

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO
PAULO CAMPUS SUZANO**

**FELIPE NOGUEIRA ABRANTES DE CASTRO
MATEUS NOGUEIRA ABRANTES DE CASTRO
VINÍCIUS NOGUEIRA ABRANTES DE CASTRO**

**DETECÇÃO DA ANTRACNOSE EM PLANTAÇÕES DE CAQUI EM MOGI DAS
CRUZES COM O USO DE INTELIGÊNCIA ARTIFICIAL E SENSORIAMENTO
REMOTO**

Suzano/SP
2025

Felipe Nogueira Abrantes de Castro
Mateus Nogueira Abrantes de Castro
Vinícius Nogueira Abrantes de Castro

**DETECÇÃO DA ANTRACNOSE EM PLANTAÇÕES DE CAQUI EM MOGI DAS
CRUZES COM O USO DE INTELIGÊNCIA ARTIFICIAL E SENSORIAMENTO
REMOTO**

Orientador (a): Vera Lúcia da Silva
Coorientador (a): Ana Paula Abrantes de Castro
Shiguemori

Suzano – SP
2025

Agradecimentos

Primeiramente, gostaríamos de agradecer a Deus, por estar sempre ao nosso lado nos dando forças e perseverança.

A nossa amada mãe Adriana Regina Nogueira, que esteve conosco em todos os momentos, apoiando nossas decisões e nos aconselhando sempre da melhor forma possível.

Ao nosso querido pai Renato Abrantes de Castro por nos auxiliar, dando suporte para as diversas questões no decorrer do trabalho e, aos demais familiares que acompanharam o desenvolvimento do projeto.

Vale mencionar também os amigos que estiveram ao nosso lado nos dando apoio motivacional e compartilhando suas experiências.

Ao produtor Takeshi Uyekita e à sua esposa por disponibilizar o sítio Uyekita para a coleta de imagens dos caquis e compartilharem seus saberes e, ao Sr. Yusi por mediar nosso contato com os produtores e por dedicar seu valioso tempo para nos acompanhar e auxiliar na ida à plantação.

À orientadora, Vera Lúcia da Silva, pela orientação técnica, críticas construtivas e dedicação ao acompanhamento acadêmico do trabalho. Ademais, por nos acompanhar durante todos esses anos, nos abrindo diversos caminhos e oportunidades.

À coorientadora, Ana Paula Abrantes de Castro Shiguemori, que acompanhou nosso crescimento desde cedo e influenciou positivamente em algumas de nossas escolhas.

Por fim, à instituição Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Suzano, por oferecer infraestrutura, recursos e ambiente acadêmico propício ao desenvolvimento do estudo.

Resumo

O cultivo do caqui possui grande relevância na cidade de Mogi das Cruzes, porém sua produção vem diminuindo devido aos impactos da antracnose, doença causada pelo fungo do gênero *Colletotrichum*. Com o objetivo de preservar esse símbolo cultural e a agrobiodiversidade local, este trabalho propõe o uso de sensoriamento remoto aliado a técnicas de visão computacional, a fim de identificar a doença nas plantações, aprimorar seu manejo, reduzir as perdas econômicas, diminuir o uso excessivo de defensivos agrícolas, preservar a produtividade e a qualidade dos frutos, e valorizar a produção local de caqui. Para isso, foram utilizadas imagens capturadas por um drone e transmitidas a um computador, onde são processadas por uma Inteligência Artificial (IA) desenvolvida com o modelo pré-treinado *YOLOv8*, em conjunto com a biblioteca *OpenCV* e outros recursos complementares.

Palavras-Chave: caqui; antracnose; visão computacional; inteligência artificial; sensoriamento remoto; *YOLOv8*.

Abstract

Persimmon cultivation is highly important in the city of Mogi das Cruzes, but its production has been declining due to the impacts of anthracnose, a disease caused by the fungus of the genus Colletotrichum. To preserve this cultural symbol and local agrobiodiversity, this work proposes the use of remote sensing combined with computer vision techniques to identify the disease in plantations, improve its management, reduce economic losses, reduce the excessive use of pesticides, preserve productivity and fruit quality, and enhance local persimmon production. To achieve this, images captured by a drone were transmitted to a computer, where they are processed by an Artificial Intelligence AI developed with the pre-trained YOLOv8 model, in conjunction with the OpenCV library and other complementary resources.

Keywords: *persimmon; anthracnose; computer vision; artificial intelligence; remote sensing; YOLOv8.*

SUMÁRIO

1. Introdução.....	7
2. Objetivos.....	8
2.1. Geral.....	8
2.2. Específicos.....	8
3. Fundamentação Teórica.....	8
3.1. Caqui.....	9
3.1.0. Rama Forte.....	9
3.1.1. Fuyu.....	9
3.1.2. Giombo.....	10
3.2. Antracnose.....	11
3.2.0. Ciclo da Antracnose.....	12
3.2.1. Manejo da Antracnose.....	13
3.3. Inteligência Artificial (IA).....	13
3.3.0. <i>Machine Learning</i> (ML).....	14
3.3.1. <i>Deep Learning</i> (DL).....	15
3.3.2. Redes Neurais Artificiais (RNAs).....	15
3.4. Visão Computacional.....	16
3.4.0. <i>YOLOv8</i>	18
3.4.1. <i>OpenCv</i>	18
4. Metodologia.....	19
4.1. Produção do dataset.....	19
4.2. Treinamento do modelo.....	21
4.3. Integração com o computador.....	25
5. Resultados e Discussão.....	27
5.1. Treinamento e dataset.....	27
5.2. Identificação ao vivo.....	31
6. Conclusão.....	32
7. Referências Bibliográficas.....	33

1. Introdução

O caqui é uma fruta originária da China, introduzida no Brasil durante a imigração japonesa, tendo Mogi das Cruzes como uma das principais cidades a recebê-los. O município é conhecido como a “Terra do Caqui” devido à sua relevância na produção da fruta (IBGE, 2013, apud Santos; Bello, 2014).

O fruto tornou-se um dos símbolos da cidade, com uma produção média de 20.660 toneladas entre 2016 e 2020, a maior do país nesse período. Além disso, o cultivo possui grande importância para a agricultura familiar, representando fonte de renda para diversos produtores. Em duas edições dos Jogos Abertos do Interior, o centro urbano teve como mascote o “Caquito”, escolhido por voto popular na internet, e por anos a cidade promoveu a Festa do Caqui, celebrando o início da colheita do fruto (Santos, 2022).

A cultura agrícola é tipicamente cultivada nas regiões Sul e Sudeste do Brasil, onde o clima subtropical favorece seu desenvolvimento. Contudo, pragas e doenças têm afetado significativamente os plantios, ameaçando a produção, a agrobiodiversidade e o agroturismo local. Entre as principais enfermidades, destaca-se a antracnose, causada pelo fungo do gênero *Colletotrichum*, responsável por grandes perdas nas lavouras (Oliveira; Pereira; Anzanello, 2021).

Portanto, diversos produtores rurais abandonam a produção de caqui e substituem por outras culturas agrícolas. Dessa forma, o caqui sofre um grande risco de ser extinto nas plantações de Mogi das Cruzes, como é o caso do estado do Paraná, onde o caqui Giombo está quase extinto.

Diante das perdas significativas causadas pela antracnose nas plantações de caqui, surge a necessidade de investigar métodos que possam minimizar esse problema. Nesse contexto, o presente estudo propõe o desenvolvimento de um modelo de Inteligência Artificial (IA), fundamentado em técnicas de *Machine Learning* (ML) e *Deep Learning* (DL), associado ao uso de visão computacional e de sensoriamento remoto. O objetivo é possibilitar o reconhecimento automático de padrões visuais da antracnose nos frutos, promovendo o diagnóstico precoce e fornecendo suporte ao manejo agrícola, de modo a contribuir para a manutenção da produtividade e a sustentabilidade do cultivo.

2. Objetivos

2.1. Geral

Aplicar técnicas de inteligência artificial (IA) e visão computacional para o reconhecimento da antracnose em caquizeiros, através do uso de sensoriamento remoto, a fim de aprimorar o monitoramento da doença em plantações de caqui em Mogi das Cruzes e manter a agrobiodiversidade no município.

2.2. Específicos

Entre os objetivos específicos, estão:

- Investigar as principais doenças que atingem cultivos de caqui, com ênfase na antracnose;
- Avaliar os impactos da antracnose em plantações do município de Mogi das Cruzes;
- Reconhecer e capturar imagens da Antracnose em caquis por meio da pesquisa de campo;
- Estudar técnicas de visão computacional e sensoriamento remoto aplicadas à agricultura;
- Selecionar e aplicar métodos de classificação e detecção de objetos utilizando Deep Learning (DL);
- Aplicar a IA para a detecção automática da Antracnose em caquis; e
- Identificar com mais eficácia cultivos contaminados com a Antracnose, utilizando imagens de drone em tempo real.

3. Fundamentação Teórica

Esta seção tem como objetivo apresentar uma visão geral dos principais conceitos que fundamentam o trabalho. Inicialmente, será feita uma contextualização sobre os tipos de caqui cultivados em Mogi das Cruzes, ressaltando os impactos da antracnose no cultivo da fruta, além do ciclo da doença e seu manejo. Em seguida, serão abordados os fundamentos da Inteligência Artificial (IA), com ênfase no *Machine Learning* (ML) e no *Deep Learning* (DL), bem como o funcionamento das Redes Neurais Artificiais (RNAs), que são a base para a maioria dos sistemas modernos de aprendizado de máquina.

No campo da Visão Computacional, serão discutidas as técnicas que permitem a interpretação automática de imagens, destacando a utilização do modelo *YOLOv8*. Também será apresentada a biblioteca *OpenCV*.

