além da formação de sombras e oclusão de partes da planta. Além disso, a segmentação de imagens nas condições de campo, tem-se ainda o desafio do fundo complexo, com presença de folhas e outras partes verde da planta no segundo plano da imagem, o que pode decréscimo de desempenho ao causar confusão do modelo (BARBEDO, 2016).

Para melhor comparação dos modelos e espinhas dorsais utilizados nesta pesquisa, todos os hiper parâmetros foram padronizados para todos os modelos. A

função de perda originalmente utilizada para trabalhos de segmentação semântica multiclasse é entropia cruzada categórica entre o valor previsto e o valor de referência para cada pixel (FREUDENBERG et al., 2019; WANG; ZHAO; YU, 2021). Porém, o desequilíbrio entre as classes amostradas demonstra também um desequilíbrio no aprendizado destas diferentes classes, portanto, utilizou-se a perda focal é uma entropia cruzada categórica ajustada, bastante recomendado para trabalhos de segmentação semântica, onde o banco de dados apresenta desequilíbrio entre as classes (JADON, 2020; LIN et al., 2020). É justamente um peso inserido à equação, o qual aumenta a confiança da predição para classes mais amostradas, permitindo focar o treinamento do modelo para as classes menos amostradas, ou seja, o sintoma em estudo (LIN et al., 2020).

A acurácia, precisão e revogação apresentaram valores mais altos que a IoU, o que já era esperado pois não são consideradas boas métricas para segmentação semântica, justamente por conta do desequilíbrio das classes amostradas, que é comum em trabalhos de segmentação semântica (CUI; CHEN; LU, 2020; RAHMAN; WANG, 2016). Geralmente se tem uma predominância de pixels pertencentes ao plano de fundo, ou seja, a classe mais amostrada ofuscando assim, o resultado geral. E quando analisado a predição final quase não haveriam pixels pertencentes à classe menos amostrada. A precisão das classes mais amostradas acaba sombreando a imprecisão das menores classes. Por isso, a IoU vem sendo recomendada para trabalhos de segmentação semântica, pois reflete a qualidade da segmentação em uma escala de 0 a 1, onde 1 significa que a máscara predita pelo modelo foi exatamente igual à máscara de referência, que é justamente a situação ideal que buscamos na segmentação semântica (CHEN et al., 2021). Sendo assim, a utilização do valor do IoU para as classes separadamente, permitiu confirmar a dificuldade de segmentação de classes menos amostradas, mesmo fazendo-se o uso da técnica de aumento de dados e utilizando-se o fator de ponderação γ na equação da perda focal.

Entre os modelos analisados, a FPN alcançou melhor desempenho na pontuação IoU médio e IoU separado para cada classe em questão. O modelo FPN alcançou resultado ligeiramente superior comparado ao modelo LinkNet. A FPN propõe um descendente e uma estrutura conectada horizontalmente que combina baixa resolução, recursos semânticos fortes com alta resolução e recursos semânticos fracos para detecção e segmentação de pequenos alvos (GONÇALVES et al., 2021). O que pode explicar o melhor desempenho de segmentação quando comparado a

rede LinkNet para predição dos objetos da classe de menor amostragem e de tamanho menor.

A arquitetura LinkNet obteve segundo melhor resultado na segmentação quando avaliado o IoU para classe sintoma. Ela é uma rede totalmente convolucional bastante utilizada para trabalhos de segmentação semântica, ela se concentra na predição rápida baseada na estrutura codificador-decodificador. Para que não haja perda de informações de localização do pixel na parte codificadora do modelo, a LinkNet propaga diretamente as informações espaciais do codificador para o decodificador em um mesmo nível ou patamar, podendo-se assim explicar o segundo melhor desempenho. Ainda como resultado, o tempo e as operações necessárias para reaprender os recursos perdidos são salvos, levando assim uma redução significativa no tempo de processamento.

Já a U-Net foi a rede que obteve o pior resultado no banco de dados de teste deste trabalho. A U-Net é uma rede neural totalmente convolucional que possui dois blocos e utiliza de convoluções linha a linha, o que performa igual à camada densa de uma CNN. No bloco de contração, a saída do mapa de características vai diminuindo com a profundidade da rede, o que leva uma perda de informação de pequenos alvos, o que explica seu pior desempenho comparado as outras redes utilizadas.

