Catalogação na publicação
Seção de Catalogação e Classificação

S725r Sousa, Antonio Isaac Araujo Firmino de.
Reconhecimento de plantas alimentícias não convencionais (PANCs) em ambientes naturais utilizando o modelo YOLO: uma abordagem baseada em visão computacional / Antonio Isaac Araujo Firmino de Sousa. - João Pessoa, 2025.
19 f. : il.

Orientação: Thaís Gaudêncio do Rego.
TCC (Graduação) - UFPB/CI.

1. PANCs. 2. Modelo. 3. Plantas. I. Rego, Thaís Gaudêncio do. II. Título.

UFPB/CI CDU 004.4

Elaborado por Michelle de Kássia Fonseca Barbosa - CRB-738

Reconhecimento de Plantas Alimentícias Não Convencionais (PANCs) em Ambientes Naturais Utilizando o Modelo YOLO: Uma Abordagem Baseada em Visão Computacional

Antonio Isaac A. F. de Sousa[1], **Thaís Gaudêncio do Rego**[1]

[1]Centro de Informática – Universidade Federal da Paraíba (UFPB)
João Pessoa – PB – Brasil

Abstract. *Non-Conventional Food Plants (PANCs) have gained prominence due to their nutritional value and sustainable cultivation, but they still face recognition challenges given Brazil's vast biodiversity and the limited educational resources available. This work aims to identify and localize PANCs in images using the YOLOv11 model, one of the most recent versions of the You Only Look Once (YOLO) architecture. For training and testing the model, seed 0 was used, with a total of 15,968 images and 19,037 instances, applying data augmentation techniques and distributing the samples across 14 classes. The images were randomly and stratifiedly divided into 70% for training, 20% for validation, and 10% for testing. All samples were obtained from the Pl@ntNet website. The effectiveness of the proposed method was evaluated using metrics such as precision, recall, confusion matrix, loss functions, mean Average Precision (mAP), and heatmaps generated with the Eigen-CAM algorithm. For the latter, the most activated regions were analyzed to identify the model's areas of attention, improving its explainability. The model achieved a precision of 0.877, a recall of 0.778, an mAP50 of 0.878, and an mAP50-95 of 0.801, facing greater difficulties with classes that are similar or have a wide variety of characteristics.*

Resumo. *As Plantas Alimentícias Não Convencionais (PANCs) vêm ganhando destaque pelo seu valor nutricional e cultivo sustentável, mas ainda enfrentam dificuldades de reconhecimento devido à vasta biodiversidade brasileira e à limitação de recursos educativos. Este trabalho tem como objetivo identificar e localizar PANCs em imagens por meio do modelo YOLOv11, uma das versões mais recentes da arquitetura You Only Look Once (YOLO). Para o treinamento e o teste do modelo, utilizou-se a seed 0, em um total de 15.968 imagens e 19.037 instâncias, aplicando técnicas de aumento de dados e distribuídas entre 14 classes. As imagens foram divididas, de forma aleatória e estratificada, em 70% para treinamento, 20% para validação e 10% para teste. Todas as amostras foram obtidas do site Pl@ntNet. A eficácia do método proposto foi avaliada com base em métricas como precisão, recall, matriz de confusão, funções de perda, mean Average Precision (mAP) e mapas de calor, utilizando o algoritmo Eigen-CAM. Para este último, atentou-se às áreas mais quentes para avaliação dos locais de maior atenção do modelo, melhorando a sua explicabilidade. O modelo obteve precisão de 0,877, recall de 0,778, mAP50 de 0,878 e mAP50-95 de 0,801, enfrentando maiores dificuldades em classes semelhantes ou com grande variedade de características.*

1. Introdução

De acordo com [Kinupp and de Barros 2007], as Plantas Alimentícias Não Convencionais (PANCs) são aquelas que possuem uma ou mais partes comestíveis, podendo ser espontâneas

ou cultivadas, nativas ou exóticas, mas que não fazem parte do cardápio cotidiano. As PANCs também podem ser definidas como plantas que não são comuns na nossa alimentação, ou não são cultivadas em sistemas agrícolas convencionais (como a agricultura industrial). Elas são igualmente conhecidas como plantas alimentícias da agrobiodiversidade, conforme [Brack 2016].

O Brasil possui uma das maiores biodiversidades do mundo, e a utilização dessa riqueza natural na alimentação amplia a oferta de nutrientes à população, fortalecendo a soberania e a segurança alimentar. O consumo PANCs é uma maneira significativa de aproveitar essa diversidade, especialmente em pequenas comunidades rurais, promovendo uma alimentação mais sustentável [Tuler et al. 2019].

Portanto, é fundamental ressaltar que uma maior divulgação das PANCs permitiria que diversas áreas de estudo aproveitassem de forma mais eficiente suas qualidades naturais. Isso abriria espaço tanto para a promoção de métodos adequados de consumo dessas plantas, quanto para o avanço de pesquisas sobre seu potencial farmacológico. Nesse contexto, o uso de inteligência artificial (IA) para reconhecer essas plantas amplia ainda mais sua visibilidade, trazendo o tema para o campo da tecnologia e inovação.

Este artigo tem como objetivo desenvolver e apresentar um modelo para o reconhecimento e detecção de PANCs, abrangendo as espécies Aranto, Azedinha, Babosa, Beldroega, Camapu, Cana-da-Índia, Caruru, Chanana, Colônia, Noni, Ora-pro-nóbis, Saião, Taioba, e Urucum, presentes na Figura 1, utilizando o modelo YOLO11[Redmon 2016] como base. O YOLO é conhecido por sua eficiência em tarefas de detecção em tempo real, fazendo possível construir um sistema para identificação dessas plantas. Com este estudo, espera-se não apenas aumentar a visibilidade das PANCs, mas também introduzir o uso de IA como uma ferramenta para a identificação de espécies vegetais em diferentes ambientes, contribuindo para a pesquisa científica, a sustentabilidade alimentar e o fortalecimento da biodiversidade. Ao explorar uma área na interseção entre tecnologia e botânica, o projeto visa abrir novas oportunidades de aplicação de IA em estudos ecológicos, farmacológicos e de segurança alimentar, promovendo inovação na área e permitindo otimizar processos e fornecer suporte mais eficiente para a tomada de decisões.

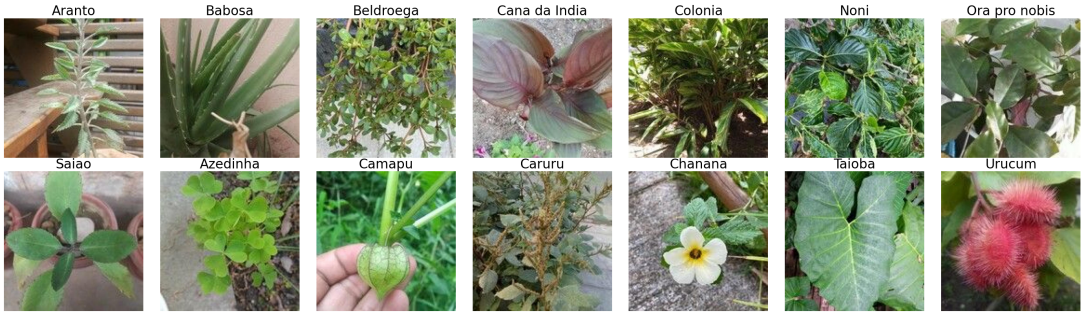

Figura 1. Exemplo representativo de imagem do conjunto de dados para cada espécie de planta.

