

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PAULO HENRIQUE DE CAMARGO DIONYSIO MARTINS

**APLICATIVO MÓVEL UTILIZANDO TÉCNICAS DE DEEP
LEARNING PARA IDENTIFICAÇÃO DE DOENÇA FOLIAR**

BAURU
Novembro/2025

PAULO HENRIQUE DE CAMARGO DIONYSIO MARTINS

APLICATIVO MÓVEL UTILIZANDO TÉCNICAS DE DEEP LEARNING PARA IDENTIFICAÇÃO DE DOENÇA FOLIAR

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Orientador: Prof. Dr. André Luis Debiaso Rossi

BAURU

Novembro/2025

M386a

Martins, Paulo Henrique de Camargo Dionysio

Aplicativo Móvel Utilizando Técnicas de Deep Learning para Identificação de Doença Foliar / Paulo Henrique de Camargo Dionysio Martins. -- Bauru, 2025
51 p.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) -
Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: André Luis Debiaso Rossi

1. Redes neurais convolucionais. 2. Dispositivo móvel. 3. Detecção doenças de plantas. 4. Visão computacional. 5. Agricultura de precisão. I. Título.

Paulo Henrique de Camargo Dionysio Martins

Aplicativo Móvel Utilizando Técnicas de Deep Learning para Identificação de Doença Foliar

Trabalho de Conclusão de Curso do Curso de Bacharelado em Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. André Luis Debiaso Rossi

Orientador

Departamento de Computação
Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

**Profa. Dra. Simone das Graças
Domingues Prado**

Departamento de Computação
Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

Prof. Dr. Clayton Reginaldo Pereira

Departamento de Computação
Faculdade de Ciências

Universidade Estadual Paulista "Júlio de
Mesquita Filho"

Bauru, 12 de novembro de 2025.

Dedico este trabalho à minha bisa vó Iracema, pelo seu amor incondicional.

Agradecimentos

Primeiramente, a Deus, por me sustentar e permitir que os objetivos traçados fossem alcançados ao longo de toda esta jornada de estudos e dedicação. Estendo minha gratidão à Nossa Senhora, por sua intercessão e proteção materna.

Aos meus amados avós, pais e irmão, por serem a base e o porto seguro. O incentivo incondicional nos momentos de dificuldade e a compreensão diante da minha ausência foram essenciais para a concretização deste curso.

Aos meus amigos, Dante, Eduardo, Enzo, Guilherme, Gustavo, João Lucas, Lucca e Manu, por todo companheirismo durante a graduação, sem vocês, essa jornada não teria mesmo sentido. O companheirismo, as risadas, o apoio mútuo e até mesmo a paciência nos momentos de maior estresse foram o motor que me impulsionou. Vocês são a prova de que, na vida acadêmica e além, as melhores conquistas são aquelas que podemos compartilhar.

Aos professores, pela paciência, pelas correções minuciosas e pelos inestimáveis ensinamentos que lapidaram meu conhecimento e conduziram a um desempenho profissional sólido.

E, com profunda saudade e gratidão, dedico este momento à memória da minha bisavó, que partiu este ano. Sua sabedoria, força e amor incondicional permanecem vivos em meu coração, sendo uma inspiração eterna.

Resumo

Este trabalho propõe o desenvolvimento de um aplicativo móvel para reconhecimento de doenças foliares a partir de imagens, visando apoiar a tomada de decisão de pequenos e médios produtores. A solução combina um modelo de *deep learning* baseado em redes neurais convolucionais (CNNs) com uma arquitetura cliente–servidor: a interface é implementada em Flutter e a inferência do modelo é disponibilizada por meio de uma API em Flask. O projeto contempla curadoria e pré-processamento de um conjunto de imagens de folhas, treinamento e ajuste de um classificador CNN, construção de serviços REST para comunicação com o aplicativo e avaliação de acurácia e desempenho com imagens reais. Espera-se demonstrar que a integração entre IA e dispositivos móveis possibilita diagnósticos rápidos e acessíveis, contribuindo para a detecção precoce de doenças, uso mais racional de insumos e aumento da eficiência produtiva na agricultura de precisão. O estudo discute limitações, como variações de iluminação e qualidade de captura, e indica melhorias futuras, incluindo aumento de base de dados e técnicas de *data augmentation*.

Palavras-chave: visão computacional; CNN; diagnóstico de doenças de plantas; Flutter; Flask; agricultura de precisão.

Abstract

This project presents a mobile application for leaf disease recognition from images to support decision-making for small and medium-scale farmers. The solution combines a deep learning model based on Convolutional Neural Networks (CNNs) with a client–server architecture: the user interface is built with Flutter, while model inference is exposed through a Flask REST API. The workflow includes curation and preprocessing of a leaf image dataset, training and tuning of a CNN classifier, implementation of REST services for communication with the app, and evaluation of accuracy and runtime performance using real images. We aim to show that integrating AI with mobile devices enables fast and accessible diagnostics, fostering early disease detection, more rational input usage, and improved efficiency in precision agriculture. The study also discusses limitations—such as illumination variability and capture quality—and outlines future improvements, including dataset expansion and data augmentation strategies.

Keywords: computer vision; CNN; plant disease diagnosis; Flutter; Flask; precision agriculture..

Lista de Figuras

Figura 1 – Arquitetura típica de uma CNN, com blocos de <i>camadas convolucionais</i> (Conv layers) e blocos de <i>camadas totalmente conectadas</i> (FC layers). . . .	23
Figura 2 – Bloco residual típico da ResNet.	25
Figura 3 – Resultados do Teste 1 com ResNet50.	37
Figura 4 – Resultados do Teste 2 com ResNet50, após ajuste de Dropout e Label Smoothing.	37
Figura 5 – Resultados do Teste 3 com ResNet50, com taxa de aprendizado aumentada.	38
Figura 6 – Resultados do Teste 4 com ResNet50, após descongelar mais camadas e remover regularização.	38
Figura 7 – Resultados do Teste 1 com MobileNet, estabelecendo a linha de base.	39
Figura 8 – Resultados do Teste 2 com MobileNet, após otimização.	40
Figura 9 – Matriz de Confusão do modelo MobileNetV2 no conjunto de teste.	43
Figura 10 – Telas do aplicativo desenvolvido: (a) tela inicial, (b) seleção da imagem e (c) resultados obtidos.	45
Figura 11 – Matriz de Confusão do modelo MobileNetV2 no conjunto de teste.	51

Lista de Tabelas

Tabela 1 – Comparativo dos trabalhos relacionados sobre detecção de doenças em plantas.	31
Tabela 2 – Comparativo de Métricas de Desempenho: ResNet50 vs. MobileNetV2. . .	41
Tabela 3 – Relatório de Classificação Detalhado do Modelo MobileNetV2 (Conjunto de Teste)	41

Lista de abreviaturas e siglas

IA	Inteligência Artificial
DL	<i>Deep Learning</i>
ML	Aprendizado de Máquina (<i>Machine Learning</i>)
TCC	Trabalho de Conclusão de Curso
CNN	Rede Neural Convolucional (<i>Convolutional Neural Networks</i>)
API	Interface de Programação de Aplicações
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
PLN	Processamento de linguagem natural
GPU	Unidade de processamento gráfico

Sumário

1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Hipóteses e Questão de Pesquisa	15
1.3	Objetivos	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	Estrutura do trabalho	16
2	FUNDAMENTAÇÃO	17
2.1	Agricultura 4.0	17
2.2	Inteligência Artificial	19
2.3	Aprendizado de Máquina	20
2.4	Aprendizado Profundo	21
2.4.1	Algoritmos de DL	22
2.4.1.1	Convolutional Neural Networks	23
2.4.1.2	VGGNet	24
2.4.1.3	<i>Residual Networks</i>	25
2.4.1.4	GoogLeNet	26
2.4.1.5	MobileNet	27
3	REVISÃO DA LITERATURA	29
3.0.1	Síntese dos Trabalhos Relacionados	31
4	MATERIAL E MÉTODOS	33
4.1	Conjunto de dados	33
4.2	Linguagens e Ferramentas	33
4.3	Treinamento e Avaliação dos Modelos	34
4.3.1	Métricas de Avaliação	35
4.4	Avaliação do Sistema Integrado	35
5	RESULTADOS	36
5.1	Desempenho dos Modelos de DL	36
5.1.1	Experimentos com a Arquitetura ResNet50	36
5.1.1.1	Teste 1: ConFiguração Inicial	36
5.1.1.2	Teste 2: Redução da Regularização	37
5.1.1.3	Teste 3: Aumento da Taxa de Aprendizado	37

5.1.1.4	Teste 4: Ajuste Fino Agressivo	38
5.1.2	Experimentos com a Arquitetura MobileNet	39
5.1.2.1	Teste 1: ConFiguração Inicial com MobileNet	39
5.1.2.2	Teste 2: Otimização de Hiperparâmetros	39
5.1.3	Comparação ResNet50 vs MobileNetV2	40
5.1.3.1	Classificação Detalhada e Matriz de Confusão	41
5.2	Desempenho do Sistema Integrado	44
5.3	Interface e Funcionalidade do Aplicativo Móvel	44
6	CONCLUSÃO	46
	REFERÊNCIAS	48
	APÊNDICE A – MATRIZ DE CONFUSÃO VERTICAL (MOBILE- NETV2)	51

1 Introdução

Com o avanço contínuo da tecnologia e a crescente demanda por soluções inovadoras na agricultura, a utilização de técnicas de Inteligência Artificial (IA) tem se destacado como uma ferramenta promissora no auxílio ao diagnóstico de doenças em plantas. A IA permite a detecção automatizada de doenças com acurácia superior a 95%, usando modelos preditivos que consideram variáveis ambientais e fitossanitárias para antecipar surtos com 81–95% de precisão, favorecendo intervenções mais sustentáveis e eficientes ([GONZÁLEZ-RODRÍGUEZ et al., 2024](#)).

Entre as diversas áreas da IA, o ML se mostra de grande relevância, pois possibilita a análise automatizada de grandes volumes de dados agrícolas. Nesse contexto, o uso de imagens digitais de folhas para detecção de anomalias tem ganhado espaço, permitindo decisões mais rápidas e precisas no manejo de lavouras. Técnicas baseadas em CNNs vêm apresentando resultados expressivos ao identificar padrões visuais relacionados a pragas e doenças, muitas vezes de forma mais eficiente do que métodos tradicionais de inspeção visual realizados manualmente. As CNNs extraem automaticamente características das imagens e alcançam desempenho de ponta nas tarefas de detecção de doenças foliares, mostrando-se superiores às abordagens convencionais ([JIANG, 2021](#)).

Um exemplo é o estudo de [Mohanty, Hughes e Salathé \(2016\)](#), que aplicou técnicas de DL - uma subárea do ML, tem se mostrado eficiente em diversas aplicações de visão computacional ([GOODFELLOW et al., 2016](#)) - para classificar imagens de folhas de plantas, atingindo uma acurácia superior a 99% na identificação de 26 doenças em 14 culturas diferentes. Esse resultado evidencia o potencial da IA como ferramenta de apoio à agricultura de precisão, reforçando sua aplicabilidade prática e relevância científica.

Diante desse cenário, este Trabalho de Conclusão de Curso (TCC) propõe o desenvolvimento de um aplicativo móvel capaz de realizar o reconhecimento de doenças foliares a partir de imagens coletadas pelo usuário. O objetivo é facilitar a identificação precoce de problemas fitossanitários, proporcionando aos agricultores um recurso acessível que contribua para tomadas de decisão mais rápidas e assertivas.

Portanto, este trabalho insere-se no esforço global de aplicar recursos tecnológicos avançados em benefício da agricultura, reafirmando o papel da IA como um grande aliado na construção de um setor mais eficiente, sustentável e produtivo.

1.1 Justificativa

A escolha pelo desenvolvimento de um aplicativo para reconhecimento de doenças foliares se justifica pela escalabilidade, acessibilidade e eficiência que essas soluções podem oferecer a pequenos e médios produtores, sobretudo em áreas rurais com acesso limitado a agrônomos especializados. O aplicativo AgroTIC, por exemplo, mostrou impacto positivo ao conectar agricultores e agrônomos por meio de smartphones e técnicas de machine learning (ML) em uma rede de mais de 130 fazendas na Colômbia ([HINOJOSA et al., 2023](#)). Na mesma linha, o aplicativo Plantix oferece diagnósticos via DL e recomendações em linguagem local para pequenos agricultores que não dispõem de assistência técnica ([CHASSIN, 2025](#)). Outro estudo, em regiões da África Oriental, demonstra que um app baseado em IA para diagnóstico de doenças em mandioca, aliado a alertas via SMS, pode ser uma solução viável para produtores com dificuldade de acesso a extensões agrícolas ([LEGG; HUGHES, 2017](#)).

