UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

ELISA ALVES VELOSO

APLICAÇÃO DE APRENDIZAGEM FEDERADA NA IDENTIFICAÇÃO DE PLANTAS DANINHAS

Ouro Preto
2026

ELISA ALVES VELOSO

APLICAÇÃO DE APRENDIZAGEM FEDERADA NA IDENTIFICAÇÃO DE PLANTAS DANINHAS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Rodrigo César Pedrosa Silva

Ouro Preto, MG
2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Elisa Alves Veloso

APLICAÇÃO DE APRENDIZAGEM FEDERADA NA IDENTIFICAÇÃO DE PLANTAS DANINHAS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 13 de Julho de 2026.

Membros da banca

Rodrigo César Pedrosa Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Rodolfo Ayala Lopes (Examinador) - Doutor - Confederação Nacional das Cooperativas do Sicoob
Ederson Naves Fernandes Gonçalves Junior (Examinador) - Bacharel - Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto

Rodrigo César Pedrosa Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 13/07/2026.

Documento assinado eletronicamente por **Rodrigo Cesar Pedrosa Silva**, **COORDENADOR(A) DO CURSO DE BACHARELADO EM INTELIGENCIA ARTIFICIAL**, em 14/07/2026, às 09:48, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1137642** e o código CRC **88B01CF7**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Agradecimentos

A conclusão desta etapa não teria sido possível sem o auxílio de pessoas especiais. À minha família, base de tudo, expresso minha eterna gratidão pelo amor incondicional e incentivo nos dias difíceis. Aos meus amigos, companheiros de jornada, agradeço por tornarem essa caminhada mais leve e alegre.

Ao meu orientador, professor Rodrigo Silva, pelas valiosas orientações e por acompanhar de perto cada etapa deste trabalho. Sou profundamente grata pelo apoio não apenas no desenvolvimento deste Trabalho de Conclusão de Curso, mas durante toda a minha trajetória na graduação. Agradeço por ter me dado a primeira oportunidade de pesquisa científica na UFOP e por ter sido o responsável por despertar o meu interesse pela área de Inteligência Artificial.

Resumo

A agricultura de precisão voltada para a cultura da cana-de-açúcar tem recebido investimentos expressivos no cenário brasileiro. Nesse contexto, um dos principais desafios das grandes produções é a infestação de plantas daninhas, que competem com a lavoura principal e comprometem a produtividade. Para mitigar o crescimento dessas espécies e otimizar os custos via aplicação localizada de herbicidas, investiga-se a identificação inteligente de plantas invasoras no campo. Atualmente, as soluções predominantes baseiam-se em modelos centralizados de aprendizado de máquina, que apresentam alta performance, mas exigem o envio de imagens para um servidor central, expondo dados corporativos e operacionais sensíveis do produtor rural. Como alternativa para a preservação da privacidade, o Aprendizado Federado *(Federated Learning)* propõe o treinamento local nos dispositivos dos clientes, compartilhando apenas as atualizações de pesos com um servidor agregador para a construção de um modelo global. Este trabalho propõe e avalia um ecossistema de aprendizado federado para a identificação de plantas daninhas, comparando-o com abordagens centralizadas tradicionais. Os resultados experimentais baseados na arquitetura DarkNet53 demonstraram que, enquanto o treinamento centralizado atingiu níveis de Acurácia e Recall próximos a 98%, o cenário federado de distribuição homogênea (IID) obteve desempenho altamente competitivo, superando 90% em ambas as métricas. Ademais, no desafiador cenário heterogêneo (Non-IID), a aplicação do algoritmo de agregação FedProx evitou a divergência dos clientes locais, garantindo a convergência estável do classificador global e validando a técnica como uma solução segura, privada e eficiente para a identificação agrícola em larga escala.

Palavras-chave: Aprendizado Federado. Modelos Distribuídos. Visão Computacional Agrícola. Detecção de Plantas Daninhas. Agricultura de Precisão. Aprendizado Profundo.

Abstract

Precision agriculture focused on sugarcane crops has received significant investments in the Brazilian scenario. In this context, one of the main challenges for large-scale production is weed infestation, which competes with the main crop and compromises crop yield. To mitigate the growth of these species and optimize costs through the localized application of herbicides, the intelligent identification of invasive plants in the field is investigated. Currently, prevailing solutions are based on centralized machine learning models, which show high performance but require sending images to a central server, exposing sensitive corporate and operational data of the rural producer. As a privacy-preserving alternative, Federated Learning proposes local training on client devices, sharing only weight updates with an aggregating server to build a global model. This work proposes and evaluates a federated learning ecosystem for weed identification, comparing it with traditional centralized approaches. Experimental results based on the DarkNet53 architecture demonstrated that, while centralized training achieved Accuracy and Recall levels close to 98%, the homogeneous distribution federated scenario (IID) achieved highly competitive performance, exceeding 90% in both metrics. Furthermore, in the challenging heterogeneous scenario (Non-IID), the application of the FedProx aggregation algorithm prevented the divergence of local clients, ensuring the stable convergence of the global classifier and validating the technique as a secure, private, and efficient solution for large-scale agricultural identification.

Keywords: Federated Learning. Distributed Models. Agricultural Computer Vision. Weed Detection. Precision Agriculture. Deep Learning.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

AF	Aprendizagem Federada.
AgTech	Empresas de Tecnologia Agrícola.
AgTechs	Empresas de Tecnologia Agrícola.
AM	Aprendizado de Máquina.
API	Interface de Programação de Aplicações.
ATR	Açúcar Total Recuperável.
BD	Big Data.
BN	*Batch Normalization*.
CNNs	Redes Neurais Convolucionais.
FedAvg	*Federated Averaging*.
FL	*Federated Learning*.
IA	Inteligência Artificial.
IBGE	Instituto Brasileiro de Geografia e Estatística.
IID	*Independent and Identically Distributed*.
IoT	Internet das Coisas.
MAE	Erro Médio Absoluto.
ML	Machine Learning.
MLP	Multilayer Perceptron.
MSE	Erro Quadrático Médio.
NAS	*Neural Architecture Search*.
ReLU	Rectified Linear Unit.
RNAs	Redes Neurais Artificiais.
RPC	*Remote Procedure Call*.
SGD	*Stochastic Gradient Descent*.
XOR	Lógica OU Exclusivo.

Lista de Símbolos

240×320	Dimensões originais das imagens do *dataset* (largura $\times$ altura)
224×224	Dimensão de redimensionamento de imagens no cenário centralizado
128×128	Dimensão de redimensionamento de imagens no cenário federado
$[0, 1]$	Intervalo de normalização dos valores de *pixels*
α	Inclinação do *LeakyReLU*
x	Variável de ativação/entrada
x_j	Entrada total (pré-ativação) do neurônio j
j	Índice do neurônio atual
y_i	Saída da unidade i da camada anterior
i	Índice da unidade da camada anterior
w_{ji}	Peso sináptico entre a unidade i e o neurônio j
$bias_j$	Termo de viés associado ao neurônio j
$f(z)$	Função de ativação (forma geral)
z	Variável de entrada da função de ativação
e	Base do logaritmo natural (na Sigmoid)
Δw	Variação/atualização do peso
η	Taxa de aprendizado
E	Função objetivo/erro (na derivada parcial)
$\frac{\partial E}{\partial w}$	Derivada parcial do erro em relação ao peso
$\mathcal{F}(x)$	Função de resíduo no bloco residual
$H(x)$	Função de mapeamento desejada no bloco residual
$F(x)$	Função residual definida como $H(x) - x$
y	Saída do bloco residual

$\{W_i\}$	Conjunto de pesos do bloco residual
ϕ	Coeficiente composto de escalonamento (EfficientNet)
t	Índice da rodada de comunicação (FL)
w_t	Modelo global na rodada t
k	Índice do cliente
P_k	Conjunto de dados local do cliente k
$F_k(w)$	Função de perda local do cliente k
Δw_k	Atualização de pesos produzida pelo cliente k
w_{t+1}	Modelo global após agregação (rodada $t+1$)
K	Número total de clientes
n_k	Volume de dados do cliente k
n	Total de dados na rede
S_t	Conjunto de clientes selecionados na rodada t
m_t	Total de exemplos nos clientes selecionados na rodada t
μ	Hiperparâmetro proximal (FedProx)
$\|w - w_t\|^2$	Norma quadrática da diferença entre pesos locais e globais

Sumário

1 Introdução

Saccharum spp. é o nome científico que designa o gênero botânico das canas-de-açúcar. São gramíneas tropicais gigantes, da família Poaceae, reconhecidas por seus caules ricos em sacarose, sendo a base fundamental para a produção global de açúcar, etanol, bioenergia e suplemento forrageiro (Miranda, 2015).

O cultivo da cana-de-açúcar desempenha um papel fundamental no agronegócio global, destacando-se não apenas por seus altos índices de produtividade, mas também por ser um dos setores que mais impulsionam a adoção de tecnologias de agricultura de precisão. No Brasil, essa cultura desempenha um papel central na história, sendo uma das primeiras produções estabelecidas no período colonial. Ao longo dos anos, consolidou-se como uma das principais atividades agrícolas do país, passando por diversos ciclos de expansão e modernização (Senior Sistemas, 2023). O impacto direto dessa expansão sobre a dimensão territorial colhida ao longo das últimas três décadas está ilustrado na Figura 1.1. Os dados evidenciam o forte ciclo de expansão da cultura agroindustrial entre 2000 e 2014, seguido por um período de consolidação e estabilização do patamar produtivo acima de 10 milhões de hectares.

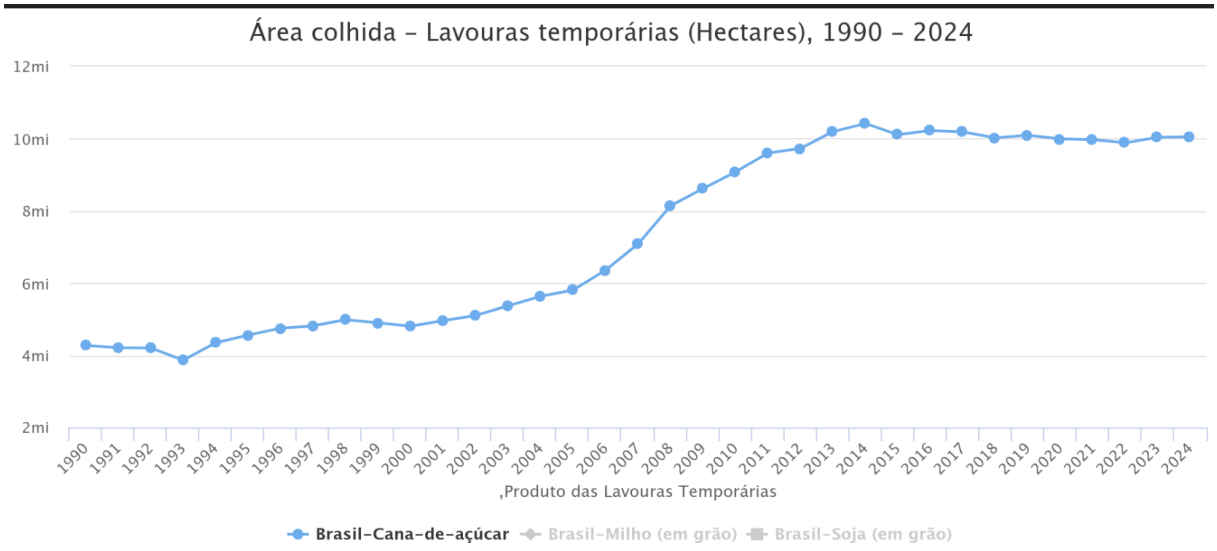

Figura 1.1 – Gráfico demonstrando a série histórica da área colhida de lavouras temporárias de cana-de-açúcar no Brasil (1990–2024). Fonte: (Instituto Brasileiro de Geografia e Estatística, 2024).

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), o valor da produção açucareira nacional em 2024 ultrapassou 100 bilhões de reais, enquanto as áreas destinadas ao cultivo, concentradas principalmente na região Sudeste, somam mais de 10 milhões de hectares (Figura 1.2). Assim sendo, desde o início do século, o crescimento do setor sucroenergético brasileiro tem impulsionado atividades de pesquisa e desenvolvimento tecnológico no país, que

visam a otimização da produção e uso da matéria prima em produtos, tais como o biocombustível etanol e o açúcar, e subprodutos como o bagaço e a torta de filtro (Nascimento et al., 2024).

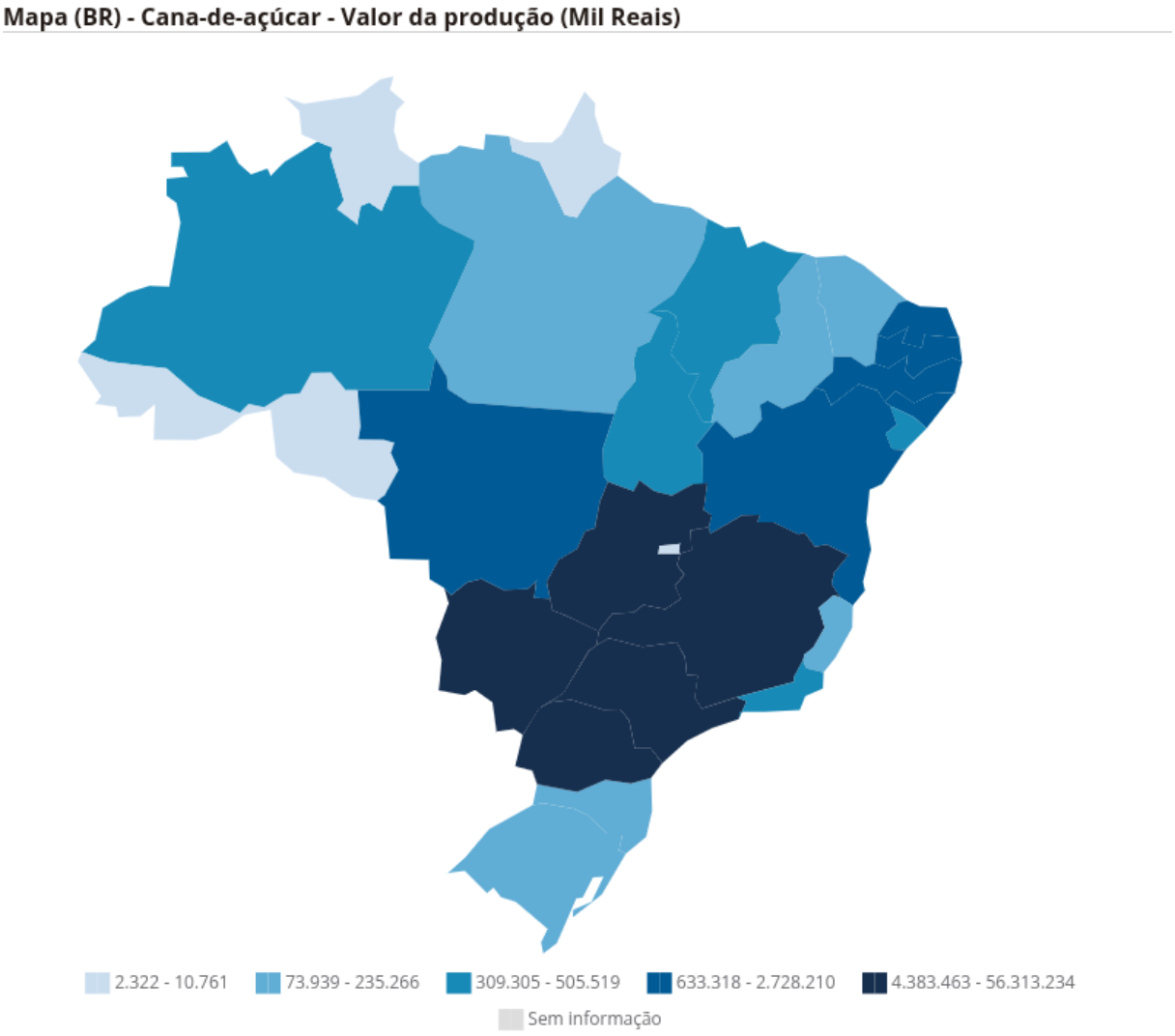

Figura 1.2 – Distribuição espacial do valor da produção de cana-de-açúcar por estados brasileiros, em milhares de reais. Fonte: (IBGE, 2024).

Neste contexto, a expansão territorial e produtiva da cana-de-açúcar desencadeia transformações diretas nas esferas ambiental e social das regiões produtoras. Do ponto de vista ambiental, a atividade demanda a supressão de vegetação nativa e o uso intensivo de insumos químicos, fatores que alteram a dinâmica físico-química do solo e dos recursos hídricos. No aspecto socioeconômico, a lavoura impulsiona a geração de empregos e a movimentação financeira local (Cunha; Pasqualetto, 2022). Dessa forma, a cultura da cana-de-açúcar possui elevada relevância econômica para o mercado brasileiro, ao mesmo tempo em que essa produção gera impactos socioeconômicos e ambientais em escala global.

A agricultura de precisão caracteriza-se pela aplicação de tecnologias para gerenciar a variabilidade espacial e temporal da produção agrícola, com o objetivo de otimizar o rendimento das safras e preservar a qualidade ambiental (Pierce; Nowak, 1999). Como uma evolução natural dessa prática, surge a agricultura digital. Ao integrar redes de sensores e tecnologia da informação, ela viabiliza a comunicação bidirecional entre gestores e máquinas, fundamentando a tomada de decisão e tornando a operação agrícola muito mais eficiente (Cheema et al., 2023). Dentre as inovações desse setor, a identificação automatizada de plantas daninhas em lavouras de cana-de-açúcar ganha cada vez mais destaque no mercado brasileiro, pois proporciona autonomia, precisão e confiabilidade na aplicação de defensivos. A adoção dessas tecnologias resulta em benefícios diretos: redução no custo de insumos, otimização das operações da fazenda, mitigação de impactos ambientais e, consequentemente, maiores margens de lucro (Cheema et al., 2023).

O termo *AgTech* abrange tanto a integração de tecnologias avançadas ao agronegócio para impulsionar a eficiência e a sustentabilidade no campo, quanto o próprio modelo de *startups* focado no desenvolvimento dessas inovações (Dutia, 2014). Paralelamente, a convergência entre a Internet das Coisas (*IoT*), a computação em nuvem, a robótica e a inteligência artificial expande o fenômeno do *Big Data* (BD) no setor. Trata-se de fluxos de dados massivos e complexos que, embora fundamentais para refinar os processos decisórios, superam a capacidade de gerenciamento dos sistemas tradicionais de processamento. Na agricultura inteligente, o escopo de aplicação do BD reconfigura toda a cadeia produtiva, atuando desde o fornecimento de análises preditivas sobre as operações e o suporte a decisões em tempo real, até o redesenho estrutural dos processos de negócio nas usinas produtoras. Essa intensa digitalização, contudo, altera profundamente os papéis e as dinâmicas de mercado entre os atores do campo, estabelecendo um cenário de assimetrias onde *AgTechs*, fundos de investimento e produtores rurais travam um verdadeiro jogo de poder pelo controle da informação (Wolfert et al., 2017). Essa disputa ganha contornos ainda mais críticos na governança desses ativos intangíveis; conforme advertem (Wiseman et al., 2019), os contratos e termos de adesão estabelecidos por essas plataformas frequentemente transferem direitos excessivos sobre os dados coletados para as empresas de tecnologia, deixando as usinas e os produtores severamente vulneráveis em relação ao uso secundário e não autorizado de suas informações estratégicas (Wiseman et al., 2019).

Em outro contexto, a Aprendizagem Federada (AF) é um conceito introduzido pelo Google em 2016, no qual múltiplos dispositivos aprendem colaborativamente um modelo de aprendizado de máquina sem compartilhar seus dados privados, sob a supervisão de um servidor central; e tem emergido como um paradigma eficaz para a preservação da privacidade durante o treinamento de modelos em ambientes distribuídos, uma vez que apenas os parâmetros e gradientes resultantes dos treinamentos locais são compartilhados com o agregador, sem expor os dados originais. Tal método oferece amplas oportunidades em domínios críticos em que é arriscado compartilhar informações privadas com outras organizações ou dispositivos (Mammen, 2021). Em comparação, o modelo de aprendizado centralizado exige a coleta e o armazenamento de dados em um único servidor, o que aumenta significativamente o risco de exposição e comprometimento das

informações (Kairouz; McMahan, 2021).

Como mencionado anteriormente, o cultivo da cana-de-açúcar é fundamental para o agronegócio brasileiro, uma vez que o país figura entre os principais exportadores desse produto, que constitui a base da produção nacional de açúcar e etanol. Esse setor contribui de forma expressiva para a economia, tanto por meio das exportações quanto pela geração de empregos diretos e indiretos, além do fornecimento de energia renovável. Nesse contexto, o setor sucroenergético consolida-se como um dos pilares do agronegócio e da matriz energética nacional, com o Brasil ocupando posição de liderança mundial na produção de açúcar e etanol (Milanez et al., 2024).

O processo bem-sucedido do cultivo de cana de açúcar envolve diversas etapas, que iniciam no bom preparo químico e físico do solo, bem como a adubação correta. Além disso, há a necessidade de escolher uma variedade adequada da planta, considerando fatores tais como a produtividade, resistência a doenças e pragas e adaptação climática. Os cuidados pós plantio são essenciais para garantir a qualidade do produto colhido e envolvem o manejo integrado de pragas e doenças, a manutenção do solo, a irrigação e o monitoramento contínuo para assegurar o vigor da lavoura (Senior Sistemas, 2023).

