

CLASSIFICAÇÃO E MEDIÇÃO AUTOMÁTICA DE CASTANHAS DE CAJU COM YOLOV8: UMA FERRAMENTA AUXILIAR PARA AVALIAÇÃO NÃO INVASIVA

Raul Vinicius Morais de Oliveira*
Alexandro Lima Damasceno**

RESUMO

A castanha de caju é um produto de grande relevância econômica para o Brasil, especialmente no contexto das exportações agrícolas. Em 2024, a produção nacional atingiu 161 mil toneladas, impulsionada principalmente pelo crescimento no estado do Ceará. A colheita ocorre entre setembro e abril, exigindo avaliações rápidas e precisas dos lotes, tradicionalmente realizadas de forma visual e subjetiva. Nesse cenário, a inteligência artificial (IA) surge como alternativa para aprimorar a avaliação, proporcionando maior eficiência e padronização. Este trabalho propôs o desenvolvimento de um algoritmo baseado em IA para a classificação e medição de castanhas de caju, utilizando métodos de visão computacional. O modelo adotado foi o YOLO (You Only Look Once), devido à sua alta eficiência na detecção em tempo real e à capacidade de aprendizado com poucos dados. A solução inclui a conversão de medidas de pixel para centímetros, permitindo estimar o tamanho das castanhas com base em um objeto de referência. O modelo foi treinado em duas etapas, totalizando 60 épocas, explorando tanto o desenvolvimento *from scratch* quanto o uso de *transfer learning*. Apresentou alta precisão e boa capacidade de generalização, com alguma confusão entre as classes “avariadas” e “castanha”, atribuída à semelhança entre algumas instâncias. A detecção da classe “réguas” viabiliza a inferência do tamanho das castanhas. No entanto, a partir do limiar de confiança de 70%, apesar de alcançar precisão superior a 80%, o *recall* varia de 0% a 90%, comprometendo a contagem precisa de objetos. Isso limita a aplicação do modelo em tarefas de quantificação automática.

Palavras-chave: Castanha de caju *in natura*; Inteligência artificial; YOLO; Avaliação não invasiva; Aprendizado de máquina.

ABSTRACT

Cashew nuts are of great economic relevance to Brazil, especially in the context of agricultural exports. In 2024, national production reached 161 thousand tons, driven mainly by growth in the state of Ceará. The harvest takes place between September and April and requires quick and

* Aluno do Bacharelado em Ciência da Computação do IFCE campus Aracati

** Professor do Bacharelado em Ciência da Computação do IFCE campus Aracati

accurate assessments of the batches, traditionally performed visually and subjectively. In this context, artificial intelligence (AI) emerges as an alternative to improve the evaluation process, offering greater efficiency and standardization. This work proposed the development of an AI-based algorithm for the classification and measurement of cashew nuts using computer vision methods. The selected model was YOLO (You Only Look Once), due to its high efficiency in real-time detection and its ability to learn from limited data. The solution includes the conversion of pixel measurements to centimeters, allowing for size estimation of the cashew nuts based on a reference object. The model was trained in two stages, totaling 60 epochs, exploring both development *from scratch* and the use of *transfer learning*. It demonstrated high accuracy and good generalization capacity, with some confusion between the “damaged” and “cashew” classes, attributed to the similarity between some instances. The detection of the “ruler” class enables the inference of the nut sizes. However, from a confidence threshold of 70%, although the model achieves over 80% precision, the *recall* varies from 0% to 90%, compromising accurate object counting. This limits the model’s applicability for automatic quantification tasks.

Keywords: Cashew nut *in natura*; Artificial intelligence; YOLO; Non-invasive evaluation; Machine learning.

1 INTRODUÇÃO

A castanha de caju é um produto de grande importância econômica para o Brasil, destacando-se principalmente entre as exportações agrícolas do país. A produção de castanha de caju tem apresentado perspectivas de crescimento. Segundo (Instituto Brasileiro de Geografia e Estatística (IBGE), 2025), a produção nacional em 2024 alcançou 161 mil toneladas. Esse incremento deve-se, em grande parte, ao crescimento observado no estado do Ceará — principal produtor do país —, responsável por aproximadamente 101,9 mil toneladas da produção nacional. Esse volume representa um aumento de 38,79% em relação ao registrado no ano anterior. De acordo com (Instituto Brasileiro de Geografia e Estatística (IBGE), 2024), a produção nacional em 2023 foi de 116 mil toneladas.

No Brasil, o período de colheita e comercialização começa entre os meses de setembro e outubro e perdura até o mês de abril. A alta sazonalidade da produção, aliada à elevada demanda internacional, condiciona a indústria a realizar compras em grande volume para formação de estoque em um curto período de tempo. Esse cenário torna inviável uma avaliação precisa e detalhada das castanhas (LIMA A. C.; VIDAL NETO, 2022).

O trabalho de (LIMA A. C.; VIDAL NETO, 2022) demonstra que a compra ainda envolve a participação de um profissional qualificado encarregado de precisar o lote a partir de uma inspeção de sua matéria-prima. Essa inspeção é predominantemente visual e baseada em critérios não invasivos, como o tamanho, o formato, a coloração, a integridade física e a ausência de impurezas. Contudo, esse método é suscetível a limitações, como subjetividade na análise,

inconsistências nos resultados e maior tempo de execução, especialmente em grandes volumes.

Nesse contexto, os trabalhos de (BANDEIRA et al., 2023) e (BORBA et al., 2022) abordam como a inteligência artificial (IA) tem sido amplamente aplicada para auxiliar diversos setores da agricultura em diferentes países. Métodos de IA, como os de *deep learning*, já são utilizados na contagem de frutos e na detecção de doenças. A capacidade da IA de processar grandes volumes de dados com rapidez e precisão abre possibilidades para reduzir a subjetividade da análise visual dos lotes de castanhas, tornando o processo mais ágil e confiável.

Este trabalho propõe o desenvolvimento de um algoritmo de classificação de castanhas de caju, baseado em um modelo de inteligência artificial, capaz de detectar, reconhecer e mensurar o tamanho das castanhas, utilizando como referência um objeto de tamanho conhecido presente na imagem; neste caso, utiliza-se uma régua. A solução poderá ser empregada como ferramenta para auxiliar profissionais na avaliação visual não invasiva, contribuindo para critérios mais precisos e objetivos. Para isso, será necessário escolher um modelo eficiente para detecção e classificação em vídeos e fotos, criar um banco de dados com imagens de castanhas e réguas e, por fim, treinar e avaliar o modelo.

