



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Lucca Machado da Silva

**Diagnóstico Precoce da Saúde em Folhas de Mudas de Bananeira por meio de
Processamento de Imagem**

Araranguá
2024

Lucca Machado da Silva

**Diagnóstico Precoce da Saúde em Folhas de Mudas de Bananeira por meio de
Processamento de Imagem**

Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Computação submetido ao Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Alexandre Leopoldo Gonçalves, Dr.

Araranguá
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Silva, Lucca Machado
Diagnóstico Precoce da Saúde em Folhas de Mudas de
Bananeira por meio de Processamento de Imagem / Lucca
Machado Silva ; orientador, Alexandre Leopoldo Gonçalves,
2024.
36 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá,
Graduação em Engenharia de Computação, Araranguá, 2024.

Inclui referências.

1. Engenharia de Computação. 2. Diagnóstico da Saúde em
Folhas. 3. Aprendizado Profundo. 4. Redes Neurais
Convolucionais. 5. Vetores Densos. I. Gonçalves, Alexandre
Leopoldo. II. Universidade Federal de Santa Catarina.
Graduação em Engenharia de Computação. III. Título.

Lucca Machado da Silva

**Diagnóstico Precoce da Saúde em Folhas de Mudas de Bananeira por meio de
Processamento de Imagem**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 20 de junho de 2024.

Prof. Jim Lau, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Alexandre Leopoldo Gonçalves, Dr.
Universidade Federal de Santa Catarina
Orientador

Prof. Alison Roberto Panisson, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Roderval Marcelino, Dr.
Avaliador
Universidade Federal de Santa Catarina

Diagnóstico Precoce da Saúde em Folhas de Mudas de Bananeira por meio de Processamento de Imagem

Lucca Machado da Silva*

Alexandre Leopoldo Gonçalves †

2024, JUNHO

Resumo

O aumento da demanda por alimentos devido ao crescimento populacional, aliado à insegurança alimentar e às perdas significativas nas culturas causadas por pragas, exige soluções eficazes para a agricultura. A detecção manual de doenças em plantas é inviável para pequenos e médios agricultores, resultando no uso excessivo de pesticidas, que estão diretamente relacionados a uma maior incidência de câncer na população. De maneira geral, os primeiros sintomas de doenças em plantas se manifestam visualmente nas folhas em seus estágios iniciais. A partir deste contexto, o presente trabalho propõe um método para detectar o estado de saúde em folhas de mudas de bananeiras. Para a construção do método, foram utilizados conceitos de Aprendizado de Máquina, como as Redes Neurais Convolucionais Profundas em conjunto com Bancos de Dados Vetoriais. O método proposto mostrou-se eficiente na segmentação individual de folhas em uma imagem de entrada, atingindo valores de mAP50 superiores a 94%. Para a classificação entre folhas doentes e saudáveis, o método obteve uma acurácia geral de 92%, com uma precisão de 40% para folhas doentes e de 97% para folhas saudáveis. Diante dos resultados, o método proposto se mostra promissor para a estimativa da quantidade de folhas saudáveis e doentes em uma estufa, auxiliando agricultores no monitoramento do ambiente.

Palavras-chaves: Diagnóstico da Saúde em Folhas, Aprendizado Profundo, Redes Neurais Convolucionais, Banco de Dados Vetoriais, Vetores Densos.

*lucca.machado@grad.ufsc.br

†a.l.goncalves@ufsc.br

Early Diagnosis of the Health of Banana Seedlings Leaves Through Image Processing

Lucca Machado da Silva* Alexandre Leopoldo Gonçalves †

2024, JUNE

Abstract

The increase in food demand due to population growth, coupled with food insecurity and significant crop losses caused by pests, requires effective solutions for agriculture. Manual detection of plant diseases is unfeasible for small and medium-sized farmers, resulting in the excessive use of pesticides, which are directly related to a higher incidence of cancer in the population. Generally, the first symptoms of plant diseases visually manifest on the leaves in their early stages. In this context, the present work proposes a method to detect the health status of banana seedling leaves. For the development of the method, concepts from Machine Learning, such as Deep Convolutional Neural Networks, were used in conjunction with Vector Databases. The proposed method proved to be efficient in the individual segmentation of leaves in an input image, achieving mAP50 values greater than 94%. For the classification between diseased and healthy leaves, the method achieved an overall accuracy of 92%, with a precision of 40% for diseased leaves and 97% for healthy leaves. Considering the results, the proposed method is promising for estimating the number of healthy and diseased leaves in a greenhouse, assisting farmers in monitoring the environment.

Key-words: Leaf Health Diagnosis, Deep Learning, Vector Database, Convolutional Neural Networks, Embeddings.

*lucca.machado@grad.ufsc.br

†a.l.goncalves@ufsc.br

1 Introdução

Ao longo das últimas décadas, o crescimento populacional intenso contribuiu significativamente com o aumento da demanda por alimentos. Estima-se que seja necessário um aumento de cerca de 60% na produção global de alimentos até o ano de 2050, a fim de garantir a segurança alimentar da população mundial (DIJK *et al.*, 2021). Além disso, dados levantados pelo Banco Mundial indicam que a prevalência de insegurança alimentar moderada ou severa já está em crescimento no planeta, tendo aumentado de 21.7% em 2015 para 29.6% em 2022. Esse fenômeno afetou, em sua maioria, países mais pobres. Na América Latina e nas ilhas do Caribe, esse número chegou a alcançar 40.3% da população no ano de 2021 (FAO, 2022). Outro fator extremamente importante a ser levado em conta e que influencia diretamente a segurança alimentar é a propagação de pragas em plantios. De acordo com FAO (2021), as principais culturas alimentares, como arroz, trigo, batata e banana, sofrem perdas de até 40% em sua produção total devido a pragas que afetam as plantas.

Embora seja possível realizar o reconhecimento e a detecção de doenças em plantas manualmente, isso depende fortemente da experiência do agricultor ou do auxílio de especialistas. Além disso, considerando a extensa área de cultivo utilizada por produtores, torna-se inviável para agricultores de pequeno e médio porte realizar análises manuais para identificar áreas de plantio afetadas por doenças. Diante disso, uma parte significativa dos agricultores tradicionalmente opta pela utilização excessiva de pesticidas para controlar a propagação de pragas em suas plantações. No entanto, esse uso excessivo de pesticidas está fortemente associado à causa de diversos tipos de câncer (BASSIL *et al.*, 2007).

De modo geral, os primeiros sintomas de doenças em plantas se manifestam fisicamente em suas folhas, sendo perceptíveis em seus estágios iniciais (SANKARAN *et al.*, 2010). Com isso em mente, muitos métodos foram desenvolvidos para detectar doenças ou avaliar a saúde das plantas, concentrando-se na análise de características visuais das folhas. Embora haja uma variedade de técnicas disponíveis, a abordagem mais comum entre os pesquisadores envolve o uso de Inteligência Artificial (do inglês *Artificial Intelligence* - AI), particularmente Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* - CNN), para extrair detalhadamente todas as características da planta e, assim, identificar doenças ou determinar o estado de saúde da planta ou da folha (AABIDI *et al.*, 2023; CHEN; PAN; WU, 2023).

Apesar de redes neurais serem amplamente adotadas para a solução deste tipo de problema, existem cenários em que a utilização delas, sem o agregamento de novas técnicas, se torna pouco eficiente, em grande parte devido ao fato de que técnicas de aprendizado profundo necessitam de uma grande quantidade de dados. Existem cenários em que o objetivo é realizar a detecção de certas doenças que são raras ou incomuns. Neste caso, a construção de um conjunto de dados de um tamanho adequado para uma rede neural complexa se torna difícil ou até mesmo impraticável (ARGÜESO *et al.*, 2020). Estudos recentes têm tido como foco a utilização de técnicas de aprendizado profundo em cenários em que o conjunto de dados é limitado, sendo usualmente denominados de *Few-Shot Learning*.

Tendo em vista esse contexto, o presente trabalho tem como objetivo a proposição de um método que estime a quantidade de folhas saudáveis e doentes considerando imagens de plantas obtidas a partir de cultivos em estufas (do inglês *greenhouse*). A estratégia abordada consiste na utilização de uma CNN em conjunto com vetores densos (do inglês *embeddings*) e banco de dados vetorial para a classificação do estado de saúde de folhas.

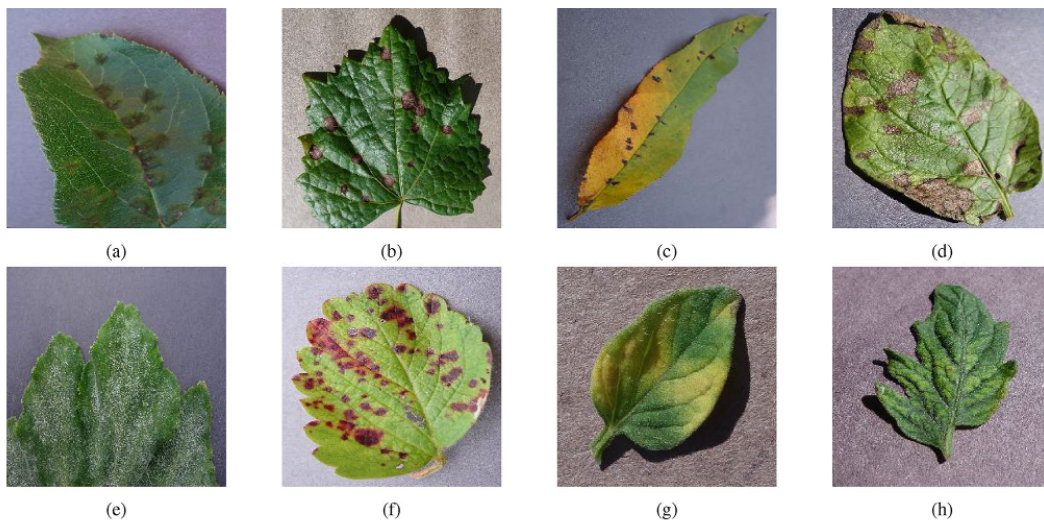
Além da seção introdutória, o trabalho conta com outras cinco seções. Na Seção 2 é realizada a apresentação de conceitos teóricos essenciais relacionados à temática do trabalho. A Seção 3 aborda os trabalhos correlatos e as técnicas utilizadas pelos autores para a resolução de problemas similares. A Seção 4 apresenta detalhadamente cada uma das etapas realizada no método proposto. Na Seção 5 são apresentados os resultados obtidos e, por fim, a Seção 6 apresenta as considerações finais e trabalhos futuros.

2 Fundamentação Teórica

2.1 Detecção de Doenças em Plantas

Doenças em plantas frequentemente se manifestam por fenótipos visuais, como manchas, deformações e descolorações nas folhas, que são indicadores vitais para detectar e combater eficazmente problemas que afetam o crescimento e a produtividade das plantas. A Figura 1 mostra exemplos típicos de tais fenótipos associados a doenças de plantas, enfatizando a importância de compreender e monitorar essas características visuais. A análise dessas características desempenha um papel fundamental na identificação e classificação das doenças, possibilitando a adoção de medidas preventivas e corretivas.

Figura 1 – Sintomas visuais em folhas de plantas doentes.



Fonte: [Ma, Chen e Xu \(2024\)](#).

Na literatura atual, o método mais prevalente para a detecção de doenças em folhas de plantas é o uso de redes neurais convolucionais. De maneira geral, existem dois cenários abordados pelos autores [Devanna et al. \(2022\)](#), [Ma, Chen e Xu \(2024\)](#). No primeiro cenário, o conjunto de imagens é simples, ou seja, as imagens contêm apenas folhas individuais, sem fundos complexos que necessitariam de algum tipo de tratamento prévio. Nesse tipo de cenário, os autores optam pela utilização de CNNs para a classificação de objetos, visando a extração de características individuais de cada folha e na predição da doença correspondente. Já no segundo cenário, quando as folhas estão localizadas em ambientes complexos, ou seja, com diversas folhas sobrepostas ou com outros objetos, é necessária uma etapa extra: a segmentação. Nesta etapa, as folhas são extraídas individualmente

ou em conjunto a partir de uma imagem e, então, são submetidas a uma rede neural que classifica as folhas em seus respectivos grupos de doenças (WALDAMICHAEL; DEBELEE; AYANO, 2022; ZHANG *et al.*, 2022).

A utilização de redes neurais para a detecção de doenças em plantas se mostra extremamente eficiente em cenários controlados, como no trabalho desenvolvido por Zhang *et al.* (2022), no qual os autores desenvolveram um modelo de detecção de doenças no trigo unindo técnicas de segmentação de imagens por limiarização de cores e a utilização de CNNs como *AlexNet*, *GoogLeNet*, entre outras. Neste trabalho, as plantas de trigo estavam contidas em uma *Greenhouse*, isto é, um cenário controlado no qual as imagens de trigo não estavam sujeitas a grandes níveis de ruído, diferenças de luminosidade ou cenários muito complexos.