Neste contexto, a proposta de utilizar um modelo de rede neural convolucional (CNN) aliado a um aplicativo móvel se justifica por sua capacidade de democratizar o acesso a uma tecnologia inteligente, prática e de baixo custo. A aplicação do DL na identificação de doenças em plantas tem se mostrado altamente eficaz em estudos recentes de [Gao, Lin e Sinnott \(2017\)](#) e [Elhassouny e Smarandache \(2019\)](#), alcançando níveis elevados de acurácia e oferecendo uma alternativa promissora para a automatização desse processo.

Do ponto de vista técnico, o projeto agrega valor por integrar diferentes áreas do conhecimento: desenvolvimento mobile com Flutter, construção de APIs com Flask e aplicação de modelos de IA com Python. Essa combinação interdisciplinar favorece a inovação, especialmente considerando que existem poucas aplicações práticas que implementam efetivamente essas técnicas no dia a dia agrícola. Um exemplo prático é o desenvolvimento de um aplicativo para manejo sustentável de café, que utiliza Flutter para a interface mobile e Flask para o servidor back-end, integrado com técnicas geoespaciais para estimar a distribuição de pragas e nutrientes, contribuindo para decisões agrícolas mais precisas e com menor impacto ambiental ([RODRÍGUEZ et al., 2025](#)).

Além disso, o trabalho promove o aprendizado prático na área de ML e desenvolvimento mobile, fortalecendo habilidades relevantes e atualizadas com as exigências do mercado tecnológico. A resolução do problema apresentado pode gerar um impacto direto na agricultura de precisão, incentivando o uso consciente de defensivos agrícolas, reduzindo perdas e contribuindo para uma produção mais sustentável.

A motivação pessoal também é um fator importante: a possibilidade de aliar tecnologia à solução de problemas reais e relevantes, como os enfrentados por produtores agrícolas, torna este trabalho não apenas desafiador, mas também socialmente significativo. Assim, justifica-se o empenho em desenvolver uma ferramenta funcional, acessível e tecnicamente robusta, que possa trazer benefícios reais tanto para a comunidade acadêmica quanto para o setor agrícola.

1.2 Hipóteses e Questão de Pesquisa

A hipótese central deste trabalho é que o uso de técnicas de IA, especificamente CNNs, integradas a um aplicativo móvel, pode auxiliar na identificação precoce de doenças foliares com alto grau de precisão, oferecendo aos agricultores uma ferramenta acessível e de fácil utilização.

Parte-se do pressuposto de que, ao disponibilizar uma solução tecnológica baseada em dispositivos móveis, será possível democratizar o acesso a diagnósticos rápidos e confiáveis, especialmente para pequenos e médios produtores localizados em regiões onde a assistência técnica especializada é limitada.

Dessa forma, a questão de pesquisa que norteia este estudo pode ser enunciada da seguinte forma:

É possível desenvolver um aplicativo móvel integrado a um modelo de IA capaz de identificar doenças foliares com precisão suficiente para apoiar a tomada de decisão de agricultores em diferentes contextos produtivos?

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver uma solução móvel que utilize técnicas de IA para o diagnóstico de doenças foliares por meio de imagens.

1.3.2 Objetivos Específicos

- Selecionar, organizar e pré-processar um conjunto de imagens foliares adequadas ao treinamento de modelos de DL;
- Implementar um modelo de CNN em Python capaz de classificar as doenças com base nas imagens das folhas;
- Construir uma API com Flask para gerenciar a comunicação entre o modelo de IA e o aplicativo móvel;
- Desenvolver a interface do aplicativo em Flutter, permitindo ao usuário capturar imagens e receber o diagnóstico da planta;
- Avaliar a acurácia e a performance do sistema por meio de testes com imagens reais e analisar seus resultados;
- Validar a aplicabilidade da solução proposta em ambientes reais de uso, destacando suas limitações e potenciais melhorias.

1.4 Estrutura do trabalho

Este Trabalho de Conclusão de Curso está estruturado em cinco capítulos, além desta introdução:

- **Capítulo 2 – Fundamentação Teórica:** apresenta os conceitos relacionados à Agricultura 4.0, IA, ML, CNNs e tecnologias utilizadas no desenvolvimento do aplicativo.
- **Capítulo 3 - Revisão da Literatura:** apresenta trabalhos relacionados a aplicativos móveis e o uso de CNNs para diagnóstico de doenças foliares. O objetivo é delinear o estado da arte e identificar lacunas, especialmente quanto à generalização dos modelos, robustez em condições reais de campo e usabilidade das soluções.
- **Capítulo 4 - Material e Métodos:** descreve os materiais (conjunto de dados PlantVillage, linguagens e hardware) e os métodos empregados no desenvolvimento do aplicativo, detalhando o pré-processamento de imagens, a metodologia de treinamento com ajuste fino e as métricas de avaliação das arquiteturas ResNet50 e MobileNet.
- **Capítulo 5 - Resultados:** apresenta os resultados dos experimentos de treinamento, detalhando a evolução da acurácia e da perda para as arquiteturas ResNet50 (atingindo 97,74%) e MobileNetV2 (atingindo 98,1%). O capítulo também inclui o Relatório de Classificação Detalhado, a Matriz de Confusão e a avaliação do desempenho do sistema integrado, com latência ponto a ponto de aproximadamente 100 ms, além da interface e funcionalidade do aplicativo móvel.
- **Capítulo 6 - Conclusão:** apresenta a síntese dos resultados, confirmando que a acurácia (MobileNet: 97,83%) e a latência (100 ms) validam a viabilidade técnica do sistema. O capítulo reforça a contribuição da pipeline completa e do artefato funcional para a agricultura de precisão, reconhece as limitações relativas à generalização e dependência de conectividade, e detalha os desdobramentos futuros, como inferência embarcada e uso de imagens de campo.

2 Fundamentação

O presente capítulo tem como objetivo apresentar a base teórica necessária para a compreensão do desenvolvimento deste trabalho. Para isso, são discutidos conceitos fundamentais que permeiam tanto a área agrícola quanto o campo da IA, com ênfase em técnicas modernas de ML e DL aplicadas à detecção de doenças foliares.

Inicialmente, é introduzido o conceito de Agricultura 4.0, um marco da transformação digital no setor agrícola, que busca aumentar a produtividade e a eficiência por meio da integração de tecnologias emergentes, como Internet das Coisas, Big Data e IA.

Na sequência, são abordados os fundamentos de IA, desde sua concepção histórica até suas principais áreas de aplicação, destacando sua relevância em sistemas capazes de aprender e tomar decisões. Dentro desse contexto, é detalhado o ML, um dos subcampos mais relevantes da IA, que permite o desenvolvimento de algoritmos capazes de identificar padrões e realizar previsões a partir de dados.

O capítulo também discute o DL, uma vertente avançada do ML que tem se destacado em aplicações de visão computacional. São apresentadas as principais arquiteturas de redes neurais profundas utilizadas na literatura para tarefas de classificação e diagnóstico, como CNNs, VGGNet, ResNet, GoogLeNet e MobileNet, evidenciando sua importância no contexto da agricultura de precisão e, em especial, no reconhecimento de doenças foliares.

Dessa forma, este capítulo fornece os fundamentos conceituais que sustentam a proposta do presente trabalho, estabelecendo a conexão entre as inovações tecnológicas da Agricultura 4.0 e os avanços da IA aplicados ao setor agrícola.

2.1 Agricultura 4.0

A Agricultura 4.0, também denominada agricultura digital, consiste em uma nova fase da evolução agrícola marcada pela integração de tecnologias digitais e sistemas inteligentes em todas as etapas da cadeia produtiva. Essa transformação acompanha o movimento mais amplo da chamada Quarta Revolução Industrial, no qual a automação, a conectividade e a análise de dados se tornam elementos centrais para a modernização de setores tradicionais ([WOLFERT et al., 2017](#)).

Os pilares fundamentais da Agricultura 4.0 incluem:

- Aumento da produtividade, por meio da utilização de algoritmos e sensores capazes de otimizar o plantio, a irrigação, o manejo e a colheita;

- Alocação racional de recursos, que busca reduzir o desperdício de insumos agrícolas como fertilizantes, defensivos e água, promovendo eficiência no uso dos recursos naturais;
- Adaptação às mudanças climáticas, com base na análise de séries históricas de dados meteorológicos e no uso de modelos preditivos para antecipar riscos ambientais e orientar o manejo agrícola;
- Redução do desperdício de alimentos, com sistemas inteligentes que acompanham a produção, o transporte e a distribuição, garantindo maior rastreabilidade e segurança alimentar.

Essa abordagem se apoia em tecnologias como Internet das Coisas (IoT), que conecta sensores e máquinas agrícolas; Big Data, que permite o processamento de grandes volumes de informações; IA, que transforma dados em conhecimento útil para a tomada de decisão; sensoriamento remoto, utilizado no monitoramento de lavouras via satélites e drones; e computação em nuvem, que possibilita armazenamento, integração e compartilhamento de dados em tempo real.

Na prática, a Agricultura 4.0 possibilita, por exemplo, a criação de mapas de prescrição de insumos, em que áreas específicas de uma plantação recebem diferentes quantidades de fertilizantes ou defensivos conforme suas necessidades. Da mesma forma, o uso de drones e imagens de satélite permite identificar pragas ou deficiências nutricionais em estágios iniciais, possibilitando intervenções mais rápidas e precisas (KAUR et al., 2022).

Outro aspecto relevante é a agricultura de precisão, considerada uma das principais vertentes da Agricultura 4.0. Essa prática utiliza tecnologias digitais para personalizar o manejo agrícola, garantindo que cada talhão de terra seja tratado conforme suas características particulares. Além disso, sistemas integrados de gestão agrícola oferecem uma visão holística da produção, auxiliando na logística, no controle de custos e na sustentabilidade ambiental.

Os dispositivos móveis também desempenham um papel essencial na consolidação da Agricultura 4.0. Com a ampla disseminação de smartphones e tablets, produtores rurais, técnicos e gestores têm acesso direto a informações e ferramentas digitais em tempo real, independentemente da localização geográfica. Aplicativos agrícolas permitem o monitoramento remoto de lavouras, o acompanhamento de dados climáticos, o registro de atividades de campo e a gestão de insumos, tudo de maneira integrada e acessível.

Além disso, esses dispositivos funcionam como interfaces entre o produtor e os sistemas inteligentes baseados em IoT e IA, possibilitando a visualização de dados provenientes de sensores, drones ou estações meteorológicas. Isso torna a tomada de decisão mais ágil e fundamentada em dados precisos. Outro benefício é a conectividade entre os diferentes agentes da cadeia produtiva, como cooperativas, agrônomos e fornecedores, promovendo maior colaboração e eficiência operacional.

A popularização dos dispositivos móveis também contribuiu para a inclusão tecnológica no meio rural, pois reduz barreiras de acesso a soluções avançadas que antes dependiam de equipamentos caros ou infraestrutura complexa. Assim, pequenos e médios produtores podem adotar práticas de agricultura digital, melhorando a produtividade e a sustentabilidade de suas operações (DUNCOMBE, 2016).

Dessa forma, a Agricultura 4.0 não se limita apenas ao aumento da produtividade, mas também busca alinhar a produção agrícola aos princípios da sustentabilidade, da segurança alimentar e da resiliência frente às mudanças climáticas. Trata-se, portanto, de um paradigma que conecta ciência, tecnologia e prática agrícola, abrindo caminho para um setor mais competitivo, eficiente e sustentável.

2.2 Inteligência Artificial

A IA é um campo interdisciplinar da Ciência da Computação dedicado ao estudo e ao desenvolvimento de sistemas e algoritmos capazes de executar tarefas que, tradicionalmente, requereriam a inteligência humana (RUSSELL; NORVIG, 1995). Entre essas tarefas, destacam-se o reconhecimento de padrões, o processamento de linguagem natural, a tomada de decisões, a visão computacional e a aprendizagem. Em linhas gerais, a IA busca reproduzir, em certo grau, processos de raciocínio, percepção e cognição que caracterizam o comportamento humano, possibilitando que máquinas atuem de maneira autônoma ou semi-autônoma em diferentes contextos.

O conceito de IA remonta ao século XX, em especial com os trabalhos pioneiros de Alan Turing, que em 1950 publicou o artigo *Computing Machinery and Intelligence*, no qual propôs a famosa questão “As máquinas podem pensar?” e introduziu o Teste de Turing como um critério para avaliar o comportamento inteligente de sistemas computacionais. Poucos anos depois, em 1956, John McCarthy cunhou o termo “inteligência artificial” durante a histórica Conferência de Dartmouth, considerada o marco oficial do nascimento da área (MCCARTHY et al., 2006). A partir desse ponto, a IA passou a atrair atenção crescente da comunidade científica e tecnológica, estabelecendo-se como uma disciplina em rápida evolução.

Desde sua origem, o campo da IA avançou significativamente, passando por diferentes fases de entusiasmo e de estagnação, conhecidas como invernos da IA. Esses períodos ocorreram devido a limitações computacionais e à falta de dados para treinar modelos mais complexos. Contudo, a partir dos anos 2000, com o aumento da capacidade de processamento, a disponibilidade de grandes volumes de dados e o surgimento de algoritmos mais sofisticados, a IA retomou seu crescimento e se consolidou como uma das áreas mais promissoras da ciência moderna.