2. Trabalhos Relacionados

Nesta seção, foram analisados artigos que apresentam semelhanças com o modelo proposto. O objetivo dessa análise foi identificar abordagens metodológicas, resultados e limitações relevantes para aprimorar o desenvolvimento do modelo de detecção de PANCs, e realizar uma comparação com o modelo desenvolvido neste estudo.

No estudo de [de Menezes Neto et al. 2021], o objetivo é utilizar IA para identificar PANCs, com ênfase em seu potencial para aplicação em práticas educativas. O trabalho para a criação da NEural IDEntifier (NEIDE) envolveu três etapas principais: a criação de uma base de dados com mais de 2500 fotos de PANCs, a aplicação de arquiteturas de redes neurais convolucionais (do inglês *Convolutional Neural Networks* - CNNs), para identificação das plantas e o ajuste de segurança do modelo para evitar o consumo de plantas indesejadas.

Para a elaboração do conjunto de dados, foram inicialmente tiradas 997 fotos ao longo de 28 dias, capturando diferentes fases do ciclo de vida das plantas e variadas condições climáticas. Após identificar semelhanças excessivas nas imagens, foram coletadas e rotuladas manualmente mais 1043 fotos da internet, além de mais 521 fotos seis meses depois, a fim de refletir mudanças sazonais. O banco de dados final incluiu 2561 fotos de dez espécies de plantas, sendo a Beldroega (Portulaca oleracea L.) e o Urucum (Bixa orellana L.) as únicas em comum com a base de dados utilizada neste estudo.

A etapa de treinamento utilizou a biblioteca fast.ai para aplicar técnicas de aumento de dados, incluindo inversão, rotação, zoom e ajustes de brilho e contraste. As CNNs escolhidas foram a ResNet [He et al. 2016] e a VGG [Simonyan and Zisserman 2014], reconhecidas por sua eficácia na identificação de plantas [Sun et al. 2017] e flores [Nguyen et al. 2016]. O melhor modelo do estudo alcançou uma acurácia de 96,35%. No entanto, um problema apontado no artigo de [de Menezes Neto et al. 2021] é que o modelo se limitava a classificar apenas entre as classes já presentes no conjunto de treinamento, mesmo quando a planta analisada não pertencia a nenhuma delas. Para solucionar essa limitação, foi adotado um nível de confiança para validar a inclusão de novas imagens na base de dados pelos usuários.

Apesar de não possuir o mesmo escopo, a pesquisa de [Sapkotaa et al. 2024] foi utilizada para avaliar algoritmos de detecção de objetos YOLO em 22 configurações dos modelos YOLOv8, YOLOv9, YOLOv10 e YOLOv11, aplicados à detecção de frutos verdes em pomares comerciais. Essa análise é relevante para este trabalho, pois permite observar o desempenho e a robustez dos algoritmos YOLO em tarefas de detecção em ambientes naturais, com alto grau de variação visual e complexidade, como também ocorre na identificação de PANCs. O estudo também validou a contagem de frutos no campo utilizando um iPhone e sensores de visão computacional em quatro variedades de maçãs: Scifresh, Scilate, Honeycrisp e Cosmic Crisp.

O estudo de [Tian et al. 2025] destaca a importância de aprimorar a arquitetura da rede do *framework* YOLO, um desafio que tem sido abordado ao longo do tempo com melhorias baseadas em CNNs, apesar da comprovada superioridade dos mecanismos de atenção em termos de capacidade de modelagem. Isso ocorre porque os modelos baseados em atenção ainda não alcançam a velocidade dos modelos baseados em CNNs. Para superar essa limitação, o estudo propõe um framework YOLO centrado em atenção, denominado YOLOv12, que iguala a velocidade das versões anteriores baseadas em CNNs, ao mesmo tempo em que aproveita os benefícios de desempenho dos mecanismos de atenção.

Embora o YOLOv12 já esteja disponível, os resultados apresentados ainda justificam o uso do YOLO11 para criar um modelo de classificação, devido ao seu desempenho em velocidade de inferência e precisão. No estudo de [Sapkotaa et al. 2024], citado anteriormente, o YOLO11n, em particular, se destacou por ser um modelo rápido, com um tempo de inferência de apenas 2,4 ms, tornando-o ideal para tarefas em tempo real, como a automação agrícola e a gestão de colheitas em estágios iniciais. Além disso, o YOLO11n demonstrou alta precisão na contagem e detecção de frutos verdes, com baixos valores de erro médio absoluto (MAE)

e erro quadrático médio (RMSE), em diferentes variedades de maçãs, destacando sua robustez em cenários de detecção complexos.

Logo, percebe-se que, embora o uso do YOLO em problemas de classificação já tenha uma trajetória consolidada, sua aplicação específica na detecção de PANCs ainda não foi explorada. Até o momento, o principal trabalho apresentado nessa área foi o apresentado por [de Menezes Neto et al. 2021], que se concentra no uso de CNNs, como ResNet e VGG, para o reconhecimento de plantas. O presente estudo, portanto, inova ao propor a utilização do YOLO11, uma versão avançada do algoritmo de detecção de objetos, com o objetivo de criar um modelo robusto e eficiente para identificar e classificar PANCs em diferentes cenários. Além disso, os cenários utilizados apresentam maior diversidade, tanto em termos da variedade de plantas, quanto na quantidade de amostras e cenários, proporcionando um aprendizado mais robusto e eficaz para o modelo.

3. Metodologia

Esta seção apresenta a metodologia aplicada no estudo, abrangendo a descrição do conjunto de dados, detalhes sobre o modelo YOLO utilizado, as técnicas de pré-processamento adotadas, as métricas de avaliação, bem como o ambiente de desenvolvimento.

3.1. Modelo

Para o desenvolvimento do modelo, foi utilizado o YOLO11. A escolha do YOLO se justifica por sua capacidade de realizar a classificação e a detecção de objetos de forma rápida e eficaz [Redmon 2016], essencial para o reconhecimento em tempo real das PANCs em ambientes variados. O modelo também permite a geração de caixas delimitadoras ao redor das plantas identificadas, auxiliando na localização precisa dos espécimes nas imagens. A arquitetura do YOLO11, ilustrada na Figura 2, representa um aprimoramento significativo em relação às arquiteturas anteriores [Wang and Liao 2024], tomando a arquitetura do YOLOv8 como base e introduzindo novas integrações e ajustes de parâmetros que aprimoram a capacidade de detecção e eficiência computacional do modelo.

Dentre as principais inovações, descritas por [Khanam and Hussain 2024], destacam-se os blocos C3k2 e C3k, que aumentam a flexibilidade no processamento de características ao permitir diferentes tamanhos de kernel, e o mecanismo C2PSA, que melhora a atenção espacial do modelo, ao captar informações posicionais de forma mais eficiente. Essas mudanças não apenas mantêm a tradição da família YOLO de oferecer modelos rápidos e precisos, mas também expandem as capacidades do YOLO11 para cenários mais complexos, especialmente no que diz respeito à detecção multiescalar e ao uso de mecanismos de atenção.

