APLICAÇÃO DE VISÃO COMPUTACIONAL E DEEP LEARNING NA DETECÇÃO PRECOCE DE DOENÇAS EM MELÃO

Matheus de Oliveira Dias
Universidade Federal Rural do Semi-Árido - UFERSA
Programa de Pós-Graduação em Engenharia Elétrica - PPGEE
Mossoró/RN, Brasil
matheus.dias@alunos.ufersa.edu.br

Idalmir de Souza Queiroz Júnior
Universidade Federal Rural do Semi-Árido - UFERSA
Programa de Pós-Graduação em Engenharia Elétrica - PPGEE
Mossoró/RN, Brasil
idalmir@ufersa.edu.br

Glauber Henrique de Sousa Nunes
Universidade Federal Rural do Semi-Árido - UFERSA
Programa de Pós-Graduação em Fitotecnia - PPGFITO
Mossoró/RN, Brasil
glauber@ufersa.edu.br

***Abstract*— A detecção precoce de doenças em plantas é essencial para a produtividade agrícola e a redução de perdas na plantação. Este estudo propõe a utilização de visão computacional e aprendizado profundo para identificar automaticamente doenças em folhas de melão. O modelo *YOLO* (*You Only Look Once*) é empregado devido à sua eficiência na detecção de objetos em tempo real. O conjunto de dados utilizado inclui imagens de folhas saudáveis e doentes, permitindo o treinamento de um modelo capaz de classificar e localizar regiões afetadas com alta precisão. O pré-processamento das imagens envolve técnicas como redimensionamento, filtragem e segmentação baseada no algoritmo *K-means*, garantindo uma melhor extração de características. Para aprimorar o desempenho da rede neural convolucional, são aplicadas estratégias de otimização, como poda de canais e fine-tuning, reduzindo a complexidade do modelo sem comprometer sua precisão. Além disso, a avaliação do modelo é realizada por meio de métricas como precisão, lembrança, *mean Average Precision* (*mAP*) e tempo de inferência, garantindo um equilíbrio entre eficiência e desempenho. A aplicação dessa tecnologia pode auxiliar agricultores na identificação rápida e precisa de doenças, permitindo intervenções mais eficazes e minimizando o uso excessivo de defensivos agrícolas. Conclui-se que a combinação de redes neurais convolucionais com técnicas de otimização representa um avanço significativo para a agricultura de precisão. Pesquisas futuras podem explorar a adaptação do modelo para diferentes culturas e condições ambientais, bem como sua integração com drones e sensores *IoT* para monitoramento contínuo das culturas.**

Keywords*—Visão Computacional, YOLO, Detecção de Doenças, Redes Neurais Convolucionais, Processamento de Imagens

I. INTRODUÇÃO

O Brasil é um grande produtor e exportador agrícola no mundo. Com sua vasta dimensão territorial e diferentes climas, há grande diversidade de culturas agrícolas por todo o país, sendo que diferentes regiões se adequam melhor a determinados cultivos [1]. Uma cultura que vem crescendo nos últimos anos é a produção de melão. Em 2023, o Brasil produziu aproximadamente 862.387 toneladas desse fruto [2]. A região Nordeste concentra cerca de 96% da produção nacional, com o Rio Grande do Norte sendo o maior produtor, responsável por 65% dessa produção e arrecadando cerca de R$ 823.025.000 [3]. A concentração da produção de melão na região noroeste do estado se deve às condições climáticas favoráveis, como o clima semiárido, que favorece frutos com maior teor de açúcar, mais sabor e maior consistência, tornando-os ideais para comercialização [4].

No entanto, várias doenças causadas por fungos, bactérias, nematoides e vírus podem afetar essa cultura. Se não forem tratadas adequadamente e de forma rápida, essas enfermidades podem reduzir tanto a produtividade quanto a qualidade dos frutos, resultando em grandes perdas para os agricultores [5]. Portanto, diagnosticar doenças no melão de maneira eficaz é essencial para implementar estratégias de controle apropriadas [6].

