

DETECÇÃO AUTOMÁTICA DE DOENÇAS EM PLANTAS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS E TRANSFER LEARNING COM RESNET50

Angelo Henrique Oneda*; Alexssandro Cordeiro**

*Acadêmico de Engenharia de Software, e-mail: angelo.oneda77@hotmail.com

**Mestre pelo Programa de Pós-Graduação em Tecnologias Computacionais para o Agronegócio - UTFPR Medianeira, alexssandrofc@gmail.com.

INFORMAÇÕES

Histórico de submissão:

Recebido em: 21 maio. 2025
Aceite: 01 jun. 2025
Publicação online: jun. 2025

RESUMO

Este trabalho tem como objetivo aplicar técnicas de aprendizado profundo para a detecção de doenças em plantas a partir de imagens digitais. Utilizando o conjunto de dados PlantVillage, que contém milhares de imagens classificadas em diversas culturas e patologias, foi desenvolvido um modelo de rede neural convolucional baseado na arquitetura ResNet50. A pesquisa foi motivada pelo desafio enfrentado por agricultores em identificar doenças de forma precoce, especialmente em regiões com acesso limitado a assistência técnica. A metodologia envolveu o pré-processamento das imagens, divisão dos dados em treino e validação, aplicação de aumento de dados e ajuste fino da rede utilizando pesos pré-treinados. O modelo foi treinado em ambientes distintos, incluindo máquina local e Google Colab, com análise de métricas como acurácia, perda, precisão e revocação. Os resultados demonstraram que a ResNet50, mesmo com oscilações na validação, alcançou desempenho satisfatório, indicando seu potencial para aplicação no campo. Conclui-se que a utilização de redes neurais pode contribuir significativamente para o monitoramento fitossanitário de forma automatizada e escalável.

Palavras-chave: redes neurais convolucionais; aprendizado profundo; reconhecimento de doenças; ResNet50; agricultura de precisão.

ABSTRACT

Este estudio tiene como objetivo aplicar técnicas de aprendizaje profundo para la detección de enfermedades en plantas a partir de imágenes digitales. Se utilizó el conjunto de datos PlantVillage, que contiene miles de imágenes clasificadas en diversas especies y patologías vegetales, para desarrollar un modelo de red neuronal convolucional basado en la arquitectura ResNet50. La investigación fue motivada por el desafío que enfrentan los agricultores al identificar enfermedades de manera temprana, especialmente en regiones con acceso limitado a soporte técnico. La metodología incluyó el preprocesamiento de imágenes, la división de los datos en conjuntos de entrenamiento y validación, el uso de técnicas de aumento de datos y el ajuste fino de la red mediante pesos previamente entrenados. El modelo fue entrenado tanto en un entorno local como en la plataforma Google Colab, y su rendimiento fue evaluado mediante métricas como precisión, pérdida, exactitud y recuperación. Los resultados demostraron que la arquitectura ResNet50 alcanzó un desempeño satisfactorio, a pesar de ciertas oscilaciones en la validación, lo que indica su potencial para aplicaciones prácticas. Se concluye que el uso de redes neuronales puede contribuir significativamente al monitoreo fitosanitario de forma automatizada y escalable.

Palabras clave: edes neuronales convolucionales; aprendizaje profundo; reconocimiento de enfermedades; ResNet50; agricultura de precision.

Copyright © 2025, Angelo Henrique Oneda; Alexssandro Cordeiro. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Citação: ONEDA, Angelo Henrique; CORDEIRO, Alexssandro. Detecção automática de doenças em plantas utilizando redes neurais convolucionais e transfer learning com RESNET50. *Iguazu Science*, São Miguel do Iguaçu, v. 3, n. 7, p. 47-58, jun. 2025.

INTRODUÇÃO

O agronegócio, nos tempos contemporâneos, configura-se como uma das principais atividades econômicas em escala global, sendo essencial para o desenvolvimento socioeconômico de diversas nações. A sua relevância ultrapassa os limites da produção primária de alimentos e matérias-primas, tornando-se um eixo estruturante para diversos setores produtivos e industriais. Ao integrar atividades como o fornecimento de insumos, a produção vegetal e animal, o beneficiamento de matérias-primas, a logística de distribuição, a comercialização nacional e internacional, bem como o consumo final, o agronegócio representa uma cadeia produtiva robusta, complexa e profundamente interligada (Cepea, 2022).

A importância estratégica do setor agrícola se evidencia ainda mais diante dos desafios globais do século XXI, como o crescimento populacional, a urbanização acelerada, a escassez de recursos naturais e as mudanças climáticas. O aumento contínuo da população mundial, que segundo estimativas da Organização das Nações Unidas (ONU) poderá ultrapassar os 9,7 bilhões de pessoas até 2050 (ONU, 2019), exige um incremento significativo na produção de alimentos. Nesse contexto, o agronegócio se torna peça-chave para garantir a segurança alimentar, promover o desenvolvimento sustentável e reduzir as desigualdades socioeconômicas entre as regiões.

Os dados sobre o impacto do setor são expressivos. De acordo com informações recentes das Nações Unidas, aproximadamente 27% da força de trabalho mundial está empregada direta ou indiretamente no agronegócio. Isso corresponde a cerca de 870 milhões de pessoas, abrangendo desde pequenos produtores familiares até grandes conglomerados agroindustriais.

Em muitos países em desenvolvimento, especialmente na África Subsaariana, Sudeste Asiático e América Latina, o setor agrícola é o principal motor da economia local, representando uma parcela substancial do Produto Interno Bruto (PIB) e sustentando milhões de famílias em áreas rurais. No Brasil, por exemplo, o agronegócio responde por mais de 23,2% do PIB, conforme dados do Centro de Estudos Avançados em Economia Aplicada (Cepea, 2022), e cerca de 49% das exportações, segundo a Agência Gov (2024), sendo um dos principais vetores de crescimento econômico e geração de divisas.

O papel do agronegócio na sustentabilidade ambiental e social também é fundamental. A produção de alimentos, fibras e bioenergia precisa ser feita de maneira que respeite os limites dos ecossistemas, reduza os impactos ambientais negativos e promova o

uso eficiente dos recursos naturais. Nesse sentido, a adoção de tecnologias sustentáveis, o uso racional da água, o manejo adequado do solo e a conservação da biodiversidade são práticas cada vez mais valorizadas. A agricultura de baixo carbono, os sistemas agroflorestais e o plantio direto são exemplos de estratégias sustentáveis que vêm ganhando espaço em diversos países, contribuindo para a mitigação das emissões de gases de efeito estufa e para o enfrentamento das mudanças climáticas (UTFPR, 2025).