Analisando as espinhas dorsais das redes utilizadas, os experimentos mostraram que a espinha dorsal DenseNet121 alcança melhores resultados de IoU em todos os modelos, pois pode amostrar mais os pixels referentes a classe do sintoma. Em teoria, redes mais profundas deveriam desempenhar melhor na tarefa de segmentação semântica, mas os resultados experimentais demonstraram o contrário, quando comparando-se o número de parâmetros treináveis por modelo e espinhas dorsais (Tabela 9). Efeito este que pode ser resultado do tamanho do banco de dados utilizado neste trabalho, podendo ser insuficiente para uma rede com maior profundidade.

Tabela 9. Quantidade de parâmetros treináveis por modelo e espinha dorsais em milhões.

Espinhas dorsais	Modelos de aprendizado profundo		
	U-Net	LinkNet	FPN

	Parâmetros treináveis	Parâmetros treináveis	Parâmetros treináveis
VGG16	23,748,531	20,318,611	17,572,547
ResNet34	24,439,094	21,620,118	23,915,590
Inceptionv3	29,896,979	26,228,243	24,994,851
DenseNet121	12,059,635	8,267,411	9,828,099

Portanto, os resultados experimentais demonstraram que a DenseNet121 como espinha dorsal pode fornecer o melhor desempenho para a rede de segmentação semântica de pequenos alvos em planos de fundos complexos, com variações na iluminação e com múltiplas oclusões.

Quanto as estimativas de severidade dos modelos treinados, os modelos que apresentaram melhores estimativas de severidade foram o FPN, sendo a ResNet34 e DenseNet121 as respinhas dorsais com melhores resultados de acordo com valor de RMSE. Os modelos treinados com arquitetura U-Net seguem com a segunda melhor estimativa de severidade, sendo a espinha dorsal Inceptionv3 a que melhor estimou a severidade dos sintomas de mosca-minadora. Estes resultados podem ser confirmados através dos maiores valores de IoU encontrados tanto para os modelos FPN quanto para o U-Net. Os modelos utilizando arquitetura LinkNet superestimaram a severidade, o que pode ser verificado com os valores de IoU encontrados para as classes apresentadas e com a comparação visual feita nesta pesquisa, onde houve uma má classificação das classes nas máscaras preditas pela arquitetura. A arquitetura LinkNet foi primariamente desenvolvida para processamento de imagens aéreas com resoluções muito altas, podendo então a resolução utilizada nesta pesquisa ter sido insuficiente ocasionando este resultado inferior comparado a FPN ou até mesmo o tamanho dos alvos que são relativamente pequenos e menos amostrados.

Os resultados obtidos demonstram a importância do aprendizado por transferência no aprendizado de máquinas, mais especificamente, na segmentação semântica em um banco de dados considerado pequeno, o que é uma das limitações do emprego dos modelos de CNN em geral, justamente a falta de dados, principalmente dados anotados (BARBEDO, 2020). O processo de anotação de imagens é totalmente supervisionado, podendo ser extremamente demorado e difícil, dependendo do tamanho e quantidade de objetos a serem anotados nas imagens e do número de classes presentes. Deve-se ter atenção para um trabalho conjunto dos

pesquisadores da área na criação do banco de dados como já vem sendo feito no contexto de doenças agrícolas. A coleta de imagens no campo, as anotações prévias e compartilhamento com toda comunidade científica, o que ajudaria ainda mais a consolidar o aprendizado profundo na agricultura, permitindo o acesso comum aos dados.

Considerações para futuras pesquisas, seria a utilização de vídeos ao invés de imagens digitais e da rede CNN chamada YOLO v2 ou v3 (*You Only Look Once*), que possui vários benefícios sendo bastante utilizada para outro tipo de segmentação, que é a segmentação em instância, que combina detecção de objetos e segmentação semântica para distinguir diferentes objetos da cena, fornecendo diferentes identificadores para diferentes objetos na cena, podendo avaliar diferentes sintomas simultaneamente. Além da utilização de diferentes espinhas dorsais.