3.1. Caqui

A fruta destaca-se pelo seu elevado teor de polifenóis e fibras alimentares, que atuam como antioxidantes e auxiliam na regulação do metabolismo lipídico. Esses compostos contribuem para reduzir o colesterol e melhorar o equilíbrio metabólico, além de favorecer a digestão e a saúde intestinal. Assim, seu consumo regular pode ser considerado uma estratégia nutricional importante para a promoção da saúde (Gorinstein et al., 1999).

Tendo isso em vista, no decorrer do projeto foram feitas entrevistas com produtores rurais, com o objetivo de compreender melhor o funcionamento do cultivo da fruta, seus diferentes tipos e, principalmente, os impactos causados pela antracnose na produção, considerando a influência benéfica da fruta para a saúde. Assim, o estudo sobre seus devidos tipos presentes nas plantações da cidade foi essencial para entender a dinâmica da doença e seus impactos econômicos.

3.1.0. Rama Forte

O Rama Forte sempre teve destaque no plantio da cidade, sendo a variedade mais produzida pelos agricultores locais. Entretanto, nos últimos anos, seu preço tem passado por um processo de desvalorização, situando-se atualmente entre R\$2,00/kg e R\$3,00/kg, segundo entrevistas com produtores rurais, o que tem tornado seu cultivo economicamente inviável.

Embora a antracnose também acometa o Rama Forte, o controle da doença nessa variedade é mais simples em comparação com outros tipos de caqui, não provocando grandes perdas na produção.



Figura 1 - Rama Forte: Mole, achatado e vermelho

Fonte: Mollicenter, 2023.

3.1.1. Fuyu

O cultivo do caqui Fuyu demanda um manejo mais complexo e planejado, incluindo práticas como o ensacamento dos frutos, que contribuem tanto para a melhoria do sabor

quanto para a proteção contra pragas, especialmente moscas, em um processo semelhante ao utilizado na produção da goiaba. Apesar de apresentar maior resistência à antracnose, essa variedade também pode ser afetada pela doença, ainda que em menor intensidade quando comparada a outros tipos de caqui.



Figura 2 - Fuyu: Crocante, Achatado e Alaranjado

Fonte: Molicenter, 2023.

3.1.2. Giombo

Essa variedade de caqui é a mais suscetível à antracnose, sendo frequentemente afetada a ponto de ocasionar o secamento das árvores e torná-las improdutivas, o que leva, em muitos casos, ao corte dos pés. Tal vulnerabilidade representa a principal razão pela qual diversos produtores rurais abandonam o cultivo dessa espécie, uma vez que a doença acarreta elevados prejuízos econômicos e apresenta grande dificuldade de manejo.

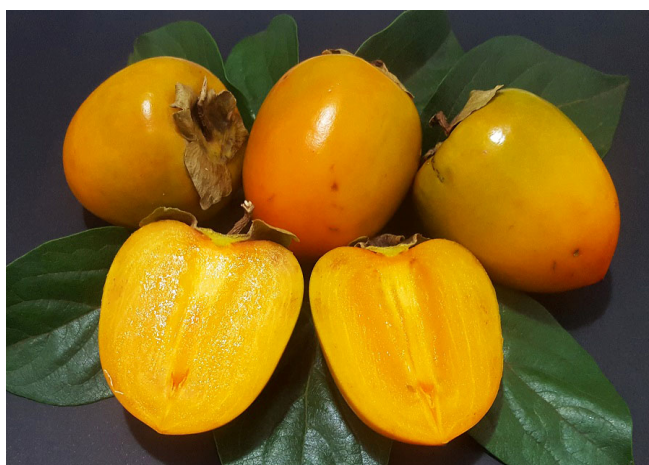


Figura 3 - Giombo: Crocante, ovóide e alaranjado

Fonte: Ciprest, 2017.

Dessa forma, observa-se que existem diferenças econômicas significativas entre os diversos tipos de caqui, bem como variações na suscetibilidade de cada variedade à antracnose. Essa realidade evidencia a necessidade de atenção especial ao caqui Giombo, que

se apresenta como o mais afetado pela doença e, conseqüentemente, o que mais demanda estratégias de manejo e monitoramento.

3.2. Antracnose

A antracnose, causada pelo fungo do gênero *Colletotrichum*, é a principal doença que afeta o cultivo de caqui no Brasil, podendo ocasionar perdas de até 100% da produção e sendo responsável pelo declínio ou até extinção de plantações (Blood et al., 2015; Paludo, 2015 apud Oliveira; Pereira; Anzanello, 2021).

A enfermidade acomete ramos, folhas e frutos, provocando manchas escuras e deprimidas, expostos na Figura 4, queda prematura, amolecimento dos frutos, seca e, em casos mais graves, a morte dos ramos, funcionando como inóculo primário e possibilitando a disseminação da doença em outras áreas da plantação.



Figura 4 - Sintomas da antracnose nos ramos (A e B), nas folhas (C e D), na superfície dos frutos (E, F, G) e na polpa dos frutos (H) de caquizeiro.

Fonte: Oliveira; Pereira; Anzanello, 2021.

Além disso, o fungo pode permanecer incubado na lavoura, principalmente durante o inverno, representando um problema não apenas em condições de campo, mas também em pós-colheita, armazenamento refrigerado prolongado e transporte dos frutos (Palou et al., 2015 apud Oliveira; Pereira; Anzanello, 2021).

3.2.0. Ciclo da Antracnose

Na fase de inóculo primário, o fungo permanece no solo ou em restos vegetais na forma de micélio e microescleródios, podendo ser disseminado por respingos de chuva, ventos, insetos, sementes e mudas contaminadas, dando início a um novo ciclo na primavera. Nessa etapa, o patógeno forma estruturas reprodutivas, desenvolvendo conidióforos que produzem conídios (esporos assexuais) em grande quantidade, causadores do rompimento da epiderme do fruto e liberando os esporos no ambiente.

A transmissão da antracnose é favorecida por altos níveis de umidade e temperaturas entre 20 e 25 °C, especialmente durante a noite, além de períodos prolongados de chuva que mantêm os órgãos aéreos da planta molhados, ataques de pragas e práticas de adubação inadequadas. Estudos realizados por May de Mio et al. (2019) demonstraram que a infecção do patógeno em frutos maduros é mais eficiente a 25 °C, embora possa ocorrer entre 10 e 30 °C, sendo a esporulação favorecida a 20, 25 e 30 °C (Oliveira; Pereira; Anzanello, 2021).

A infecção inicial ocorre quando o conídio germina sobre o fruto, o que forma um apressório, estrutura especializada responsável pela penetração no tecido do hospedeiro. Na fase inicial, o fungo desenvolve-se de forma biotrófica, estabelecendo-se no hospedeiro sem causar sintomas visíveis. Posteriormente, ocorre a transição para a fase necrotrófica, na qual o patógeno provoca a morte dos tecidos e passa a extrair nutrientes, resultando nos sintomas característicos da doença.

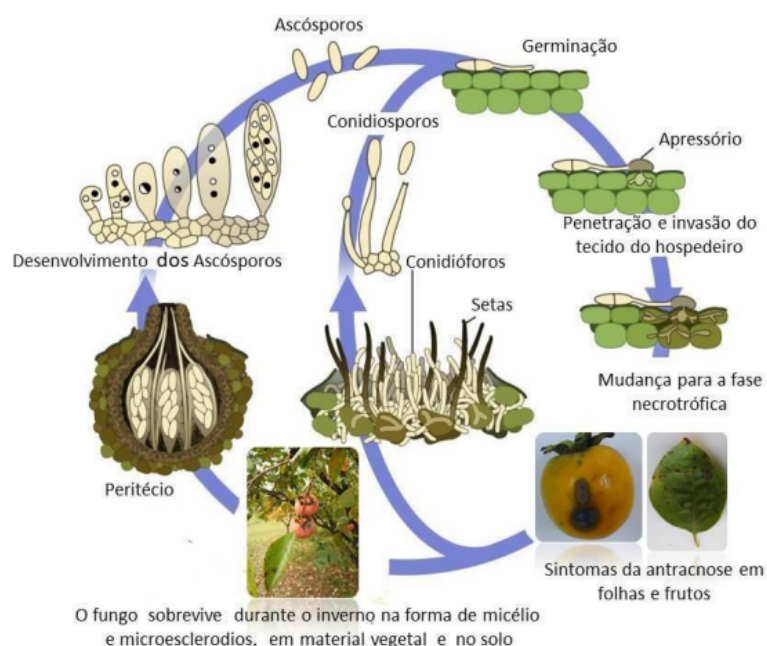


Figura 5 - Ciclo de vida de *Colletotrichum spp.*

Fonte: Silva et al., 2017 apud Oliveira; Pereira; Anzanello, 2021.

3.2.1. Manejo da Antracnose

Para a prevenção da antracnose, diversas práticas culturais podem ser adotadas, visando reduzir a incidência da doença e minimizar perdas na produção. Entre elas, destaca-se a realização de poda verde de ramos fracos, “ladrões” e doentes durante o período vegetativo, com o objetivo de melhorar a aeração e a entrada de luz no dossel das plantas. Além disso, recomenda-se a redução do porte das árvores, bem como a retirada de frutos mumificados, doentes ou caídos ao solo, enterrando-os quando possível (Lourenço Jr.; Leite Jr.; May de Mio, 2014 apud Oliveira; Pereira; Anzanello, 2021).

O tratamento dos cortes resultantes da poda de inverno também é fundamental, podendo ser realizado com a aplicação de pasta bordalesa ou cúprica. Complementarmente, a pulverização das plantas com calda sulfocálcica ou bordalesa após a poda contribui para a prevenção da doença. Outro ponto importante é a desinfestação de ferramentas utilizadas nos tratamentos culturais, feita por imersão em soluções de hipoclorito de sódio ou água sanitária, evitando a disseminação do patógeno entre as plantas.

A escolha de áreas de plantio sem histórico da doença, assim como o uso de cultivares mais resistentes, como Kyoto, Mikado e Yamamoto, também auxilia na redução da incidência da antracnose (Lourenço Jr.; Leite Jr.; May de Mio, 2014 apud Oliveira; Pereira; Anzanello, 2021).

