

Identificação de Doenças na Soja Utilizando Redes Neurais Artificiais em Dispositivos Móveis *

Fernando A. Port¹, Roger Luis Hoff Lavarda¹

¹Ciência da Computação – Instituto Federal do Rio Grande do Sul (IFRS)
Campus Ibirubá

Abstract. *The soybean culture, one of the most crucial components of agribusiness and the Brazilian economy, faces challenges stemming from diseases that can directly impact its productivity. However, the application of digital technologies by farmers and field professionals offers a way to mitigate the effects caused by pests affecting the plants. This study presents a mobile application for the Android operating system capable of assisting producers in identifying Bacterial Blight, Rust, and Downy Mildew diseases, using computer vision techniques and an Artificial Neural Network model for their classification. At the end of the research, the application demonstrated its ability to identify and classify soybean leaves affected by these diseases through images, achieving an accuracy of approximately 99% in tests conducted on a smartphone.*

Resumo. *A cultura da soja, uma das mais importantes do agronegócio e para a economia brasileira, enfrenta desafios decorrentes de doenças que podem ter um impacto direto em sua produtividade. No entanto, a aplicação de tecnologias digitais por agricultores e profissionais do campo oferece uma forma de mitigar os efeitos causados pelas pragas que afetam as plantas. O presente trabalho, apresenta um aplicativo para o sistema operacional Android capaz de auxiliar o produtor na identificação das doenças Crestamento Bacteriano, Ferrugem Asiática e Mildio, utilizando técnicas de visão computacional e um modelo de Rede Neural Artificial para realizar a classificação das mesmas. Ao término da pesquisa, o aplicativo demonstrou sua capacidade em identificar e classificar, por meio de imagens, as folhas de soja afetadas por estas doenças, alcançando uma acurácia de aproximadamente 99% nos testes conduzidos em um Smartphone.*

1. Introdução

No atual contexto agrícola, o Brasil se destaca como o maior produtor mundial de soja, atingindo uma produção de 154,8 milhões de toneladas em uma área plantada de 43,8 milhões de hectares na safra de 2022/23, conforme dados da CONAB (2023). Esse notável crescimento é impulsionado pelo avanço da digitalização e da conectividade no campo, fatores fundamentais, segundo Filho et al. (2011), para determinar o desempenho econômico-financeiro das propriedades rurais.

Entretanto, a produção de soja no Brasil enfrenta desafios significativos, como a competição com plantas daninhas e a luta contra pragas e doenças que impactam negativamente o desempenho produtivo (CONTE; POSSAMAI; FILHO, 2020). Estima-se que

*Trabalho de Conclusão de Curso (TCC) do curso Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Ibirubá, 2023.

doenças na planta da soja possam causar perdas substanciais, variando de 15% a 20% na produção de grãos, podendo, em situações específicas, resultar na perda total da lavoura (OLIVEIRA et al., 2021).

A falta de capacitação adequada da mão de obra nas lavouras, conforme observado por Cardoso e Louzada (2012), muitas vezes resulta em práticas inadequadas de manejo, agravando a incidência de doenças e pragas. Essa falta de preparo evidencia a necessidade de uma identificação correta dessas doenças para a implementação eficaz de programas de controle, segundo Henning et al. (2014). Diante desse cenário, o presente projeto surge com o propósito de abordar a seguinte questão: como detectar de maneira prática e eficiente algumas das principais doenças que afetam a soja?

Uma das formas de realizar o monitoramento de algumas doenças é por meio do uso de IA (Inteligência Artificial), utilizando técnicas de visão computacional aliadas ao aprendizado de máquina com RNA (Redes Neurais Artificiais). Esses têm se mostrado métodos eficientes para a classificação de doenças em diversas culturas no campo, conforme evidenciado no estudo de Ferreira (2017).

Nesse contexto, o objetivo dessa pesquisa foi desenvolver um aplicativo móvel para Android, utilizando a RNA MobileNetV3, para auxiliar os produtores na identificação das doenças Crestamento Bacteriano, Ferrugem e Míldio, justificando-se pela sua potencial contribuição em aprimorar o manejo de doenças na cultura da soja, podendo mitigar a dependência de especialistas e conhecimento prévio na área, proporcionando um diagnóstico ágil e prático.

O presente trabalho está organizado da seguinte maneira: na Seção 2 estão apresentados os conceitos básicos para a compreensão deste trabalho, com a Revisão Bibliográfica. Na sequência, na Seção 3, estão os Trabalhos Correlatos. Na Seção 4, são apresentados os Materiais e Métodos, divididos em 5 subseções. Na Seção 5, são apresentados os resultados produzidos com essa investigação. Por fim, na Seção 6, temos a Conclusão.

2. Revisão Bibliográfica

Nesta seção, são apresentados os conceitos que fundamentaram o desenvolvimento do presente projeto.

2.1. Cultura da Soja

A soja, segundo Bonato e Bonato (1987), é uma leguminosa de origem não definitivamente conhecida, mas que todos indicam, no entanto, ser originária da China, cultivada a milhares de anos. Atualmente, apresenta-se como uma das principais *commodities* comercializadas no mundo, conforme Campeão, Sanches e Maciel (2020).

A utilização da soja é muito conhecida pela extração do óleo vegetal, que pode ser usado como óleo de salada, de cozinha e de fritura. O óleo também apresenta aplicações industriais como tinta de caneta, bio-diesel, tintas de pintura em geral, xampus, sabões e detergentes. Além de seu subproduto, o farelo, que é usado basicamente como suplemento para a criação de gado, suínos, alimento de peixe na aquicultura, na produção de ração de animais domésticos e como substitutos do leite para bezerros (MISSÃ et al., 2006).

As perdas anuais causadas por doenças na cultura da soja são estimadas em cerca de 15% a 20%, por afetarem diretamente o índice de área foliar das plantas, principal componente da produção, devido à realização da fotossíntese (NASCIMENTO et al., 2018).

Segundo Corrêa-Ferreira (2012), se os programas de controle de doenças não forem conduzidos de maneira adequada, existe a possibilidade de decisões incorretas em relação ao uso de agrotóxicos. Isso pode resultar em aplicações desnecessárias ou na dispensa de aplicações essenciais.

2.2. Visão Computacional

Visão computacional, segundo Milano e Honorato (2010), é a responsável pela forma como um computador enxerga o meio à sua volta, através de imagens capturadas por câmeras de vídeo, sensores, scanners, entre outros dispositivos. Dessa maneira, a visão computacional oferece ao computador uma ampla gama de informações a partir de imagens e vídeos, permitindo que ele execute tarefas inteligentes e se aproxime da inteligência humana, simulando-a.