3.2. Base de dados e treinamento do modelo

Para criar uma base de dados diversificada, foi utilizada uma coleção de imagens do site Pl@ntNet [Pl@ntNet 2025], que abrange uma grande variedade de espécies vegetais. No entanto, como as imagens são enviadas por usuários, sem uma curadoria quanto à qualidade e padronização das identificações, pode haver inconsistências nos nomes científicos atribuídos às plantas. A pesquisa por nomes científicos, gêneros, famílias e características visuais facilitou a obtenção de imagens específicas para todas as PANCs selecionadas para o projeto. Para garantir a padronização dos nomes científicos apresentados no site, foi realizada uma verificação manual, comparando os nomes populares e científicos atribuídos às plantas em diversas fontes na internet. Esse cuidado foi essencial para minimizar erros de identificação e inconsistência

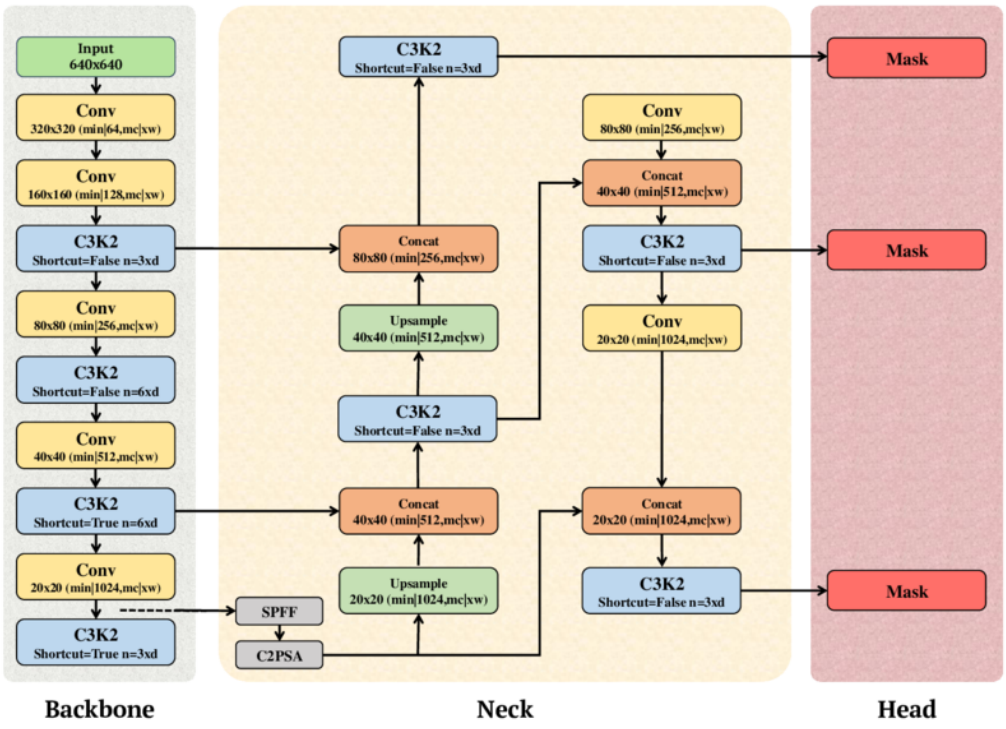

Figura 2. Arquitetura do YOLOv11

das informações na base de dados do projeto. Entretanto, ainda é necessária uma análise rigorosa da base de dados por um biólogo, o que significa que pode haver erros de identificação no conjunto de imagens criado.

Após a criação da base, realizamos um ajuste fino (*fine-tuning*) no modelo YOLO para otimizar o seu desempenho. O treinamento foi iniciado com o número máximo de 100 épocas, utilizando os hiperparâmetros padrões da Ultralytics, incluindo a semente (*seed*) 0, que também foi aplicada ao conjunto de teste. No entanto, com a aplicação da técnica de parada antecipada (*early stopping*), o processo foi interrompido na época 79, pois não houve melhora no desempenho no conjunto de validação após 10 épocas consecutivas. Ao final, foram disponibilizadas as curvas de aprendizado. Parâmetros como o tamanho do *batch*, otimizador, taxa de aprendizado e *momentum* também seguiram os valores padrões definidos pela Ultralytics, buscando obter o melhor aprendizado possível ao final do processo.

Para o treinamento, aproximadamente 70% das imagens foram utilizadas, enquanto 20% foram destinadas à validação e 10% ao teste, totalizando 15.968 imagens e 19.037 instâncias (ocorrências de objetos anotados dentro das imagens). Todas as imagens foram treinadas usando *RGB*, sendo divididas entre 14 classes: Aranto, Babosa, Beldroega, Canada-Índia, Colônia, Noni, Ora-pro-nóbis, Saião, Azedinha, Camapu, Caruru, Chanana, Taioba e Urucum. Para melhorar a capacidade de distinguir entre as plantas e o fundo, foram adicionadas imagens sem nenhuma das plantas anteriores, denominadas *background*, o que contribuiu para a redução de falsos positivos e para o aprimoramento da precisão das previsões, mesmo estando em menor número que as demais classes. Essa divisão visou proporcionar uma quantidade suficiente de dados para aprendizado, ao mesmo tempo em que reservamos conjuntos para avaliar seu desempenho, avaliando a ocorrência de *overfitting*. As imagens foram divididas aleatoriamente e, em seguida, estratificadas para evitar um possível viés do modelo em prever com maior frequência as classes que possuíam mais imagens, sendo todo o processo

realizado manualmente. A divisão de imagens por classe, juntamente com o total de imagens de *background*, pode ser observada na Figura 3.

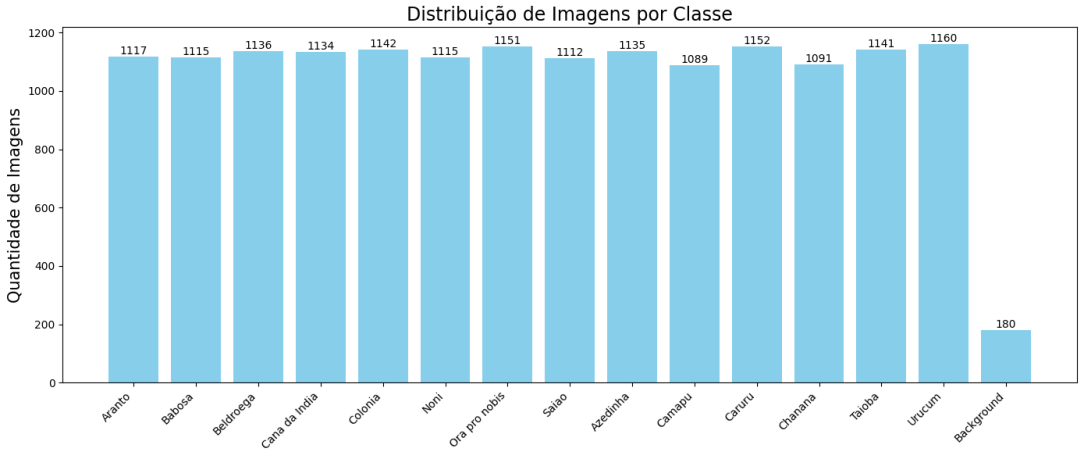

Figura 3. Distribuição da quantidade de imagens por espécie de planta, incluindo o total de imagens utilizadas como *background*.

3.3. Pré-processamento

Para garantir o uso eficaz do YOLO, foi necessário rotular as imagens das plantas, identificando a espécie presente e sua localização exata na imagem. Utilizou-se o site Roboflow, que permite criar um projeto personalizado e realizar a rotulação automática das imagens, sendo esta última opção utilizada para adiantar a criação de algumas caixas delimitadoras. No entanto, para evitar possíveis inconsistências, cada imagem rotulada foi revisada manualmente, corrigindo eventuais erros gerados pelo sistema automatizado.