A estrutura do artigo está organizada de forma a abordar os tópicos principais em seções distintas. A Seção 2 apresenta uma revisão teórica concisa, explicando conceitos fundamentais de inteligência artificial e suas conexões com o projeto desenvolvido. Na Seção 3, são discutidos trabalhos relacionados, analisando estudos prévios realizados por outros autores sobre temas semelhantes. A Seção 4 detalha a metodologia empregada para implementar a proposta apresentada. Na Seção 5, são apresentados os resultados obtidos após o treinamento da rede neural convolucional. Por fim, a Seção 6 conclui o trabalho com uma síntese dos principais achados e indica direções para pesquisas futuras.

2 REFERENCIAL TEÓRICO

Para melhor compreensão sobre o tema abordado, esta seção apresenta conceitos sobre Inteligência Artificial (IA), Aprendizado de Máquina, Redes Neurais Artificiais, *Deep Learning*, e *Convolutional Neural Networks* (CNN), ferramentas utilizadas no sistema.

2.1 Inteligência Artificial

Como demonstrado por Russell e Norvig (2021), o conceito de Inteligência Artificial evoluiu ao longo da história, seguindo diferentes abordagens e refletindo distintas compreensões científicas sobre inteligência. Enquanto alguns pesquisadores a definem pela capacidade de reproduzir o desempenho humano, outros adotam uma visão mais abstrata e formal, baseada na racionalidade.

O trabalho de (GOMES, 2023) aborda o Teste de Turing, proposto por Alan Turing em 1950. O teste possui três participantes: dois humanos — um interrogador e outro que responderá às perguntas — e uma máquina, que também será interrogada. O interrogador não sabe quem está respondendo às perguntas e, em sua segunda versão, não sabe sequer que há uma máquina

entre os participantes. Para passar no teste, a máquina deve enganar o interrogador humano, fazendo-se passar por uma pessoa. Gomes conclui que, ao passar no teste, a máquina demonstra ser capaz de simular ou imitar aspectos do comportamento humano que requerem inteligência.

Além disso, Russell e Norvig (2021) citam Bellman, que define a Inteligência Artificial como o campo dedicado ao estudo e criação de agentes capazes de operar autonomamente, adaptar-se, compreender o ambiente e buscar o melhor resultado para atingir um objetivo. Esse paradigma, predominante na área, constitui o modelo padrão. Assim, a IA possibilita que máquinas realizem operações complexas, como reconhecer padrões e resolver problemas sem soluções predefinidas, tomando sempre a melhor decisão com base nos dados disponíveis.

2.2 Aprendizado de Máquina

Russell e Norvig (2021) explicam que o aprendizado é a capacidade de um agente extrair uma regra geral a partir de observações feitas no ambiente em que está inserido. No contexto do aprendizado de máquina, esse agente é um computador que, ao analisar dados, gera uma regra, constrói um modelo com base nela e usa esse modelo para resolver problemas.

Embora o computador — considerado o agente no contexto do aprendizado de máquina — possa receber como entrada qualquer tipo de estrutura de dados, os problemas de aprendizado podem ser divididos em dois tipos principais: classificação, quando a saída pertence a um conjunto finito de valores, e regressão, quando a saída é um número, representado por um valor inteiro ou real. Além disso, existem diferentes tipos de feedback que podem acompanhar as entradas, os quais determinam os principais tipos de aprendizado (RUSSELL; NORVIG, 2021).

No aprendizado supervisionado, o agente recebe um conjunto de entradas, cada uma associada a um rótulo de saída, permitindo que desenvolva uma função para o problema (ALZU-BAIDI et al., 2021). No aprendizado não supervisionado, o agente identifica padrões nos dados de entrada e realiza a tarefa sem depender de feedback explícito (FLECK; BENEDET; SILVA, 2016). A técnica mais comum, segundo Russell e Norvig (2021), é o clustering, que organiza os dados em grupos com características semelhantes.

O aprendizado por reforço ocorre quando o agente aprende por meio de reforços, recebendo recompensas ao atingir seus objetivos e punições ao cometer erros (OPENAI et al., 2019). Conforme descrito por Russell e Norvig (2021), o agente deve identificar quais ações realizadas levaram ao resultado final e ajustar seu comportamento para maximizar as recompensas futuras. Por fim, o aprendizado por transferência permite que o agente utilize o conhecimento de uma tarefa anterior para aprender uma nova de maneira mais eficiente, especialmente quando ambas são relacionadas, exigindo menos exemplos para a adaptação Russell e Norvig (2021).

2.2.1 *Redes Neurais Artificiais*

De forma geral, uma rede neural artificial é um sistema projetado para imitar um cérebro humano, desde sua forma até a maneira como realiza determinadas tarefas e sua topologia, que é uma clara analogia a uma rede neural natural. Esse tipo de rede é capaz de aprender por meio

da interação com o ambiente em que está inserida, adquirindo conhecimento progressivamente. A adaptação da rede ao problema ocorre por meio das conexões entre os neurônios, que são alteradas em decorrência da aprendizagem (FLECK; BENEDET; SILVA, 2016).

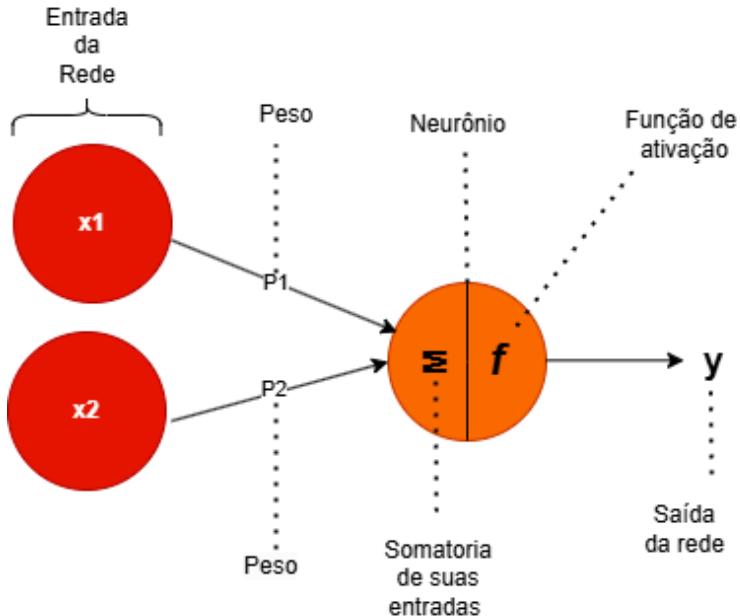
Em 1958, Frank Rosenblatt desenvolveu a rede Perceptron, cuja topologia simples é composta por três camadas. A primeira camada recebe as entradas, que são transmitidas para a segunda camada por meio de sinapses com pesos ajustáveis, onde um neurônio processa essas informações. Por fim, o neurônio gera uma saída que corresponde à terceira camada. Sua topologia simples a limitava à resolução de problemas linearmente separáveis (GOUVEIA, 2012). A topologia da rede é representada na Figura 1. Gouveia (2012) demonstra que, posteriormente, a fim de resolver problemas mais complexos, foi desenvolvida a Perceptron de Multicamadas (MLP), que, graças à presença de uma ou mais camadas ocultas, é capaz de resolver problemas não linearmente separáveis. Sua topologia, mais avançada em comparação ao Perceptron, é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. A Figura 2 ilustra essa topologia.