Dentre os cuidados pós-plantio, que constituem o foco de aplicação da proposta apresentada no presente trabalho, destaca-se o controle de plantas daninhas. Tal categoria de plantas contempla diversas espécies, como a grama-seda, o capim-colonião e a mamona, que surgem e se desenvolvem devido a uma combinação de fatores, sendo os principais o banco de sementes no solo e as condições ambientais favoráveis. Uma vez estabelecidas, as plantas daninhas competem pelo nicho ecológico, liberando substâncias alelopáticas que podem inibir o brotamento da cana-de-açúcar. Sua presença em alta densidade, formando um tapete, impede que a cultura absorva totalmente os recursos disponíveis, resultando na redução da área foliar e no desperdício de água e nutrientes (Figura 1.3) . Ademais, a convivência da cana-de-açúcar com espécies agressivas, a exemplo do capim-colonião, resulta em uma drástica redução da produtividade de colmos e em severas perdas no acúmulo de biomassa (Kuva et al., 2003). Além do prejuízo em campo, as invasoras geram impactos críticos na qualidade industrial da matéria-prima, uma vez que a interferência fisiológica reduz diretamente o teor de Açúcar Total Recuperável (ATR), que representa a quantidade em quilos de açúcares contidos em uma tonelada de cana que pode ser efetivamente transformada em açúcar ou etanol. Esse índice é o principal parâmetro de qualidade utilizado para precificar a produção. Somado a isso, a biomassa daninha eleva o índice de impurezas na colheita mecanizada, encarecendo o processamento do caldo na usina (Agropecuária, 2024). Assim sendo, o manejo assertivo dessas plantas torna-se fundamental para garantir a rentabilidade e a eficiência do ecossistema sucroenergético.

Figura 1.3 – Matocompetição na cultura da cana-de-açúcar por Ipomoea sp. Fonte: (Elevagro, 2013).

Há também as dificuldades na colheita que relacionam-se à presença das invasoras. Nesse âmbito, as plantas daninhas, especialmente espécies trepadeiras, isto é, plantas de crescimento volúvel que utilizam raízes aéreas e enroscamento para cobrir e sufocar outras, como a “corda-de-viola”, se enroscam nos colmos da cana, dificultando a operação da colheita mecânica. Isso pode levar ao chamado "embuchamento"ou entupimento das máquinas, reduzindo a eficiência e o rendimento operacional. O controle das daninhas exige investimento significativo em herbicidas, mão de obra e equipamentos, elevando o custo final de produção. Por fim, algumas daninhas servem como hospedeiras para pragas e doenças comuns à cana-de-açúcar, ampliando os desafios de manejo sanitário do canavial (Azania et al., 2021).

Tradicionalmente, o controle de plantas daninhas baseia-se principalmente em métodos físicos, mecânicos e, mais recentemente, químicos. O controle manual consiste na remoção física das plantas daninhas do solo, utilizando as mãos ou ferramentas simples como enxadas. É um método eficaz para pequenas áreas, mas inviável para grandes propriedades devido à mão de obra intensiva. O controle mecânico envolve o uso de equipamentos tracionados por animais ou tratores para cortar, arrancar ou enterrar as plantas daninhas. Exemplos incluem a aração, gradagem e cultivo mecanizado, realizado entre linhas da cultura. O preparo adequado do solo antes do plantio também se encaixa nesta categoria, visando o enterrio das plantas daninhas existentes. Esse método ainda exige uma mão de obra mais intensiva devido ao percurso extensivo por todo o campo cultivado, bem como a necessidade de preparo do solo em alguns métodos. Com o avanço da tecnologia, o uso de herbicidas, definidos como substâncias químicas desenvolvidas para eliminar ou inibir o crescimento de plantas daninhas, tornou-se um método tradicional em grande escala na agricultura moderna. A aplicação pode ser feita antes da emergência da cultura (pré-emergência) ou após (pós-emergência), dependendo do tipo de herbicida e da cultura (Constantin, 2011).

O método de controle baseado em defensivos agrícolas se popularizou rapidamente devido a sua eficácia e rentabilidade. Em linhas gerais, o custo de obtenção dos herbicidas é baixo e sua aplicação ao longo do campo de colheita resulta em aparente efetiva eliminação das invasoras. Em contrapartida, os efeitos ao longo prazo da aplicação excessiva de agrotóxicos envolvem a resistência das plantas daninhas e consequente aumento dos custos financeiros ao produtor, além de riscos ambientais e impactos na sustentabilidade e riscos à saúde pública. O uso contínuo do mesmo herbicida ou de herbicidas com o mesmo modo de ação seleciona e favorece o crescimento de plantas daninhas resistentes. Como consequência, o produtor que passa a utilizar novos tipos ou maiores quantidades de herbicidas para compensar a perda de eficiência dos produtos anteriormente empregados acaba arcando com custos mais elevados, o que impacta diretamente a rentabilidade da produção. Herbicidas podem afetar negativamente o solo, reduzindo a quantidade de organismos benéficos, e a persistência de seus resíduos (*carryover*) pode prejudicar culturas subsequentes. Esses produtos também podem contaminar lençóis freáticos a partir do processo de lixiviação, isto é, o movimento descendente da água através das camadas do solo, a percolação, que arrasta consigo as moléculas do herbicida, trazendo grandes preocupações no âmbito ambiental (Agropecuária, 2024).

Com o avanço da tecnologia, a agricultura de precisão obteve destaque em diversos contextos. No controle de plantas daninhas, a aplicação localizada dos insumos é um exemplo que tem tomado força no mercado agrícola mundial. As tecnologias são diversas, e envolvem modernizações tais como pulverizadores inteligentes, equipados com sensores e câmeras que identificam espécies individualmente e aplicam o herbicida automaticamente no local exato; sensores espectrais que utilizam luz infravermelha e vermelha para detectar plantas verdes, diferenciando-as do solo ou de outras plantas não-alvo; e os drones, que permitem a pulverização localizada, especialmente em áreas de difícil acesso. Eles podem ser usados para pulverização de contato em plantas daninhas estabelecidas ou para aplicar agentes de controle biológico. Em geral, essas tecnologias baseiam-se na prévia identificação inteligente das espécies de plantas daninhas, baseada na construção de modelos de inteligência artificial.

Os métodos mais estabelecidos para a construção dos modelos mencionados são fundamentados no aprendizado centralizado. Nesta metodologia, todos os dados são coletados e armazenados em um único servidor, onde ocorre o treinamento do modelo. Embora essa abordagem possa simplificar o gerenciamento das informações, ela também aumenta significativamente o risco de exposição de dados sensíveis, uma vez que concentra grandes volumes de informações em um ponto único de vulnerabilidade.

Por outro lado, a Aprendizagem Federada adota um modelo distribuído, no qual o treinamento é realizado localmente em cada dispositivo ou instituição participante, e apenas os parâmetros resultantes do processo são compartilhados com um servidor central. Dessa forma, os dados originais permanecem protegidos em suas respectivas origens, reduzindo o risco de vazamentos e garantindo maior preservação da privacidade, aspecto essencial no contexto agrícola,

em que informações sobre produtividade, manejo e localização têm alto valor estratégico.

Dadas as problemáticas da matocompetição nos canaviais e o dilema da privacidade de dados no avanço da agricultura digital, o presente trabalho avalia a hipótese de que a adoção de uma arquitetura de AF para a classificação de imagens de plantas daninhas é capaz de assegurar a privacidade dos dados locais sem comprometer de forma significativa a acurácia e a confiabilidade do modelo, quando comparado à abordagem tradicional de aprendizado centralizado.

Nesta via, a fim de guiar as etapas de desenvolvimento e validação desta pesquisa, estabelecem-se três perguntas de pesquisa centrais: (I) No contexto de classificação de imagens de plantas daninhas em lavouras de cana-de-açúcar, qual é o impacto no desempenho preditivo ao se migrar de um ambiente de aprendizado centralizado para um ambiente de Aprendizado Federado? (II) Considerando que dados agrícolas tendem a apresentar forte variabilidade espacial, como cenários de distribuição de dados heterogêneos entre os clientes no ambiente federado afetam a convergência e a estabilidade do modelo treinado? (III) É possível que o modelo federado alcance métricas de desempenho competitivas sendo treinado exclusivamente a partir do compartilhamento de gradientes, preservando assim a confidencialidade das informações das propriedades rurais?

Para responder a tais questões e validar a hipótese proposta, experimentos foram conduzidos utilizando um dataset público de imagens de cana-de-açúcar e plantas daninhas. Inicialmente foi estabelecida um modelo de referência por meio do treinamento e otimização de modelos de aprendizado profundo *(Deep Learning)* em regime centralizado. Posteriormente, o *framework Flower* foi utilizado para simular a execução distribuída, possibilitando o particionamento do *dataset*, distribuição entre diferentes clientes, execução dos treinamentos, agregações e testes em diferentes rodadas. Nesses cenários, os mesmos modelos foram treinados sob uma arquitetura federada, permitindo a comparação direta do desempenho preditivo, da viabilidade de implementação e da estabilidade de convergência entre as duas abordagens de treinamento tecnológico.

Os resultados demonstram que, ao comparar a metodologia federada com a arquitetura de melhor desempenho no método centralizado em 15 épocas, a *DarkNet*, a convergência alcança métricas de acurácia, *recall*, *F1-score* e *loss* satisfatórias, visto que o processo de estabilização ocorre dentro das 15 rodadas de treinamento propostas. Assim sendo, há indicativos de que a aplicação do treinamento federado em campo pode sanar uma desafio do setor, ao mesmo tempo em que a alta performance da solução de classificação de daninhas é mantida.

1.1 Justificativa

Conforme mencionado anteriormente, o cultivo da cana-de-açúcar possui grande relevância econômica no Brasil, o que impulsiona investimentos expressivos em agricultura de precisão, fundamentais para a otimização da cadeia produtiva. Nesse cenário, a segurança das informações

corporativas dos grandes produtores representa um desafio constante. Dessa forma, a utilização e o processamento de dados de maneira distribuída surgem como alternativas viáveis para assegurar a privacidade e a confiança na tomada de decisão das empresas.

Como será detalhado na revisão de literatura, a aplicação da Aprendizagem Federada *(Federated Learning)* no setor agrícola é uma área de pesquisa ainda pouco explorada. No que tange à identificação de plantas daninhas em canaviais, a literatura mostra-se ainda mais incipiente. Diante dessa lacuna, este trabalho propõe a adoção dessa abordagem de treinamento distribuído para a classificação de plantas daninhas, buscando manter um nível de desempenho preditivo comparável ao de modelos treinados em ambientes centralizados.

A escolha e a viabilidade deste tema justificam-se, também, pela disponibilidade de um conjunto de dados público de alta qualidade e representatividade (Domah, 2025). Por fim, a experiência profissional prévia da autora no setor permitiu a identificação desta oportunidade de pesquisa, unindo uma necessidade real do mercado a uma abordagem tecnológica inovadora

1.2 Objetivos

O objetivo geral deste trabalho é propor um modelo de Aprendizado Federado para a identificação de plantas daninhas em lavouras de cana-de-açúcar, alcançando uma acurácia competitiva em relação à abordagem centralizada.

Como objetivos específicos, tem-se:

1. Estabelecer um modelo de alta qualidade, obtido por meio de treinamento centralizado. De acordo com a literatura, os melhores modelos reportados para esse problema alcançam acurácia de 96,6% e *F1-score* de 99%, valores que servirão como referência do estado da arte para a comparação com o *baseline* deste estudo.

2. Verificar a validade de aplicação da Aprendizagem Federada no contexto de identificação de plantas daninhas, comparando distribuições de dados identicamente distribuídos e não identicamente distribuídos.

3. Comparar o desempenho da proposta federada com a proposta centralizada tradicional

1.3 Organização do Trabalho

Para fins de organização, o restante deste documento está estruturado em cinco capítulos. O Capítulo 2 apresenta a Revisão Bibliográfica, que embasa teoricamente a pesquisa. Em seguida, o Capítulo 3 detalha a Metodologia, descrevendo os procedimentos e ferramentas utilizados no estudo. O Capítulo 4 expõe os Resultados e Discussões obtidos até o presente momento. O Capítulo 5 traz as Considerações Finais, sintetizando as conclusões.

1.3.1 Estrutura da Monografia

Segue a estrutura da presente pesquisa:

Capítulo 1: Introdução.

Capítulo 2: Revisão Bibliográfica.

Capítulo 3: Metodologia ou Desenvolvimento.

Capítulo 4: Resultados e Discussões.

Capítulo 5: Considerações Finais

2 Revisão Bibliográfica

Esta seção apresenta a Revisão da Literatura, constituindo a fundamentação teórica necessária para a compreensão dos conceitos, modelos e paradigmas de aprendizado profundo utilizados neste trabalho. O encadeamento dos tópicos foi estruturado de forma a construir uma linha de raciocínio que parte dos fundamentos matemáticos isolados até a sua aplicação em arquiteturas distribuídas complexas. Nesse sentido, a primeira parte (2.1) concentra os conceitos e temas basilares que fundamentam esta pesquisa, enquanto a segunda subseção (2.2) expõe os trabalhos correlacionados que se alinham diretamente com os objetivos propostos.

2.1 Fundamentação Teórica

2.1.1 Aprendizado de Máquina

A área de Aprendizado de Máquina (AM), frequentemente referenciada pelo termo em inglês *Machine Learning* (ML), é dedicada ao estudo e ao desenvolvimento de programas de computador capazes de melhorar o próprio desempenho automaticamente por meio da experiência (Mitchell, 1997). Esse é um campo eminentemente multidisciplinar, fundamentado na integração de conhecimentos de áreas como Estatística, Inteligência Artificial (IA), Teoria da Informação, Teoria de Controle, Complexidade Computacional, Filosofia, Ciência Cognitiva e Biologia. Nesse contexto computacional, o termo "aprendizado"diz respeito ao processo pelo qual um modelo ajusta o seu comportamento interno ao ser exposto a um conjunto de dados, isto é, à sua "experiência", aprimorando iterativamente a sua capacidade de generalização e de resolução de tarefas.

Os métodos de ML são, em geral, divididos em três tipos, com base na forma como os modelos aprendem com os dados: Aprendizado Supervisionado, Não Supervisionado e por Reforço. No Aprendizado Supervisionado, os dados de entrada são pareados com as respectivas saídas corretas. Dessa forma, o algoritmo tenta entender padrões na relação entre entrada e saída, de modo que possa prever o resultado para um novo dado futuro. O modelo é treinado com os pares conhecidos e ajusta o seu funcionamento durante o treinamento para minimizar os erros na previsão, visando acertar conforme novas entradas são apresentadas. Diferentemente, no Aprendizado Não Supervisionado, o modelo recebe dados sem os respectivos rótulos, ou seja, o treinamento não possui as saídas corretas. Dessa maneira, o algoritmo tenta identificar padrões, agrupamentos ou relações de modo independente. Em seu funcionamento, o sistema examina as entradas para encontrar conjuntos com o mesmo padrão e agrupa os pontos de dados com base em suas similaridades. Há ainda uma variação que combina os dois métodos citados. O Aprendizado Semi Supervisionado é uma abordagem híbrida na qual parte da entrada é rotulada e a outra

parte não possui rótulos, com o objetivo de melhorar o aprendizado. Por fim, o Aprendizado por Reforço é caracterizado pela existência de um agente que interage com um ambiente e executa tarefas. Esse agente aprende por meio das recompensas ou penalidades provenientes das próprias ações, otimizando a sua função de recompensa, isto é, busca maximizar os ganhos e minimizar as perdas em um determinado espaço de tempo.

Ademais, para compreender as tarefas executadas em um treinamento, é necessário entender os dois principais problemas envolvidos em AM: Regressão e Classificação. A Regressão consiste na tarefa de prever valores numéricos contínuos, tais como preços de ações e temperaturas. Para essa modelagem, algoritmos como Regressão Linear, *Random Forest* e Regressão *Ridge* são utilizados, podendo ser avaliados por métricas como o Erro Médio Absoluto (MAE) e o Erro Quadrático Médio (MSE), entre outras. Já os problemas de Classificação preveem valores categóricos discretos, como identificar se um email é spam ou se um determinado animal é um gato ou um cachorro. Alguns classificadores comuns são as Árvores de Decisão e os algoritmos baseados em Floresta Aleatória. As avaliações desse tipo de modelagem são feitas com base em métricas como Acurácia, Precisão, *Recall* e *F1 Score*.

No presente trabalho, os dados provenientes do *dataset* público (Domah, 2025) seguem a estrutura de uma entrada com os respectivos rótulos. Por esse motivo, o tipo de treinamento utilizado é o Supervisionado. Além disso, a partir de uma imagem, o modelo irá inferir se a entrada pertence à classe de Plantas Daninhas ou não. Essa decisão binária caracteriza o estudo atual como um problema de Classificação.

2.1.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são sistemas computacionais inspirados na arquitetura biológica do sistema nervoso central, projetados para processar informações de forma distribuída e paralela (Wu; Feng, 2018). Embora os primeiros modelos datem da década de 1940, foi na década de 1980 que estas estruturas se consolidaram como ferramentas robustas de modelagem.

A unidade fundamental desses modelos é o neurônio artificial, ou perceptron, que atua como um dispositivo lógico funcional. Em uma estrutura de Multilayer Perceptron (MLP), esses neurônios são organizados em camadas: uma camada de entrada, que recebe os sinais externos; uma ou mais camadas ocultas, responsáveis pela extração de representações internas; e uma camada de saída, que fornece o resultado final do processamento (Wu; Feng, 2018).

O fluxo de dados através dessa arquitetura (Figura 2.1) é denominado *Forward Propagation*. Nesse processo, a entrada total x_j de um neurônio j é calculada como uma função linear das saídas y_i das unidades da camada anterior, ponderadas pelos respectivos pesos sinápticos w_{ji}, acrescida de um termo de viés (*bias*) (Rumelhart; Hinton; Williams, 1986). Matematicamente, a entrada é expressa por:

$$x_j = \sum_i y_i w_{ji} + bias_j \tag{2.1}$$

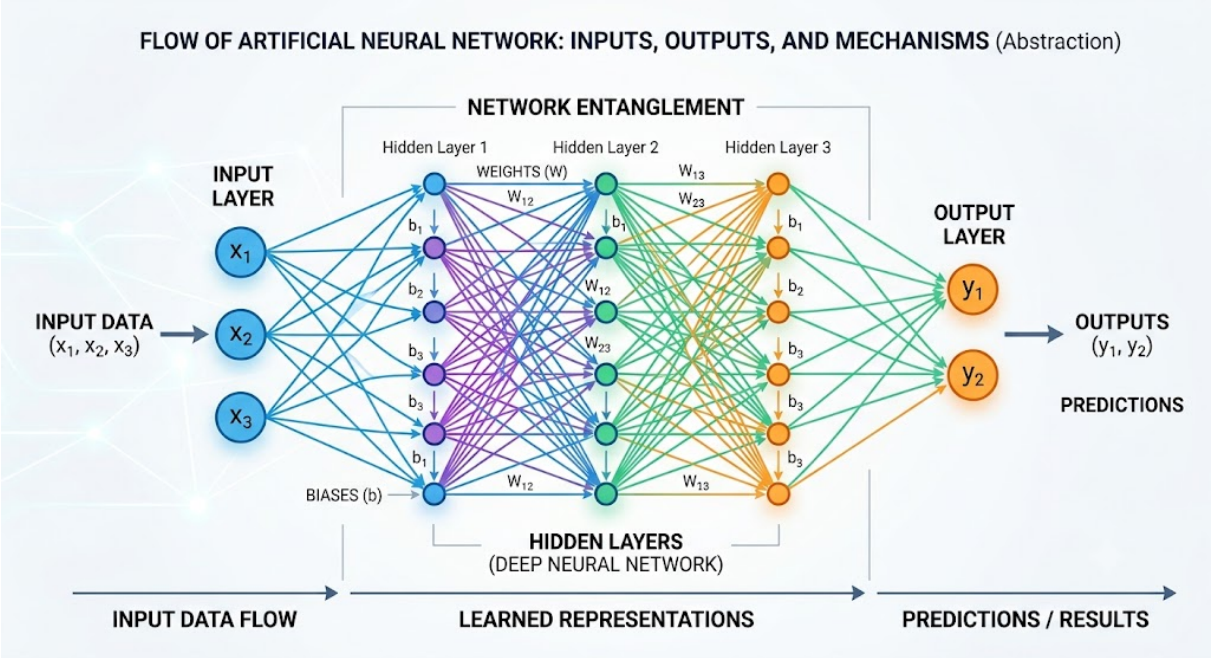

Figura 2.1 – Fluxo de processamento e arquitetura abstrata de uma Rede Neural Artificial profunda. Fonte: Gerada pela autora utilizando a plataforma Google Gemini Versão 3.1 Pro (2026)

Os neurônios são as unidades básicas de processamento das redes neurais, operando em três etapas sequenciais: a recepção, o processamento e a transmissão das informações. Em linhas gerais, o neurônio capta dados numéricos que são repassados pela camada imediatamente anterior e, em seguida, tais valores são multiplicados por seus respectivos pesos e somados linearmente e acrescentando a constante de viés. Em seguida, o resultado é submetido a uma função de ativação não-linear. Por fim, o dado gerado é transmitido para a camada posterior, propagando o processamento pela rede de forma contínua (Almada, 2019). A Figura 2.2 esquematiza o fluxo de dados em um neurônio.