O manejo nutricional da cultura desempenha papel relevante na prevenção da doença. A correção adequada do solo antes do plantio garante que a planta receba os nutrientes necessários para um desenvolvimento equilibrado, enquanto o excesso de adubação nitrogenada deve ser evitado, pois pode favorecer a proliferação do fungo.

Além das práticas culturais e nutricionais, o controle químico continua sendo uma ferramenta importante no manejo da doença. Fungicidas à base de cobre, azoxistrobina ou difenoconazol podem ser utilizados, sendo essencial realizar a rotação de grupos químicos para evitar o surgimento de resistência por parte do patógeno.

Por fim, o controle biológico surge como uma alternativa sustentável, com o uso de microrganismos como *Trichoderma spp.* e *Bacillus spp.*, que contribuem para a redução da queda de frutos e para o aumento da produtividade, reforçando a importância de um manejo integrado da cultura. Portanto, é possível controlar a antracnose em plantações de caqui.

3.3. Inteligência Artificial (IA)

A inteligência artificial (IA) constitui um campo da ciência da computação voltado ao desenvolvimento de sistemas capazes de reproduzir aspectos do raciocínio humano. O

objetivo é criar máquinas e programas que possam executar funções tradicionalmente relacionadas com a inteligência de pessoas, como identificar padrões de fala, interpretar informações visuais, realizar escolhas diante de diferentes cenários e até mesmo traduzir automaticamente conteúdos entre idiomas distintos (Siegel et al., 2018, apud Almeida, 2019).

A proposta central dessa área é dotar computadores de recursos que simulam processos cognitivos, permitindo que se tornem ferramentas mais autônomas e eficazes em diversas aplicações.

A inteligência artificial é uma noção geral que abrange diversos conceitos. As principais divisões dela estão representadas na Figura 6.

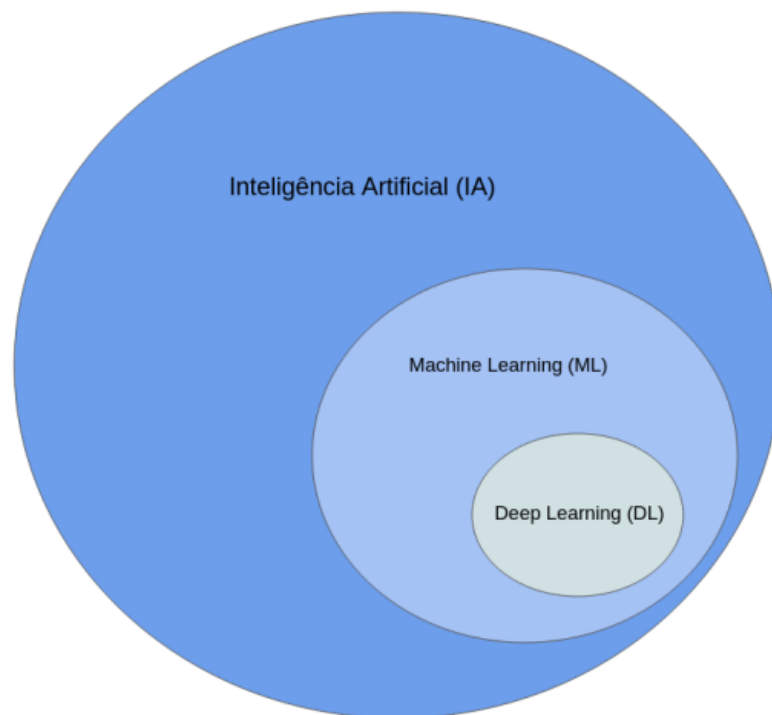


Figura 6 - Distinção entre: AI, ML e DL
Fonte: Silva; Oliveira, 2020.

3.3.0. *Machine Learning* (ML)

A aprendizagem de máquina (ML) é uma área da inteligência artificial que permite que sistemas aprendam padrões a partir de dados e façam previsões sem programação explícita. Ela pode ser supervisionada (com dados rotulados), não supervisionada (sem rótulos) ou por reforço (aprendendo com a interação com o ambiente). A abordagem probabilística é central, permitindo que os modelos enfrentem incertezas e façam previsões informadas (Bishop, 2006, apud Almeida, 2019).

O principal desafio da ML é a generalização, ou seja, a capacidade do modelo de aplicar o que aprendeu a dados não vistos, evitando o *overfitting* (ajuste excessivo aos dados de treinamento).

3.3.1. *Deep Learning* (DL)

O *Deep learning* é uma área do machine learning que utiliza redes neurais com múltiplas camadas para aprender representações de dados. Essa abordagem permite que o sistema identifique automaticamente padrões e características a partir de grandes volumes de informação, sem a necessidade de intervenção humana na definição de regras. É especialmente eficaz em tarefas complexas como reconhecimento de imagens, tradução automática e processamento de linguagem natural (Arroyo-Figueroa et al., 2000, apud Almeida, 2019).

O *deep learning* aprende diretamente com os dados por meio de treinamento e ajuste automático de parâmetros. Isso o torna mais flexível e escalável em contextos onde o volume de dados é alto e o comportamento do sistema é difícil de modelar com regras fixas.

3.3.2. Redes Neurais Artificiais (RNAs)

As redes neurais artificiais são modelos computacionais inspirados no funcionamento do cérebro humano. Assim como os neurônios biológicos processam e transmitem informações por meio de conexões sinápticas, as RNAs utilizam unidades de processamento, denominadas neurônios artificiais, que possuem função semelhante aos neurônios biológicos, interligadas por pesos sinápticos que determinam a intensidade da conexão entre elas. Essa estrutura permite que o sistema seja capaz de processar sinais de entrada e gerar respostas de saída de forma semelhante ao processo cognitivo humano (Silva; Spatti; Flauzino, 2016).

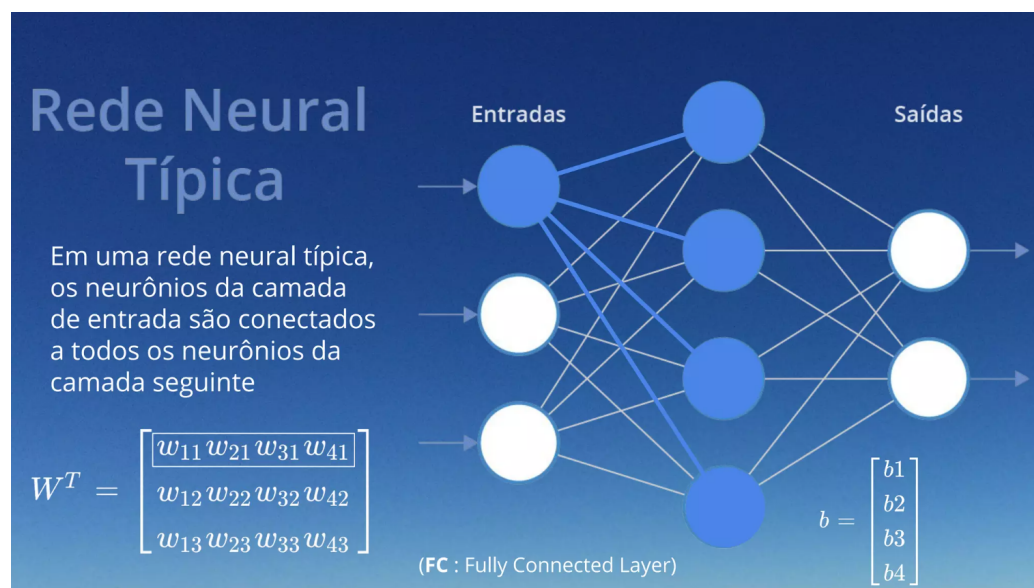


Figura 7 - Rede Neural Típica

Fonte: Menezes, 2018, apud Almeida, 2019.

O aprendizado em uma rede neural ocorre pelo ajuste desses pesos, de modo que o modelo consiga melhorar seu desempenho à medida que é exposto a novas situações. Esse processo geralmente é realizado por algoritmos de treinamento. Dessa forma, a rede neural não depende de um modelo matemático explícito do problema, mas sim da experiência adquirida a partir dos dados, possibilitando a generalização para situações ainda não vistas.

As RNAs são aplicadas em problemas que envolvem reconhecimento de padrões, classificação e previsão, como no reconhecimento de imagens, no diagnóstico médico, em sistemas de controle e na análise de séries temporais. Sua principal vantagem é a capacidade de lidar com problemas complexos e não lineares.

Portanto, as redes neurais artificiais são sistemas de processamento da informação que buscam simular a inteligência humana de maneira simplificada. Sua habilidade de aprender com exemplos e de generalizar soluções torna essa tecnologia uma ferramenta poderosa para diversas áreas da engenharia e das ciências aplicadas.

3.4. Visão Computacional

A visão computacional (VC) é uma área da inteligência artificial que ajuda os computadores a entenderem imagens ou dados complexos, como em sistemas de reconhecimento facial. O objetivo da VC é fazer com que as máquinas consigam “ver” e identificar objetos através de uma câmera, assim como os seres humanos fazem.

Para isso, a visão computacional analisa cada pixel da imagem, transformando as cores em números. Depois, ela procura padrões nesses números, como cores e texturas parecidas, para identificar e diferenciar os objetos na imagem. Essa técnica é usada em várias áreas, como segurança, indústria e saúde (Marengoni; Stringhini, 2009; Backes; Junior, 2019, apud Almeida, 2019).

Em relação ao uso de visão computacional é justo afirmar que:

A visão computacional é uma técnica de Inteligência Artificial que permite que os computadores processem, analisem e entendam imagens e vídeos de diversas áreas. No contexto do agronegócio, essa área de IA pode ser usada para a identificação de doenças e pragas em plantas, previsão das condições climáticas, análise da qualidade do solo, automações na colheita, entre outras aplicações (Alves; Lacerda; Piantoni, 2023, p.4).