Embora a topologia de cada rede possa variar bastante, podemos classificar três classes de arquiteturas de rede fundamentalmente diferentes: redes alimentadas para frente com camada única, redes alimentadas para frente com múltiplas camadas e redes recorrentes. Essas arquiteturas normalmente são compostas por alguns componentes essenciais:

- **Neurônio:** Unidade de processamento da rede responsável por processar as entradas em conjunto com os pesos das sinapses e os valores de bias, gerando uma saída que pode ser utilizada como entrada para outro neurônio ou como o resultado final da rede (FLECK; BENEDET; SILVA, 2016).
- **Conjunto de sinapses:** São as conexões entre os neurônios de uma rede. Elas transformam a saída de um neurônio na entrada do seguinte. Cada uma dessas conexões possui um peso, que ajusta a influência que a saída do neurônio anterior terá no próximo neurônio. Esse peso é um valor numérico que multiplica a saída recebida, determinando o quanto aquela informação é relevante para o processamento subsequente (HOSAKI; RIBEIRO, 2021).
- **Função de ativação:** Por lidar com problemas complexos, as redes neurais podem gerar valores de saída muito discrepantes ao processar as entradas ao longo de sua topologia. Para mitigar esse problema, utiliza-se a função de ativação, que tem como objetivo restringir a amplitude dos valores gerados pelos neurônios (HOSAKI; RIBEIRO, 2021).
- **Bias:** Valor aplicado externamente a cada neurônio e que tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação (FLECK; BENEDET; SILVA, 2016).

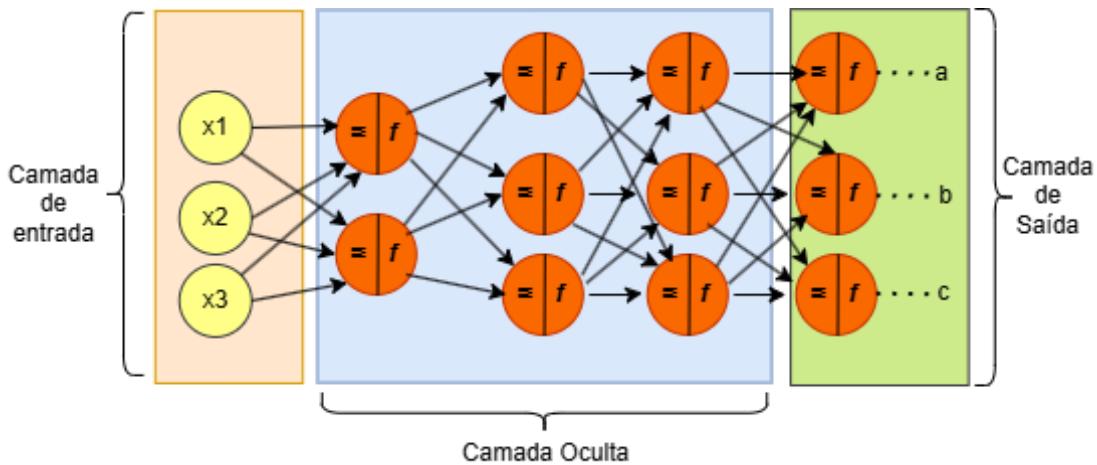
Segundo Fleck et al. (2016) o aprendizado da rede MLP (Perceptron Multicamadas) envolve o processo de ajuste dos pesos das sinapses, conhecido como *backpropagation*. Esse processo ocorre durante o treinamento em duas etapas: propagação e retropropagação. Na primeira etapa, um conjunto de entradas é propagado pela rede, camada por camada, até produzir

Figura 1 – Exemplo de arquitetura de uma rede Perceptron



Fonte: Elaborado pelo autor (2025)

Figura 2 – Exemplo de arquitetura de uma MLP totalmente conectada



Fonte: Elaborado pelo autor (2025)

um conjunto de saídas ao final. Na segunda etapa, todos os pesos sinápticos são ajustados com base em uma regra de correção de erro. O erro é retropropagado pela rede, da camada de saída em direção à camada de entrada, permitindo o ajuste dos pesos sinápticos. Esse ajuste melhora a capacidade da rede de se adequar à tarefa, gerando uma saída mais próxima da resposta desejada.

2.3 Deep Learning

Deep learning trata-se de um conjunto de técnicas de *Machine Learning* que utilizam redes neurais artificiais profundas, com muitas camadas ocultas entre a camada de entrada e a de saída.

De acordo com Hosaki e Ribeiro (2021), modelos clássicos de *machine learning* são

limitados por sua capacidade de extração de características, o que restringe a quantidade de dados que podem processar. Em contraste, o *deep learning*, devido à sua topologia complexa, consegue lidar com grandes volumes de dados sem enfrentar essa limitação. No entanto, essa abordagem exige um alto poder computacional para ser aplicada de forma eficaz.

Segundo Pichler e Hartig (2023), apesar de esses modelos compartilharem princípios e conceitos fundamentais com outros algoritmos de *machine learning*, eles são frequentemente considerados um campo distinto devido às suas características específicas e arquiteturas especializadas. Modelos de *deep learning*, como as *Convolutional Neural Networks* (CNNs), apresentam estruturas que diferem significativamente dos modelos tradicionais de aprendizado de máquina, sendo projetados para atender a desafios específicos de forma mais eficiente.