Entretanto, apesar de sua importância econômica, ambiental e social, o agronegócio enfrenta diversos desafios estruturais e operacionais que comprometem seu pleno desenvolvimento. Um dos mais relevantes diz respeito à vulnerabilidade das lavouras à ocorrência de doenças e pragas que comprometem a sanidade das plantas e afetam diretamente a produtividade agrícola. A presença desses agentes bióticos pode provocar perdas expressivas na produção, impactar a qualidade dos produtos e até mesmo inviabilizar colheitas inteiras, com consequências econômicas graves para os produtores e para o abastecimento alimentar. Segundo a Organização das Nações Unidas para a Alimentação e a Agricultura (FAO), cerca de 40% da produção agrícola global é perdida anualmente devido a pragas e doenças (FAO, 2021a).

O impacto desses problemas fitossanitários é particularmente preocupante em regiões tropicais e subtropicais, onde as condições climáticas favorecem o desenvolvimento e a propagação de organismos patogênicos. Culturas como soja, milho, trigo, arroz, café, tomate, batata e frutas são especialmente suscetíveis a infecções por fungos, bactérias, vírus e à infestação por insetos e ácaros. A dificuldade em detectar precocemente esses problemas e em tomar medidas eficazes para seu controle agrava ainda mais a situação. Em muitas áreas agrícolas, o monitoramento das lavouras ainda é realizado de forma manual, baseada na observação visual por técnicos ou agricultores, o que demanda tempo, exige conhecimento especializado e pode apresentar falhas humanas.

As doenças das plantas roubam a economia global em mais de 220 bilhões de dólares anualmente. As pragas invasoras custam aos países pelo menos 70 bilhões de dólares e também são um dos principais responsáveis pela perda de biodiversidade (FAO, 2021a). Espécies como a lagarta-do-cartucho, que se alimenta de plantações que incluem milho, sorgo e milheto, já se espalharam devido ao clima mais quente. Outros, como os gafanhotos do deserto, que são as pragas migratórias mais destrutivas do mundo, devem mudar suas rotas migratórias e distribuição geográfica devido às mudanças climáticas. O impacto desses problemas fitossanitários é particularmente preocupante em regiões tropicais e subtropicais, onde

as condições climáticas favorecem o desenvolvimento e a propagação de organismos patogênicos. Culturas como soja, milho, trigo, arroz, café, tomate, batata e frutas são especialmente suscetíveis a infecções por fungos, bactérias, vírus e à infestação por insetos e ácaros. A dificuldade em detectar precocemente esses problemas e em tomar medidas eficazes para seu controle agrava ainda mais a situação. Em muitas áreas agrícolas, o monitoramento das lavouras ainda é realizado de forma manual, baseada na observação visual por técnicos ou agricultores, o que demanda tempo, exige conhecimento especializado e pode apresentar falhas humanas.

Com o objetivo de mitigar esses riscos, diversas estratégias de controle vêm sendo adotadas ao longo dos anos. O controle químico, baseado na aplicação de defensivos agrícolas, é uma das abordagens mais utilizadas no combate a pragas e doenças. Embora seja eficaz em muitos casos, esse método apresenta limitações importantes, como o alto custo, o impacto ambiental, a contaminação de recursos naturais e alimentos, a exposição da saúde humana a substâncias tóxicas e o desenvolvimento de resistência por parte dos organismos-alvo. Como alternativa, o Manejo Integrado de Pragas (MIP) surgiu como uma abordagem mais sustentável, combinando diferentes métodos de controle – biológicos, químicos, culturais e genéticos – com o objetivo de manter as populações de pragas em níveis economicamente aceitáveis.

Nos últimos anos, a agricultura de precisão tem se destacado como uma revolução tecnológica no setor agropecuário. Essa abordagem baseia-se na coleta, análise e interpretação de dados em tempo real para apoiar a tomada de decisão nas atividades agrícolas. Com o uso de sensores, drones, imagens de satélite, sistemas de posicionamento global (GPS) e softwares especializados, é possível monitorar variáveis ambientais e fisiológicas das plantas, identificar zonas de manejo específicas e aplicar insumos de forma localizada, reduzindo desperdícios e aumentando a eficiência produtiva. No entanto, apesar de seu grande potencial, a adoção da agricultura de precisão ainda é limitada em muitas regiões, devido ao alto custo dos equipamentos, à necessidade de infraestrutura tecnológica e à falta de capacitação técnica dos produtores.

Nesse contexto de transformação digital e busca por soluções inovadoras, as tecnologias baseadas em inteligência artificial (IA) têm se mostrado promissoras para enfrentar os desafios da agricultura moderna. A IA, especialmente por meio do aprendizado profundo (deep learning), vem permitindo o desenvolvimento de sistemas computacionais capazes de aprender padrões complexos a partir de grandes volumes de dados, o que abre novas possibilidades para o monitoramento e diagnóstico de doenças em plantas (Embrapa, 2019). Uma das aplicações mais relevantes dessa tecnologia

está na análise de imagens digitais capturadas por câmeras, drones ou sensores ópticos, que podem ser processadas por algoritmos de visão computacional para identificar sintomas de infecções e infestações de forma automatizada.

As tecnologias de inteligência artificial estão permitindo identificar com mais precisão doenças em folhas de plantas por meio da análise de imagens, conforme destacado por pesquisadores da PUC-SP (2020). Essa abordagem tem revolucionado a forma como os agricultores monitoram a saúde de suas lavouras, permitindo intervenções mais rápidas e precisas.

O uso de redes neurais convolucionais (CNNs), uma das principais arquiteturas de deep learning voltadas para a análise de imagens, tem ganhado destaque em pesquisas científicas e em aplicações práticas no campo. Essas redes são inspiradas na estrutura do sistema visual humano e são capazes de extrair características visuais relevantes, como cores, formas, texturas e padrões, mesmo em ambientes complexos e com variabilidade nas condições de iluminação, ângulo de captura e tipo de planta. Ao treinar modelos de CNNs com bancos de dados compostos por milhares de imagens rotuladas, é possível desenvolver sistemas com alta acurácia na detecção de doenças, como oídio, ferrugem, míldio, mancha foliar, entre outras (Bento, 2019).

Além da acurácia, essas ferramentas oferecem vantagens significativas em termos de escalabilidade, automação e custo-benefício. Diferente das inspeções manuais, que são demoradas, onerosas e suscetíveis a erros humanos, os sistemas baseados em deep learning podem operar de forma contínua, com alta velocidade de processamento e padronização nos diagnósticos. Isso representa um ganho expressivo na eficiência do monitoramento fitossanitário, permitindo que os produtores tomem decisões mais rápidas e fundamentadas, com base em informações precisas e atualizadas. Com isso, é possível antecipar intervenções, aplicar defensivos apenas onde necessário e evitar perdas maiores, contribuindo para o uso racional dos recursos e para a sustentabilidade da produção agrícola.

Além do reconhecimento de doenças, as CNNs podem ser utilizadas em outras aplicações relacionadas à agricultura, como a contagem de frutos, a classificação de espécies, a estimativa de produtividade, a identificação de deficiências nutricionais e o monitoramento de estágios fenológicos. A flexibilidade e a capacidade adaptativa dessas redes tornam-nas ferramentas versáteis, que podem ser integradas a sistemas embarcados em drones, robôs agrícolas, estações meteorológicas e dispositivos móveis. Dessa forma, os agricultores podem acessar os diagnósticos diretamente em seus celulares ou tablets, facilitando o uso prático da

tecnologia no dia a dia do campo (AgroAdvance, 2024).