As aplicações que utilizam visão computacional são, em sua maioria, provindas de outras áreas de pesquisa, dessa forma, não existe um modelo padrão para implementação destas funcionalidades. Porém, basicamente todos os sistemas de visão computacional envolvem reconhecimento de objetos em imagens, transformando-os em informações que serão processadas e utilizadas por algum sistema especialista. Funcionalidades comuns na maioria dos sistemas de visão computacional: aquisição da imagem, pré-processamento, extração de características, detecção e segmentação, processamento de alto nível (MILANO; HONORATO, 2010).

No contexto da agricultura, a visão computacional pode ser empregada na detecção de doenças e pragas, na estimação de safra e na avaliação não invasiva de atributos como qualidade, aparência e volume, além de ser componente essencial em sistemas robóticos agrícolas (SANTOS et al., 2020).

2.3. Redes Neurais Artificiais

As Redes Neurais Artificiais são técnicas computacionais de Aprendizado de Máquina que apresentam um modelo matemático inspirado na estrutura neural biológica, tendo surgimento com o trabalho de Fitch (1944). O primeiro modelo de Rede Neural teve apenas um neurônio, chamado de Perceptron no trabalho de Rosenblatt (1958).

Uma de suas principais características é a capacidade de aprender por meio de exemplos e de generalizar a informação aprendida, dessa forma, as RNAs são capazes de tomar decisões de acordo com o próprio aprendizado. Elas são compostas por uma determinada quantidade de entradas e unidades de processamento, as quais são ligadas através de pesos sinápticos (FLECK et al., 2016). As RNAs são comumente utilizadas na resolução de problemas complexos, onde o comportamento das variáveis não é definitivamente conhecido.

Assim como para os humanos, o aprendizado de máquina ocorre quando as Redes Neurais são expostas a dados, aprendendo por meio de exemplos e definindo um padrão. Se tratando de imagens, os principais fatores a serem analisados pela Rede Neural são cores, formas, localização e contraste (MARTINS; SCHOR, 2022).

Redes Neurais Artificiais são estruturas computacionais projetadas com a finalidade de modelar a maneira pela qual o cérebro desempenha alguma tarefa de seu interesse. As RNAs consistem da interconexão de uma grande quantidade de unidades de processamento, chamadas de neurônios. Elas obtém conhecimento através de um processo de aprendizado, e armazenam o conhecimento obtido nos pesos sinápticos e parâmetros (CASTRO; CASTRO, 2001).

Algumas características relevantes das RNAs são:

- Possibilidade de considerar a não-linearidade de fenômenos físicos responsáveis pela geração dos dados de entrada
- Capacidade de aprendizado através de um processo de treinamento
- Adaptatividade
- Capacidade de generalização
- Tolerância a falhas

Uma das métricas para avaliar o funcionamento de uma rede neural é a função de perda (*loss*), que representa a medida de quão distantes as previsões da rede estão em relação aos valores reais (rótulos) dos dados de treinamento. Essa métrica é fundamental durante o processo de treinamento, pois o objetivo é minimizar o *loss* ajustando os pesos da rede, de modo a melhorar a precisão das previsões. Em termos simples, quanto menor o *loss*, mais eficiente é a capacidade da rede neural em aprender e realizar a tarefa específica para a qual está sendo treinada (ZHAO et al., 2016). Por outro lado, a acurácia em uma rede neural é uma medida crucial que avalia a precisão geral do modelo ao classificar dados de teste. Ela representa a proporção de previsões corretas em relação ao total. Ao interpretar a acurácia, é importante considerar o contexto específico do problema e, se necessário, complementar a análise com outras métricas para uma avaliação mais completa (SILVA et al., 2023).

2.3.1. Rede Neural Convolutacional (CNN)

As CNNs, são tipos de Redes Neurais Artificiais Profundas. São amplamente utilizadas para resolver problemas do mundo real, como por exemplo, no reconhecimento de padrões em imagens. No entanto, tarefas de classificação em dispositivos móveis não exigem apenas alta precisão, mas também, baixos custos de memória e alta eficiência computacional (QIAN; NING; HU, 2021).

Segundo Gu et al. (2018), existem inúmeras variantes de arquiteturas CNN. No entanto, seus componentes básicos são muito semelhantes, que são: a camada de convolução, camada de *pooling* e camada totalmente conectada.

As camada de convolução é responsável por extrair os dados de entrada e os repassar para a próxima camada. A camada de *pooling* é usada para reduzir as dimensões dos dados aplicando camadas de agrupamento, essa camada recebe cada saída da camada convolutacional. A camada totalmente conectada é que realiza a tarefa de classificação, as pontuações de probabilidade são calculadas por uma função de ativação (RODRIGUES, 2018).

A MobileNetV3 é uma CNN criada no ano de 2019, aprimorando as suas antecessoras MobileNetV1 e MobileNetV2. As MobileNets são arquiteturas que foram cri-

adas para CPUs de dispositivos móveis, visando manter um equilíbrio entre precisão e eficiência na tarefa de classificar imagens (QIAN; NING; HU, 2021).

Na versão 3, a MobileNet foi dividida em 2 modelos, a MobileNetV3-Large e a MobileNetV3-Small, visando atender a diferentes dispositivos, a versão Small possui menos camadas do que a versão Large. Também foram redesenhadas algumas camadas, com a finalidade de melhorar o seu desempenho, além de ser introduzida uma nova não-linearidade na rede, o que torna a MobileNetV3 capaz de realizar os procedimentos com uma menor latência, porém mantendo a precisão (HOWARD et al., 2019).

2.4. Tensor Flow

TensorFlow é uma biblioteca de código aberto, desenvolvida pelo Google, para facilitar a implementação de aprendizado de máquina e redes neurais. A biblioteca permite a criação de modelos complexos e oferece suporte a várias tarefas, como classificação, processamento de linguagem natural e visão computacional. De acordo com Abadi et al. (2016), o TensorFlow foi projetado para facilitar a pesquisa e a implementação de algoritmos de aprendizado de máquina avançados e para oferecer suporte a uma ampla gama de aplicativos em produção.