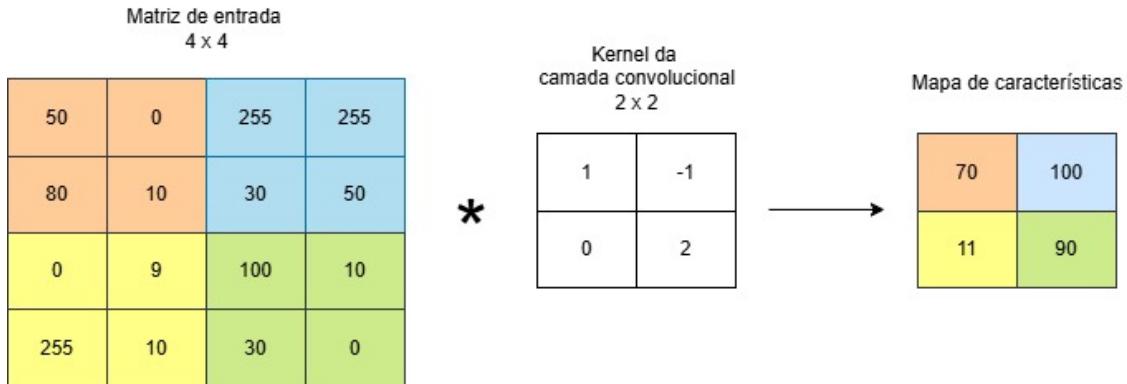
2.4 Convolutional Neural Networks

Segundo Hosaki e Ribeiro (2021), as *Convolutional Neural Networks* (CNNs) são a forma de *deep learning* mais utilizada para tarefas de visão computacional. A principal aplicação das CNNs está no processamento de informações visuais, como imagens e vídeos, pois a operação de convolução permite analisar dados preservando sua estrutura bidimensional.

O trabalho de (ALZUBAIDI et al., 2021) demonstra que a arquitetura da CNN é composta por três tipos de camadas, cada uma com uma função específica. Embora o número de camadas possa variar, elas são geralmente classificadas em três tipos:

- **Camada de Convolução:** A camada convolucional possui um *kernel*, que é uma matriz de pesos. Cada neurônio é responsável por processar esse *kernel* em uma região específica da imagem, denominada campo receptivo local. Nesse processo, os valores do campo receptivo local são multiplicados pelos pesos correspondentes do *kernel* e, em seguida, somados para formar um único valor escalar. A saída da camada é o conjunto das saídas de todos os neurônios, resultando em uma matriz bidimensional denominada mapa de características. O processo de convolução é demonstrado na Figura 3.
- **Camada de Pooling:** A camada de *pooling* é responsável por reduzir a dimensão do mapa de características, preservando o maior número possível de informações extraídas pelas camadas convolucionais. Ela funciona de maneira semelhante à camada de convolução, substituindo uma região do mapa de características por um único valor, geralmente utilizando operações como *max pooling* (que seleciona o maior valor da região) ou *average pooling* (que calcula a média dos valores).
- **Camada Totalmente Conectada:** É utilizada como o classificador da CNN. Ela segue o método básico da rede neural perceptron de múltiplas camadas convencional. A sua entrada vem da camada anterior de *pooling* ou convolucional. Esta entrada está na forma de um vetor, que é criado a partir dos mapas de características após o achatamento. A saída da última camada representa a saída final da CNN.

Figura 3 – Exemplo de convolução



Fonte: Elaborado pelo autor (2025)

2.5 Yolo

Diante das particularidades deste trabalho, optou-se pelo uso do algoritmo YOLO devido à sua eficiência na detecção em tempo real. Além disso, sua oitava versão integra máscaras de segmentação, proporcionando maior precisão nas caixas delimitadoras. A altura e a largura das caixas serão convertidas de pixels para centímetros, permitindo assim mensurar o tamanho das castanhas.

Criado por Joseph Redmon em 2015, o YOLO, já em sua primeira versão, demonstrava a capacidade de detectar objetos com apenas uma passagem da imagem pela rede convolucional, o que originou seu nome: YOLO (*You Only Look Once*). Sua *pipeline* simplificada permite uma detecção rápida sem comprometer a precisão, e sua execução de 45 quadros por segundo torna possível realizar detecções em tempo real com o dobro da média de precisão em relação aos modelos concorrentes. Esse desempenho foi alcançado graças à sua abordagem inovadora, que trata a detecção de objetos como um problema de regressão, convertendo diretamente os pixels da imagem em coordenadas das caixas delimitadoras.

Além disso, o YOLO utiliza o contexto global da imagem para identificar objetos, diferenciando-se de métodos que empregam um contexto regional, como o R-CNN (*Region-based Convolutional Neural Networks*) ou o *Faster R-CNN*, que utilizam o método de janela deslizante. Graças a isso, o YOLO comete menos da metade dos erros de fundo em comparação ao *Fast R-CNN* (REDMON et al., 2015).

Na sua oitava versão, o YOLOv8¹ conta com uma abordagem sem âncoras, o que contribui para uma melhor precisão e um processo de detecção mais eficiente. Sua arquitetura de *backbone* e *neck* é de última geração e oferece uma compensação otimizada entre precisão e velocidade, tornando-o ideal para diversas aplicações (ULTRALYTICS, 2024). De acordo com (ZOU et al., 2023) o *backbone* é responsável pela extração de características, gerando o mapa de características, enquanto o *neck* atua como um funil, reduzindo e refinando esse mapa.

¹ <https://github.com/ultralytics/ultralytics>

3 TRABALHOS RELACIONADOS

Na agricultura, a inteligência artificial tem sido utilizada como facilitadora em diversos setores, conforme demonstrado nos trabalhos de (Sá et al., 2023), (LIN et al., 2024) e (WEN et al., 2022).

O trabalho de (Sá et al., 2023) aborda o contexto da pós-colheita e do beneficiamento de frutos, destacando que a identificação e a rastreabilidade são fundamentais para garantir a qualidade do produto e reduzir perdas. Métodos convencionais de monitoramento dependem fortemente da inspeção humana, o que pode resultar em inconsistências e ineficiências. Nesse cenário, técnicas de *deep learning* têm sido aplicadas para otimizar o controle de produção.

O estudo implementa um sistema de monitoramento automatizado utilizando o YOLOv8, capaz de processar imagens em tempo real, identificar e classificar contentores de frutos, com o objetivo de reduzir erros humanos, aumentar a rastreabilidade e melhorar a eficiência operacional no setor agrícola. A triagem manual é suscetível a erros, variações subjetivas e limitações operacionais.

O modelo foi treinado com um banco de dados contendo imagens de contentores de frutos em diferentes cenários, incluindo variações de iluminação e obstruções parciais. As métricas de avaliação utilizadas foram precisão, *recall* e média de precisão. Além disso, o sistema foi testado em ambientes simulados para verificar sua viabilidade em condições reais de operação.

Os resultados obtidos indicam que o modelo é altamente eficiente para a detecção de contentores de frutos, mesmo em condições desafiadoras. O modelo apresentou altos índices de precisão e demonstrou ser uma solução viável para aplicação na indústria agrícola.

Por sua vez, o trabalho de (LIN et al., 2024) propõe a detecção automática de frutas cítricas em ambientes naturais, que apresentam desafios significativos devido à oclusão causada por folhas, galhos e sobreposição entre os frutos. O estudo desenvolve o AG-YOLO, um algoritmo de detecção rápida de frutas cítricas que incorpora fusão de contexto global para melhorar a precisão em cenários complexos.