É importante destacar que o sucesso dessas aplicações depende da qualidade dos dados utilizados no treinamento dos modelos. A diversidade de imagens, representando diferentes espécies, estágios da doença, condições ambientais e tipos de cultivo, é essencial para garantir que os sistemas sejam robustos e generalizáveis. Por isso, a construção de bancos de dados agrícolas representativos, com anotações precisas e curadoria especializada, é uma etapa crítica no desenvolvimento de soluções baseadas em IA para o agronegócio. Iniciativas colaborativas, envolvendo universidades, centros de pesquisa, empresas de tecnologia e produtores rurais, são fundamentais para viabilizar essa infraestrutura de dados e fomentar a inovação no setor.

Dessa forma, este trabalho tem como objetivo avaliar a aplicação de técnicas de aprendizado profundo no reconhecimento de doenças em plantas por meio da análise de imagens digitais. Especificamente, busca-se explorar o desempenho de diferentes modelos de redes neurais convolucionais na tarefa de classificação de sintomas visuais, considerando aspectos como acurácia, sensibilidade, especificidade, robustez e capacidade de generalização. A proposta é investigar como essas ferramentas podem contribuir para o aprimoramento do monitoramento fitossanitário nas lavouras, reduzindo o tempo de diagnóstico, minimizando perdas e promovendo uma agricultura mais inteligente, precisa e sustentável.

A incorporação de inteligência artificial no contexto agrícola representa não apenas um avanço tecnológico, mas uma mudança de paradigma na forma como se conduz a produção de alimentos. Ao aliar ciência de dados, automação e conhecimento agrônomo, é possível construir soluções integradas e personalizadas, que respondam aos desafios específicos de cada sistema produtivo. Mais do que uma tendência, trata-se de uma necessidade emergente, diante de um cenário global marcado por transformações profundas, pressões ambientais e demandas crescentes por alimentos seguros, saudáveis e acessíveis.

Portanto, compreender, desenvolver e aplicar tecnologias baseadas em deep learning na agricultura é um passo essencial para garantir a resiliência, a competitividade e a sustentabilidade do agronegócio no século XXI. A modernização do campo, guiada pela inovação e pela ciência, tem o potencial de transformar realidades, empoderar produtores, preservar recursos naturais e promover um futuro mais justo e equilibrado para as próximas gerações.

METODOLOGIA

Este estudo teve como principal objetivo explorar e aplicar técnicas modernas de aprendizado profundo, com foco em redes neurais convolucionais (CNNs), para o reconhecimento automático de doenças em plantas a partir de imagens digitais. Com o avanço da inteligência artificial e a ampla disponibilidade de bases de dados públicas, tornou-se possível desenvolver soluções robustas para problemas complexos como a identificação precoce de patologias em culturas agrícolas. Neste contexto, foi utilizada a base de dados PlantVillage e implementado um modelo baseado na arquitetura ResNet50, com treinamentos realizados tanto em ambiente local (máquina pessoal) quanto na plataforma de computação em nuvem Google Colab. A seguir, são descritas detalhadamente todas as etapas envolvidas, desde a preparação dos dados até o processo de avaliação do desempenho do modelo treinado, destacando as bibliotecas e técnicas utilizadas. (Noyan, 2022; *Frontiers in Plant Science*, 2023).

A implementação do modelo e todo o fluxo de trabalho foram realizados com o auxílio de bibliotecas consolidadas no ecossistema Python, amplamente utilizadas em aplicações de aprendizado de máquina e visão computacional. Entre elas, destacam-se: TensorFlow Framework principal utilizado para a construção e treinamento da rede neural. Especificamente, a API `tensorflow.keras` foi empregada por oferecer uma interface de alto nível que facilita a criação de modelos de deep learning de forma modular e eficiente. Ela provê classes e funções para camadas, modelos, otimizadores, funções de perda e métricas. Scikit-learn Biblioteca utilizada para tarefas auxiliares fundamentais, como a divisão da base de dados em conjuntos de treinamento e validação (`train_test_split`) e o cálculo de pesos para as classes (`compute_class_weight`). Esses recursos são essenciais para a preparação de dados e para o tratamento de desbalanceamentos no conjunto de imagens. NumPy utilizada para a manipulação de arrays e estruturas de dados numéricas. Sua integração com as demais bibliotecas garante desempenho otimizado em operações vetoriais e matriciais. Matplotlib responsável pela geração de gráficos que permitiram visualizar o comportamento do modelo durante o treinamento. A visualização das métricas de acurácia e perda é fundamental para compreender o processo de aprendizagem e identificar possíveis problemas como overfitting ou underfitting (Machine Learning Mastery, 2022; TensorFlow, 2023).

OS e Shutil Bibliotecas da biblioteca padrão do Python utilizadas para manipulação de arquivos e diretórios, essenciais na etapa de organização da base de dados. Permitem automatizar a criação de pastas e a cópia de imagens para os diretórios corretos de treinamento e validação.

Para o treinamento do modelo, foi utilizada a base de dados PlantVillage, uma base de imagens pública amplamente utilizada em projetos de detecção de doenças em plantas. O subconjunto utilizado foi a versão colorida, que oferece maior riqueza de informação visual, permitindo que o modelo aprenda características relevantes relacionadas à textura, coloração e padrões visuais das folhas. A estrutura da base é composta por subpastas, onde cada uma representa uma classe, ou seja, uma combinação de cultura agrícola e tipo de doença. Exemplos incluem "Apple__Apple_scab", "Potato__Early_blight" e "Tomato__Late_blight". Esse tipo de organização facilita a tarefa de classificação multiclasse, pois permite a utilização direta de funções como `flow_from_directory` para o carregamento das imagens. (Papers With Code, 2024; TensorFlow Datasets, 2024)

A divisão do conjunto de dados foi feita em dois subconjuntos principais: 80% para treinamento e 20% para validação, utilizando a função `train_test_split` com uma semente aleatória fixa (`random_state=42`), garantindo a reprodutibilidade dos experimentos. Para cada classe, foram criadas pastas específicas nos diretórios de treino e validação, e as imagens foram copiadas utilizando a função `shutil.copy`, mantendo a estrutura hierárquica necessária para o funcionamento correto do pipeline de carregamento. (Analytics Vidhya, 2022; Medium, 2025)

Antes do treinamento, as imagens passaram por um processo de pré-processamento e aumento de dados com o objetivo de melhorar a generalização do modelo. O aumento de dados, realizado com a classe `ImageDataGenerator`, tem como finalidade simular variações reais nas imagens, como mudanças na orientação, iluminação e escala, evitando que o modelo aprenda apenas os padrões presentes no conjunto original.