Desta forma, a utilização da visão computacional é benéfica para as atividades rurais e deve ser estimulada e explorada de maneira mais extensiva, tendo em vista sua capacidade de lidar com diversas situações e possíveis problemas no campo.

Existem trabalhos que utilizam a visão computacional e a aprendizagem profunda para a detecção e contagem automática de *Euschistus heros* (Percevejo marrom) em lavouras de soja, mostrando a eficácia dessas técnicas na agricultura (Lima, 2024). De forma semelhante,

Vasconcelos (2019) aplicou a visão computacional e drones autônomos para identificar o bicho mineiro em plantações de café, novamente evidenciando o potencial dessas tecnologias para o monitoramento agrícola.

A Figura 8 apresenta o trabalho realizado por Lima (2024), já a Figura 9 o trabalho de Vasconcelos (2019).

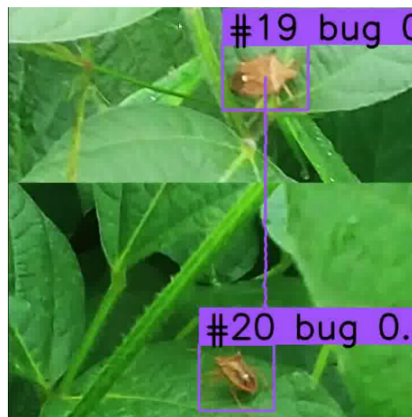


Figura 8 - Dois PMSs visíveis na mesma cena.
Fonte: Lima, 2024

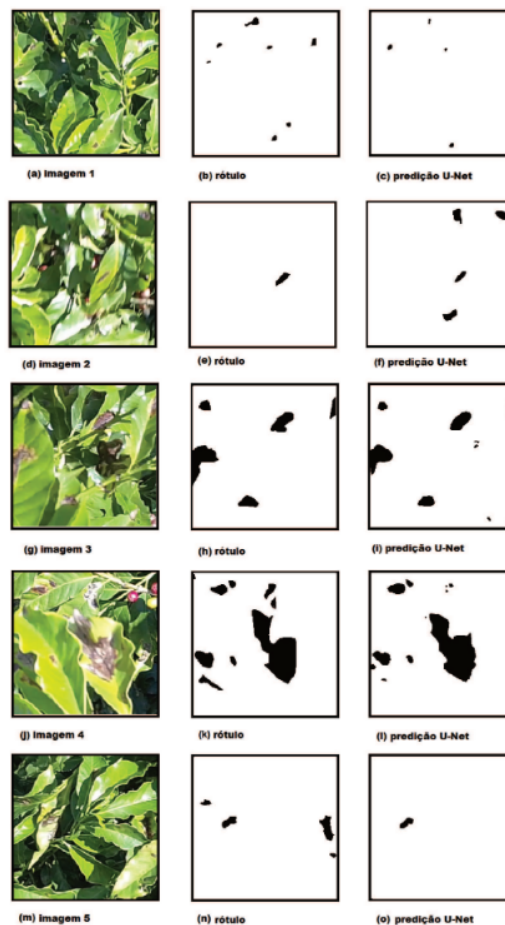


Figura 9 - Resultados da arquitetura U-Net com diferentes níveis de dificuldade.
Fonte: Vasconcelos, 2019.

Embora o uso de drones automáticos facilite a captura de imagens, muitas vezes a aplicação prática desses métodos depende de recursos disponíveis, como drones manuais, que também podem ser usados para coleta de dados visuais. Essas abordagens demonstram como a visão computacional e o aprendizado profundo são ferramentas importantes para o controle e manejo de pragas nas plantações.

3.4.0. YOLOv8

O *YOLOv8* é uma versão dos mais diversos modelos de redes neurais *YOLO* (*You Only Look Once*), criada pela equipe da *Ultralytics*, que também desenvolveu o *YOLOv5*. Lançado em 10 de janeiro de 2023, proporcionou melhorias importantes em precisão, velocidade e facilidade de uso para tarefas como detecção de objetos, classificação de imagens e segmentação de partes das imagens (Solawezt; Francesco, 2024).

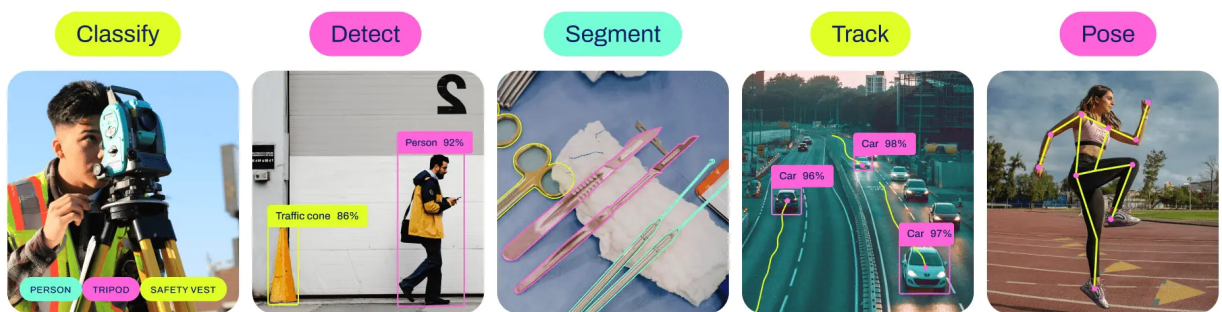


Figura 10 - Exemplos de detecção realizadas pela *YOLOv8*.

Fonte: Ultralytics (2023).

Uma das vantagens do *YOLOv8* é sua arquitetura moderna e eficiente, que torna o modelo mais rápido e preciso comparado às versões anteriores. Além disso, é um modelo “sem âncoras”, o que significa que prevê diretamente o centro dos objetos, sem precisar de caixas pré-definidas. Isso facilita o treinamento e melhora os resultados na detecção.

Testes feitos no conjunto de dados *Roboflow 100* mostraram que o *YOLOv8* é mais eficiente que o *YOLOv5*, tornando-se uma ótima escolha para projetos de visão computacional.

3.4.1. OpenCv

A *OpenCV* (*Open Source Computer Vision Library*) é uma biblioteca de código aberto muito usada em visão computacional e aprendizado de máquina. Oferece ferramentas para trabalhar com imagens e vídeos em tempo real, permitindo realizar tarefas como detecção de objetos, reconhecimento facial, segmentação de imagens e rastreamento de movimentos. Com

mais de 2.500 algoritmos disponíveis, a *OpenCV* é uma das bibliotecas mais populares nessa área, sendo utilizada tanto em pesquisas quanto em projetos comerciais (Vina, 2024).

Uma grande vantagem do *OpenCV* é seu funcionamento em várias linguagens de programação, como *Python*, *C++*, *Java* e *MATLAB*, além de diferentes sistemas operacionais, incluindo *Windows*, *Linux*, *Android* e *macOS*. Isso facilita a integração da biblioteca em diversos tipos de projetos, desde aplicativos simples até sistemas industriais complexos, permitindo que tanto iniciantes quanto profissionais aproveitem seus recursos.

Ademais, os componentes da *OpenCV* podem ser usados de forma separada ou combinada. Ele cobre áreas como processamento de imagens, aprendizado de máquina e criação de interfaces gráficas, permitindo que os desenvolvedores adaptem os algoritmos às necessidades específicas de cada projeto.

Com base nisso, a biblioteca mencionada foi escolhida para realizar as tarefas necessárias para o projeto.

4. Metodologia

O desenvolvimento do trabalho foi estruturado em três etapas principais: produção do *dataset*, treinamento do modelo e integração com o computador.

Nesta seção, serão detalhadas essas etapas, apresentando as tecnologias empregadas e os métodos adotados em cada fase do projeto.

4.1. Produção do *dataset*

Um *dataset* é um conjunto de dados estruturados que serve como base para o treinamento e a validação de modelos de Inteligência Artificial. Em projetos de Visão Computacional, ele geralmente é composto por imagens ou vídeos rotulados, permitindo que o algoritmo aprenda a reconhecer padrões, diferenciar classes e realizar previsões de forma precisa. A criação de *datasets* próprios é fundamental quando não existem conjuntos de dados disponíveis ao público que atendam às necessidades do projeto, garantindo maior flexibilidade e qualidade nos resultados.

Dessa forma, a criação de um *dataset* exclusivo mostrou-se essencial para garantir maior versatilidade no desenvolvimento da rede, já que não foram encontrados conjuntos de dados prontos que atendessem às necessidades específicas do projeto.

A coleta de dados para este trabalho foi realizada por meio de uma visita a uma plantação de caqui localizada em Mogi das Cruzes, resultando em aproximadamente 1.500 imagens, sendo que aproximadamente 1000 delas foram utilizadas na construção do *dataset* próprio.



Figura 11 - Imagens coletadas nas visitas

Fonte: Autoria Própria.

Após a captura das fotos, as imagens foram rotuladas na plataforma online *Roboflow*, especializada no apoio a projetos de Inteligência Artificial. Nesse processo, os registros foram classificados em dois grupos: caquis saudáveis e caquis doentes (afetados pela antracnose), de modo a evitar conflitos nas predições e tornar o modelo mais funcional. Assim, consolidou-se um *dataset* personalizado para o treinamento de uma rede de detecção de objetos.