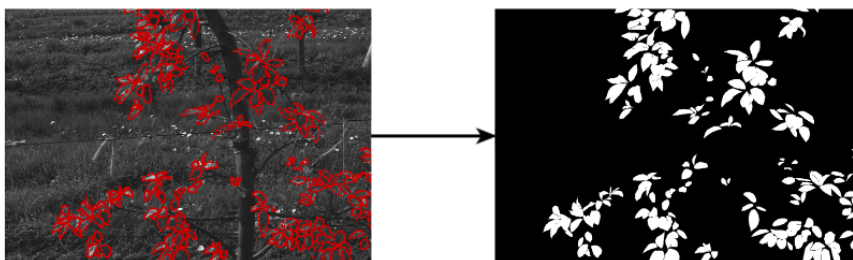
No entanto, na grande maioria dos casos, para aplicações práticas da vida real, tem-se um ambiente menos controlado sendo necessário lidar com diversos aspectos que dificultam tanto a segmentação dos objetos quanto a inferência do tipo de doença que está afetando a saúde da planta em questão. Um dos fatores que mais afetam a segmentação das imagens é a diferença de luminosidade entre as áreas de interesse de cada imagem. Para superar tais obstáculos, autores têm utilizado diferentes técnicas. No estudo realizado por Devanna *et al.* (2022), foi empregada a técnica de aprendizado por transferência em estágios. Inicialmente, o modelo de rede neural foi treinado com imagens de frutos de romã em um fundo branco sem iluminação natural. Posteriormente, o treinamento foi realizado com o mesmo fundo e iluminação natural e, por fim, em cenários reais com iluminação natural, abordando assim o problema de imagens complexas de forma progressiva.

Em Uryasheva *et al.* (2022), os autores tiveram como foco criar uma aplicação que possibilitasse realizar a segmentação de folhas de macieiras, a qual permite ao usuário observar o estado de saúde das mesmas. Para enfrentar a complexidade dos cenários práticos citados anteriormente, os autores desenvolveram uma aplicação baseada em diversos filtros nas imagens com o objetivo de remover ruídos, diferenças de luminosidade e artefatos que possam prejudicar a segmentação das folhas do restante dos objetos na imagem.

2.2 Segmentação de imagens

De acordo com Debelee *et al.* (2019), a segmentação de imagens pode ser definida como o processo de particionamento de uma imagem digital em diversos segmentos ou áreas de interesse que podem ser obtidos através da análise de certas propriedades visuais que os objetos de interesse possuem em comum, como a cor, textura ou formato dos objetos. A Figura 2 demonstra a segmentação de folhas de macieiras através do uso de uma CNN.

Figura 2 – Segmentação de folhas de macieira



Fonte: Uryasheva *et al.* (2022).

Entretanto, a segmentação é um processo desafiador, tendo em vista os cenários citados anteriormente. Cada imagem fotográfica está sujeita a diferentes níveis de ruídos, iluminação variável e uma grande diversidade de condições ambientais que podem afetar significativamente a qualidade e precisão do processo de segmentação. Além disso, a complexidade das estruturas e objetos presentes nas imagens pode variar consideravelmente, tornando ainda mais crucial o desenvolvimento de técnicas robustas e flexíveis para a segmentação.

Por outro lado, o principal motivo para a ampla adoção da segmentação de imagens está relacionado à sua capacidade de eliminar informações desnecessárias, tornando diversos processos posteriores mais eficientes. Um exemplo disso pode ser encontrado no trabalho de [Bhujel *et al.* \(2022\)](#), em que a segmentação de imagens foi utilizada para analisar a severidade de doenças em folhas de morango. Nesse contexto, os autores removeram objetos desnecessários da imagem, obtendo assim uma imagem fotográfica contendo apenas as folhas, otimizando consequentemente a análise posterior. Essa capacidade de refinar e isolar objetos de interesse torna a segmentação de imagens uma ferramenta valiosa em várias aplicações.

No estudo conduzido por [Waldamichael, Debelee e Ayano \(2022\)](#), a segmentação de imagens foi alcançada por meio da técnica de limiarização de cores. Realizou-se uma análise individual das componentes de cores de cada imagem obtida. Após a análise desses valores, foram estabelecidos intervalos para que os pixels fossem considerados como áreas de interesse. Dessa forma, obteve-se imagens segmentadas contendo apenas folhas de plantas, descartando informações desnecessárias por meio da análise das características dos pixels.

Embora o uso de CNNs para segmentação seja consideravelmente mais complexo do que a limiarização de cores, principalmente devido à maior demanda por poder computacional, sua utilização possibilita o desenvolvimento de aplicações mais sofisticadas. Isso ocorre porque as CNNs são mais eficientes na segmentação de áreas de interesse em cenários complexos. Isso pode ser observado no trabalho realizado por [Afonso *et al.* \(2020\)](#), onde os autores exploraram diferentes arquiteturas de CNNs chamada *Mask R-CNN* para segmentar frutos de tomate em uma estufa inteligente, com o objetivo de estimar a quantidade de frutos maduros e verdes.

Tendo em vista as diferentes técnicas apresentadas nesta seção, é possível observar a ampla gama de métodos disponíveis para a segmentação de objetos em uma determinada imagem, variando significativamente conforme as características específicas de cada problema a ser tratado. Neste trabalho, foram exploradas técnicas de segmentação utilizando limiarização de cores e também a aplicação da CNN profunda YOLO para a segmentação de folhas a partir de uma imagem de entrada.

2.2.1 Limiarização de cores

Dentre os diversos métodos de segmentação de imagens disponíveis, a limiarização de cor se destaca como um dos mais eficientes. Segundo [Feng e He \(2017\)](#), a limiarização de cor pode ser definida como o particionamento de uma imagem digital em regiões que possuem similaridades por meio de um critério de cor pré-estabelecido. Para que isso seja possível, é necessário determinar valores numéricos para os pixels a serem mantidos, permitindo a obtenção de um intervalo dos pixels que devem ser preservados na imagem final. Contudo, esse tipo de técnica não é eficiente em todos os cenários. A limiarização de cor se destaca na segmentação de objetos que possuem uma cor muito diferente dos outros objetos em uma determinada imagem, conseguindo eliminar itens indesejados de maneira

simples. No entanto, se o objeto a ser segmentado possuir cores muito similares às dos objetos de fundo, a limiarização de cor deixa de ser a melhor opção.

Segundo [Raof et al. \(2008\)](#) a regra geral para a segmentação de objetos utilizando a limiarização de cor pode ser definida como:

$$Threshold(R, G, B) = \begin{cases} 1 & \text{se } (R > R_{\min} \text{ e } G > G_{\min} \text{ e } B > B_{\min}) \\ & \text{e } (R < R_{\max} \text{ e } G < G_{\max} \text{ e } B < B_{\max}) \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

Onde R , G e B são os valores para cada componente do espaço de cor RGB, e R_{\min} , G_{\min} , B_{\min} , R_{\max} , G_{\max} e B_{\max} são os limiares definidos pela análise prévia da imagem, resultando no valor final para a limiarização de um determinado pixel. A Equação 1 verifica se todos os parâmetros de um pixel se encaixam nos limites de intervalos propostos. Se o pixel estiver dentro desses limites, o valor da máscara obtida para aquele pixel será um, ou seja, é um pixel válido para a limiarização. Caso contrário, o valor será definido como zero e o pixel será descartado. Um exemplo disso pode ser encontrado na Figura 3, onde, por meio do estabelecimento de limiares de cores e da aplicação de máscaras, foi possível remover o fundo da imagem, obtendo uma segmentação que contém apenas a folha.

Outro fator importante a ser destacado é a diferença que cada espaço de cor proporciona. O espaço mais comum a ser utilizado é o RGB, no qual cada componente lista a intensidade das cores Vermelho, Verde e Azul. Todavia, existem outros modelos de espaço de cor, como o HSV (Matiz, Saturação e Valor), que oferece uma representação alternativa das cores. Enquanto o espaço RGB é mais adequado para representar cores em termos de luz aditiva, o espaço HSV permite descrever cores com base em características perceptuais, como matiz, saturação e brilho. A escolha do espaço de cor apropriado depende das necessidades específicas da aplicação e das propriedades das cores que se deseja analisar.

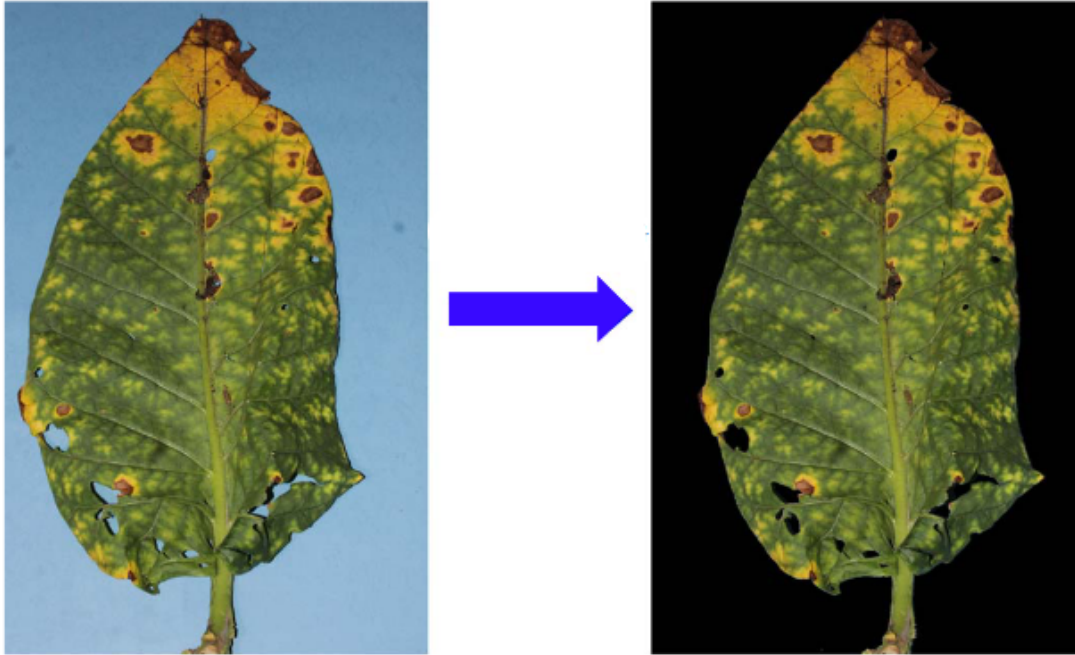
Em [Zhang et al. \(2022\)](#), os autores propuseram a utilização de duas abordagens para realizar a segmentação das imagens, tendo em vista que foram utilizadas duas câmeras fotográficas com características diferentes, uma que captura as imagens em RGB e outra em infravermelho. Inicialmente, os autores converteram as imagens em RGB para o espaço de cor Lab (L para luminosidade e a,b para dimensões de cores) e realizaram a segmentação das mesmas. Já para as imagens em infravermelho, os autores optaram pela utilização de um algoritmo de *K-means clustering* em vez da limiarização de cor.

[Waldamichael, Debelee e Ayano \(2022\)](#) aplicaram a limiarização de cores em dois momentos diferentes em seu trabalho. Inicialmente, o método foi aplicado com o objetivo de remover o fundo de uma imagem, mantendo apenas uma folha na mesma. Após esse processo, a limiarização foi realizada novamente com o intuito de obter apenas os pontos de doenças que afetavam a folha em questão. Para isso, os autores utilizaram o espaço de cor HSV em ambas as segmentações.

2.3 Aprendizado Profundo

De acordo com [Sheikh, Prins e Schrijvers \(2023\)](#), a Inteligência Artificial (do inglês *Artificial Intelligence* - AI) pode ser definida de maneira mais estrita como a imitação da inteligência humana por parte das máquinas. No entanto, existem diversas definições para

Figura 3 – Processo de limiarização de cor



Fonte: Setiawan *et al.* (2023).

o mesmo termo, algumas sendo mais abrangentes que outras. A definição mais comum na literatura pode ser descrita como a seguinte: a inteligência artificial é uma tecnologia que permite às máquinas imitarem diversas tarefas complexas realizadas por seres humanos.