O TensorFlow Lite, por outro lado, é uma versão otimizada do TensorFlow projetada para dispositivos móveis, sistemas embarcados e outros dispositivos com recursos computacionais limitados. Segundo Lite (2022), o TensorFlow Lite permite que modelos de aprendizado de máquina sejam executados em tempo real, com baixa latência e usando menos recursos computacionais. Ele fornece uma estrutura eficiente para implantar modelos em dispositivos móveis, permitindo que aplicativos executem inferência localmente, sem a necessidade de conexão com a nuvem.

2.5. Google Colaboratory

O Google Colaboratory, ou Colab, é uma plataforma de código aberto lançada pelo Google em 2017. Projetado para facilitar a execução de código Python na nuvem, o Colab oferece acesso gratuito a GPUs e TPUs, proporcionando um ambiente colaborativo para o desenvolvimento e compartilhamento de notebooks Jupyter no Google Drive (COLABORATORY, 2023).

Diversas bibliotecas populares do Python, como TensorFlow, PyTorch e OpenCV, são pré-instaladas, tornando-o compatível com uma ampla gama de projetos de desenvolvimento.

O Colab é frequentemente empregado em aprendizado de máquina devido à sua interface intuitiva e capacidades de computação. Sua utilidade é reforçada pelo suporte integrado a GPUs e TPUs, agilizando significativamente o treinamento de modelos de aprendizado de máquina. Essas características tornam o Colab uma escolha popular para estudantes e profissionais que buscam uma plataforma acessível e eficiente para suas atividades computacionais (BISONG; BISONG, 2019).

3. Trabalhos Correlatos

Nesta seção, serão apresentados dois trabalhos que fazem o uso de Redes Neurais Convolucionais para solução de problemas envolvendo detecção de doenças na folha da soja, e um trabalho que realiza um comparativo entre algumas RNAs.

3.1. Detecção de Doenças em Folhas de Soja Utilizando Redes Neurais Convolucionais

O trabalho de Silva (2021), intitulado "Detecção de Doenças em Folhas de Soja Utilizando Redes Neurais Convolucionais", escrito como Trabalho de Conclusão de Curso de Engenharia da Computação na Universidade Tecnológica Federal do Paraná, tem por objetivo realizar a detecção de doenças na folha da soja utilizando Redes Neurais Convolucionais. Míldio, Ferrugem, Oídio, Folha Carijó e Crestamento Bacteriano foram as doenças escolhidas para serem detectadas, além de folhas saudáveis.

Foi utilizada uma base de dados da EMBRAPA, contendo um total de 258 imagens divididas em 6 classes. Inicialmente a base de dados foi considerada pequena para se alcançar bons resultados no treinamento da Rede Neural, então, optou-se por utilizar mais de uma textura da mesma folha, aumentando a quantidade de dados significativamente. Foi realizado a separação da região de interesse do fundo da imagem e as mesmas foram separadas em 3 conjuntos: treino, validação e teste.

Após ter a base de dados organizada, foi realizado o treinamento da RNA. A Rede Neural escolhida para a classificação das texturas presentes na folha da soja foi a VGG-16. Para a validação final do resultado foi usado o modelo de aprendizagem supervisionado *Support Vector Machine*. A Rede Neural foi treinada na plataforma de desenvolvimento de código Google Colab, implementada em Python, utilizando a Biblioteca Keras no TensorFlow. De acordo com o gráfico do treinamento, a Rede Neural obteve acurácia superior a 90%.

A partir da localização da região de interesse na folha da soja, foi efetuado um algoritmo para realizar a extração das texturas a serem classificadas pela Rede Neural. A quantidade de texturas retiradas de cada folha é proporcional ao tamanho da mesma.

Com a Rede Neural treinada e as amostras extraídas, foi realizada a classificação das texturas e então gerado um vetor para cada folha da base de testes, desta forma, todo o processo foi supervisionado pelo *Support Vector Machine*.

Os resultados obtidos foram considerados satisfatórios para solução do problema, onde obteve-se precisão de 100% para Folha Carijó, 92% para Crestamento Bacteriano, 98% para Ferrugem, 94% para Míldio e 100% para Oídio.

3.2. Detectando a Ferrugem Asiática na Folha da Soja Utilizando Redes Neurais Convolucionais

O trabalho de Funck (2019), intitulado "Detectando a Ferrugem Asiática na Folha da Soja Utilizando Redes Neurais Convolucionais", escrito como Trabalho de Conclusão de Curso de Ciência da Computação na Universidade Tecnológica Federal do Paraná, apresenta uma solução de um aplicativo móvel para o sistema operacional Android capaz de identificar a ferrugem asiática na folha da soja, através de Redes Neurais Convolucionais, utilizando a câmera de um smartphone.

Para a realização desse trabalho, foi utilizado dois grupos de imagens, o primeiro grupo, com imagens da Invasive, Ipmimages e outras coletadas em campo, continha 770 imagens de folhas contaminadas com a ferrugem asiática em diferentes níveis de severidade. O segundo grupo abrangia 2000 imagens de folhas de soja que foram retiradas da

Plant Village, tanto de folhas saudáveis quando folhas contaminadas com outras doenças da cultura da soja, exceto com a ferrugem asiática.

Os dois grupos de imagens foram submetidos a etapas de pré-processamento antes de realizar o treinamento da Rede Neural Artificial, deixando-as com o tamanho de 224 pixels de largura por 224 pixels de altura, visando deixar a classificação das imagens mais rápida, pois o trabalho foi desenvolvido para dispositivos móveis, que nem sempre tem uma alta capacidade de processamento. Foi retirado tudo que não fosse a área de interesse na imagem, ficando apenas a folha da soja, em seguida, retirado as texturas da folha da soja e última etapa do pré-processamento foi de transformar todas as imagens para a escala de cinza para realizar o treinamento da RNA.

A biblioteca TensorFlow foi utilizada em todo o processo de desenvolvimento do aplicativo, desde o processo de treinamento da Rede Neural Artificial, até a identificação da doença em tempo real na folha da soja. O modelo de Rede Neural Convolutiva utilizado foi o Inception-V3. A acurácia obtida no treinamento foi de 87,9%.

Após o processo de treinamento da Rede Neural, se deu início ao desenvolvimento do aplicativo Android, o mesmo foi integrado ao classificador desenvolvido. O aplicativo tem a função de capturar a imagem através da câmera do celular e retornar para o usuário o resultado da classificação da imagem realizada pela RNA.

Foram utilizadas 458 imagens para realização dos testes aplicados. Ao final dos testes realizados o aplicativo desenvolvido atingiu uma acurácia de 84,61% e uma taxa de erro geral de 15,39% na detecção da ferrugem asiática na folha da soja.