Para o conjunto de treino, foram aplicadas as seguintes transformações:

`rescale=1./255`: normaliza os valores dos pixels para o intervalo $[0, 1]$, essencial para facilitar o processo de treinamento da rede neural.

`rotation_range=50`: aplica rotações aleatórias de até 50 graus nas imagens.

`width_shift_range=0.4` e `height_shift_range=0.4`: deslocamentos horizontais e verticais de até 40% da dimensão da imagem.

`shear_range=0.4`: aplica uma transformação de cisalhamento nas imagens.

`zoom_range=0.4`: realiza aproximações e afastamentos simulados.

`horizontal_flip=True`: realiza a inversão horizontal das imagens.

`fill_mode='nearest'`: define o modo de preenchimento para os pixels vazios gerados pelas transformações.

`brightness_range=[0.5, 1.5]` e `channel_shift_range=30.0`: variações na iluminação e canais de cor, simulando condições de iluminação diversas.

`brightness_range=[0.5, 1.5]` e `channel_shift_range=30.0`: variações na iluminação e canais de cor, simulando condições de iluminação diversas.

O carregamento das imagens foi realizado com a função `flow_from_directory`, tanto para os conjuntos de treino quanto de validação. Os parâmetros principais definidos foram:

`target_size=(299, 299)`: aumento da resolução para capturar detalhes visuais mais refinados, compatível com a arquitetura ResNet50.

`batch_size=32`: número de imagens por lote.

`class_mode='categorical'`: codificação one-hot para problemas de classificação multiclasse.

Essa abordagem garante que as imagens sejam carregadas em tempo real e processadas em lotes, otimizando o uso da memória RAM e acelerando o treinamento.

Com o objetivo de mitigar os efeitos de possíveis desbalanceamentos entre as classes (quantidade desigual de imagens por categoria), foram computados pesos de classe utilizando a função `compute_class_weight` da biblioteca Scikit-learn. Os pesos atribuídos foram posteriormente integrados ao processo de treinamento por meio do parâmetro `class_weight` na função `model.fit()`, assegurando que o modelo não favorecesse classes com maior número de amostras.

A arquitetura escolhida para este trabalho foi a ResNet50, uma rede residual profunda com 50 camadas, reconhecida por sua eficiência na extração de características visuais complexas e por sua capacidade de generalização. A ResNet50 foi importada da biblioteca `tensorflow.keras.applications`, utilizando os seguintes parâmetros:

`weights='imagenet'`: permite o carregamento de pesos pré-treinados na base de dados ImageNet.

`include_top=False`: exclui a última camada densa da arquitetura original, permitindo personalização para o número de classes do problema atual.

`input_shape=(299, 299, 3)`: define o tamanho das imagens de entrada.

O modelo base foi parcialmente congelado para preservar os pesos já treinados e evitar o sobreajuste nas primeiras camadas. Apenas as últimas 50 camadas foram descongeladas para permitir o ajuste fino (fine-tuning) dos pesos com base nas imagens do PlantVillage.

Após o modelo base, foram adicionadas camadas customizadas para adaptação à tarefa de classificação multiclasse. `GlobalAveragePooling2D` reduz a dimensionalidade das saídas da ResNet, mantendo as informações espaciais, `Dense(512, activation='relu')`:

camada densa com 512 neurônios e função de ativação ReLU, Dropout(0.5) técnica de regularização que desativa aleatoriamente 50% dos neurônios durante o treinamento, reduzindo a chance de overfitting e Dense(len(classes), activation='softmax') camada de saída com número de neurônios igual ao número de classes, utilizando softmax para gerar uma distribuição de probabilidade.

O modelo foi compilado utilizando o otimizador Adam, configurado com uma taxa de aprendizado de 0.00001, valor ideal para o processo de fine-tuning, no qual são ajustados os pesos de uma rede pré-treinada em uma nova tarefa. A função de perda adotada foi a `categorical_crossentropy`, indicada para problemas de classificação multiclasse com codificação one-hot, pois mede a divergência entre a distribuição prevista pelo modelo e a distribuição real das classes. Para avaliação do desempenho durante o treinamento, foram utilizados três métricas principais: accuracy, precision e recall. Essa escolha visa proporcionar uma análise mais abrangente da performance do modelo, especialmente relevante em contextos com classes desbalanceadas, nos quais apenas a acurácia pode ser insuficiente para uma avaliação precisa. (GeeksforGeeks, 2025; Science Direct, 2021)

Com o intuito de tornar o treinamento mais eficiente e robusto, foram incorporados três callbacks principais ao processo. O primeiro deles, `EarlyStopping`, tem como função interromper o treinamento automaticamente caso a perda de validação não apresente melhoria após 10 épocas consecutivas, evitando desperdício de tempo computacional e o risco de overfitting. O segundo, `ModelCheckpoint`, atua salvando o modelo com a melhor acurácia de validação alcançada até o momento, garantindo a preservação dos melhores pesos durante as iterações. Por fim, foi utilizado o callback `ReduceLROnPlateau`, responsável por reduzir a taxa de aprendizado pela metade sempre que a perda de validação permanecer estável por cinco épocas, até o limite mínimo de 0.000001. Essa estratégia dinâmica permite que o modelo continue a se ajustar finamente à medida que se aproxima da convergência. (Danielsen, 2019; IEEE Xplore, 2022)

O treinamento do modelo foi realizado ao longo de até 30 épocas, por meio da função `model.fit()`. Os parâmetros `steps_per_epoch` e `validation_steps` foram definidos com base no número total de amostras dos conjuntos de treinamento e validação, respectivamente, garantindo que todas as imagens fossem processadas em cada época. O parâmetro `class_weight` também foi aplicado durante o treinamento, de modo a compensar o desbalanceamento existente entre as classes, atribuindo maior importância às aquelas com menor representatividade. Todos os callbacks mencionados anteriormente foram integrados ao treinamento para

otimização contínua do processo. (Science Direct, 2024; Viso.ai, 2024)

Para fins de comparação de desempenho e análise da eficiência computacional, o modelo foi treinado em dois ambientes distintos. Primeiramente, os testes foram conduzidos em uma máquina local, sem o uso de GPU dedicada, o que permitiu observar as limitações e a viabilidade do treinamento em contextos com menor capacidade computacional. Em seguida, o mesmo processo foi executado na plataforma Google Colab, que oferece acesso gratuito a GPUs como a Tesla T4, proporcionando aceleração significativa do treinamento e evidenciando os ganhos obtidos com o uso de infraestrutura computacional otimizada.