As Redes Neurais Artificiais (do inglês *Artificial Neural Network* - ANN) buscam imitar o processamento de informações observado nos neurônios biológicos. Em sua forma mais simples, uma unidade de processamento em uma ANN recebe estímulos ponderados como entradas, as quais são somadas e passadas por uma função de ativação (GURESEN; KAYAKUTLU, 2011). Esta função de ativação introduz não linearidade na resposta da unidade, permitindo que a rede modele relações complexas nos dados. Uma representação matemática simplificada desse processo pode ser expressa pela seguinte equação:

$$Y = f \left(\sum_{i=1}^n W_i X_i + b \right) \quad (2)$$

Onde Y é a saída da unidade, X_i são as entradas, W_i são os pesos sinápticos correspondentes a cada entrada, b é o viés (bias), e f é a função de ativação. A Equação 2 ilustra a transformação não linear realizada por cada unidade de processamento em uma rede neural artificial.

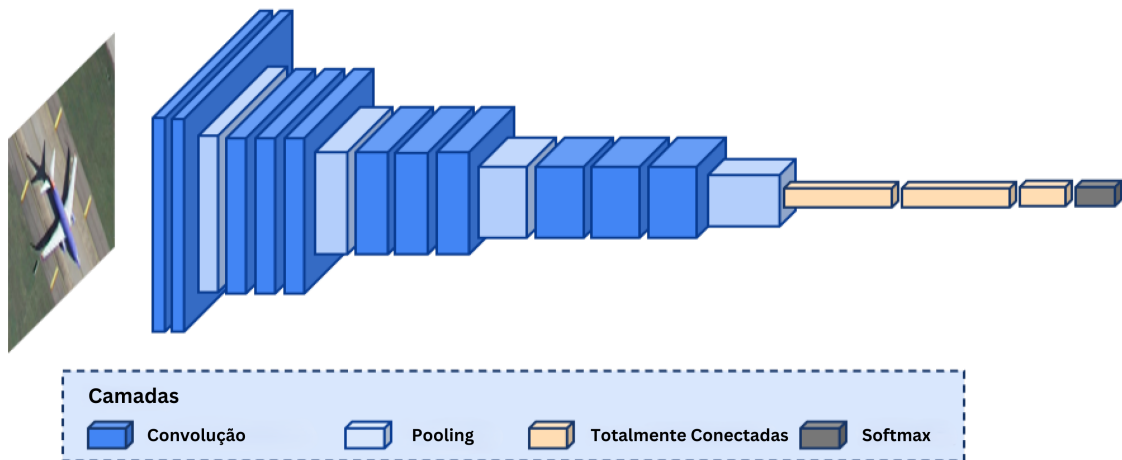
Já o Aprendizado Profundo (do inglês *Deep Learning* - DL) é um conjunto de métodos de Aprendizado de Máquina (do inglês *Machine Learning* - ML) nos quais a computação é conduzida através de múltiplas camadas de Redes Neurais Profundas (do inglês *Deep Neural Network* - DNN) conectadas entre si. O termo “profundo” faz alusão à ideia de empregar múltiplos estágios para processar os dados, com o objetivo de construir um modelo de DL (SARKER, 2021).

Além disso, outra característica extremamente importante do DL é que as DNNs são muito mais influenciadas pela quantidade e características dos dados do que ANNs tradicionais. Por exemplo, no trabalho desenvolvido por Devanna *et al.* (2022), os autores realizaram o ajuste fino (do inglês *fine-tuning*) da rede neural profunda pré-treinada "DeepLabv3+" em três estágios, inicialmente com um conjunto de dados simples e escalonando a complexidade. No final, obtiveram três arquiteturas de redes neurais profundas. A primeira, que considerou uma pequena quantidade de dados, obteve uma precisão de apenas 48.10%, enquanto as duas próximas redes que consideraram uma quantidade maior de dados tiveram uma precisão de 96.87% e 93.3%.

2.3.1 Convolação

Um dos principais conceitos utilizados por DNNs em relação à imagens é a convolução. As camadas de convolução são responsáveis por realizar a detecção de padrões locais e estruturas em uma imagem de entrada. Através da utilização de diversas camadas de convolução conectadas em série, como observado na Figura 4, é possível extrair diferentes características de baixo nível da imagem, obtendo ao fim um mapa de características da mesma. A obtenção destas características é realizada através da utilização de kernels (matrizes) que identificam padrões específicos, como bordas, cor, textura ou outras características da imagem de entrada (WANG *et al.*, 2024; SHARMA *et al.*, 2024). As DNNs que utilizam diversas camadas convolucionais são chamadas de CNNs (do inglês *Convolutional Neural Network*).

Figura 4 – Arquitetura de uma rede neural com diversas camadas de convolução



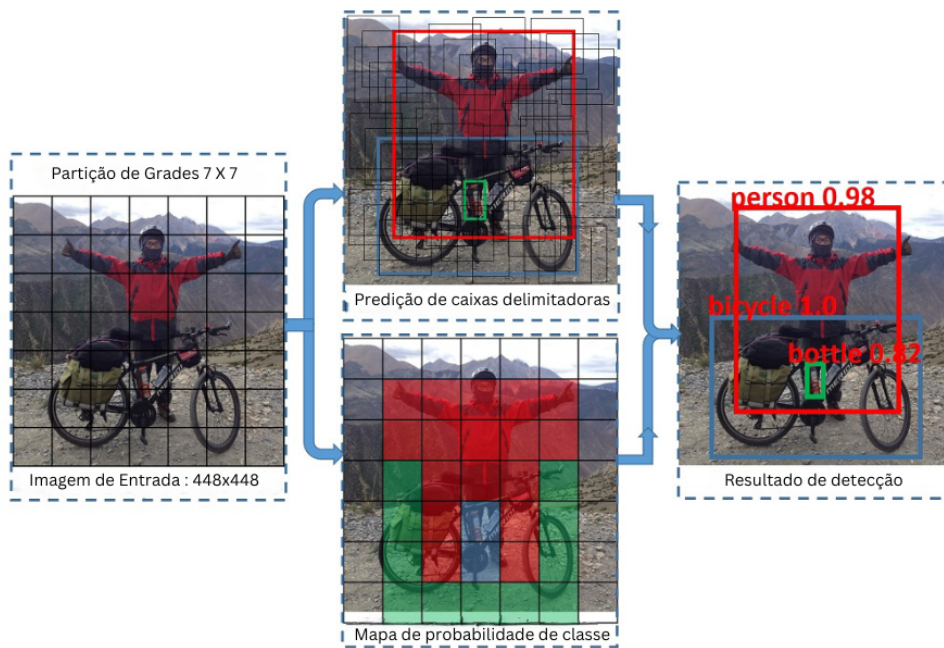
Fonte: Adaptado de Thapa *et al.* (2023)

2.3.2 Yolo (*You Only Look Once*)

Para a realização de segmentação e detecção de objetos neste trabalho, a rede neural profunda convolucional YOLO foi escolhida. A principal premissa da rede YOLO é, como seu próprio nome indica, a observação única de uma imagem de entrada. Isso é possível porque a rede divide a imagem de entrada em diversas grades quadriculares identificadas como células. Dessa maneira, cada uma das células destas grades é responsável pela detecção ou segmentação de objetos em sua grade, tornando-as independentes uma das outras (REDMON *et al.*, 2016).

Inicialmente, é realizado a divisão de uma imagem em uma grade com $S \times S$ células, onde S representa a quantidade de células. Cada uma destas células pode prever até “B” caixas delimitadoras (*bounding boxes*). Essas caixas indicam a localização de um objeto na imagem. As caixas também possuem uma pontuação de confiança, que indica a probabilidade de que a caixa realmente contenha um objeto. Além disso, cada célula prediz “C” probabilidades de classes. Isso significa que o YOLO pode identificar qual tipo de objeto está presente na imagem, com base em uma lista de “C” classes predefinidas. A Figura 5 demonstra o funcionamento do YOLO por etapas. Neste exemplo, foi utilizada uma grade de 7×7 células para realizar a detecção de objetos.

Figura 5 – Representação do funcionamento do YOLO



Fonte: Adaptado de Wu e Zhou (2019).

2.4 Vetores Densos

Vetores densos, também conhecidos como *embeddings*, têm se tornado cada vez mais populares em aplicações que envolvem AI. De maneira simples, é possível defini-los como uma forma de representar dados complexos, como textos, imagens e vídeos, de forma puramente numérica, compacta e eficiente. Através de *embeddings* preserva-se características essenciais e semânticas dos dados, ocupando um espaço significativamente menor comparado à sua forma original. A obtenção dos vetores densos pode ser realizada por meio da aplicação de DNNs. Estas redes utilizam múltiplas camadas de convolução sequencialmente para extrair as características mais importantes de uma imagem e condensá-las em um vetor de alta dimensionalidade, representando a imagem original (SUN *et al.*, 2023; GALLAGHER, 2023).

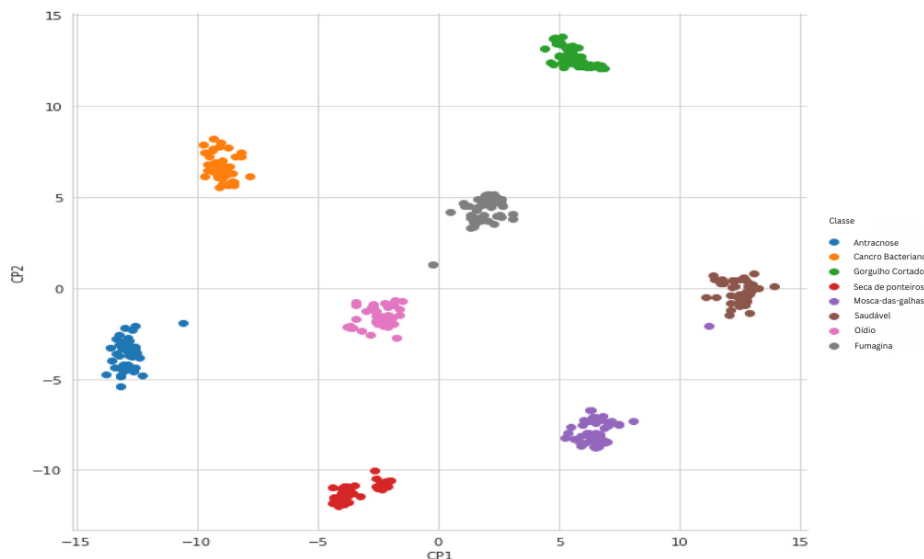
Uma característica crucial das ANNs é a exigência de uma vasta quantidade e diversidade de dados para alcançar resultados satisfatórios. No entanto, em certos casos, devido à natureza intrínseca do problema, é praticamente impossível obter uma quantidade significativa de dados. Isso é evidente na classificação de doenças em plantas

(ZABIHZADEH; MASOUDIFAR, 2023). Para tais problemas, os métodos tradicionais de ML podem não ser eficientes. A utilização de *embeddings* é altamente benéfica nessas situações, pois permite a obtenção das características semânticas individuais de cada planta. Assim, é viável compará-las umas com as outras, mesmo quando há poucas instâncias de dados disponíveis, facilitando a classificação do objeto em questão.

2.5 Banco de Dados Vetoriais

Bancos de dados vetoriais possibilitam o armazenamento de diversos vetores densos de mesma dimensionalidade em um mesmo ambiente. Esse tipo de abordagem tem como principal objetivo viabilizar o agrupamento de múltiplos *embeddings* semelhantes em determinadas áreas de um espaço vetorial. Dessa maneira, é viável realizar consultas extremamente rápidas para obter a similaridade entre um vetor denso recém-gerado e todos os vetores densos já armazenados em um banco de dados, permitindo o desenvolvimento de várias aplicações, como sistemas de recomendações, pesquisas semânticas e classificações de maneira rápida e eficiente (HAN; LIU; WANG, 2023).

Figura 6 – Gráfico da representando o algoritmo T-SNE em um grupo de vetores densos



Fonte: Adaptado de [Salamai \(2023\)](#).

A Figura 6 ilustra a distribuição em um espaço 2D de vetores densos de 8 tipos distintos de folhas de plantas doentes. A aplicação de algoritmos como o t-SNE (*t-distributed stochastic neighbor embedding*) permite visualizar as similaridades entre diferentes tipos de objetos de forma mais clara. Através desse tipo de análise, é possível observar características específicas do conjunto de dados utilizado e, conseqüentemente, escolher um algoritmo de busca de similaridade mais adequado a ser utilizado em conjunto ao banco de dados vetorial.