3.3. MobileNetV3 for Image Classification

No artigo "MobileNetV3 for Image Classification", do autor Qian, Ning e Hu (2021), publicado na IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering - ICBAIE 2021, é descrito a estrutura principal da Rede Neural MobileNetV3 e também é realizada uma comparação de desempenho e precisão entre as Redes Neurais MobileNetV3, AlexNet, InceptionV3 e ShuffleNetV2 para classificação de imagens em dispositivos móveis.

Três conjuntos de dados diferentes foram utilizados para realizar o treinamento e testes das Redes Neurais. O primeiro conjunto de dados é o fruit360, incluindo 131 categorias de frutas e 67.692 imagens de treinamento. O segundo conjunto de dados são 1097 imagens de 10 macacos distintos. O terceiro conjunto de dados contém 16 categorias de pássaros e 150 imagens no total.

Para o conjunto de dados de frutas, o tempo de treinamento das RNAs foi de aproximadamente 40 minutos para MobileNetV3-Large, 33 minutos para MobileNetV3-Small, 417 minutos para AlexNet, 225 minutos para InceptionV3 e 42 minutos para ShuffleNetV2. Os modelos MobileNetV3-Large, MobileNetV3-Small, AlexNet e InceptionV3 atingiram quase 100% de precisão, apenas ShuffleNetV2 ficou abaixo, com precisão próxima dos 79%. Para este conjunto de dados, apenas as Redes MobileNetV3 obtiveram uma boa relação entre tempo de treinamento e teste com precisão.

No conjunto de dados de macacos, tanto a MobileNetV3 Small quanto a Large, atingiram a mesma precisão da InceptionV3, que é uma rede consideravelmente maior. Na fase de teste, a MobileNetV3-Large obteve os melhores resultados, seguido da

MobileNetV3-Small, InceptionV3, AlexNet e ShuffleNet, respectivamente. Esperava-se que InceptionV3 teria a melhor capacidade de generalização. No entanto, não foi o que aconteceu. A suposição é que o motivo esteja no tamanho de treinamento do conjunto de dados do macaco ser muito pequeno para que um modelo tão grande como a Inception alcance seu melhor desempenho, o que também mostra que o MobileNetV3 pode ser treinado com conjuntos de dados pequenos e alcançar alta precisão.

O conjunto de dados de pássaros é difícil de ser treinado pois contém apenas 150 imagens e possui menos de 10 imagens para cada categoria. Comparado com os dois conjuntos de dados anteriores, o tempo de treinamento foi menor devido ao tamanho do conjunto de imagens. O tempo de treinamento para todas as Redes Neurais foi muito semelhante, ficando próximo dos 14 minutos.

No processo de validação, as Redes MobileNetV3 Large e Small obtiveram resultados muito superiores em relação as demais Redes, muito disso se deve ao fato do conjunto de dados ser pequeno e desequilibrado.

Ao comparar e analisar os resultados experimentais, foi concluído que os modelos MobileNetV3 são capazes de executar tarefas de classificação de imagens com eficiência muito maior que as demais redes, mantendo uma boa precisão.

3.4. Comparação dos Trabalhos Correlatos

Após a leitura dos trabalhos correlatos, foi realizado um comparativo entre os mesmos e também com o presente projeto, conforme mostra a Tabela 1.

Tabela 1. Comparativo entre trabalhos

	Funck (2019)	Silva (2021)	Qian, Ning e Hu (2021)	Presente projeto
Rede Neural	Inception-V3	VGG-16	MobileNetV3, AlexNet, InceptionV3 e ShuffleNetV2	MobileNetV3
Dataset	Invasive, Ipmimages e coletadas em campo	EMBRAPA	Fruit360, espécies de macacos e espécies de pássaros	EMBRAPA e Kaggle
Tipos de Doenças	Ferrugem Asiática	Míldio, Ferrugem, Oídio, Folha Carijó e Crestamento Bacteriano, além de folhas saudáveis.	Não se aplica	Crestamento Bacteriano, Ferrugem e Míldio.
Aplicativo Mobile	Sim	Não	Não	Sim

Comparando com o trabalho de Funck (2019), o diferencial do presente projeto está na detecção de 3 doenças que podem se fazer presentes na folha da soja, além da utilização da Rede Neural MobileNetV3.

Já em relação ao trabalho de Silva (2021), o presente projeto diferencia-se ao realizar a detecção de doenças na folha da soja utilizando um aplicativo para dispositivos móveis e também pela Rede Neural MobileNetV3.

Com o trabalho de Qian, Ning e Hu (2021), não se aplica a comparação se tratando de doenças na folha da soja, porém, serviu como fundamentação para a escolha da Rede Neural MobileNetV3 no desenvolvimento deste projeto.

4. Materiais e Métodos

O propósito desta seção é introduzir os materiais e métodos empregados no desenvolvimento deste trabalho. A metodologia utilizada neste estudo foi baseada nos trabalhos de

Funck (2019) e Silva (2021). Conforme ilustrado na Figura 1, ela consiste nos seguintes passos: estudo e escolha do *dataset* de imagens rotuladas da soja; pré-processamento das imagens; definição e treinamento do modelo de RNA; desenvolvimento do aplicativo; validação do funcionamento do sistema.

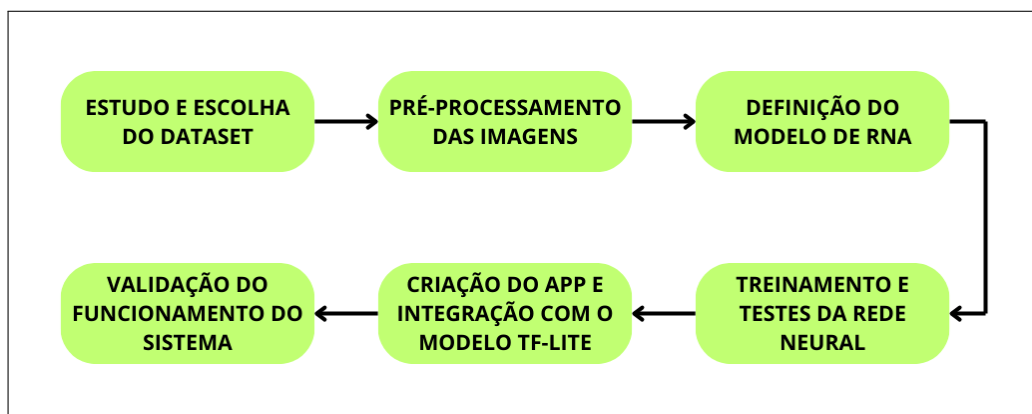


Figura 1. Visão geral da metodologia utilizada

4.1. Base de Imagens

Para realizar o treinamento de uma Rede Neural Artificial capaz de entregar resultados satisfatórios, é fundamental se ter uma base de dados com uma grande quantidade de imagens (SHORTEN; KHOSHGOFTAAR, 2019). O número de imagens pode variar de acordo com o problema ou com a margem de erro que se deseja para a solução proposta.