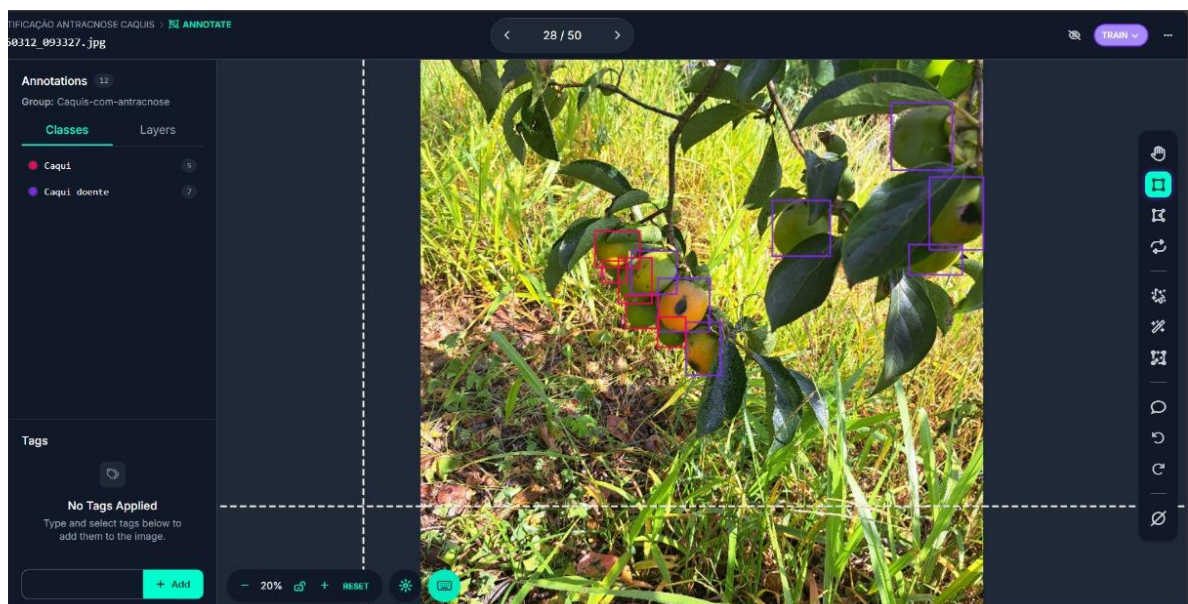


Figura 12 - Plataforma de anotação *Roboflow*

Fonte: Autoria Própria.

Além do *Roboflow*, também foram avaliados outros ambientes para anotação das imagens, como o *Edge Impulse* e o *Label Studio*. No entanto, a adaptação a essas ferramentas se mostrou um problema devido a limitações dessas plataformas.

Em contrapartida, o *Roboflow* demonstrou-se mais versátil e adequado às necessidades do projeto: sendo online, permite que vários usuários trabalhem simultaneamente na anotação das imagens, além de oferecer exportação direta no formato compatível com o *YOLOv8* e possuir uma interface intuitiva e de fácil acesso. Dessa forma, a escolha pelo *Roboflow* contribuiu para a agilidade e eficiência na construção do *dataset*.

O *dataset* foi dividido em torno de 70% para o conjunto de treinamento e 30% para o conjunto de validação, valor padrão em estudos de inteligência artificial. Ademais, foram utilizados alguns dos filtros para as imagens disponíveis na plataforma, a fim de melhorar o desempenho do treinamento. Entre os filtros utilizados estão: Ajuste de orientação, ajuste de resolução, ajuste de contraste, entre outros.

Foram realizados diversos ajustes e testes no *dataset* com o objetivo de melhorar os resultados dos treinamentos. Entre as principais modificações que foram feitas estão: o balanceamento das classes rotuladas, a correção de rótulos incorretos e a exclusão de caquis muito pequenos nas imagens, que poderiam comprometer a precisão do modelo.

Mesmo com um maior balanceamento do *dataset*, a classe de caquis saudáveis continuou maior, sendo dela cerca de 3000 rótulos e mais de 1000 rótulos de caquis doentes. Porém, é possível compensar esse desbalanceamento durante o treinamento do modelo. Além disso, o *dataset* com 1000 imagens teve um aumento para quase 3000 imagens com o uso dos filtros, aumentando também a quantidade de rótulos da classe minoritária.

4.2. Treinamento do modelo

Para o desenvolvimento da Inteligência Artificial, foi utilizado o modelo pré-treinado *YOLOv8*, reconhecido por sua elevada precisão em tarefas de detecção de objetos e por oferecer uma implementação eficiente e moderna. Além disso, o modelo conta com uma comunidade ativa e ampla documentação, o que facilita a adaptação, pesquisa e integração em projetos de Visão Computacional (Solawezt; Francesco, 2024).

A escolha do tamanho da arquitetura *YOLOv8* também foi importante para esse processo. A tabela 1 contém os tamanhos com suas respectivas vantagens e desvantagens.

Tamanho do modelo	Vantagens	Desvantagens
<i>yolov8n</i>	Extremamente rápido e leve, ideal para dispositivos com pouco poder de processamento.	Menos preciso, principalmente em objetos pequenos ou cenas complexas.
<i>yolov8s</i>	Bom equilíbrio entre velocidade e precisão.	Falhas em objetos muito pequenos ou com muitos detalhes.
<i>yolov8m</i>	Mais preciso, lida melhor com objetos pequenos e complexos.	Requer mais memória e GPU mais potente.
<i>yolov8l</i>	Alta precisão, boa captura de detalhes, ótimo para objetos pequenos e cenários complexos.	Muito pesado, lento em tempo real e exige GPU potente.
<i>yolov8x</i>	Máxima precisão disponível, ótimo para <i>datasets</i> muito complexos.	Extremamente pesado, muito lento para aplicações em tempo real, só indicado para máquinas com GPU de últimas gerações.

Tabela 1 - Vantagens e desvantagens dos tamanhos do *Yolov8*

Fonte: Autoria Própria.

Observa-se que o *yolov8n* não seria a escolha mais adequada para a detecção da antracnose, uma vez que a tarefa exige a identificação de detalhes importantes nas imagens. Por esse motivo, optou-se pelo *yolov8m*, modelo de porte intermediário entre os disponíveis, que apresenta melhor desempenho na detecção de objetos pequenos. Além disso, esse modelo, por mais que demande uma boa GPU, não necessita de equipamentos tão potentes quanto as versões maiores, permitindo que o treinamento seja realizado de forma eficiente com a GPU disponível, garantindo precisão sem comprometer recursos computacionais.

O treinamento do modelo foi realizado no *Google Colab*, uma plataforma em nuvem disponibilizada pelo *Google* que permite escrever e executar códigos em Python diretamente no navegador, sem necessidade de instalação ou configuração prévia. É uma ferramenta bastante utilizada em áreas como aprendizado de máquina, ciência de dados e no ensino, por oferecer praticidade e acesso gratuito a recursos computacionais avançados (Google, 2025).

Foram conduzidos diversos treinamentos, com variações tanto no *dataset* quanto nos hiperparâmetros utilizados. Contudo, nesta seção serão apresentados exclusivamente os hiperparâmetros referentes ao último treinamento, realizado com o *dataset* final construído para o modelo. A tabela abaixo indica os hiperparâmetros utilizados.

Hiperparâmetro	Valor
<i>epochs</i>	<i>120</i>
<i>patience</i>	<i>20</i>
<i>batch</i>	<i>16</i>
<i>lr0</i>	<i>0.0012</i>
<i>imgsz</i>	<i>960</i>
<i>workers</i>	<i>8</i>
<i>pretrained</i>	<i>True</i>
<i>resume</i>	<i>False</i>
<i>single_cls</i>	<i>False</i>
<i>box</i>	<i>0.1</i>
<i>cls</i>	<i>1.0</i>
<i>dfl</i>	<i>1.2</i>
<i>val</i>	<i>True</i>
<i>degrees</i>	<i>0.1</i>
<i>hsv_s</i>	<i>0.1</i>
<i>hsv_v</i>	<i>0.1</i>
<i>scale</i>	<i>0.3</i>
<i>fliplr</i>	<i>0.5</i>
<i>mosaic</i>	<i>0.8</i>
<i>mixup</i>	<i>0.3</i>
<i>save</i>	<i>True</i>

Tabela 2 - Hiperparâmetros utilizados para o treinamento

Fonte: Autoria Própria

Na sequência, são expostas as razões que fundamentam a escolha dos valores definidos:

1. *epochs* = 120: Número máximo de épocas de treinamento, definido para permitir que o modelo aprenda melhor a classe minoritária, considerando o desbalanceamento do *dataset*.

2. *patience* = 20: Evita que o treinamento seja interrompido prematuramente, proporcionando mais oportunidades de melhoria do modelo e previne o overfitting e ciclos de treino desnecessários.
3. *batch* = 16: Quantidade de imagens processadas por vez na GPU, garantindo equilíbrio entre estabilidade do gradiente e uso eficiente de memória.
4. *lr0* = 0.0012: Taxa de aprendizado inicial, valor seguro que evita instabilidades no gradiente e permite aprendizado eficaz.
5. *imgsz* = 960: Tamanho das imagens, identifica detalhes minuciosos para a classificação mais precisa, porém demanda mais memória da GPU.
6. *workers* = 8: Número de threads para leitura e pré-processamento das imagens, acelerando o carregamento do *dataset* e evitando que a GPU fique ociosa.
7. *pretrained* = True: Utiliza os pesos pré-treinados do *yolov8m* no *COCO*, acelerando o treinamento e melhorando a precisão do modelo. Essa abordagem evita iniciar o aprendizado do zero, sendo especialmente útil para *datasets* pequenos ou desbalanceados.
8. *resume* = False: Indica que o treinamento não retoma de um checkpoint anterior, mas inicia a partir dos pesos do último modelo salvo.
9. *single_cls* = False: Mantém múltiplas classes no modelo, essencial para cálculo de métricas por classe.
10. *box* = 0.1: Peso da loss para bounding boxes, menor por se tratar de uma tarefa relativamente simples para o modelo.
11. *cls* = 1.0: Peso da perda de classificação, valor padrão.
12. *dfl* = 1.2: Distribuição Focal Loss, auxilia no aprendizado de objetos pequenos e de classes desbalanceadas.
13. *val* = True: Ativa a avaliação de métricas como precisão e recall a cada época, fundamental para acompanhar o aprendizado das classes menos representadas.
14. *degrees* = 0.1: Pequenas rotações aleatórias das imagens para aumentar a diversidade do *dataset*.
15. *hsv_s*, *hsv_v* = 0.1: Pequenas variações de saturação e brilho, contribuindo para a generalização do modelo.
16. *scale* = 0.3: Variação de zoom nas imagens, útil para classes com objetos pequenos.
17. *fliplr* = 0.5: Flip horizontal aleatório, que dobra efetivamente o número de exemplos disponíveis.

