Aplicação de Redes Neurais Convolucionais para o Diagnóstico de Doenças Foliares na Cultura da Manga

Sávio Augusto Dias Pereira
Departamento de Computação
Universidade Federal de Viçosa (UFV)
Rio Paranaíba, Minas Gerais, Brasil
savioust.dev@gmail.com

***Abstract*—Este artigo apresenta o desenvolvimento de uma rede neural convolucional (CNN) aplicada à visão computacional, com o objetivo de reconhecer pragas e doenças em folhas de mangueiras, auxiliando agricultores no diagnóstico precoce. Para o experimento, utilizou-se um dataset do Kaggle contendo 4.000 imagens distribuídas em 8 classes distintas. O processo de treinamento foi implementado em Python com PyTorch, empregando duas arquiteturas: AlexNet e EfficientNetB0. O conjunto de dados foi dividido em treino, validação e teste, e os modelos foram treinados por 10 épocas, com batch size de 32 e taxa de aprendizado de 0.001, utilizando o otimizador Adam. Durante o treinamento, foram monitoradas métricas como acurácia, precisão macro, recall e F1-score, além da matriz de confusão para análise detalhada do desempenho por classe. Ambos os modelos atingiram resultados satisfatórios, demonstrando potencial para aplicação prática no apoio ao diagnóstico agrícola e na automação de sistemas de detecção de pragas.**

***Index Terms*—Visão computacional, CNN, Detecção de pragas, Agricultura de precisão, Deep Learning**

I. Introdução

A agricultura moderna demanda soluções tecnológicas capazes de otimizar processos produtivos e reduzir perdas causadas por pragas e doenças. Entre as culturas de grande relevância econômica no Brasil, destaca-se a mangueira, cuja produção pode ser significativamente comprometida por patógenos que afetam folhas, frutos e ramos. Métodos tradicionais de diagnóstico, baseados em inspeção visual, são frequentemente lentos, subjetivos e dependem da experiência do agricultor ou técnico especializado. Nesse contexto, técnicas de visão computacional e aprendizagem profunda têm se mostrado promissoras no apoio ao monitoramento automatizado de lavouras.

Nos últimos anos, redes neurais convolucionais (CNNs) consolidaram-se como o estado da arte em tarefas de classificação de imagens, oferecendo alta capacidade de extração de padrões visuais complexos. Modelos como AlexNet e EfficientNet demonstram desempenho robusto em diversos cenários e têm sido aplicados com sucesso em diagnósticos agrícolas, detecção de pragas e identificação de doenças foliares.

Este trabalho apresenta o desenvolvimento e avaliação de duas arquiteturas de CNN — AlexNet e EfficientNetB0 — aplicadas ao reconhecimento de pragas e doenças em folhas de mangueiras. Utilizou-se um dataset contendo 4000 imagens distribuídas em oito classes distintas, organizado com conjuntos de treino, validação e teste. Os modelos foram treinados em ambiente Python utilizando a biblioteca PyTorch, com acompanhamento das métricas de acurácia, precisão, recall e F1-score, além da análise da matriz de confusão.

O objetivo deste estudo é investigar o desempenho das arquiteturas selecionadas e verificar sua viabilidade como ferramenta de apoio ao diagnóstico precoce no campo, contribuindo para práticas de agricultura de precisão e para o desenvolvimento de sistemas automatizados de monitoramento fitossanitário. Os resultados obtidos indicam que ambas as CNNs apresentam desempenho satisfatório, demonstrando potencial para integração em soluções práticas voltadas a produtores rurais e pesquisadores.

II. Revisão Bibliográfica

A literatura recente demonstra um avanço significativo no uso de técnicas de visão computacional e aprendizado profundo para detecção de doenças foliares. Diversos trabalhos exploram arquiteturas modernas, como EfficientNet, ResNet, MobileNet e Vision Transformers (ViT), destacando melhorias em desempenho, robustez e eficiência computacional quando comparadas às CNNs tradicionais.