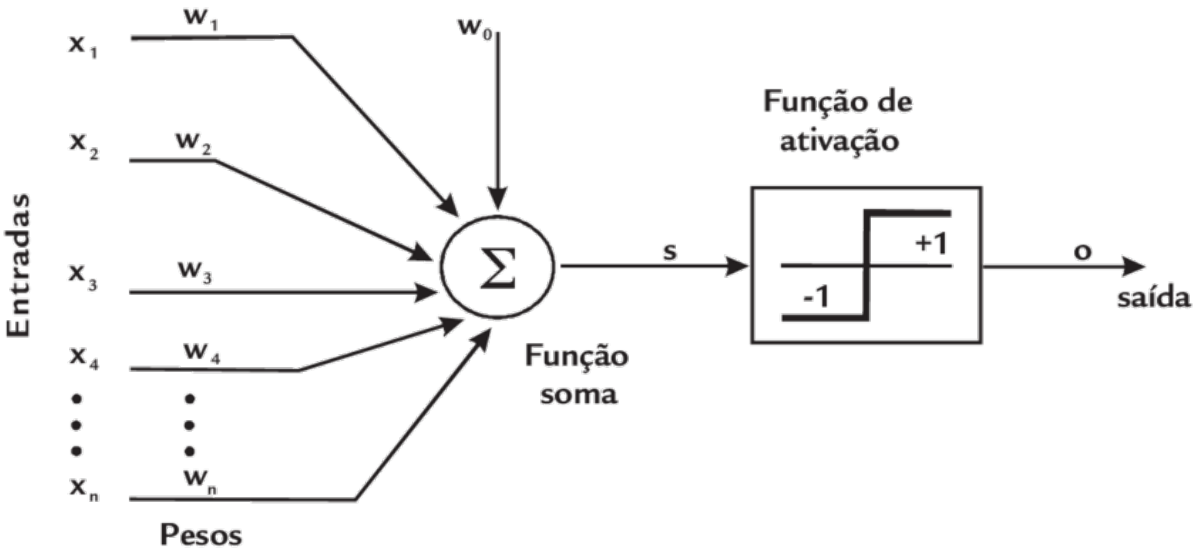

Figura 2.2 – Diagrama do fluxo de processamento de um neurônio artificial, ilustrando como os dados de entrada são transformados em uma previsão final através de pesos, viés e cálculos matemáticos. Fonte: (Rosa; Luz, 2012)

2.1.2.1 Funções de Ativação

Embora as RNAs operem essencialmente através de transformações lineares dadas pelas multiplicações matriciais de pesos e adições de vieses, a composição de múltiplas camadas puramente lineares é matematicamente equivalente a uma única camada linear (LeCun; Bengio; Hinton, 2015). Assim sendo, sem a introdução de uma Função de Ativação não-linear, o modelo seria incapaz de aprender superfícies de decisão complexas, limitando-se a separar classes que podem ser divididas por um hiperplano no espaço de entrada (LeCun; Bengio; Hinton, 2015).

Este limite foi historicamente evidenciado pelo problema da lógica XOR (ou-exclusivo). Classificadores lineares simples, como os perceptrons originais, falham em resolver o XOR porque as classes não são linearmente separáveis (Wu; Feng, 2018). A função de ativação atua distorcendo o espaço de entrada em cada camada oculta, de modo que, ao chegar na camada final, os dados se tornem linearmente separáveis.

Entre as funções mais comuns, destacam-se a Sigmoid ($f(z) = 1/(1 + e^{-z})$), tradicionalmente utilizada por sua suavidade; a Softmax, aplicada em camadas de saída para fornecer distribuições de probabilidade entre classes; e a ReLU (*Rectified Linear Unit*, $f(z) = max(0, z)$), que se tornou o padrão em redes profundas por acelerar significativamente a convergência do treinamento.

Os autores do artigo supracitado destacam que as RNAs são essenciais para o estado da arte da IA, tendo em vista seu processamento distribuído em paralelo. Diferentemente da computação tradicional sequencial, as RNAs processam as informações simultaneamente em múltiplos nós, o que confere alta velocidade e eficiência. Além disso, devido a essa natureza distribuída, o sistema mantém sua funcionalidade mesmo se partes da rede sofrerem danos ou

se os dados de entrada apresentarem ruídos. Por fim, tal relevância também é fundamentada na capacidade de aprendizado e adaptabilidade, uma vez que as redes ajustam os pesos de suas conexões internas com base em dados de entrada, permitindo que o modelo aprenda padrões sem ser explicitamente programado para cada regra.

2.1.2.2 Otimização e Aprendizado

O objetivo do treinamento é encontrar um conjunto de pesos que minimize a discrepância entre a predição da rede e o rótulo real, quantificada por uma Função de Custo (ou Perda), como a Entropia Cruzada para tarefas de classificação (Rumelhart; Hinton; Williams, 1986). A minimização dessa função ocorre iterativamente através do algoritmo de Descida de Gradiente (*Gradient Descent*, Figura 2.3), onde os pesos são ajustados na direção oposta ao vetor gradiente por meio da seguinte equação:

$$\Delta w = -\eta \frac{\partial E}{\partial w} \tag{2.2}$$

onde η representa a taxa de aprendizado. Tal hiperparâmetro deve ser balanceado para que o tamanho do passo dado na otimização da função não seja tão grande a ponto de causar instabilidade, tampouco tão pequeno a ponto de inviabilizar a convergência dentro das épocas preestabelecidas no treinamento (LeCun; Bengio; Hinton, 2015).

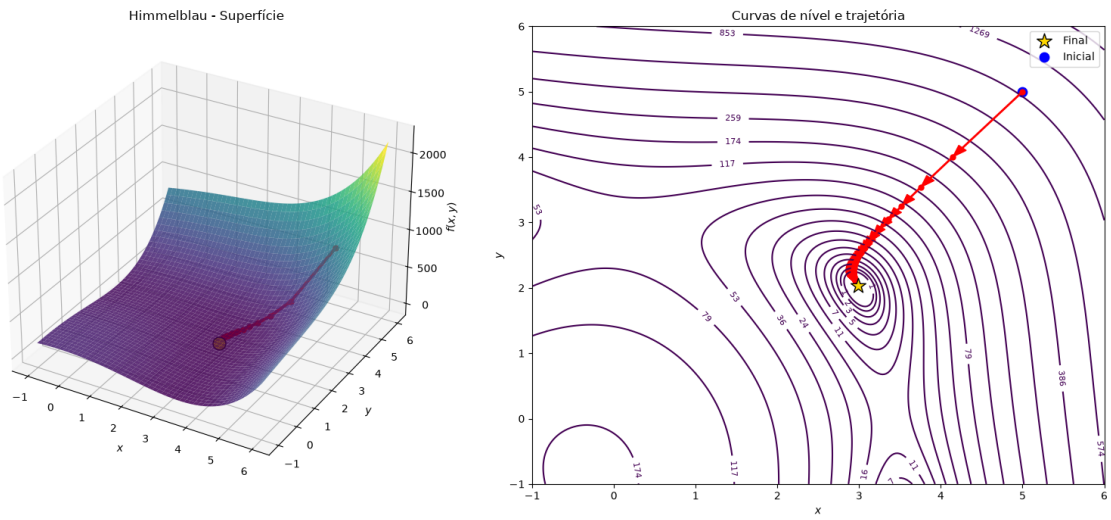

Figura 2.3 – Trajetória da descida do gradiente na função de Himmelblau. Fonte: elaboração própria. Nota: O método do gradiente descendente é um algoritmo para encontrar o mínimo de uma função em N dimensões. As setas representam as atualizações sucessivas do algoritmo. (Sipio, 2019)

Para calcular as derivadas parciais necessárias, utiliza-se o algoritmo de *Backpropagation* (retropropagação do erro) (Rumelhart; Hinton; Williams, 1986). Baseado na regra da cadeia, o *Backpropagation* propaga o erro da camada de saída de volta para as camadas iniciais, permitindo

identificar a contribuição de cada peso para o erro total (LeCun; Bengio; Hinton, 2015; Wu; Feng, 2018). O processo permite que as unidades ocultas aprendam representações internas que capturam as regularidades dos dados.

A evolução dos algoritmos de treinamento, como o *backpropagation*, isto é, regra de aprendizagem baseada na correção do erro pelo método do gradiente, aliada ao aumento do poder computacional, permitiu a transição das RNAs de modelos teóricos para aplicações práticas em reconhecimento de imagem, previsão e processamento de sinais. No contexto de modelos avançados como a *EfficientNet*, as RNAs evoluíram para estruturas convolucionais profundas, mas ainda retêm esses princípios fundamentais de interconexão e ajuste de pesos descritos por (Wu; Feng, 2018).

2.1.2.3 Redes Neurais Convolucionais

O trabalho de (LeCun et al., 2002) é amplamente reconhecido como o marco fundamental no uso de redes neurais para o reconhecimento de padrões visuais. Em sua obra seminal, os autores argumentaram que as redes neurais totalmente conectadas (MLPs) eram inadequadas para o processamento de imagens, pois ignoravam a topologia bidimensional dos dados e sofriam com a explosão no número de parâmetros. Para mitigar essas limitações, os autores formalizaram a arquitetura das Redes Neurais Convolucionais (CNNs), exemplificada pelo modelo *LeNet-5*. A grande inovação consistiu na introdução de três mecanismos arquiteturais projetados para garantir invariância a translações e distorções, explorando a correlação espacial local das imagens. Assim sendo, foram propostas conexões restritas a uma vizinhança local de pixels, permitindo a extração de características elementares como bordas e cantos, pesos compartilhados, isto é, a aplicação do mesmo conjunto de filtros em toda a imagem, o que não apenas garante a detecção de características independentemente de sua posição, mas também reduz drasticamente a complexidade do modelo,e, por fim, a subamostragem *(Pooling)*, que consiste na redução progressiva da resolução espacial dos mapas de características para diminuir a sensibilidade a deslocamentos e distorções geométricas. A eficácia dessa abordagem na identificação de dígitos manuscritos por meio de um treinamento supervisionado via *backpropagation* estabeleceu o paradigma do aprendizado de características ponta a ponta, eliminando a dependência de extratores manuais e dando origem a arquiteturas modernas de classificação de imagens.

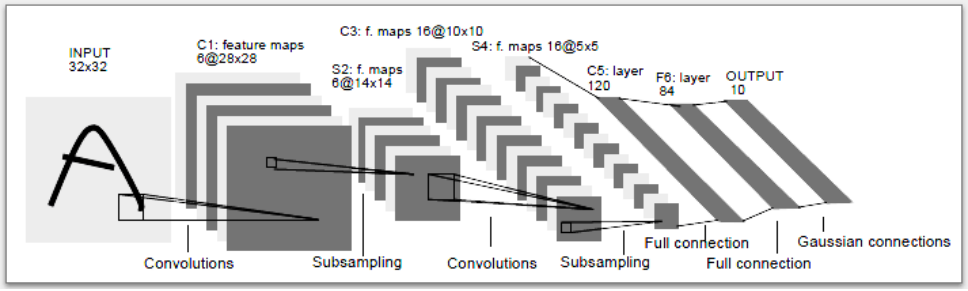

Figura 2.4 – Arquitetura de uma CNN segundo LeCun et al (LeNet5). Fonte: (LeCun et al., 2002).

Conforme demonstrado na Figura 2.4, as camadas de processamento contêm três camadas de convolução (C1, C3 e C5), intercaladas com duas camadas de subamostragem *(sub-sampling)* (S2 e S4), e a camada de saída (F6). Essas camadas de convolução e subamostragem são organizadas em planos chamados mapas de características *(feature maps)*. Na camada de convolução, cada neurônio está localmente vinculado a um campo receptivo local da camada anterior. Todos os neurônios com mapas de características semelhantes obtêm dados de diferentes regiões de entrada até que toda a entrada do plano seja percorrida, mas os mesmos pesos são utilizados em conjunto. Os mapas de características são subamostrados espacialmente na camada de subamostragem, na qual o tamanho do mapa é reduzido por um fator 2. Por exemplo, o mapa de características na camada C3 de tamanho 10×10 é subamostrado para um mapa correspondente de tamanho 5×5 na camada subsequente S4. A última camada é a F6, que realiza o processo de classificação.

2.1.3 ResNet

Antes das chamadas *ResNets*, já se sabia que CNNs profundas eram cruciais para o reconhecimento visual. Redes como *AlexNet* e VGG haviam demonstrado que aumentar a profundidade enriquecia os níveis de características extraídas de bordas simples a formas complexas, uma vez que adicionar mais camadas deveria, teoricamente, aumentar a capacidade da rede de modelar problemas complexos. Em contrapartida, ao progredir desta maneira, era observada uma piora no desempenho, em que otimizadores tinham dificuldade em encontrar a solução ideal em redes muito profundas, de modo que com o aumento da profundidade, a acurácia de treinamento saturava e, inesperadamente, degradava-se de forma acelerada (He et al., 2016a).

É fundamental distinguir esse fenômeno do problema de gradientes evanescentes ou explosivos (*vanishing/exploding gradients*), que impedem a convergência inicial. O problema da degradação manifesta-se mesmo quando a convergência é viabilizada por meio de inicialização normalizada e camadas de normalização intermediária (*Batch Normalization*), indicando que nem todos os sistemas são igualmente fáceis de otimizar.

Os gráficos da Figura 2.5 ilustram o fenômeno da degradação em arquiteturas de redes neurais profundas convencionais, comparando os erros de treinamento, à esquerda, e de teste, à direita, de modelos com 20 e 56 camadas ao longo das iterações. Contra o comportamento esperado de que maior profundidade resultaria em melhor capacidade de generalização, a rede mais profunda de 56 camadas, representada pela linha vermelha, apresenta taxas de erro de treino e de teste consistentemente superiores às da rede mais rasa de 20 camadas, representada pela linha amarela. Como esse aumento de erro ocorre já na fase de treinamento, o comportamento não caracteriza um caso de *overfitting*, mas sim uma severa dificuldade de otimização matemática decorrente do aumento da profundidade, justificando a necessidade histórica do desenvolvimento do aprendizado residual (ResNet) para viabilizar redes ainda maiores.

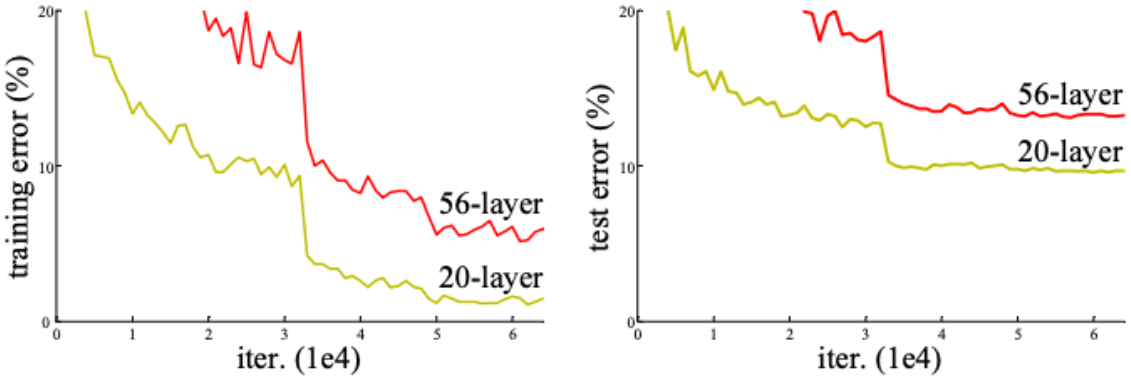

Figura 2.5 – Fenômeno da Degradação vs. Aprendizado Residual. Fonte: (He et al., 2016a).

Para resolver a dificuldade de otimização, os autores introduziram uma mudança arquitetural fundamental, que consistiu em reformular o aprendizado das camadas para que estas aproximassem uma função de resíduo $\mathcal{F}(x)$, em vez da função de mapeamento completa. Assim sendo , em vez de esperar que as camadas empilhadas ajustem-se diretamente a um mapeamento desejado $H(x)$, a rede é explicitamente reformulada para aprender uma função residual $F(x) := H(x) - x$. O mapeamento original é, portanto, redefinido como:

$$y = F(x, \{W_i\}) + x$$

.

A premissa era que seria mais fácil ajustar a diferença em relação à entrada original do que aprender uma nova função do zero. Essa estratégia foi implementada através de "conexões de atalho"(*shortcut connections*), que realizam um mapeamento de identidade somando elemento a elemento entre a entrada do bloco e a saída das camadas convolucionais. A principal vantagem teórica desta abordagem é que, caso um mapeamento de identidade seja a solução ótima, torna-se computacionalmente mais simples para o sistema conduzir os pesos das camadas não lineares a zero do que tentar aproximar uma função de identidade a partir de uma estrutura puramente não linear (He et al., 2016a).

A inclusão de camadas de *Batch Normalization* (BN) constitui a solução primária para mitigar os problemas de gradientes evanescentes ou explosivos, viabilizando a convergência de redes profundas via descida de gradiente estocástica. Ao reduzir o desvio interno de covariáveis, a BN acelera significativamente o treinamento (He et al., 2016a). No fluxo do sinal residual, essa técnica desempenha um papel crucial para a estabilidade do modelo, sendo aplicada imediatamente após cada convolução e antes da função de ativação *Rectified Linear Unit* (ReLU), que zera valores negativos e mantém os positivos para acelerar o aprendizado do modelo. Embora a ReLU seja tradicionalmente adotada após a adição do atalho (shortcut), variantes de "pré-ativação", nas quais a BN e a ReLU precedem a camada convolucional, demonstram facilitar ainda mais a propagação do sinal em redes extremamente profundas (He et al., 2016b).

2.1.4 EfficientNet

Anteriormente à proposição da arquitetura *EfficientNet*, o aumento da acurácia em Redes Neurais Convolucionais (CNNs) baseava-se predominantemente na expansão das dimensões do modelo, seja pela adição de camadas (profundidade) ou de canais (largura). Um exemplo clássico dessa abordagem é a evolução da família *ResNet*, escalada desde a *ResNet-18* até versões mais profundas como a *ResNet-152*. Entretanto, esse método de escalonamento unidimensional frequentemente resulta em um crescimento exponencial do custo computacional e do consumo de memória, apresentando retornos de acurácia cada vez menores.

A arquitetura *EfficientNet*, apresentada por (Tan; Le, 2019), estabeleceu um novo paradigma ao propor o "Escalonamento Composto". Os autores demonstraram que o escalonamento isolado de largura, profundidade ou resolução tende a saturar a performance do modelo. Para mitigar isso, propuseram um coeficiente composto (ϕ) que escala uniformemente todas as dimensões da rede. Intuitivamente, essa abordagem assegura que, ao processar imagens maiores, a rede possua profundidade suficiente para expandir seu campo receptivo e largura adequada para identificar padrões detalhados. Ao aplicar esse método sobre uma rede base denominada *EfficientNet-B0* desenvolvida via *Neural Architecture Search* (NAS), isto é, uma técnica de Inteligência Artificial que automatiza o design de redes neurais profundas, obteve-se uma família de modelos (B0-B7) que atingiu eficiência superior, reduzindo a quantidade de parâmetros em até 8,4 vezes em comparação a arquiteturas como o GPipe, mantendo a precisão no estado da arte.

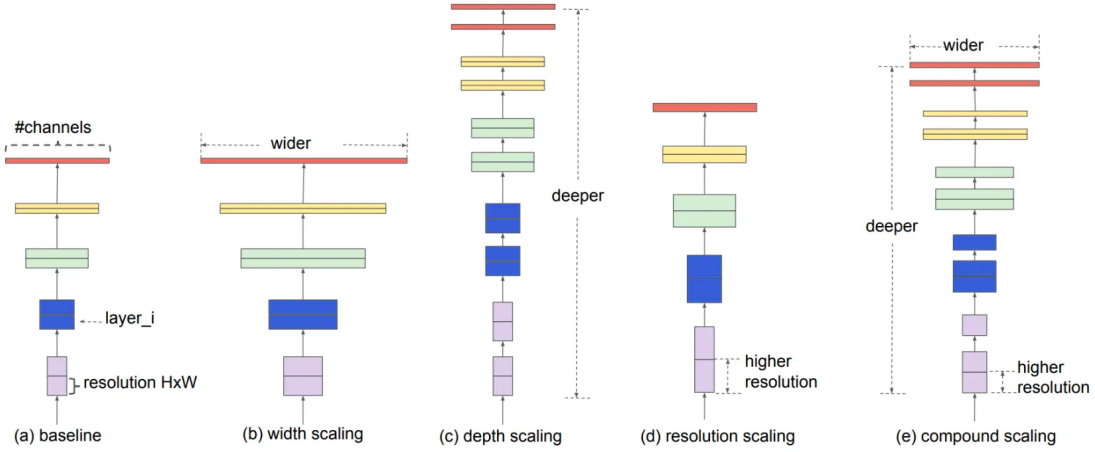

Figura 2.6 – Escalonamento de modelo. Fonte: (Tan; Le, 2019).

A Figura 2.6 esquematiza a proposta de escalonamento composto explicitado anteriormente. Logo, tem-se que (a) é um exemplo de rede de referência (*baseline*); (b)-(d) são escalonamentos convencionais que aumentam apenas uma dimensão da rede: largura, profundidade ou resolução. (e) é o nosso método de escalonamento composto proposto em (Tan; Le, 2019), que dimensiona uniformemente todas as três dimensões com uma proporção fixa.

2.1.5 DarkNet53

A arquitetura *DarkNet53* como o núcleo extrator de caracterísitcas, também chamado de *backbone*, do algoritmo de detecção de objetos YOLOv3. A sua criação foi motivada pela necessidade de encontrar um equilíbrio ideal entre a precisão na classificação de imagens e a eficiência no tempo de processamento, visando aplicações em tempo real, de modo que foi uma evolução híbrida da arquitetura anterior, *DarkNet19*, e que integrou conceitos fundamentais das redes residuais, a *ResNet* (Redmon, 2018).

Enquanto a *DarkNet19* baseia-se em camadas convolucionais sucessivas de 3×3 e 1×1, a *DarkNet53* expande essa profundidade para um total de 53 camadas convolucionais. A principal inovação técnica incorporada foi a utilização de conexões residuais, as quais permitem que o gradiente flua mais facilmente através das camadas durante o processo de retropropagação, mitigando o problema do gradiente evanescente, um fenômeno comum em redes neurais muito profundas que impede a convergência e a atualização eficiente dos pesos nas camadas iniciais.