A base de dados utilizada para o treinamento e validação da Rede Neural, contém imagens retiradas de um banco de dados da Embrapa (2022) e outro da Kaggle (2023), somando um total de 299 imagens, divididas em 3 tipos de doenças na folha soja, conforme mostra a Tabela 2.

Tabela 2. Base de dados

Classe	Quantidade
Ferrugem asiática	100
Mildio	86
Creteamento Bacteriano	113

4.2. Pré-processamento das imagens

Primeiramente, todas as imagens foram submetidas a um processo de corte, adotando cortes verticais para aquelas de orientação vertical e cortes horizontais para as de orientação horizontal. Esse procedimento foi conduzido com o propósito de padronizar todas as imagens para um formato quadrado. Em seguida, todas elas foram redimensionadas para 380x380 pixels.

Padronizadas no mesmo tamanho e formato, procedeu-se à eliminação do fundo nas imagens da folha, visando realçar a área de interesse destinada ao treinamento da

Rede Neural. Conforme abordado por (KC et al., 2021), treinar Redes Neurais com conjuntos de dados que possuem fundo homogêneo resulta em uma precisão superior em comparação com imagens cujos fundos são desordenados, contendo elementos indesejados ao redor da área de interesse. Para realizar essa tarefa, empregou-se a biblioteca remgb¹.

Após a etapa de segmentação da folha da soja ao fundo, foram utilizadas técnicas de *Data Augmentation* (Aumento de Dados), com o objetivo de aumentar o volume da base de dados. Segundo Claro et al. (2020), este é o processo capaz de gerar novos dados virtuais para a base de imagens a partir dos dados reais, gerando assim uma maior quantidade e diversidade de imagens para realizar o treinamento da Rede Neural.

A biblioteca Keras, que segundo Ketkar e Ketkar (2017), oferece recursos para construir redes de aprendizado profundo, foi utilizada para aplicar as técnicas de aumento de dados. Através da classe *ImageDataGenerator* foram geradas novas imagens a partir de cada imagem original. Alterando configurações como rotação, deslocamentos vertical e horizontal, brilho, contraste e zoom, conforme exemplo na Figura 2. Dessa forma, aumentou significativamente a quantidade de dados disponíveis, passando a ter uma quantidade total de 3.962 imagens.

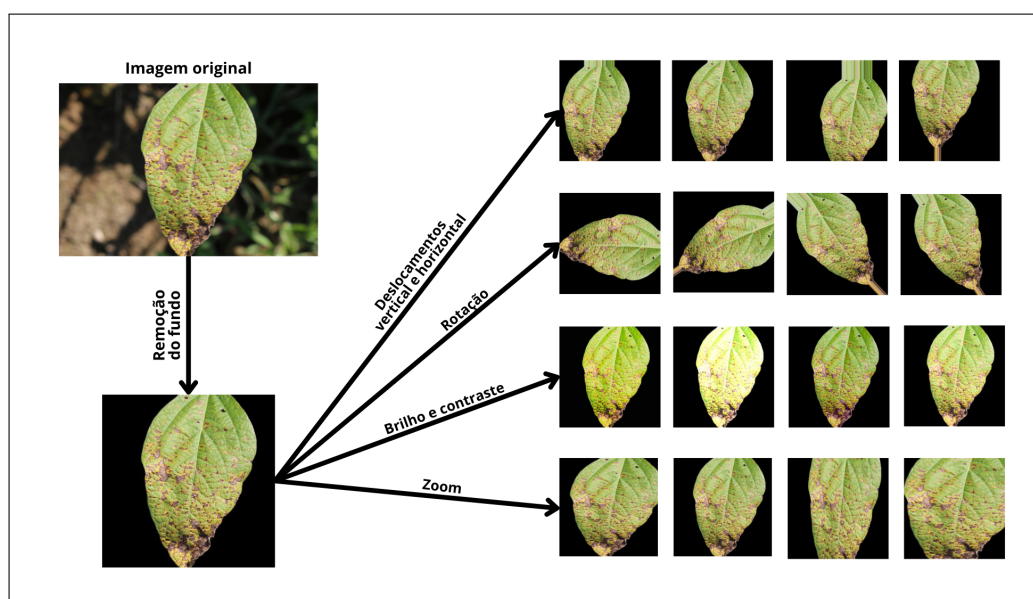


Figura 2. Pré-processamento das imagens

Através do Google Colab que foram realizados todos os procedimentos de manipulação e pré-processamento das imagens, bem como os de treinamento e validação da Rede Neural, descrito na *subseção 4.3*.

4.3. Definição e Treinamento do modelo de RNA

Com a base de imagens organizada e padronizada, foi realizado a definição e treinamento do modelo de RNA. No processo de treinamento, as imagens foram divididas aleato-

¹API para remover fundo de imagens (REMBG, 2023)

riamente em três grupos: treinamento, validação e teste. Com 80% das imagens para treinamento, 10% para validação e 10% para teste.

Por se tratar de um projeto desenvolvido para dispositivos móveis, a Rede Neural Artificial escolhida para realizar a classificação das imagens foi a MobileNetV3, que segundo Howard et al. (2019), possui uma arquitetura desenvolvida para CPUs deste tipo de dispositivo, em concordância com Qian, Ning e Hu (2021), que afirma que a presente Rede Neural Artificial apresenta um melhor equilíbrio entre eficiência e precisão para a classificação de imagens em dispositivos móveis comparado a outras redes neurais.

O treinamento da MobileNetV3 foi realizado no Google Colab, através do TensorFlow Lite - uma versão otimizada do TensorFlow -, que permitiu a conversão do modelo treinado para um formato compatível com o aplicativo Android.

Após ter sido realizado o treinamento da RNA, o modelo treinado foi exportado, para posteriormente ser integrado ao aplicativo.