Neste contexto, a aplicação de técnicas de visão computacional pode agilizar o processo de detecção de doenças no melão e evitar tais problemas, considerando que, atualmente, a identificação ainda é feita por observação humana. Essa abordagem propõe a utilização do modelo chamado *YOLO* (*You Only Look Once*), um modelo de aprendizado profundo eficiente para detecção de objetos em tempo real [7]. A implementação da *YOLO* permite a análise automatizada de imagens para encontrar de maneira mais rápida plantas doentes em meio a grandes plantações. Com isso, torna-se possível identificar precocemente o nível de infestação nas plantas, possibilitando a tomada de medidas preventivas antes que as doenças se espalhem [8].

Este artigo explora a aplicação da biblioteca *YOLO* em Python para identificar doenças nas folhas do melão, discutindo suas vantagens, desafios e perspectivas futuras para aprimorar a agricultura de precisão.

II. REFERENCIAL TEÓRICO

A. Visão Computacional

A visão computacional é um campo da inteligência artificial que usa aprendizado de máquina e redes neurais para ensinar computadores e sistemas a extrair informações significativas de imagens digitais, vídeos e outras entradas visuais e para fazer recomendações ou tomar medidas quando defeitos ou problemas são identificados [9].

Para permitir que as máquinas vejam, analisem e compreendam o que estão observando, são utilizados algoritmos de visão computacional e aprendizado de máquina, que processam imagens e extraem informações relevantes para tomada de decisões em diferentes aplicações. A visão computacional funciona de maneira semelhante à visão humana, mas os humanos se beneficia de anos de experiência e contexto, o que nos ajuda a identificar objetos, avaliar distâncias, detectar movimentos ou perceber se algo está errado em uma imagem [10]. Por outro lado, treina máquinas

para realizar essas tarefas de forma mais rápida, utilizando câmeras, dados e algoritmos, em vez de retinas, nervos ópticos e o córtex visual. Um sistema treinado para inspecionar produtos ou monitorar um processo produtivo pode analisar milhares de itens ou operações por minuto, identificando defeitos ou problemas que muitas vezes passam despercebidos pelos humanos, superando rapidamente nossas capacidades [11].

B. Deep Learning

O *deep learning* é um subconjunto do aprendizado de máquina que usa redes neurais de várias camadas, chamadas de redes neurais profundas, para simular o complexo poder de tomada de decisão do cérebro humano. Alguma forma de *deep learning* alimenta a maioria das aplicações de inteligência artificial (IA) [9].

A principal diferença entre *deep learning* e aprendizado de máquina está na arquitetura da rede neural utilizada. Os modelos tradicionais de aprendizado de máquina, conhecidos como "não profundos", utilizam redes neurais simples, com uma ou duas camadas computacionais. Já os modelos de *deep learning* empregam redes com três ou mais camadas, frequentemente chegando a centenas ou até milhares, para realizar o treinamento [12].

Enquanto os modelos de aprendizado supervisionado requerem dados de entrada estruturados e rotulados para gerar resultados precisos, os modelos de *deep learning* podem operar com aprendizado não supervisionado. Com essa abordagem, os modelos de *deep learning* conseguem identificar características, particularidades e relações a partir de dados brutos e não estruturados, sem a necessidade de rótulos definidos. Além disso, esses modelos têm a capacidade de avaliar e ajustar seus resultados para melhorar sua precisão de forma contínua [13].

C. Redes Neurais Convolucionais (CNN)

Um dos modelos de aprendizado são as *convolutional neural networks* (*CNNs*), traduzindo para o português, redes neurais convolutionais, sendo o principal modelo utilizado na visão computacional e na classificação de imagens, permitindo a detecção de padrões em fotos e vídeos para reconhecimento de objetos e rostos [7]. Compostas por camadas convolucionais, de agrupamento e totalmente conectadas, essas redes podem conter milhares de camadas para capturar padrões detalhados e estruturais [8]. Sua eficiência no processamento de imagens supera métodos tradicionais de extração de características, otimizando a troca de informações entre camadas, embora haja perda de dados nas camadas de agrupamento, mitigada pela redução da complexidade e do risco [6].