6 CONCLUSÃO

O objetivo principal deste trabalho foi estimar a severidade do sintoma da mosca-minadora na folha do tomate. Para isso, fez-se a segmentação semântica utilizando-se diferentes redes neurais convolucionais profundas e diferentes espinhas dorsais. Por meio de comparação experimental, demonstrou a rede FPN possui maior a capacidade de segmentação, apresentando melhor desempenho na predição de alvos pequenos e menos amostrados, referentes a classe do sintoma, mesmo utilizando-se um banco de dados pequeno. Ainda demonstrou a DenseNet121 sendo a melhor espinha dorsal em todos os modelos, apresentando melhor desempenho que os demais. As espinhas dorsais VGG16 e ResNet34 apresentaram os piores resultados.

Assim torna-se válida a segmentação dos sintomas com as condições complexas apresentadas nas imagens capturadas no campo como fundo complexo e iluminação irregular, podendo-se posteriormente, desenvolver sistemas para identificação automática de pragas acoplados no trator agrícola encurtando o tempo entre a detecção e a tomada de decisão pelo produtor, além de uma aplicação mais eficiente dos defensivos agrícolas.

REFERÊNCIAS

- BARBEDO, J. G. A. **A review on the main challenges in automatic plant disease identification based on visible range images.** Biosystems Engineering, v. 144, p. 52–60, 2016.
- BARBEDO, J. G. A. **Detection of nutrition deficiencies in plants using proximal images and machine learning: A review.** Computers and Electronics in Agriculture, v. 162, n. April, p. 482–492, 2019.
- BARBEDO, J. G. A. **Detecting and Classifying Pests in Crops Using Proximal Images and Machine Learning: A Review.** Ai, v. 1, n. 2, p. 312–328, 2020.
- BARROS, A. P. **BIOLOGIA DE Liriomyza sativae Blanchard , 1938 (Diptera : Agromyzidae) EM TOMATEIROS SUBMETIDOS À APLICAÇÃO DE BIOLOGIA DE Liriomyza sativae Blanchard , 1938 (Diptera : Agromyzidae) EM TOMATEIROS SUBMETIDOS À APLICAÇÃO DE.** v. 1938, 2016.
- BECKER, W. F. et al. **Sistema de produção integrada para o tomate tutorado em Santa Catarina.** [s.l: s.n.].
- BOCK, C. H. et al. **Characteristics of the perception of different severity measures of citrus canker and the relationships between the various symptom types.** Plant Disease, v. 92, n. 6, p. 927–939, 2008.
- BOCK, C. H. et al. **Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging.** Critical Reviews in Plant Sciences, v. 29, n. 2, p. 59–107, 2010.
- BOCK, C. H. et al. **From visual estimates to fully automated sensor-based measurements of plant disease severity: status and challenges for improving accuracy.** Phytopathology Research, v. 2, n. 1, 2020.
- CATAPAN, V. et al. **Principais pragas de hortaliças-fruto nas famílias das Solanáceas, Cucurbitáceas e Fabáceas.** [s.l: s.n.].
- CHAN, R. et al. **Application of maximum likelihood decision rules for handling class imbalance in semantic segmentation.** 30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM 2020, n. January, p. 3065–3072, 2020.

- CHANG, J. et al. **Deep Adaptive Image Clustering**. Proceedings of the IEEE International Conference on Computer Vision, v. 2017- Octob, p. 5880–5888, 2017.
- CHAURASIA, A.; CULURCIELLO, E. **LinkNet: Exploiting encoder representations for efficient semantic segmentation**. 2017 IEEE Visual Communications and Image Processing, VCIP 2017, v. 2018- Janua, p. 1–4, 2018.
- CHEN, J. et al. **Automatic Segmentation and Counting of Aphid Nymphs on Leaves Using Convolutional Neural Networks**. Agronomy, v. 8, n. 8, 2018.
- CHEN, S. et al. **An approach for rice bacterial leaf streak disease segmentation and disease severity estimation**. Agriculture (Switzerland), v. 11, n. 5, 2021.
- CHENG, X. et al. **Pest identification via deep residual learning in complex background**. Computers and Electronics in Agriculture, v. 141, p. 351–356, 2017.
- CHOWDHURY, M. E. H. et al. **Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques**. AgriEngineering, v. 3, n. 2, p. 294–312, 2021.
- CONAB. **Comercialização no Mercado Mundial**, Tomate: 2019.
- CUI, B.; CHEN, X.; LU, Y. **Semantic Segmentation of Remote Sensing Images Using Transfer Learning and Deep Convolutional Neural Network with Dense Connection**. IEEE Access, v. 8, p. 116744–116755, 2020.
- DAWEI, W. et al. **Recognition pest by image-based transfer learning**. Journal of the Science of Food and Agriculture, v. 99, n. 10, p. 4524–4531, 2019.
- DIVYA, B.; SANTHI, M. **SVM-based pest classification in agriculture field**. International Journal of Recent Technology and Engineering, v. 7, n. 5, p. 150–155, 2019.
- DOS SANTOS, A. A. et al. **Assessment of CNN-based methods for individual tree detection on images captured by RGB cameras attached to UAVS**. Sensors (Switzerland), v. 19, n. 16, p. 1–11, 2019.
- FAO - Food and Agriculture Organization of the United Nations. **Statistical database 2018**. FAO. 2020. Disponível em: <http://www.fao.org/faostat/en/#data/QC>. Acesso em: 17 de dezembro de 2020.
- FAO - Food and Agriculture Organization of the United Nations. **Statistical database 2021**. FAO. 2021. Disponível em: <https://www.fao.org/faostat/en/#data/QCL>. Acesso