18. *mosaic* = 0.8: Combinação de quatro imagens em uma só, aumentando a frequência de aparição de classes raras.
19. *mixup* = 0.3: Mistura objetos de imagens diferentes, reforçando o aprendizado de classes pouco representadas.

Assim, a escolha do modelo *YOLOv8m* aliada à configuração criteriosa dos hiperparâmetros permite estabelecer um equilíbrio entre desempenho, tempo de processamento e capacidade de generalização. A adoção de técnicas de *data augmentation* e o uso de pesos pré-treinados no *COCO* reforçam a robustez do treinamento, o que possibilita o modelo de tratar melhor o desbalanceamento das classes e as variações presentes no conjunto de dados. Dessa forma, a metodologia adotada garante uma base para a análise dos resultados e para a avaliação da eficácia da rede na detecção da enfermidade nos frutos.

4.3. Integração com o computador

Inicialmente, tentou-se estabelecer uma conexão direta entre o drone e o computador, de modo que a Inteligência Artificial pudesse processar as imagens utilizando o endereço *IP* fornecido pelo equipamento. No entanto, essa alternativa não foi possível, pois o *IP* do drone é restrito, impossibilitando o acesso direto ao fluxo de vídeo. Diante dessa limitação, foi necessário desenvolver uma solução alternativa, que viabilizou a integração do drone com o *notebook* de forma eficiente.

O processo inicia com a captura das imagens no campo por meio de um drone, tal qual exposto na Figura 13. Essas imagens são enviadas em tempo real para um celular utilizando o aplicativo *WiFi_CAM*, que permite a transmissão ao vivo dos registros feitos pela câmera do equipamento. Em seguida, o celular é conectado ao *notebook* por meio de um cabo USB. Dessa forma, com o auxílio do *software scrpy*, é possível transmitir a tela do aparelho diretamente para o computador, permitindo que as imagens captadas pelo drone sejam processadas em um ambiente adequado.



Figura 13 - Drone modelo YLR/C
Fonte: Autoria Própria

Num código *Python*, feito para realizar o processamento, a biblioteca *PyAutoGUI* é utilizada para capturar a tela inteira do computador. Essa biblioteca é bastante conhecida por permitir a automação de tarefas, como cliques, digitação e captura de tela, o que facilita a manipulação do conteúdo exibido.

Posteriormente, a biblioteca *OpenCV* é responsável por transformar as imagens em frames e recortar apenas a área da tela exibida pelo *scrpy*, isolando a região de interesse para análise.

As linhas abaixo contém o código *Python* desenvolvido para realizar as predições ao vivo:

```
import pyautogui
import cv2
import numpy as np
from ultralytics import YOLO

print("Iniciando captura da tela... Pressione ESC para sair.")

# Carrega o modelo treinado
model = YOLO("C:/Users/felip/Downloads/best.pt")
# Dicionário de classes com cores
classes = {
    0: ("Caqui", (255, 0, 0)),      # azul
    1: ("Caqui Doente", (0, 0, 255)) # vermelho
}
while True:
    # Captura a tela
    screenshot = pyautogui.screenshot()
    frame = cv2.cvtColor(np.array(screenshot), cv2.COLOR_RGB2BGR)

    frame = frame[100:800, 500:1000] # Recorte da área do vídeo do drone
    # Detecção de objetos
    results = model(frame)
    for result in results:
        boxes = result.boxes
        for box in boxes:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            conf = box.conf[0]
            cls = int(box.cls[0])
            # Nome e cor da classe
            label, color = classes.get(cls, (f"Classe {cls}", (255, 255, 255)))
            # Desenha caixa e label
            cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
            cv2.putText(frame, f"{label} {conf:.2f}", (x1, y1-10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
    cv2.imshow("Vídeo do Drone", frame)
    if cv2.waitKey(1) == 27:
        break
cv2.destroyAllWindows()
```

Outro ponto importante a ser destacado é a presença do modelo treinado para o processamento das imagens, possibilitando a identificação automática dos caquis saudáveis e daqueles afetados pela antracnose.

Por fim, o resultado do processamento realizado pela Inteligência Artificial é exibido em uma janela no *notebook*. Esse fluxo possibilita que as imagens captadas no campo sejam processadas em tempo real, integrando ferramentas de automação e visão computacional para apoiar a análise.

A Figura 14 demonstra todo esse processo de forma esquematizada para melhor entendimento.

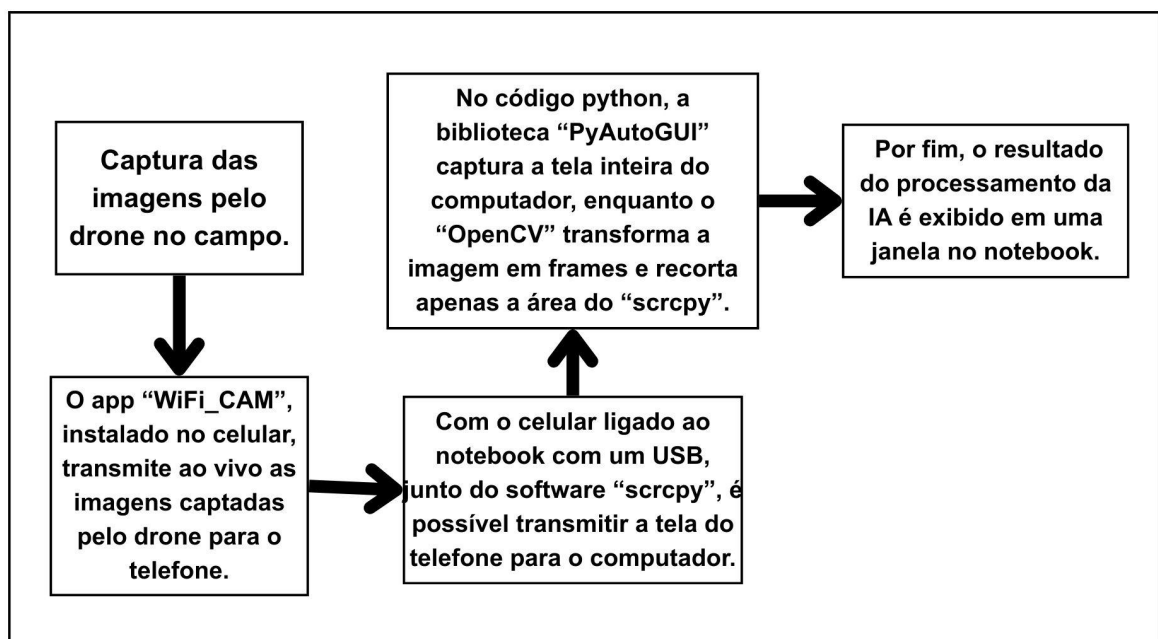


Figura 14 - Esquema de funcionamento integrado

Fonte: Autoria Própria

5. Resultados e Discussão

Nesta seção são apresentados e discutidos os principais resultados obtidos ao longo do desenvolvimento do trabalho. Inicialmente, são abordados os aspectos do treinamento do modelo, considerando os hiperparâmetros aplicados e o impacto das escolhas realizadas no desempenho da rede, seguidas da integração do modelo com o computador.

5.1. Treinamento e *dataset*

A avaliação de modelos de detecção e classificação de objetos exige métricas que consigam medir não apenas a quantidade de acertos, mas também a qualidade desses acertos. Entre as mais importantes estão *Precision*, *Recall* e o *F1-score*, que se complementam para fornecer uma visão equilibrada do desempenho do modelo.

A *Precision* (ou precisão) mede a proporção de predições corretas entre todos os casos que o modelo classificou como positivos. Essa métrica é importante quando os falsos positivos (erros em que o modelo detecta algo que não existe) trazem grandes consequências. A seguir é apresentada a fórmula para o cálculo dessa métrica, onde TP (*True Positives*) representa os verdadeiros positivos e FP (*False Positives*) os falsos positivos.

$$Precision = \frac{TP}{TP+FP}$$

Já o *Recall* (ou sensibilidade) avalia a capacidade do modelo em identificar todos os exemplos positivos que realmente existem. Essa métrica é fundamental quando é mais problemático deixar passar casos reais, como na detecção de doenças em plantas. A seguir a fórmula do *recall*, onde FN (*False Negatives*) são os falsos negativos, ou seja, casos que o modelo deixou de identificar.

$$Recall = \frac{TP}{TP+FN}$$

Dessa forma, o *F1-score* combina *Precision* e *Recall* em uma única métrica. Ele é particularmente útil em cenários de desbalanceamento de classes, já que só terá valores altos quando ambas as métricas também forem altas. Assim, o *F1-score* equilibra a exigência de um modelo que seja ao mesmo tempo preciso e capaz de identificar a maioria dos casos relevantes.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Além disso, utilizou-se a métrica *mAP* (*mean Average Precision*), considerada padrão em *benchmarks* de visão computacional, que corresponde à média da precisão em diferentes limiares de sobreposição entre caixas preditas e reais (*IoU - Intersection over Union*) (Padilla; Netto; Silva, 2020).

Existem dois tipos de *mAP*: *mAP50* e *mAP50-95*.

O *mAP50* calcula a média da *Average Precision* considerando apenas predições cuja sobreposição com o objeto real, medida pelo *IoU* (*Intersection over Union*), seja igual ou superior a 0,5. Essa métrica indica a capacidade do modelo em identificar corretamente os objetos com um nível mínimo de sobreposição, sendo uma referência comum para desempenho básico em detecção de objetos.