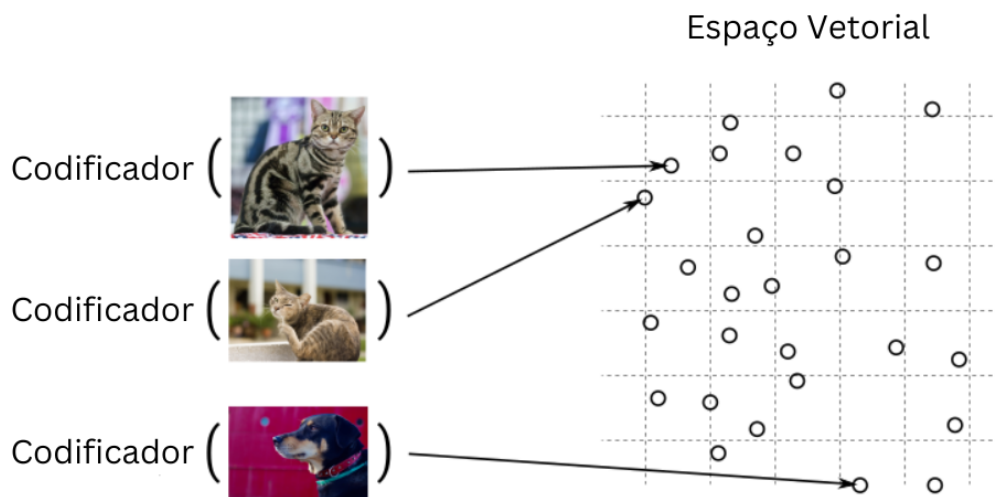
2.6 Busca aproximada

Uma ferramenta extremamente importante em banco de dados vetoriais é a busca aproximada. Essa técnica visa encontrar os vetores mais semelhantes a um dado vetor de entrada, em comparação a todos os vetores disponíveis em um banco de dados. Embora

existam diversos tipos de métricas utilizadas para calcular a similaridade entre vetores, a mais aplicada é a similaridade de cossenos, que utiliza o ângulo entre dois vetores para determinar o grau de similaridade entre eles (SIDOROV GRIGORI; GELBUKH, 2014).

A busca aproximada é uma técnica extremamente poderosa e amplamente utilizada, em grande parte devido à sua versatilidade e eficiência computacional. Através dela, é possível construir sistemas de recomendação, processamento de linguagem natural, busca de imagens, entre outros. Em Romisha e Vashist (2023), os autores propõem um novo método de busca de similaridade com o objetivo de ser utilizado para realizar a detecção de doenças em plantas através do processamento de imagens de plantas doentes e de cálculos de similaridades entre elas. A Figura 7 ilustra como os vetores densos são distribuídos no espaço vetorial. Neste caso, vetores com características similares possuem um alto grau de proximidade em comparação com *embeddings* de outras classes.

Figura 7 – Demonstração do espaço vetorial e similaridade entre vetores densos



Fonte: Elaborado pelo autor (2024)

3 Trabalhos Correlatos

Para a elaboração desta seção, realizou-se uma busca de artigos científicos nas seguintes bases: *Scopus*[®], *Web of Science*[®], *Science Direct*[®], *Springer Link*[®] e *IEEE Xplore*[®], utilizando a seguinte expressão de busca: “deep learning” AND “image segmentation” AND (“plant disease” OR “leaf disease” OR “plant pathology” OR “plant growth”). A Tabela 1 exibe a quantidade de artigos obtidos em cada uma das bases de dados consultadas.

Após a exclusão inicial, utilizando os critérios do Quadro 1, procedeu-se à análise dos títulos dos trabalhos, o que resultou na redução do número de 195 para um total de 86 estudos. Posteriormente, realizou-se uma segunda filtragem com base nos resumos de cada artigo, nas características específicas de cada trabalho, tais como seus objetivos gerais e os tipos de dados utilizados. Após essa segunda filtragem, um total de 6 artigos foram selecionados para a análise.

Subsequentemente ao início do desenvolvimento, constatou-se a necessidade de

Tabela 1 – Base de dados e números de artigos

Base de Dados	Número Original	Número após Critérios de Exclusão
Scopus [®]	188	88
Web of Science [®]	23	19
Science Direct [®]	18	16
Springer Link [®]	126	60
IEEE Explore [®]	113	12

Fonte: Elaborado pelo autor (2024)

Quadro 1 – Critérios de exclusão de estudos

Critério de Exclusão
Critério 1: Tipo de artigo Para a leitura, considerou-se apenas artigos publicados em periódicos.
Critério 2: Idioma Para a leitura, considerou-se apenas trabalhos na língua inglesa.
Critério 3: Data A busca nas bases de dados considerou apenas artigos publicados entre 2018 e o dia da realização das buscas, 16 de setembro de 2023.

Fonte: Elaborado pelo autor (2024)

empregar o conceito de *embeddings* para obter melhores resultados, tendo em vista as limitações impostas pelo conjunto de dados utilizado, e considerando que os estudos contidos na pesquisa inicial não abordaram técnicas que viabilizassem a melhoria dos resultados no cenário enfrentado neste trabalho. Nesse sentido, realizou-se uma nova busca por artigos científicos apenas na base de dados *Scopus*[®] utilizando a seguinte expressão: “deep learning” AND “embedding” AND (“plant disease” OR “leaf disease” OR “plant patholog” OR “plant growth”). A Tabela 2 exibe a quantidade de artigos obtidos para a base de dados Scopus antes e após a aplicação dos critérios do Quadro 1. Após a filtragem inicial, foi realizada a análise dos títulos e dos resumos dos trabalhos, reduzindo o número de artigos relevantes de 14 para 2.

Tabela 2 – Base de dados e números de artigos

Base de Dados	Número Original	Número após Critérios de Exclusão
Scopus [®]	31	14

Fonte: Elaborado pelo autor (2024)

No estudo de [Waldamichael, Debelee e Ayano \(2022\)](#), foi proposta uma metodologia para detectar quatro diferentes tipos de doenças nas folhas de café utilizando segmentação de imagens baseada em *Color Thresholding* e aprendizado por transferência com o modelo pré-treinado *MobileNetV2*. A abordagem apresentada consiste em tratamentos nas imagens das folhas, para então realizar a segmentação das imagens e, em seguida, essas informações são utilizadas pelo modelo *MobileNetV2* para produzir a inferência e determinar o tipo de doença. Os resultados demonstram que o processo de segmentação desenvolvido no trabalho superou outros métodos de estado da arte, como *YChCr* e *k-means algorithms*, considerando as métricas de *MIoU* (*Mean Intersection over Union*) com 72,13% e um *f1*

score com 82,54%. Além disso, o modelo de inferência *MobileNetV2* obteve uma precisão média de 96%.

No trabalho realizado por [Devanna et al. \(2022\)](#), os autores propuseram um sistema de identificação automática dos frutos de romã, a fim de monitorar a saúde e o crescimento dos frutos, além de realizar uma contagem da quantidade deles. Para isso, a coleta das imagens foi realizada com o auxílio de um robô equipado com sensores de distância a laser e a câmera de profundidade do modelo Intel RealSense D435. A abordagem consistiu na criação de três conjuntos de dados: um em que as fotografias dos frutos eram obtidas com um fundo branco e sem iluminação natural, outro com o mesmo fundo branco, contudo com iluminação natural e, por último, o conjunto obtido através do robô em campo. Tendo isso em vista, para realizar a segmentação, os autores utilizaram o método de aprendizado por transferência em vários estágios, onde a rede é treinada inicialmente no conjunto de dados com imagens mais simples e progressivamente chega ao conjunto final mais complexo. A rede escolhida para a segmentação foi a DeepLabv3+, obtendo um *F1 score* de 86.42% e um IoU de 97.94%.

[Uryasheva et al. \(2022\)](#) desenvolveram um sistema de detecção de folhas genérico, utilizando redes convolucionais em combinação com imagens multiespectrais. Para isso, foram coletadas mais de 360 mil imagens de folhas doentes e saudáveis de macieiras usando três diferentes câmeras: uma câmera no espectro visível, outra multiespectral contendo os espectros azul, verde, vermelho, borda vermelha e NIR (*Near Infra Red*), além de uma terceira câmera no espectro LWIR (*Long Wavelength Infrared*). Após a coleta das imagens, o modelo *Eff-UNet* foi escolhido para ser utilizado na realização da segmentação das folhas nas demais imagens. A escolha desse modelo ocorreu principalmente devido à sua combinação de características eficazes de duas arquiteturas de ponta, *EfficientNet* para extração de atributos e *U-Net* para criação de mapas de segmentação. O estudo apresentou resultados semelhantes a outros trabalhos recentes na área, alcançando um IoU médio de 0.72, enquanto em outros estudos variou entre 0.7 e 0.74.

Em [Afzaal et al. \(2021\)](#), os autores propõem a criação de um modelo para a segmentação de instâncias com o objetivo de detectar sete tipos diferentes de doenças em morangos. Um dos principais objetivos dos autores foi desenvolver um conjunto de dados contendo as sete classes de doenças de morangos, uma vez que tal conjunto não estava disponível publicamente. Além disso, eles se concentraram em propor um modelo de rede neural otimizado com base na arquitetura *MaskRCNN*. Também foram testadas várias técnicas de aumento de dados e sua influência na mAP (*Mean Average Precision*), que variou de 68,34% a 72,37%. Os resultados finais foram obtidos com duas arquiteturas diferentes para o *MaskRCNN*. A arquitetura ResNet50 obteve uma mAP de 81,37%, enquanto a ResNet101 alcançou uma mAP de 82,43%.

[Afonso et al. \(2020\)](#) teve como objetivo apresentar uma abordagem que viabiliza a detecção e contagem de frutos de tomate durante as suas fases de crescimento em uma *Greenhouse*. Além disso, os autores procuraram responder a duas questões durante o desenvolvimento do trabalho: é viável detectar frutos de tomate em ambientes práticos da vida real e é possível atingir resultados de nível de estado da arte na detecção de tomates utilizando a rede neural pré-treinada *MaskRCNN*. Para isso, inicialmente foram adquiridas imagens dos frutos de tomate em diferentes estágios de maturação utilizando câmeras Intel RealSense D435, e, posteriormente, utilizou-se o modelo de rede neural pré-treinado *MaskRCNN* para realizar a segmentação dos frutos. Além disso, os autores optaram por comparar várias arquiteturas diferentes do *MaskRCNN* e também testar diferentes

limiares IoU, que ocasionam variações nas métricas de cada arquitetura. Os resultados obtidos demonstram que a alteração do limiar afeta consideravelmente a precisão. Foram selecionados três níveis distintos: 0.75, 0.50 e 0.25. O limiar de 0.25 obteve os melhores resultados, alcançando uma precisão média na segmentação dos frutos de tomate maduros de 95% para a melhor arquitetura.

Em (ZABIHZADEH; MASOUDIFAR, 2023), os autores abordaram a detecção automática de doenças em plantas usando aprendizado profundo sem a necessidade de treinar uma rede neural para novas classes de doenças. A técnica utiliza redes neurais profundas pré-treinadas para extrair características das imagens de folhas. Com o uso de uma função de perda baseada em *proxies*, chamada *soft-triple loss*, o modelo consegue capturar variações detalhadas dentro de cada classe de folha doente. Além disso, o treinamento do modelo foi realizado no conjunto de dados *Plant Village*. Os resultados obtidos mostram que, apresentando poucas amostras de novas classes ao modelo, é possível atingir acurácias de 99% a 80.64% para a classificação de doenças.

Hepsağ (2023) teve como foco propor um método eficiente para identificar doenças em plantas usando *Few-Shot Learning*. O método proposto utiliza transferência de aprendizado, começando com um modelo pré-treinado no conjunto de dados ImageNet e refinando-o com o conjunto PlantCLEF2022 para extrair características específicas das plantas. Além disso, o autor optou pela substituição de CNNs por um classificador de máquinas de suporte (do inglês *Support Vector Machine - SVM*) para a classificação das doenças. Os experimentos foram realizados em dois conjuntos de dados, PlantVillage e PDD271. O método proposto alcançou uma precisão média de 88,4% com 10 exemplos e 75,5% com 5 exemplos no PlantVillage. No PDD271, que possui mais classes, o framework atingiu 56,3% com um único exemplo, 67,5% com 3 exemplos e 74,2% com 5 exemplos.