As aplicações da IA atualmente abrangem uma ampla gama de domínios: na saúde, algoritmos auxiliam no diagnóstico de doenças, na descoberta de medicamentos e na análise de

exames médicos; na indústria automotiva, impulsionam o desenvolvimento de veículos autônomos; na agricultura, são empregados para prever safras, monitorar lavouras e identificar pragas ou doenças; no comércio eletrônico, alimentam sistemas de recomendação que personalizam a experiência dos usuários; e no cotidiano, estão presentes em assistentes virtuais, como Siri, Alexa e Google Assistente, que processam comandos de voz em tempo real.

De maneira geral, as abordagens em IA podem ser classificadas em dois grandes grupos:

- **IA fraca (ou estreita):** refere-se a sistemas projetados para executar tarefas específicas, como jogar xadrez, reconhecer imagens ou sugerir filmes em plataformas de streaming. Esses sistemas não possuem consciência nem entendimento além da função para a qual foram programados.
- **IA forte (ou geral):** refere-se a uma forma de inteligência hipotética que, em teoria, seria capaz de aprender, compreender e aplicar conhecimentos em diferentes contextos, de forma semelhante ao ser humano. A IA forte implicaria em raciocínio abstrato, criatividade e até mesmo consciência, mas ainda se mantém como um objetivo distante e um campo de investigação em aberto ([RUSSELL; NORVIG, 1995](#)).

Além dessa classificação, pesquisadores também distinguem entre IA simbólica, baseada em regras e representações explícitas do conhecimento, e IA conexionista, que utiliza redes neurais artificiais inspiradas no funcionamento do cérebro humano. Atualmente, a IA conexionista, especialmente com o avanço do DL, tem ganhado maior destaque devido ao seu desempenho em tarefas complexas, como visão computacional e processamento de linguagem natural.

Por fim, é importante ressaltar que a IA, ao mesmo tempo em que apresenta oportunidades transformadoras para a sociedade, também levanta discussões éticas e sociais relevantes. Questões como privacidade de dados, viés algorítmico, transparência e impacto no mercado de trabalho fazem parte do debate contemporâneo, sendo necessárias regulamentações e políticas que garantam o uso responsável e sustentável dessas tecnologias.

2.3 Aprendizado de Máquina

O ML, ou aprendizado de máquina, constitui um subcampo da IA que se dedica ao desenvolvimento de algoritmos capazes de aprender a partir de dados e realizar previsões ou tomar decisões sem a necessidade de programação explícita para cada tarefa ([BISHOP; NASRABADI, 2006](#)). Diferentemente dos sistemas tradicionais, nos quais o comportamento é definido por regras fixas, os modelos de ML ajustam seus próprios parâmetros com base nos padrões identificados nos dados.

A essência do ML está na capacidade de um sistema reconhecer padrões e regularidades a partir da experiência, de modo a realizar inferências ou previsões sobre novos conjuntos de dados. Seus algoritmos possuem aplicações em diversas áreas, que vão desde sistemas de recomendação de filmes e diagnósticos médicos (RODER et al., 2023), até a detecção automatizada de doenças foliares em plantas (SARDOGAN; TUNCER; OZEN, 2018).

Os métodos de ML podem ser classificados em três categorias principais, de acordo com o paradigma de aprendizado (SUTTON; BARTO, 1998):

- **Aprendizado supervisionado:** utiliza conjuntos de dados rotulados, nos quais entradas estão associadas a saídas corretas. Um exemplo é a classificação de e-mails como “spam” ou “não spam”.
- **Aprendizado não supervisionado:** identifica padrões em dados não rotulados, como na análise de agrupamentos, em que os algoritmos segmentam dados em grupos com características semelhantes.
- **Aprendizado por reforço:** baseia-se na aprendizagem por tentativa e erro, na qual o algoritmo recebe recompensas ou penalidades de acordo com suas ações. Essa abordagem é amplamente utilizada em áreas como jogos e robótica.

2.4 Aprendizado Profundo

O DL, ou aprendizado profundo, é um subcampo do ML que se concentra no uso de redes neurais artificiais com múltiplas camadas ocultas, permitindo a modelagem de relações complexas entre dados de entrada e saída. Diferentemente de métodos tradicionais de ML, nos quais a extração de características geralmente depende de especialistas humanos, o DL é capaz de realizar a extração automática de atributos em diferentes níveis de abstração. Essa característica possibilita a construção de representações hierárquicas dos dados, tornando-o altamente eficaz em tarefas de reconhecimento de padrões complexos (GOODFELLOW et al., 2016).

O surgimento do DL está diretamente relacionado à evolução das redes neurais artificiais, inicialmente propostas nos anos 1940 e 1950, mas que só alcançaram impacto significativo a partir da década de 2010. Esse crescimento foi impulsionado pela combinação de três fatores fundamentais:

1. Avanços em arquiteturas de redes neurais, como as CNNs, que revolucionaram a visão computacional;
2. Aumento da capacidade computacional, sobretudo com o uso de unidades de processamento gráfico (GPUs) - processador especializado em fazer milhares de cálculos simples ao mesmo tempo - que aceleraram o treinamento de modelos complexos;

3. Disponibilidade de grandes volumes de dados, fundamentais para que os modelos pudessem aprender padrões robustos em cenários reais.

Esses elementos permitiram que o DL superasse, em diversas áreas, técnicas tradicionais de IA. Por exemplo, em competições de visão computacional como o ImageNet Large Scale Visual Recognition Challenge (ILSVRC), modelos baseados em redes neurais profundas alcançaram desempenhos superiores ao de métodos anteriores, marcando uma mudança de paradigma no campo da IA.

As aplicações do DL se estendem por uma ampla variedade de domínios:

- **Visão computacional:** reconhecimento facial, detecção de objetos, diagnóstico médico por imagem e identificação de doenças foliares em plantas;
- **Processamento de linguagem natural (PLN):** tradução automática, análise de sentimentos, sistemas de recomendação e assistentes virtuais;
- **Robótica e controle:** navegação autônoma, manipulação de objetos e veículos autônomos;
- **Ciências aplicadas:** bioinformática, previsão climática e análise de séries temporais complexas.

Outro aspecto fundamental do DL é sua capacidade de generalização em cenários complexos, mesmo quando os dados apresentam grande variabilidade. Isso o torna particularmente relevante para aplicações na agricultura de precisão, como o monitoramento de lavouras e a detecção automatizada de doenças foliares, nas quais as imagens podem variar em iluminação, ângulo ou estágio de desenvolvimento da planta.

Em síntese, o DL representa um dos maiores avanços recentes da IA. Sua capacidade de extrair representações de dados de forma hierárquica e automatizada, aliada ao desenvolvimento de arquiteturas cada vez mais sofisticadas, estabelece o DL como um dos pilares da inovação tecnológica contemporânea.

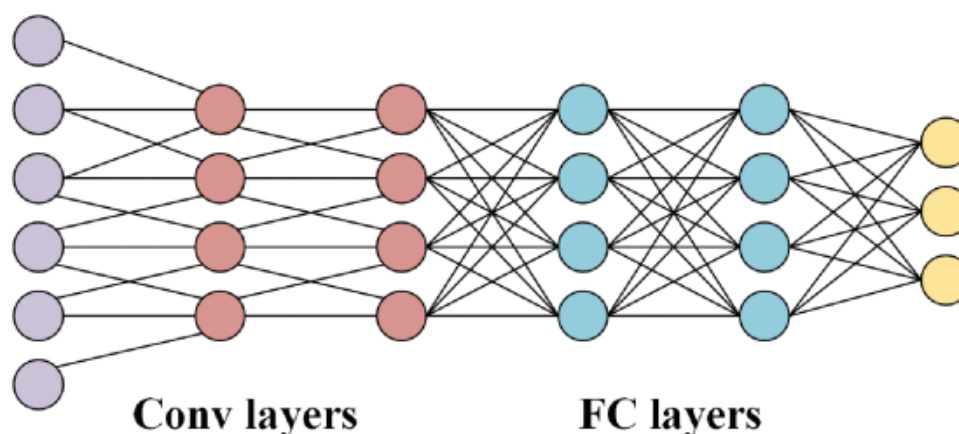
2.4.1 Algoritmos de DL

Nesta seção, são apresentados alguns dos principais algoritmos de DL utilizados em problemas de previsão e classificação de doenças foliares. Os algoritmos apresentados a seguir são amplamente utilizados na literatura, mas não necessariamente serão utilizados neste trabalho.

2.4.1.1 Convolutional Neural Networks

As CNNs constituem uma das arquiteturas mais relevantes do DL, sendo amplamente utilizadas em tarefas de processamento e análise de imagens. Diferentemente das redes neurais tradicionais, que tratam cada entrada de forma independente, as CNNs exploram a estrutura espacial dos dados visuais, permitindo a detecção de padrões locais e a extração progressiva de características hierárquicas. A Figura 1 ilustra, de forma esquemática, esse fluxo: à esquerda, as camadas convolucionais extraem mapas de características; à direita, as camadas totalmente conectadas transformam essas representações em saídas (por exemplo, classes).

Figura 1 – Arquitetura típica de uma CNN, com blocos de *camadas convolucionais* (Conv layers) e blocos de *camadas totalmente conectadas* (FC layers).



Fonte: [Li et al. \(2022\)](#)

O funcionamento das CNNs é baseado em três componentes principais:

1. **Camadas convolucionais:** responsáveis por aplicar filtros (ou kernels) que percorrem a imagem, extraíndo padrões visuais como bordas, texturas e formas geométricas. Cada filtro é aprendido automaticamente durante o treinamento, possibilitando que a rede identifique características relevantes para a tarefa. Essas camadas correspondem ao bloco “Conv layers” na Figura 1.
2. **Camadas de agrupamento:** utilizadas para reduzir a dimensionalidade dos mapas de ativação gerados nas convoluções, preservando as informações mais importantes e diminuindo o custo computacional. Normalmente são intercaladas com as convoluções no lado esquerdo do diagrama.
3. **Camadas totalmente conectadas:** localizadas nas etapas finais, responsáveis pela tomada de decisão, transformando as representações extraídas em previsões ou classificações, como mostrado no bloco “FC layers” da Figura 1.

Um aspecto fundamental das CNNs é a sua invariância a translações, rotações e variações de iluminação, características comuns em bases de dados reais. Essa robustez é particularmente importante na agricultura de precisão, em que imagens de folhas podem ser capturadas em condições diversas, tanto em campo quanto em laboratório.

O impacto das CNNs no avanço da visão computacional foi consolidado em 2012, quando o modelo AlexNet venceu o ILSVRC, alcançando desempenho significativamente superior às técnicas tradicionais da época. Desde então, as CNNs se tornaram a arquitetura dominante em tarefas visuais, sendo a base para o desenvolvimento de arquiteturas mais complexas, como VGGNet, ResNet e InceptionNet.

Na agricultura, as CNNs têm sido aplicadas em diferentes problemas, incluindo a classificação de espécies vegetais, a identificação de pragas e o diagnóstico de doenças foliares. Em um estudo, [Mohanty, Hughes e Salathé \(2016\)](#) utilizaram CNNs para classificar imagens de folhas em 26 doenças diferentes distribuídas em 14 culturas, obtendo uma acurácia superior a 99%. Esses resultados demonstram o potencial das CNNs como ferramentas eficazes para apoiar agricultores na detecção precoce de doenças, contribuindo para a redução de perdas agrícolas e para a promoção de uma agricultura mais sustentável.

Assim, as CNNs consolidam-se como a base de modelos modernos de visão computacional, sendo amplamente utilizadas em aplicações que exigem análise de imagens em larga escala, incluindo o contexto agrícola e o reconhecimento de doenças em plantas.