A exportação do modelo não empregou técnicas de quantização pós-treinamento, as quais, conforme mencionado por (KLUSKA; ZIEBA, 2020), buscam diminuir a latência do modelo com uma perda mínima de precisão. Contudo, testes realizados no aplicativo desenvolvido neste projeto demonstraram que, para este contexto específico, a diferença de latência era praticamente imperceptível, enquanto a perda de precisão era considerável.

4.4. Desenvolvimento do Aplicativo

Após treinar a RNA com a biblioteca TensorFlow, deu-se início à fase de elaboração da aplicação para Android. A implementação foi conduzida por meio da linguagem de programação Java, utilizando a plataforma de desenvolvimento Android Studio. O modelo previamente treinado foi importado ao aplicativo.

O aplicativo tem a funcionalidade de coletar a imagem da folha da soja e realizar a classificação da mesma, tendo a possibilidade de carregar uma imagem da galeria ou realizar a captura através da câmera.

O usuário tem a opção de realizar a classificação da imagem com ou sem fundo, dado que as imagens utilizadas para treinar o modelo foram todas preparadas sem fundo, remover o fundo das imagens de teste durante a classificação cria uma coerência na entrada do modelo. Porém, vale ressaltar realizando este processo, o aplicativo levará mais tempo para o apresentar o resultado, além de necessitar de conexão com a internet, visto que remoção do fundo é realizada pela API `remove.bg`². Na Figura 3, é apresentada a estrutura do funcionamento do aplicativo.

²API para remover fundo de imagens (REMOVE.BG, 2023)

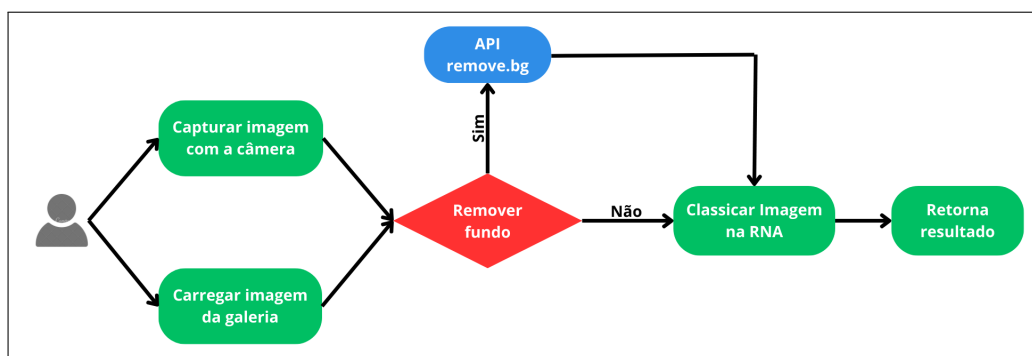


Figura 3. Funcionamento do aplicativo

Assim que a imagem for capturada/carregada e classificada, o aplicativo exibe na tela o resultado da classificação. Esse resultado inclui a classe com a maior probabilidade, conforme determinado pela rede neural, e a porcentagem de confiança do classificador sobre este resultado, conforme ilustrado na Figura 4.



Figura 4. Resultado da classificação da imagem para Ferrugem

4.5. Testes e Validação do Aplicativo

Na fase de testes e validação do aplicativo, primeiramente foi testado o modelo no Google Colab, e depois de exportado, foram feitos testes diretamente no aplicativo Android, com o objetivo de comparar o funcionamento da RNA antes e depois de sua exportação.

Após pré-processar as imagens, treinar a RNA e desenvolver o aplicativo, foram feitos testes para verificar o funcionamento do sistema. Inicialmente, o conjunto de testes (10% do total do *dataset*, que foi separado do conjunto de treinamento e validação na etapa 4.2), foi testado no Google Colab, logo após o treinamento e validação da RNA.

Com o modelo de Rede Neural Artificial (RNA) treinado já integrado ao aplicativo, foi possível realizar testes diretamente no Smartphone. Os testes no aplicativo foram feitos com as mesmas imagens presentes no conjunto de dados que foi testado no Google Colab. Ao total foram testadas 144 imagens manualmente no aplicativo, 48 de cada categoria.

Foram efetuados ajustes em todos os passos descritos na metodologia, incluindo modificações na etapa de Aumento de Dados, alterações no número de épocas de treinamento, tamanho do lote, taxa de aprendizagem, bem como a execução de experimentos com outros modelos de redes neurais, até que fosse encontrado um resultado considerado satisfatório para a conclusão do trabalho.

5. Resultados e discussão

Nesta seção, são apresentados os resultados obtidos no desenvolvimento e avaliação do trabalho.

5.1. Acurácia Geral

Durante o treinamento e validação do modelo no ambiente do Google Colab, foram obtidos resultados promissores. A acurácia alcançada durante o treinamento foi de 99%, e a acurácia na validação atingiu 98,5%. Além disso, o valor da função de perda (*loss*) foi reduzido para 36,8% em treinamento e 35,3% em validação, conforme mostra o gráfico da Figura 5. Com o conjunto de testes, o modelo alcançou uma acurácia de 99,5% e perda de 34,5%.