Sua arquitetura é composta fundamentalmente por blocos residuais repetidos que utilizam camadas convolucionais consecutivas (Figura 2.7) . A lógica operacional da rede utiliza camadas convolucionais com deslocamento 2 para realizar a redução da dimensionalidade espacial de forma aprendida, o que preserva mais informações estruturais dos mapas de características ao longo da profundidade da rede (Redmon, 2018).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figura 2.7 – Distribuição de Camadas e Parâmetros da DarkNet53. Fonte: (Redmon, 2018)

A aplicação da DarkNet53 no contexto do mapeamento de plantas daninhas na cultura da

cana-de-açúcar foi dada por (Modi et al., 2023) e justifica-se por sua robustez superior frente a outros modelos de aprendizagem profunda. Em testes comparativos, a DarkNet53 superou arquiteturas consagradas como AlexNet, GoogLeNet e ResNet50, alcançando um F1-score superior a 99% e uma precisão de 99,3% na distinção entre a cultura principal e as espécies invasoras (Modi et al., 2023). No contexto agrícola, essa alta performance é crucial devido aos desafios inerente ao ambiente, onde plantas daninhas e folhas de cana apresentam cores e texturas similares, o que dificulta a extração de características distintivas por métodos tradicionais. Ademais, a capacidade da *DarkNet53* de processar informações em múltiplas escalas e sua alta velocidade de inferência, sendo aproximadamente duas vezes mais rápida que modelos ResNet equivalentes com precisão similar, tornam-na a escolha ideal para sistemas embarcados em pulverizadores de precisão e robôs agrícola (Modi et al., 2023).

2.1.6 Comparação das Arquiteturas

Embora as três arquiteturas, *ResNet*, *EfficientNet* e *DarkNet53*, representem marcos fundamentais na evolução das CNNs e no aprimoramento da extração de características visuais, cada uma aborda os desafios de otimização, escalonamento e aplicação prática sob perspectivas metodológicas distintas.

A *ResNet* foi a pioneira na viabilização de redes extremamente profundas. Seu desenvolvimento focou em solucionar o fenômeno da degradação do modelo, um problema de otimização onde o aumento indiscriminado de camadas aumentava a taxa de erro. A introdução das conexões de atalho, que buscam aproximar a função residual, permitiu que o gradiente fluísse livremente, eliminando as barreiras matemáticas para o treinamento de modelos com centenas de camadas, consolidando a profundidade como um vetor de precisão (He et al., 2016a).

Em contrapartida, a *EfficientNet* demonstrou que focar exclusivamente na expansão unidimensional, como a profundidade na *ResNet*, gera custos computacionais proibitivos e retornos decrescentes de acurácia. A arquitetura rompeu com o escalonamento tradicional ao propor o "Escalonamento Composto". Ao expandir uniformemente a profundidade, a largura e a resolução da imagem através de um coeficiente composto fixo (ϕ), e partindo de uma rede base otimizada via NAS, a *EfficientNet* entregou resultados no estado da arte com um número drasticamente menor de parâmetros, consagrando-se como a referência em eficiência arquitetural (Tan; Le, 2019).

Por fim, a *DarkNet53* apresenta uma abordagem pragmática e híbrida, desenvolvida sob a necessidade de suportar sistemas de detecção em tempo real, como o algoritmo YOLOv3. Ela assimila as lições da *ResNet*, incorporando conexões residuais para mitigar gradientes evanescentes em suas 53 camadas, mas distancia-se ao priorizar a velocidade de inferência sem comprometer a extração de detalhes espaciais, alcançada pelo uso de camadas convolucionais com deslocamento ao invés de camadas de *pooling* convencionais (Redmon, 2018). Esse balanço ideal entre robustez e agilidade torna a *DarkNet53* a arquitetura preferencial para sistemas embarcados em ambientes

dinâmicos e complexos, como verificado no mapeamento de plantas daninhas na cultura da cana-de-açúcar (Modi et al., 2023).

Para sistematizar as diferenças fundamentais entre os três modelos, a Tabela 2.1 resume suas principais características analíticas.

Critério Analítico	ResNet	EfficientNet	DarkNet53
Problema Original Mitigado	Degradação da acurácia e dificuldade de otimização em redes muito profundas.	Custos computacionais altos e saturação de performance ao escalar redes unidimensionalmente.	Gargalo entre velocidade de processamento e precisão em aplicações de tempo real.
Inovação Estrutural Primária	Aprendizado residual via conexões de atalho.	Escalonamento composto uniforme em largura, profundidade e resolução.	Blocos residuais repetidos integrados com convoluções de redução espacial aprendida.
Foco de Desempenho	Expansão segura da profundidade e capacidade do modelo.	Maximização da precisão com eficiência de parâmetros.	Equilíbrio ótimo entre velocidade de inferência (FPS) e acurácia.
Principal Vantagem ou Aplicação	Base fundamental para extração de características complexas em múltiplas tarefas visuais.	Classificação de imagens de alto desempenho em ambientes restritos por memória.	*Backbone* de detecção de objetos para processamento rápido e embarcado (ex: robótica agrícola).

Tabela 2.1 – Comparativo arquitetural entre ResNet, EfficientNet e DarkNet53.

2.1.7 Aprendizado Federado

O trabalho de (McMahan et al., 2017) é a obra seminal que formalizou o conceito de Aprendizado Federado (*Federated Learning* - FL), introduzindo o algoritmo *Federated Averaging (FedAvg)* e consolidando-se como um paradigma proeminente de aprendizado de máquina distribuído, fundamentado no princípio de descentralização e preservação de privacidade. Ao contrário das arquiteturas centralizadas convencionais, o FL adota a premissa de que o treinamento de modelo seja realizado no mesmo dispositivo em que os dados são coletados, mitigando riscos de segurança e governança de dados ao garantir que as informações brutas permaneçam localmente nos dispositivos de origem.

A mecânica do FL opera por meio de rodadas de comunicação iterativas. Em cada rodada t, o servidor central distribui o modelo global atual w_t para um subconjunto de clientes selecionados. Cada cliente k realiza um treinamento local utilizando seu conjunto de dados privado P_k para minimizar uma função de perda local $F_k(w)$. Ao final da computação local, os clientes transmitem apenas as atualizações de pesos (Δw_k) ou gradientes para o servidor. O servidor então agrega essas atualizações para produzir o modelo global w_{t+1}, descartando as contribuições individuais logo após a aplicação, o que reforça a minimização de dados (McMahan et al., 2017). A dinâmica

de treinamento e agregações é esquematizada na Figura 2.8.

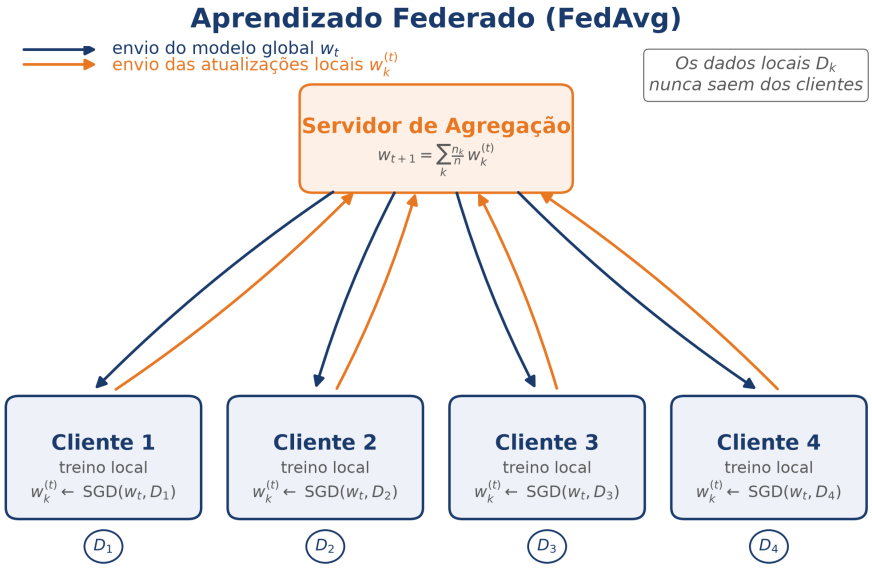

Figura 2.8 – Arquitetura e fluxo operacional do *pipeline* de Aprendizado Federado. Fonte: elaboração própria. Nota: O fluxo é dado pelas seguintes etapas. (1) download do modelo global pelos nós; (2) treinamento local com dados privados; (3) envio dos gradientes atualizados ao servidor; e (4) atualização e agregação global do modelo.

2.1.7.1 Funções de Agregação

De acordo com (Qi et al., 2024), funções de agregação são funções matemáticas responsáveis por juntar modelos locais produzidos após uma rodada de treinamento federado e gerando, assim, um novo modelo global. Assim sendo, a função *Federated Average* (FedAvg) é o método agregador mais comum em FL, e consiste na média ponderada de pesos dos modelos locais, baseada no tamanho do dataset de cada respectivo cliente. O objetivo global do algoritmo é minimizar a média ponderada das funções de perda locais:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{2.3}$$

Onde n_k é o volume de dados no cliente k e n é o total de dados na rede. No FedAvg, cada cliente executa E épocas de Gradiente Descendente Estocástico (SGD) localmente. O servidor realiza a agregação dos parâmetros recebidos através de uma média ponderada:

$$w_{t+1} = \sum_{k \in S_t} \frac{n_k}{m_t} w_k^{t+1} \tag{2.4}$$

Onde m_t é o total de exemplos nos clientes selecionados na rodada t.

Embora eficiente, o FedAvg apresenta limitações severas em redes heterogêneas, tanto sistêmicas quanto estatísticas. Em termos sistêmicos, o FedAvg tende a descartar dispositivos

lentos que não completam as E épocas no tempo estipulado, o que pode introduzir viés no modelo e prejudicar a convergência (Li et al., 2020).

Para mitigar as deficiências do FedAvg, (Li et al., 2020) introduziram o *framework* FedProx. Tal método generaliza o FedAvg ao permitir que cada cliente realize uma quantidade variável de trabalho local, integrando soluções parciais de dispositivos lentos em vez de descartá-los. A inovação central é a adição de um termo de proximidade proximal à função de perda local:

$$\min_{w} h_k(w; w_t) = F_k(w) + \frac{\mu}{2}\|w - w_t\|^2$$

Este termo proximal, controlado pelo hiperparâmetro μ, restringe o treinamento local para que os pesos resultantes não divirjam excessivamente do modelo global inicial w_t. O FedProx estabiliza a convergência em cenários de alta heterogeneidade, impedindo que atualizações locais muito profundas afastem o modelo global do ótimo comum (Li et al., 2020). Os resultados do autor demonstram que no cenário de FedAvg $(\mu = 0)$, a curva apresenta oscilações drásticas ou divergência em ambientes heterogêneos, ao passo que na curva de FedProx $(\mu > 0)$, a trajetória é visivelmente mais estável e suave, demonstrando a robustez conferida pelo termo proximal na limitação da deriva do cliente.

Além das duas supracitadas, outras estratégias de agregação podem ser introduzidas nos protocolos federados. Existem no mercado diferentes *frameworks* para viabilizar a implementação de FL, que facilitam a implementação dos protocolos, bem como as funções de agregações do servidor central, já integradas nas aplicações. Um exemplo de uma dessas ferramentas é o 2.1.7.3 (Beutel et al., 2020), que conta com 23 estratégias de agregação em sua Interface de Programação de Aplicações (API).

2.1.7.2 Distribuição de Dados: IID vs. *Non-IID*

O cenário ideal para a otimização federada é a distribuição IID *(Independent and Identically Distributed)*, onde a partição de dados de cada cliente é uma amostra estatisticamente representativa da população total (McMahan et al., 2017).

Contudo, na prática federada, os dados são inerentemente *Non-IID*. Isso ocorre porque o uso de dispositivos móveis ou sensores é pessoal e contextual, gerando distribuições locais que não representam a distribuição global (Li et al., 2020). Em suma, em ambientes *Non-IID*, o FedAvg sofre do fenômeno de divergência do cliente *(client drift)*, onde as atualizações locais desviam o modelo para ótimos locais divergentes, prejudicando o gradiente global (Figura 2.9). O FedProx responde melhor a esses cenários porque o termo proximal atua como uma âncora, permitindo que o modelo global aprenda características generalistas sem ser dominado por características locais extremas de um único nó (Li et al., 2020).

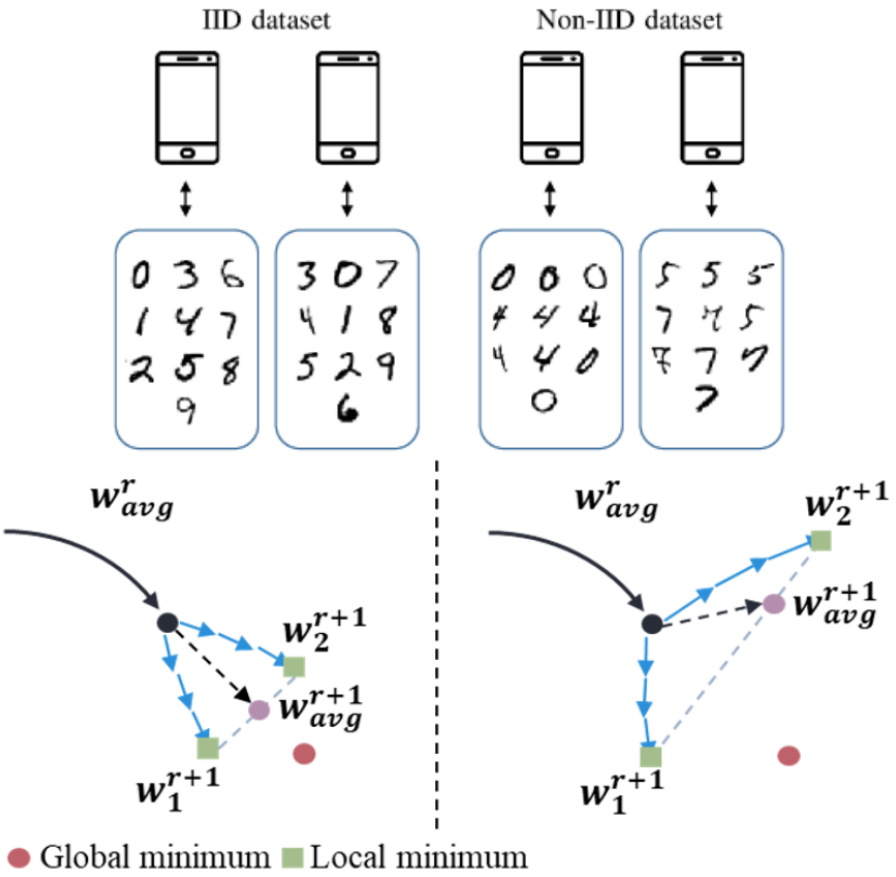

Figura 2.9 – Impacto da Distribuição Não-IID na convergência de um modelo ao utilizar a estratégia FedAVG. Fonte: (Marcondes, 2024)

No âmbito do aprendizado federado, a distribuição estatística dos dados entre as entidades locais exerce um papel determinante sobre a eficiência e a convergência do modelo consolidado. Ainda que o cenário de dados Independentes e Identicamente Distribuídos represente a condição ideal de otimização, as implementações práticas enfrentam inerentemente partições não-identicamente distribuídas, decorrentes de heterogeneidades geográficas e temporais que induzem a vieses nos modelos locais e degradam a generalização global (Arafeh et al., 2022). De acordo com a taxonomia descrita por (Iyer, 2024), essa desconexão estatística manifesta-se sob três vertentes principais: o viés de rótulo (*label skew*), caracterizado por assimetrias nas distribuições marginais de classes; o viés de atributos (*feature skew*), que expressa variações nas probabilidades condicionais entre os nós; e o viés de quantidade (*quantity skew*), definido pela disparidade no volume total de amostras por cliente. Por conseguinte, a mitigação desses desbalanceamentos exige modificações nos algoritmos de agregação e técnicas de expansão de dados para estabilizar a função de otimização distribuída.

2.1.7.3 *Framework Flower*

Para a materialização dos conceitos de Aprendizado Federado em ambientes de produção e pesquisa, destaca-se o *framework Flower* (`flwr`), criado para suprir a necessidade de reduzir

a disparidade entre as simulações acadêmicas e as implementações em sistemas reais de larga escala. Trata-se de uma biblioteca de código aberto, licenciada sob Apache 2.0, concebida para ser escalável e, fundamentalmente, agnóstica a plataformas de Machine Learning (Beutel et al., 2020). Essa característica permite que desenvolvedores utilizem ecossistemas distintos, como PyTorch, TensorFlow, JAX ou Scikit-learn, integrando-os de forma transparente em uma mesma infraestrutura federada.

Enquanto a maioria das pesquisas em FL limita-se a simulações com menos de 100 clientes, o Flower foi projetado para suportar cargas de trabalho heterogêneas que abrangem desde dispositivos móveis e sistemas embarcados, como Raspberry Pi e NVIDIA Jetson, até servidores de borda, permitindo experimentos com até 15 milhões de clientes (Beutel et al., 2020).

A arquitetura do Flower fundamenta-se em um modelo Cliente-Servidor altamente modularizado. No lado do servidor, o gerenciamento é realizado pelo FL Loop, que orquestra o ciclo de aprendizagem, e pelo ClientManager, responsável por gerenciar os objetos ClientProxy, que representam os nós conectados. O servidor é agnóstico à natureza do cliente, o que permite a coexistência de dispositivos com capacidades computacionais e sistemas operacionais variados no mesmo processo de treinamento. A comunicação entre os nós e o servidor é estabelecida por meio de fluxos bidirecionais utilizando RPC *(Remote Procedure Call)*. A escolha desse protocolo justifica-se pela eficiência na serialização binária, fator crítico em conexões de baixa largura de banda típicas de ambientes rurais ou móveis (Beutel et al., 2020). Tecnicamente, os parâmetros do modelo são transmitidos como vetores de *bytes* brutos, os quais são frequentemente convertidos em arranjos NumPy no lado do cliente para processamento local, garantindo uma interface simples para a atualização dos pesos e gradientes. Essa abstração, esquematizada na Figura 2.10, permite que o *framework* gerencie complexidades de rede, como *timeouts* e quedas de conexão, de forma transparente para o pesquisador.

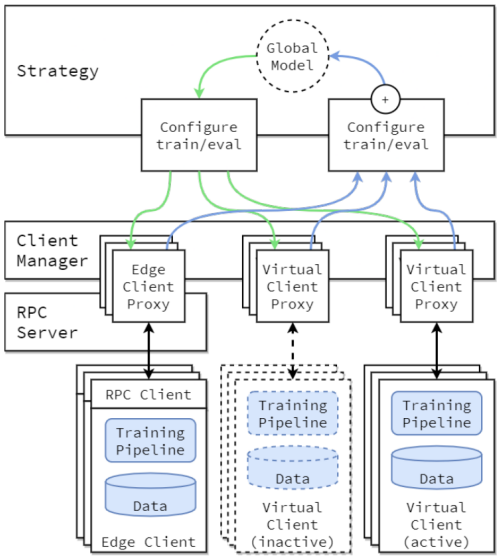

Figura 2.10 – Arquitetura do framework Flower Core com Edge Client Engine e Virtual Client Engine. Fonte: (Beutel et al., 2020). Nota: Os clientes de borda residem em dispositivos de borda reais e se comunicam com o servidor via RPC. Os clientes virtuais, por outro lado, consomem praticamente nenhum recurso quando inativos e carregam o modelo e os dados na memória somente quando o cliente é selecionado para treinamento ou avaliação.

2.1.8 *Overfitting* e Aumento de Dados

Um desafio crítico no treinamento de redes profundas é o *Overfitting*, fenômeno em que o modelo apresenta baixo erro nos dados de treino, mas falha em generalizar para dados novos, essencialmente "memorizando"o ruído do conjunto de treinamento (Shorten; Khoshgoftaar, 2019). Em linhas gerais, ocorre um sobreajuste dos parâmetros treináveis de um modelo de acordo com os dados de treino. Esse fenômeno é comumente desencadeado por três fatores primários: conjuntos de dados muito ruidosos, bases amostrais desbalanceadas e arquiteturas de rede excessivamente complexas. Para mitigar essas causas, (Ying, 2019) estabelece soluções diretas. A complexidade desnecessária pode ser tratada através da redução estrutural da rede. a partir da redução de camadas e nós, ou pela aplicação de métodos de regularização, como penalidades L1, L2 e o Dropout, que forçam a rede a ignorar características irrelevantes e a aprender de forma mais generalista. Já as deficiências e vieses dos dados podem ser corrigidos pela expansão do conjunto de treinamento, com destaque para o *data augmentation*, que injeta variabilidade artificial nas amostras. Por fim, para evitar a memorização contínua de ruídos, pode ser aplicado o *early-stopping*, uma técnica que monitora o aprendizado e interrompe o treinamento no exato momento em que o modelo deixa de aprender padrões úteis e começa a apenas decorar os dados (Ying, 2019).

No que tange à estratégia de Aumento de Dados, esta técnica consiste em aplicar transformações preservadoras de rótulo para inflar artificialmente o conjunto de treinamento (Shorten;

Khoshgoftaar, 2019). As transformações podem ser geométricas, como rotações, espelhamentos (*flipping*), cortes (*cropping*) e translações; ou fotométricas, envolvendo alterações no espaço de cores, brilho e contraste para simular diferentes condições de iluminação. Como tal proposta promove aumento do volume e variabilidade dos dados de treinamento, compreende-se que a medida em questão pode evitar a ocorrência de sobreajuste dos hiperparâmetros de treinamento (Mumuni; Mumuni, 2022).

Ainda segundo (Ying, 2019), outro fenômeno pode ser observado. O *underfitting* manifesta-se quando um modelo de aprendizado de máquina possui capacidade expressiva insuficiente para mapear a real complexidade e os padrões subjacentes presentes nos dados de treinamento. Esse fenômeno é caracterizado por um alto viés (*bias*), onde as hipóteses simplistas do algoritmo impedem a convergência para um erro residual aceitável, resultando em métricas de desempenho insatisfatórias tanto na fase de ajuste quanto na validação. Em contrapartida, o ajuste correto (*good fitting*) representa o ponto ideal de generalização, situado exatamente no limiar anterior ao *overfitting*. A Figura 2.11 ilustra como as regressões de modelo se adaptam em cada fenômeno de ajuste aos dados de treinamento.