2.4.1.2 VGGNet

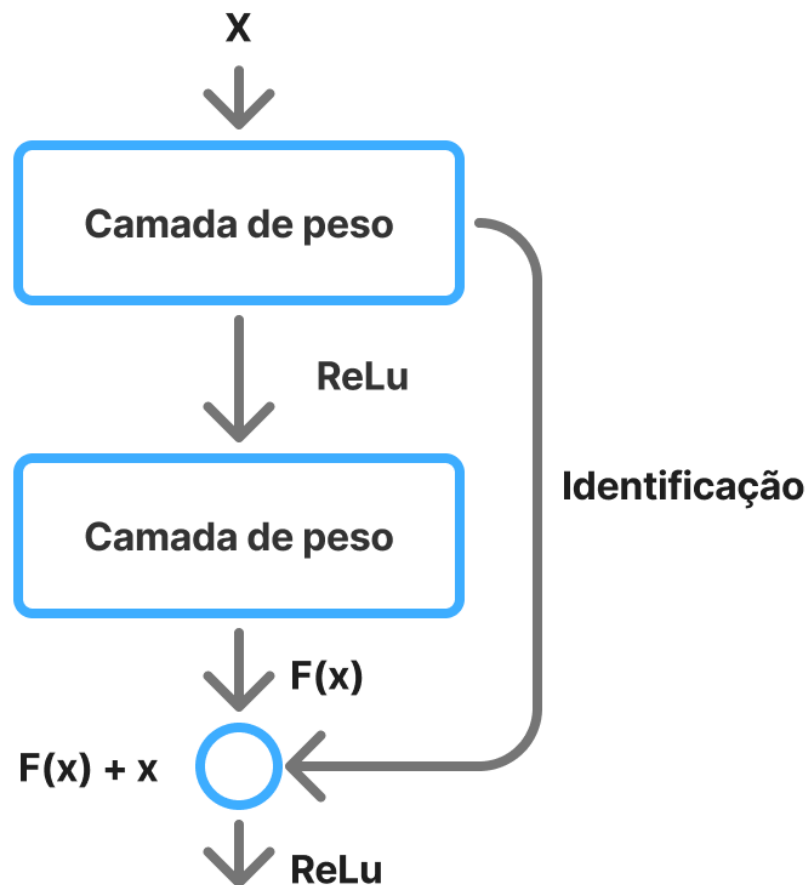
A arquitetura VGGNet (Visual Geometry Group Network), proposta por [Simonyan e Zisserman \(2014\)](#), destacou-se por introduzir o uso sistemático de filtros convolucionais de tamanho reduzido (3x3), empilhados em múltiplas camadas, seguidos de camadas de agrupamento. Essa abordagem mostrou que redes mais profundas, ainda que com operações simples, podem alcançar uma representação altamente expressiva dos dados visuais.

Apesar de exigir elevado poder computacional e memória, a VGGNet consolidou-se como um marco no desenvolvimento de arquiteturas de visão computacional, servindo de base para inúmeros trabalhos subsequentes. Em aplicações agrícolas, especialmente no reconhecimento de doenças foliares, a VGGNet foi amplamente utilizada em fases iniciais de pesquisa, devido à sua eficiência na extração de padrões visuais complexos presentes nas folhas. Estudos demonstraram que, mesmo com custos computacionais mais altos, a arquitetura é capaz de atingir resultados competitivos, tornando-se uma referência no uso de CNN para tarefas de classificação de imagens em larga escala.

2.4.1.3 Residual Networks

As Redes Residuais (ResNet), introduzidas por He et al. (2016), representaram um marco na evolução das arquiteturas de DL. Seu principal diferencial foi a introdução das conexões residuais, também chamadas de *skip connections*, que permitem que o gradiente seja propagado diretamente através das camadas, reduzindo de forma significativa o problema do desaparecimento ou explosão do gradiente em redes muito profundas. Essa inovação possibilitou o treinamento de modelos com centenas ou até milhares de camadas, algo impraticável em arquiteturas anteriores. A Figura 2 ilustra um bloco residual: a transformação aprendida $F(x)$ (duas camadas com ReLU) é somada à entrada original x por uma conexão de identificação, produzindo $F(x) + x$, seguida de nova ativação ReLU.

Figura 2 – Bloco residual típico da ResNet.



Fonte: Sajwan e Mishra (2023)

Na prática, a ResNet funciona aprendendo funções residuais em vez de funções diretas. Em outras palavras, ao invés de aprender a transformação $H(x)$ de uma entrada x , a rede aprende a diferença $F(x) = H(x) - x$, sendo então combinada com a entrada original através

da conexão residual mostrada na Figura 2. Essa estratégia simplifica o processo de otimização e favorece a generalização, resultando em modelos mais robustos e menos suscetíveis a *overfitting*.

A ResNet alcançou resultados revolucionários no desafio ImageNet de 2015, superando modelos anteriores como VGGNet e GoogLeNet, estabelecendo novos patamares de acurácia em classificação e detecção de objetos. Desde então, suas variantes (ResNet-18, ResNet-34, ResNet-50, ResNet-101 e ResNet-152) tornaram-se arquiteturas de referência em diversas áreas da visão computacional, incluindo análise biomédica, reconhecimento facial e, mais recentemente, aplicações em agricultura digital.

No contexto do diagnóstico de doenças foliares, a ResNet se destaca por sua capacidade de extrair características visuais profundas e hierárquicas, mesmo em cenários de grande variabilidade, como diferentes condições de iluminação, estágios de desenvolvimento da planta ou sobreposição de folhas. Estudos recentes têm demonstrado sua eficácia: Ferentinos (2018) treinou CNNs baseadas em arquiteturas profundas, incluindo variantes da ResNet, para o reconhecimento de 58 classes de doenças e pragas em 25 culturas diferentes, atingindo acurácia superior a 99%. Da mesma forma, Too et al. (2019) realizaram uma análise comparativa entre diferentes arquiteturas de CNNs aplicadas à detecção de doenças em plantas e concluíram que modelos baseados em ResNet apresentaram desempenho competitivo, conciliando acurácia elevada com eficiência computacional.

Portanto, a ResNet não apenas solucionou um problema histórico do DL, mas também se consolidou como uma das arquiteturas mais eficazes para classificação de imagens no campo da agricultura 4.0, oferecendo suporte robusto a sistemas de apoio à decisão em ambientes reais de produção agrícola.

2.4.1.4 GoogLeNet

A InceptionNet, também conhecida como GoogLeNet, foi um marco no avanço das CNNs ao vencer o desafio ImageNet em 2014, superando arquiteturas anteriores como a AlexNet e a VGGNet (SZEGEDY et al., 2015). Seu principal diferencial foi a introdução dos módulos *Inception*, que permitem a execução paralela de convoluções com diferentes tamanhos de filtros (1x1, 3x3 e 5x5), além de operações de agrupamento, dentro de uma mesma camada. Essa abordagem possibilita a extração de informações em múltiplas escalas, aumentando a expressividade do modelo sem elevar drasticamente o custo computacional.

Outro aspecto relevante do GoogLeNet é sua profundidade: com 22 camadas treináveis, a arquitetura conseguiu ser mais eficiente que a VGGNet, utilizando menos parâmetros graças ao uso extensivo de convoluções 1×1 , que reduzem a dimensionalidade antes da aplicação de filtros mais custosos. Essa otimização permitiu que a rede fosse mais leve e viável para aplicações práticas, mesmo em ambientes com restrições computacionais.

No campo da visão computacional aplicada à agricultura, a GoogLeNet tem se destacado

em tarefas como classificação de doenças foliares, detecção de pragas e identificação de espécies de plantas. Por exemplo, [Mohanty, Hughes e Salathé \(2016\)](#) utilizaram arquiteturas como GoogLeNet para classificar imagens de folhas com diferentes doenças, obtendo acurácia superior a 99%. De forma semelhante, [Sladojevic et al. \(2016\)](#) aplicaram uma rede baseada em GoogLeNet para reconhecer 13 tipos de doenças em folhas de plantas, também alcançando resultados altamente satisfatórios.

Com o passar do tempo, a InceptionNet evoluiu em versões subsequentes (Inception v2, v3 e v4), que incorporaram melhorias como normalização em camadas intermediárias, fatoração de convoluções e conexões residuais, combinando ideias que também inspiraram arquiteturas como a ResNet. Apesar do surgimento de modelos mais recentes e sofisticados, o GoogLeNet continua sendo uma arquitetura relevante, tanto pelo seu impacto histórico quanto pela sua eficiência em aplicações agrícolas, onde a disponibilidade de recursos computacionais pode ser limitada.

2.4.1.5 MobileNet

A MobileNet foi projetada para aplicações em dispositivos com recursos computacionais limitados, como smartphones, drones e sistemas embarcados, mantendo alta eficiência e desempenho em tarefas de classificação e detecção de imagens ([HOWARD et al., 2017](#)). Sua principal inovação é a utilização das convoluções separáveis em profundidade (*depthwise separable convolutions*), que dividem a operação convolucional tradicional em duas etapas: uma convolução *depthwise*, que aplica um filtro a cada canal de entrada separadamente, seguida por uma convolução *pointwise* 1x1, que combina os resultados de cada canal. Essa abordagem reduz drasticamente o número de parâmetros e operações, permitindo que a rede seja muito mais leve e rápida, sem comprometer significativamente a acurácia.

A arquitetura MobileNet também introduz parâmetros de ajuste, como o multiplicador de largura e o multiplicador da resolução, que permitem balancear o trade-off entre precisão e eficiência computacional, adaptando o modelo a diferentes cenários e dispositivos. Dessa forma, é possível reduzir ainda mais o tamanho da rede ou acelerar o processamento, conforme a necessidade da aplicação.

No contexto da agricultura de precisão, a MobileNet tem se mostrado especialmente útil em sistemas de detecção de doenças foliares em tempo real, em que imagens de folhas são capturadas diretamente em campo, utilizando smartphones ou drones. Estudos demonstram que a MobileNet consegue alcançar acurácia comparável a redes maiores, como VGGNet ou ResNet, enquanto mantém a viabilidade para dispositivos móveis, tornando-a ideal para soluções de monitoramento agrícola de baixo custo ([SARMADI; NEJAD; SAMAVI, 2018](#)).

Além disso, a MobileNet tem sido amplamente utilizada como base para modelos de transferência de aprendizado, nos quais pesos pré-treinados em grandes bases de dados, como ImageNet, são adaptados para tarefas específicas de classificação de plantas e detecção de

doenças. Essa abordagem permite desenvolver sistemas precisos mesmo com conjuntos de dados relativamente pequenos, acelerando o desenvolvimento de aplicações práticas.

Portanto, a MobileNet combina eficiência computacional, flexibilidade e robustez, consolidando-se como uma das arquiteturas mais adequadas para integração em aplicações móveis e de campo, alinhando-se perfeitamente ao conceito de Agricultura 4.0 e à necessidade de soluções acessíveis e escaláveis para o monitoramento de culturas agrícolas.

3 Revisão da Literatura

A rápida evolução de técnicas de DL e a difusão de dispositivos móveis têm impulsionado aplicações de visão computacional na agricultura, especialmente no diagnóstico de doenças foliares. Este capítulo apresenta os trabalhos relacionados a aplicativos móveis e estudos de uso de CNNs para detecção ou quantificação de doenças. O objetivo é delinear o estado da arte e identificar lacunas quanto à generalização dos modelos, robustez em condições reais de campo e usabilidade das soluções.

A aplicação de técnicas de IA na agricultura de precisão tem se mostrado fundamental para a automação de diagnósticos fitossanitários. Uma área de especial relevância é a identificação de doenças em folhas de tomateiro, onde a precisão e a agilidade são cruciais para evitar perdas na produção. Nesse cenário, as CNNs destacam-se como a tecnologia mais promissora, dado seu desempenho notável em tarefas de visão computacional, que pode, inclusive, superar a acurácia da percepção humana.

Um exemplo prático dessa aplicação é o trabalho de [Elhassouny e Smarandache \(2019\)](#), que desenvolveram um aplicativo móvel inteligente especificamente para o reconhecimento de doenças em folhas de tomate utilizando CNNs. O estudo deles demonstra a viabilidade de embarcar modelos de DL em dispositivos móveis, criando uma ferramenta de diagnóstico acessível que pode ser utilizada diretamente no campo pelo produtor.

Aplicativos móveis têm sido empregados tanto para quantificação de severidade quanto para detecção e classificação de doenças. O *Leaf Doctor* destacou-se como solução pioneira e amplamente citada para quantificar a severidade de sintomas foliares em imagens de campo ([PETHYBRIDGE; NELSON, 2015](#)). Na linha de diagnóstico assistido por IA, estudos com mandioca investigaram desde o treinamento de modelos de DL até sua implantação em aplicativos móveis e posterior avaliação em campo; resultados indicam que o sistema alcançou desempenho comparável ou superior ao de usuários humanos em determinadas tarefas ([RAMCHARAN et al., 2017](#)).

Além de aplicações móveis, diversos trabalhos mostraram a eficácia das CNNs diretamente em bases de imagens de plantas. O estudo de [Mohanty, Hughes e Salathé \(2016\)](#), um dos mais referenciados na área, avaliou o desempenho de CNNs em mais de 50 mil imagens de folhas de 14 espécies de plantas com 26 doenças distintas, atingindo taxas de acurácia superiores a 99% em condições controladas. Resultados semelhantes foram encontrados por [Ferentinos \(2018\)](#), que treinou CNNs em um conjunto de 87 mil imagens, alcançando acurácias acima de 99% para diagnóstico de diferentes culturas agrícolas.

Outro trabalho relevante é o de [Sladojevic et al. \(2016\)](#), que aplicou CNNs no reconhecimento automático de 13 doenças em 25 espécies de plantas. Apesar de utilizar um

conjunto de dados relativamente limitado, os resultados demonstraram o potencial da técnica para automatizar diagnósticos, reforçando o papel das redes profundas no apoio a agricultores.