Já o *mAP50-95* é uma média do *Average Precision* calculada para múltiplos limiares de *IoU*, variando de 0,5 até 0,95, em incrementos de 0,05. Essa métrica oferece uma avaliação mais rigorosa e detalhada da precisão do modelo, exigindo que as predições estejam mais

próximas da posição e dimensão corretas dos objetos. Portanto, esse indicador é considerado mais completo e exigente do desempenho do modelo em tarefas de detecção.

Por fim, a Matriz de confusão é uma ferramenta para avaliar o desempenho do modelo, fazendo a organização dos resultados das previsões em uma tabela que mostra a relação entre os valores reais e os valores previstos. Dessa forma, é possível identificar não apenas a taxa de acertos, mas também em quais classes o modelo está cometendo erros.

Com base nos indicadores apresentados, é possível observar que o modelo atingiu 0,81053 de precisão, indicando que a grande maioria das previsões positivas realizadas pelo modelo estavam corretas. Já o *recall* de 0,76015 demonstra que o modelo conseguiu identificar aproximadamente 76% de todos os casos reais de caquis saudáveis e caquis com antracnose, mostrando boa capacidade de detecção, ainda que haja espaço para melhoria na captura de todos os exemplos positivos.

Dessa forma, o *F1-score* do modelo foi calculado em aproximadamente 78,45%, indicando que a rede treinada é capaz de identificar tanto caquis saudáveis quanto doentes. No entanto, os resultados também revelam que algumas previsões não são realizadas quando deveriam e que ocorrem detecções incorretas, evidenciando a necessidade de ajustes e refinamentos no modelo.



Figura 15 - Resultados de previsões feitas durante a validação do treinamento.

Fonte: Autoria Própria.

Em seguida, o $mAP50$ de 0,80546 reforça que, considerando um limiar de sobreposição $IoU \geq 0,5$, o modelo consegue localizar corretamente e com precisão a maior parte dos objetos de interesse. Já o $mAP50-95$ de 0,67246 indica que, ao exigir predições mais precisas e próximas das posições corretas dos objetos, o desempenho cai, o que é esperado em tarefas de detecção complexas, especialmente com classes pequenas ou desbalanceadas.

A matriz de confusão, apresentada na Figura 16, evidencia que a classe “Caqui saudável” obteve o maior número de acertos, enquanto a classe “Caqui doente” apresentou confusões mais frequentes, sendo erroneamente classificada como saudável ou confundida com o *background*. Também foram observadas algumas falsas detecções em regiões sem objetos, indicando a necessidade de ajustes adicionais para reduzir alarmes falsos. Esse comportamento pode ser atribuído, em parte, ao desbalanceamento do *dataset*, que dificultou o aprendizado adequado da classe minoritária.

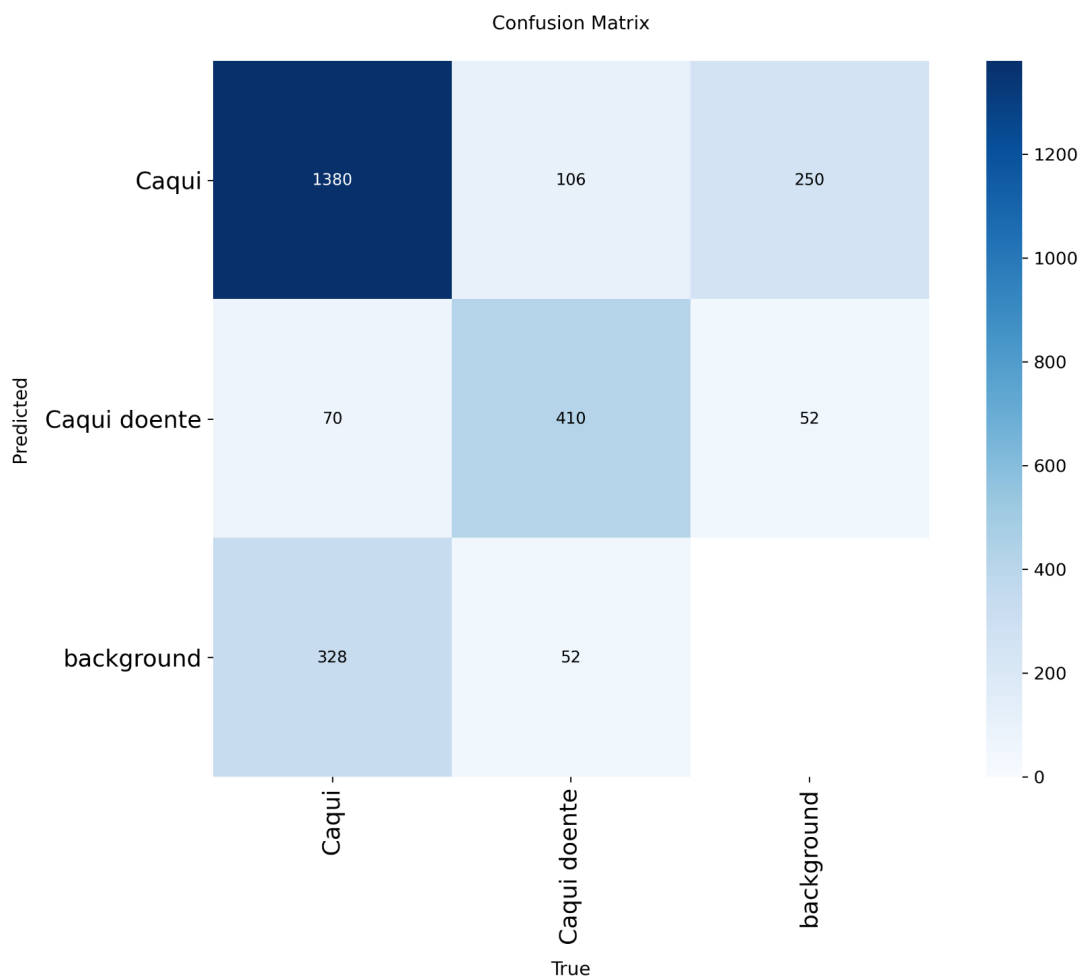


Figura 16 -.Matriz de confusão do treinamento

Fonte: Autoria Própria

5.2. Identificação ao vivo

A proposta de realizar a detecção em tempo real foi efetivamente implementada, conforme ilustrado no esquema da Figura 14, permitindo o envio e o processamento das imagens capturadas. Além disso, o fato de o drone possuir conectividade *Wi-Fi*, possibilitando o uso do aplicativo *WiFi_CAM*, foi fundamental para viabilizar a transmissão das imagens em tempo real.

O espelhamento de tela realizado pelo *screpy* funcionou de forma eficiente, garantindo que o notebook recebesse a imagem ao vivo do drone sem perdas significativas.

Além disso, a utilização do *PyAutoGUI* e do *OpenCV* se mostrou plenamente satisfatória, permitindo capturar a tela, transformar as imagens em frames e processá-las em tempo real com o modelo treinado, atingindo exatamente o objetivo proposto.

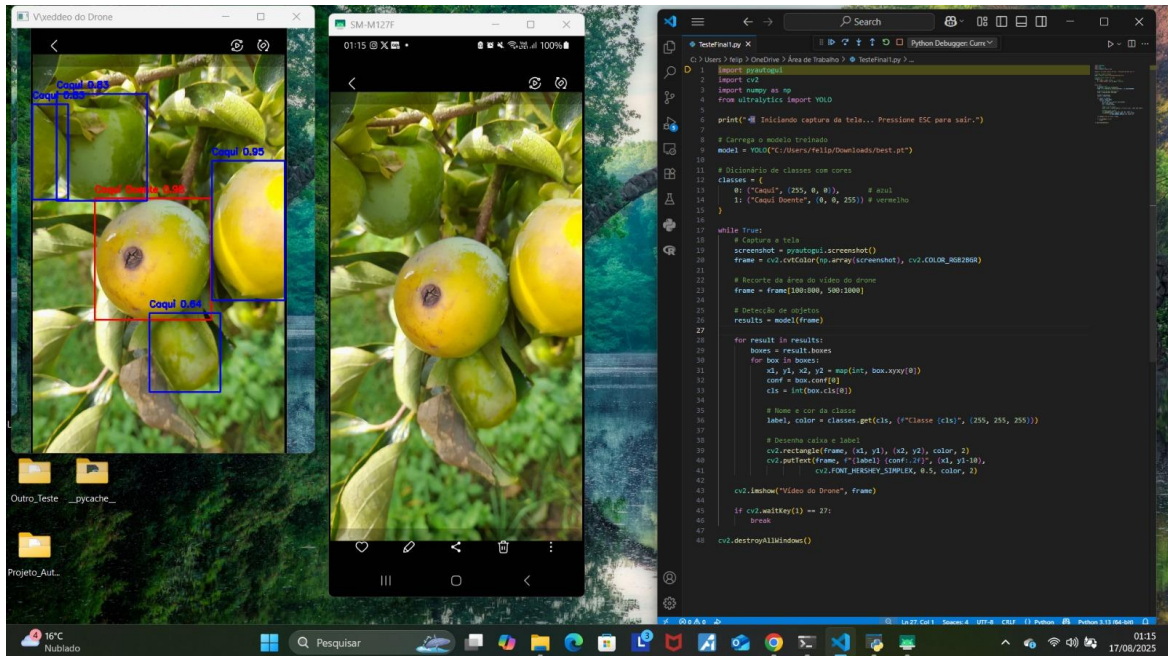


Figura 17 - Predição em tempo real com o uso do *screpy*, *OpenCv* e *pyautogui*

Fonte: Autoria Própria.