No caso específico da cultura da mangueira, estudos recentes reportam o uso de CNNs e transformers na identificação automática de doenças em folhas, avaliando diferentes configurações de modelos e estratégias de pré-processamento. Hossain et al. investigam a aplicação de Vision Transformers e realizam comparações diretas com CNNs amplamente utilizadas, demonstrando ganhos em capacidade de representação e generalização. [1]

Entre as arquiteturas avaliadas na literatura, EfficientNet-B0 e suas variantes destacam-se por apresentarem uma relação favorável entre precisão e custo computacional. Trabalhos comparativos mostram que essa família de modelos supera, em diversos cenarios, arquiteturas clássicas como ResNet e DenseNet na classificação de doenças foliares. [2]

Embora arquiteturas modernas tenham ganhado protagonismo, modelos clássicos como AlexNet continuam presentes em estudos aplicados devido à sua simplicidade, baixo custo computacional e facilidade de adaptação. Pesquisas recentes propõem variantes aprimoradas que incorporam módulos Inception, mecanismos de atenção ou ajustes estruturais, resultando em melhorias de desempenho especialmente em contextos agrícolas. [3]

Revisões sistemáticas e estudos de síntese reforçam a tendência de adoção de modelos cada vez mais robustos, destacando desafios como variação de iluminação, ruído nas imagens, diferenças entre ambientes controlados e condições reais de campo, além da necessidade de ampliação de datasets e de técnicas de aumento de dados. Esses trabalhos também evidenciam a crescente utilização de arquiteturas híbridas e de sensores multiespectrais no monitoramento agrícola. [4]

III. Material e Métodos

A. Dataset

Os experimentos foram conduzidos utilizando o *MangoLeafBD Dataset*, um conjunto público de imagens de folhas de mangueira disponibilizado por Ali et al. [5]. O dataset contém 4.000 imagens em formato JPG, com resolução aproximada de 240×320 pixels. Destas, cerca de 1.800 correspondem a folhas distintas, enquanto o restante foi obtido por técnicas de aumento de dados aplicadas pelos autores, como zoom e rotações.

As imagens estão organizadas em oito classes: sete doenças (*Anthracnose*, *Bacterial Canker*, *Cutting Weevil*, *Die Back*, *Gall Midge*, *Powdery Mildew* e *Sooty Mould*) e uma classe de folhas saudáveis, cada classe contendo 500 instâncias. As fotografias foram coletadas em quatro pomares de Bangladesh: Sher-e-Bangla Agricultural University, Jahangir Nagar University, Udaypur Village e Itakhola Village, apresentando variação realista de iluminação e condições ambientais.

Para os experimentos, o dataset foi dividido em treinamento, validação e teste seguindo o esquema estratificado 70%/15%/15%, garantindo balanceamento entre classes. O módulo `load_dataset` foi utilizado para carregar as imagens a partir do diretório organizado, retornando os respectivos `DataLoaders` e a lista de classes.

B. Pré-processamento e DataLoader

O pré-processamento das imagens foi realizado pelo módulo `dataset_loader.py`, que aplica uma série de transformações antes do envio dos dados aos modelos. As etapas incluem:

- **Redimensionamento** das imagens para a resolução de entrada exigida pelo modelo, seguido de *center crop* ou *random crop* durante o treinamento.
- **Normalização** utilizando médias e desvios padrão compatíveis com redes pré-treinadas no ImageNet, garantindo compatibilidade quando inicializações *pretrained* são utilizadas.
- **Data augmentation** no conjunto de treino, incluindo rotações, inversões horizontais/verticais, e ajustes de brilho e contraste, com o objetivo de aumentar a robustez do modelo e reduzir sobreajuste.

Os conjuntos foram carregados por meio de *DataLoaders* do PyTorch, configurados com `batch_size = 32`, embaralhamento (*shuffle*) ativado para o treino e carregamento paralelo via `num_workers` definido conforme o hardware disponível.