O AG-YOLO mescla o contexto global com o local utilizando a arquitetura NextViT como *backbone*, permitindo a extração de características globais das imagens. Além disso, incorpora o Módulo de Fusão de Contexto Global (GCFM), que facilita a interação entre características locais e globais por meio de mecanismos de autoatenção. Essa abordagem demonstrou melhorias significativas na detecção de frutas cítricas, especialmente em condições de oclusão severa.

A detecção automática de pragas agrícolas em ambientes naturais é um desafio significativo, especialmente devido à alta densidade e ao tamanho reduzido dos insetos. Nesse contexto, (WEN et al., 2022) propuseram o Pest-YOLO, um modelo de detecção voltado para múltiplas classes de pragas densas e pequenas, visando melhorar a precisão em cenários complexos.

O Pest-YOLO incorpora a função de perda focal (*focal loss*) para lidar com o desequilíbrio entre classes e melhorar a atenção a amostras difíceis. Essa abordagem permite que o modelo foque em exemplos mais desafiadores durante o treinamento, resultando em melhor desempenho na detecção de pragas pequenas e densamente distribuídas.

Além disso, o modelo foi avaliado em um conjunto de dados abrangente, demonstrando sua eficácia na detecção e contagem de pragas em larga escala. Os resultados indicaram melhorias significativas em relação a modelos anteriores, destacando o potencial do Pest-YOLO para aplicações práticas na agricultura de precisão.

4 METODOLOGIA

Este trabalho adota uma abordagem metodológica aplicada e experimental. A abordagem experimental se dá por meio dos testes do modelo YOLO na detecção de castanhas *in natura*, enquanto a abordagem aplicada consiste na utilização do modelo testado como ferramenta auxiliar na avaliação não invasiva.

O processo metodológico se inicia com a coleta e anotação das imagens das castanhas de caju, conforme descrito na Seção 4.1. Em seguida, foi realizada a escolha e o desenvolvimento do modelo utilizando a arquitetura YOLOv8, que foi treinado, conforme descrito na Seção 4.2, posteriormente testado na Seção 4.3 e avaliado na Seção 5.

4.1 Dataset

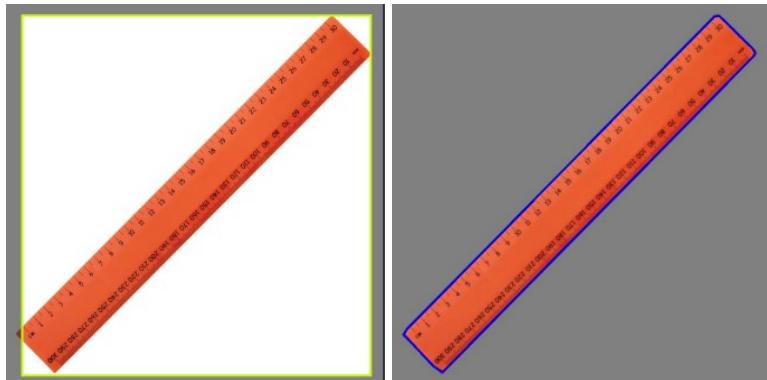
O conjunto de dados utilizado neste trabalho é composto por 681 imagens contendo castanhas e réguas, organizadas em quatro classes: “castanha” (inteiras, sem comprometimento e maduras), “avariada” (com a presença de furos), “maturi” (imaturas) e “réguas”, com um total de 742, 348, 467 e 211 objetos anotados, respectivamente.

Com o intuito de garantir maior diversidade e representatividade das castanhas, as imagens foram capturadas a diferentes distâncias — até 60 cm de altura —, sob variadas condições de iluminação e qualidade de imagem. Além disso, foram utilizados múltiplos ângulos de captura, com diferentes quantidades de castanhas por imagem, a fim de ampliar a complexidade do acervo.

As 164 imagens correspondentes à classe Réguas foram extraídas do projeto Ruler Computer Vision Project, disponível na plataforma Roboflow Universe. As imagens foram reanotadas com a finalidade de se adequar aos padrões do novo conjunto de dados. A anotação original utiliza *bounding box*, onde o objeto se encontra no centro dela. Já na adaptação, a réguas é contornada, conforme ilustrado na Figura 4. As réguas presentes variam quanto à cor, tamanho, formato e número por imagem. Exemplos visuais das quatro classes estão apresentados na Figura 5.

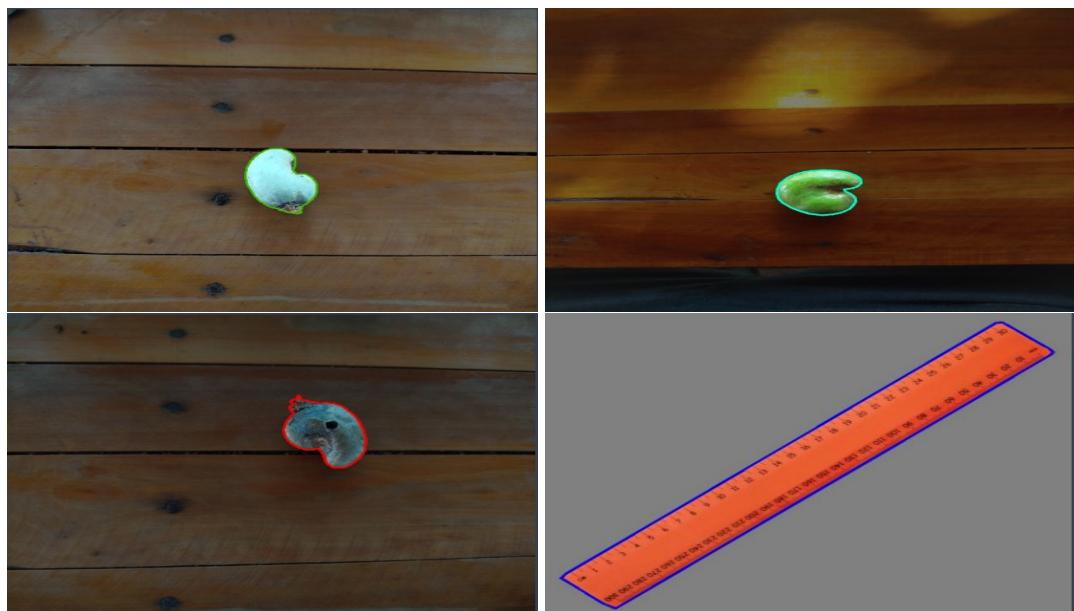
Para aumentar o conjunto de dados, foram aplicadas técnicas de *data augmentation*, como espelhamento horizontal, rotação de 90° no sentido horário, rotação de 90° no sentido anti-horário e rotações aleatórias entre -12° e +12°. Com essas técnicas, o número de imagens no *dataset* foi ampliado de 691 para 1286 e as classes de 348 para 513, 742 para 1513, 467 para 734. As classes passaram de 28 para 70, 186 para 409 e 85 para 182. Do total, 76% das imagens foram destinadas ao treinamento e 24% à validação.