em: 25 de janeiro de 2022.

FREUDENBERG, M. et al. **Large scale palm tree detection in high resolution satellite images using U-Net.** Remote Sensing, v. 11, n. 3, p. 1–18, 2019.

FRÖHLICH, B.; RODNER, E.; DENZLER, J. **A fast approach for pixelwise labeling of facade images.** Proceedings - International Conference on Pattern Recognition, p. 3029–3032, 2010.

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.

FUENTES, A. F. et al. **High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank.** Frontiers in Plant Science, v. 9, n. August, p. 1–15, 2018.

GILBERTSON, R. L.; BATUMAN, O. **Emerging viral and other diseases of processing tomatoes: Biology, diagnosis and management.** Acta Horticulturae, v. 971, p. 35–48, 2013.

GONÇALVES, J. P. et al. **Deep learning architectures for semantic segmentation and automatic estimation of severity of foliar symptoms caused by diseases or pests.** Biosystems Engineering, v. 210, p. 129–142, 2021.

GUSMÃO, M. R. et al. **Seletividade de inseticidas a predadores de pulgões.** Horticultura Brasileira, v. 18, n. 2, p. 130–133, 2000.

HIDAYATULOH, A.; NURSALMAN, M.; NUGRAHA, E. **Identification of Tomato Plant Diseases by Leaf Image Using Squeezenet Model.** 2018 International Conference on Information Technology Systems and Innovation, ICITSI 2018 - Proceedings, p. 199–204, 2018.

HUANG, M. et al. **Detection of insect-damaged vegetable soybeans using hyperspectral transmittance image.** Journal of Food Engineering, v. 116, n. 1, p. 45–49, 2013.

IGLOVIKOV, V.; SHVETS, A. **TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation.** 2018.

JADON, S. **A survey of loss functions for semantic segmentation.** 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational

Biology, CIBCB 2020, 2020.

JOHNSON, M. W.; WELTER, S. C.; TOSCANO, N. C.; TING, P. I.; TRUMBLE, J. T. **Reduction of tomato leaf photosynthesis rates by mining activity of *Liriomyza sativae* (Diptera:Agromyzidae)**. *Journal of Economic Entomology*, Volume 76, Issue 5, 1, Pages 1061–1063, 1983.

KARISTO, P. et al. **Ranking quantitative resistance to septoria tritici blotch in elite wheat cultivars using automated image analysis**. *Phytopathology*, v. 108, n. 5, p. 568–581, 2018.

KAZMI, W. et al. **Detecting creeping thistle in sugar beet fields using vegetation indices**. *Computers and Electronics in Agriculture*, v. 112, p. 10–19, 2015.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. *Nature*, v. 521, n. 7553, p. 436–444, 2015.

LIN, T. L.; CHANG, H. Y.; CHEN, K. H. **The pest and disease identification in the growth of sweet peppers using faster r-cnN and mask r-CNn**. *Journal of Internet Technology*, v. 21, n. 2, p. 605–614, 2020.