C. Arquiteturas

Foram avaliadas duas arquiteturas amplamente utilizadas em tarefas de classificação de imagens:

- **AlexNet** — implementada via `torchvision.models.alexnet`. Trata-se de uma arquitetura clássica, de baixa complexidade e rápida de treinar, frequentemente utilizada como linha de base em experimentos de visão computacional. Sua estrutura relativamente simples — cinco camadas convolucionais seguidas por três camadas totalmente conectadas — permite avaliar rapidamente o comportamento do conjunto de dados sem elevado custo computacional.
- **EfficientNet-B0** — carregada através de `torchvision.models.efficientnet_b0`. Pertence à família EfficientNet, que utiliza otimização composta (*compound scaling*) para escalar largura, profundidade e resolução de forma equilibrada. O modelo B0 é reconhecido por apresentar uma boa relação entre desempenho e eficiência computacional, sendo adequado para cenários com recursos limitados ou datasets de médio porte.

Em ambos os casos, os modelos foram instanciados com `weights=None` e configurados com `num_classes = 8`, substituindo a camada final para se adequar ao número de classes do dataset.

Embora a literatura frequentemente recomende o uso de pesos pré-treinados no ImageNet para melhorar a capacidade de generalização, neste trabalho optou-se por treinar todas as arquiteturas *from scratch*, de modo a avaliar exclusivamente o desempenho intrínseco de cada rede no domínio de folhas de mangueira, sem transferência de conhecimento externo.

O estudo também considerou abordagens avançadas — tais como *feature extraction*, *fine-tuning* e uso de resoluções maiores no EfficientNet —, porém tais técnicas não foram empregadas nos experimentos principais, uma vez que requerem pesos pré-treinados. São, entretanto, sugeridas como trabalhos futuros, tendo em vista seu potencial de acelerar a convergência, reduzir sobreajuste e melhorar o desempenho em aplicações reais.

D. Treinamento

Os modelos foram treinados utilizando **50 épocas**, valor definido para permitir maior estabilidade na convergência devido ao fato de as arquiteturas terem sido inicializadas **sem pesos pré-treinados** (`weights=None`). Dessa forma, todo o aprendizado foi obtido exclusivamente a partir do dataset empregado.

O otimizador adotado foi o **Adam** com taxa de aprendizado inicial de `lr = 0.001`. A função de perda utilizada foi a

CrossEntropyLoss. Para melhorar a adaptação do modelo ao longo do treinamento, aplicou-se o scheduler **ReduceLROnPlateau**, que reduz automaticamente a taxa de aprendizado com base na perda de validação.

O processo de treinamento foi executado em **GPU** quando disponível (`device = 'cuda'`), e cada época contou com monitoramento em tempo real por meio de barras de progresso `tqdm`, exibindo a evolução da loss por batelada.

Além da acurácia, as métricas de **Precisão, Recall e F1-score** foram calculadas ao final de cada época para melhor avaliação da performance. Também foi implementado **Early Stopping** baseado no melhor F1 de validação, com salvamento automático do melhor modelo.

O histórico completo de treinamento foi salvo em arquivos JSON no formato `history_{model_name}.json`, enquanto os pesos dos melhores modelos foram armazenados em `models/{model_name}.pt`.

E. Avaliação

As métricas utilizadas para avaliar o desempenho dos modelos foram:

- **Acurácia** (`accuracy_score`);
- **Precisão macro** (`precision_score`);
- **Recall macro** (`recall_score`);
- **F1-score macro** (`f1_score`);
- **Matriz de confusão** (via `sklearn.metrics.confusion_matrix`).

A avaliação foi realizada por meio do método `evaluate()`, que percorre o *data loader* correspondente em modo `eval` e com `torch.no_grad()`, garantindo que não haja atualização de gradientes. O método calcula as métricas de forma agregada para cada época durante a validação e, ao final do treinamento, gera também os resultados no conjunto de teste.