Após a conclusão do treinamento, foi implementada a função `plot_training_history()`, com o objetivo de visualizar graficamente o desempenho do modelo ao longo das épocas. Essa função gera dois gráficos principais: o primeiro mostra a acurácia de treinamento e validação por época, enquanto o segundo exibe a função de perda (loss) para os mesmos conjuntos. A análise visual desses gráficos é fundamental para compreender o comportamento do modelo durante o processo de aprendizagem, permitindo identificar padrões como convergência, indícios de overfitting (quando o desempenho no treinamento é muito superior ao da validação) ou underfitting (quando o modelo não consegue aprender adequadamente nem no conjunto de treinamento). Essas informações orientam decisões sobre possíveis ajustes nos hiperparâmetros, no número de épocas ou nas estratégias de regularização.

RESULTADOS E DISCUSSÃO

São apresentados e discutidos os principais resultados obtidos durante o treinamento e validação do modelo baseado na arquitetura ResNet50, aplicado ao dataset PlantVillage para a classificação multiclasse de doenças em plantas.

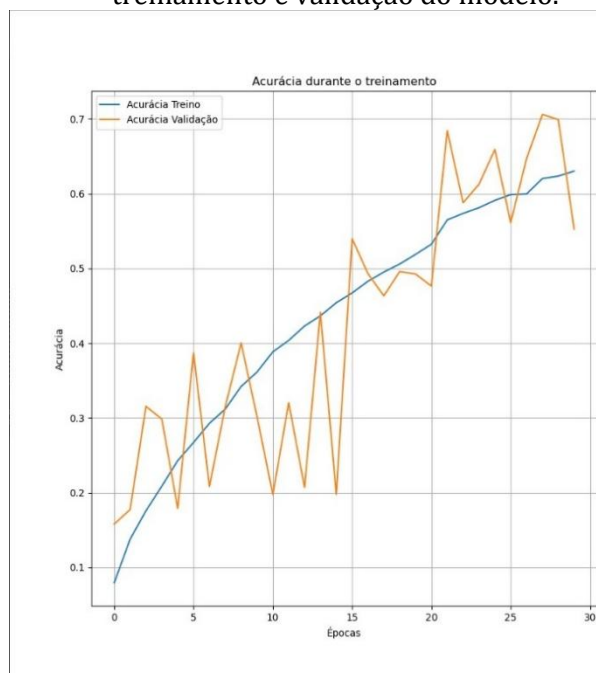
A análise concentra-se no desempenho do modelo ao longo das épocas, utilizando métricas como acurácia, perda (loss), precisão e recall. A partir dos gráficos gerados e do log detalhado do processo de treinamento, busca-se compreender a efetividade do modelo, suas limitações e as possibilidades de aprimoramento.

A interpretação dos resultados é orientada pelas métricas centrais de avaliação em problemas de classificação supervisionada. Especial atenção é dada à diferença de desempenho entre os conjuntos de treinamento e validação, ao comportamento das métricas ao longo do tempo, à estabilidade do aprendizado e à ocorrência de possíveis indícios de overfitting ou underfitting. Essas análises permitem não apenas mensurar o sucesso da abordagem implementada, mas também avaliar o impacto das

decisões metodológicas adotadas ao longo do projeto. (Bioinfo, 2021; Sigmoidal, 2023; Medium, 2023).

O gráfico gerado pela função `plot_training_history()` (Figura 1) ilustra a evolução da acurácia durante o processo de treinamento. Observa-se que a curva de acurácia no conjunto de treino apresenta um crescimento contínuo e relativamente suave, indicando que o modelo está de fato aprendendo padrões relevantes a partir dos dados. Desde a primeira época, onde a acurácia era inferior a 10%, até a trigésima época, em que ultrapassa os 63%, há uma clara progressão.

Figura 1 – Evolução da acurácia durante o treinamento e validação do modelo.



Fonte: Autoria própria (2025).

Por outro lado, a curva referente ao conjunto de validação exibe uma alta oscilação entre as épocas. Apesar de alcançar picos superiores a 70% de acurácia em determinadas épocas, também apresenta quedas abruptas, chegando a valores próximos de 20% em outras. Essa variação pode ser interpretada como um indicativo de instabilidade na generalização do modelo. Ou seja, enquanto o desempenho sobre o conjunto de treinamento melhora de forma contínua, o mesmo não ocorre de forma estável no conjunto de validação.

Essa flutuação pode estar associada a diversos fatores. Em primeiro lugar, a presença de desbalanceamento nas classes — mesmo com a aplicação de pesos de classe via `class_weight` — pode ter levado o modelo a focar em classes majoritárias, dificultando a previsão precisa de classes menos representadas. Em segundo lugar, o alto grau de aumento de dados (data augmentation) durante o treinamento pode ter contribuído para o aprendizado de representações mais variadas, o que pode dificultar

o processo de validação se as imagens de teste não apresentarem variações similares.

Mesmo com essas oscilações, é possível notar uma tendência geral de crescimento na acurácia de validação ao longo das épocas, especialmente a partir da 20ª. Nos momentos de maior estabilidade, o modelo demonstra capacidade de generalização, superando os 70% de acurácia, valor bastante significativo considerando a complexidade do problema de classificação multiclasse e o fato de que o modelo está sendo refinado com fine-tuning apenas nas últimas 50 camadas da ResNet50.

Além da acurácia, o treinamento gerou um conjunto de métricas mais completas, como precision, recall e loss, disponíveis no log de execução (Figura 2). Ao final do treinamento, os valores alcançados foram:

Acurácia de treino aproximadamente 63,2%, Acurácia de validação 55,2%, Perda de validação (`val_loss`): 1.6216, Precisão de validação (`val_precision`): 66,3% e Revocação de validação (`val_recall`): 49,1%.

