

### **MODELO DE VISÃO COMPUTACIONAL PARA PREDIÇÃO DE DOENÇAS EM FOLHAS**

Bernardo L. F. Padua<sup>1</sup> (EG), Egon L. M. Júnior (PQ)<sup>1</sup>, Lenisa M. C. Sousa (EG)<sup>1</sup> ,Roberto R. G. Júnior<sup>1</sup> (EG)

<sup>1</sup>Universidade Federal de Itajubá.

**Palavras-chave:** U-net, visão computacional, agricultura de precisão.

#### **Introdução**

A agricultura, essencial para a segurança alimentar global, enfrenta perdas anuais significativas devido a doenças em plantas, que comprometem a produtividade e a economia desse setor. A Organização das Nações Unidas para a Alimentação e a Agricultura (FAO) estima que a produção agrícola global tenha sofrido perdas de aproximadamente US\$ 123 bilhões por ano. A detecção tradicional dessas patologias, baseada na inspeção visual, é frequentemente subjetiva, de baixa escalabilidade e requer conhecimento especializado, muitas vezes resultando em diagnósticos incorretos devido à similaridade de sintomas entre diferentes doenças. Diante disso, a Inteligência Artificial (IA), especificamente o aprendizado profundo (*deep learning*), é uma solução tecnológica para automatizar e qualificar o diagnóstico fitossanitário de forma precisa e eficiente.

Modelos de *deep learning*, como as redes neurais convolucionais (CNNs), são capazes de extrair automaticamente características visuais, como forma, cor e textura das lesões, minimizando o viés da seleção manual de características. Esta abordagem vai além da simples classificação de imagens, que apenas rotula uma folha como "doente", e se concentra na segmentação semântica, uma técnica que classifica cada pixel da imagem para delinear com exatidão as áreas afetadas. Essa capacidade de segmentação é fundamental, pois permite a quantificação objetiva da severidade da doença, um passo crucial para a agricultura de precisão e a aplicação direcionada de tratamentos. Este trabalho tem como objetivo principal o desenvolvimento de um modelo de IA baseado na arquitetura de rede neural convolucional U-Net para a detecção e segmentação de lesões de doenças em folhas de plantas.

A escolha da U-Net é justificada por sua alta

performance em tarefas de segmentação, mesmo com conjuntos de dados limitados, graças à sua estrutura de codificador-decodificador com conexões de salto (*skip connections*), que preservam detalhes espaciais finos enquanto capturam o contexto global da imagem. Essa capacidade permite que a U-Net delimita com exatidão os contornos das áreas foliares afetadas, como demonstrado em estudos que alcançaram elevadas acurárias de validação.

O método a ser utilizado envolverá o treinamento e a validação do modelo com bancos de imagens, seguindo procedimentos de pré-processamento de dados e avaliação de desempenho por meio de métricas de acurácia de segmentação.

#### **Metodologia**

O desenvolvimento e validação do modelo de segmentação de doenças em folhas de plantas foi baseado na implementação da arquitetura U-Net. A abordagem empregada nessa arquitetura descreve a sequência de etapas, desde o preparo dos dados até o treinamento e avaliação do modelo.

A pesquisa utilizou o conjunto de dados público "**PlantDoc: A Dataset for Visual Plant Disease Detection**". Este dataset é composto por imagens de folhas de diversas espécies de plantas, contendo tanto exemplares saudáveis quanto afetados por diferentes doenças. Para cada imagem de folha, há exatamente uma máscara de segmentação correspondente, que delimita a área da folha em relação ao fundo da imagem.

Antes de serem utilizadas no treinamento do modelo, as imagens e suas respectivas máscaras passaram por uma etapa de pré-processamento. Primeiramente, todas as imagens foram redimensionadas

## ***“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”***

para uma dimensão padrão de 128x128 pixels, conforme formato da camada de entrada da rede neural. Então foram normalizadas, os valores dos pixel das imagens, originalmente no intervalo [0, 255], foram normalizados para o intervalo [0,1]. Este procedimento é uma prática padrão indicada para estabilizar e acelerar a convergência do treinamento. As máscaras de segmentação foram mantidas com seus valores binários [0, 1].

O conjunto de dados foi dividido em dois subconjuntos: 80% para treinamento e 20% para validação. Esta divisão permite treinar o modelo com a maior parte dos dados, tendo uma maior variabilidade dos dados, e avaliando seu desempenho em um conjunto de dados não visto durante o treinamento, a fim de monitorar casos de *overfitting*.