Foram utilizados os recursos de pré-processamento das imagens oferecido pelo Roboflow. No projeto, optou-se pela orientação automática das imagens, ajustando-as conforme a rotulação, e pelo redimensionamento para 640x640 pixels, padronizando as imagens e facilitando o treinamento do modelo. Após esse processo, a base de dados rotulada foi disponibilizada para download, pronta para ser utilizada no modelo.

3.4. Aumento de Dados

Para aumentar a diversidade do conjunto de dados e melhorar a robustez do modelo, diversas transformações foram aplicadas às imagens durante o treinamento. O YOLO, por meio da biblioteca Ultralytics, utiliza *online data augmentation* [Enkvetchakul and Surinta 2022], ou seja, as imagens não são duplicadas, mas sim modificadas dinamicamente durante o processo de treinamento, gerando uma variedade de versões de cada imagem sem aumentar o tamanho do conjunto de dados. A função *augment* permite configurar uma série de parâmetros responsáveis por essas transformações. Entre eles, os seguintes foram utilizados, acompanhados de seus respectivos valores:

- **hsv_h** (Ajuste de tonalidade): 0,015 — Ajusta a tonalidade da imagem por uma fração da roda de cores, introduzindo variabilidade de cor.
- **hsv_s** (Ajuste de saturação): 0,7 — Altera a saturação da imagem por uma fração, afetando a intensidade das cores.
- **hsv_v** (Ajuste de brilho): 0,4 — Modifica o valor (brilho) da imagem por uma fração, ajudando o modelo a ter um bom desempenho em várias condições de iluminação.

- **translate** (Translação): 0,1 — Desloca a imagem horizontal e verticalmente por uma fração do seu tamanho, auxiliando na detecção de objetos parcialmente visíveis.
- **scale** (Escala): 0,5 — Redimensiona a imagem aplicando um fator de escala, simulando objetos em diferentes distâncias da câmera.
- **fliplr** (Espelhamento horizontal): 0,5 — Inverte a imagem da esquerda para a direita com a probabilidade especificada, útil para aprender objetos simétricos e aumentar a diversidade do conjunto de dados.
- **erasing** (Remoção aleatória): 0,4 — Apaga aleatoriamente uma parte da imagem durante o treinamento, incentivando o modelo a focar em características menos óbvias para o reconhecimento.

3.5. Ambiente de desenvolvimento

Para desenvolvimento do projeto, foi utilizada a linguagem de programação Python em sua versão 3.10.12, acompanhada das seguintes bibliotecas *open-source*:

- **Matplotlib** - versão **3.7.1**: responsável pela plotagem de dados.
- **Numpy** - versão **1.26.4**: responsável pelo processamento de dados multidimensionais e matrizes.
- **Pandas** - versão **2.2.2**: responsável pela manipulação e análise de dados.
- **Pillow (PIL)** - versão **10.4.0**: responsável pelo processamento e manipulação de imagens.
- **OpenCV** - versão **4.10.0**: voltada para visão computacional e processamento de imagens.
- **Ultralytics** - versão **8.3.17**: utilizado para detecção de objetos utilizando o modelo YOLO.
- **Seaborn** - versão **0.13.1**: baseada no Matplotlib, voltada para visualização de dados estatísticos.

O hardware utilizado para o treinamento do modelo foi o oferecido pelo Google Colab Pro+, com as seguintes especificações:

- **Processador**: Intel(R) Xeon(R) CPU @ 2.20GHz
- **Memória RAM**: 83GB
- **Placa de vídeo**: NVIDIA A100-SXM4-40GB

3.6. Métricas de avaliação

Para poder analisar adequadamente a capacidade do modelo em identificar PANCs, é importante estabelecer métricas a fim de realizar essa análise. Sendo assim, estão descritas a seguir as métricas utilizadas como forma de analisar o resultado apresentado pelo modelo.

3.6.1. Matriz de confusão normalizada

A matriz de confusão normalizada [Hardin and Shumway 1997] é uma ferramenta usada para avaliar o desempenho de modelos de classificação, especialmente quando se trabalha com problemas de múltiplas classes. Ela resume o número de previsões corretas e incorretas feitas pelo modelo, categorizando-as por classe. A matriz normalizada exibe os valores em porcentagens, facilitando a visualização do desempenho relativo do modelo em cada classe, independentemente do número absoluto de amostras por classe. Neste estudo, a matriz foi construída de modo a destacar a classificação dada ao objeto de maior confiança em uma imagem, mostrando se ela condiz com a classificação real dada à planta.

3.6.2. Precisão

A precisão representa a capacidade do modelo de identificar apenas os objetos relevantes. Indica a porcentagem de previsões positivas corretas feitas pelo modelo [Padilla et al. 2020]. É calculada como:

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{1}$$

- TP é o número de verdadeiros positivos;
- FP é o número de falsos positivos.

3.6.3. *Recall*

O *recall* representa a capacidade do modelo de encontrar todos os casos relevantes, ou seja, todas as caixas delimitadoras existentes no conjunto de verdade. Indica a porcentagem de previsões positivas corretas entre todas as anotações reais [Padilla et al. 2020]. É definido como:

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{2}$$

- TP é o número de verdadeiros positivos;
- FN é o número de falsos negativos.

3.6.4. *Mean Average Precision* (mAP)

A média da Precisão Média (*mean Average Precision* - mAP) é uma métrica utilizada para medir a precisão de detectores de objetos em todas as classes de um determinado banco de dados. O mAP é simplesmente a média da Precisão Média (*Average Precision* - AP) em todas as classes, ou seja:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i, \tag{3}$$

onde AP_i representa a Precisão Média da i-ésima classe, e N é o número total de classes avaliadas.

A AP é obtida considerando a precisão interpolada máxima $P_{\text{interp}}(R)$, cujo *recall* seja maior que R. Para avaliar a precisão e o *recall* das detecções em relação às caixas de referência (*ground truth*) presentes nas imagens, é necessário definir um limiar de interseção sobre união (*Intersection over Union* – IoU), denotado como t [Padilla et al. 2020]. Neste trabalho, foram adotadas duas métricas para o cálculo do mAP: o mAP50, que considera uma predição correta quando a IoU entre a caixa predita e a real é de pelo menos 50%, e o mAP50-95, que representa a média da precisão calculada em dez limiares de IoU, variando de 50% a 95% em incrementos de 5%, proporcionando uma avaliação mais completa e rigorosa do desempenho do modelo.

3.6.5. Funções de perda

O YOLO11 trabalha com três funções de perda para caixas delimitadoras orientadas (*Oriented Bounding Box*): a perda de caixa delimitadora (*Bounding Box Loss - box loss*), a perda de classificação (*Classification Loss - cls loss*) e a perda de distribuição focal (*Distribution Focal Loss - dfl loss*).

A *box loss* mede a diferença geométrica entre a caixa delimitadora prevista e a real (*ground truth*). Seu objetivo é garantir que as previsões sejam precisas em termos de posição, tamanho e ângulo de rotação do objeto, melhorando o desempenho geral da detecção. A *cls loss* avalia a capacidade do modelo de classificar corretamente os objetos detectados. Ela compara as categorias previstas com as reais, ajudando o modelo a diferenciar melhor os diversos alvos, mesmo em cenários onde há um desequilíbrio entre as classes. A *dfl loss* aprimora a precisão da regressão da caixa delimitadora, tornando a localização dos objetos mais precisa. Isso é especialmente útil em cenários complexos, onde os contornos dos objetos são irregulares. A função aprende a distribuição discreta das previsões para alinhar melhor os limites das caixas ao formato real do objeto [He et al. 2024].