Figura 4 – Exemplo de adaptação da anotação



Fonte: Elaborado pelo autor (2025)

Figura 5 – Exemplos das classes Castanha, Maturi, Avariada e Regua anotadas no Roboflow



Fonte: Elaborado pelo autor (2025)

4.2 Treinamento

O modelo foi treinado ao longo de 60 épocas, utilizando a técnica de *transfer learning*, na qual os melhores pesos obtidos na primeira etapa de treinamento foram reaproveitados na segunda. No YOLO, esses pesos são selecionados com base nas métricas $mAP@0.5$ e $mAP@0.5:0.95$. Essa abordagem foi adotada em virtude das limitações da versão gratuita do Google Colab, que não permite treinamentos contínuos com duração superior a 40 épocas. Assim, o processo foi dividido em duas etapas: a primeira com 25 épocas e a segunda com 35.

O treinamento foi realizado por meio de aprendizado supervisionado, com possibilidade de ativação do mecanismo de *early stopping*, que interrompe automaticamente o processo caso o desempenho no conjunto de validação deixe de apresentar melhorias. Esse recurso é essencial para evitar o *overfitting*.

Na primeira etapa, o modelo foi treinado com a versão inicial do conjunto de dados ao

longo de 25 épocas. O objetivo principal dessa fase foi possibilitar que a rede neural aprendesse características fundamentais, respeitando as limitações de tempo de execução impostas pela plataforma utilizada. Na segunda etapa, os melhores pesos da fase anterior foram carregados e utilizados em um novo treinamento, agora com duração de 35 épocas. Essa fase teve como finalidade a potencial ativação do *early stopping* e o refinamento do aprendizado do modelo.

4.3 Testes

O modelo foi avaliado utilizando aproximadamente 309 imagens do conjunto de validação, além de algumas imagens inéditas que não haviam sido vistas pelo modelo durante o treinamento. A Figura 6 apresenta exemplos de detecções realizadas pelo modelo. A classe “castanha” obteve previsões com mais de 80% de confiança; a classe “maturi” alcançou 93%; e a classe “avariada” obteve valores de 93% e 96%. Esses resultados demonstram a capacidade do modelo em rotular corretamente as imagens, mesmo diante de variações visuais e classes com características distintas.

Figura 6 – Exemplo de imagens rotuladas pelo modelo



Fonte: Elaborado pelo autor (2025)

4.4 Métricas de avaliação

As métricas de avaliação utilizadas neste trabalho são a precisão, o *recall*, o *mean average precision* (mAP) e a matriz de confusão. De acordo com (PADILLA et al., 2021), a precisão é definida como a proporção de instâncias classificadas positivamente que são, de fato, corretas, indicando a capacidade do modelo de selecionar apenas objetos relevantes. Por sua vez, o *recall* corresponde à proporção de instâncias relevantes que foram corretamente identificadas, refletindo a capacidade do modelo de localizar todas as ocorrências pertinentes.

Para o cálculo da precisão e do *recall*, cada caixa delimitadora prevista deve ser classificada em uma das três categorias: Verdadeiros Positivos (VP), Falsos Positivos (FP) ou Falsos Negativos (FN). Os verdadeiros positivos correspondem às detecções corretas de objetos existentes. Já os falsos positivos representam detecções incorretas, seja pela identificação de um objeto inexistente ou pela localização imprecisa de um objeto real. Por fim, os falsos negativos

ocorrem quando uma caixa delimitadora da verdade de base não é detectada pelo modelo. Essas classificações são essenciais para a avaliação precisa do desempenho dos detectores de objetos. As Equações 1 e 2 representam os cálculos de precisão e *recall*, respectivamente:

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (1)$$

$$\text{Recall} = \frac{VP}{VP + FN} \quad (2)$$

O *mean Average Precision* (mAP) corresponde à média da área sob a curva de *precisão* em função do *recall* para todas as classes, representando o quanto preciso é o modelo à medida que realiza um número crescente de detecções. Existem duas versões mais utilizadas dessa métrica: o *mAP@0.5* e o *mAP@0.5:0.95*. A primeira considera uma predição correta quando o valor de *IoU* (Intersection over Union) é igual ou superior a 0,5, enquanto a segunda aplica critérios mais rigorosos, calculando a média do *AP* para valores de *IoU* variando de 0,5 a 0,95, com intervalos de 0,05 (PADILLA et al., 2021).

O *IoU* é uma medida que avalia o grau de sobreposição entre a caixa delimitadora predita e a caixa real, sendo calculado pela razão entre a área da interseção e a área da união dessas duas caixas. Seu valor varia de 0 a 1, sendo 1 representativo de uma correspondência perfeita entre a predição e a anotação real (PADILLA et al., 2021).

Por fim, a matriz de confusão, conforme descrito por (AMARAL, 2023), serve para verificar a qualidade do modelo na classificação. Sua estrutura é composta por linhas e colunas: neste trabalho, as linhas representam as classes preditas pelo modelo, enquanto as colunas correspondem às classes reais. Trata-se de uma ferramenta interpretativa, na qual qualquer previsão fora da diagonal principal é considerada incorreta.

5 AVALIAÇÃO DOS RESULTADOS

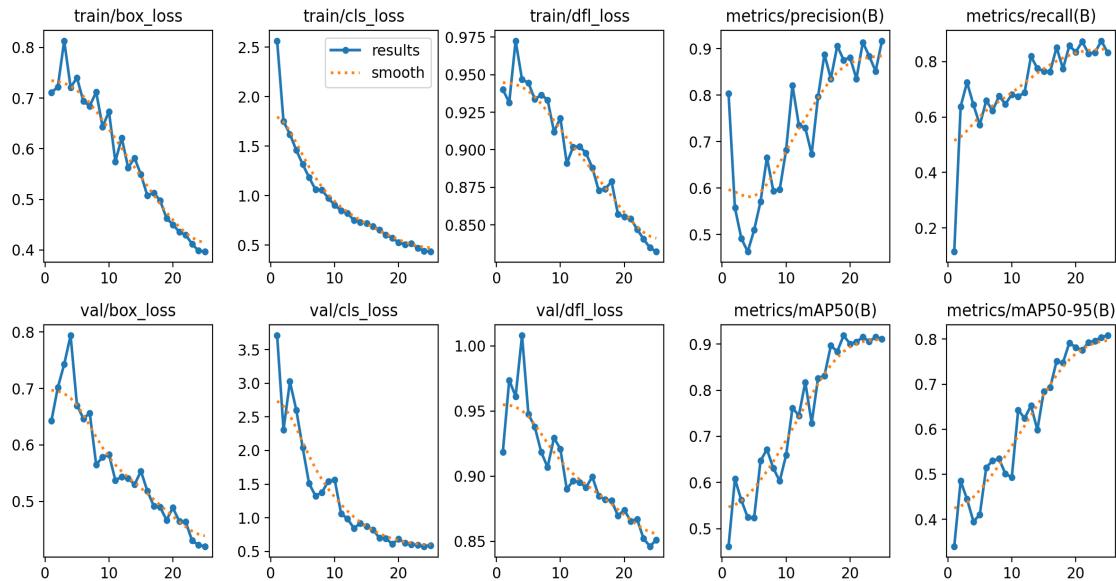
As Figuras 7 e 8 ilustram o desempenho do modelo ao longo do processo de treinamento e validação, realizados em 25 e 35 épocas, respectivamente. Os gráficos estão organizados em duas linhas: a linha superior refere-se aos dados de treinamento, enquanto a inferior representa os dados de validação.