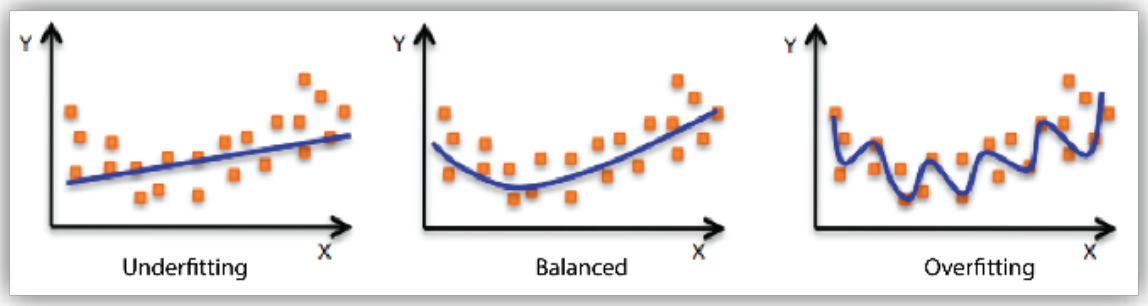

Figura 2.11 – Ilustração da adaptação das curvas nos casos de *Underfitting* e *overfitting*. Fonte: (V et al., 2020)

2.2 Trabalhos Relacionados

A agricultura de precisão (AP) começou a ser estudada conceitualmente no início do século XX, com pesquisas iniciais sobre pedologia e variabilidade do solo. No entanto, seu desenvolvimento técnico e aplicação prática com o nome de *"Precision Farming"* teve início nos anos 1980, ganhando força com a tecnologia de GPS e computação no começo dos anos 1990.

No que tange a colheita de cana-de-açúcar, (Cox; Harris; Cox, 1999) analisa a aplicação da agricultura de precisão na cultura da cana-de-açúcar por meio de um estudo de caso realizado em uma área agrícola na Austrália, com o objetivo de reduzir a variabilidade espacial da produtividade e otimizar os insumos utilizados. Inicialmente, foi desenvolvido um sistema inédito de mapeamento de produtividade para cana-de-açúcar, já que não existiam monitores comerciais disponíveis para essa cultura. Apesar de buscar otimizar insumos, o foco deste estudo não está na

identificação localizada de invasoras na colheita.

A detecção de plantas daninhas também já tem sido estudada na academia. Em (Muangkasem et al., 2010), é apresentado um sistema de visão computacional para a detecção desse tipo de infestação em áreas de entrelinhas de cana-de-açúcar, com foco na aplicação de herbicidas em taxa variável. Os autores destacam que o uso convencional de herbicidas em doses uniformes é economicamente ineficiente e ambientalmente insustentável, uma vez que as plantas daninhas ocorrem de forma espacialmente não uniforme nos canaviais. Tal estudo propõe um treinamento centralizado, que combina técnicas de identificação de vegetação verde e de segmentação do solo.

Em um estudo voltado para o cultivo de cana-de-açúcar, os autores de (Modi et al., 2023) investigaram o uso de modelos de *Deep Learning* para a identificação automática de plantas daninhas. Ao comparar seis arquiteturas diferentes, os autores constataram que o modelo *DarkNet53* apresentou desempenho significativamente superior aos demais (como *ResNet50, InceptionV3 e Xception*), alcançando um *F1-score* maior que 99% e uma acurácia de 96,6% em testes com dados independentes. Os resultados evidenciam a viabilidade de integrar essa tecnologia a sistemas de pulverização inteligente de baixo custo, otimizando o manejo agrícola.

Em (Zheng et al., 2024), é proposto um modelo de detecção de plantas daninhas em campos de algodão, denominado *YOLO-WL*, baseado na arquitetura *YOLOv8* e aprimorado com *EfficientNet* e mecanismos de atenção. O objetivo é aumentar a precisão e a velocidade de detecção em ambientes agrícolas complexos, mantendo viabilidade para dispositivos de borda. Os resultados mostraram que, após uma otimização do *TensorRT*, o modelo se mostrou adequado para a aplicação. Ainda assim, trata-se de uma aplicação centralizada, que não prioriza a manutenção da privacidade de dados.

Outro estudo relacionado é exposto em (Yano et al., 2016), que propõe um sistema de identificação de plantas daninhas em áreas de cultivo de cana-de-açúcar utilizando imagens obtidas por veículos aéreos não tripulados (UAVs) e técnicas de aprendizado de máquina, com destaque para o classificador *Random Forest*. A metodologia empregada baseia-se na aquisição de imagens RGB de alta resolução capturadas por UAVs em baixa altitude, permitindo a detecção de plantas daninhas mesmo em níveis reduzidos de infestação e sem interferência de nuvens, uma limitação comum em imagens de satélite. A partir dessas imagens, amostras das principais classes vegetais foram extraídas manualmente e utilizadas para o treinamento supervisionado do classificador *Random Forest*. Nesse caso, tem-se maior autonomia para o produtor na aquisição dos dados, mas ainda se emprega um treinamento centralizado, no qual a proteção dos dados não é priorizada.

(Arangi; Padhy, 2025) é um estudo mais recente, que aplica técnicas de Aprendizagem Federada para a detecção e manejo das plantas daninhas na agricultura de precisão. Esse estudo foi baseado na integração de *Federated Edge Learning* (FEL), aprendizado profundo e fusão de sensores multimodais. O objetivo central é melhorar a acurácia da detecção de plantas daninhas

ao mesmo tempo em que se reduzem a latência, o consumo energético e os riscos associados ao compartilhamento de dados sensíveis em ambientes agrícolas distribuídos. De acordo com o autor, no contexto da agricultura de precisão, especificamente na detecção de plantas daninhas utilizando fusão de sensores multimodais, as limitações técnicas descritas são críticas, uma vez que os dados coletados em campo são inerentemente *Non-IID*: as diferentes fazendas ou regiões geográficas apresentam variações significativas no solo, clima e espécies de pragas predominantes. Uma cooperativa pode ter uma infestação massiva de uma espécie resistente a herbicidas, enquanto outra fazenda vizinha possui uma flora completamente distinta, resultando em sintomas visuais e assinaturas espectrais divergentes nas folhagens (Arangi; Padhy, 2025). Assim sendo, O uso de uma função de agregação robusta como o FedProx é fundamental para garantir que o modelo global de detecção de ervas daninhas não seja enviesado pelas características de um único nó dominante que possua um volume massivo de dados. Além disso, a capacidade do FedProx de lidar com *stragglers*, isto é, dispositivos clientes cujo *hardware* limitado ou conexão de internet lenta atrasa a conclusão do treinamento local, é vital para o agronegócio, onde sensores de borda operam sob condições instáveis de conectividade rural e energia. Conforme demonstrado nos experimentos da pesquisa, a implementação do FedProx permitiu atingir uma acurácia de 94,1% na detecção distribuída, superando o FedAvg (93,7%) e garantindo uma solução eficiente, com menor latência (120 ms) e consumo energético reduzido (300 mWh) para o monitoramento agrícola em tempo real.

Embora existam muitos estudos sobre o uso de Inteligência Artificial (IA) e Aprendizado de Máquina (ML) na cultura da cana-de-açúcar, a aplicação específica de Aprendizado Federado (FL) para a identificação de plantas daninhas ainda representa um nicho acadêmico pouco explorado. Em linhas gerais, os estudos concentram-se na aplicação de arquiteturas centralizadas. Essa dependência cria um impasse prático, no qual a relutância dos grandes produtores em compartilhar dados agronômicos estratégicos impede a criação de modelos colaborativos robustos.

Tendo em vista a notória relevância do mercado açucareiro nacional, bem como a prioridade na preservação de dados sensíveis dos grandes produtores, este estudo propõe uma abordagem baseada em Aprendizado Federado para a identificação de plantas daninhas em campos de colheita. Portanto, oferece-se uma solução inédita que concilia o desempenho dos modelos de estado da arte com a segurança de dados exigida pelo competitivo setor sucroenergético nacional.

3 Desenvolvimento

Para atender aos objetivos deste trabalho e garantir a reprodutibilidade dos experimentos, a metodologia foi organizada em um fluxo de cinco etapas sequenciais: (1) Aquisição e Preparação dos Dados; (2) Definição da Arquitetura dos Modelos; (3) Configuração do Treinamento Centralizado; (4) Protocolo de Aprendizagem Federada; e (5) Métricas de Avaliação. A seguir, detalha-se cada uma dessas etapas.

3.1 Etapa 1: Aquisição e Pré-processamento dos Dados

O conjunto de dados utilizado foi o *Sugarcane and Weed*, disponível publicamente na plataforma *Kaggle* (Domah, 2025). O *dataset* original é composto por imagens com dimensões originais de 240×320 *pixels* (Figura 3.1. O problema foi modelado como uma classificação binária, contendo as classes:

- **Daninha**: 522 imagens contendo infestação de plantas invasoras.
- **Não-Daninha**: 200 imagens contendo apenas a cultura da cana-de-açúcar.

(a) Exemplo da Classe Daninha (b) Exemplo da Classe Não-Daninha

Figura 3.1 – Exemplo de imagens do *Dataset*.

Para adequar os dados à entrada da rede neural e garantir a robustez do treinamento, foi aplicado um *pipeline* de pré-processamento e limpeza:

1. **Limpeza e Remoção de Arquivos Corrompidos**: Antes de qualquer transformação, um *script* automatizado varreu os diretórios testando a integridade de cada imagem por meio da biblioteca *Pillow* (PIL), removendo instâncias corrompidas que poderiam causar falhas durante o treinamento.

2. **Balanceamento e Aumento de Dados (*Data Augmentation*)**: Tendo em vista o desbalanceamento inicial entre as classes, identificou-se a proporção máxima de desequilíbrio e aplicaram-se técnicas de aumento de dados sintético na classe subamostrada. O objetivo foi equalizar a distribuição de amostras de forma estrita. Utilizando a biblioteca `imgaug`, o *pipeline* de transformações incluiu o espelhamento horizontal (`Fliplr(1.0)`) (Figura 3.2) e transformações de perspectiva (`PerspectiveTransform` com escala entre 0.01 e 0.15), de modo a resultar em 200 variações da classe minoritária (Não-Daninha). Ao final, o dataset contou com 922 imagens, sendo 400 para a classe "Não-Daninha".

(a) Exemplo original da Não-Daninha (b) Exemplo espelhado da Classe Não-Daninha

Figura 3.2 – Uso da aumentação de imagens para balancear o *Dataset*.

3. **Prevenção de Vazamento de Dados**: Uma etapa crítica na preparação foi a implementação de uma lógica de agrupamento por chave base. Como o *data augmentation* gera variações de uma mesma imagem original (com prefixos como `aug_`), desenvolveu-se uma função de particionamento estratificado. Isso assegurou que uma imagem original e suas versões

aumentadas fossem alocadas inteiramente para o mesmo conjunto (treino, validação ou teste), eliminando o risco do modelo "decorar"imagens e mascarar resultados no teste.

4. **Particionamento, Carregamento e Redimensionamento**: No contexto centralizado, os grupos de imagens foram divididos em **Treino (70%)**, **Validação (15%)** e **Teste (15%)**. As imagens foram lidas, redimensionadas para as dimensões 224×224 pixels (ou 128×128 para o cenário federado) no formato *RGB*, e os valores de *pixels* normalizados para o intervalo $[0, 1]$. O carregamento se deu em lotes (*batches*) de tamanho 4 para otimizar o uso da memória, empregando técnicas de *prefetching* com `tf.data.AUTOTUNE` para paralelizar a Entrada/Saída (E/S) de dados.

3.2 Etapa 2: Arquitetura dos Modelos

No ambiente centralizado, foram instanciadas e avaliadas três arquiteturas base: **EfficientNetB0**, **ResNet50**, ambas com pesos pré-treinados via carregamento dos pesos da base *ImageNet*, e uma implementação customizada da **DarkNet53**.

A arquitetura Da DarkNet53 implementada neste trabalho (Algoritmo 1) foi adaptada a partir do modelo proposto por (Yakhyo, 2024). Essa adaptação fez-se necessária uma vez que a versão original do código foi construída com base no framework PyTorch, enquanto a primeira parte deste projeto havia sido desenvolvida utilizando o framework TensorFlow.

Com base nos resultados satisfatórios dos experimentos iniciais, e visando um modelo mais flexível, eficiente e computacionalmente adequado para o intercâmbio de pesos em múltiplos clientes no Aprendizado Federado, adotou-se o modelo **DarkNet53** como base principal da pesquisa. O fluxo de dados da rede foi estruturado da seguinte forma:

1. **Camadas Convolucionais Base (*Darknet Conv*)**: Operações de convolução bidimensionais sem viés, seguidas sequencialmente por camadas de Normalização em Lote (*Batch Normalization*) e ativações não-lineares *LeakyReLU* com inclinação $\alpha = 0.1$.

2. **Blocos Residuais (*Darknet Blocks*)**: Utilização intensiva de conexões residuais (*shortcuts*). Cada bloco consiste em uma convolução 1×1 para redução de dimensionalidade, seguida de uma convolução 3×3, cujos mapas de características resultantes são somados à entrada original do bloco. A arquitetura empilha esses blocos sequencialmente nas proporções de 1, 2, 8, 8 e 4 blocos para capturar características de baixíssimo a altíssimo nível.

3. **Cabeçalho de Classificação (*Top Head*)**: Para adaptar a rede à tarefa binária, foram adicionadas ao topo da base convolucional:

    - ***Global Average Pooling 2D***: Reduz a dimensionalidade espacial para um vetor de médias.

- ***Dropout***: Camada de regularização com taxa de 0,2 (20%) para mitigar o sobreajuste (*overfitting*).
- **Camada de Saída**: Uma camada densa (*Dense*) com 1 neurônio e função de ativação **Sigmoide**, responsável por retornar a probabilidade da imagem pertencer à classe "daninha".

Algoritmo 1 Construção da Arquitetura Customizada DarkNet53

Entrada: FormatoEntrada (dimensões da imagem, ex: $128 \times 128 \times 3$)
Saída: Modelo de Rede Neural compilado para classificação binária

function DarknetConv(x, filtros, tamanhoKernel, passos)
  $x \leftarrow$ Conv2D(x, filtros $=$ filtros, kernel $=$ tamanhoKernel, strides $=$ passos)
  $x \leftarrow$ BatchNormalization(x)
  $x \leftarrow$ LeakyReLU($x, \alpha = 0.1$)
  return x
end function

function DarknetBlock(x, filtros, numBlocos)
  for $i \leftarrow 1$ **até** numBlocos **do**
    atalho $\leftarrow x$
    $x \leftarrow$ DarknetConv(x, filtros$/2, 1, 1$) ▷ Redução de dimensionalidade
    $x \leftarrow$ DarknetConv(x, filtros, $3, 1$) ▷ Extração de características
    $x \leftarrow$ Soma(atalho, x) ▷ Conexão residual (shortcut)
  return x
end function

function ConstroiDarkNet53(FormatoEntrada)
  entrada $\leftarrow$ Input(FormatoEntrada)
  $x \leftarrow$ DarknetConv(entrada, $32, 3, 1$)
  $x \leftarrow$ DarknetConv($x, 64, 3, 2$)
  $x \leftarrow$ DarknetBlock($x, 64, 1$)
  $x \leftarrow$ DarknetConv($x, 128, 3, 2$)
  $x \leftarrow$ DarknetBlock($x, 128, 2$)
  $x \leftarrow$ DarknetConv($x, 256, 3, 2$)
  $x \leftarrow$ DarknetBlock($x, 256, 8$)
  $x \leftarrow$ DarknetConv($x, 512, 3, 2$)
  $x \leftarrow$ DarknetBlock($x, 512, 8$)
  $x \leftarrow$ DarknetConv($x, 1024, 3, 2$)
  $x \leftarrow$ DarknetBlock($x, 1024, 4$)
  ▷ Cabeçalho de Classificação Binária
  $x \leftarrow$ GlobalAveragePooling2D(x)
  $x \leftarrow$ Dropout(x, taxa $= 0.2$)
  saída $\leftarrow$ Dense(x, neurônios $= 1$, ativação $=$ 'sigmoid')
  return Modelo(entrada, saída)
end function

3.3 Etapa 3: Configuração do Treinamento Centralizado

Nesta etapa, estabeleceu-se o *baseline* de desempenho utilizando a abordagem clássica e centralizada.

3.3.1 Ambiente Computacional

Todos os experimentos, tanto centralizados quanto federados, foram executados e validados no *Cluster* do Laboratório CSILab. Especificamente, utilizou-se o servidor batizado de *Nevasca*, uma máquina robusta composta por um processador *AMD Ryzen Threadripper 3960X* de 24 núcleos físicos (48 *threads*) operando a 3.70GHz, acompanhado de 128GB de memória RAM DDR4. O processamento gráfico contou com uma *GPU NVIDIA RTX 3090* com 24GB de VRAM GDDR6X e mais de 10 mil *CUDA cores*. O armazenamento central das imagens foi orquestrado via *Synology NAS* de 10TB acessível via rede de alto desempenho.

Para otimizar o uso desse ambiente, ativou-se o crescimento dinâmico de memória em *GPU* e utilizou-se a política de Precisão Mista (`mixed_float16`) do *TensorFlow*.

3.3.2 Protocolo Experimental e Hiperparâmetros

O treinamento centralizado operou como parâmetro de controle. A configuração base de hiperparâmetros foi estabelecida conforme a Tabela 3.1:

Parâmetro	**Configuração**
Otimizador	*Adam*
Taxa de Aprendizado (*Learning Rate*)	1×10^{-4} (0.0001)
Função de Perda	*Binary Cross-Entropy*
Tamanho do Lote (*Batch Size*)	4
Número de Épocas	15
Dimensão da Imagem	$224 \times 224 \times 3$

Tabela 3.1 – Hiperparâmetros utilizados no treinamento centralizado

Ademais, o fluxo de treinamento centralizado (Algoritmo 2) executa múltiplos experimentos repetidos (3 rodadas), para atestar estabilidade estatística:

Algoritmo 2 Protocolo de Treinamento Centralizado Baseline

Entrada: Dados de entrada ($Dados_{\text{Treino}}$, $Dados_{\text{Val}}$, $Dados_{\text{Teste}}$)
Saída: Modelos treinados e métricas de desempenho consolidadas

```
ModelosBase ← [EfficientNetB0, ResNet50, DarkNet53]
for cada Arquitetura em ModelosBase do
    for run ← 1 até 3 do                    ▷ Garantir robustez via múltiplas execuções
        Inicializar pesos da Arquitetura (ImageNet ou aleatórios)
        Configurar política de Precisão Mista do TensorFlow (mixed_float16)
        Compilar modelo utilizando Otimizador Adam
        for epoch ← 1 até 15 do
            Ajustar pesos iterativamente em lotes de tamanho 4 com Dados_Treino
            Avaliar e monitorar desempenho usando Dados_Val
        Avaliar modelo consolidado no conjunto independente Dados_Teste
        Registrar Métricas Finais: Acurácia, Recall, F1-Score e Loss
```

3.4 Etapa 4: Protocolo de Aprendizagem Federada

Para simular o ecossistema descentralizado, empregou-se o *framework* ***Flower*** (`flwr`), dividindo a arquitetura em um Servidor Global e múltiplos Clientes independentes. Visando abranger diferentes contextos agrícolas corporativos, o estudo foi estruturado em dois cenários metodológicos distintos:

3.4.1 Experimento 1: Cenário IID com estratégia FedAvg

Neste experimento basal (Algoritmo 3), partiu-se da premissa conceitual onde os dados estão distribuídos de maneira Independente e Identicamente Distribuída (*IID*).

- **Distribuição**: Os dados gerais do foram fragmentados entre 5 clientes (`num-partitions = 5`) com uma distribuição homogênea de classes. Todos os clientes possuíam uma parcela similar de imagens representativas da cultura e da planta invasora.
- **Agregação**: O servidor global operou com o algoritmo *Federated Averaging* (**FedAvg**), que realiza a consolidação iterativa do modelo tirando a média simples, ponderada pela quantidade de amostras de cada cliente.

Algoritmo 3 Aprendizado Federado: Cenário Homogêneo (IID) com FedAvg

Entrada: $K = 5$ clientes ativos, Fração de participação $C = 0.6$, Rodadas Globais $R = 15$
Saída: Modelo global agregado w_R atualizado ao fim das rodadas

Servidor distribui o *dataset* uniformemente entre os K clientes
Servidor inicializa os pesos globais iniciais w_0 da arquitetura **DarkNet53**
for cada rodada global $t = 1, 2, \ldots, R$ **do**
    Servidor seleciona aleatoriamente um subconjunto S_t de tamanho $\max(C \cdot K, 1)$ clientes
    for cada cliente $k \in S_t$ **em paralelo do**
        Baixar modelo global atualizado w_{t-1} do servidor
        Instanciar dados locais redimensionados para $128 \times 128 \times 3$
        for cada época local de 1 até 10 **do**
            Treinar modelo local via Otimizador SGD (LR $= 0.0001$, Momento $= 0.9$)
        Devolver novos pesos locais w_t^k e o número de exemplos treinados n_k ao servidor
    $n \leftarrow \sum_{k \in S_t} n_k$
    $w_t \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_t^k$ ▷ Agregação clássica via algoritmo FedAvg

3.4.2 Experimento 2: Cenário Non-IID com estratégia FedProx

Buscando mapear a imprevisibilidade do campo, onde uma área monitorada por um cliente pode apresentar infestação severa enquanto outra se encontra totalmente limpa, introduziu-se a heterogeneidade estatística (Algoritmo 4).