IV. Resultados e Discussão

A. Histórico de Treinamento

O acompanhamento do processo de aprendizagem foi realizado por meio dos valores de *loss* no conjunto de treinamento e das métricas calculadas no conjunto de validação. As Figuras 1, 2, 3 e 7 apresentam os históricos de treinamento dos modelos EfficientNetB0 e AlexNet, mostrando a evolução do *train loss* e da *val accuracy* ao longo das épocas.

A EfficientNetB0 demonstrou convergência estável mesmo quando treinada do zero, reduzindo progressivamente o erro e alcançando métricas consistentes de validação. O mesmo comportamento pode ser observado no modelo AlexNet, que também apresenta queda regular do *loss* e estabilização das métricas ao longo das épocas, embora com desempenho inferior — algo esperado devido às limitações arquiteturais do AlexNet em relação a modelos mais modernos.

Observa-se que o AlexNet, apesar de ser uma arquitetura clássica, ainda é capaz de capturar características visuais relevantes para a classificação de doenças foliares, evidenciando que modelos mais antigos continuam úteis quando corretamente treinados. Entretanto, sua capacidade de generalização e a velocidade de convergência são inferiores, devido à arquitetura mais simples e à ausência de mecanismos modernos de normalização e atenção presentes na EfficientNetB0. O acompanhamento detalhado do *train loss* e das métricas por época permitiu identificar padrões de estabilidade e flutuações temporárias, fornecendo insights importantes sobre o comportamento do modelo durante o aprendizado.

Outro ponto relevante é que o AlexNet apresentou maior sensibilidade às flutuações iniciais do *train loss*, necessitando de mais épocas para estabilização. Algumas classes com menor quantidade de amostras também foram classificadas com precisão ligeiramente inferior. Esses aspectos reforçam a vantagem de arquiteturas mais atuais em cenários mais desafiadores, ao mesmo tempo em que demonstram como modelos tradicionais seguem úteis para análises experimentais, comparação de desempenho e contextualização da evolução dos métodos de visão computacional ao longo do tempo.

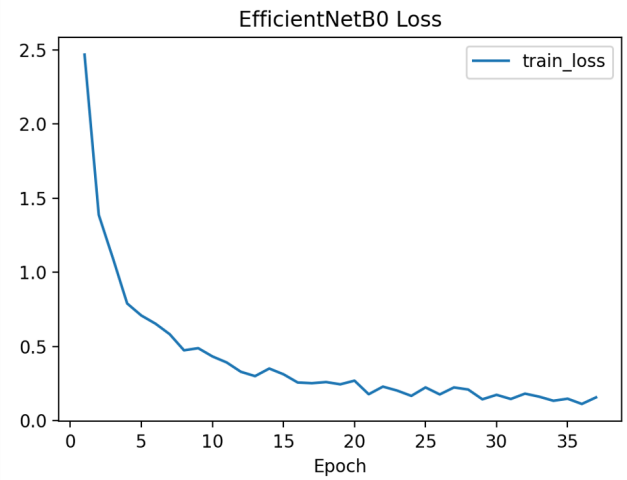

Fig. 1. Histórico de *train loss* por época para o modelo EfficientNetB0.

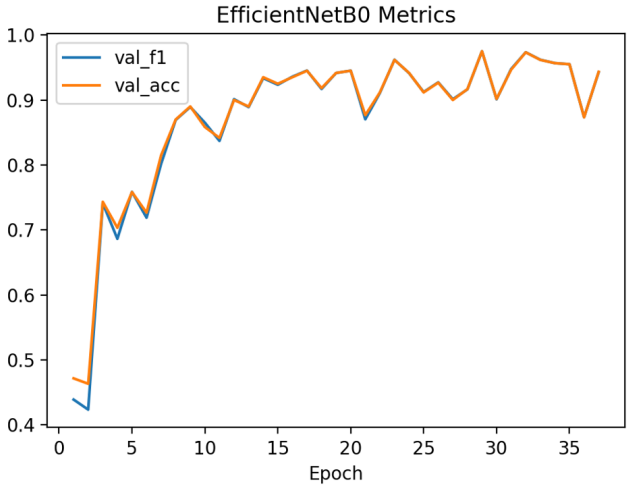

Fig. 2. Histórico de *val accuracy* por época para o modelo EfficientNetB0.