3.6.6. Geração de Mapas de Calor com Eigen-CAM

Com o objetivo de aumentar a explicabilidade e identificar possíveis ambiguidades nas predições do modelo, foram gerados mapas de calor em algumas imagens. Para isso, utilizou-se o Eigen-CAM [Muhammad and Yeasin 2020], um método que permite visualizar quais regiões da imagem tiveram maior influência na tomada de decisão do modelo. Essa abordagem é útil para compreender como as características das plantas foram detectadas, facilitando a análise de possíveis erros e melhorando a interpretabilidade do modelo.

4. Resultados e Discussões

As análises foram feitas com base nos resultados das curvas de perda, na avaliação do modelo a partir do conjunto de testes e na matriz de confusão normalizada. Em seguida, foi realizada uma análise mais detalhada das plantas que apresentaram maiores erros no teste do modelo.

4.1. Curvas de perda

Na Figura 4, observa-se que as curvas de perda no treinamento apresentam uma tendência decrescente ao longo das épocas, indicando que o modelo está se ajustando aos dados e minimizando o erro na previsão. No entanto, as perdas na validação apresentam um comportamento diferente: após uma fase inicial de redução, estabilizam-se e mantêm valores superiores às perdas de treinamento, o que pode ocorrer devido ao conjunto de validação não representar tão bem os dados de treino, a alguma diferença entre os dois conjuntos ou, até mesmo, por ser apenas o início de um possível *overfitting*.

Analisando separadamente cada métrica de perda, a *box loss*, que mede a precisão na localização das caixas delimitadoras, exibe uma redução consistente, mas observa-se que, nas últimas épocas, a perda na validação se mantém acima da perda de treinamento, o que pode indicar que a generalização da localização das caixas ainda pode ser aprimorada.

A *cls loss*, responsável por quantificar os erros de classificação dos objetos, apresenta um forte decaimento inicial, estabilizando-se posteriormente, com uma diferença visível entre as curvas de treinamento e validação, o que pode sugerir que o modelo está aprendendo a

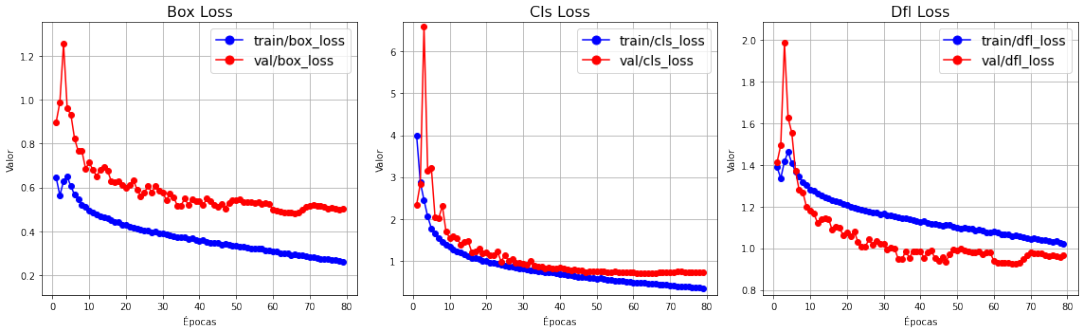

Figura 4. Curvas de perda para as diferentes funções de perda do modelo. Na primeira coluna, apresenta-se a perda da caixa delimitadora (*box loss*); na segunda, a perda de classificação (*cls loss*); e, na terceira, a perda da distribuição focal (*dfl loss*). As curvas de treinamento estão representadas em azul, enquanto as curvas de validação estão em vermelho.

classificar melhor os objetos, mas ainda enfrenta desafios em exemplos que diferem muito do conjunto de treino.

Já a *dfl loss*, que refina a distribuição das caixas preditas, apresenta um comportamento mais oscilatório, especialmente nas primeiras épocas, o que é esperado devido ao ajuste dos parâmetros iniciais. Com o avanço do treinamento, a perda oscila com menor frequência, mas a discrepância entre as curvas de validação e treinamento indica que o modelo pode não estar generalizando perfeitamente a qualidade do ajuste fino das caixas delimitadoras, possivelmente devido a variações na escala, perspectiva ou sobreposição dos objetos entre os conjuntos de treino e validação.

Apesar das diferenças entre as curvas de perda de treinamento e validação sugerirem que o modelo ainda poderia se beneficiar de ajustes adicionais ou de um tempo maior de treinamento, a análise geral indica que os pesos foram salvos em um ponto adequado. As perdas de validação permaneceram estáveis nas últimas épocas e os valores finais de erro são considerados baixos, o que demonstra que o modelo alcançou um bom nível de desempenho.

4.2. Avaliação do modelo

Por padrão, o YOLO usa um limiar de confiança mínimo de 0,001 e um limiar de IoU para supressão de não-máximos de 0,6. Isso faz com que as caixas delimitadoras que estão abaixo desse limiar não sejam consideradas para as predições na parte de avaliação. A avaliação do modelo foi realizada com um conjunto de teste de 1.586 imagens, resultando em um total de 1.862 instâncias. É importante destacar, também, que o conjunto de dados disponível para o modelo possui muitas imagens com uma única instância, o que pode enviesar o modelo a fazer menos predições em uma só imagem. As métricas de desempenho, apresentadas na Tabela 1, revelaram uma precisão média de 0,877 e uma taxa de *recall* de 0,778, com um mAP50 de 0,878 e um mAP50-95 de 0,801. Essa avaliação revelou variações significativas no desempenho entre as diferentes classes de PANCs.

Para uma avaliação mais simplificada, as classes com precisão e *recall* acima de 0,85 serão consideradas as de melhor desempenho, enquanto aquelas com *recall* inferior a 0,75 serão classificadas como as de pior desempenho. O *recall* foi escolhido como critério para essa última avaliação por ser a métrica que indica se o modelo está identificando corretamente as instâncias presentes na imagem.

As plantas que apresentaram bom desempenho foram a **Beldroega** e a **Ora-pro-nóbis**,

Tabela 1. Métricas de desempenho por classe

Classe	Precisão	Recall	mAP50	mAP50-95
Todas	0,877	0,778	0,878	0,801
Aranto	0,844	0,751	0,830	0,758
Babosa	0,933	0,780	0,901	0,827
Beldroega	0,933	0,856	0,934	0,843
Cana da Índia	0,922	0,832	0,938	0,862
Colônia	0,842	0,843	0,887	0,838
Noni	0,911	0,767	0,872	0,774
Ora-pro-nóbis	0,933	0,862	0,948	0,882
Saião	0,886	0,812	0,904	0,857
Azedinha	0,853	0,785	0,881	0,801
Camapu	0,871	0,667	0,839	0,787
Caruru	0,799	0,811	0,875	0,820
Chanana	0,873	0,774	0,885	0,813
Taioba	0,809	0,703	0,815	0,710
Urucum	0,876	0,648	0,779	0,637

com precisão de 0,933 e 0,933 e *recall* de 0,856 e 0,862, respectivamente. Isso indica que o modelo possui alta precisão na maioria das caixas delimitadoras, além de uma boa correspondência entre as identificações feitas pelo modelo e as verdadeiras. Além disso, o mAP50 de 0,934 e 0,948 e o mAP50-95 de 0,843 e 0,882, apresentados por **Beldroega** e **Ora-pro-nóbis**, respectivamente, evidenciam a precisão das detecções do modelo para essas duas plantas. Esses resultados reforçam a robustez do modelo na identificação dessas classes, demonstrando que ele mantém uma boa precisão em uma variedade de condições, como pode ser visto nas Figuras 5 e 6.