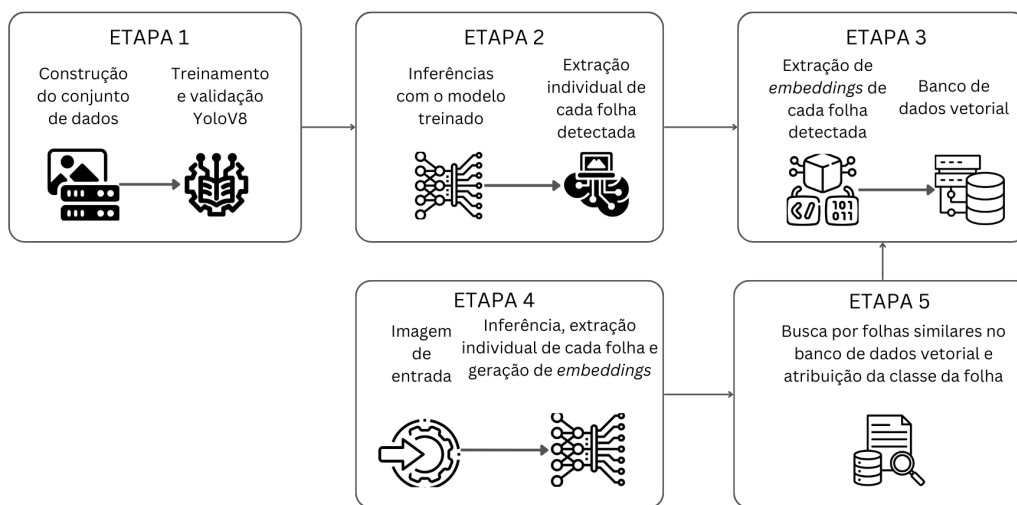
Por fim, Stavness (2018) propuseram a elaboração de um modelo de redes neurais convolucionais para a contagem de folhas de plantas do tipo roseta. A fim de alcançar esse objetivo, também abordaram a criação de um método que possibilita a geração de imagens 3D artificiais de folhas de plantas do tipo roseta, a fim de aumentar o conjunto de dados final. Para a realização da contagem, os autores elaboraram sua própria rede neural convolucional, com três camadas de convolução, cada uma com uma resolução espacial de 5x5, seguidas por uma camada de *Pooling*. Os resultados mostraram que a utilização da geração de imagens artificiais para o aumento de dados pode melhorar significativamente as métricas do modelo, reduzindo em aproximadamente 27% o erro médio absoluto na contagem das folhas.

Ao analisar os trabalhos relacionados, observa-se a tendência crescente da utilização de CNNs para a segmentação de folhas ou frutos de plantas, combinadas com técnicas de pré-processamento de imagens. De maneira similar, o método proposto também inclui o uso de uma CNN para segmentar folhas em uma imagem de entrada. Além disso, outra similaridade relevante entre o presente trabalho e os analisados é a utilização de modelos de redes neurais profundas pré-treinados para a extração de vetores densos, diferenciando-se na realização do ajuste fino por parte dos autores, o que não foi possível no presente trabalho devido à baixa quantidade de dados. Um fator de destaque do método proposto, que o diferencia dos trabalhos abordados nesta seção, é a utilização de um banco de dados vetorial em conjunto com o conceito de busca aproximada. Através de ambos os conceitos, é possível realizar a consulta de similaridade entre as folhas para que então possam ser classificadas como saudáveis ou doentes.

4 Método Proposto

O método proposto neste trabalho tem como principal fundamento a análise de folhas de bananeira em uma imagem de entrada com o objetivo de classificar cada folha individualmente como saudável ou não saudável, sem especificar o tipo de doença que a aflige. De forma ampla, o método proposto pode ser definido por um total de cinco etapas, conforme observado na Figura 8. A etapa inicial envolve a construção do conjunto de dados e o treinamento e validação de um modelo neural. Em seguida, ocorre a inferência por meio desse modelo nas imagens disponíveis no banco de dados, permitindo a extração individual de cada uma das folhas nelas contidas. A terceira etapa compreende a extração de vetores densos para cada uma dessas folhas detectadas e a criação do banco de dados vetorial com os *embeddings* extraídos. A quarta etapa consiste na utilização do sistema já construído para avaliar o método como um todo, abrangendo a inferência, a segmentação de cada folha e a extração individual de vetores densos. Por fim, a etapa final utiliza as informações obtidas de cada folha na etapa quatro para realizar consultas no banco de dados vetorial e, então, classificar cada uma das folhas como saudável ou doente. É importante ressaltar que as folhas armazenadas no banco de dados vetorial para as consultas foram classificadas como saudáveis ou doentes pelo autor do presente trabalho, sem o auxílio de especialistas

Figura 8 – Etapas realizadas para o método proposto



Fonte: Elaborado pelo autor (2024)

4.1 Etapa 1

4.1.1 Construção do conjunto de dados para treinamento

Embora já se tenha um conjunto de imagens, para que seja possível realizar o treinamento do modelo de rede neural profunda YOLOv8 (arquitetura utilizada neste trabalho), é necessário construir um conjunto de dados de forma que a rede possa interpretar. Com isso em mente, utilizou-se a plataforma Roboflow®, que possibilita a anotação eficiente e prática de itens em uma imagem, otimizando o processo e fornecendo guias intuitivos sobre como realizar o treinamento da rede. Foram utilizadas um total de 8 imagens para construir o conjunto inicial. Após a aplicação de técnicas de aumento de dados, o

número de imagens foi aumentado para um total de 14, das quais 9 foram reservadas para treinamento, 3 para validação e 2 para testes. As técnicas de aumento de dados aplicadas incluíram rotações de -15 e +15 graus, zoom mínimo de 0 e máximo de 20%, bem como rotações horizontais e verticais na imagem. No total, foram anotadas 1859 folhas através das 8 imagens originais.

4.1.2 Treinamento do modelo

Apesar de existirem diversos modelos de redes neurais de estado da arte disponíveis atualmente, os modelos da arquitetura YOLO têm ganhado grande visibilidade devido à sua arquitetura inovadora. Além disso, associada à grande facilidade de treinamento através da plataforma Ultralytics[®] e aos diversos guias disponíveis *online*, a rede YOLOv8, versão mais recente disponível quando este trabalho teve início, foi o modelo escolhido. O treinamento foi realizado utilizando a versão *YOLOv8x-seg*, que possui a maior quantidade de parâmetros, 71.8 milhões, comparado às outras versões disponíveis. O treinamento foi realizado por um total de 200 épocas, levando cerca de 22 minutos para a conclusão do treinamento até obter a convergência de resultado ótimo com o modelo treinado.

4.2 Etapa 2

4.2.1 Inferência com o modelo

Após a validação do modelo, torna-se viável a utilização do mesmo para realizar inferências em imagens, visando obter a segmentação de novas folhas. Na Figura 9, é apresentado o resultado da aplicação do modelo em uma imagem. Após a execução da inferência, o modelo retorna uma lista contendo todas as folhas detectadas, juntamente com a classe do objeto detectado, a confiança da inferência realizada e as coordenadas das caixas de detecção e dos polígonos referentes ao objeto identificado. Por meio da linguagem de programação Python[®], é possível converter essa lista em um arquivo com formato JSON. A Figura 10 ilustra como cada uma das folhas detectadas é representada como um objeto JSON.

4.2.2 Extração individual de cada folha

Utilizando a linguagem de programação Python[®], em conjunto com sua biblioteca CV2 e o arquivo JSON obtido na etapa anterior, realiza-se a extração individual de cada folha da imagem original. A Figura 11 apresenta uma folha extraída da sua imagem original. Embora uma grande parte da imagem seja apenas preta, ou seja, sem informação, a extração do vetor denso não considera essas seções da imagem, pois elas apresentam uma informação uniforme que não contribui significativamente para a característica semântica do vetor denso final. Após a extração, cada folha foi armazenada em uma pasta específica. Posteriormente, uma amostra foi selecionada arbitrariamente: 29 folhas extraídas e consideradas doentes e 66 consideradas saudáveis. É importante ressaltar que as folhas da amostra foram classificadas como doentes ou saudáveis por meio da análise manual do autor, sem a assistência de especialistas.

4.3 Etapa 3

4.3.1 Extração de vetores densos

Após a extração individual de cada imagem das folhas da imagem original, são criados vetores densos para cada uma das folhas da amostra selecionada. A biblioteca

Figura 9 – Resultado da inferência do modelo



Fonte: Elaborado pelo autor (2024)

Img2Vec foi utilizada para essa tarefa, fornecendo a opção de utilizar diversas arquiteturas de CNNs pré-treinadas para a obtenção dos vetores densos. Cada um desses *embeddings* é então associado individualmente à sua respectiva folha no arquivo JSON. A Tabela 3 apresenta um *embedding* criado a partir de uma imagem de folha extraída. É importante destacar que a criação de vetores densos é uma etapa fundamental do método. Ela é utilizada tanto para popular o banco de dados quanto para a classificação de novas folhas nunca antes vistas no sistema. Isso garante que a classificação entre folhas saudáveis e doentes tenha o menor viés possível.

Tabela 3 – Exemplo de um vetor denso de uma folha

0.32203	0.27724	0.78606	0.078616	0.056686
⋮	⋮	⋮	⋮	⋮
0.26870	0.17168	0.18564	0.09670	0.16346

Fonte: Elaborado pelo autor (2024)

4.3.2 Banco de dados vetorial

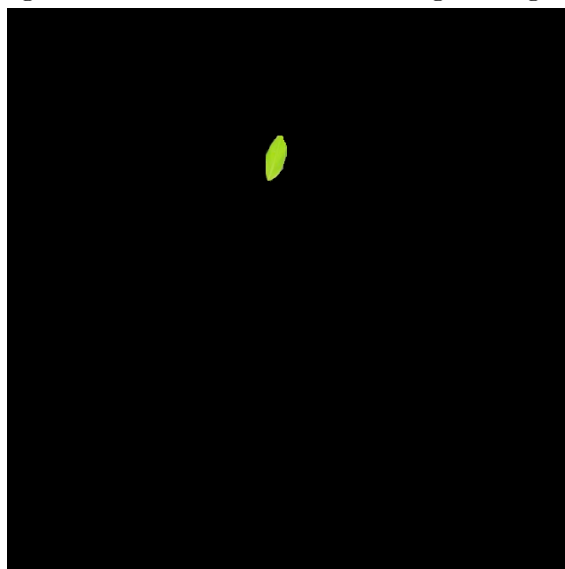
Em seguida, com a utilização das imagens pré-selecionadas na etapa anterior, é criada uma instância de banco de dados vetorial contendo um total de 95 *embeddings* de imagens, sendo 29 referentes a folhas consideradas doentes e as 66 restantes consideradas

Figura 10 – JSON obtido através da inferência

```
{
  "name": "folha",
  "class": 0,
  "confidence": 0.9353748559951782,
  "box": {
    "x1": 180.99575805664062,
    "y1": 474.26123046875,
    "x2": 220.21633911132812,
    "y2": 511.3416748046875
  },
  "segments": {
    "x": [185.0, 184.0, 184.0, 185.0, 185.0, 188.0, 188.0, 189.0, 190.0, 192.0, 192.0, 193.0, 193.0, 196.0, 196.0, 198.0, 213.0, 217.0, 218.0, 218.0, 219.0, 219.0, 218.0, 217.0, 215.0, 215.0, 214.0, 213.0, 211.0, 211.0, 210.0, 208.0, 207.0, 206.0, 204.0, 203.0, 202.0, 199.0, 198.0, 197.0, 196.0, 195.0, 194.0, 192.0, 191.0],
    "y": [476.0, 477.0, 492.0, 493.0, 494.0, 497.0, 498.0, 499.0, 499.0, 501.0, 503.0, 504.0, 505.0, 508.0, 509.0, 511.0, 511.0, 507.0, 507.0, 506.0, 505.0, 493.0, 492.0, 492.0, 490.0, 489.0, 488.0, 488.0, 486.0, 485.0, 484.0, 484.0, 483.0, 483.0, 481.0, 481.0, 480.0, 480.0, 479.0, 479.0, 478.0, 478.0, 477.0, 477.0, 476.0]
  }
}
```

Fonte: Elaborado pelo autor (2024)

Figura 11 – Folha extraída da imagem original



Fonte: Elaborado pelo autor (2024)

saudáveis. O código responsável pela instanciação do banco de dados e armazenamento dos vetores pode ser observado na Figura 12. O banco de dados vetorial Qdrant[®] foi escolhido para utilização neste trabalho, destacando-se sua fácil utilização, sem necessidade da instalação de qualquer *software*, apenas sua biblioteca em Python[®].

4.4 Etapa 4

Após a conclusão das etapas anteriores, uma nova imagem pode ser fornecida ao sistema. Utilizando um fluxo similar ao das etapas anteriores, a imagem é fornecida ao modelo de rede neural, que por sua vez realiza uma inferência para detectar folhas na imagem. Em seguida, um arquivo JSON é obtido. Através da leitura deste arquivo, cada folha da imagem original é extraída e vetores densos são criados para cada folha detectada na imagem.