Figura 2 – Log de execução exibindo métricas durante o treinamento do modelo.

```

MINGW64/c/Users/User/Desktop/PROJETO PLANTVILLAGE
1290/1358 -- 5:50 1s/step - accuracy: 0.6327 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1291/1358 -- 5:45 1s/step - accuracy: 0.6327 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1292/1358 -- 5:40 1s/step - accuracy: 0.6327 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1293/1358 -- 5:35 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1294/1358 -- 5:30 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1295/1358 -- 5:25 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1296/1358 -- 5:19 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1297/1358 -- 5:14 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1298/1358 -- 5:09 1s/step - accuracy: 0.6326 - loss: 1.2409 - precision: 0.8010 - recall: 0.4890
1299/1358 -- 5:04 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1300/1358 -- 4:59 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1301/1358 -- 4:54 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1302/1358 -- 4:48 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1303/1358 -- 4:43 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1304/1358 -- 4:38 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1305/1358 -- 4:33 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1306/1358 -- 4:28 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1307/1358 -- 4:23 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1308/1358 -- 4:17 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1309/1358 -- 4:12 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1310/1358 -- 4:07 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1311/1358 -- 4:02 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1312/1358 -- 3:57 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1313/1358 -- 3:52 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1314/1358 -- 3:47 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1315/1358 -- 3:41 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1316/1358 -- 3:36 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1317/1358 -- 3:31 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1318/1358 -- 3:26 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1319/1358 -- 3:21 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1320/1358 -- 3:16 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1321/1358 -- 3:10 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1322/1358 -- 3:05 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1323/1358 -- 3:00 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1324/1358 -- 2:55 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1325/1358 -- 2:50 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1326/1358 -- 2:45 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1327/1358 -- 2:39 1s/step - accuracy: 0.6326 - loss: 1.2408 - precision: 0.8010 - recall: 0.4890
1328/1358 -- 2:34 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1329/1358 -- 2:29 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1330/1358 -- 2:24 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1331/1358 -- 2:19 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1332/1358 -- 2:14 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1333/1358 -- 2:09 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8010 - recall: 0.4890
1334/1358 -- 2:03 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1335/1358 -- 1:58 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1336/1358 -- 1:53 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1337/1358 -- 1:48 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1338/1358 -- 1:43 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1339/1358 -- 1:38 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1340/1358 -- 1:32 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4890
1341/1358 -- 1:27 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1342/1358 -- 1:22 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1343/1358 -- 1:17 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1344/1358 -- 1:12 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1345/1358 -- 1:07 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1346/1358 -- 1:01 1s/step - accuracy: 0.6326 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1347/1358 -- 56 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1348/1358 -- 51 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1349/1358 -- 46 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1350/1358 -- 41 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1351/1358 -- 36 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1352/1358 -- 30 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1353/1358 -- 25 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1354/1358 -- 20 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1355/1358 -- 15 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1356/1358 -- 10 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1357/1358 -- 5 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
1358/1358 -- 0 1s/step - accuracy: 0.6325 - loss: 1.2407 - precision: 0.8011 - recall: 0.4889
val_accuracy: 0.5527 - val_loss: 1.6216 - val_precision: 0.6632 - val_recall: 0.4927 - training_rate: 2.5000e-06

```

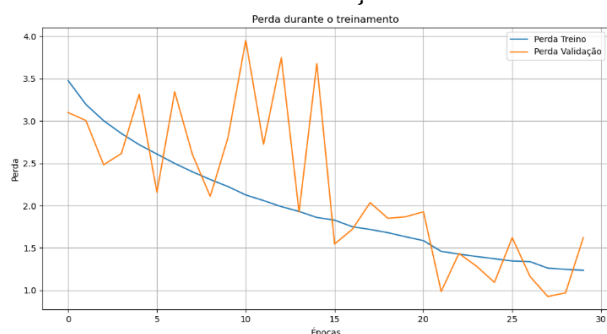
Fonte: Autoria própria (2025).

A análise conjunta dessas métricas permite extrair conclusões mais robustas sobre o desempenho do modelo. A precisão elevada indica que, das previsões feitas como pertencentes a uma determinada classe, uma alta proporção está correta. Por outro lado, a revocação mais baixa sugere que o modelo não está sendo igualmente eficaz em identificar todos os exemplos reais de cada classe. Isso pode ser um

reflexo do desbalanceamento de classes, mesmo com a ponderação aplicada por `class_weight`.

A função de perda, que permanece acima de 1.6 no conjunto de validação, sinaliza que o modelo ainda encontra dificuldades em fazer previsões com alta confiança. Esse valor relativamente elevado pode ser resultado da grande quantidade de classes e da complexidade intrínseca do conjunto de dados, especialmente considerando que cada classe representa não apenas uma cultura diferente, mas também uma condição fitossanitária distinta.

Figura 3 – Evolução da perda (loss) durante o treinamento e validação do modelo.



Fonte: Autoria própria (2025)

Durante várias épocas do treinamento, nota-se que o modelo entra em uma fase de saturação, onde a acurácia, a perda e as demais métricas tornam-se praticamente constantes. Como mostrado no log, isso ocorre a partir da época 20, quando os valores de acurácia no treino estabilizam em torno de 63,2% e as métricas de precisão e recall param de oscilar. Esse comportamento é típico de quando o modelo atinge um platô no espaço de otimização. O mecanismo `ReduceLROnPlateau` foi corretamente utilizado, reduzindo a taxa de aprendizado progressivamente até atingir o valor mínimo de $2.5e-6$, mas não foi suficiente para escapar da estagnação.

Com base nas evidências gráficas e numéricas, é possível realizar uma análise sobre os fenômenos de overfitting e underfitting. O crescimento consistente da acurácia de treino, contrastando com a grande variabilidade na acurácia de validação, pode sugerir overfitting: o modelo aprende muito bem os dados de treinamento, mas não consegue generalizar de maneira confiável. Contudo, o nível de acurácia de validação não é tão inferior ao de treino, especialmente a partir da 20ª época. Isso indica que o modelo não está totalmente superajustado, mas sim que ainda há espaço para melhorias em termos de regularização e otimização do treinamento. Medidas adicionais, como o ajuste da taxa de dropout (de 0.5 para 0.3), uso de técnicas de normalização de batch (batch normalization) ou aumento da diversidade de imagens no conjunto de validação, podem contribuir para mitigar essa lacuna entre treino e validação. Já o underfitting — situação na qual o modelo não

consegue aprender nem os dados de treino — não parece ocorrer aqui. O fato de a acurácia no conjunto de treino atingir mais de 63% com consistência mostra que o modelo está aprendendo representações significativas (Analytics Vidhya, 2020; Oppermann, 2021; GeeksforGeeks, 2025).

O uso da arquitetura ResNet50 com pesos pré-treinados no ImageNet e o ajuste das últimas 50 camadas via fine-tuning mostraram-se adequados para a tarefa proposta. Essa abordagem tem como principal vantagem a reutilização de representações visuais genéricas aprendidas a partir de um grande corpus de imagens, o que acelera o processo de convergência e reduz a quantidade de dados necessários para treinar o modelo com eficácia.

A adição de camadas densas com ativação ReLU, pooling global e dropout entre a ResNet e a camada de saída foi uma estratégia bem fundamentada. No entanto, é possível que a camada densa com 512 neurônios, combinada com um dropout elevado (0.5), tenha causado certa perda de informação ou dificuldade de aprendizado. Uma possível sugestão para trabalhos futuros é testar arquiteturas mais leves ou ajustar a arquitetura final com um número menor de neurônios ou ativar mecanismos de regularização L2 para manter a complexidade sob controle.

O treinamento foi realizado tanto em ambiente local quanto na plataforma Google Colab. Em ambiente local, sem GPU dedicada, o tempo de execução foi significativamente maior, tornando inviável a realização de experimentos em larga escala. Já no Google Colab, o acesso a GPUs como a Tesla T4 possibilitou a execução de cada época em aproximadamente 5 a 6 segundos por batch, totalizando um tempo aceitável mesmo para 30 épocas completas.

Essa diferença de desempenho comprova a importância de infraestruturas baseadas em aceleração por hardware para experimentos com redes neurais profundas. Além disso, o Colab facilita a integração com bibliotecas e armazenamento em nuvem, favorecendo o fluxo de trabalho e reprodutibilidade dos experimentos.