Figura 5. Predições feitas para Beldroega

Figura 6. Predições feitas para Ora-pro-nóbis

As plantas com pior desempenho foram o **Urucum**, a **Taioba** e o **Camapu**, com *recall* de 0,648, 0,703 e 0,667, respectivamente, indicando dificuldade na detecção de todas as instâncias dessas classes nas imagens da base de dados. Algo que reforça essa observação é o fato de que o mAP50-95 do **Urucum** e da **Taioba** foi de 0,637 e 0,710, respectivamente, evidenciando que, à medida que o limiar do mAP aumenta, a confiança do modelo na predição dessas classes tende a diminuir. O **Camapu** foi o único a apresentar um resultado um pouco mais elevado, com mAP50-95 de 0,787. Um exemplo de predições incorretas do **Urucum** e da **Taioba** podem ser observadas nas Figuras 7 e 8, enquanto exemplos de predições incorretas do **Camapu** são apresentadas e explicadas mais adiante.

Figura 7. Predições erradas do Urucum

Figura 8. Predições erradas da Taioba

Em suma, os resultados apresentados revelam que o modelo demonstrou uma boa capacidade de detecção em classes como **Ora-pro-nóbis** e **Beldroega**, que se destacaram por suas altas métricas de precisão e *recall*. Isso indica que o modelo é eficaz na identificação dessas PANCs, o que pode ter implicações significativas para promover o consumo e o estudo dessas plantas.

Contudo, as classes **Urucum** e **Taioba** evidenciam a necessidade de melhorias, especialmente em condições variadas, sugerindo que ajustes no modelo e a inclusão de dados mais diversificados podem aprimorar a eficácia geral. A análise abrangente das métricas não apenas valida a utilidade do modelo na detecção de PANCs, mas também orienta futuros esforços para otimizar a precisão e a robustez nas aplicações práticas.

4.3. Matriz de Confusão Normalizada

Em uma matriz de confusão normalizada, as linhas representam as classes preditas pelo modelo, e as colunas indicam as classes verdadeiras. Os valores nas células representam as proporções de predições corretas e incorretas, com maior intensidade de cor indicando melhor desempenho em uma determinada classe. A seguir, serão expostas as informações mais importantes e que mais se destacaram na matriz.

A análise da matriz de confusão, apresentada na Figura 9, revela um desempenho geral satisfatório do modelo, com a grande maioria das plantas apresentando valores acima de 90%. No entanto, ainda há dificuldades na distinção entre classes, com destaque para o **Aranto** e o **Camapu**, que são as duas plantas com as maiores porcentagens de erro, o que mostra que o modelo pode não classificar corretamente a planta presente em uma imagem.

No caso do **Aranto**, isso se dá principalmente por conta do erro de predição da planta como **Saião**, totalizando 6% das predições totais. Esse erro ocorre devido à semelhança entre as duas plantas, com as bordas sendo a principal característica em comum, ambas apresentando formato ondulado. Com o **Camapu**, houve uma porcentagem de erro ainda maior, sendo confundido com o **Caruru** em 9% das predições. Nesse caso, não há uma semelhança tão evidente entre as duas plantas, porém o modelo mostra uma clara tendência a cometer esse erro, o que pode estar relacionado ao fato de ambas apresentarem estruturas destacadas, como fruto no

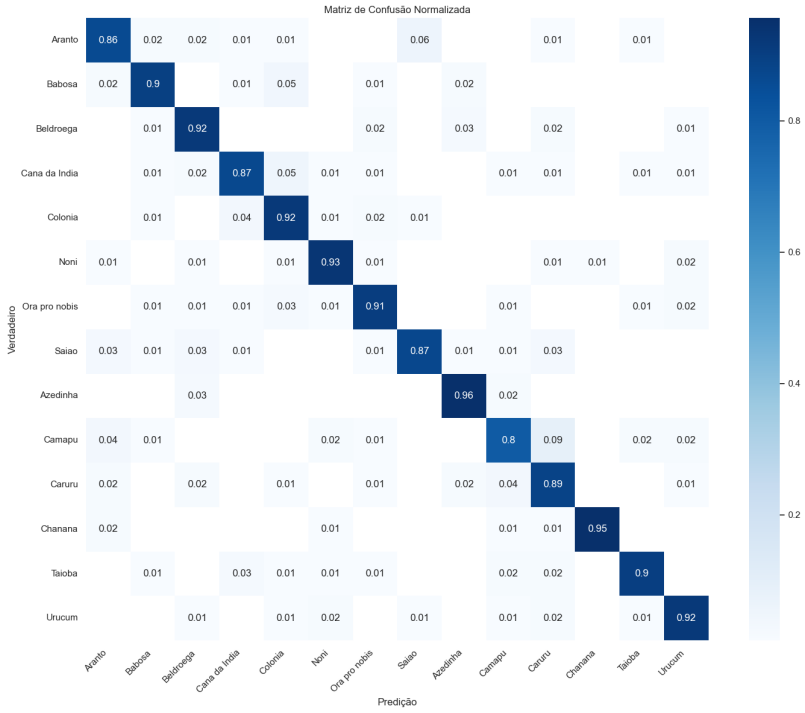

Figura 9. Matriz de confusão normalizada para a classificação de PANCs.

caso do **Camapu** e espiga no caso do **Caruru**, o que pode levar à confusão. Alguns desses erros de **Aranto** e **Camapu** podem ser vistos nas Figuras 10 e 11, respectivamente.

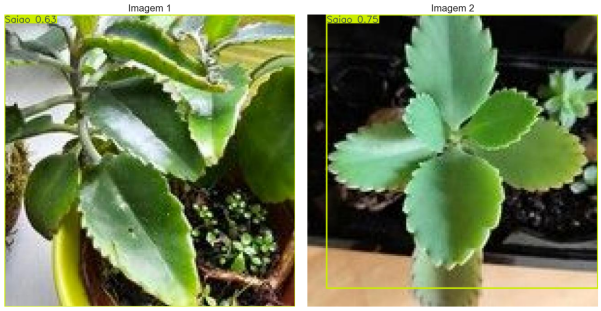

Figura 10. Predições erradas do Aranto

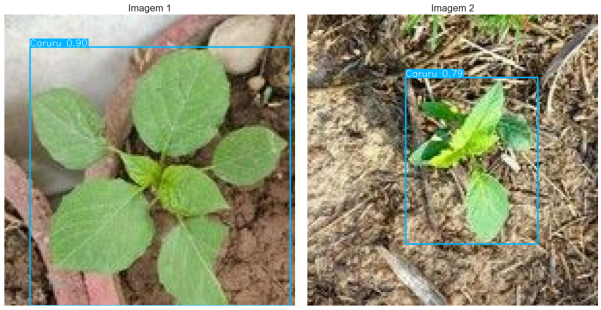

Figura 11. Predições erradas do Camapu

Melhorias futuras podem focar no aumento e diversificação do conjunto de dados, principalmente para as classes mais problemáticas como **Aranto** e **Camapu**, que apresentam erros

maiores. Outra possibilidade é o uso de técnicas como *ensembles* de modelos, que podem combinar diferentes modelos para melhorar a robustez das predições e reduzir os erros observados.