Para a segmentação de imagem, foi implementado uma arquitetura de Rede Neural Convolucional (CNN) baseada no modelo U-NET. A U-NET é uma rede adequada para realização de segmentações semânticas em imagens biomédicas e tarefas similares, isso é devido a sua estrutura simétrica composta por um caminho de contração (codificador) e um caminho de expansão (decodificador).

O caminho de Contração consiste em blocos sequenciais de camada de convolução (Conv2D) com filtros de 3x3 e função de ativação ReLU (*Rectified Linear Unit*), seguidos por camadas de *Batch Normalization* para estabilizar o treinamento. Ao final de cada bloco, uma camada de *Max Pooling* de 2x2 é aplicada para reduzir as dimensões especiais da imagem e extrair características hierárquicas. A cada bloco, a resolução espacial da imagem é reduzida pela metade, enquanto os números de canais/filtros aumentam a cada bloco, começando em 32 e progredindo para 64, 128, 256 e, por fim, 512 no bloco de conexão (*bottleneck*) permitindo a captura de características mais complexas.

O caminho de Expansão irá reconstruir a máscara de segmentação a partir das características extraídas pelo codificador, cada bloco do decodificador é composto por camadas de convolução transpostas (Conv2DTranspose) 2x2 que realizam o aumento da resolução espacial (*upsampling*) dos mapas de características. Em seguida, o mapa resultante é concatenado com seu mapa de característica correspondente ao caminho de contração. Uma

característica fundamental do modelo é o uso das “conexões de salto” (*Skip connections*), que concatenam os mapas de características do caminho da contração com os do caminho de expansão. Isso permite que a rede recupere informações espaciais de alta resolução perdidas durante o *downsampling*, resultando em segmentações mais precisas.

A arquitetura é finalizada com uma camada de saída do modelo consiste em uma convolução de 1x1 dos dois filtros que mapeia os canais de características para o número de classes desejado, neste (no caso deste projeto, duas, fundo e doença). Uma função de ativação *softmax*, que produz um mapa de probabilidade para cada pixel indicando se ele pertence à uma das classes, resultando na máscara de segmentação.

Como forma de regularização e aceleração do treinamento, a Batch Normalization foi aplicada após cada camada de convolução.

O modelo foi compilado utilizando o otimizador Adam, uma escolha devido a sua eficiência computacional e baixa exigência de memória, e a função de perda binary *Cross Entropy*, adequada para problemas de classificação binária pixel a pixel. Tendo como métrica monitorada durante o processo de treinamento a acurácia.

O treino foi configurado para uma máxima de 50 épocas, e teve o uso de callbacks para otimizar o processo de treinamento e evitar condições de *overfitting*. Os métodos de callbacks utilizados foram o *EarlyStopping*, que monitorava as perdas no conjunto de validação e interrompia o treinamento caso não houvesse melhora após X épocas consecutivas e o *ModelCheckpoint*, que salva o modelo com melhor desempenho no conjunto de validação, monitorando também as perdas.

## **Resultados e discussão**

O modelo U-net foi treinado por 14 épocas, sendo o treinamento interrompido precocemente na 14<sup>a</sup> época, conforme indicado pelo callback *EarlyStopping*. Durante o treinamento, as medidas de perda de validação (val loss) e de acurácia foram monitoradas para salvar o modelo com melhor desempenho, resultando no arquivo do modelo.