Outro estudo relevante é o de [Shrestha, Das e Dey \(2020\)](#), que aplicou técnicas de CNN para a detecção de doenças em diferentes culturas, incluindo pimentão, batata e tomate. O modelo foi treinado com 15 classes, abrangendo tanto folhas saudáveis quanto doentes, com sintomas de diferentes patologias como *Early Blight* (Requeima precoce), *Late Blight* (Requeima tardia), *Leaf Mold* (Mofo da folha) e infestações por ácaros. A acurácia de teste alcançada foi de 88,80%, demonstrando a viabilidade do uso de CNNs em cenários multiclasse. Além disso, os autores exploraram métricas de desempenho e análise da área infectada, o que contribui para uma avaliação mais abrangente do sistema em termos de robustez e aplicabilidade prática.

Mais recentemente, pesquisas têm buscado explorar arquiteturas mais leves para cenários de campo. Por exemplo, [Falaschetti et al. \(2022\)](#) demonstram a viabilidade de CNNs compactas integradas a *toolchains* como TensorFlow Lite para execução em dispositivos de baixo custo, permitindo inferência *on-device*. Esse direcionamento é essencial para superar restrições de conectividade em áreas rurais, viabilizando o uso prático de IA na agricultura de precisão.

A tabela 1 apresenta um comparativo entre os principais trabalhos identificados na literatura, destacando os tipos de dados utilizados e as técnicas empregadas para o reconhecimento de doenças foliares.

Tabela 1 – Comparativo dos trabalhos relacionados sobre detecção de doenças em plantas.

Referência	Tipo de Dados	Técnicas Utilizadas
Elhassouny e Smarandache (2019)	Imagens de folhas de tomateiro capturadas por dispositivo móvel	CNN embarcada em aplicativo móvel para diagnóstico em campo.
Pethybridge e Nelson (2015)	Imagens de folhas com sintomas em diferentes culturas agrícolas	Análise de severidade foliar por segmentação e quantificação de área afetada (não utiliza CNN).
Ramcharan et al. (2017)	Imagens de folhas de mandioca coletadas em campo	CNNs treinadas para classificação de doenças e implantação em aplicativo móvel com teste em campo.
Mohanty, Hughes e Salathé (2016)	54.306 imagens de 14 espécies de plantas com 26 doenças (<i>PlantVillage</i>)	CNNs para classificação multiclasse de doenças foliares; abordagem pioneira em DL aplicado à agricultura.
Ferentinos (2018)	87.000 imagens de diversas culturas agrícolas	CNN profundas para diagnóstico automático de doenças em múltiplas espécies.
Sladojevic et al. (2016)	13 doenças em 25 espécies de plantas	CNNs para reconhecimento automático de doenças com base em imagens foliares; uso de conjunto de dados reduzido.
Shrestha, Das e Dey (2020)	Imagens de folhas de pimenta, batata e tomate (15 classes)	CNNs para detecção multiclasse de doenças; análise de métricas de desempenho e área infectada.
Falaschetti et al. (2022)	Imagens de folhas de campo (diversas culturas)	CNNs compactas integradas ao TensorFlow Lite para inferência em dispositivos móveis de baixo custo (<i>on-device</i>).

Fonte: Elaborada pelo autor.

3.0.1 Síntese dos Trabalhos Relacionados

A análise dos estudos apresentados na tabela 1 evidencia uma evolução significativa nas abordagens de diagnóstico de doenças foliares. Inicialmente, os trabalhos focavam em métodos de segmentação e quantificação de área lesionada, como observado em [Pethybridge e Nelson \(2015\)](#). Com o avanço das CNNs, observou-se uma transição para modelos de DL, que passaram a dominar as pesquisas subsequentes.

Os estudos de [Mohanty, Hughes e Salathé \(2016\)](#), [Ferentinos \(2018\)](#) e [Sladojevic et al. \(2016\)](#) consolidaram as CNNs como ferramentas robustas para a classificação de doenças, especialmente em ambientes controlados e com bases de dados extensas. Em contrapartida, trabalhos mais recentes como os de [Ramcharan et al. \(2017\)](#), [Elhassouny e Smarandache \(2019\)](#) e [Falaschetti et al. \(2022\)](#) avançaram em direção à aplicação prática, explorando a integração entre modelos de IA e dispositivos móveis, o que representa um passo crucial para o

uso em campo.

Nota-se também um movimento em direção à leveza e eficiência dos modelos, buscando compatibilidade com dispositivos de baixo custo e operação *offline*, conforme destacado em [Falaschetti et al. \(2022\)](#). Essa tendência reforça a necessidade de modelos mais generalizáveis, capazes de lidar com variações de iluminação, ângulos e ruídos típicos de ambientes agrícolas reais.

Em síntese, os trabalhos revisados revelam que, embora as CNNs tenham atingido excelente desempenho em bases controladas, ainda existem lacunas quanto à sua aplicação em cenários reais de campo e quanto à usabilidade das soluções por agricultores, especialmente em contextos com restrições de conectividade e recursos computacionais.

4 Material e Métodos

Este capítulo descreve os materiais utilizados e os métodos empregados no desenvolvimento do aplicativo para detecção de doenças foliares, incluindo o conjunto de dados, as linguagens e ferramentas, a metodologia de treinamento dos modelos de ML e a forma de avaliação dos modelos de DL e do sistema em geral.

4.1 Conjunto de dados

O conjunto de dados utilizado neste trabalho consiste em imagens de folhas com diferentes condições de saúde, incluindo folhas saudáveis e folhas afetadas por diversas doenças.

- **Fonte:** Para a análise das imagens, foi utilizado o conjunto de dados *PlantVillage*, originalmente apresentado por [Mohanty, Hughes e Salathé \(2016\)](#) e disponibilizado publicamente na plataforma Kaggle ([BHATTARAI, 2020](#)). No total, foram utilizadas 87.829 imagens distribuídas em 38 pastas rotuladas para as etapas de treinamento, validação e teste.
- **Pré-processamento:** Na etapa de pré-processamento, todas as imagens do conjunto de dados foram submetidas a uma sequência de transformações. Primeiramente, as imagens foram redimensionadas para uma resolução uniforme de 224×224 pixels, visando padronizar as dimensões de entrada para o modelo. Em seguida, foi aplicada a conversão das imagens para tensores do PyTorch, que também normaliza os valores dos pixels do intervalo original de $[0, 255]$ para um novo intervalo de $[0.0, 1.0]$.
- **Formato das imagens:** As imagens do conjunto de dados, originalmente no formato .JPG, foram processadas no espaço de cores RGB (3 canais).

4.2 Linguagens e Ferramentas

Para o desenvolvimento do aplicativo e dos modelos de DL, foram utilizadas as seguintes linguagens e ferramentas:

- **Linguagens de programação:** Python para o desenvolvimento da IA e do servidor; e Dart/Flutter para o aplicativo móvel.
- **Frameworks e bibliotecas:** Matplotlib, pandas, numpy, os, pickle, torch, random, cupertino_icons, image_picker, http e flutter_dotenv.

- **Ambiente de desenvolvimento:**
 - **Desenvolvimento do Aplicativo Móvel:** foram utilizados o Visual Studio Code e o Android Studio, com o Dart SDK (Versão 3.6.0) e o Flutter (Versão 3.27.0) executando no sistema operacional Windows 10.
 - **Treinamento e Execução dos modelos de IA:** foram realizados em ambiente Python (Versão 3.10.16), também no Windows 10, utilizando o Visual Studio Code como IDE principal.
- **Hardware utilizado:** O treinamento foi realizado em um desktop com processador Intel Core i3-10105F, placa de vídeo Nvidia GTX 1050 de 2GB e 16GB de RAM. Os testes no aplicativo foram feitos em um Galaxy S20FE (Snapdragon 865, 6GB RAM) e em um emulador do Pixel 7a no Android Studio.

4.3 Treinamento e Avaliação dos Modelos

O treinamento dos modelos foi conduzido de forma iterativa, explorando duas arquiteturas de CNN: ResNet50 e MobileNetV2. A abordagem experimental consistiu em ajustar sistematicamente hiperparâmetros e técnicas de regularização, como a taxa de aprendizado (*learning rate*), *Dropout* e *label smoothing*, buscando otimizar o desempenho preditivo. A técnica de ajuste fino foi aplicada, inicialmente treinando apenas as camadas finais das redes e, em experimentos posteriores, descongelando camadas intermediárias e profundas para permitir um ajuste mais refinado aos dados específicos do problema. A rede foi ajustada para 38 classes, correspondentes a 14 espécies de plantas, incluindo classes de folhas saudáveis quando disponíveis.

O treinamento final utilizou o otimizador *Adam*, com taxa de aprendizado (*learning rate*) de 0,0005 e *batch size* de 64. A função de ativação adotada nas camadas internas foi a *ReLU*, e a função de perda (*loss function*) foi a *CrossEntropyLoss*. O modelo foi treinado por 15 épocas, o que demonstrou ser suficiente para a convergência sem ocorrência significativa de *overfitting*.

O conjunto de dados foi dividido na proporção de 60% para treinamento, 20% para validação e 20% para teste, seguindo uma prática amplamente adotada em experimentos de DL. Essa divisão visa garantir que o modelo aprenda com uma quantidade representativa de exemplos, enquanto é avaliado em dados não vistos durante o treinamento, permitindo uma estimativa mais confiável de sua capacidade de generalização. Essa estratégia também contribui para reduzir o risco de *overfitting* e aumentar a robustez do modelo em aplicações reais.

4.3.1 Métricas de Avaliação

Para monitorar o processo de treinamento e avaliar o desempenho final dos modelos, foram utilizadas as seguintes métricas quantitativas:

- **Função de Perda (Loss Function):** A otimização dos modelos foi guiada pela função de perda *CrossEntropyLoss*, adequada para problemas de classificação multi-classe. A minimização do valor de perda (*Loss*) ao longo das épocas de treinamento foi o principal indicador de que o modelo estava aprendendo o mapeamento desejado. A evolução do *Training Loss* e do *Validation Loss* foi utilizada para monitorar e mitigar problemas de *overfitting*.
- **Acurácia (Accuracy):** A acurácia foi a principal métrica escolhida para avaliar o desempenho geral do modelo no conjunto de validação. Ela representa a proporção de predições corretas sobre o total de amostras e é definida formalmente pela Equação (4.1):

$$\text{Acurácia} = \frac{\text{Número de Predições Corretas}}{\text{Número Total de Amostras}} \quad (4.1)$$

A maximização da acurácia de validação (*Val Acc*) foi o objetivo central do processo de otimização.

- **Métricas Adicionais:** Para uma análise mais detalhada do comportamento do modelo em cada classe, foram empregadas a Matriz de Confusão e métricas derivadas dela, como Precisão (*Precision*), Revocação (*Recall*) e F1-Score. Essas ferramentas são essenciais para identificar classes que o modelo tende a confundir e para avaliar o desempenho em cenários com possível desbalanceamento de dados.

4.4 Avaliação do Sistema Integrado

A etapa final de avaliação focou na análise do sistema completo em operação. Para isso, foi mensurada a latência ponto a ponto, que corresponde ao tempo total decorrido desde o envio de uma imagem pelo cliente até o recebimento da predição. Este teste foi conduzido utilizando a ferramenta Postman para simular requisições à API do *backend*, permitindo medir o tempo de resposta em um cenário de uso realista. Os componentes analisados dentro dessa latência foram: o tempo de transferência de dados na rede, o processamento da requisição pelo servidor e o tempo de inferência do modelo.

5 Resultados

Este capítulo apresenta os resultados obtidos tanto no treinamento dos modelos de DL quanto na avaliação de desempenho do sistema integrado. Primeiramente, são detalhados os resultados dos experimentos iterativos com as arquiteturas ResNet50 e MobileNet. Em seguida, são apresentados os dados de latência do sistema em uma simulação de uso real.

5.1 Desempenho dos Modelos de DL

A performance dos modelos foi avaliada por meio de uma série de experimentos, ajustando-se hiperparâmetros e arquiteturas com o objetivo de maximizar a acurácia de validação. Os resultados são apresentados a seguir.