4.4. Classes problemáticas

Entre todas as classes, as que apresentaram mais problemas, de modo geral, foram **Urucum**, **Taioba**, **Camapu** e **Aranto**. Nesta seção, faremos uma análise mais detalhada de cada uma dessas classes.

O **Urucum** apresentou o pior *recall* entre todas as classes, como observado na avaliação do modelo, mostrando que a planta enfrenta dificuldades em reconhecer todas as instâncias verdadeiras em uma imagem. Esse problema ocorre devido às diversas formas em que o **Urucum** pode se apresentar no ambiente. Como ilustrado na Figura 12, a planta pode aparecer de diferentes maneiras, exibindo folhagem, frutos e flores, com essas características de maneira isolada ou com todas em conjunto. Além disso, seus frutos podem ter três colorações distintas (verde, vermelho e marrom), o que exige que o modelo reconheça múltiplas características dentro da mesma classe, tornando a predição correta de todas as instâncias mais difícil.

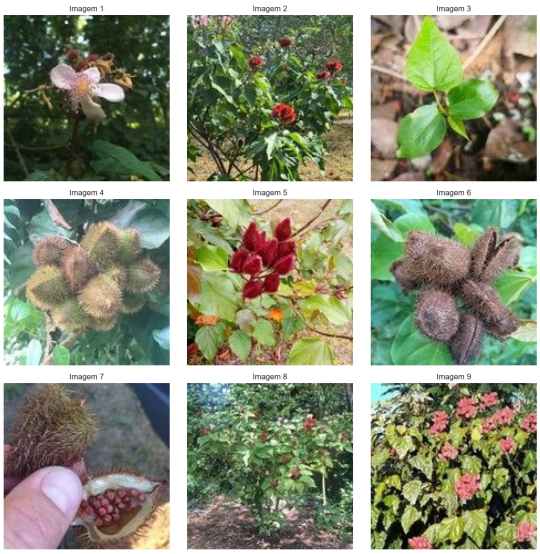

Figura 12. Exemplos de Urucum presentes na base de dados.

No caso da **Taioba**, a planta é popularmente conhecida por suas folhas extremamente grandes, que podem atingir até 65,8 cm de comprimento e 53,6 cm de largura em ambientes de cultivo com restrição de luz [Colombo et al. 2018], algo que não se destaca tanto em uma imagem. Além disso, a planta não apresenta um padrão constante na disposição de suas folhas, como pode ser observado na Figura 13, o que reduz a taxa de predições corretas para suas instâncias e dificulta ainda mais a detecção pelo modelo. Isso ocorre, principalmente, porque a **Taioba** possui pequenas variações visuais que podem não ser evidentes para um leigo no assunto, tornando difícil a diferenciação entre a **Taioba** comestível e a não comestível, por exemplo, o que pode ter enviesado negativamente o conjunto de dados criado para a planta.

Com o **Aranto** e o **Camapu**, o principal problema está na confusão com outras classes, com destaque para o **Saião** e o **Caruru**, respectivamente. No caso do **Aranto**, sua semelhança com o **Saião** dificulta a detecção, especialmente nos estágios iniciais da planta. O Aranto é caracterizado pelo surgimento de pequenas mudas ao longo da borda de suas folhas, mas, nos estágios iniciais, essas mudas ainda não estão presentes, fazendo com que a borda da planta se assemelhe à do Saião. Para proporcionar uma melhor explicabilidade, as Figuras 14 e 15

Figura 13. Exemplos de Taioba presentes na base de dados.

apresentam duas predições incorretas do **Aranto** como **Saião**, seguidas de um mapa de calor e de uma imagem representando como realmente é o **Saião**. Na Figura 14, pode-se perceber que o modelo dá maior importância às bordas da imagem, sem considerar informações importantes, favorecendo o erro de predição apresentado. Entretanto, na Figura 15, mesmo com o modelo focando em partes relevantes da planta, ainda é feita uma predição incorreta do **Aranto** como **Saião**, o que mostra que, mesmo com informações da planta, o modelo ainda pode confundir as duas espécies.

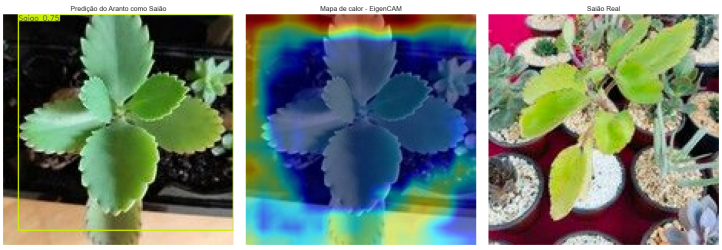

Figura 14. Predição errada do Aranto como Saião, seguido de mapa de calor e imagem real do Saião.

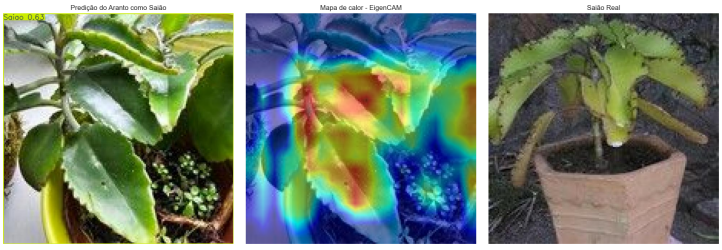

Figura 15. Predição errada do Aranto como Saião, seguido de mapa de calor e imagem real do Saião.

No caso do **Camapu**, os exemplos em que a planta aparece sozinha, sem seu fruto, apresentam maior dificuldade para o modelo distinguir o **Camapu** do **Caruru**. Nas Figuras

16 e 17, é possível perceber, por meio do mapa de calor, que o modelo tende a dar mais importância ao redor da planta. Essa característica faz com que o modelo não capture nenhuma informação realmente importante, mostrando que, quando se atenta mais às regiões ao redor de um **Camapu**, ele tende a predizer que a planta apresentada é um **Caruru**, evidenciando que o conjunto de dados utilizado não foi suficiente para que o modelo aprendesse a focar nas partes relevantes de um **Camapu** em todos os casos.

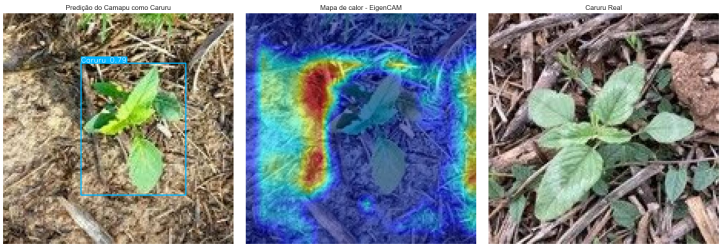

Figura 16. Predição errada do Camapu como Caruru, seguido de mapa de calor e imagem real do Caruru.