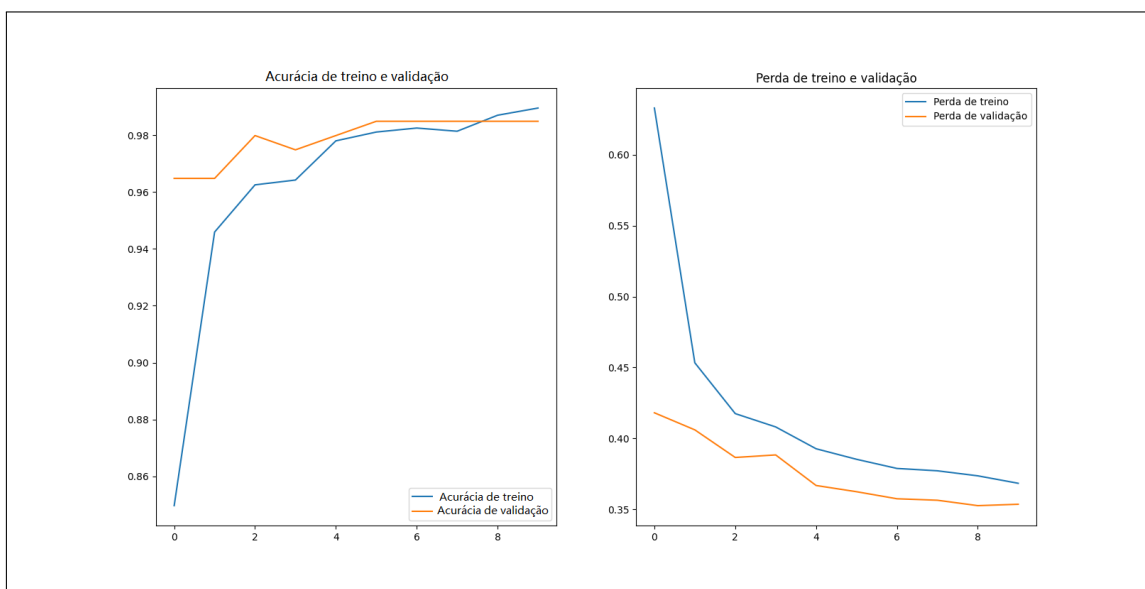


Figura 5. Gráficos de treinamento e validação

Esses resultados iniciais demonstraram a capacidade do modelo em aprender a tarefa de detecção das doenças Crestamento Bacteriano, Ferrugem e Míldio, através da folha da soja.

Realizando testes no aplicativo, com as 144 imagens provindas do conjunto de teste, o modelo classificou corretamente em 99,30% dos testes realizados, confirmando

os resultados obtidos no Google Colab. Entretanto, vale destacar que, por mais que sejam imagens que não foram utilizadas em treinamento, outras imagens semelhantes a elas estavam nos conjuntos de treinamento e validação, devido ao processo de *Data Augmentation*. Outro aspecto a salientar é que, em situações práticas de teste, os resultados podem variar, pois o modelo pode ser influenciado por mudanças nas condições de iluminação e na qualidade das imagens.

Para uma análise mais detalhada dos resultados, foi criada uma matriz de confusão que ilustra a distribuição das classificações do modelo em relação às classes reais das amostras, apresentada na Tabela 3.

Tabela 3. Matriz de confusão

		Valor Predito		
		Ferrugem	Míldio	Crestamento Bacteriano
Valor Real	Ferrugem	47	0	1
	Míldio	0	48	0
	Crestamento Bacteriano	0	0	48

Conforme mostra a matriz de confusão, as categorias Míldio e Crestamento Bacteriano foram classificadas corretamente em 100% das vezes. A classe Ferrugem, foi identificada corretamente em 47 de 48 testes, o que representa uma acurácia de 97,9%.

O presente projeto alcançou uma acurácia geral superior em comparação aos estudos conduzidos por Funck (2019) e Silva (2021), abordados na seção de Trabalhos Correlatos.

5.2. Limitações e Trabalho Futuro

Este estudo reconhece algumas limitações, como a necessidade de um conjunto de dados mais preciso e equilibrado, considerando a variação nas condições da folha da soja. Além disso, por mais que a análise da Rede Neural seja feita localmente, a etapa de remoção do fundo da imagem é feita através de uma API, sendo necessário conexão com a internet para tal. Outra melhoria significativa para este trabalho seria implementação de um mecanismo para verificar se a imagem fornecida é realmente uma folha de soja antes de tentar realizar a classificação. Atualmente ele retornará um resultado para qualquer imagem disponibilizada a ele. Sugere-se também, a realização de uma avaliação prática da aplicação pelos usuários, visando validar não apenas a eficácia técnica da solução proposta, mas também a usabilidade e aceitação por parte dos agricultores.

Portanto, é importante destacar que o trabalho pode se beneficiar de melhorias contínuas, incluindo:

- Coleta de dados mais abrangente e diversificada.
- Aprimoramento da capacidade de detecção de doenças em estágios iniciais.
- Desenvolvimento de uma interface de usuário mais amigável.
- Implementação da remoção de fundo da imagem localmente, permitindo que o aplicativo funcione independente de sua conexão com a internet.
- Desenvolver um mecanismo para identificar se a imagem fornecida é de uma folha de soja.

- Avaliação prática da aplicação pelos usuários.

Essas áreas representam oportunidades para pesquisa futura e desenvolvimento contínuo do modelo e do aplicativo.

6. Conclusão

O presente projeto explorou alguns dos problemas que são enfrentados na produção de soja, destacando que a digitalização e a conexão no campo são essenciais para impulsionar o crescimento agrícola.

A aplicação conjunta de técnicas de processamento de imagens e aprendizado de máquina demonstraram que Redes Neurais Artificiais são excelentes ferramentas de análise e identificação de padrões em imagens.

Embora tenha sentido a limitação de não dispor de um conjunto de dados mais diversificado e abrangente, o estudo também mostrou que a utilização de técnicas de Aumento de Dados pode efetivamente enriquecer um conjunto pequeno de imagens, transformando-o em uma base sólida para o treinamento de uma Rede Neural.

A eficaz implementação dessas abordagens não apenas fornece uma ferramenta de auxílio no monitoramento da saúde das plantações de soja, mas também contribui para o desenvolvimento de estratégias mais eficientes e sustentáveis no manejo de doenças.

Os resultados obtidos foram considerados satisfatórios para a resolução do problema. Com uma acurácia geral de 99,30% em testes no aplicativo, a MobileNetV3, conforme estudado na literatura, demonstrou eficácia ao oferecer boa precisão mesmo em dispositivos com recursos computacionais limitados. Isso reforça sua relevância em cenários onde a otimização de recursos é crucial, destacando-a como uma boa escolha para o desenvolvimento deste projeto.

Este estudo ressalta a importância da inovação tecnológica na agricultura, destacando seu papel crucial na busca por soluções que abordem os desafios complexos enfrentados pelo setor agrícola.

Referências

- ABADI, M. et al. Tensorflow: a system for large-scale machine learning. In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. [S.l.: s.n.], 2016. p. 265–283.
- BISONG, E.; BISONG, E. Google colaboratory. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, Springer, p. 59–64, 2019.
- BONATO, E. R.; BONATO, A. L. V. A soja no brasil: história e estatística. Londrina: EMBRAPA-CNPSo, 1987., 1987.
- CAMPEÃO, P.; SANCHES, A. C.; MACIEL, W. R. E. Mercado internacional de commodities: uma análise da participação do brasil no mercado mundial de soja entre 2008 e 2019. *Desenvolvimento em Questão*, v. 18, n. 51, p. 76–92, 2020.