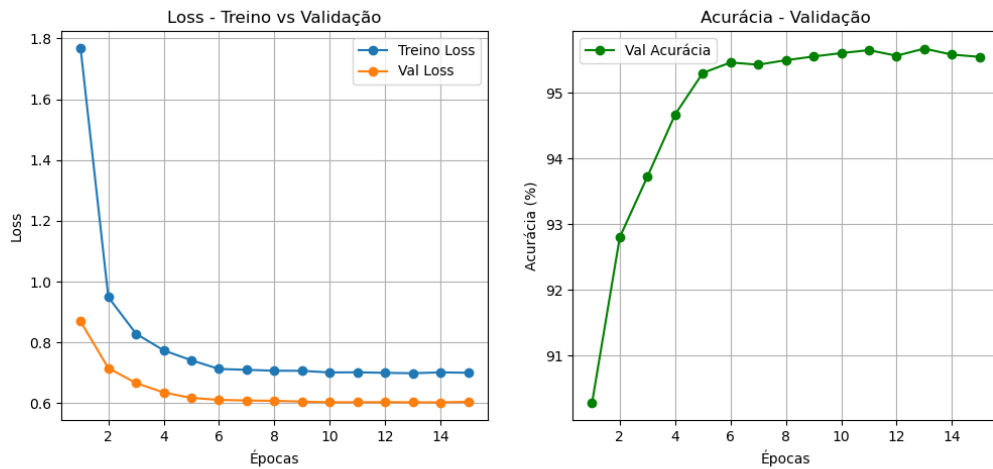
5.1.1 Experimentos com a Arquitetura ResNet50

Os testes iniciais foram conduzidos utilizando a arquitetura ResNet50. Foram realizadas quatro iterações principais para otimizar seu desempenho.

5.1.1.1 Teste 1: ConFiguração Inicial

O primeiro experimento com a ResNet50 estabeleceu uma linha de base, com um *Dropout* de 0.5 e *label_smoothing* de 0.05, alcançou uma acurácia de validação de 95,5%. Conforme observado na Figura 3, a curva de perda de treinamento (*Treino Loss*) indicou a possibilidade de melhorias, ao estabilizar em um valor próximo a 0.7.

Figura 3 – Resultados do Teste 1 com ResNet50.

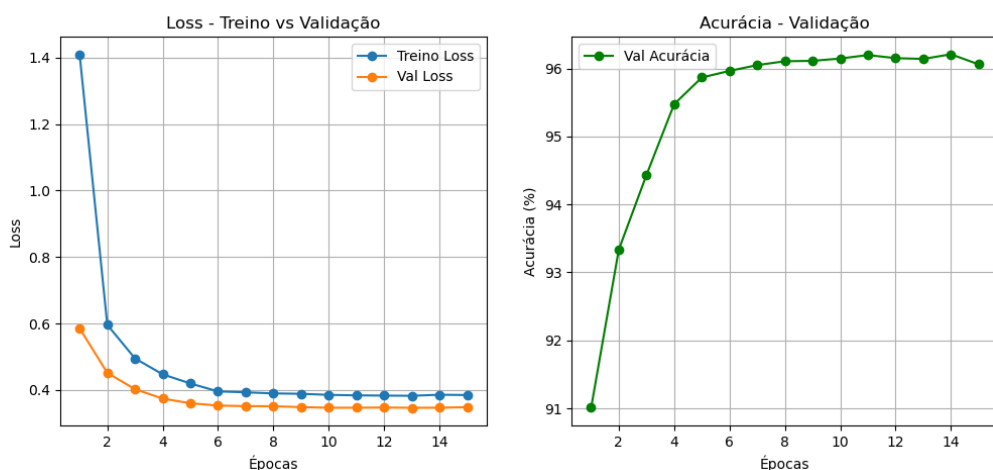


Fonte: Elaborada pelo autor.

5.1.1.2 Teste 2: Redução da Regularização

No segundo teste, os hiperparâmetros de regularização foram suavizados, com o *Dropout* reduzido para 0.3 e o *label_smoothing* para 0.02. Tais ajustes resultaram em um ganho modesto de performance, elevando a acurácia de validação para 96,21%. A Figura 4 demonstra que, apesar do aumento na acurácia, a perda de validação (*Val Loss*) permaneceu próxima à de treinamento.

Figura 4 – Resultados do Teste 2 com ResNet50, após ajuste de Dropout e Label Smoothing.



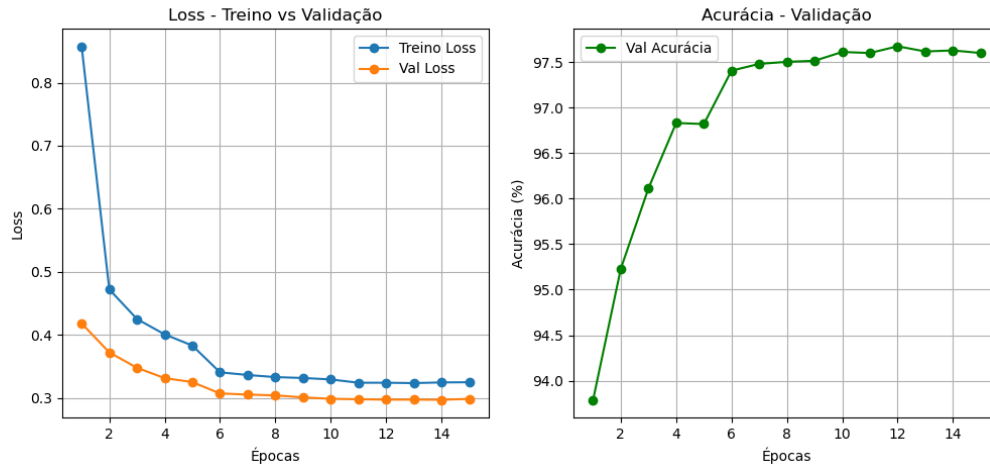
Fonte: Elaborada pelo autor.

5.1.1.3 Teste 3: Aumento da Taxa de Aprendizado

A terceira iteração envolveu o aumento da taxa de aprendizado (*learning rate*) de 0.001 para 0.005. Essa modificação gerou um avanço expressivo, impulsionando a acurácia de

validação para 97,67%. Os gráficos na Figura 5 mostram uma convergência mais rápida do modelo para um menor valor de perda.

Figura 5 – Resultados do Teste 3 com ResNet50, com taxa de aprendizado aumentada.

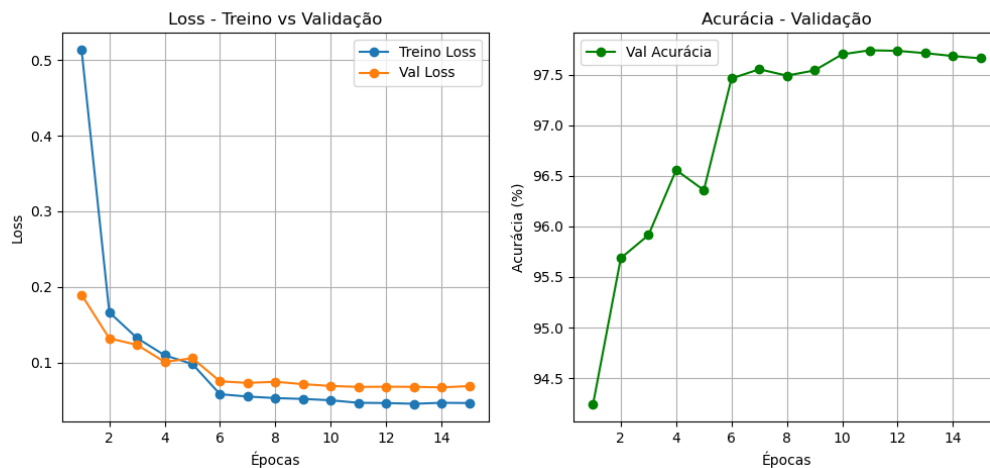


Fonte: Elaborada pelo autor.

5.1.1.4 Teste 4: Ajuste Fino Agressivo

No último teste com esta arquitetura, adotou-se uma abordagem de ajuste fino mais agressiva, descongelando as camadas layer3 e layer4 e removendo as técnicas de regularização (*Dropout* e *label_smoothing*). O resultado foi a maior acurácia obtida com este modelo, atingindo 97,74%. A Figura 6 revela que a perda de treinamento se tornou notavelmente inferior à de validação, sugerindo um leve sobreajuste (*overfitting*) em troca do ganho de desempenho.

Figura 6 – Resultados do Teste 4 com ResNet50, após descongelar mais camadas e remover regularização.



Fonte: Elaborada pelo autor.

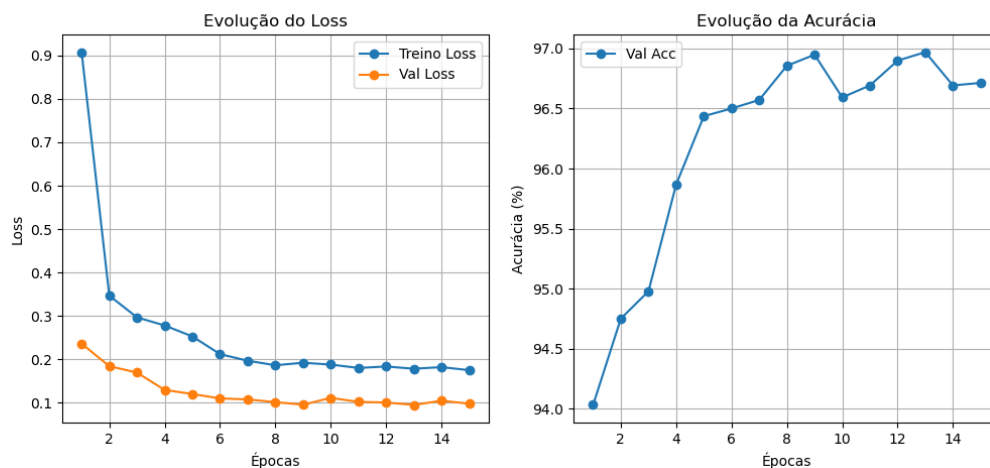
5.1.2 Experimentos com a Arquitetura MobileNet

Com o objetivo de encontrar um modelo mais leve e eficiente, foram conduzidos experimentos com a arquitetura MobileNet, cujos resultados são detalhados a seguir.

5.1.2.1 Teste 1: ConFiguração Inicial com MobileNet

O primeiro experimento com a MobileNet estabeleceu um ponto de partida para a otimização. Conforme exibido na Figura 7, o modelo alcançou uma acurácia de validação estável em torno de 96,8%. Embora seja um resultado robusto, as curvas de aprendizado sugerem que havia espaço para melhorias, motivando ajustes nos hiperparâmetros para o teste seguinte.

Figura 7 – Resultados do Teste 1 com MobileNet, estabelecendo a linha de base.

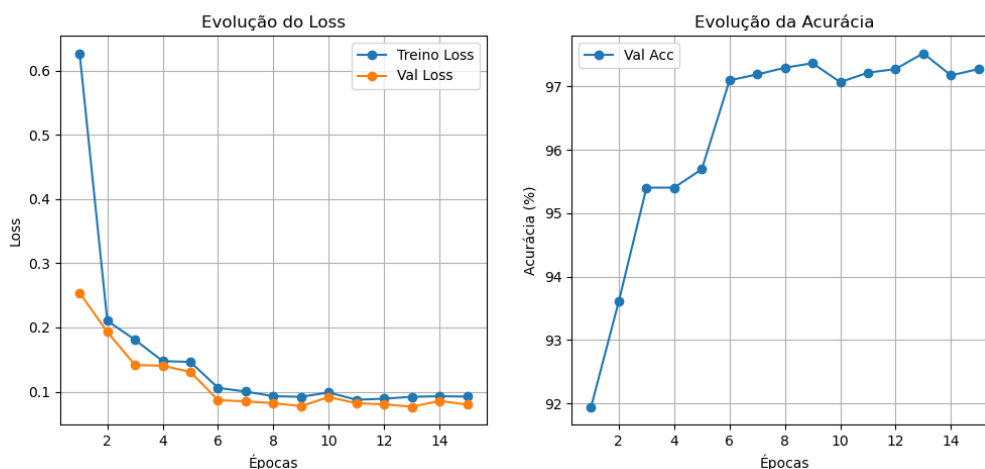


Fonte: Elaborada pelo autor.

5.1.2.2 Teste 2: Otimização de Hiperparâmetros

No segundo teste, após um ajuste fino nos hiperparâmetros, o desempenho do modelo melhorou significativamente. Os resultados, apresentados na Figura 8, mostram uma convergência mais eficaz, com a acurácia de validação ultrapassando 97% e se estabilizando em um patamar superior ao do teste anterior. Isso demonstra que a arquitetura MobileNet respondeu positivamente à otimização, consolidando-se como uma alternativa eficiente.

Figura 8 – Resultados do Teste 2 com MobileNet, após otimização.



Fonte: Elaborada pelo autor.

5.1.3 Comparação ResNet50 vs MobileNetV2

Ao comparar os resultados finais de ambas as arquiteturas, é evidente que o modelo ResNet50 alcançou uma performance superior em todas as métricas de avaliação. A tabela 2 detalha essa comparação, mostrando que a ResNet50 obteve uma vantagem consistente de aproximadamente 1,5% em acurácia, precisão, revocação e F1-score em relação à MobileNetV2.