Entretanto, *CNNs* demandam alto poder computacional, exigindo múltiplas *GPUs* e recursos significativos para treinamento. Além disso, sua implementação e ajuste requerem especialistas altamente qualificados para alcançar um desempenho ideal [14]. Mas para contornar esses problemas existem algumas bibliotecas que facilitam no trabalho, como o caso da biblioteca *YOLO* no reconhecimento de imagens.

D. Biblioteca YOLO

O modelo chamado *You Only Look Once*, traduzindo para o português, Você Olha Apenas Uma Vez, ou mais conhecido apenas por *YOLO*, é um modelo de aprendizado profundo projetado para a detecção de objetos em tempo real. Diferentemente de outros modelos de detecção que analisam imagens em múltiplas etapas, o *YOLO* processa a imagem inteira em uma única etapa, tornando a inferência mais rápida e eficiente [7].

Esse algoritmo de detecção de ponta a ponta aumentou significativamente a velocidade de detecção. A série *YOLO* evoluiu do *YOLOv1* ao *YOLOv5* à medida que novos métodos foram desenvolvidos. O *YOLOv5* é o modelo do algoritmo que hoje melhor funciona e alcança excelentes resultados em diversos projetos de detecção de objetos em tempo real [8], assim apresentando as melhores características para o objetivo do trabalho, que precisa de analisar imagens de plantas capturadas em movimento.

O modelo *YOLOv5* é composto por quatro partes: entrada, *backbone* (estrutura principal), *neck* (pescoço) e previsão. Para a parte de entrada, além do método de aumento de dados *Mosaic* introduzido no *YOLOv4*, são incorporadas computações de auto âncora e dimensionamento adaptativo de imagens, permitindo maior flexibilidade no tamanho das imagens de entrada e aumentando a robustez da rede [14]. Tendo como suas principais versões:

- *YOLOv3*: Popular por seu equilíbrio entre velocidade e precisão [7];
- *YOLOv4*: Melhor otimização para precisão em tempo real [8];
- *YOLOv5*: Implementação otimizada em *PyTorch* para fácil treinamento e ajuste fino [14].

III. MATERIAIS E MÉTODOS

Para treinar um modelo *YOLO* para identificar doenças na folha, primeiro faz toda a coleta de dados:

- Coleta de imagens com uso de câmera *RGB* acoplada em drone, para sobrevoar a plantação de forma eficiente e rápida, capturando o maior número de imagens possíveis de folhas saudáveis e infectadas [6]. Assim essas imagens são divididas em duas categorias principais: imagens de folhas saudáveis e imagens de folhas doentes. O principal objetivo desse processo é armazenar as imagens de forma estruturada para análise. Todas as imagens utilizadas no conjunto de dados devem estar no formato *JPEG* [17];
- O pré-processamento é uma etapa fundamental para executar tarefas como recorte, filtragem, redimensionamento de imagens, realce de bordas e operações morfológicas. Nesse contexto, o principal objetivo do pré-processamento é ajustar o tamanho das imagens das folhas de melão, visto que, inicialmente, essas imagens possuem alta resolução, o que pode aumentar significativamente o tempo de processamento. Além disso, ruídos presentes nas imagens podem comprometer a segmentação e a extração de características, sendo necessário removê-los por meio de técnicas de filtragem apropriadas. Durante essa fase, também ocorre a conversão das imagens do formato *RGB* para valores binários [17].
- A segmentação é utilizada para dividir imagens digitais de folhas em objetos específicos ou áreas desejadas nas folhas das plantas. O principal objetivo desse processo é identificar e quantificar as regiões da imagem que apresentam indícios de doenças. A

segmentação é finalizada apenas quando o objeto de interesse no sistema é completamente isolado. Uma das técnicas empregadas nesse processo é a segmentação por agrupamento *K-means*, que permite dividir as imagens em diferentes grupos, separando as áreas afetadas. Com esse método, a parte desejada da imagem é preservada, enquanto as regiões irrelevantes são removidas [17].