## *“Do conhecimento acadêmico à transformação sustentável: inovação com validação científica”*

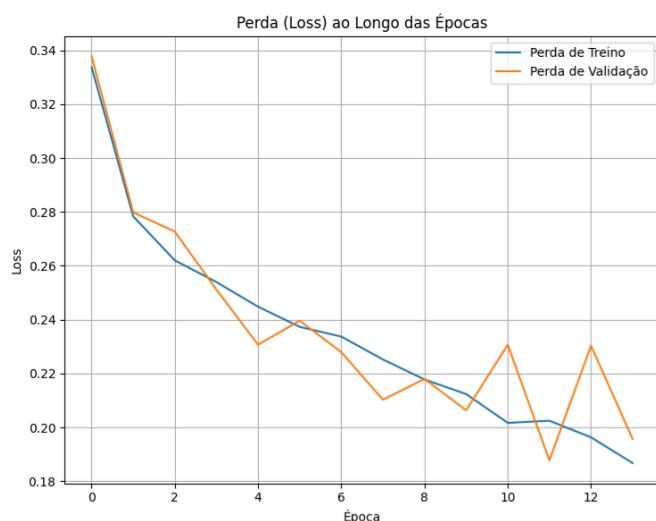


Figura 1 – Gráfico da perda por época

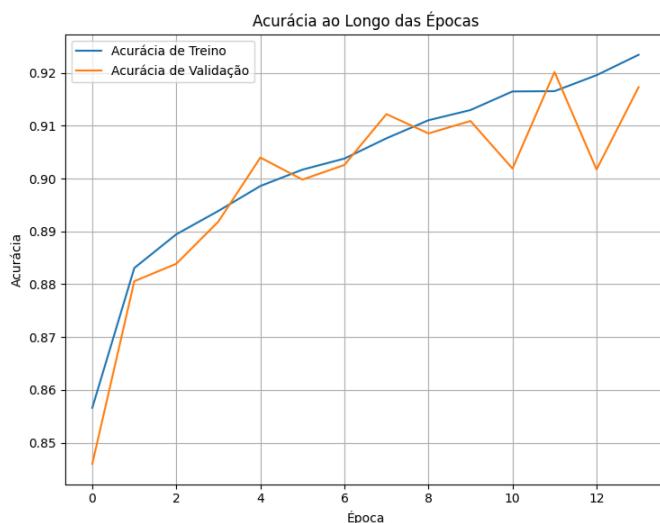


Figura 2 – Gráfico da acurácia por época

A tabela de resumo do modelo (`model.summary()`) mostrou a arquitetura U-net, com um `input_layer` de 128x128x3 (altura, largura, canais de cor) e um `output_shape` de 128x128x2 (altura, largura, número de classes), possui um total de **7.771.906 parâmetros**, dos quais **7.766.018 são treináveis**. Isso indica uma arquitetura robusta e capaz de aprender características complexas das imagens.

A análise do histórico do treinamento revelou inicialmente que a perda de validação melhorou progressivamente nas primeiras épocas, passando de 0.3377 na época 1 para 0.1878 na época 12 como é possível ver na Figura 1. A acurácia, em mesmo âmbito demonstrou melhora, alcançando 0.9202 na época 12

representado pela Figura 2. Sendo interrompido na época 14 porque a `val_loss` não melhorou em relação ao valor mínimo na época 12.

Os resultados de predição em dados de validação mostraram uma segmentação visualmente convincente, com a máscara predita se assemelhando à máscara verdadeira para as amostras exibidas, como presentes nas figuras seguintes.

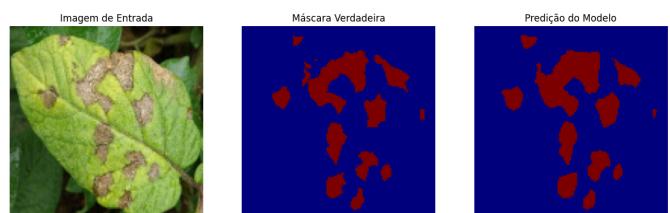


Figura 3 – Predição em folha de batata

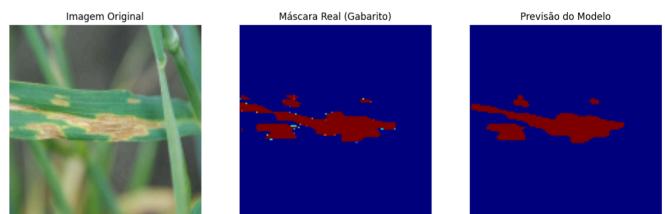


Figura 4 - predição do modelo em soja

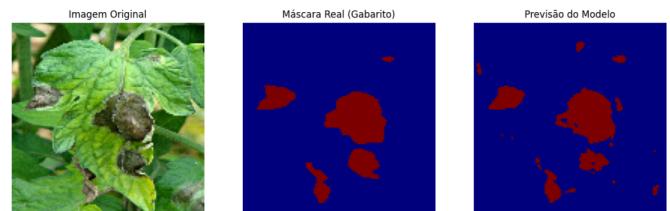


Figura 5 - predição em folha de tomate

O modelo U-Net demonstrou ser uma arquitetura eficaz para a tarefa de segmentação de imagens de folhas, como evidenciado pela melhora constante na acurácia e na perda durante o treinamento inicial. A utilização de um gerador de dados permitiu o treinamento com um conjunto de dados maior e mais diversificado por meio de aumentações, como a inversão que contribui para a robustez do modelo.