Apesar da superioridade numérica da ResNet50, a arquitetura MobileNetV2 foi a escolhida para a implementação final no aplicativo móvel. Esta decisão é fundamentada em um compromisso estratégico entre a acurácia do modelo e sua eficiência computacional. A MobileNetV2 é uma arquitetura projetada especificamente para ambientes com recursos limitados, como smartphones, oferecendo vantagens cruciais:

- **Leveza e Eficiência:** O modelo possui um número significativamente menor de parâmetros, o que resulta em um arquivo de modelo mais leve e menor consumo de memória RAM.
- **Velocidade de Inferência:** Requer menos poder de processamento, permitindo que a classificação de imagens seja executada mais rapidamente no dispositivo do usuário e com menor impacto no consumo da bateria.

Dessa forma, a pequena redução na acurácia é considerada um *trade-off* aceitável em troca da viabilidade técnica, da performance de execução e da melhor experiência do usuário em uma plataforma móvel.

Tabela 2 – Comparativo de Métricas de Desempenho: ResNet50 vs. MobileNetV2.

Modelo	Precisão	Revocação	F1-Score	Acurácia
ResNet50	0,9900	0,9900	0,9900	0,9901
MobileNetV2	0,9749	0,9746	0,9746	0,9749

Fonte: Elaborada pelo autor.

5.1.3.1 Classificação Detalhada e Matriz de Confusão

Uma vez que o modelo MobileNetV2 foi selecionado para a aplicação final devido à sua eficiência, esta seção se aprofunda em uma análise detalhada de seu desempenho no conjunto de teste. O objetivo é compreender a performance do modelo em cada uma das classes individuais e identificar possíveis pontos de dificuldade.

A fim de complementar a análise quantitativa, avaliou-se o desempenho preditivo por classe, conforme apresentado na tabela 3.

Tabela 3 – Relatório de Classificação Detalhado do Modelo MobileNetV2 (Conjunto de Teste)

Classe	Precisão	Revocação	F1-score	Suporte
Maçã - Sarna	0,9876	0,9876	0,9876	404
Maçã - Podridão negra	0,9950	0,9950	0,9950	398
Maçã - Ferrugem do cedro	0,9943	0,9858	0,9900	352
Maçã - Saudável	0,9875	0,9851	0,9863	402
Mirtilo - Saudável	0,9864	0,9973	0,9918	364
Cereja - Oídio	0,9940	0,9881	0,9911	337
Cereja - Saudável	0,9945	0,9945	0,9945	366
Milho - Mancha de cercospora	0,9675	0,9058	0,9356	329
Milho - Ferrugem comum	0,9948	0,9948	0,9948	382
Milho - Queima do norte	0,9254	0,9738	0,9490	382
Milho - Saudável	1,0000	1,0000	1,0000	372
Uva - Podridão negra	0,9764	0,9868	0,9816	378
Uva - Esca (Sarampo negro)	0,9869	0,9792	0,9830	384
Uva - Queima da folha	0,9971	0,9971	0,9971	345
Uva - Saudável	0,9912	1,0000	0,9956	339
Laranja - Huanglongbing (Greening)	1,0000	0,9975	0,9988	402
Pêssego - Mancha bacteriana	0,9944	0,9701	0,9821	368
Pêssego - Saudável	0,9857	0,9942	0,9899	346
Pimentão - Mancha bacteriana	0,9973	0,9817	0,9895	383
Pimentão - Saudável	0,9680	0,9874	0,9776	398

Continua na próxima página

Tabela 3 – continuação

Classe	Precisão	Revocação	F1-score	Suporte
Batata - Requeima precoce	0,9872	0,9923	0,9897	388
Batata - Requeima tardia	0,9815	0,9562	0,9687	388
Batata - Saudável	0,9833	0,9699	0,9766	365
Framboesa - Saudável	0,9944	1,0000	0,9972	357
Soja - Saudável	0,9828	0,9877	0,9852	405
Abóbora - Oídio	1,0000	1,0000	1,0000	348
Morango - Queimadura da folha	0,9944	1,0000	0,9972	355
Morango - Saudável	0,9973	1,0000	0,9986	365
Tomate - Mancha bacteriana	0,9676	0,9648	0,9662	341
Tomate - Requeima precoce	0,8912	0,8958	0,8935	384
Tomate - Requeima tardia	0,9160	0,9111	0,9135	371
Tomate - Mofo da folha	0,9627	0,9576	0,9601	377
Tomate - Mancha de septoria	0,9318	0,8997	0,9155	349
Tomate - Ácaros-aranha	0,9296	0,9456	0,9375	349
Tomate - Mancha alvo	0,9036	0,8962	0,8999	366
Tomate - Vírus do enrolamento	0,9871	0,9746	0,9808	393
Tomate - Vírus do mosaico	0,9491	0,9888	0,9685	358
Tomate - Saudável	0,9623	0,9922	0,9770	386
Acurácia			0,9749	14076
Média Macro	0,9749	0,9746	0,9746	14076
Média Ponderada	0,9749	0,9749	0,9748	14076

Fonte: Elaborada pelo autor.

Observa-se que o modelo apresentou um desempenho geral equilibrado, com as médias de precisão, revocação e F1-score próximas de 97,5%, indicando elevada consistência nas predições. A acurácia global de 97,49% reforça a eficiência do modelo em distinguir corretamente as diferentes classes foliares.

Para validar visualmente a distribuição dos acertos e erros especificamente nas culturas de milho e tomate, foi gerada a Matriz de Confusão correspondente, apresentada na Figura 9. Essa matriz permite observar o comportamento do classificador em cada categoria individualmente, destacando o número de predições corretas na diagonal principal e os eventuais casos de confusão entre classes dentro dessas duas culturas.

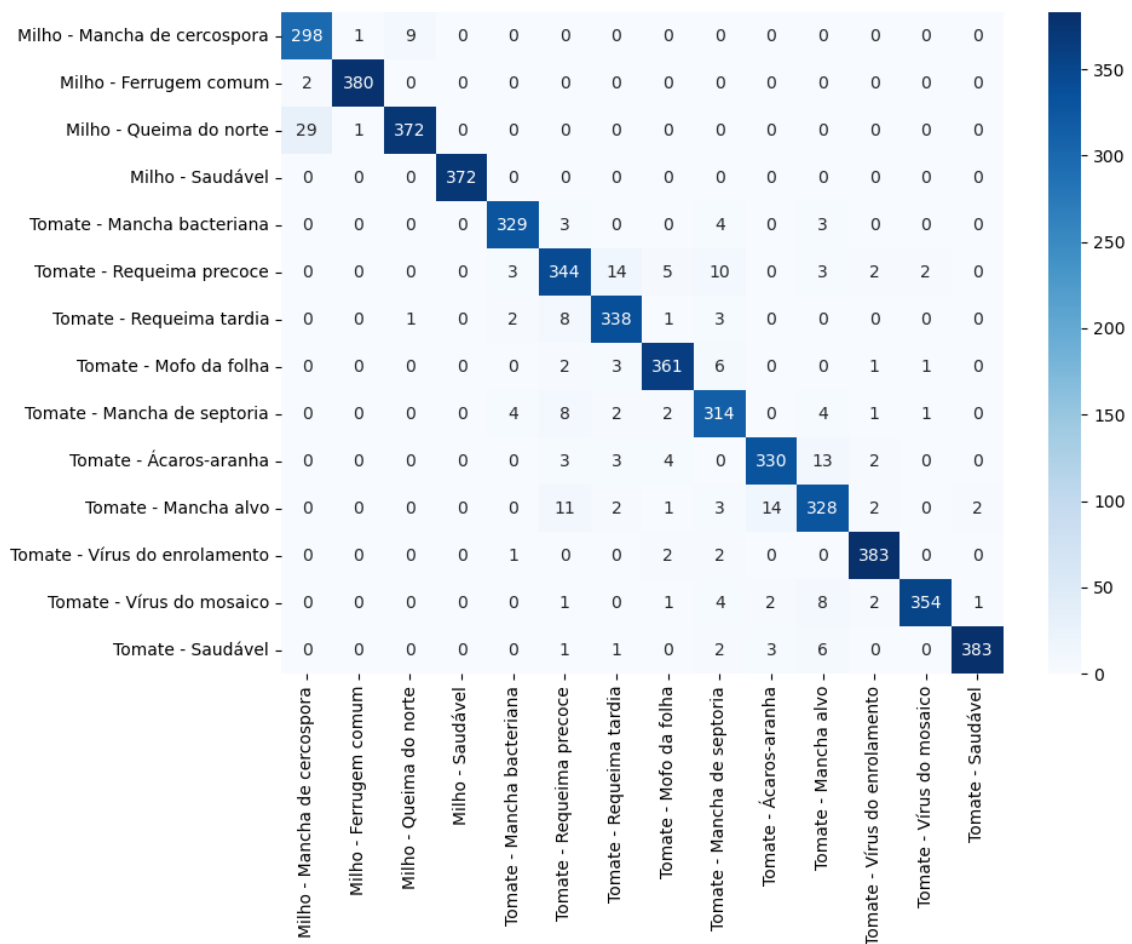
Uma análise detalhada da matriz revela que o modelo alcançou uma alta taxa de acerto para a maioria das classes de milho e tomate. No entanto, é notável que a maior parte dos

erros de classificação ocorreu entre doenças da mesma cultura. No caso do milho, por exemplo, observam-se confusões entre as classes “Milho – Mancha de cercospora”, “Milho – Ferrugem comum” e “Milho – Queima do norte”, que apresentam padrões visuais relativamente similares.

Da mesma forma, para o tomate, o modelo apresentou dificuldade em distinguir entre doenças como “Tomate – Requeima precoce”, “Tomate – Requeima tardia” e “Tomate – Mancha alvo”, além de pequenas confusões envolvendo “Tomate – Mancha de septoria” e “Tomate – Ácaros-aranha”. Essas sobreposições refletem a elevada semelhança morfológica entre algumas patologias, mesmo para observadores humanos treinados.

Esse padrão sugere que, embora o modelo seja eficaz em identificar corretamente as culturas e em distinguir a maior parte das doenças foliares presentes nas classes de milho e tomate, ele encontra maior desafio quando precisa diferenciar doenças que compartilham características visuais muito próximas dentro de uma mesma espécie.

Figura 9 – Matriz de Confusão do modelo MobileNetV2 no conjunto de teste.



Fonte: Elaborada pelo autor.

5.2 Desempenho do Sistema Integrado

A avaliação final concentrou-se no desempenho do sistema completo, medindo a latência ponto a ponto desde o envio da imagem pelo cliente até o recebimento da predição. Utilizando o Postman para simular requisições à API, o tempo de resposta foi quantificado.

O resultado obtido foi um tempo médio de resposta de aproximadamente 100 milissegundos (ms). Este valor agrega a latência de rede, o processamento da requisição pelo servidor Flask e o tempo de inferência do modelo.

Este baixo tempo de resposta é considerado excelente para uma aplicação interativa, validando as escolhas arquitetônicas do projeto. Em especial, a utilização do modelo MobileNet provou-se ideal, pois além de alcançar a maior acurácia, sua eficiência computacional permitiu inferências rápidas, viabilizando a aplicação em tempo real.

5.3 Interface e Funcionalidade do Aplicativo Móvel

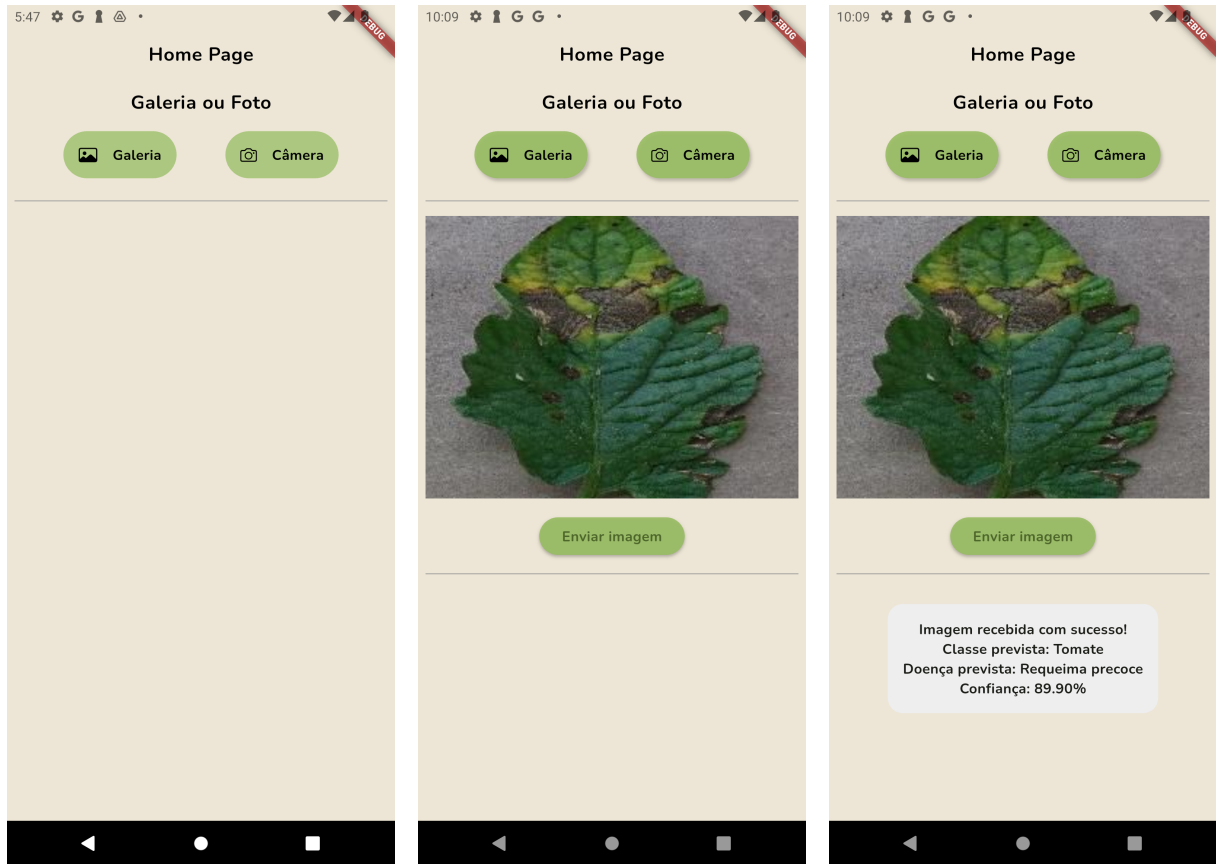
Como resultado final do desenvolvimento, foi criado um aplicativo móvel funcional que integra o modelo de DL para a detecção de doenças foliares. A interface foi projetada para ser intuitiva, guiando o usuário por um fluxo simples desde a captura da imagem até a obtenção do diagnóstico. As telas principais da aplicação são apresentadas a seguir.