- **Distribuição via Dirichlet**: Para particionar o conjunto original garantindo viés local forte e caracterizando o cenário *Non-IID*, aplicou-se uma função da distribuição de probabilidade de *Dirichlet* estipulando uma variável de concentração $\alpha = 0.5$. Essa formulação garante matematicamente um profundo desbalanceamento no nível de silagem.
- **Agregação**: Sabendo que a heterogeneidade extrema compromete a convergência do *FedAvg*, este experimento implementou o algoritmo **FedProx**. Ao otimizador, foi adicionado o termo de regularização proximal $\mu = 5.0$. Essa penalidade intrínseca restringe e ancora as atualizações dos gradientes locais durante as épocas, impedindo que os pesos do cliente desviem exorbitantemente do escopo do modelo global.

Parâmetros Operacionais Federados: Para viabilizar a fluidez de dados pela arquitetura de rede e atenuar interrupções, ambas as abordagens utilizaram o redimensionamento das entradas para dimensões de $128 \times 128 \times 3$. O ciclo determinou 15 rodadas globais de comunicação. A cada ciclo, foi definida a participação de uma fração de clientes treinando o *backbone DarkNet53* simultaneamente ao longo de 10 épocas locais, valendo-se do otimizador *SGD* (momento de 0.9 e *LR* de 0.0001). A capacidade da máquina *Nevasca* provou-se essencial aqui, visto que suas 48 *threads* possibilitaram que as funções de *pipeline* (`AUTOTUNE`) operassem assincronamente os mapeamentos IID e Non-IID diretamente para os tensores sem gargalos na CPU.

Algoritmo 4 Aprendizado Federado: Cenário Heterogêneo (Non-IID) com FedProx

Entrada: Clientes $K = 5$, Concentração $\alpha = 0.5$, Termo Proximal $\mu = 5.0$, Rodadas $R = 15$
Saída: Modelo global agregado w_R mitigado contra desvios locais

Servidor divide dados aplicando a **Distribuição de Dirichlet**(α) por classe entre os clientes
Servidor inicializa os pesos globais iniciais w_0 da arquitetura **DarkNet53**
for cada rodada global $t = 1, 2, \ldots, R$ **do**
    Servidor seleciona o subconjunto S_t contendo $\max(C \cdot K, 1)$ clientes
    for cada cliente $k \in S_t$ **em paralelo do**
        Receber os pesos globais consolidados w_{t-1}
        for cada época local de 1 até 10 **do**
            Otimizar o modelo minimizando a função de perda modificada do **FedProx**:
            $\mathcal{L}_{\text{prox}}(w) = \mathcal{L}_{\text{local}}(w) + \frac{\mu}{2}\|w - w_{t-1}\|^2$ ▷ Termo restritivo contra desvios
            Atualizar os pesos locais usando SGD (LR $= 0.0001$, Momento $= 0.9$)
        Enviar novos pesos modificados w_t^k e cardinalidade de dados n_k ao servidor
    $n \leftarrow \sum_{k \in S_t} n_k$
    $w_t \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_t^k$ ▷ Média ponderada dos pesos limitados pelo termo proximal

3.5 Etapa 5: Métricas de Avaliação

As métricas de avaliação de Modelos de *Machine Learning* são utilizadas para avaliar os padrões de aprendizado e performance dos classificadores. Destacam-se as métricas registradas durante os treinamentos: Acurácia, *Recall*, Precisão, *F1-Score*, e *Loss*. Elas avaliam tanto a capacidade de acerto global quanto a segurança na detecção da classe positiva (daninha), dado o custo do Falso Negativo na agricultura.

Para que tais conceitos sejam compreendidos, deve-se entender a Matriz de Confusão. Trata-se de uma tabela na qual as linhas representam as classes previstas e as colunas as reais. Pode ser representada da seguinte maneira:

	Predito: Negativo	**Predito: Positivo**
Real: Negativo	Verdadeiro Negativo (VN)	Falso Positivo (FP)
Real: Positivo	Falso Negativo (FN)	Verdadeiro Positivo (VP)

Tabela 3.2 – Matriz de Confusão para Classificação Binária

- Verdadeiro Positivo (VP): O modelo previu a classe positiva corretamente.
- Verdadeiro Negativo (VN): O modelo previu a classe negativa corretamente.
- Falso Positivo (FP): O modelo previu positivo, mas a classe real era negativa (Erro Tipo I).
- Falso Negativo (FN): O modelo previu negativo, mas a classe real era positiva (Erro Tipo II).

3.5.1 Acurácia

Representa a proporção de predições corretas sobre o total de casos observados. É utilizada como indicador geral de performance.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.1}$$

3.5.2 *Recall* (Revocação)

Quantifica a sensibilidade do modelo na detecção de instâncias pertencentes à classe de interesse. Sua aplicação é prioritária quando a ocorrência de Falsos Negativos implica riscos críticos. No caso específico deste estudo, a não identificação de plantas daninhas pode comprometer a produtividade, logo, a maximização desta métrica é crucial.

$$\text{Recall} = \frac{VP}{VP + FN} \tag{3.2}$$

3.5.3 Precisão

Indica a confiabilidade das detecções positivas, ou seja, quantas das plantas classificadas como daninhas realmente o eram.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{3.3}$$

3.5.4 *F1-Score*

É a média harmônica entre a Precisão e o *Recall*. Sua aplicação é particularmente vantajosa em cenários nos quais a Acurácia pode mascarar deficiências do modelo. Ao utilizar a média harmônica, o *F1-Score* penaliza resultados extremos, garantindo que o modelo apresente um desempenho equilibrado.

$$F1\text{-}Score = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{3.4}$$

3.5.5 *Binary Cross-Entropy (Log Loss)*

Avalia a incerteza das predições ao considerar as probabilidades contínuas estimadas pelo modelo. Quanto mais distante a probabilidade estiver do rótulo verdadeiro, maior será a penalidade aplicada. Uma baixa *Log Loss* indica que o modelo acerta a classe com alta confiança estatística. Sua expressão é dada por:

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}i)] \tag{3.5}$$

Onde:

- $J(\theta)$: representa a função de custo a ser minimizada;
- N: é o número total de observações no conjunto de dados;
- $\sum i = 1^N$: representa o somatório das perdas individuais;
- y_i: é o valor real da classe (0 ou 1) para a amostra i;
- $\hat{y}_i$: é a probabilidade estimada pelo modelo para a classe positiva.

4 Resultados

Neste capítulo constam os resultados dos experimentos descritos em 3, bem como as suas respectivas interpretações e discussões.

4.1 Arquiteturas Centralizadas

Os experimentos com arquiteturas centralizadas demonstram convergência eficiente para as configurações da *DarkNet53*, ao passo que evidenciam dificuldades severas para os modelos *EfficientNetB0* e *ResNet50*, cujas curvas de validação se mostraram bastante ruidosas. A *EfficientNet*, especificamente, apresenta picos drásticos de erro e quedas abruptas nas métricas de acerto (Figura 4.1). Da mesma maneira, a *ResNet50* também apresentou instabilidade no treinamento, tendo em vista a grande faixa de desvio dos indicadores de performance e o aumento da métrica de *Loss* ao final (Figura 4.2).

A adoção de um lote de 4 imagens se deu como uma estratégia metodológica imposta pelo escopo restrito do *dataset*. Contendo apenas 922 imagens no total, lotes maiores resultariam em poucas atualizações de pesos por época, dificultando a convergência. Contudo, esse choque de desempenho observado entre as arquiteturas reflete limitações bem documentadas na literatura de Aprendizado Profundo. Redes baseadas em ativações *ReLU* e com forte dependência de *Batch Normalization*, como a *ResNet50* e a *EfficientNetB0*, sofrem drástica degradação quando submetidas a lotes de treinamento reduzidos durante o *fine-tuning*.

Conforme demonstrado por (Wu; He, 2018) e endossado na prática estrutural da *Mask R-CNN* (He et al., 2017), ao forçar lotes pequenos para maximizar as iterações, o cálculo da média e variância na *Batch Normalization* torna-se estatisticamente impreciso (Singh; Krishnan, 2020). Esse recálculo com amostras diminutas entra em conflito com os pesos pré-treinados, introduzindo ruído severo nos gradientes e corrompendo a rede nas épocas iniciais, o que justifica o baixo desempenho da *EfficientNet* e da *ResNet*. Além disso, o módulo *Squeeze-and-Excitation* da *EfficientNet* demonstra-se altamente sensível à falta de representatividade estatística em lotes pequenos para a recalibração de seus canais (Hu; Shen; Sun, 2018). Somado a isso, as flutuações violentas nos gradientes sob a ativação *ReLU* padrão (*ResNet50*) levam à inatividade permanente de neurônios quando expostos a valores fortemente negativos, fenômeno conhecido como *Dying ReLU* (Lu et al., 2020).

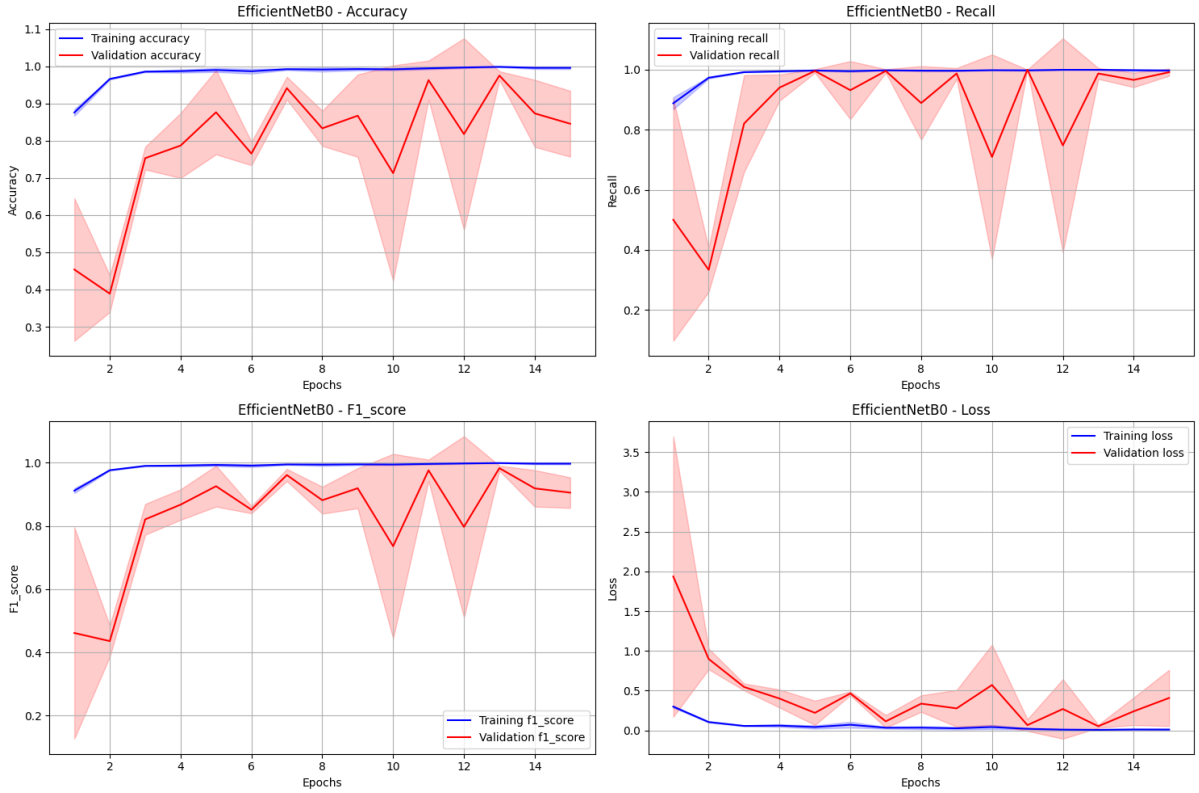

Figura 4.1 – Experimento executado com a arquitetura *EfficientNetB0*. Fonte: elaboração própria.

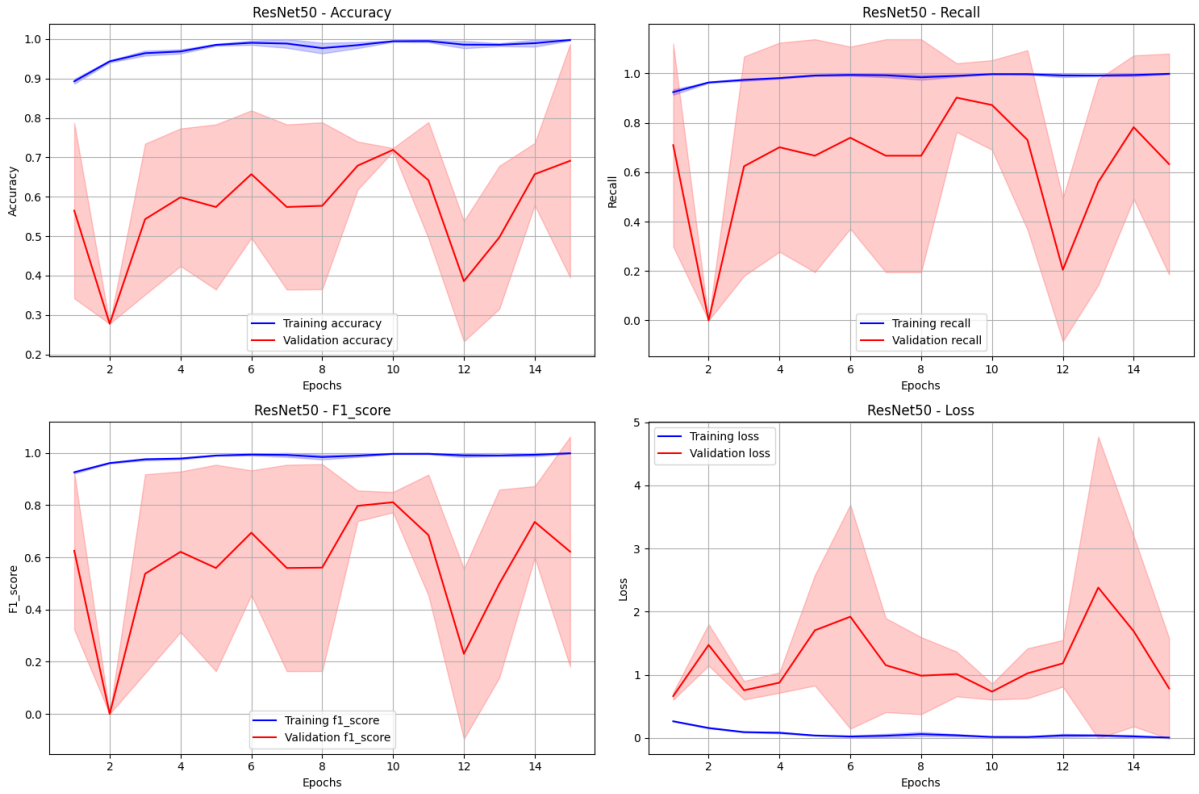

Figura 4.2 – Experimento executado com a arquitetura *ResNet50*. Fonte: elaboração própria.

Em contrapartida, no caso da *DarkNet53*, as curvas de treinamento e validação seguem a

mesma tendência de crescimento ou decaimento. A *Loss* cai de forma consistente, enquanto as métricas de Acurácia, *Recall* e *F1-Score* sobem de forma estável, convergindo rapidamente para valores próximos a 1.0 e com uma pequena faixa de desvio padrão (Figura 4.3).

A *DarkNet53* superou tais restrições devido à sua topologia baseada na alternância bruta de filtros convolucionais 1×1 e 3×3 com conexões residuais (Redmon, 2018), aliada ao uso da ativação *LeakyReLU*. O emprego desta função evita o colapso neuronal ao permitir que um gradiente atenuado passe para valores negativos (Xu et al., 2015), mitigando o problema de morte dos neurônios frente ao ruído estocástico dos lotes menores. Além disso, por ter sido inicializada com pesos aleatórios, a evolução dos pesos e os parâmetros de sua normalização coadaptaram-se harmonicamente ao comportamento dos lotes reduzidos desde a primeira época, garantindo uma convergência estável e completamente isolada das falhas geradas pelo choque de domínios pré-treinados.

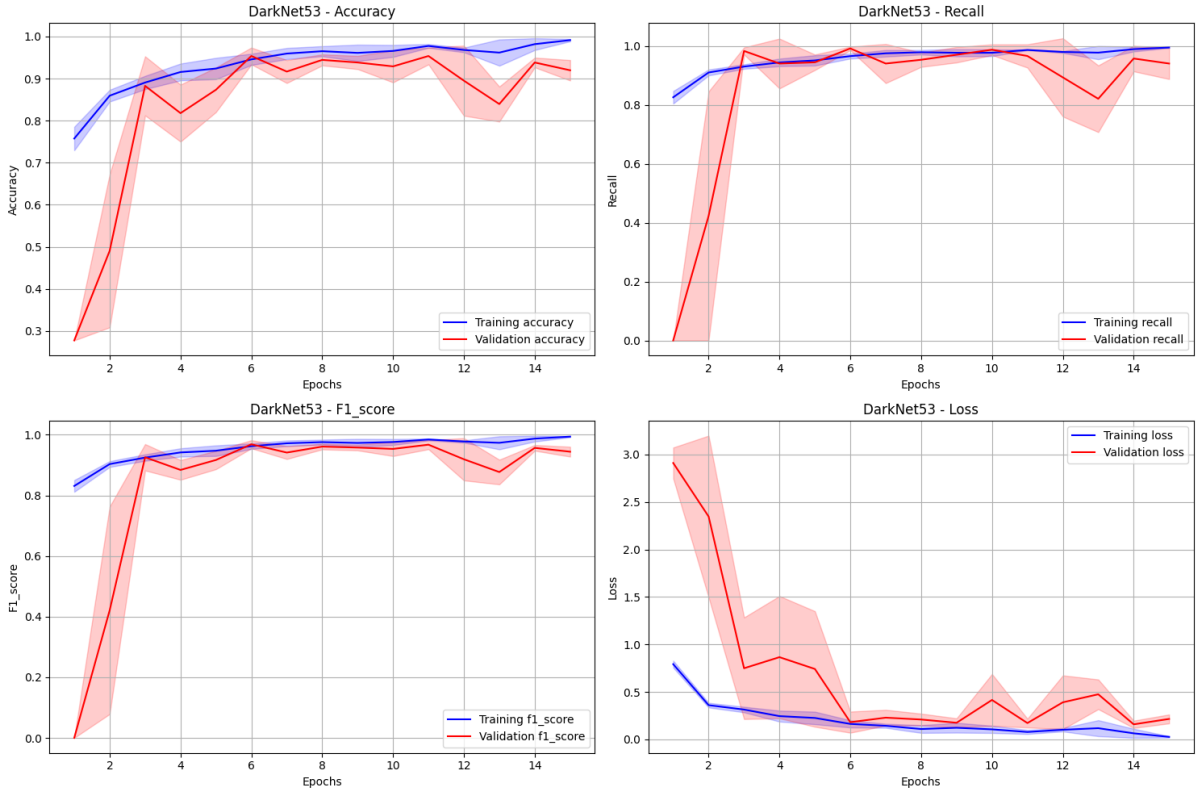

Figura 4.3 – Experimento executado com a arquitetura *DarkNet53*. Fonte: elaboração própria.

A fim de mitigar vieses estocásticos, cada experimento foi executado independentemente 3 vezes. A análise da distribuição de desempenho, ilustrada pelos *boxplots* (Figuras 4.4, 4.5 e 4.6), consolida a diferença de estabilidade entre as arquiteturas avaliadas ao longo das três execuções independentes. Observa-se que os modelos EfficientNetB0 e ResNet50 apresentam caixas de acurácia mais alongadas, o que evidencia uma alta dispersão e instabilidade nos resultados. Essa instabilidade no teste é o reflexo direto da degradação das camadas de *Batch Normalization* supra-citada, visto que uma execução pode apresentar desempenho aceitável enquanto a seguinte pode

falhar de forma drástica. Em contrapartida, a DarkNet53 exibe caixas notavelmente compactas e com valores medianos superiores, comprovando a sua robustez arquitetural e insensibilidade às flutuações estocásticas de inicialização.

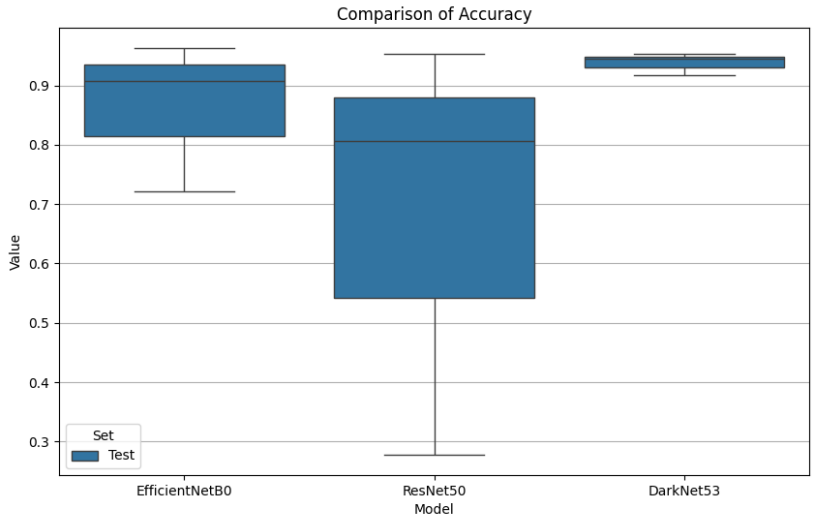

Figura 4.4 – Distribuição da métrica de Acurácia no conjunto de teste após 3 execuções independentes. A *DarkNet53* demonstra superioridade e variância quase nula. Fonte: elaboração própria.