LIN, T. Y. et al. **Feature pyramid networks for object detection**. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, v. 2017-Janua, p. 936–944, 2017.

LIN, T. Y. et al. **Focal Loss for Dense Object Detection**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 42, n. 2, p. 318–327, 2020.

LIU, H.; LEE, S. H.; CHAHL, J. S. **An evaluation of the contribution of ultraviolet in fused multispectral images for invertebrate detection on green leaves**. *Precision Agriculture*, v. 18, n. 4, p. 667–683, 2017.

LIU, J.; WANG, X. **Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network**. *Frontiers in Plant Science*, v. 11, n. June, p. 1–12, 2020.

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.

LIU, T. et al. **Detection of aphids in wheat fields using a computer vision**

technique. Biosystems Engineering, v. 141, p. 82–93, 2016.

LLORCA, C.; YARES, M.; MADERAZO, C. **Image-Based Pest and Disease Recognition of Tomato Plants Using a Convolutional Neural Network.** E-Jikei.Org, 2018.

MA, Y. et al. **Automatic threshold method and optimal wavelength selection for insect-damaged vegetable soybean detection using hyperspectral images.** Computers and Electronics in Agriculture, v. 106, p. 102–110, 2014.

MABVAKURE, B. et al. Ongoing geographical spread of Tomato yellow leaf curl virus. Virology, v. 498, p. 257–264, 2016.

MADDEN, L.; HUGHES, G.; VAN DEN BOSCH, F. **Measuring Plant Diseases.** p. 11–31, 2007.

MAHARLOOEI, M. et al. **Detection of soybean aphids in a greenhouse using an image processing technique.** Computers and Electronics in Agriculture, v. 132, p. 63–70, 2017.

MILIOTO, A.; LOTTES, P.; STACHNISS, C. **Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs.** Proceedings - IEEE International Conference on Robotics and Automation, p. 2229–2235, 2018.

MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. **Using deep learning for image-based plant disease detection.** Frontiers in Plant Science, v. 7, n. September, p. 1–10, 2016.

MOURA, A. P. et al. **Manejo integrado de pragas do tomateiro para processamento industrial.** Embrapa, p. 1415–3033, 2014.

MUPPALA, C.; GURUVIAH, V. **Machine vision detection of pests, diseases and weeds: A review.** Journal of Phytology, v. 12, p. 9–19, 2020.

MUSTAFA, W. A.; YAZID, H. **Illumination and contrast correction strategy using bilateral filtering and binarization comparison.** Journal of Telecommunication, Electronic and Computer Engineering, v. 8, n. 1, p. 67–73, 2016.

NALAM, V.; LOUIS, J.; SHAH, J. **Plant defense against aphids, the pest extraordinaire.** Plant Science, v. 279, n. 2010, p. 96–107, 2019.

PEREZ, L.; WANG, J. **The Effectiveness of Data Augmentation in Image Classification using Deep Learning.** 2017.

PICANÇO, M. C. et al. **Effect of integrated pest management practices on tomato production and conservation of natural enemies.** *Agricultural and Forest Entomology*, v. 9, n. 4, p. 327–335, 2007.

PRATISSOLI, D. et al. **Incidência de mosca-minadora e insetos vetores em sistemas de manejo de pragas em tomateiro.** *Revista Ciencia Agronomica*, v. 46, n. 3, p. 607–614, 2015.

RAHMAN, M. A.; WANG, Y. **Optimizing intersection-over-union in deep neural networks for image segmentation.** *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 10072 LNCS, p. 234–244, 2016.

RONNEBERGER, O.; FISCHER, P.; BROX, T. **U-net: Convolutional networks for biomedical image segmentation.** In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention.* [S.l.], 2015. p. 234–241.

SCHEFFER, S. J.; HAWTHORNE, D. J. **Molecular evidence of host-associated genetic divergence in the holly leafminer *Phytomyza glabricola* (Diptera: Agromyzidae): Apparent discordance among marker systems.** *Molecular Ecology*, v. 16, n. 13, p. 2627–2637, 2007.

SEFERBEKOV, S. et al. **Feature pyramid network for multi-class land segmentation.** *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, v. 2018- June, p. 272–275, 2018.