- Rotulagem e classificação das imagens utilizando filtros que melhor se adequa para marcar regiões afetadas [14]. Fazendo extração das características da imagem que auxiliam no processo de classificação com precisão. Elementos como cor, textura, correlação e homogeneidade são algumas das características utilizadas na análise. Esse procedimento é essencial para acelerar tarefas como correspondência e recuperação de imagens. Diversos métodos são empregados para a classificação de imagens, incluindo diferentes estratégias de análise de componentes principais. Para a operação de classificação, é utilizada a rede neural recorrente empilhada (*Stacked RNN*), que categoriza as imagens processadas em duas condições: folhas saudáveis e não saudáveis. Assim no caso das folhas não saudáveis, a área afetada pela doença é identificada e extraída [17].
- Divisão do *dataset*, fazendo a separação em conjuntos de treinamento, validação e teste [5].

A. *Modelo do Filtro PYSS*

Modelos de redes neurais convolucionais mais complexos e maiores podem proporcionar melhores resultados em detecção, mas são difíceis de serem implementados em hardware com recursos limitados. Portanto, é necessário diminuir o número de parâmetros e o uso de recursos computacionais, sem comprometer o desempenho do modelo. Para isso, utiliza-se a otimização dos pesos dos canais. Isso permite manter camadas importantes para a extração de características, ajustando a taxa de espaçamento. Além disso, um ajuste fino é realizado para compensar a perda de precisão causada pela compressão do modelo [15]. Como mostra a Figura 1.

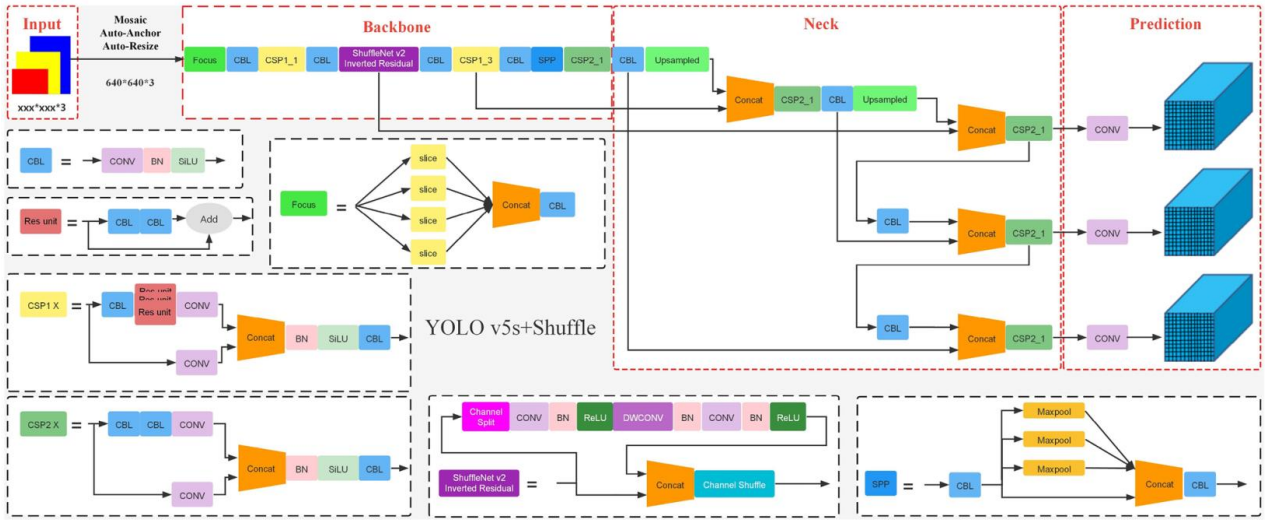

Fig. 1. Estrutura do *YOLO v5s+Shuffle* [16].