Embora a métrica de Recall seja crítica no contexto agronômico, visto que a não identificação de uma planta daninha, isto é, um FN, gera perdas diretas de produtividade por matocompetição, a escolha definitiva do algoritmo base não pode se sustentar apenas na sensibilidade. Caso contrário, correr-se-ia o risco de selecionar um classificador enviesado, propenso a rotular qualquer folhagem como invasora. Por esse motivo, a justificativa central para a adoção da DarkNet53 fundamenta-se na métrica de F1-Score, calculada pela média harmônica entre a Precisão e o Recall. Ao penalizar desbalanceamentos extremos, o F1-Score garante que o modelo possua um desempenho equilibrado, minimizando falsos positivos ao mesmo tempo em que detecta eficientemente as infestações.

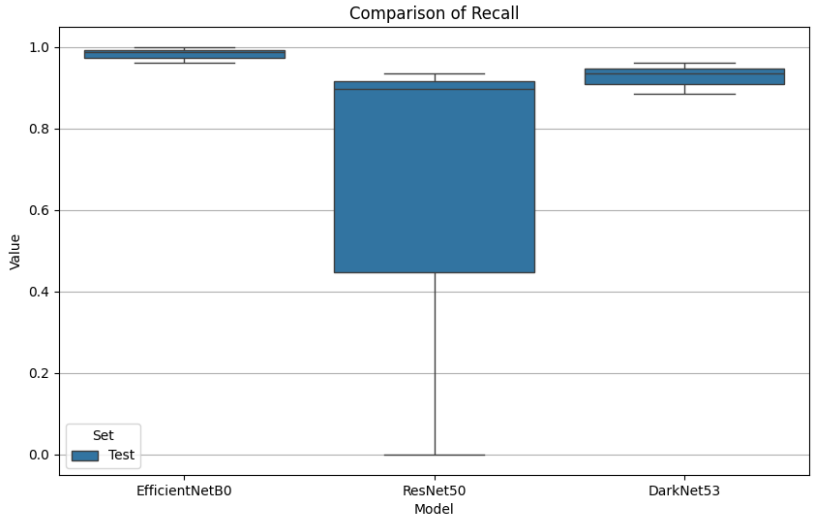

Figura 4.5 – Distribuição da métrica de *Recall* no conjunto de teste após 3 execuções independentes. Fonte: elaboração própria.

Como demonstrado pela análise de distribuição do F1-Score (Figura 4.6), a DarkNet53 manteve a maior consistência e os maiores valores absolutos da métrica entre as redes testadas. Esse comportamento atesta que o balanceamento sintético do dataset foi eficaz e que a arquitetura é capaz de discriminar ambas as classes com exatidão no conjunto de teste. Diante da comprovada capacidade de generalização equilibrada e da baixa variância atestada pelos boxplots, a DarkNet53 consagra-se formalmente como o baseline ideal e seguro para instanciar os clientes no ecossistema de Aprendizado Federado.

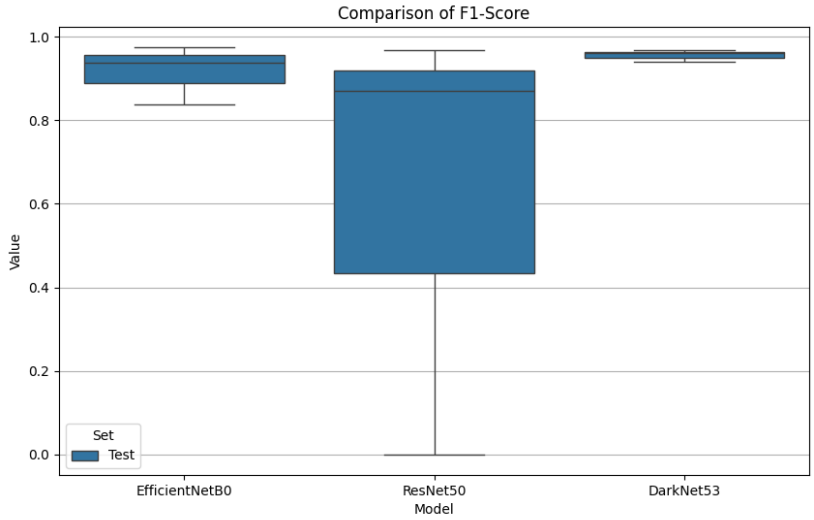

Figura 4.6 – Distribuição da métrica de *F1-Score* no conjunto de teste. A *DarkNet53* atesta sua capacidade de discriminação equilibrada entre as classes. Fonte: elaboração própria.

Em suma, as análises de distribuição no conjunto de teste validam a arquitetura *Dark-*

Net53 como o *baseline* ideal deste estudo. Sua alta taxa de acerto e baixa variância justificam formalmente sua seleção para instanciar os clientes nas subsequentes simulações do ecossistema de Aprendizado Federado.

4.2 Treinamento Federado no Cenário IID

Os experimentos conduzidos no ecossistema federado, partindo do cenário com distribuição Independente e Identicamente Distribuída (IID), corroboram os preceitos fundamentais estabelecidos na literatura de Aprendizado Descentralizado. Nesta configuração, o *dataset* foi homogeneamente particionado entre 5 clientes, garantindo que cada nó retivesse proporções equivalentes das classes "daninha"e "não-daninha". O algoritmo de agregação eleito para o Servidor Global foi o *Federated Averaging (FedAvg)*.

Conforme ilustrado nas curvas de desempenho globais (Figura 4.7), o modelo *DarkNet53* obteve uma convergência global notavelmente suave e estável. A *Loss* decai monotonamente enquanto as métricas de Acurácia e *Recall* ascendem de forma consistente ao longo das 15 rodadas de comunicação, superando a marca de 90% de assertividade. Esse comportamento reflete a proposição original de (McMahan et al., 2017), que comprova que o *FedAvg* alcança eficiência simétrica à de um treinamento centralizado sob configurações de dados IID, uma vez que a direção dos gradientes calculados localmente aponta para o mesmo objetivo ótimo de minimização.

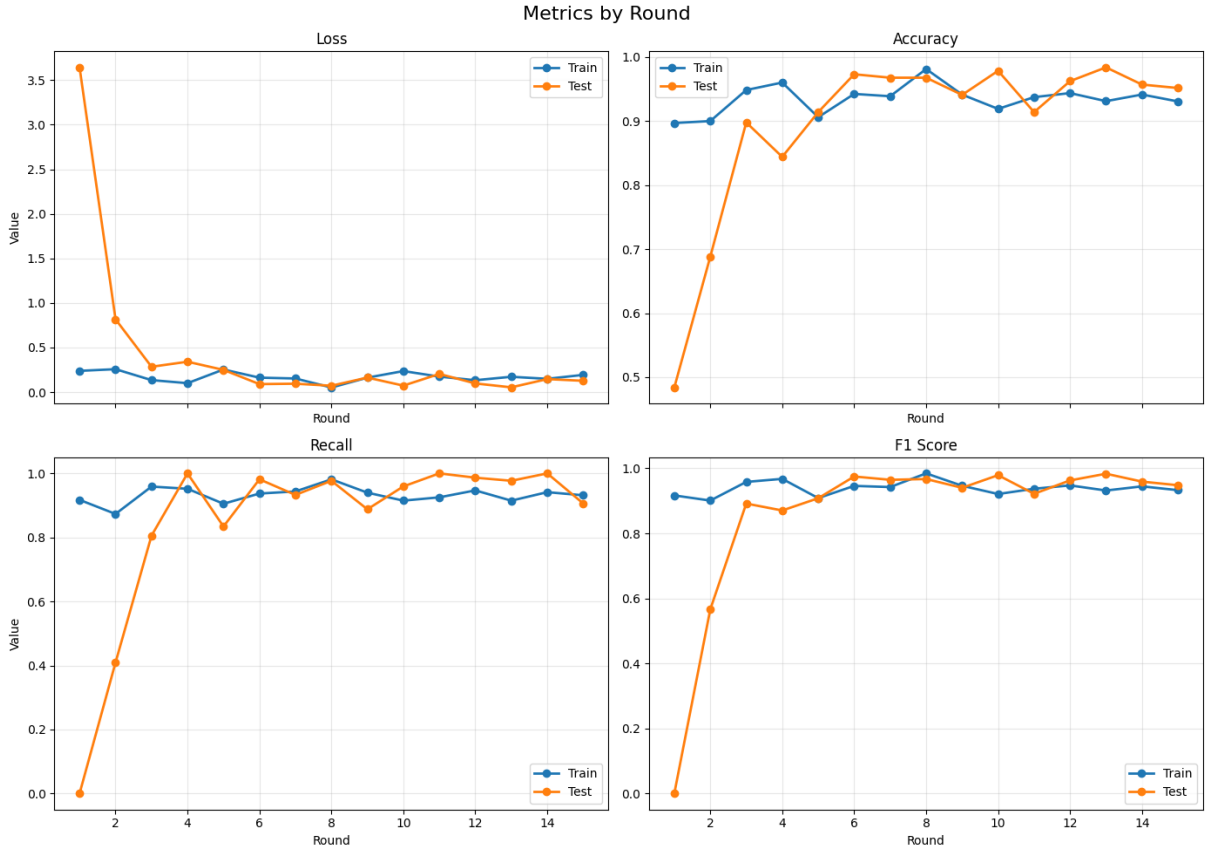

Figura 4.7 – Evolução das métricas globais (Acurácia, Loss, Precision e Recall) ao longo de 15 rodadas no cenário IID. Fonte: elaboração própria.

Um comportamento evidenciado nesta etapa diz respeito ao efeito regularizador da agregação global frente à escassez de dados locais. Tendo em vista o tamanho restrito do *dataset* original, cada cliente processou volumes inferiores a 200 amostras locais por rodada. Como consequência inerente da estocasticidade, as métricas individuais dos clientes apresentam oscilações entre as rodadas (Figura 4.8). Contudo, o Servidor Global atua como uma curva suavizada de desempenho ótimo. Segundo (Kairouz; McMahan, 2021), isso ocorre porque a agregação paramétrica no servidor mitiga o viés e cancela os ruídos individuais resultantes da baixa amostragem dos clientes locais, produzindo um modelo global detentor de maior capacidade de generalização espacial do que qualquer nó isolado.

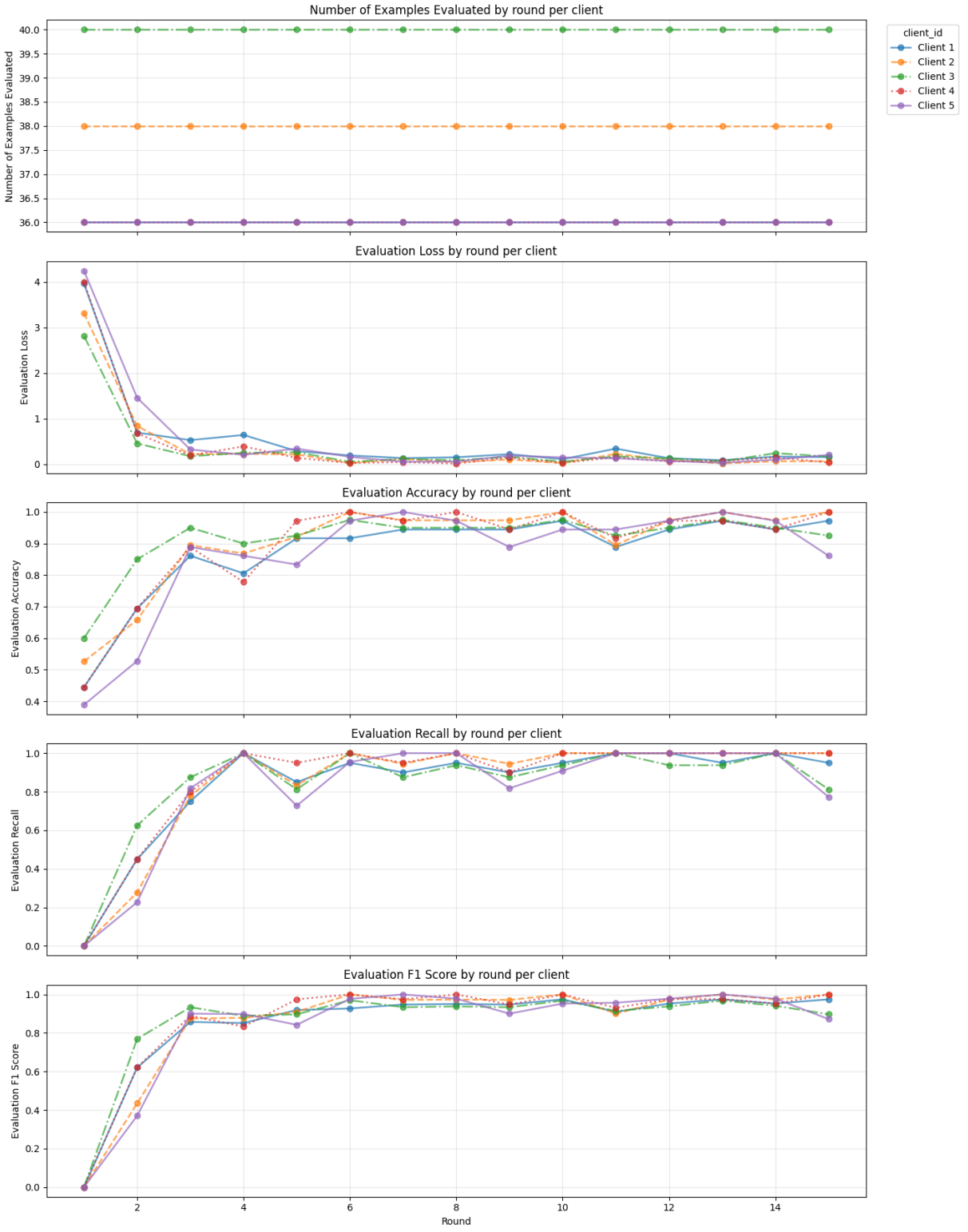

Figura 4.8 – Comparação da métrica de Acurácia e Loss entre o modelo Global (Servidor) e os modelos locais (Clientes). Fonte: elaboração própria.

4.3 Treinamento Federado no Cenário *Non-IID*

Diferentemente do cenário homogêneo, a aplicação real do Aprendizado Federado na agricultura lida com extrema heterogeneidade: uma área monitorada pode estar severamente infestada, enquanto outra não apresenta plantas daninhas. Para emular essa complexidade do mundo real, instituiu-se o cenário Non-IID (*Non-Independent and Non-Identically Distributed*). A distribuição do *dataset* entre os clientes foi governada por uma distribuição de probabilidade de *Dirichlet* com fator de concentração $\alpha = 0.5$. Matematicamente, essa configuração força um alto desbalanceamento estatístico, fazendo com que clientes locais possuam quase que exclusivamente imagens de uma única classe.

A consequência direta desta distribuição assimétrica se mostra visível nas curvas de avaliação individuais dos clientes (Figura 4.9). Em contraste com a relativa proximidade das linhas no cenário IID, os nós locais no ambiente Non-IID apresentam uma dispersão caótica e extrema. Alguns clientes atingem rapidamente altas métricas de forma ilusória, por possuírem majoritariamente apenas uma classe e o modelo local viciar sua predição, enquanto outros nós colapsam.

Segundo (Zhao et al., 2018), esse fenômeno é conhecido como Desvio de Cliente (*Client Drift*). Em linhas gerais, durante as épocas locais, os gradientes de cada cliente são puxados em direções opostas no espaço de otimização, devido aos fortes vieses de seus dados parciais. A utilização do algoritmo *FedAvg* clássico neste contexto por parte do servidor agregador resultaria em uma média de pesos tão discrepantes que anularia o conhecimento global, impedindo a convergência da rede (Li et al., 2020).

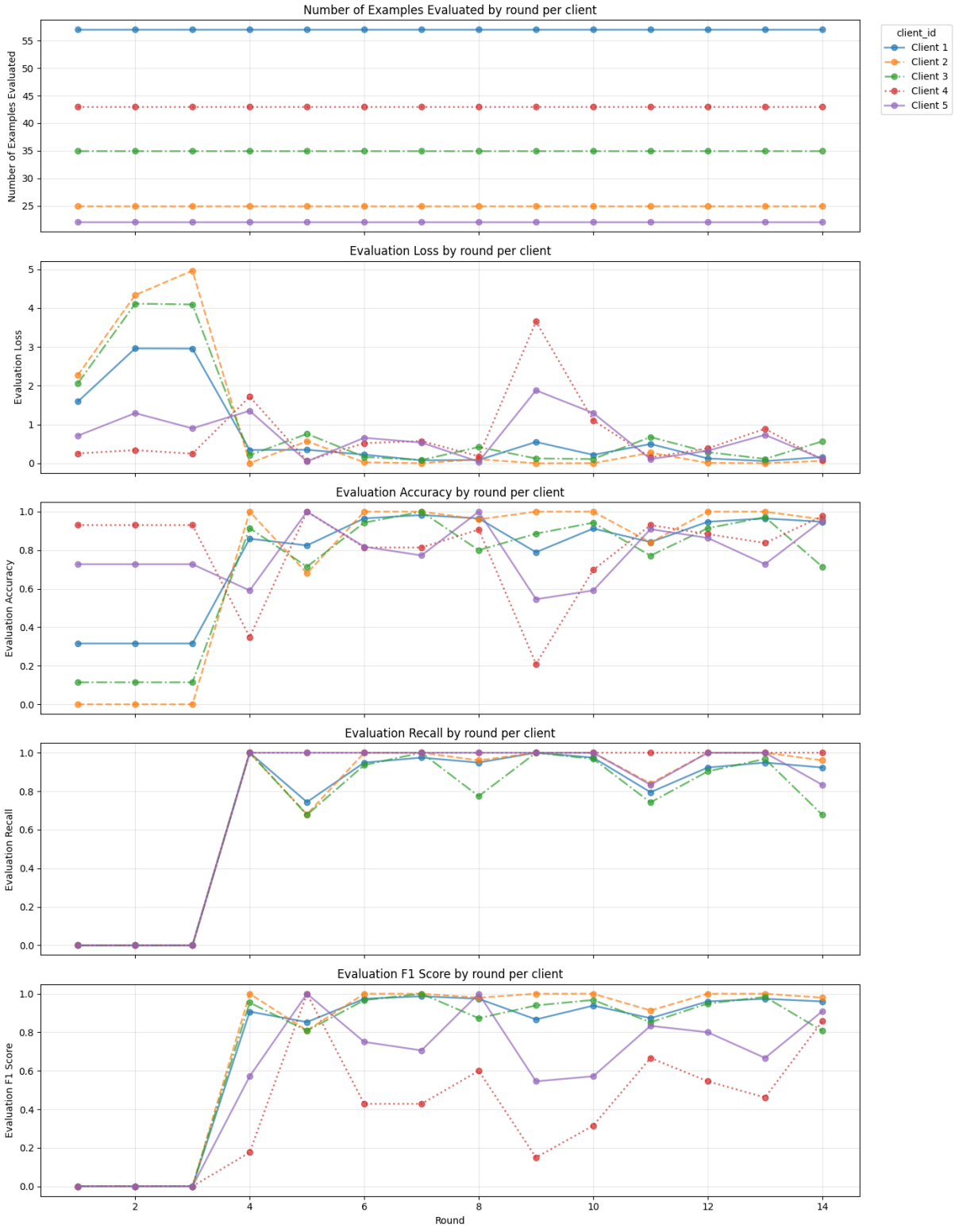

Figura 4.9 – Dispersão acentuada (*Client Drift*) das métricas de Acurácia e Loss entre os modelos locais e o Servidor no cenário Non-IID. Fonte: elaboração própria.

Para contornar o colapso paramétrico inerente aos dados Non-IID, foi aplicada no servidor global a agregação e a atualização local utilizando o algoritmo **FedProx**, configurado com um termo de regularização proximal $\mu = 5.0$. Em estágios preliminares de experimentação, valores

inferiores de μ revelaram-se insuficientes para restringir as atualizações locais a uma região de proximidade segura do modelo global. Consequentemente, magnitudes menores do termo proximal induziram a uma alta volatilidade no processo de otimização, resultando na instabilidade da convergência global e em métricas de desempenho insatisfatórias.

Os resultados globais consolidados (Figura 4.10) demonstram o sucesso da abordagem. Apesar da curva do modelo global ser visivelmente mais ruidosa em comparação ao cenário IID, a convergência foi assegurada. Esse ruído mais acentuado decorre da própria natureza heterogênea da distribuição de *Dirichlet*: mesmo sob penalidade matemática, a extrema distorção das classes faz com que os clientes puxem agressivamente os gradientes para ótimos locais divergentes, gerando flutuações residuais a cada agregação. Ainda assim, a *Loss* global decai de forma controlada e a Acurácia ascende acima dos patamares de aceitabilidade, suportando a complexidade geométrica do aprendizado.

Um evento que ilustra claramente essa dinâmica estocástica ocorreu na rodada de comunicação 9. Neste instante, o modelo global apresentou uma generalização tendenciosa e um declínio temporário. Isso ocorreu pois os clientes locais selecionados para o treinamento daquela iteração apresentavam uma partição de dados altamente desfavorável: receberam uma proporção significativamente maior de dados alocados para validação e, por consequência, operaram com um volume insuficiente de dados para o treinamento local. Essa escassez amostral resultou em uma queda abrupta na performance de aprendizado desses nós específicos. Assim, evidencia-se que, embora o uso da estratégia *FedProx* seja efetivo no cômputo geral, ele apresentou dificuldades pontuais para neutralizar completamente o viés introduzido pela deficiência extrema de dados de treino nesses clientes.