Figura 12 – Código responsável pela instanciação do banco de dados e armazenamento dos vetores densos

```
1 from qdrant_client import QdrantClient
2 from qdrant_client.models import Distance, VectorParams, PointStruct
3
4 qdrant = QdrantClient(":memory:")
5
6 qdrant.recreate_collection(
7     collection_name="my_collection",
8     vectors_config=VectorParams(size=2048, distance=Distance.COSINE),
9 )
10
11
12 for index, file_info in enumerate(list_objetos):
13     vector = file_info['vector']
14     class_label = file_info['class']
15
16     qdrant.upsert(
17         collection_name="my_collection",
18         points=[
19             PointStruct(
20                 id=index,
21                 vector=vector,
22                 payload={"classe": class_label}
23             )
24         ]
25     )
26
```

Fonte: Elaborado pelo autor (2024)

4.5 Etapa 5

Nesta etapa, os *embeddings* obtidos na etapa anterior são utilizados para realizar consultas no banco de dados vetorial. Através desses vetores densos, são efetuadas consultas de similaridade individuais para cada folha no banco de dados. O objetivo é encontrar as folhas com características mais similares à folha que está sendo classificada. Para isso, cada consulta retorna um número escolhido de folhas similares, juntamente com o nível de similaridade e a classe da folha similar. Dessa forma, determina-se se a classe da folha é saudável ou doente, analisando qual classe é mais proeminente entre todas as folhas retornadas na consulta ao banco de dados. Após isso, é realizado o cálculo da porcentagem total de folhas saudáveis e de folhas doentes de uma imagem de entrada.

5 Resultados Experimentais

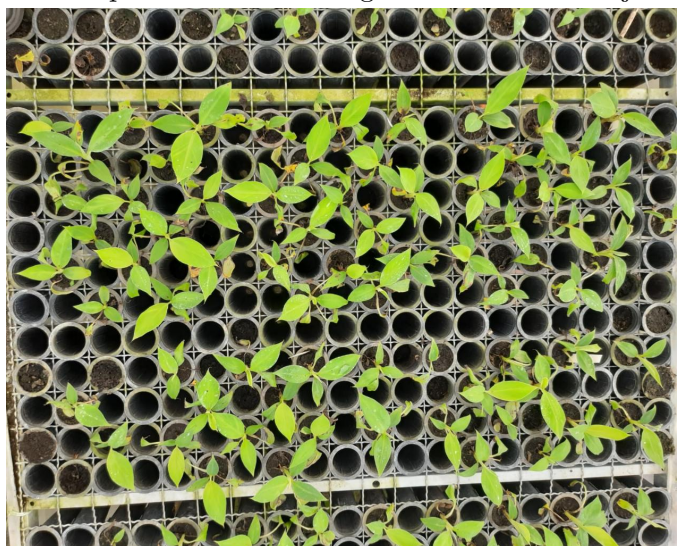
A seção atual tem como objetivo explorar os resultados obtidos através da implementação do método descrito na etapa anterior. Além disso, é realizada a discussão sobre as métricas utilizadas para a avaliação da qualidade do método proposto e também sobre o conjunto de dados utilizado para a realização do trabalho.

5.1 Cenário de estudo

O conjunto de dados utilizado para a realização deste trabalho foi obtido em parceria com o projeto “Desenvolvimento de estufas inteligentes e eficientes energeticamente para cultivo de plantas de alto valor agregado aplicadas a comunidades lindeiras ou assentadas de usinas hidrelétricas”. Neste *dataset* foram obtidas um total de 8 imagens RGB de folhas de bananeira em seus estágios iniciais de crescimento. Além disso, foram obtidas 3 imagens, as quais possuíam luminárias LED apontadas para as folhas, realizando a emissão de luzes na faixa de frequência azul e vermelha, afetando diretamente o espectro de cores, sendo então descartadas do trabalho. A escassez de imagens no cenário estudado é decorrente do longo intervalo de tempo entre a chegada de novas mudas de bananeira na estufa. Neste estudo, foram utilizadas imagens de mudas recém-chegadas e, ao longo do desenvolvimento do trabalho, não houve aquisição de novas mudas na estufa inteligente.

A Figura 13 é um exemplo de uma das imagens contida no conjunto de dados utilizado no trabalho. Ademais, uma característica importante deste conjunto de dados é a grande diferença entre as classes de folhas. Pode-se observar na Figura 13, que a quantidade de folhas saudáveis é muito maior do que a quantidade de folhas doentes, o que influencia diretamente a qualidade dos modelos de detecção de objetos, devido à discrepância entre as observações das classes. No total, ao longo das 78 imagens, foram observadas 41 folhas doentes.

Figura 13 – Exemplo de uma das imagens contidas no conjunto de dados



Fonte: Elaborado pelo autor (2024)

5.2 Materiais e Métodos

Para desenvolver o método, foi utilizado Python[®] como linguagem de programação. A escolha dessa linguagem se deve à vasta quantidade de bibliotecas disponíveis. No contexto deste trabalho, as principais bibliotecas utilizadas foram: cv2, que possibilita a utilização de algoritmos de visão computacional; numpy, para manipulação e criação de máscaras binárias para a extração de folhas individuais de cada imagem; json, para a leitura de arquivos contendo informações sobre os objetos detectados pelo modelo de rede neural; qdrant, que permite a integração com o banco de dados vetorial escolhido; e Img2Vec, uma biblioteca extremamente importante que possibilita a utilização de diversos modelos de rede neural pré-treinados para a extração de vetores densos de maneira simplificada.

Conforme mencionado anteriormente, o banco de dados vetorial escolhido para o método foi o Qdrant[®], permitindo o armazenamento de vetores de diferentes dimensões e a realização de consultas de busca aproximada de maneira eficiente.

Durante as épocas de treinamento de um modelo de rede neural, é crucial utilizar métricas para determinar se o modelo está ou não convergindo para um resultado satisfatório ao longo do tempo. Para isso, são empregados dados de validação, os quais não são diretamente utilizados durante o treinamento, mas apenas ao término de cada fase, não afetando diretamente o modelo. Após a conclusão do treinamento, o mesmo processo é aplicado a um conjunto de testes que também não foi previamente observado pela rede. Neste estudo, foram utilizadas diversas métricas para a avaliação do modelo que podem ser observadas nas Figuras 16, 17, 18. Contudo, vale destacar a métrica mais importante, o mAP (*Mean Average Precision*).

O mAP (*Mean Average Precision*) pode ser definido como a média do AP (*Average Precision*) ao longo de todas as classes previstas pelo modelo. O cálculo dessa métrica é realizado pela Equação 3, onde C representa o número total de classes e AP_c a AP para cada classe.

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C AP_c \quad (3)$$

A métrica AP é fundamental para o cálculo do mAP. Para calculá-la, utiliza-se cinco conceitos principais. O conceito de IoU (*Intersection over Union*), mede a sobreposição entre a caixa delimitadora (*bounding box*) predita pelo modelo e a caixa delimitadora verdadeira do objeto. O IoU é definido como a razão entre a área de interseção e a área de união das *bounding boxes*. A área de interseção é a região que se sobrepõe entre as duas caixas, enquanto a área de união é a soma das áreas individuais de cada caixa. A Figura 14 demonstra como é realizado o cálculo e exemplifica casos em que esta métrica é considerada como ruim ou excelente.

Outro conceito fundamental para o cálculo do AP é a matriz de confusão. Utilizando a matriz de confusão, é possível extrair diversas métricas relevantes das previsões feitas por um modelo de DNN. O Quadro 2 apresenta uma matriz de confusão genérica e os nomes de cada uma das variáveis que ela contém. Outros dois conceitos essenciais são *precision* (precisão) e *recall* (revocação). A métrica de precisão, derivada da Equação 4, indica a proporção de instâncias corretamente identificadas como positivas em relação a todas as instâncias previstas como positivas. A revocação, obtida a partir da Equação 5 visa indicar a proporção das instâncias positivas corretamente identificadas em relação ao valor total de instâncias que são realmente positivas. Embora a métrica de Acurácia, demonstrada na

Equação 6, não seja utilizada para o cálculo da AP, ela ainda possui grande importância, pois consegue medir a proporção de instâncias corretamente classificadas em relação ao total de instâncias.

Quadro 2 – Matriz de Confusão

Real Previsto	Previsto Positivo (P)	Previsto Negativo (N)
Real Positivo (P)	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Real Negativo (N)	Falso Positivo (FP)	Verdadeiro Negativo (VN)

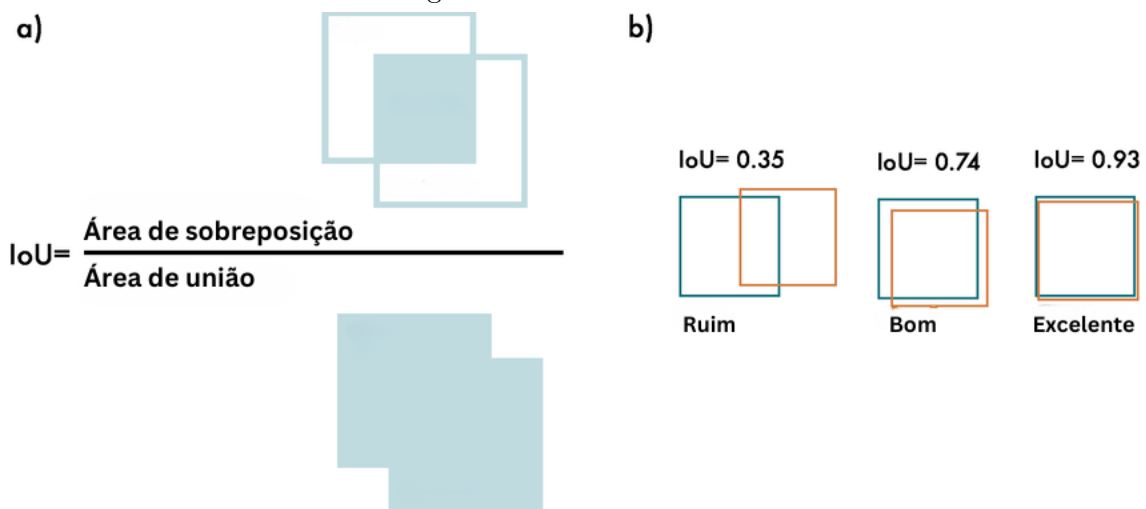
Fonte: Elaborado pelo autor (2024)

$$Precision \text{ (Precisão)} = \frac{VP}{VP + FP} \quad (4)$$

$$Recall \text{ (Revocação)} = \frac{VP}{VP + FN} \quad (5)$$

$$Accuracy \text{ (Acurácia)} = \frac{VP + VN}{VP + VN + FP + FN} \quad (6)$$

Figura 14 – Cálculo do IoU

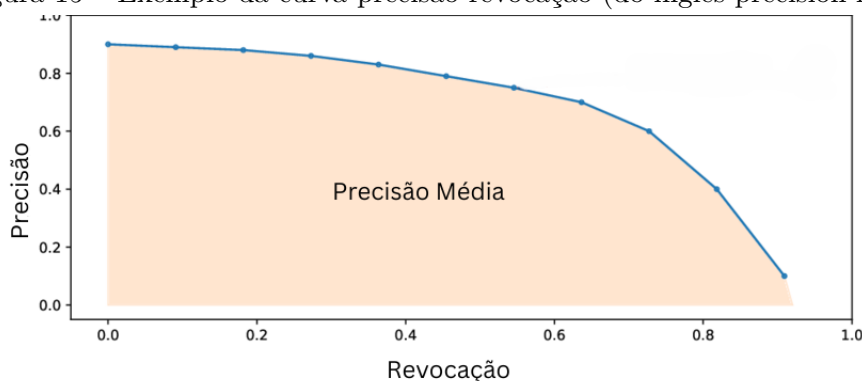


Fonte: Adaptado de Terven, Córdova-Esparza e Romero-González (2023).

Após a extração das métricas de precisão e revocação, é possível construir a curva *precision-recall* similar à exemplificada na Figura 15. A AP (*Average Precision*) é obtido calculando a área sob essa curva. A curva *precision-recall* representa a relação entre Precisão, a proporção de verdadeiros positivos entre os exemplos classificados como positivos e Revocação, a proporção de verdadeiros positivos entre todos os exemplos positivos reais, ao variar o limiar de decisão. A AP é uma métrica importante, especialmente em cenários com desequilíbrio de classes, pois um valor de AP maior indica melhor desempenho do modelo na identificação da classe positiva sem aumentar muitos falsos positivos, sintetizando o *trade-off* entre precisão e revocação ao longo de diferentes limiares.

Existem dois tipos principais de mAP utilizados para avaliar a qualidade de modelos de DNNs. O mAP50 calcula a média do AP para cada classe detectada, utilizando um IoU (Intersection over Union) de 0,5 como base. Já o mAP50-95 calcula a média do AP

Figura 15 – Exemplo da curva precisão-revocação (do inglês precision-recall)



Fonte: Adaptado de [Mendes-Neves, Meireles e Mendes-Moreira \(2023\)](#)

para cada classe, mas com um IoU variando de 0,5 a 0,95 em incrementos de 0,05. Embora ambas as métricas sejam válidas e sirvam como guias para a qualidade do modelo, o mAP50 é utilizado para medir a acurácia do modelo em cenários de predição considerados fáceis, enquanto o mAP50-95 oferece uma visão mais abrangente da acurácia do modelo, considerando todos os níveis de dificuldade de detecção.