Entretanto, não foi possível realizar testes práticos diretamente na plantação, que constituem o cenário ideal para validar a ferramenta em condições reais de uso, devido às restrições de tempo que limitaram o avanço do projeto e pelo fato de a antracnose estar incubada no período de inverno, época em que foi treinada a rede.

A Figura 18 apresenta os resultados de algumas predições feitas em tempo real.

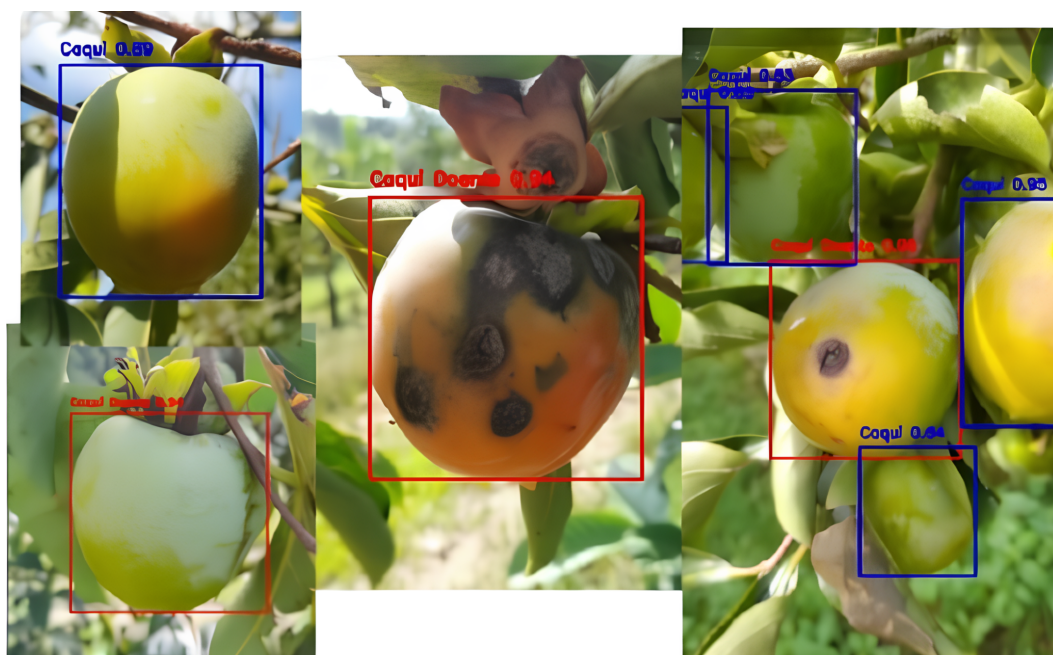


Figura 18 - Resultados das predições realizadas pelo modelo desenvolvido

Fonte: Autoria Própria

6. Conclusão

O estudo apresentou bom desempenho na detecção automática da antracnose em frutos de caquizeiro utilizando sensoriamento remoto por drone junto de técnicas de visão computacional e aprendizado profundo.

Desse modo, foi possível demonstrar a viabilidade técnica de identificar sintomas visuais da doença em imagens aéreas e em transmissões ao vivo, por meio da construção de um *dataset* próprio com imagens coletadas em campo, anotadas no *Roboflow*, e do treinamento do modelo pré-treinado *YOLOv8m*. Esse resultado confirma a hipótese central do trabalho de que a IA pode ser empregada como ferramenta auxiliar no monitoramento contra a antracnose do caqui em Mogi das Cruzes, com potencial para reduzir perdas e apoiar decisões de manejo.

A análise qualitativa das predições e da matriz de confusão revelou comportamento típico em cenários com classes desbalanceadas: a classe “Caqui saudável” obteve maior número de acertos, enquanto a classe “Caqui doente” apresentou mais confusões. Esse padrão indica dois pontos essenciais para aplicação prática: a necessidade de reduzir falsos negativos a fim de não ignorar áreas infectadas e, mitigar falsos positivos com o objetivo de não confundir as classes e balancear de maneira errada os pesos do modelo.

Tendo em vista os resultados obtidos, propõem-se recomendações para trabalhos futuros, como aumentar a amostragem de imagens de caquis doentes, coletar imagens em

várias estações do ano e condições meteorológicas, além de experimentos com calibração de hiperparâmetros no treinamento do modelo, que podem reduzir erros em classes minoritárias.

Partindo da perspectiva de implantação, recomenda-se trabalhar a otimização do modelo, como técnicas de *pruning*, quantização e o uso de versões menores do *YOLO* para inferência embarcada, além de integrar georreferenciamento automático das detecções e uma interface simples para que os produtores possam visualizar mapas de risco e receber recomendações de manejo.

Além dos aspectos técnicos, o trabalho apontou implicações socioeconômicas e operacionais relevantes, visto que, para a tecnologia ser efetivamente implementada no município, é necessário envolver produtores locais em programas de implementação, promover parcerias com órgãos de extensão rural e avaliar a viabilidade econômica da adoção do sistema.

A proposta demonstrou a aplicabilidade de técnicas modernas de visão computacional para a detecção de antracnose em caqui e forneceu métricas de treinamento que embasam as decisões de melhoria do modelo. Sendo assim, os próximos passos devem priorizar a validação em campo ao longo de safras, o aumento e diversificação do *dataset* e a criação de um fluxo de uso centrado no produtor.

Portanto, só assim o sistema poderá evoluir de apenas um conceito para uma ferramenta de apoio ao manejo sustentável e à valorização da produção local de caqui.

7. Referências Bibliográficas

ALMEIDA, Carlos Caetano de. *Identificação e classificação de imagens usando rede neural convolucional e machine learning: implementação em sistema embarcado*. 2019. 202 f. Tese (Doutorado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, 2019.

ALVES, Adson; LACERDA, André; PIANTONI, Jane. *Agrovisão: analisando imagens com Inteligência Artificial: apostila do aluno*. Sorocaba: Fit Instituto de Tecnologia, 2023.

CIPREST. *Caqui Giombô (Diospyros kaki)*. [S. l.], 13 mar. 2017. Imagem presente em: *Viveiro Ciprest – Plantas Nativas e Exóticas*. Disponível em: <https://ciprest.blogspot.com/2017/03/caqui-giombo-diospyrus-kaki.html>. Acesso em: 20 maio 2025.

GOOGLE. Google Colaboratory. Disponível em: <https://colab.research.google.com/>. Acesso em: 17 abril 2025.

GORINSTEIN, S. et al. Comparative content of total polyphenols and dietary fiber in tropical fruits and persimmon. *The Journal of Nutritional Biochemistry*, v. 10, n. 6, p. 367-371, 1999. DOI: [https://doi.org/10.1016/S0955-2863\(99\)00017-0](https://doi.org/10.1016/S0955-2863(99)00017-0). Acesso em: 24 março 2025.

LIMA, Bruno Pinheiro de Melo. *Detecção e contagem automáticas de Euschistus heros (Percevejo marrom) em lavouras de soja usando imagens e aprendizagem profunda*. 2024. 79 f. Dissertação (Mestrado em Sistemas Mecatrônicos) — Universidade de Brasília, Brasília, 2024.

MOLICENTER. Os 4 caquis do Brasil. 2023. [S. l.], [s. d.]. Disponível em: <https://molicenter.com.br/os-4-caquis-do-brasil/>. Acesso em: 20 maio 2025.

OLIVEIRA, Andréia Mara Rotta de; PEREIRA, Sonia Regina de Mello; ANZANELLO, Rafael. Antracnose do caquizeiro: agente causal, sintomas e estratégias de manejo. Porto Alegre: SEAPDR/DDPA, 2021. 26 p. (Circular: divulgação técnica, 9).

PADILLA, Rafael; NETTO, Sergio L.; SILVA, Eduardo A. B. A Survey on Performance Metrics for Object-Detection Algorithms. In: INTERNATIONAL CONFERENCE ON SYSTEMS, SIGNALS AND IMAGE PROCESSING (IWSSIP), 2020.

SANTOS, Rubens Guilherme. Entenda a razão de Mogi das Cruzes ser chamada de “Terra do Caqui”. G1 – Mogi das Cruzes e Suzano, 10 abr. 2022. Disponível em: <https://g1.globo.com/sp/mogi-das-cruzes-suzano/noticia/2022/04/10/entenda-a-razao-de-mogi-das-cruzes-ser-chamada-de-terra-do-caqui.ghhtml>. Acesso em: 23 set. 2025.

SANTOS, Valdirene F. Neves dos; BELLO, Ellen Martins. Aspectos gerais da cultura alimentar do município de Mogi das Cruzes – SP. UNISEPE – Faculdades Integradas do Vale do Ribeira. Disponível em: <https://linkania.org/master/article/view/207>. Acesso em: 23 março 2025.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. *Redes neurais artificiais: para engenharia e ciências aplicadas — fundamentos teóricos e aspectos práticos*. 2. ed. rev. e ampl. São Paulo: Artliber, 2016. ISBN 978-85-88098-887-9.

SILVA, João; OLIVEIRA, Maria. Revisão bibliográfica do uso de inteligência artificial na indústria nacional brasileira e regional do Espírito Santo. Vitória: Universidade Federal do Espírito Santo, 2020.

SOLAWEYT, Jacob; FRANCESCO. What is YOLOv8? A Complete Guide. Roboflow Blog, 23 out. 2024. Disponível em: <https://blog.roboflow.com/what-is-yolov8/>. Acesso em: 25 maio 2025.

ULTRALYTICS. Introducing Ultralytics YOLOv8, 8 março 2023. Disponível em: <https://www.ultralytics.com/pt/blog/introducing-ultralytics-yolov8>. Acesso em: 25 maio 2025.

VASCONCELOS, Gabriel de Castro. Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e deep learning. 2019.

VINA, Abirami. *What is OpenCV? A Guide for Beginners*. Roboflow Blog, 21 março 2024. Disponível em: <https://blog.roboflow.com/what-is-opencv/>. Acesso em: 23 set. 2025.