- CARDOSO, D. C. B.; LOUZADA, G. A. d. S. Lagarta falsa-medideira pseudoplusia includens, importância econômica e seu difícil controle na cultura da soja. *Acta Iguazu*, v. 4, n. 3, p. 48–65, 2012.
- CASTRO, F. D.; CASTRO, M. D. Redes neurais artificiais. *DCA/FEEC/Unicamp*, 2001.
- CLARO, M. et al. Utilização de técnicas de data augmentation em imagens: Teoria e prática. *Sociedade Brasileira de Computação*, 2020.
- COLLABORATORY, G. *Google Colaboratory*. 2023. Disponível em: <https://colab.google/>. Acesso em: Jun. 2023.
- CONAB. *BOLETIM DA SAFRA DE GRÃOS*. 2023. Disponível em: <https://www.conab.gov.br/info-agro/safras/graos/boletim-da-safra-de-graos>. Acesso em: Jun. 2023.
- CONTE, O.; POSSAMAI, E. J.; FILHO, P. C. Resultados do monitoramento integrado da colheita da soja na safra 2019/2020 no paran . *Embrapa Soja-Circular T cnica (INFOTECAE)*, 2020.
- CORR A-FERREIRA, B. S. Amostragem de pragas da soja. *Soja: manejo integrado de insetos e outros artr podes-praga. Bras lia-DF: Embrapa Soja*, p. 631–672, 2012.
- EMBRAPA. *Reposit rio Digiphatos*. 2022. Disponível em: <https://www.digipathos-rep-cnptia.embrapa.br/jspui/simple-search?filterquery=Soja+%28Soybean%29&filtername=crop&filtertype>equals>. Acesso em: Jun. 2022.
- FERREIRA, A. d. S. Redes neurais convolucionais profundas na detec o de plantas daninhas em lavoura de soja. 2017.
- FILHO, H. M. de S. et al. Condicionantes da ado o de inova es tecnol gicas na agricultura. *Cadernos de Ci ncia & Tecnologia*, v. 28, n. 1, p. 223–255, 2011.
- FITCH, F. B. Warren s. mcculloch and walter pitts. a logical calculus of the ideas immanent in nervous activity. bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. *The Journal of Symbolic Logic*, Cambridge University Press, v. 9, n. 2, p. 49–50, 1944.
- FLECK, L. et al. Redes neurais artificiais: Princ pios b sicos. *Revista Eletr nica Cient fica Inova o e Tecnologia*, v. 1, n. 13, p. 47–57, 2016.
- FUNCK, F. C. *Detectando a ferrugem asi tica na folha da soja utilizando redes neurais convolucionais*. Disserta o (B.S. thesis) — Universidade Tecnol gica Federal do Paran , 2019.
- GU, J. et al. Recent advances in convolutional neural networks. *Pattern recognition*, Elsevier, v. 77, p. 354–377, 2018.
- HENNING, A. A. et al. Manual de identifica o de doen as de soja. *Documentos 256*, v. 5  edic o, 2014.
- HOWARD, A. et al. Searching for mobilenetv3. In: *Proceedings of the IEEE/CVF international conference on computer vision*. [S.l.: s.n.], 2019. p. 1314–1324.
- KAGGLE. *Soybean Disease Dataset*. 2023. Disponível em: <https://www.kaggle.com/datasets/lucaskishima/soybean-disease-data>. Acesso em: Jun. 2023.

- KC, K. et al. Impacts of background removal on convolutional neural networks for plant disease classification in-situ. *Agriculture*, MDPI, v. 11, n. 9, p. 827, 2021.
- KETKAR, N.; KETKAR, N. Introduction to keras. *Deep learning with python: a hands-on introduction*, Springer, p. 97–111, 2017.
- KLUSKA, P.; ZIEBA, M. Post-training quantization methods for deep learning models. In: SPRINGER. *Intelligent Information and Database Systems: 12th Asian Conference, ACIIDS 2020, Phuket, Thailand, March 23–26, 2020, Proceedings, Part I 12*. [S.l.], 2020. p. 467–479.
- LITE, T. Tensorflow lite. *Retrieved May*, Springer, v. 20, p. 2022, 2022.
- MARTINS, T. G. d. S.; SCHOR, P. *Aprendizagem de máquina em análise de imagens na oftalmologia*. [S.l.]: SciELO Brasil, 2022.
- MILANO, D. de; HONORATO, L. B. Visao computacional. *Faculdade de Tecnologia, Universidade Estadual de Campinas*, 2010.
- MISSÃ, M. R. et al. Soja, origem, classificaÇÃo, utilizaÇÃo e uma visÃo abrangente do mercado. *Maringá Management*, v. 3, n. 1, 2006.
- NASCIMENTO, F. G. et al. Controle de doenças em soja com diferentes manejos e momentos de aplicação de fungicidas baseados no índice de área foliar. Universidade Federal de Uberlândia, 2018.
- OLIVEIRA, L. F. et al. Programas de fungicidas no controle de antracnose na cultura da soja. *Tecno-Lógica*, v. 25, n. 2, p. 209–220, 2021.
- QIAN, S.; NING, C.; HU, Y. Mobilenetv3 for image classification. In: IEEE. *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. [S.l.], 2021. p. 490–497.
- REMBG. *Rembg*. 2023. Disponível em: <https://github.com/danielgatis/rembg>. Acesso em: Mai. 2023.
- REMOVE.BG. *Remove.bg*. 2023. Disponível em: <https://www.remove.bg/pt-br/api>. Acesso em: Nov. 2023.
- RODRIGUES, D. A. Deep learning e redes neurais convolucionais: reconhecimento automático de caracteres em placas de licenciamento automotivo. Universidade Federal da Paraíba, 2018.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- SANTOS, T. T. et al. Visão computacional aplicada na agricultura. *Embrapa Agricultura Digital-Capítulo em livro científico (ALICE)*, In: MASSRUHÁ, SMFS; LEITE, MA de A.; OLIVEIRA, SR de M.; MEIRA, CAA . . . , 2020.
- 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.
- SILVA, E. C. d. et al. Técnicas de deep learning: autoencoder e long short-term memory (lstm) no desenvolvimento de sensores virtuais e na detecção e diagnóstico de falhas em processos reacionais complexos. Universidade Federal de Alagoas, 2023.

SILVA, G. V. G. *Detecção de doenças em folhas de soja utilizando redes neurais convolucionais*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2021.

ZHAO, H. et al. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, IEEE, v. 3, n. 1, p. 47–57, 2016.