Apesar dessa oscilação, o resultado geral corrobora a tese fundadora da estratégia *FedProx* desenvolvida por (Li et al., 2020). O termo proximal (μ) inserido no otimizador restringe matematicamente a magnitude das atualizações locais, isto é, ele atua como uma restrição âncora, permitindo que o cliente aprenda com seus dados locais enviesados durante as 10 épocas, mas penalizando os pesos caso eles se afastem excessivamente do modelo global da rodada anterior. Essa ancoragem impede o esquecimento catastrófico nos clientes isolados e garante que, ao serem agregados no servidor, os pesos ainda pertençam a um subespaço compatível, viabilizando a construção de um classificador global robusto capaz de generalizar sobre a lavoura inteira.

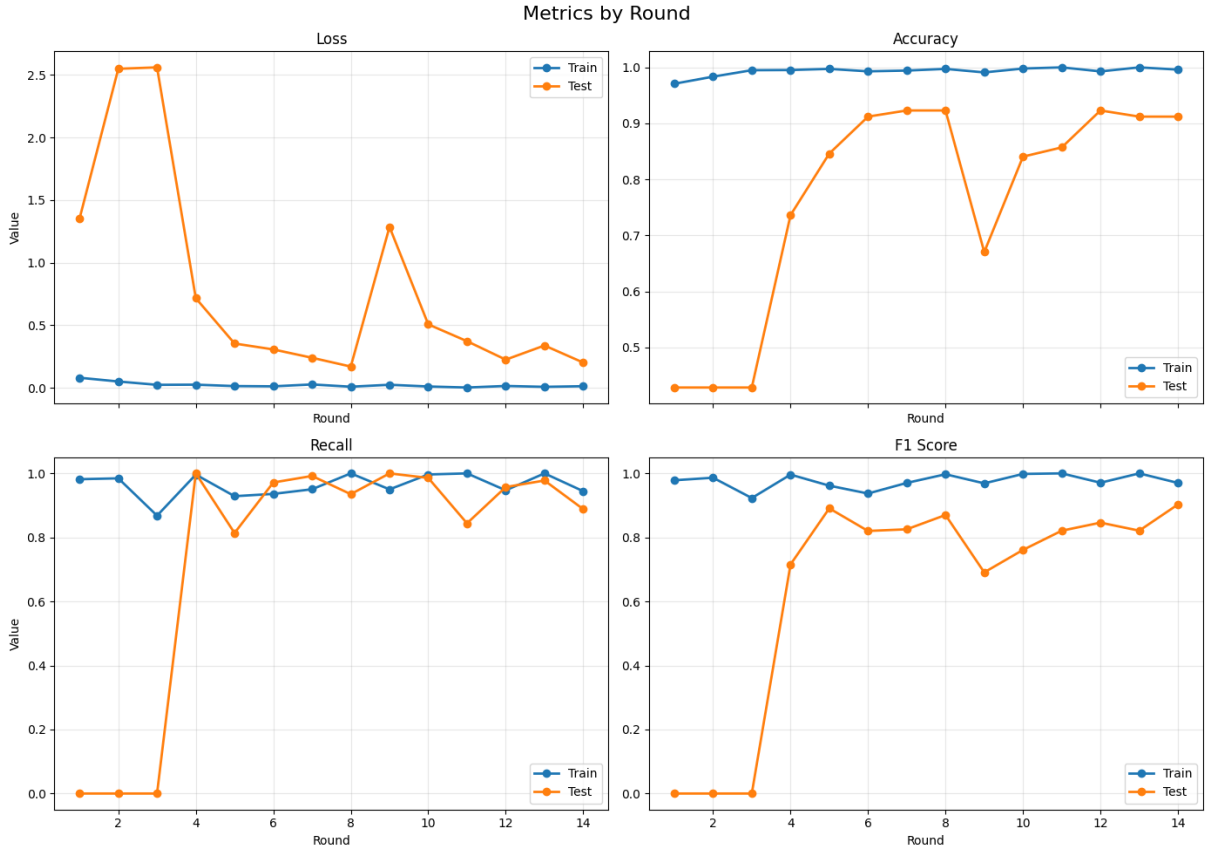

Figura 4.10 – Evolução global das métricas sob a agregação FedProx. Nota-se a convergência mantida apesar das oscilações inerentes à heterogeneidade dos dados. Fonte: elaboração própria.

4.4 Comparação de Resultados

Para sintetizar os resultados experimentais obtidos e facilitar a análise comparativa entre os modelos de aprendizado, a Tabela 4.1 apresenta a consolidação das métricas finais alcançadas pela arquitetura *DarkNet53* em suas respectivas abordagens. Observa-se que, embora o ambiente centralizado estabeleça o limite superior de performance (*baseline*), a transição para o Aprendizado Federado penaliza as métricas de forma marginal e controlada. A robustez mantida nos cenários IID e Non-IID atesta que a orquestração descentralizada, especialmente com a ancoragem do algoritmo *FedProx*, é capaz de entregar confiabilidade agronômica aliada à preservação da privacidade.

Cenário de Treinamento	**Estratégia / Algoritmo**	**Acurácia**	**Recall**	**F1-Score**
Centralizado (*Baseline*)	Treinamento Unificado	0.938	0.927	0.955
Aprendizado Federado (IID)	*FedAvg*	0.951	0.906	0.947
Aprendizado Federado (Non-IID)	*FedProx* ($\mu = 5.0$)	0.912	0.888	0.902

Tabela 4.1 – Comparativo de desempenho da arquitetura *DarkNet53* nos diferentes cenários experimentais.

5 Considerações Finais

5.1 Conclusão

O presente trabalho teve como objetivo investigar e validar a aplicação do Aprendizado Federado (*Federated Learning*) na detecção de plantas daninhas em lavouras de cana-de-açúcar. A premissa central foi solucionar o dilema entre a necessidade de modelos de alta performance para a agricultura de precisão e a urgência em preservar a privacidade dos dados corporativos e operacionais dos produtores rurais.

A metodologia adotada partiu do estabelecimento de um *baseline* centralizado. Observou-se que a restrição quantitativa da base de dados e, consequentemente, a necessidade de treinamentos com lotes reduzidos (*batch sizes*) impactaram negativamente redes pré-treinadas baseadas em *Batch Normalization*, como *EfficientNetB0* e *ResNet50*. Em contrapartida, a arquitetura *DarkNet53*, inicializada do zero e baseada na ativação *LeakyReLU*, demonstrou notável resiliência, isolando o ruído estocástico e atingindo estabilidade, consistência (*Recall* e *F1-Score* elevados) e capacidade de generalização excepcionais.

Com a validação da *DarkNet53*, os experimentos no ecossistema federado comprovaram a viabilidade da proposta, tendo em vista a convergência em cenários IID e *Non-IID*. A partir dos resultados obtidos, é possível responder diretamente às perguntas de pesquisa que nortearam este estudo:

- **(I) No contexto de classificação de imagens de plantas daninhas, qual é o impacto no desempenho preditivo ao se migrar de um ambiente centralizado para o Aprendizado Federado?** O impacto preditivo demonstrou-se aceitável e viável. Enquanto o modelo centralizado alcançou métricas beirando a totalidade de acertos devido ao acesso irrestrito aos dados, a migração para o ambiente federado no cenário IID manteve um desempenho robusto, com Acurácia e *Recall* superiores a 90%. O treinamento federado provou ser capaz de construir um modelo global altamente competitivo, revelando que a troca de centralização por privacidade resulta em uma penalidade marginal no desempenho preditivo, justificável pelos benefícios de segurança.
- **(II) Como cenários de distribuição de dados heterogêneos *(Non-IID)* afetam a convergência e a estabilidade do modelo treinado?** A variabilidade espacial característica da agricultura (Cenário Non-IID) afeta criticamente o treinamento ao induzir o *Client Drift*. Os experimentos comprovaram que nós locais tendem ao *overfitting* e ao viés devido ao desbalanceamento de suas partições de dados, gerando extrema instabilidade nas validações locais. Contudo, ao substituir a agregação tradicional pelo algoritmo *FedProx*, a

introdução do termo de regularização proximal ancorou as atualizações locais. Isso mitigou o esquecimento catastrófico e assegurou que, apesar do forte ruído inicial, o Servidor Global mantivesse uma convergência estável e resiliente, absorvendo a heterogeneidade e estabilizando a rede ao longo das rodadas.

- **(III) É possível que o modelo federado alcance métricas de desempenho competitivas compartilhando apenas gradientes, preservando a confidencialidade das propriedades?** Sim, de forma categórica. O sucesso convergente das simulações, tanto via *FedAvg* quanto via *FedProx*, atesta que é possível orquestrar o aprendizado profundo transferindo estritamente matrizes de pesos pela rede. Nenhuma imagem real de cana ou planta daninha precisou deixar seu nó de origem. Isso garante a privacidade das propriedades do produtor, viabilizando consórcios cooperativos entre fazendas para a criação de inteligência artificial de ponta para a agricultura de precisão sem ferir o sigilo industrial.

Em síntese, os resultados obtidos nesta pesquisa comprovam a validação de modelos computacionais, apresentando uma contribuição prática, viável e segura para a agricultura de precisão. A orquestração bem-sucedida do Aprendizado Federado prova que a inovação tecnológica no agronegócio não exige a abdicação da privacidade do produtor rural.

Ao viabilizar a detecção de plantas daninhas de forma estritamente descentralizada, este trabalho oferece uma fundação arquitetural para o desenvolvimento de maquinários e drones agrícolas colaborativos. Abre-se, assim, um horizonte promissor no qual cooperativas e propriedades rurais poderão treinar inteligências artificiais conjuntas, otimizando a aplicação de defensivos, promovendo a sustentabilidade ambiental e reduzindo custos operacionais, sob a garantia da proteção de seus dados e estratégias de manejo.

5.2 Trabalhos Futuros

Embora os resultados atestem a eficácia da abordagem, o escopo quantitativo do *dataset*, limitado a 922 imagens, figura como a principal restrição deste trabalho. Com a expansão do volume de dados em trabalhos futuros, as flutuações estatísticas inerentes ao treinamento com *mini-batches* poderiam ser mitigadas.

Uma base de imagens ampliada permitiria a estabilização natural das avaliações locais no contexto federado, reduzindo a variância dos testes em cada cliente, e viabilizaria otimizações mais complexas no cenário centralizado, como o aumento dos lotes e a reavaliação de redes como a *EfficientNet* sem o choque das normalizações pré-treinadas. Além disso, a robustez de um *dataset* maior abriria margem para evoluir esta pesquisa da classificação binária de imagens estáticas para tarefas de detecção e segmentação de objetos em tempo real em maquinários agrícolas.

Referências

AGROPECUáRIA, E. E. B. de P. *Plantas Daninhas: Sobre o tema*. 2024. Disponível em: <https://www.embrapa.br/tema-plantas-daninhas/sobre-o-tema>.

ALMADA, M. Modelos de neurônios em redes neurais artificiais. 11 2019.

ARAFEH, M.; HAMMOUD, A.; OTROK, H.; MOURAD, A.; TALHI, C.; DZIONG, Z. Independent and identically distributed (iid) data assessment in federated learning. In: IEEE. *GLOBECOM 2022-2022 IEEE Global Communications Conference*. [S.l.], 2022. p. 293–298.

ARANGI, D.; PADHY, N. Federated edge learning for distributed weed detection in precision agriculture using multimodal sensor fusion. *Engineering Proceedings*, MDPI, v. 118, n. 1, p. 33, 2025.

AZANIA, C. A. M. et al. *Fundamentos para Aplicação de Herbicidas em Cana-de-Açúcar*. [S.l.], 2021. Disponível em: <https://www.iac.sp.gov.br/media/publicacoes/iacbt229.pdf>.

BEUTEL, D. J.; TOPAL, T.; MATHUR, A.; QIU, X.; FERNANDEZ-MARQUES, J.; GAO, Y.; SANI, L.; LI, K. H.; PARCOLLET, T.; GUSMÃO, P. P. B. de et al. Flower: A friendly federated learning research and practice framework. *arXiv preprint arXiv:2007.14390*, 2020. Disponível em: <https://arxiv.org/abs/2007.14390>.

CHEEMA, M. J. M.; IQBAL, T.; DACCACHE, A.; HUSSAIN, S.; AWAIS, M. Precision agriculture technologies: present adoption and future strategies. In: *Precision agriculture*. [S.l.]: Elsevier, 2023. p. 231–250.

CONSTANTIN, J. *Métodos de Controle*. [S.l.], 2011. Disponível em: <https://www.esalq.usp.br/LPV/sites/default/files/10\%20-\%20Leitura\%20metodos\%20de\%20controle\%201.pdf>.

COX, G.; HARRIS, H.; COX, D. Application of precision agriculture to sugar cane. In: WILEY ONLINE LIBRARY. *Proceedings of the Fourth International Conference on Precision Agriculture*. [S.l.], 1999. p. 753–765.

CUNHA, G. N.; PASQUALETTO, A. Impactos socioeconómicos e ambientais, do plantio à colheita, da cana-de-açúcar na mesorregião norte de goiás. *COLÓQUIO-Revista do Desenvolvimento Regional*, v. 19, n. 4, out./dez., p. 234–257, 2022.

DOMAH, D. K. *Sugarcane and Weed Dataset*. 2025. Kaggle Dataset. Disponível em: <https://www.kaggle.com/datasets/dhirendrakumardomah/sugarcane\discretionary{-}{}{}and\discretionary{-}{}{}weed>.

DUTIA, S. Agtech: Challenges and opportunities for sustainable growth. *Available at SSRN 2431316*, 2014.

Elevagro. *Matocompetição na cultura da Cana-de-açúcar por Ipomoea sp.* 2013. Disponível em: <https://elevagro.com/matocompeticao-na-cultura-da-cana-de-acucar-por-ipomoea-sp/>.

HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 2961–2969.

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*. [S.l.: s.n.], 2016. p. 770–778.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Identity mappings in deep residual networks. In: SPRINGER. *European Conference on Computer Vision (ECCV)*. [S.l.], 2016. p. 630–645.

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 7132–7141.

IBGE. *Produção Agropecuária: Cana-de-açúcar*. 2024. Instituto Brasileiro de Geografia e Estatística (. Disponível em: <https://www.ibge.gov.br/explica/producao-agropecuaria/cana-de-acucar/br>.

Instituto Brasileiro de Geografia e Estatística. *Produção Agrícola Municipal: Culturas temporárias e permanentes - Séries Históricas*. 2024. Disponível em: <https://www.ibge.gov.br/estatisticas/economicas/agricultura-e-pecuaria/9117-producao-agricola-municipal-culturas-temporarias-e-permanentes.html?=&t=series-historicas>. Acesso em: 29 jun. 2026.

IYER, V. N. A review on different techniques used to combat the non-iid and heterogeneous nature of data in fl. *arXiv preprint arXiv:2401.00809*, 2024.

KAIROUZ, P.; MCMAHAN, H. B. Advances and open problems in federated learning. *Foundations and trends in machine learning*, Emerald Publishing Limited, v. 14, n. 1-2, p. 1–210, 2021.

KUVA, M. A.; PITELLI, R. A.; CHRISTOFFOLETI, P. J.; ALVES, P. L. C. A. Períodos de interferência das plantas daninhas na cultura da cana-de-açúcar: Iii - capim-braquiária e capim-colonião (panicum maximum). *Planta Daninha*, SciELO Brasil, v. 21, n. 1, p. 37–44, 2003. Disponível em: <https://doi.org/10.1590/S0100-83582003000100005>.

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

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 2002.

LI, T.; SAHU, A. K.; ZAHEER, M.; SANJABI, M.; TALWALKAR, A.; SMITH, V. Federated optimization in heterogeneous networks. In: *Proceedings of Machine Learning and Systems (MLSys)*. [S.l.: s.n.], 2020. v. 2, p. 429–450.

LU, L.; SHIN, Y.; SU, Y.; KARNIADAKIS, G. Dying relu and initialization: Theory and numerical examples. cicp 28 (5): 1671–1706. *arXiv preprint arXiv:1903.06733*, 2020.

MAMMEN, P. M. Federated learning: Opportunities and challenges. *arXiv preprint arXiv:2101.05428*, 2021.

MARCONDES, L. d. S. *FedBN : Federated Learning on Non-IID Features via Local Batch Normalization*. 2024. Disponível em: <https://ic.unicamp.br/~allanms/mo809/trab/trabalho11/>.

MCMAHAN, B.; MOORE, E.; RAMAGE, D.; HAMPSON, S.; ARCAS, B. A. y. Communication-efficient learning of deep networks from decentralized data. In: PMLR. *Artificial intelligence and statistics*. [S.l.], 2017. p. 1273–1282.

MILANEZ, A. Y.; GUIMARÃES, D. D.; SILVA, M. M. d.; NAKANO, V. T. P. O protagonismo do brasil no mercado global de açúcar: evolução recente e perspectivas. *Banco Nacional de Desenvolvimento Econômico e Social*, 2024.

MIRANDA, P. A. B. P. Características forrageiras de variedades de cana-de-açúcar (saccharum spp.) desenvolvidas no estado de alagoas. *Universidade Federal de Alagoas*, 2015.

MITCHELL, T. M. *Machine Learning*. [S.l.]: New York: McGraw-Hill, 1997.

MODI, R. U.; KANCHETI, M.; SUBEESH, A.; RAJ, C.; SINGH, A. K.; CHANDEL, N. S.; DHIMATE, A. S.; SINGH, M. K.; SINGH, S. An automated weed identification framework for sugarcane crop: A deep learning approach. *Crop Protection*, v. 173, p. 106360, 2023. ISSN 0261-2194. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0261219423001837>.

MUANGKASEM, A.; THAINIMIT, S.; KEINPRASIT, R.; ISSHIKI, T.; TANGWONGKIT, R. Weed detection over between-row of sugarcane fields using machine vision with shadow robustness technique for variable rate herbicide applicator. *Energy Research Journal*, v. 1, n. 2, p. 141–145, 2010.

MUMUNI, A.; MUMUNI, F. Data augmentation: A comprehensive survey of modern approaches. *Array*, Elsevier, v. 16, p. 100258, 2022.

NASCIMENTO, D. Prescilio do; BUENO, M. P.; JÚNIOR, J. G. C.; GOMES, W. S.; FILHO, J. A. F. Panorama da cultura da cana-de-açúcar utilizando prospecção tecnológica. *GeSec: Revista de Gestao e Secretariado*, v. 15, n. 7, 2024.

PIERCE, F. J.; NOWAK, P. Aspects of precision agriculture. *Advances in agronomy*, Elsevier, v. 67, p. 1–85, 1999.

QI, P.; CHIARO, D.; GUZZO, A.; IANNI, M.; FORTINO, G.; PICCIALLI, F. Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems*, Elsevier, v. 150, p. 272–293, 2024.

REDMON, J. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

ROSA, G.; LUZ, J. da. Simulação de moagem mista por rede neural artificial. *Rem: Revista Escola de Minas*, v. 65, p. 247–256, 06 2012.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

Senior Sistemas. *O plantio de cana-de-açúcar e a sua importância para o agronegócio*. 2023. Disponível em: <https://www.senior.com.br/blog/o-plantio-de-cana-de-acucar-e-a-sua-importancia-para-o-agronegocio>.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, SpringerOpen, v. 6, n. 1, p. 1–48, 2019.

SINGH, S.; KRISHNAN, S. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 11237–11246.

SIPIO, R. D. *A Quick Guide to Gradient Descent and its Variants*. 2019. <https://pub.towardsai.net/a-quick-guide-to-gradient-descent-and-its-variants-afa181d5b97b>. Publicado em Towards AI.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. [S.l.], 2019. p. 6105–6114.

V, S. M.; MULERIKKAL, J.; PB, R.; THARAKAN, P. Fuzzy concepts and machine learning algorithms for car park occupancy and route prediction. In: ____. [S.l.: s.n.], 2020. p. 35–51. ISBN 978-981-15-3324-2.

WISEMAN, L.; SANDERSON, J.; ZHANG, A.; JAKKU, E. Farmers and their data: An examination of farmers' reluctance to share their data through the lens of the laws impacting smart farming. *NJAS-Wageningen Journal of Life Sciences*, Elsevier, v. 90, p. 100301, 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.

WU, Y.; HE, K. Group normalization. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 3–19.

WU, Y.-c.; FENG, J.-w. Development and application of artificial neural network. *Wireless Personal Communications*, Springer, v. 102, n. 2, p. 1645–1656, 2018.

XU, B.; WANG, N.; CHEN, T.; LI, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

YAKHYO, J. *darknet-pytorch: Implementation of DarkNet models in PyTorch*. [S.l.]: GitHub, 2024. <https://github.com/yakhyo/darknet-pytorch>.

YANO, I. H.; ALVES, J. R.; SANTIAGO, W. E.; MEDEROS, B. J. T. Identification of weeds in sugarcane fields through images taken by uav and random forest classifier. *IFAC-PapersOnLine*, Elsevier, v. 49, n. 16, p. 415–420, 2016.

YING, X. An overview of overfitting and its solutions. In: IOP PUBLISHING. *Journal of physics: Conference series*. [S.l.], 2019. v. 1168, n. 2, p. 022022.

ZHAO, Y.; LI, M.; LAI, L.; SUDA, N.; CIVIN, D.; CHANDRA, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

ZHENG, L.; LONG, L.; ZHU, C.; JIA, M.; CHEN, P.; TIE, J. A lightweight cotton field weed detection model enhanced with efficientnet and attention mechanisms. *Agronomy*, MDPI, v. 14, n. 11, p. 2649, 2024.

Declaração de Uso de Inteligência Artificial Generativa

Durante a preparação deste documento, eu, ELISA ALVES VELOSO, estudante de graduação em Ciência da Computação da Universidade Federal de Ouro Preto, declaro o uso de IAGen Gemini versão 3.1 Pro para tradução, revisão textual, geração de códigos LaTeX e BibTeX e geração da figura 2.1.

Após o uso desta ferramenta, eu revisei e editei o conteúdo em conformidade com os princípios éticos para uso de IAGen (Resolução CONPEP nº 144) e com os acordos estabelecidos com o orientador desta pesquisa. Dessa forma, assumo total responsabilidade pelo conteúdo da publicação.