Para a detecção em tempo real de doenças nas folhas de melão será utilizado o modelo de filtro para esse reconhecimento de imagens é o chamado *Python Screen Shots* (*PYSS*), que é o refinamento com melhoras para a função proposta neste artigo [16]. Após receber todos os dados, para a melhora do modelo é feito o treinamento, a poda e o refinamento. Esses três processos de treinamento da melhora do modelo serão repetidos várias vezes para obter o modelo podado com menor perda de precisão. Além disso, o desenvolvimento do modelo inclui a validação e depuração do modelo no servidor e no *Jetson Nano* [16].

No modelo *PYSS*, é empregada uma método para alcançar a espaçamento na camada de Normalização de Batch (*BN*) incorporando uma restrição de regularidade *L1* no fator de escala γ do método de detecção *YOLO v5+Shuffle* [16]. Essa adição modifica a função de perda ao introduzir γ como um fator de escala em cada canal. Assim a função de perda geral utilizada no algoritmo de poda de canais consiste em dois componentes principais:

$$Loss = \sum l(f(x, W), y) + \lambda \sum g(\gamma) \quad (1)$$

A perda de treinamento padrão relacionada à rede, representada como W (os pesos treináveis), e um segundo termo restringido por um coeficiente equilibrado λ. A função $g()$ atua como uma penalidade para abordar o coeficiente de razão resultante do processo de poda, definido como $g(s) = |s|$, com os valores de γ tipicamente escolhidos na faixa de $10-1$ a $10-5$. A camada *BN* depende de estatísticas de mini partes para normalizar as ativações internas, designando z_{in} e z_{out} como a entrada e a saída da camada *BN*, respectivamente, enquanto B denota o mini partes atual.

$$\hat{Z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2)$$

$$z_{out} = \gamma \hat{Z} + \beta \quad (3)$$

A média μ_B e o desvio padrão σ_B da entrada para a função de ativação são críticos para esse processo. Os parâmetros γ e β são parâmetros de transformação afim treináveis que permitem que as ativações normalizadas sejam escaladas e deslocadas de acordo. Além disso, um pequeno valor de ϵ, definido como o tamanho do canal, contribui para minimizar a ativação de saída z_{out}, levando, em última análise, a um modelo mais leve [16].

Após o treinamento esparso, muitos canais do modelo apresentam fatores de escala próximos de zero. Esses canais são eliminados removendo os pesos correspondentes em todas as suas conexões de entrada e saída, tornando o modelo de detecção de objetos mais leve. No entanto, essa poda da rede neural convolucional pode reduzir a precisão da detecção.

Para minimizar essa perda de precisão, aplica-se uma etapa de refinamento (*fine-tuning*), na qual o modelo é treinado novamente e monitorado para avaliar mudanças no *mAP* (*mean Average Precision*) e no histograma γ. O objetivo é encontrar o valor ideal de λ, que equilibra a taxa de poda e a precisão do modelo. Dessa forma, busca-se maximizar a eficiência computacional, reduzindo o tamanho e aumentando a velocidade do modelo sem comprometer significativamente a qualidade da detecção [16].

Para demonstrar de forma eficaz o desempenho do modelo *PYSS*, sua performance é avaliada por meio de métricas como Precisão, Lembrança, $mAP@0.5$ ($IoU = 0.5$), *F1-score* e Velocidade de Inferência. Com métricas definidas da seguinte forma:

$$Precisão = \frac{TP}{TP+FP} * 100\% \quad (4)$$

$$Lembrança = \frac{TP}{TP+FN} * 100\% \quad (5)$$

$$F1 = \frac{2TP}{2TP+FP+FN} * 100\% \quad (6)$$

$$mAP = \frac{1}{nc}\sum_{nc}^{nc=1} AP(nc) \quad (7)$$

$$IoU = \frac{area(C) \cap area(G)}{area(C) \cup area(G)} \quad (8)$$

Onde os verdadeiros positivos (TP), falsos positivos (FP) e falsos negativos (FN) representam, respectivamente, amostras positivas classificadas corretamente, amostras negativas classificadas incorretamente e amostras positivas classificadas incorretamente. A Precisão Média (AP) corresponde à área sob a curva Precisão-Lembrança, medindo o grau de sobreposição entre a área de fronteira prevista ($areaC$) e a área de fronteira real do objeto ($areaG$). A Média da Precisão Média (mAP) é obtida calculando o valor de AP para cada classe e dividindo pelo número total de classes (nc) [16]. Neste artigo será considerado as mesmas condições, com o mAP foi calculado considerando valores de IoU entre 0.5 e 0.95, divididos em 10 intervalos. A média desses dez valores resulta no $mAP@0.5{:}\,0.95$, enquanto o primeiro intervalo é utilizado para o $mAP@0.5$ [16].

IV. Resutados Esperados

Com a implementação do modelo *YOLO* para detecção de doenças em folhas de melão, espera-se alcançar melhorias significativas na identificação automática de plantas afetadas, reduzindo a necessidade de inspeção manual e aumentando a eficiência do diagnóstico agrícola. São esperados obter uma alta taxa de precisão média (*mean Average Precision - mAP*), principalmente no intervalo $mAP@0.5{:}\,0.95$, garantindo que o modelo identifique corretamente folhas saudáveis e doentes, assim contribuirá para um manejo mais eficiente da cultura, reduzindo perdas, otimizando o uso de defensivos agrícolas e melhorando a produtividade.

Estudos anteriores demonstram que modelos da série *YOLOv5* apresentam alta eficiência na detecção de objetos em tempo real, sendo amplamente utilizados para aplicações agrícolas. Girshick, et al [7] destacam que as versões aprimoradas do *YOLO* aumentaram significativamente a precisão e a velocidade de inferência em comparação com abordagens tradicionais. Além disso, Bochkovskiy [8] introduziram técnicas de aumento de dados e otimização que melhoram a robustez do modelo para diferentes condições ambientais.

Nesses estudos, como os feitos por Kong, et al [16], foram avaliados os modelos *YOLOv5s*, *YOLOv5m*, *YOLOv5l* e *YOLOv5s6*, utilizando *transfer learning* para ajustar os pesos previamente treinados. O conjunto de dados *CMD* foi utilizado para o treinamento, realizado ao longo de 300 épocas com parâmetros específicos ajustados. Estudos anteriores sugerem que a escolha adequada dos melhores parâmetros que pode impactar diretamente no desempenho do modelo, conforme demonstrado por Mohanty [6], que utilizaram aprendizado profundo para a detecção de doenças em plantas e observaram melhorias significativas melhoras ao ajustar parâmetros específicos.

As imagens de entrada foram processadas em dois tamanhos distintos, 640×640 e 1280×1280 *pixels*, para avaliar o impacto da resolução na detecção. Jocher [14] aponta que o uso de imagens maiores pode aumentar a precisão do modelo, porém, também exige maior capacidade computacional. Após o treinamento, os arquivos de resultados e os pesos dos modelos foram armazenados para posterior análise. A Figura 2 apresenta a comparação de desempenho entre os modelos da série *YOLOv5* [16].

Model	Precision/%	Recall/%	mAP@0.5/%	Speed/ms	F1/%
Retina Net-ResNet50-FPN	71.6	69.7	61.1	847	70.1
YOLO v3-DarkNet	69.6	75.4	69.8	803	72.4
Faster R-CNN-ResNet50-FPN	83.4	78.1	72.9	609	80.7
Cascade R-CNN-ResNet50-FPN	80.2	84.9	77.5	434	82.4
YOLO v4	82.5	79.4	88.2	206	80.9
YOLO v5s	84.5	82.9	90.3	4	90.0
PYSS	89.9	91.8	95.7	3.7	90.9

Fig. 2. Tabela de resultados dos testes das versões do *YOLO* [16].