Observa-se que as curvas de perda (*loss*) apresentam uma tendência de queda à medida que o treinamento avança, enquanto as métricas de desempenho demonstram picos cada vez mais altos. Esses resultados indicam que o modelo possui boa capacidade de aprendizado, com redução consistente da função de perda e melhoria progressiva das métricas, tanto nos dados de treinamento quanto nos de validação. Isso sugere que o modelo está conseguindo generalizar adequadamente, sem sinais aparentes de *overfitting*.

Além disso, os gráficos identificados como *metrics/mAP50(B)* e *metrics/mAP50-95(B)* referem-se, respectivamente, às métricas *mAP@0.5* e *mAP@0.5:0.95*. Nota-se que ambas apresentam evolução acentuada nas primeiras iterações, atingindo valores superiores a 0,90 no gráfico

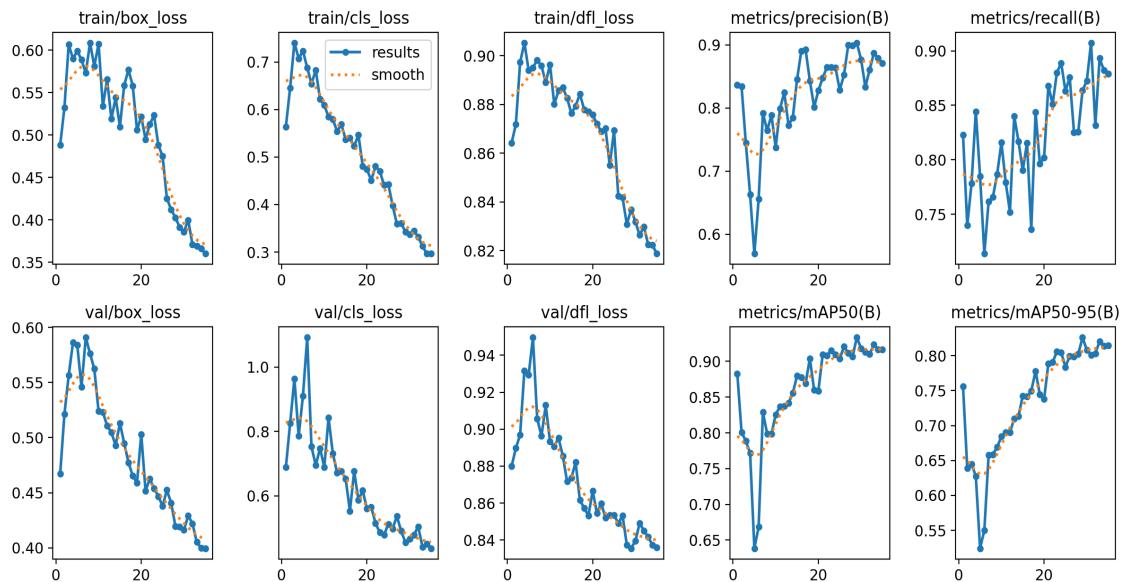
metrics/mAP50(B) e acima de 0,80 nas últimas épocas no gráfico metrics/mAP50-95(B). Esses resultados indicam que o modelo tornou-se altamente eficaz na detecção de objetos, mesmo sob critérios mais rigorosos de sobreposição (IoU), especialmente durante a segunda etapa do treinamento, na qual os picos das curvas atingem seus valores mais elevados.

Figura 7 – Gráficos de Insights do Treinamento do Modelo na primeira etapa



Fonte: Elaborado pelo autor (2025)

Figura 8 – Gráficos de Insights do Treinamento do Modelo na segunda etapa



Fonte: Elaborado pelo autor (2025)

As Figuras 9 e 10 apresentam a matriz de confusão do modelo de classificação para as classes “avariada”, “castanha”, “maturi”, “réguia” e “background”. Na classe “avariada”, ambas as matrizes mostram um número semelhante de falsos positivos (FP), principalmente confundidos

com a classe “castanha”. Apesar disso, a matriz da segunda etapa apresenta um maior número de verdadeiros positivos (VP). Para a classe “castanha”, também há uma boa quantidade de VP em ambas as matrizes, com um aumento na segunda etapa.

A classe “maturi” apresentou o melhor desempenho, com 242 verdadeiros positivos (VP). Em contraste, a classe “réguia” obteve apenas 32 VP e registrou 16 falsos positivos (FP), que foram majoritariamente confundidos com a classe “background”. Esta última representa os casos em que o modelo não detecta ou não classifica nenhum objeto relevante.

Nas duas matrizes de confusão analisadas, observa-se que o modelo falha na detecção de algumas classes, resultando em FN. Essas falhas provavelmente estão relacionadas ao desbalanceamento do conjunto de dados, no qual a classe “castanha” possui um número significativamente maior de instâncias em comparação às demais. Além disso, a semelhança visual entre as classes “castanha” e “avariada” pode dificultar a predição da classe com menor representatividade, contribuindo para os erros de classificação.

De modo geral, o modelo apresentou maior dificuldade na distinção entre as classes “avariada”, “castanha” e “background”, enquanto “maturi” demonstrou alta precisão, e os erros relacionados à “réguia” concentraram-se na confusão com o “background”.