A tela inicial, exibida na Figura 10a, serve como portal de entrada para o usuário. Ela apresenta o propósito do aplicativo e contém dois botões centralizados, "Galeria" e "Câmera", que permitem ao usuário escolher uma foto da galeria do seu dispositivo ou capturar uma nova imagem utilizando a câmera.

Após a seleção ou captura, a imagem da folha é exibida na tela para confirmação, como mostrado na Figura 10b. Esta etapa permite que o usuário verifique se a imagem escolhida está adequada para a análise. Um botão "Enviar imagem" é então disponibilizado para que o usuário envie a imagem ao servidor, que processará a requisição e executará a inferência com o modelo MobileNet treinado.

Finalmente, a tela de resultados (Figura 10c) apresenta o diagnóstico fornecido pelo modelo. A classe e a doença prevista são exibidas de forma clara, juntamente com a confiança da predição. Este retorno rápido e direto cumpre o objetivo central do projeto de fornecer uma ferramenta de auxílio acessível para a identificação de doenças em plantas.

Figura 10 – Telas do aplicativo desenvolvido: (a) tela inicial, (b) seleção da imagem e (c) resultados obtidos.



(a) Tela inicial.

(b) Seleção da folha.

(c) Tela de resultados.

Fonte: Elaborada pelo autor.

6 Conclusão

Este TCC propôs e implementou um sistema integrado para reconhecimento de doenças foliares. O sistema é composto por um modelo de DL, a arquitetura **MobileNetV2**, que foi selecionada por sua eficiência e disponibilizada via API em Flask para ser consumida por um aplicativo móvel em Flutter. O objetivo geral — viabilizar a identificação de doenças a partir de imagens capturadas pelo usuário — foi atingido, abrangendo desde a seleção e o pré-processamento do conjunto de dados até o treinamento do modelo, a construção da API e a avaliação do desempenho de todo o sistema.

O estudo comparou a robusta arquitetura ResNet50 com a eficiente MobileNetV2, aplicando ajuste fino e ajustes progressivos de hiperparâmetros. Embora a ResNet50 tenha alcançado uma acurácia de validação ligeiramente superior (97,74%), a **MobileNetV2 foi a arquitetura escolhida para a implementação final** devido ao seu excelente balanço entre performance e eficiência computacional. Sua leveza e inferência rápida são características essenciais para uma boa experiência de usuário em smartphones, que possuem hardware e capacidade de bateria limitados. A acurácia de 97,49% obtida pela MobileNetV2 foi considerada altamente competitiva e adequada para o propósito da aplicação. Métricas complementares, como a matriz de confusão, precisão, revocação e F1-Score, esclareceram o comportamento do modelo por classe e orientaram os ajustes.

Os resultados sustentam a hipótese central: é possível desenvolver um aplicativo móvel, integrado ao modelo MobileNetV2, capaz de identificar doenças foliares com precisão suficiente para apoiar a tomada de decisão de agricultores. A acurácia obtida, combinada ao desenho do sistema e à experiência de uso, indica viabilidade técnica e potencial de impacto prático, sobretudo para pequenos e médios produtores com acesso limitado à assistência especializada.

As contribuições deste trabalho incluem: uma pipeline completa e replicável que cobre preparação de dados, treinamento e ajuste fino de CNNs, disponibilização via API (Flask) e consumo móvel (Flutter); a validação da MobileNetV2 como uma arquitetura que oferece um compromisso ideal entre acurácia e eficiência para diagnósticos em campo; e um artefato funcional (app + *backend*) que democratiza o acesso ao diagnóstico assistido por IA e abre caminho para integrações com fluxos agrônômicos.

Algumas limitações devem ser reconhecidas: generalização possivelmente limitada, dado que o conjunto de dados pode não refletir plenamente variações de campo (iluminação, ângulos, estágios fenológicos, dispositivos); classes visualmente similares que elevam confusões em condições adversas; dependência de conectividade para inferência no servidor; e ausência de calibração de probabilidades e estudo de *thresholds* para diferentes custos de erro.

Como desdobramentos futuros, sugerem-se: coleta e curadoria de imagens de campo

para reduzir *domain shift*, com *data augmentation* orientada ao contexto agrícola; inferência no dispositivo por meio de *quantization* e *pruning* (p.ex., TensorFlow Lite), reduzindo latência e dependência de rede; aprendizado ativo e atualização contínua do modelo, com realimentação a partir de novas amostras rotuladas por especialistas; explicabilidade (p.ex., Grad-CAM) para apoiar interpretação do diagnóstico; calibração de probabilidades e definição de *thresholds* por cultura/doença; módulos de recomendação agrônômica e trilhas educativas; avaliação de usabilidade e estudos de campo mensurando impacto em tempo de resposta, perdas e uso de insumos; e suporte a uso offline com fila de requisições e sincronização, além de políticas de privacidade e segurança adequadas ao contexto rural.

Em síntese, a integração do modelo MobileNetV2 a um aplicativo móvel configura uma alternativa tecnicamente sólida para apoiar o diagnóstico de doenças foliares. A acurácia alcançada e a arquitetura proposta indicam viabilidade de transferência para cenários reais, desde que acompanhadas por expansão do conjunto de dados com imagens de campo, mecanismos de atualização do modelo e cuidados de usabilidade e conectividade. Esta pesquisa contribui para a agricultura de precisão ao oferecer um caminho prático para diagnósticos rápidos e confiáveis, fomentando decisões mais assertivas e sustentáveis.

Referências

BHATTARAI, S. *New Plant Diseases Dataset*. 2020. <<https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset>>. Acessado em: 12 agosto 2025.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.

CHASSIN, L. *Detecting and managing crop pests and diseases with AI: Insights from Plantix*. 2025. GSMA AgriTech blog.

DUNCOMBE, R. Mobile phones for agricultural and rural development: A literature review and suggestions for future research. *The European Journal of Development Research*, Springer, v. 28, n. 2, p. 213–235, 2016.

ELHASSOUNY, A.; SMARANDACHE, F. Smart mobile application to recognize tomato leaf diseases using convolutional neural networks. In: IEEE. *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*. [S.l.], 2019. p. 1–4.

FALASCHETTI, L.; MANONI, L.; LEO, D. D.; PAU, D.; TOMASELLI, V.; TURCHETTI, C. A cnn-based image detector for plant leaf diseases classification. *HardwareX*, Elsevier, v. 12, p. e00363, 2022.

FERENTINOS, K. P. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, Elsevier, v. 145, p. 311–318, 2018.

GAO, M.; LIN, L.; SINNOTT, R. O. A mobile application for plant recognition through deep learning. In: IEEE. *2017 IEEE 13th International Conference on e-Science (e-Science)*. [S.l.], 2017. p. 29–38.

GONZÁLEZ-RODRÍGUEZ, V. E.; IZQUIERDO-BUENO, I.; CANTORAL, J. M.; CARBÚ, M.; GARRIDO, C. Artificial intelligence: A promising tool for application in phytopathology. *Horticulturae*, v. 10, n. 3, 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016.

HINOJOSA, C.; SANCHEZ, K.; CAMACHO, A.; ARGUELLO, H. Agrotic: Bridging the gap between farmers, agronomists, and merchants through smartphones and machine learning. *arXiv preprint arXiv:2305.12418*, 2023.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

JIANG, H. t. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture*, v. 11, n. 8, p. 707, 2021.

- KAUR, R.; SINGH, S.; GARG, A.; SINGH, J. Role of drones in agriculture: A review. *Agricultural Reviews*, Agricultural Research Communication Centre, v. 43, n. 3, p. 295–302, 2022.
- LEGG, J.; HUGHES, D. *New mobile app diagnoses crop diseases in the field and alerts rural farmers*. 2017. Penn State News / CGIAR.
- LI, Z.; LIU, F.; YANG, W.; PENG, S.; ZHOU, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, v. 33, n. 12, p. 6999–7019, 2022.
- MCCARTHY, J.; MINSKY, M. L.; ROCHESTER, N.; SHANNON, C. E. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, v. 27, n. 4, p. 12–12, 2006.
- MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, frontiers, v. 7, p. 215232, 2016.
- MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, Volume 7 - 2016, 2016. ISSN 1664-462X. Disponível em: <<https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2016.01419>>.
- PETHYBRIDGE, S. J.; NELSON, S. C. Leaf doctor: A new portable application for quantifying plant disease severity. *Plant disease*, Am Phytopath Society, v. 99, n. 10, p. 1310–1316, 2015.
- RAMCHARAN, A.; BARANOWSKI, K.; MCCLOSKEY, P.; AHMED, B.; LEGG, J.; HUGHES, D. P. Deep learning for image-based cassava disease detection. *Frontiers in plant science*, Frontiers Media SA, v. 8, p. 1852, 2017.
- RODER, M.; GOMES, N.; YOSHIDA, A.; PAPA, J. P.; COSTEN, F. Multimodal convolutional deep belief networks for stroke classification with fourier transform. In: IEEE. *2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2023. p. 163–168.
- RODRÍGUEZ, J. L. H.; DÁVILA, J. F. R.; CUEVAS, R. S.; LÓPEZ, D. d. J. P.; JIMÉNEZ, F. B. G. Desenvolvimento de um aplicativo móvel para o manejo sustentável de café. *Pesquisa Agropecuária Tropical*, p. e81329, jun. 2025. Acessado em: 16 novembro 2025. Disponível em: <<https://revistas.ufg.br/pat/article/view/81329>>.
- RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: A modern approach; [the intelligent agent book]*. [S.l.]: Prentice hall, 1995.
- SAJWAN, A.; MISHRA, G. Comparative analysis of resnet and densenet for differential cryptanalysis of speck 32/64 lightweight block cipher. In: SPRINGER. *International Conference on Cryptology & Network Security with Machine Learning*. [S.l.], 2023. p. 495–504.
- SARDOGAN, M.; TUNCER, A.; OZEN, Y. Plant leaf disease detection and classification based on cnn with lvq algorithm. In: IEEE. *2018 3rd international conference on computer science and engineering (UBMK)*. [S.l.], 2018. p. 382–385.
- SARMADI, A.; NEJAD, A.; SAMAVI, S. Mobilenet-based approach for plant leaf disease detection in real-time applications. *Computers and Electronics in Agriculture*, Elsevier, v. 156, p. 510–518, 2018.

SHRESTHA, G.; DAS, M.; DEY, N. t. Plant disease detection using cnn. In: IEEE. *2020 IEEE applied signal processing conference (ASPCON)*. [S.l.], 2020. p. 109–113.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SLADOJEVIC, S.; ARSENOVIC, M.; ANDERLA, A.; CULIBRK, D.; STEFANOVIC, D. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, Hindawi, v. 2016, p. 3289801, 2016.

SUTTON, R. S.; BARTO, A. G. t. *Reinforcement learning: An introduction*. [S.l.]: MIT press Cambridge, 1998. v. 1.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.]: IEEE, 2015. 1–9 p.

TOO, E. C.; YUJIAN, L.; NJUKI, S.; YINGCHUN, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, Elsevier, v. 161, p. 272–279, 2019.

WOLFERT, S.; GE, L.; VERDOUW, C.; BOGAARDT, M.-J. Big data in smart farming—a review. *Agricultural systems*, Elsevier, v. 153, p. 69–80, 2017.