SENA JÚNIOR, D. G. **SISTEMA DE VISÃO ARTIFICIAL PARA IDENTIFICAÇÃO DE PLANTAS DE MILHO ATACADAS PELA LAGARTA DO CARTUCHO (*Spodoptera frugiperda*).** p. 1–81, 2002.

SHIJIE, J. et al. **Automatic detection of tomato diseases and pests based on leaf images.** *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, v. 2017-Janua, p. 3507–3510, 2017.

SRIWASTWA, A. et al. **Detection of Pests Using Color Based Image Segmentation.** *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, n. Iccict, p. 1393–

1396, 2018.

SZEGEDY, C. et al. **Rethinking the Inception Architecture for Computer Vision**. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, v. 2016- Decem, p. 2818–2826, 2016.

TAKAHASHI, R.; MATSUBARA, T.; UEHARA, K. **Data Augmentation Using Random Image Cropping and Patching for Deep CNNs**. IEEE Transactions on Circuits and Systems for Video Technology, v. 30, n. 9, p. 2917–2931, 2020.

UMA RANI, R.; AMSINI, P. **Pest Identification in Leaf Images using SVM Classifier**. International Journal of Computational Intelligence and Informatics, v. 6, n. 1, p. 30–41, 2016.

WANG, C.; ZHAO, Z.; YU, Y. **Fine retinal vessel segmentation by combining Nest U-net and patch-learning**. Soft Computing, v. 25, n. 7, p. 5519–5532, 2021.

WANG, K. et al. **Mobile smart device-based vegetable disease and insect pest recognition method**. Intelligent Automation and Soft Computing, v. 19, n. 3, p. 263–273, 2013.

WANG, P. et al. **Understanding Convolution for Semantic Segmentation**. Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, v. 2018- Janua, p. 1451–1460, 2018.

WENG, W.; ZHU, X. **INet: Convolutional Networks for Biomedical Image Segmentation**. IEEE Access, v. 9, p. 16591–16603, 2021.

XIA, C. et al. **Automatic identification and counting of small size pests in greenhouse conditions with low computational cost**. Ecological Informatics, v. 29, n. P2, p. 139–146, 2015.

ZHANG, J. et al. **Monitoring plant diseases and pests through remote sensing technology: A review**. Computers and Electronics in Agriculture, v. 165, n. June, p. 104943, 2019.

ZHENG, H. et al. **Early season detection of rice plants using RGB, NIR-G-B and multispectral images from unmanned aerial vehicle (UAV)**. Computers and Electronics in Agriculture, v. 169, n. March 2019, p. 105223, 2020.

APÊNDICES

APÊNDICE A – Bibliotecas utilizadas

```

import os
import glob
import cv2
import numpy as np
import pandas as pd
import random
import keras
import tensorflow
from PIL import Image
from matplotlib import pyplot as plt
from keras.models import Model
from keras.models import load_model
from keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, concatenate
, Conv2DTranspose, BatchNormalization, Dropout, Lambda
import segmentation_models as sm
from tensorflow.keras.utils import normalize
from skimage import exposure, img_as_ubyte
from tensorflow.keras.utils import normalize
from tensorflow.keras.utils import get_file
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, EarlyStopping, CSVLogger
from keras.metrics import MeanIoU
from keras.metrics import Precision
from keras.metrics import Recall

```

APÊNDICE B – Pré-processamento dos dados

Codificar Legendas

```

labelencoder = LabelEncoder()
n, h, w = train_masks.shape
train_masks_reshaped = train_masks.reshape(-1,1)
train_masks_reshaped_encoded = labelencoder.fit_transform(train_masks_reshaped
)
train_masks_encoded_original_shape = train_masks_reshaped_encoded.reshape(n,
h, w)

```

np.unique(train_masks_encoded_original_shape)

Normalizar imagens e expandir dimensões máscaras

```

train_images = img_as_ubyte(exposure.rescale_intensity(train_images))

```

```
train_masks_input = np.expand_dims(train_masks_encoded_original_shape, axis=3)
Categorizar as máscaras
```

```
train_masks_cat = to_categorical(y_train, num_classes=n_classes)
y_train_cat = train_masks_cat.reshape((y_train.shape[0], y_train.shape[1], y_train.shape[2], n_classes))
```