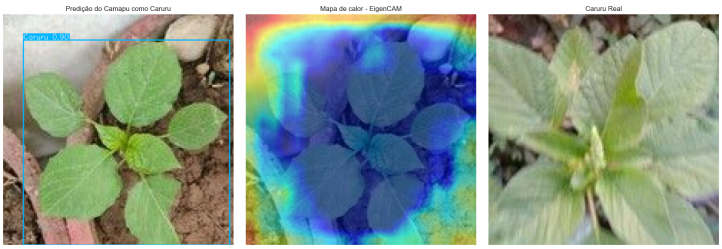

Figura 17. Predição errada do Camapu como Caruru, seguido de mapa de calor e imagem real do Caruru.

Esses fatos mostram que, embora a base de dados criada possua quase 16.000 imagens de treinamento, ainda não há um desempenho consistente em todas as classes. Isso evidencia a necessidade de um conjunto de imagens com maior variedade, incluindo a repetição de casos específicos e a inserção de exemplos que representem melhor situações que divergem significativamente dos dados de treinamento. Esse ajuste proporcionaria ao modelo mais oportunidades para aprender padrões distintos, especialmente em cenários visuais menos representados, aumentando sua capacidade de generalização e melhorando a precisão das predições. Além disso, seria necessária uma revisão criteriosa do conjunto de dados por parte de um profissional da área, para que, quando o modelo for disponibilizado para uso público, haja um maior grau de confiança nas informações utilizadas no treinamento e sejam evitados possíveis problemas em seu uso pelo público.

5. Considerações finais

Neste trabalho, foi desenvolvido um modelo capaz de identificar PANCs, com o objetivo de proporcionar maior visibilidade ao tema, tanto no meio acadêmico, quanto entre a população. Para isso, foi utilizado o modelo YOLO, mais especificamente o YOLO11, devido à sua alta precisão e bom desempenho em ambientes desafiadores. Além disso, foi realizado um ajuste fino para aprimorar o reconhecimento de PANCs pelo modelo, além do uso da técnica de aumento de dados para aumentar a robustez do modelo.

O modelo apresentou precisão e *recall* médios de 0,877 e 0,778, mas apresentaram certa variação entre as classes, como visto na matriz de confusão e nos gráficos de métricas e

perdas, como também na avaliação do modelo. As curvas de perda (*loss*) mostram uma queda constante nas perdas de caixa delimitadora, classificação e distribuição focal, no conjunto de treinamento, enquanto o conjunto de validação se estabiliza após um tempo até parar, quando não há melhoras por 10 épocas seguidas, mostrando que o modelo começou a decorar os dados do treinamento.

A avaliação do modelo foi realizada com um conjunto de 1.586 imagens e um total de 1.862 instâncias. Como foi possível observar na Tabela 1, as métricas de desempenho revelaram uma precisão de 0,877, *recall* de 0,778, conforme mencionado anteriormente, mAP50 de 0,878 e mAP50-95 de 0,801. A avaliação evidenciou variações significativas no desempenho entre as diferentes classes de PANCs, com plantas como **Beldroega** e **Ora-pro-nóbis** apresentando precisão e *recall* acima de 0,85. Por outro lado, **Camapu**, **Taioba** e **Urucum** apresentaram *recall* abaixo de 0,71, o que evidencia a dificuldade de classificar corretamente todas as instâncias dessas classes em uma única imagem. Isso, em conjunto com a análise da matriz de confusão normalizada, presente na Figura 9, revelou que classes como **Aranto** e **Camapu** apresentaram dificuldades na predição feita pelo modelo. O **Aranto** e o **Camapu** foram confundidos com o **Saião** e o **Caruru**, respectivamente.

Possíveis melhorias futuras incluem a ampliação e diversificação do conjunto de dados, especialmente para classes problemáticas como **Aranto**, **Camapu**, **Urucum** e **Taioba**, nas quais a taxa de erro é mais elevada. Estratégias como o treinamento com imagens em escala de cinza e o uso de mais imagens com múltiplas instâncias e múltiplas classes também podem contribuir para um desempenho mais robusto. Além disso, técnicas como o aumento de dados com imagens sintéticas, o uso de redes especializadas, a adoção de modelos em conjunto (*ensembles*) e a aplicação de validação cruzada (*k-fold*), utilizada para gerar resultados mais confiáveis e avaliar melhor a capacidade de generalização, podem ser exploradas para aprimorar os resultados, especialmente nas classes mais desafiadoras. Também seria necessária uma revisão mais rigorosa do conjunto por um biólogo, corrigindo possíveis inconsistências ainda existentes, o que contribuiria para a utilização do modelo como ferramenta educativa e permitiria maior confiabilidade em aplicações práticas. Por fim, com uma base mais consistente, a realização de testes em dispositivos móveis seria uma alternativa interessante, visando facilitar o acesso ao conteúdo gerado pelo modelo.

Referências

Brack, P. (2016). Plantas alimentícias não convencionais. *Agriculturas*, 13:4–6.

Colombo, J. N., Puiatti, M., Altoé, L. M., Haddade, I. R., and Vieira, J. C. B. (2018). Desempenho da taioba cultivada sob diferentes materiais de cobertura. *Revista Brasileira de Ciências Agrárias*, 13(4):1–8.

de Menezes Neto, E. J., de Lima, D. G., Feitosa, I. S., Gomes, S. M., and Jacob, M. C. M. (2021). Plant identification using artificial intelligence: Innovative strategies for teaching food biodiversity. In *Local food plants of Brazil*, pages 379–393. Springer.

Enkvetchakul, P. and Surinta, O. (2022). Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition. *Applied Science and Engineering Progress*, 15(3):3810–3810.

Hardin, P. J. and Shumway, J. M. (1997). Statistical significance and normalized confusion matrices. *Photogrammetric engineering and remote sensing*, 63(6):735–739.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, L.-h., Zhou, Y., Liu, L., and Zhang, Y.-q. (2024). Research on the directional bounding box algorithm of yolov11 in tailings pond identification. *Available at SSRN 5055415*.

Khanam, R. and Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.

Kinupp, V. F. and de Barros, I. B. I. (2007). Riqueza de plantas alimentícias não-convencionais na região metropolitana de porto alegre, rio grande do sul. *Revista Brasileira de Biociências*, 5(S1):63–65.

Muhammad, M. B. and Yeasin, M. (2020). Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE.

Nguyen, T. T. N., Le, V., Le, T., Hai, V., Pantuwong, N., and Yagi, Y. (2016). Flower species identification using deep convolutional neural networks. In *AUN/SEED-Net Regional Conference for Computer and Information Engineering*.

Padilla, R., Netto, S. L., and Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE.

Pl@ntNet (2025). `https://identify.plantnet.org/pt-br`.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sapkotaa, R., Mengb, Z., Churuvijaa, M., Dub, X., Mab, Z., and Karkeea, M. (2024). Comprehensive performance evaluation of yolo11, yolov10, yolov9 and yolov8 on detecting and counting fruitlet in complex orchard environments.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sun, Y., Liu, Y., Wang, G., and Zhang, H. (2017). Deep learning for plant identification in natural environment. *Computational intelligence and neuroscience*, 2017(1):7361042.

Tian, Y., Ye, Q., and Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

Tuler, A. C., Peixoto, A. L., and Silva, N. C. B. d. (2019). Plantas alimentícias não convencionais (panc) na comunidade rural de são josé da figueira, durandé, minas gerais, brasil. *Rodriguésia*, 70:e01142018.

Wang, C.-Y. and Liao, H.-Y. M. (2024). Yolov1 to yolov10: The fastest and most accurate real-time object detection systems. *arXiv preprint arXiv:2408.09332*.