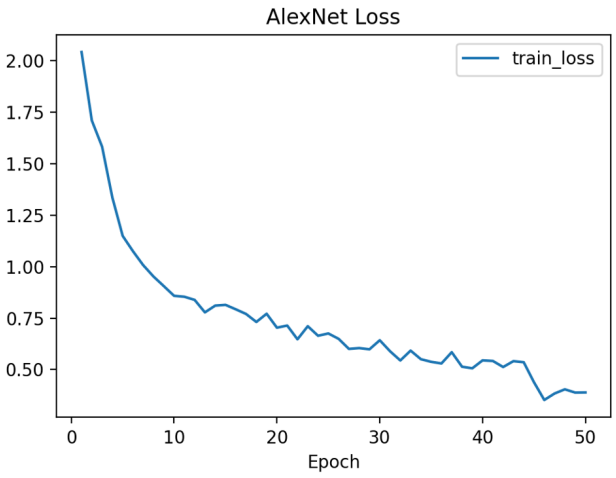

Fig. 3. Histórico de *train loss* por época para o modelo AlexNet.

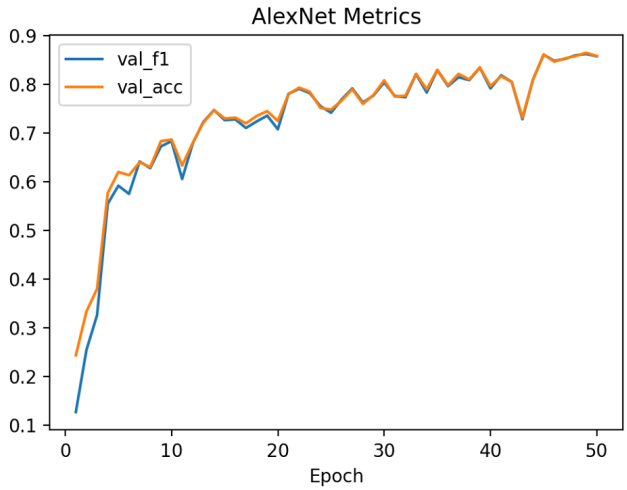

Fig. 4. Histórico de *val accuracy* por época para o modelo AlexNet.

B. Matriz de Confusão e Métricas por Classe

As matrizes de confusão permitem analisar de forma clara como cada modelo se comporta em relação às oito classes avaliadas, evidenciando tanto os acertos quanto as ambiguidades entre categorias visualmente semelhantes. As Figuras 5 e 6 apresentam, respectivamente, os resultados obtidos pela EfficientNetB0 e pela AlexNet, possibilitando uma comparação direta entre as duas arquiteturas.

A EfficientNetB0 demonstrou excelente capacidade de generalização, alcançando acurácia de 95,33%, precisão de 95,55%, recall de 95,33% e F1-score de 95,35%. Sua matriz de confusão revela uma forte concentração de acertos na diagonal principal, indicando que o modelo foi capaz de separar adequadamente a maior parte das classes. As poucas confusões observadas ocorreram principalmente entre categorias com padrões foliares muito semelhantes, como entre a Classe 0 e as Classes 2, 3 e 4, além de pequenos erros entre as Classes 6 e 7, que compartilham características visuais de textura e coloração.

Por outro lado, a AlexNet apresentou desempenho inferior, mas ainda coerente com sua arquitetura mais simples. O modelo obteve acurácia de 86,50%, precisão de 87,33%, recall de 86,50% e F1-score de 85,99%. Sua matriz de confusão evidencia maior dispersão fora da diagonal, com destaque para a Classe 4, que apresentou erros recorrentes direcionados às Classes 1, 6 e 7. Esse comportamento indica dificuldade em capturar nuances visuais mais sutis, reforçando as limitações do AlexNet quando comparado à capacidade de extração de características da EfficientNetB0.