Augmentation e ImageDataGenerator

```
seed=24
```

```
from keras.preprocessing.image import ImageDataGenerator
```

```
img_data_gen_args = dict(rotation_range=90,
                          width_shift_range=0.3,
                          height_shift_range=0.3,
                          #brightness_range=[0.2,0.8],
                          shear_range=0.5,
                          zoom_range=0.3,
                          horizontal_flip=True,
                          vertical_flip=True,
                          fill_mode='reflect')
```

```
mask_data_gen_args = dict(rotation_range=90,
                           width_shift_range=0.3,
                           height_shift_range=0.3,
                           #brightness_range=[0.2,0.8],
                           shear_range=0.5,
                           zoom_range=0.3,
                           horizontal_flip=True,
                           vertical_flip=True,
                           fill_mode='reflect')
```

```
image_data_generator = ImageDataGenerator(**img_data_gen_args)
image_data_generator.fit(X_train, augment=True, seed=seed)
```

```
image_generator = image_data_generator.flow(X_train, seed=seed)
valid_img_generator = image_data_generator.flow(X_val, seed=seed)
```

```
mask_data_generator = ImageDataGenerator(**mask_data_gen_args)
mask_data_generator.fit(y_train_cat, augment=True, seed=seed)
mask_generator = mask_data_generator.flow(y_train_cat, seed=seed)
valid_mask_generator = mask_data_generator.flow(y_test_cat, seed=seed)
```

```
def my_image_mask_generator(image_generator, mask_generator):
    train_generator = zip(image_generator, mask_generator)
    for (img, mask) in train_generator:
        yield (img, mask)
```

```
my_generator = my_image_mask_generator(image_generator, mask_generator)
```

```
validation_datagen = my_image_mask_generator(valid_img_generator, valid_mask_
generator)
```

APÊNDICE C – Hiperparâmetros e condições

Parâmetros utilizados

```
#Parâmetros usados em todos modelos
```

```
n_classes = 3
```

```
activation = 'softmax'
```

```
INPUT_SHAPE = (X_train.shape[1],X_train.shape[2],X_train.shape[3])
```

```
LR = 0.0001
```

```
optim = tensorflow.keras.optimizers.Adam(LR)
```

```
# Segmentation models losses
```

```
dice_loss = sm.losses.DiceLoss(class_weights = ([0.25,0.25,0.5]))
```

```
focal_loss = sm.losses.CategoricalFocalLoss()
```

```
total_loss = dice_loss + (1*focal_loss)
```

```
# Métricas
```

```
metrics = ['acc', sm.metrics.IOUScore(threshold=0.5), sm.metrics.FScore(threshold=
0.5), sm.metrics.Precision(threshold=0.5), sm.metrics.Recall(threshold=0.5)]
```

Adicionar checkpoints, EarlyStopping e CSVLogger

```
filepath=('/content/drive/MyDrive/DB_mosca_minadora/models_es_logs/UNET_dens
enet121_weights-improvement-{epoch:02d}-{val_acc:.2f}.hdf5')
```

```
checkpoint = ModelCheckpoint(filepath, monitor='val_iou_score', verbose=1, save_b
est_only=True, mode='max')
```

```
early_stop = EarlyStopping(monitor='val_iou_score', patience=5, verbose=1)
```

```
log_csv = CSVLogger ('/content/drive/MyDrive/DB_mosca_minadora/logs/UNET_den
senet121.csv', separator = ',', append=False)
```

```
callbacks_list = [checkpoint, early_stop, log_csv]
```

APENDICE D – Arquitetura e espinhas dorsais

Espinhas dorsais

```
BACKBONE = 'vgg16' #'vgg16','resnet34','inceptionv3', 'densenet121'
```

```
preprocess_input = sm.get_preprocessing(BACKBONE)
```

Definição das arquiteturas

```
model = sm.Unet(BACKBONE, input_shape=INPUT_SHAPE, encoder_weights='imagenet', classes=n_classes, activation=activation)
```

```
model_2 = sm.FPN(BACKBONE, input_shape=INPUT_SHAPE, encoder_weights='imagenet', classes=n_classes, activation=activation)
```

```
model_3 =  
sm.Linknet(BACKBONE, input_shape=INPUT_SHAPE, encoder_weights='imagenet',  
classes=n_classes, activation=activation)
```

```
model.compile(optimizer, total_loss, metrics=metrics)
```

```
print(model.summary())
```