A estratégia de *EarlyStopping* foi crucial para evitar o *overfitting* (ajuste excessivo aos dados de treinamento). O fato de o treinamento ter parado na Época 14, depois de a `val_loss` ter parado de melhorar, sugere que o modelo já havia aprendido as

## *"Do conhecimento acadêmico à transformação sustentável: inovação com validação científica"*

características essenciais para generalizar bem para novos dados. Continuar o treinamento poderia ter resultado em um modelo que memorize os dados de treinamento, prejudicando seu desempenho em dados não vistos.

A baixa quantidade de memória da GPU (1765 MB) e o aviso sobre a fragmentação de memória ("OOM due to memory fragmentation") indicam que o tamanho do lote pode ter sido um fator limitante. Um hardware com mais memória poderia permitir a utilização de lotes maiores e que potencialmente aceleraria o treinamento, e, em alguns casos, levaria a uma convergência mais estável.

Em suma, a implementação do modelo se mostrou bem sucedida, com o modelo aprendendo a segmentar as folhas com alta acurácia nos dados de validação. Futuros estudos poderiam explorar o Ajuste fino da arquitetura como a alteração de filtros ou blocos de convolução, utilizar métricas mais específicas para a segmentação como *Dice* ou *IoU* (*Intersection over Union*) e outras técnicas de aumento de dados com otimizadores diferentes do 'adam'.

### Conclusões

O presente trabalho demonstrou com eficácia a implementação de um modelo de visão computacional, baseado na arquitetura U-Net, para a segmentação de lesões em folhas de plantas. O objetivo do trabalho de desenvolver um modelo para a identificação precoce de doenças em folhas foi obtido, conforme apresentado e discutido ao longo do resumo.

O modelo treinado alcançou uma acurácia de validação expressiva, atingindo 0.9202 na 12<sup>a</sup> época, e as previsões visuais se mostraram convincentes ao delinear as áreas afetadas de forma similar às máscaras de referência. A utilização de técnicas como *EarlyStopping* foi fundamental para otimizar o treinamento e prevenir o sobreajuste (*overfitting*), garantindo que o modelo mantivesse a capacidade de generalizar para novos dados. A pesquisa valida a arquitetura U-Net como uma solução robusta para tarefas de segmentação em agricultura de precisão, capaz de quantificar objetivamente a severidade de doenças e auxiliar na aplicação direcionada de tratamentos. Limitações de hardware foram identificadas como um ponto que poderia restringir a aceleração do treinamento.

Para trabalhos futuros, sugere-se a exploração de otimizações na arquitetura, a utilização de métricas de avaliação mais específicas para segmentação, como *Dice* ou *IoU*, e a experimentação com diferentes técnicas de aumento de dados e otimizadores para aprimorar ainda mais o desempenho do modelo.

### Agradecimentos

Expressamos nossos agradecimentos à Universidade Federal de Itajubá, cuja infraestrutura acadêmica e apoio institucional foram fundamentais para a realização deste trabalho. Estendemos também nossa gratidão ao PET-TEC pelo incentivo constante à formação acadêmica e à pesquisa. Reconhecemos, ainda, o apoio do FNDE, que, por meio dos recursos disponibilizados, viabilizou a execução do projeto, contribuindo diretamente para o avanço das atividades desenvolvidas. Sem a colaboração dessas instituições, a concretização deste estudo não teria sido possível. Agradecemos a todos que, direta ou indiretamente, contribuíram para a conclusão deste projeto.

### Referências

FAKHRE, Alam. Leaf disease segmentation dataset. Kaggle, 2024. Disponível em: <https://www.kaggle.com/datasets/fakhrealam9537/leaf-disease-segmentation-dataset/data?select=data>. Acesso em 20 jul. 2025.

FAO BRASIL. *O impacto dos desastres na agricultura e na segurança alimentar*. Brasília: FAO, 2023. Disponível em: <https://www.fao.org/brasil/noticias/detail-events/es/c/1653890/>. Acesso em: 27 jul. 2025.

SHOAIB, M. et al. An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, v. 14, 21 mar. 2023.

TIVES, Heloise Acco; MARINI, Andreia; ORTONCELLI, André Roberto. Visão computacional aplicada na identificação de doenças na fruticultura: uma revisão sistemática da literatura. In: **CONGRESSO BRASILEIRO DE AGROINFORMÁTICA (SBIAGRO)**, 14., 2023, Lavras. *Anais do Congresso Brasileiro de Agroinformática (SBIAGRO)*. Porto Alegre: Sociedade Brasileira de Computação, 2023. Disponível em: <https://sol.sbc.org.br/index.php/sbiagro/article/view/26561>. Acesso em 28 jul. 2025.