A combinação dessas análises confirma a clara vantagem da EfficientNetB0, especialmente em classes que apresentam alto grau de similaridade visual, consolidando seu desempenho superior tanto nos indicadores globais quanto na discriminação entre as categorias avaliadas.

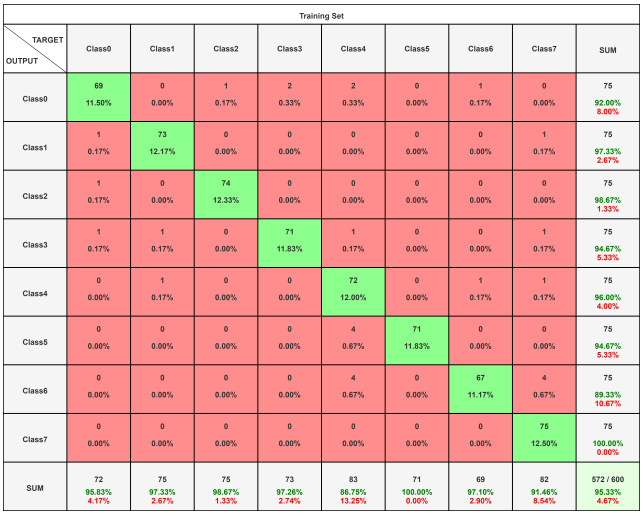

Fig. 5. Matriz de confusão do modelo EfficientNetB0 no conjunto de teste.

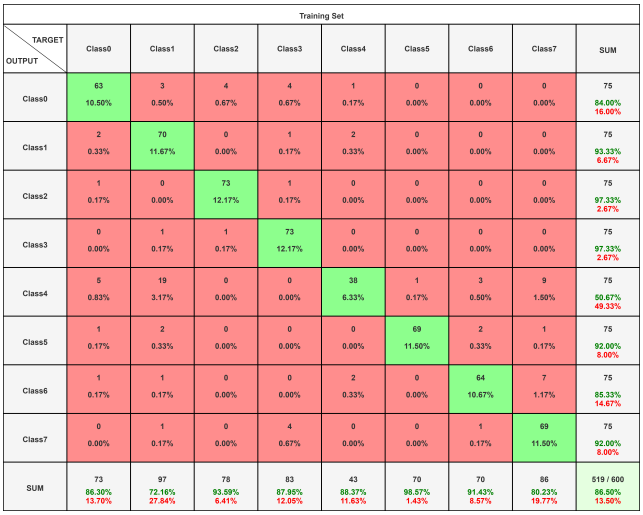

Fig. 6. Matriz de confusão do modelo AlexNet no conjunto de teste.

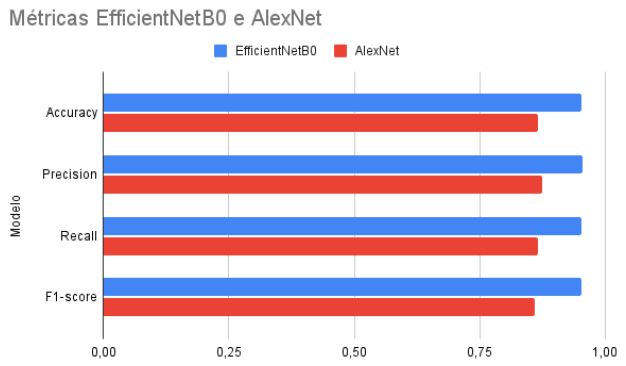

Fig. 7. Métricas globais de avaliação dos modelos EfficientNetB0 e AlexNet.

C. Comparação com a Literatura

Os resultados obtidos são consistentes com a literatura sobre classificação de doenças foliares, que aponta a família EfficientNet como uma das mais eficazes para tarefas de visão computacional aplicadas à agricultura. Diversos estudos indicam que essa arquitetura apresenta desempenho superior em relação a redes clássicas, devido à sua estratégia de escalonamento composto, que equilibra simultaneamente profundidade, largura e resolução de forma otimizada.