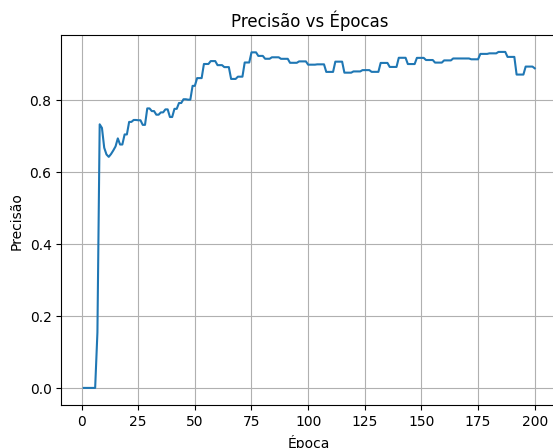
5.3 Apresentação dos resultados

Inicialmente, foi desenvolvido um método que consistia na aplicação da técnica de limiarização de cores para segmentar folhas em uma imagem específica, seguido pela utilização do modelo YOLOv8 para detecção de objetos, em vez de segmentação. Devido à natureza do conjunto de dados, ou seja, à grande discrepância entre a quantidade de folhas saudáveis e doentes, o modelo final apresentou resultados muito abaixo do esperado, com um mAP50 de 0,44, uma precisão de 41,3% e uma revocação de 39,9%. Levando isso em consideração, foi realizada a implementação do método proposto, apresentado na Seção 4, que produziu resultados satisfatórios.

Tendo em vista o método proposto, existem duas etapas que necessitam da análise dos resultados, a segmentação das imagens e, posteriormente, a classificação entre uma folha saudável ou não saudável. As Figuras 16, 17, 18 exibem, respectivamente, a progressão das métricas de precisão, revocação e de mAP50/mAP50-95 ao longo das épocas de treinamento do modelo YOLO. É possível observar que logo após as primeiras épocas, o modelo começa a convergir rapidamente, tendo poucas melhoras ao longo do tempo. Ao final do treinamento, o modelo obteve os valores de 0,94 e 0,73 para o mAP50 e mAP50-95, respectivamente, além de uma precisão (*Precision*) de 87% e uma revocação (*Recall*) de 89%. Essas estatísticas indicam que o modelo consegue detectar com precisão grande parte das folhas em uma imagem. No entanto, ele enfrenta alguma dificuldade com folhas mais difíceis de detectar, o que é esperado considerando o cenário em que várias folhas se sobrepõem, tornando a detecção de algumas instâncias da imagem extremamente difícil.

Em seguida, a fim de obter estatísticas relacionadas a precisão da classificação se uma folha está saudável ou não, foi necessário a criação de um *ground truth*, ou seja, uma informação que se sabe que é verdadeira através da observação manual. Para a criação da mesma foi selecionado um arquivo JSON de folhas extraídas e criado um novo campo chamado de "classe real". Então, após a criação deste campo comparou-se os resultados obtidos através da consulta no banco de dados vetorial versus os resultados reais.

Figura 16 – Precisão do modelo ao longo das épocas



Fonte: Elaborado pelo autor (2024)

Figura 17 – Revocação do modelo ao longo das épocas

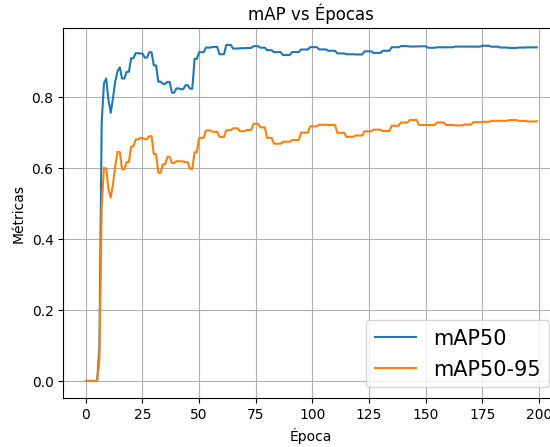


Fonte: Elaborado pelo autor (2024)

Através da matriz de confusão apresentada na Figura 19, é possível extrair métricas relacionadas à precisão da classificação do modelo. É importante ressaltar que, conforme explicado na subseção 4.5, essa matriz de confusão foi gerada a partir de consultas com um número similar de 20 folhas. Em outras palavras, cada folha foi classificada de acordo com a classe mais predominante entre as 20 folhas retornadas para cada consulta ao banco de dados vetorial. Embora a matriz de confusão exibida na Figura 19 demonstre os resultados apenas para um número de 20 folhas similares, a Tabela 6 aborda as mudanças das métricas ao variar o número de retorno.

Para o cenário com um número de retorno de 20 folhas similares, foi possível obter uma Acurácia de 91%, e precisões de 40% e 97% para as classes de folhas doentes e saudáveis, respectivamente. As Tabelas 4 e 5 mostram a comparação entre os dados reais e os dados obtidos pelo modelo de classificação. Ambas as tabelas revelam que o modelo tem uma leve tendência a classificar uma maior quantidade de folhas como doentes em vez de saudáveis, com uma margem de erro de 3.5% em relação à contagem real. Utilizando essas estatísticas, foram realizados ajustes no método, incluindo a alteração do modelo de rede neural pré-treinado utilizado para a criação dos vetores densos, bem como a modificação

Figura 18 – mAP do modelo ao longo das épocas



Fonte: Elaborado pelo autor (2024)

da quantidade de folhas similares que a consulta ao banco retornava.

Tabela 4 – Comparação da contagem de folhas saudáveis e doentes

Categoria	Contagem	
	Real	Previsto
Saudável	189	182
Doente	13	20

Fonte: Elaborado pelo autor (2024)

Tabela 5 – Comparação das porcentagens de folhas saudáveis e doentes

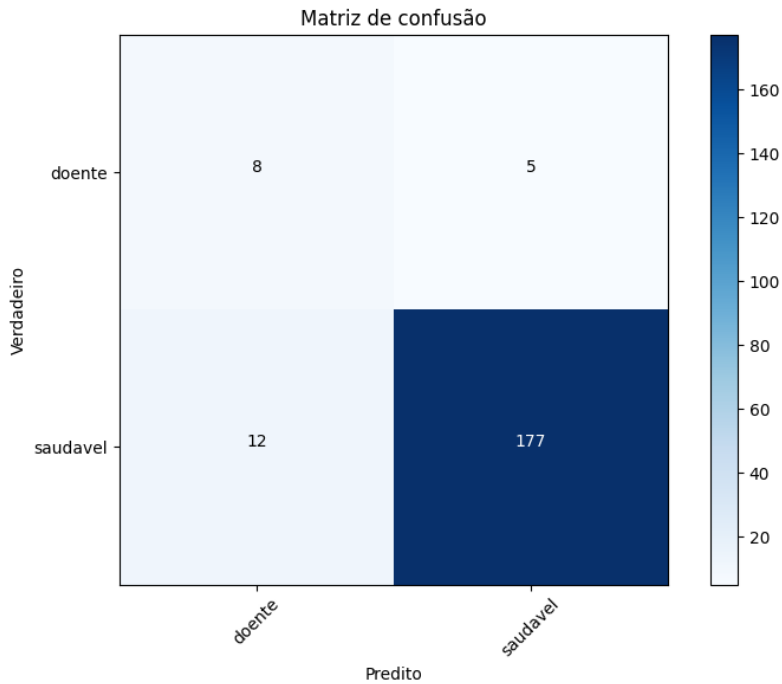
Categoria	Porcentagem	
	Real	Previsto
Saudável	93,6%	90,1%
Doente	6,4%	9,9%

Fonte: Elaborado pelo autor (2024)

Embora tenha sido realizado testes para a precisão da classificação da saúde de folhas utilizando todos os modelos disponíveis para a obtenção de vetores densos pela biblioteca *Img2Vec*, incluindo os modelos resnet18, resnet34, resnet50, resnet101 e alexnet, o modelo que obteve melhores resultados comparando diversos número de retornos K, ou seja, a quantidade de folhas similares retornadas para cada busca no banco de dados vetorial, foi o modelo resnet152. A Tabela 6 exibe os resultados obtidos considerando diferentes números de retorno na consulta ao banco de dados, onde os primeiros elementos de cada linha considerando as colunas Precisão, Revocação e *F1-Score* indicam estatísticas relacionadas a classificação de folhas doentes, e o segundo elemento referente a folhas saudáveis.

De modo geral, observa-se que o método tem uma acurácia geral muito boa. Todavia, embora a precisão para a classificação de folhas saudáveis seja alta, a precisão para folhas doentes é relativamente baixa, obtendo uma máxima de 40%. Resultado similar também ocorre nas demais estatísticas (revocação e *f1-score*). A fim de tentar obter melhores resultados o número de instâncias de imagens de folhas saudáveis e doentes no banco de dados vetorial também foi alterado.

Figura 19 – Matriz de confusão obtida de experimentos iniciais



Fonte : Elaborado pelo autor (2024)

Tabela 6 – Métricas de performance utilizando resnet152

Modelo	N. Retorno	Acurácia	Precisão	Revocação	<i>F1-Score</i>
resnet152	5	0.85	[0.28 0.99]	[0.85 0.85]	[0.42 0.91]
resnet152	10	0.91	[0.38 0.98]	[0.69 0.92]	[0.49 0.95]
resnet152	15	0.91	[0.37 0.97]	[0.54 0.94]	[0.44 0.95]
resnet152	20	0.92	[0.4 0.97]	[0.62 0.94]	[0.48 0.95]

Fonte: Elaborado pelo autor (2024)

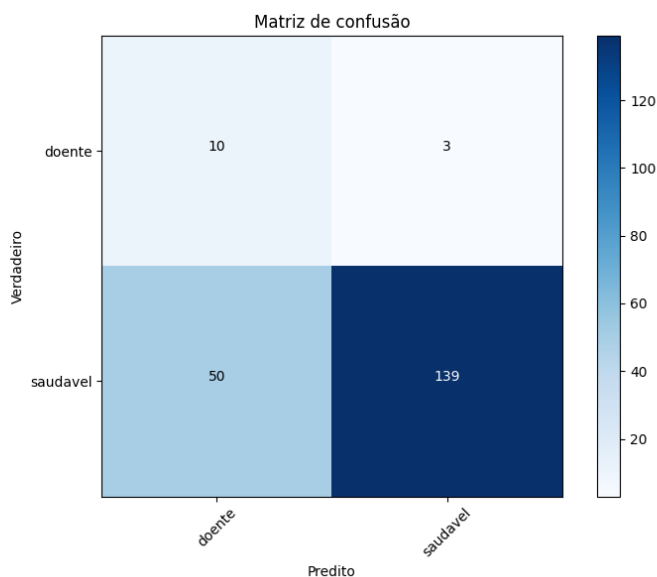
A quantidade de imagens utilizadas no banco de dados vetorial foi alterada de maneira arbitrária para um total de 82, sendo 32 imagens de folhas doentes e 50 de folhas saudáveis. A matriz de confusão obtida após esse ajuste pode ser observada na Figura 20. É possível observar que a diminuição na quantidade de folhas saudáveis disponíveis para consulta no banco de dados vetorial influencia diretamente nas estatísticas do modelo, com as novas estatísticas sendo 0.73% de acurácia, 0.16% para a precisão de classe doente e 97.8% para a classe saudável. As Tabelas 7 e 8 exibem a comparação entre os dados reais e as novas classes previstas.

De modo geral, é possível observar que a diminuição da representatividade no banco de dados da classe mais incidente no conjunto de dados de imagens, ou seja, as folhas saudáveis, torna o método mais suscetível à classificação de folhas saudáveis como doentes.

6 Considerações Finais e Trabalhos Futuros

O presente trabalho teve como objetivo a proposição de um método para a estimativa percentual de folhas de mudas de bananeira saudáveis e doentes em uma imagem.

Figura 20 – Matriz de confusão obtida após alteração de instâncias



Fonte : Elaborado pelo autor (2024)

Tabela 7 – Comparação da contagem de folhas saudáveis e doentes

Categoria	Contagem	
	Real	Previsto
Saudável	189	142
Doente	13	60

Fonte: Elaborado pelo autor (2024)

Tabela 8 – Comparação das porcentagens de folhas saudáveis e doentes

Categoria	Porcentagem	
	Real	Previsto
Saudável	93,6%	70.1%
Doente	6,4%	29.70%

Fonte: Elaborado pelo autor (2024)

Para tal, se utiliza de técnicas de Aprendizado Profundo para segmentação de imagens e a classificação do estado das folhas baseado no uso de vetores densos e buscas aproximadas.