Usando como parâmetro o modelo de escolha no estudo de Kong, et al [16]. O melhor desempenho em precisão foi

alcançando $mAP@0.5$ de 89,0%. No entanto, seu tamanho de $97{,}4\ MB$ limita sua flexibilidade para implantação em dispositivos de menor capacidade. Considerando esse fator, foi escolhido o *YOLOv5s* com tamanho de entrada de 640×640 *pixels*, que apresentou $mAP@0.5$ de 85,9%, tempo de inferência de 4,7 ms e um tamanho de modelo reduzido de $14{,}7\ MB$. Essa escolha foi baseada no equilíbrio entre precisão, velocidade e eficiência computacional, permitindo melhor adaptação a ambientes de hardware restrito [16].

V. Conclusão

A implementação de redes neurais convolucionais (*CNNs*), especificamente o modelo *YOLO*, para a detecção automática de doenças em folhas de melão demonstrou ser uma abordagem promissora para a agricultura de precisão. Através da utilização de técnicas avançadas de processamento de imagens e aprendizado profundo, espera-se obter um sistema eficiente, capaz de identificar doenças com alta precisão e velocidade de inferência.

Os resultados esperados indicam que a aplicação do *YOLO* pode proporcionar uma identificação mais rápida e precisa das doenças foliares, reduzindo a necessidade de inspeção manual e auxiliando os agricultores na tomada de decisões mais assertivas sobre o manejo da cultura. Isso não apenas otimiza o uso de defensivos agrícolas, como também minimiza perdas e melhora a produtividade.

Por fim, o uso da visão computacional na detecção de doenças agrícolas reforça o potencial da inteligência artificial no setor, tornando os processos agrícolas mais eficientes e sustentáveis. Pesquisas futuras podem explorar melhorias na adaptação do modelo a diferentes culturas, condições ambientais e integração com sistemas autônomos para monitoramento em tempo real.

Referências

[1] F. M. Silva, "Agricultura Brasileira: Produção, Exportação e Desafios," Revista Brasileira de Agricultura Sustentável, vol. 11, no. 3, pp. 98-112, 2021.

[2] IBGE, "Produção Agrícola Municipal," 2023. Disponível em: https://www.ibge.gov.br.

[3] CONAB, "Monitoramento da Produção Agrícola," 2023. Disponível em: https://www.conab.gov.br.

[4] J. R. Lima, "Cultivo do Melão no Semiárido Brasileiro," Revista Agropecuária do Nordeste, vol. 5, no. 2, pp. 45-59, 2020.

[5] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," Computers and Electronics in Agriculture, vol. 153, pp. 46-53, 2018.

[6] S. P. Mohanty, D. P. Hughes, e M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.

[7] R. Girshick, J. Redmon, S. Divvala, e A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2015.

[8] A. Bochkovskiy, C.-Y. Wang, e H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint, arXiv: 2004.10934, 2020.

[9] I. Goodfellow, Y. Bengio, e A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[10] R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Springer, 2022.

[11] S. Russell e P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.

[12] C. M. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006.

[13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436-444, May 2015.

[14] G. Jocher, "YOLOv5 by Ultralytics," Disponível em: https://github.com/ultralytics/yolov5, 2021.

[15] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, e C. Zhang, “Learning efficient convolutional networks through network slimming,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2755–2763.

[16] S. Kong, L. Xing, Q. Wang, X. Cong, Y. Zhou, "Object detection model based on YOLO v5 and ShuffleNet v2 for plant leaf diseases," Journal of Real-Time Image Processing, vol. 19, no. 4, pp. 985-995, 2022. DOI: 10.1007/s11554-022-01239-7.

[17] J. Jayakumar, "Plant Disease Detection in Melon Leaves Using Stacked RNN," 2020. Disponível em: https://ieeexplore.ieee.org/document/9262414.