É importante destacar que, diferentemente da maior parte dos trabalhos existentes, que utilizam **fine-tuning com pesos pré-treinados no ImageNet**, este estudo treinou a EfficientNetB0 **inteiramente do zero**. Ainda assim, o modelo alcançou desempenho elevado (F1-score acima de 95% no conjunto de teste), evidenciando não apenas a robustez da arquitetura, mas também a qualidade do dataset empregado e a eficácia do pipeline de treinamento implementado.

D. Limitações

Embora os resultados obtidos sejam expressivos, algumas limitações devem ser consideradas na interpretação do desempenho dos modelos. Primeiramente, o dataset utilizado contém apenas 4000 imagens, o que pode restringir a capacidade de generalização para cenários reais, especialmente em condições de campo que envolvem maior variabilidade de iluminação, ângulos de captura e presença de ruídos contextuais.

Outro ponto relevante é que os modelos foram treinados com `weights=None`, ou seja, inteiramente do zero. Essa abordagem, embora interessante do ponto de vista experimental, tende a exigir quantidades maiores de dados e mais épocas de treinamento para atingir o desempenho ideal. Assim, uma continuação natural deste trabalho seria avaliar o uso de modelos pré-treinados (`pretrained=True`) seguido de *fine-tuning*, estratégia amplamente validada na literatura.

Por fim, não foi realizada validação em campo com imagens coletadas por agricultores, técnicos ou dispositivos móveis, o que representa uma etapa essencial para verificar a robustez do modelo em condições reais de uso. A aplicação prática da solução deve incluir essa fase como um próximo passo importante para consolidar sua aplicabilidade em ambientes agrícolas.

V. Conclusão

Este trabalho apresentou um pipeline completo para a classificação de doenças em folhas de mangueira utilizando redes neurais convolucionais implementadas em PyTorch, com foco nas arquiteturas AlexNet e EfficientNet-B0. A avaliação baseada em acurácia, precisão, recall, F1-score e análise das matrizes de confusão demonstrou que ambos os modelos são capazes de realizar a tarefa com desempenho satisfatório. No entanto, a EfficientNet-B0 mostrou-se claramente superior, apresentando maior estabilidade, melhor capacidade de generalização e erros substancialmente menos dispersos entre as classes.

Os resultados obtidos reforçam o potencial do uso de CNNs como ferramenta de apoio ao diagnóstico precoce de doenças foliares, contribuindo para práticas agrícolas mais eficientes. Como continuidade deste estudo, destacam-se quatro direções principais: (1) investigar o uso de modelos pré-treinados, aplicando *transfer learning* e *fine-tuning*; (2) expandir o conjunto de dados e validar o modelo em imagens capturadas em campo; (3) avaliar arquiteturas leves e técnicas de compressão, como MobileNet e quantização, visando implantação em dispositivos móveis; e (4) integrar o classificador a um pipeline de segmentação capaz de isolar automaticamente as lesões nas folhas.

References

[1] M. A. Hossain, S. Sakib, H. M. Abdullah, and S. E. Arman, "Deep learning for mango leaf disease identification: A vision transformer perspective," *Heliyon*, vol. 10, no. 17, p. e36361, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844024123924

[2] Ümit Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954120301321

[3] Z. Li, C. Li, L. Deng, Y. Fan, X. Xiao, H. Ma, J. Qin, and L. Zhu, "Improved alexnet with inception-V4 for plant disease diagnosis," *Computational Intelligence and Neuroscience*, vol. 2022, p. 5862600, 2022. [Online]. Available: https://doi.org/10.1155/2022/5862600

[4] A. Upadhyay, N. S. Chandel, K. P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, and A. Elbeltagi, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artificial Intelligence Review*, vol. 58, no. 3, p. 92, 2025. [Online]. Available: https://doi.org/10.1007/s10462-024-11100-x

[5] S. Ali, M. Ibrahim, S. I. Ahmed, M. Nadim, M. Mizanur Rahman, M. M. Shejunti, and T. Jabid, "Mangoleafbd dataset," 2022.