Com base nos resultados obtidos, é possível sugerir algumas estratégias com potencial para melhorar o desempenho do modelo de classificação. Um dos primeiros ajustes recomendados seria a modificação da taxa de dropout, reduzindo-a de 0.5 para 0.3. Essa alteração visa evitar a perda excessiva de informações úteis durante o treinamento, contribuindo para uma melhor generalização sem comprometer a capacidade de aprendizado da rede. Outra abordagem relevante seria o aumento do conjunto de validação, tanto em volume quanto em diversidade. A inclusão de mais imagens, aliada a um balanceamento explícito entre as classes, pode contribuir significativamente para a estabilidade das métricas de validação e reduzir flutuações indesejadas no desempenho. Além disso, a

aplicação de técnicas de aumento de dados (data augmentation) diretamente no conjunto de validação pode ser uma alternativa interessante. Ao introduzir pequenas variações visuais nas imagens — como rotações, ajustes de brilho ou pequenas transformações geométricas — o modelo se depara com cenários mais desafiadores e realistas, o que contribui para uma avaliação mais robusta de sua capacidade de generalização. (Machine Learning Mastery, 2019; Stack Overflow, 2021; Medium, 2024)

Também vale considerar a exploração de outras arquiteturas de redes neurais convolucionais modernas e mais eficientes, como EfficientNetB0, MobileNetV2 ou DenseNet121, que podem oferecer um melhor equilíbrio entre desempenho preditivo e custo computacional, especialmente em ambientes com recursos limitados.

Por fim, uma estratégia promissora seria a adoção de aprendizado por transferências múltiplas. Isso poderia envolver a realização de pré-treinamentos intermediários com bases de dados agrícolas que compartilhem características visuais com o PlantVillage, permitindo que o modelo desenvolva representações mais especializadas e relevantes para o domínio agrícola antes de ser ajustado especificamente para a tarefa final. A combinação dessas abordagens tem o potencial de superar as limitações observadas durante os testes e ampliar consideravelmente a eficácia do modelo em aplicações reais.

CONCLUSÕES

Este trabalho teve como objetivo desenvolver e avaliar um modelo de classificação multiclasse de doenças em plantas utilizando a arquitetura ResNet50 com fine-tuning, aplicado ao dataset PlantVillage. Os resultados obtidos demonstraram que a abordagem é viável e apresenta desempenho satisfatório, com acurácia de validação superior a 70% em algumas épocas e métricas complementares, como precisão e recall, dentro de faixas coerentes para problemas de alta complexidade e com desbalanceamento de classes.

A análise das curvas de acurácia e perda evidenciou que o modelo foi capaz de aprender padrões relevantes nos dados de treinamento, embora tenha apresentado instabilidades durante a validação. Tais oscilações indicam uma necessidade de melhorias na capacidade de generalização, especialmente em relação ao equilíbrio entre classes e à robustez do conjunto de validação.

A estratégia de transfer learning, aliada ao ajuste das últimas camadas da rede, mostrou-se eficaz para acelerar o aprendizado e obter bons resultados mesmo com um número limitado de épocas. Entretanto, a ocorrência de possíveis indícios de

overfitting, evidenciada pela divergência entre as curvas de treino e validação, aponta para oportunidades de aprimoramento no processo de regularização.

Considerando os resultados e limitações observadas, conclui-se que o modelo baseado na ResNet50 constitui uma base sólida para aplicações práticas no diagnóstico automatizado de doenças em plantas. Para trabalhos futuros, recomenda-se a adoção de técnicas adicionais de balanceamento, ajustes na taxa de dropout, ampliação do conjunto de validação e experimentação com arquiteturas mais leves e modernas, como EfficientNet ou MobileNetV2, de modo a alcançar maior precisão e generalização com menor custo computacional.

REFERÊNCIAS

- ABAG-RP. (2024). **Números do Agronegócio**. Associação Brasileira do Agronegócio da Região de Ribeirão Preto. Disponível em: <https://www.abagrp.org.br/numeros-do-agro>
- Agência Gov. (2024). **Exportações do agronegócio fecham 2023 com US\$ 166,55 bilhões em vendas**. Disponível em: <https://agenciagov.ebc.com.br/noticias/202401/exportacoes-do-agronegocio-fecham-2023-com-us-166-55-bilhoes-em-vendas>
- AgroAdvance. (2024). **Machine Learning e Deep Learning: Usos e aplicações na agricultura**. Disponível em: <https://agroadvance.com.br/blog-machine-learning-deep-learning-agricultura/>
- AgroReceita. (2025). **Inteligência artificial na agricultura: aplicações e benefícios**. Disponível em: <https://agoreceita.com.br/inteligencia-artificial-na-agricultura/>
- Analytics Vidhya. (2020). **How to Treat Overfitting in Convolutional Neural Networks**. Disponível em: <https://www.analyticsvidhya.com/blog/2020/09/overfitting-in-cnn-show-to-treat-overfitting-in-convolutional-neural-networks/>
- Analytics Vidhya. (2022). **Image Classification Using Resnet-50 Deep Learning Model**. Disponível em: <https://www.analyticsvidhya.com/blog/2022/09/image-classification-in-stl-10-dataset-using-resnet-50-deep-learning-model/>
- Bento, D. (2019). **Deteção e identificação de doenças em plantas utilizando Deep Learning**. Instituto Politécnico do Porto. Disponível em:

- https://recipp.ipp.pt/bitstream/10400.22/15480/1/DM_DanielBento_2019_MEI.pdf
- Bioinfo. (2021). Métricas de avaliação em machine learning - acurácia, sensibilidade, precisão, especificidade e F-score. **Revista BIOINFO**. Disponível em: <https://bioinfo.com.br/metricas-de-avaliacao-em-machine-learning-acuracia-sensibilidade-precisao-especificidade-e-f-score/>
- Cepea. (2022a). **Afinal, quanto o agronegócio representa no PIB brasileiro?** Centro de Estudos Avançados em Economia Aplicada. Disponível em: <https://www.cepea.org.br/br/opiniao-cepea/afinal-quanto-o-agronegocio-representa-no-pib-brasileiro.aspx>
- Cepea. (2022b). **PIB do Agronegócio Brasileiro. Centro de Estudos Avançados em Economia Aplicada**. Disponível em: <https://www.cepea.esalq.usp.br/br/pib-do-agronegocio-brasileiro.aspx>
- Danielsen, N. (2019). **Simple Image Classification with ResNet-50**. Medium. Disponível em: <https://medium.com/@nina95dan/simple-image-classification-with-resnet-50-334366e7311a>
- Embrapa. (2019). **Inteligência artificial aplicada na agricultura de precisão e digital**. Empresa Brasileira de Pesquisa Agropecuária. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/doc/1170729/1/P-Inteligencia-artificial-aplicada-na.pdf>
- FAO. (2020). **The impact of pests and diseases on food production**. Rome: FAO. Disponível em: <https://www.fao.org>. Acesso em: 18 mai. 2025.
- FAO. (2021a). **Mudança climática influencia na perda da produção agrícola para pragas, conclui estudo apoiado pela FAO**. Organização das Nações Unidas para a Alimentação e a Agricultura. Disponível em: <https://www.fao.org/brasil/noticias/detail-events/en/c/1411810/>
- FAO. (2021b). **FAO aponta perdas anuais de US\$ 314 bi com pragas e doenças em seis países, incluindo o Brasil**. Disponível em: <https://sopesp.com.br/2021/03/19/fao-aponta-perdas-anuais-de-us-314-bi-com-pragas-e-doencas-em-seis-paises-incluindo-o-brasil/>
- FAO. (2022). **FAO lista 5 doenças de plantas que a crise climática está agravando**. Organização das Nações Unidas para a Alimentação e a Agricultura. Disponível em: <https://brasil.un.org/pt-br/184058-fao-lista-5-doen%C3%A7as-de-plantas-que-crise-clim%C3%A1tica-est%C3%A1-agravando>
- FAO. (2023). **Pense antes de clicar**. Organização das Nações Unidas para a Alimentação e a Agricultura. Disponível em: <https://www.fao.org/brasil/noticias/detail-events/en/c/1639373/>
- FERENTINOS, K. P. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, v. 145, p. 311-318, 2018.
- GeeksforGeeks. (2025a). **Image Classification using ResNet**. Disponível em: <https://www.geeksforgeeks.org/image-classification-using-resnet/>
- GeeksforGeeks. (2025b). **ML | Underfitting and Overfitting in Machine Learning**. Disponível em: <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>
- HUGHES, D. P.; SALATHÉ, M. (2015). **An open access repository of images on plant health to enable the development of mobile disease diagnostics**. arXiv preprint arXiv:1511.08060.
- IEEE Xplore. (2022). **Plant Disease Identification Using a Novel Convolutional Neural Network**. Disponível em: <https://ieeexplore.ieee.org/document/9674894>
- Machine Learning Mastery. (2019). **How to Avoid Overfitting in Deep Learning Neural Networks**. Disponível em: <https://www.machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/>
- Medium. (2023). **Principais métricas de avaliação de modelos em Machine Learning**. Disponível em: <https://medium.com/data-hackers/principais-m%C3%A9tricas-de-classifica%C3%A7%C3%A3o-de-modelos-em-machine-learning-94eeb4b40ea9>
- Medium. (2024). **Day 49: Overfitting and Underfitting in DL — Regularization Techniques**. Disponível em: <https://medium.com/@bhatadithya54764118/day-49-overfitting-and-underfitting-in-dl-regularization-techniques-8ded20baa3d6>

- MOHAMMED, N. A. A.; AL-ANI, A. (2021). Deep learning techniques for detection and classification of plant diseases. In: INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER APPLICATIONS (ACA), 2021, Baghdad. **Anais [...].** IEEE, p. 146–151.
- Noyan, M. A. (2022). **Uncovering Bias in the PlantVillage Dataset.** Medium. Disponível em: <https://medium.com/data-science/uncovering-bias-in-the-plantvillage-dataset-7ac564334526>
- Oligos Biotec. (2021). **Pragas agrícolas: o que são e como combatê-las?** Disponível em: <https://oligosbiotec.com.br/pragas-agricolas-o-que-sao-e-como-combate-las/>
- ONU. (2019). **População mundial deve chegar a 9,7 bilhões de pessoas em 2050, diz relatório da ONU. Organização das Nações Unidas.** Disponível em: <https://brasil.un.org/pt-br/83427-popula%C3%A7%C3%A3o-mundial-deve-chegar-97-bilh%C3%B5es-de-pessoas-em-2050-diz-relat%C3%B3rio-da-onu>
- ONU. (2022a). **8 bilhões de humanos: quais são as previsões até 2030.** Forbes. Disponível em: <https://forbes.com.br/forbeslife/2022/11/8-bilhoes-de-humanos-quais-sao-as-previsoes-ate-2030/>
- ONU. (2022b). **World population prospects 2022: summary of results. New York: United Nations Department of Economic and Social Affairs.** Disponível em: <https://www.un.org/development/desa/pd/content/World-Population-Prospect-2022>. Acesso em: 18 mai. 2025.
- ONU. (2024). **População mundial chegará a 9,9 bilhões em 2054.** Organização das Nações Unidas. Disponível em: <https://news.un.org/pt/story/2024/04/1830966>
- Oppermann, A. (2021). **Underfitting and Overfitting in Deep Learning.** Medium. Disponível em: <https://artem-oppermann.medium.com/underfitting-and-overfitting-in-deep-learning-687b1b7eb738>
- Papers With Code. (2024). **PlantVillage Dataset.** Disponível em: <https://paperswithcode.com/dataset/plantvillage>
- PUC-SP. (2020). **Visão computacional na agricultura: APIs de detecção e reconhecimento de doenças das plantas.** Pontifícia Universidade Católica de São Paulo. Disponível em: <https://revistas.pucsp.br/index.php/teccogs/article/view/48548>
- RAJPUT, D. S.; BODHE, S. K. (2020). Transfer learning with convolutional neural networks for real-time plant disease detection. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATION, AND INTELLIGENT SYSTEMS (ICCCIS), 2020, Greater Noida. **Anais [...].** IEEE, p. 500–505.
- Science Direct. (2021). **Deep Learning in Image Classification using Residual Network (ResNet).** Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921000284>
- Science Direct. (2024). **An ensemble of deep learning architectures for accurate plant disease classification.** Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574954124001602>
- Sigmoidal. (2023). **Métricas de Avaliação em Modelos de Classificação em Machine Learning.** Disponível em: <https://sigmoidal.ai/metricas-de-avaliacao-em-modelos-de-classificacao-em-machine-learning/>
- SIMONYAN, K.; ZISSERMAN, A. (2014). **Very deep convolutional networks for large-scale image recognition.** arXiv preprint arXiv:1409.1556.
- Stack Overflow. (2021). **How to improve performance of CNN and reduce overfitting?** Disponível em: <https://stackoverflow.com/questions/66289658/how-to-improve-performance-of-cnn-and-reduce-overfitting>
- TAN, M.; LE, Q. V. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2019, Long Beach. Proceedings **Anais...** PMLR, p. 6105–6114.
- TensorFlow Datasets. (2024). **plant_village.** Disponível em: https://www.tensorflow.org/datasets/catalog/plant_village
- TensorFlow. (2023). **Keras: The high-level API for TensorFlow.** Disponível em: <https://www.tensorflow.org/guide/keras>
- UTFPR. (2025). **Agricultura sustentável e inteligência artificial.** Universidade Tecnológica

Federal do Paraná. Disponível em:
<https://revistas.utfpr.edu.br/recit/article/download/19217/19217%20pdf>

Viso.ai. (2024). **Deep Residual Networks (ResNet, ResNet50) A Complete Guide**. Disponível em:
<https://viso.ai/deep-learning/resnet-residual-neural-network/>