Mesmo utilizando um conjunto de dados extremamente restrito, conforme explicado na subseção 5.1, o método proposto conseguiu obter resultados satisfatórios. Através da utilização do modelo YoloV8 para a segmentação individual das folhas, foi possível obter alta precisão na extração de cada folha. Conforme demonstrado na subseção 5.3, o modelo conseguiu segmentar de maneira eficiente a grande maioria das folhas, obtendo um mAP50 de 0,94, embora tenha enfrentado certa dificuldade em cenários onde diversas folhas se encontram sobrepostas.

A etapa de classificação entre folhas saudáveis e doentes obteve resultados inferiores em comparação à segmentação. As Tabelas 4 e 5 exibem os resultados obtidos utilizando o banco de dados vetorial para a classificação entre folhas doentes e saudáveis. Embora as acurácias variem entre 85% a 92%, é possível observar uma tendência em classificar uma quantidade maior de folhas como doentes do que a realidade, o que se deve, em grande

parte, à baixa quantidade de exemplos de folhas doentes em comparação às saudáveis.

A análise dos resultados revelou que a representatividade das classes no banco de dados vetorial influencia diretamente o desempenho do modelo. A diminuição da quantidade de folhas saudáveis no banco de dados levou a um aumento na suscetibilidade do método em classificar folhas saudáveis como doentes.

Embora o método proposto neste trabalho tenha atingido resultados que demonstrem potencial de aplicação prática, ainda que com limitações devido à baixa quantidade de dados, existe espaço para aprimoramentos. No método proposto, a construção dos vetores densos é realizada utilizando modelos de redes neurais genéricos, ou seja, treinados em conjuntos de dados que não possuem conhecimento específico sobre folhas de plantas. O desenvolvimento de um modelo de DNN especificamente treinado para este problema, possivelmente promoveria resultados mais satisfatórios. Esse tipo de modelo, associado à utilização de técnicas de *few-shot Learning*, pode melhorar a precisão do método de maneira significativa, uma vez que essa técnica visa treinar modelos eficientes com uma quantidade reduzida de dados, resultando na obtenção de vetores densos mais eficientes e semanticamente descritivos.

Referências

AABIDI, M. H. *et al.* A Model Proposal for Enhancing Leaf Disease Detection Using Convolutional Neural Networks (CNN): Case Study. *International journal of online and biomedical engineering*, v. 19, n. 12, p. 127 – 143, 2023. Type: Article. Disponível em: <<https://doi.org/10.3991/ijoe.v19i12.40329>>. Citado na página [7].

AFONSO, M. *et al.* Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning. *Frontiers in Plant Science*, v. 11, 2020. Disponível em: <<https://www.frontiersin.org/articles/10.3389/fpls.2020.571299>>. Citado (2) vezes nas páginas [10 e 18].

AFZAAL, U. *et al.* An instance segmentation model for strawberry diseases based on mask R-CNN. *Sensors*, v. 21, n. 19, 2021. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116023448&doi=10.3390%2fs21196565&partnerID=40&md5=d07704dfc94142f8ba265c9d6a2f4c72>>. Citado na página [18].

ARGÜESO, D. *et al.* Few-shot learning approach for plant disease classification using images taken in the field. *Computers and Electronics in Agriculture*, v. 175, p. 105542, 2020. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169920302544>>. Citado na página [7].

BASSIL, K. *et al.* Cancer health effects of pesticides: systematic review. *Canadian family physician Medecin de famille canadien*, College of Family Physicians of Canada, v. 53, n. 10, p. 1704–1711, Oct 2007. Citado na página [7].

BHUJEL, A. *et al.* Detection of gray mold disease and its severity on strawberry using deep learning networks. *Journal of Plant Diseases and Protection*, v. 129, n. 3, 2022. Disponível em: <<https://doi.org/10.1007/s41348-022-00578-8>>. Citado na página [10].

CHEN, Y.; PAN, J.; WU, Q. Apple leaf disease identification via improved CycleGAN and convolutional neural network. *Soft Computing*, v. 27, n. 14, p. 9773 – 9786, 2023. Type: Article. Disponível em: <<https://doi.org/10.1007/s00500-023-07811-y>>. Citado na página [7].

DEBELEE, T. G. *et al.* Evaluation of modified adaptive k-means segmentation algorithm. *Computational Visual Media*, v. 5, n. 4, p. 347–361, 2019. ISSN 2096-0662. Disponível em: <<https://doi.org/10.1007/s41095-019-0151-2>>. Citado na página [9].

DEVANNA, R. P. *et al.* In-Field Automatic Identification of Pomegranates Using a Farmer Robot. *Sensors*, v. 22, n. 15, 2022. Disponível em: <<https://www.mdpi.com/1424-8220/22/15/5821>>. Citado (4) vezes nas páginas [8, 9, 13 e 18].

DIJK, M. van *et al.* A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050. *Nature Food*, v. 2, n. 7, p. 494–501, jul. 2021. ISSN 2662-1355. Disponível em: <<https://doi.org/10.1038/s43016-021-00322-9>>. Citado na página [7].

FAO. *FAO's Plant Production and Protection Division*. Rome, Italy: FAO, 2021. 24 p. ISBN 978-92-5-137003-2. Disponível em: <<https://doi.org/10.4060/cc2447en>>. Citado na página [7].

FAO. *Tracking progress on food and agriculture-related SDG indicators 2023*. Rome, Italy: FAO, 2022. 215 p. ISBN 978-92-5-138013-0. Disponível em: <<https://doi.org/10.4060/cc7088en>>. Citado na página [7].

FENG, Y.; HE, K. A color image based fast segmentation method of crop leaf lesion. In: *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. [S.l.: s.n.], 2017. p. 116–121. Citado na página [10].

GALLAGHER, J. *What is an Image Embedding?* 2023. <RoboflowBlog>. Disponível em: <<https://blog.roboflow.com/what-is-an-image-embedding/>>. Citado na página [14].

GURESEN, E.; KAYAKUTLU, G. Definition of artificial neural networks with comparison to other networks. *Procedia Computer Science*, v. 3, p. 426–433, 2011. ISSN 1877-0509. World Conference on Information Technology. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050910004461>>. Citado na página [12].

HAN, Y.; LIU, C.; WANG, P. *A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge*. 2023. Disponível em: <<https://doi.org/10.48550/arXiv.2310.11703>>. Citado na página [15].

HEPSAĞ, P. U. Efficient plant disease identification using few-shot learning: a transfer learning approach. *Multimedia Tools and Applications*, 2023. Type: Article. Disponível em: <<https://doi.org/10.1007/s11042-023-17824-2>>. Citado na página [19].

MA, X.; CHEN, W.; XU, Y. ERCP-Net: a channel extension residual structure and adaptive channel attention mechanism for plant leaf disease classification network. *Scientific Reports*, v. 14, n. 1, p. 4221, fev. 2024. ISSN 2045-2322. Disponível em: <<https://doi.org/10.1038/s41598-024-54287-3>>. Citado na página [8].

MENDES-NEVES, T.; MEIRELES, L.; MENDES-MOREIRA, J. *A Survey of Advanced Computer Vision Techniques for Sports*. 2023. Citado na página [28].

- RAOF, R. *et al.* Color thresholding method for image segmentation algorithm of ziehl-neelsen sputum slide images. In: *2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control*. [S.l.: s.n.], 2008. p. 212–217. Citado na página [11].
- REDMON, J. *et al.* *You Only Look Once: Unified, Real-Time Object Detection*. 2016. Citado na página [13].
- ROMISHA, R.; VASHIST, S. Detection of plant diseases using image-based similarity measures of pythagorean fuzzy sets. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 32, p. 244, 10 2023. Citado na página [16].
- SALAMAI, A. A. Enhancing mango disease diagnosis through eco-informatics: A deep learning approach. *Ecological Informatics*, v. 77, 2023. Type: Article. Disponível em: <<https://doi.org/10.1016/j.ecoinf.2023.102216>>. Citado na página [15].
- SANKARAN, S. *et al.* A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, v. 72, n. 1, p. 1–13, 2010. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169910000438>>. Citado na página [7].
- SARKER, I. H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, v. 2, n. 6, p. 420, ago. 2021. ISSN 2661-8907. Disponível em: <<https://doi.org/10.1007/s42979-021-00815-1>>. Citado na página [12].
- SETIAWAN, B. *et al.* Identification of CNN hyper-parameters for tobacco leaf quality classification on Nvidia Jetson Nano. *Eastern-European Journal of Enterprise Technologies*, v. 6, n. 2 (126), p. 17–24, 2023. Disponível em: <<https://doi.org/10.15587/1729-4061.2023.289017>>. Citado na página [12].
- SHARMA, P. *et al.* A survey on cancer detection via convolutional neural networks: Current challenges and future directions. *Neural Networks*, v. 169, p. 637–659, 2024. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0893608023006287>>. Citado na página [13].
- SHEIKH, H.; PRINS, C.; SCHRIJVERS, E. Artificial intelligence: Definition and background. In: _____. *Mission AI: The New System Technology*. Cham: Springer International Publishing, 2023. p. 15–41. ISBN 978-3-031-21448-6. Disponível em: <https://doi.org/10.1007/978-3-031-21448-6_2>. Citado na página [11].
- SIDOROV GRIGORI; GELBUKH, A. G.-A. H. P. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, v. 18, n. 3, p. 2043, 2014. Disponível em: <<https://doi.org/10.13053/CyS-18-3-2043>>. Citado na página [16].
- STAVNESS, J. U. C. P. The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant Methods*, v. 14, n. 1, 2018. Disponível em: <<http://link.springer.com/article/10.1186/s13007-018-0273-z>>. Citado na página [19].
- SUN, J. *et al.* Few-shot learning for plant disease recognition: A review. *Agronomy Journal*, 2023. Type: Article. Disponível em: <<https://doi.org/10.1002/agj2.21285>>. Citado na página [14].

TERVEN, J.; CÓRDOVA-ESPARZA, D.-M.; ROMERO-GONZÁLEZ, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, MDPI AG, v. 5, n. 4, p. 1680–1716, nov. 2023. ISSN 2504-4990. Disponível em: <<http://dx.doi.org/10.3390/make5040083>>. Citado na página [27].

THAPA, A. *et al.* Deep learning for remote sensing image scene classification: A review and meta-analysis. *Remote Sensing*, v. 15, n. 19, 2023. ISSN 2072-4292. Disponível em: <<https://www.mdpi.com/2072-4292/15/19/4804>>. Citado na página [13].

URYASHEVA, A. *et al.* Computer vision-based platform for apple leaves segmentation in field conditions to support digital phenotyping. v. 201, 2022. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169922005828>>. Citado (2) vezes nas páginas [9 e 18].

WALDAMICHAEL, F. G.; DEBELEE, T. G.; AYANO, Y. M. Coffee disease detection using a robust HSV color-based segmentation and transfer learning for use on smartphones. *International Journal of Intelligent Systems*, v. 37, n. 8, 2022. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85125623164&doi=10.1002%2fint.22747&partnerID=40&md5=1ed44fc3e8cc6305bee08842c17934f3>>. Citado (4) vezes nas páginas [9, 10, 11 e 17].

WANG, Y. *et al.* Deep leaning in food safety and authenticity detection: An integrative review and future prospects. *Trends in Food Science and Technology*, v. 146, 2024. Disponível em: <<https://doi.org/10.1016/j.tifs.2024.104396>>. Citado na página [13].

WU, Q.; ZHOU, Y. Real-time object detection based on unmanned aerial vehicle. In: *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*. [s.n.], 2019. p. 574–579. Disponível em: <<https://doi.org/10.1109/DDCLS.2019.8908984>>. Citado na página [14].

ZABIHZADEH, D.; MASOUDIFAR, M. ZS-DML: Zero-Shot Deep Metric Learning approach for plant leaf disease classification. *Multimedia Tools and Applications*, 2023. Type: Article. Disponível em: <<https://doi.org/10.1007/s11042-023-17136-5>>. Citado (2) vezes nas páginas [15 e 19].

ZHANG, Z. *et al.* Enhancing Wheat Disease Diagnosis in a Greenhouse Using Image Deep Features and Parallel Feature Fusion. *Frontiers in Plant Science*, v. 13, 2022. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127415476&doi=10.3389%2ffpls.2022.834447&partnerID=40&md5=1f757ded5357e3683dcd4492cb2e2d53>>. Citado (2) vezes nas páginas [9 e 11].