Figura 9 – Matriz de Confusão do Modelo na primeira etapa

		Confusion Matrix				
		avariada	castanha	maturi	réguia	background
Predicted	avariada	37	6	0	0	7
	castanha	11	127	0	0	13
	maturi	2	0	242	0	15
	réguia	0	0	0	33	13
	background	10	2	4	1	1
		avariada	castanha	maturi	réguia	background
		True				

Fonte: Elaborado pelo autor (2025)

As Figuras 11 e 12 apresentam a Curva de Precisão-Confiança (*Precision-Confidence Curve*), correspondentes à primeira e à segunda etapa de treinamento, respectivamente. Essa curva é utilizada para avaliar o desempenho de um modelo de detecção de objetos com base no nível de confiança atribuído às previsões. O gráfico permite visualizar a relação entre a confiança nas predições realizadas pelo modelo e sua precisão, fornecendo subsídios importantes para a definição de um limiar de confiança ideal, com o objetivo de otimizar o desempenho.

No gráfico, o eixo horizontal representa o valor de confiança (*Confidence*) associado às predições do modelo, variando de 0 a 1. O eixo vertical indica a precisão (*Precision*), que corresponde à proporção de predições corretas entre todas as predições positivas realizadas.

Figura 10 – Matriz de Confusão do Modelo na segunda etapa

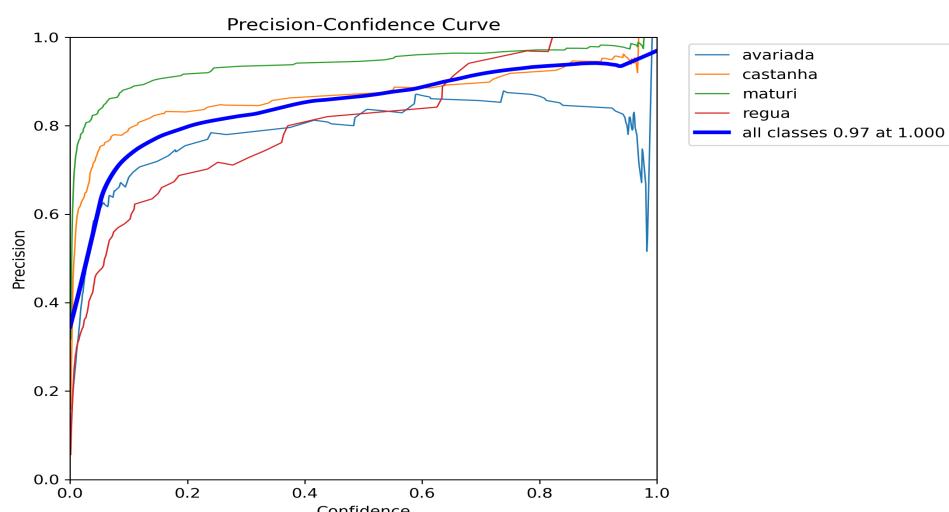
		Confusion Matrix				
		avariada	castanha	maturi	regua	background
Predicted	avariada	39	5	0	0	10
	castanha	14	130	0	0	9
maturi	1	0	242	0	22	
regua	0	0	0	32	16	
background	6	4	2	0	0	
True	avariada	39	5	0	0	10

Fonte: Elaborado pelo autor (2025)

As curvas individuais referem-se às seguintes classes: “avariada” apresenta uma precisão mais instável, especialmente em altos níveis de confiança, indicando maior variação nos resultados; “castanha” e “regua” demonstram um comportamento crescente e mais consistente, com maior precisão a partir de níveis intermediários de confiança; “maturi” se destaca pelo desempenho superior, mantendo alta precisão mesmo em baixos níveis de confiança, com estabilidade ao longo de toda a curva.

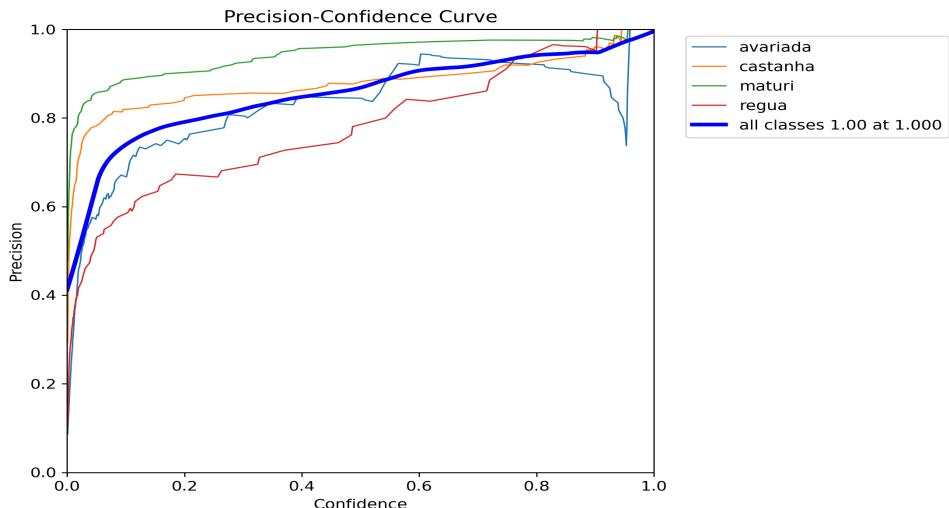
Na prática, esse tipo de gráfico é essencial para determinar um limiar de confiança ideal para a filtragem das detecções. Quanto maior for o valor de confiança exigido para aceitar uma predição, maior será a precisão; no entanto, isso pode acarretar uma redução no *recall*. Assim, identificar o ponto em que o modelo alcança alta precisão sem eliminar muitas detecções válidas é fundamental para obter um bom equilíbrio entre desempenho e abrangência.

Figura 11 – Curva de Precisão-Confiança do Modelo na primeira etapa



Fonte: Elaborado pelo autor (2025)

Figura 12 – Curva de Precisão-Confiança do Modelo na segunda etapa



Fonte: Elaborado pelo autor (2025)

As Figuras 13 e 14 exibem a Curva *Recall*-Confiança, que relaciona o nível de confiança atribuído às previsões do modelo com o valor de revocação (*recall*) obtido.

O eixo das abscissas representa os diferentes níveis de confiança definidos no processo de inferência, enquanto o eixo das ordenadas mostra os valores correspondentes de *recall*. O gráfico permite observar o comportamento do modelo por classe, bem como sua média global.

Ao analisar as curvas, observa-se que as classes “maturi”, “castanha” e “régua” alcançam valores elevados de *recall* em praticamente todos os níveis de confiança, evidenciando que o modelo possui boa capacidade de detecção para essas categorias.

Por outro lado, a classe “avariada” apresenta desempenho consideravelmente inferior, com o *recall* diminuindo progressivamente à medida que se eleva o nível de confiança exigido. Esse comportamento sugere um maior número de falsos negativos nessa classe, indicando que o modelo tem mais dificuldade em identificar corretamente os objetos desse tipo quando se aplica um filtro mais rigoroso.

A curva média mostra que o modelo atinge sua maior taxa de recuperação de objetos positivos em níveis mais baixos de confiança.

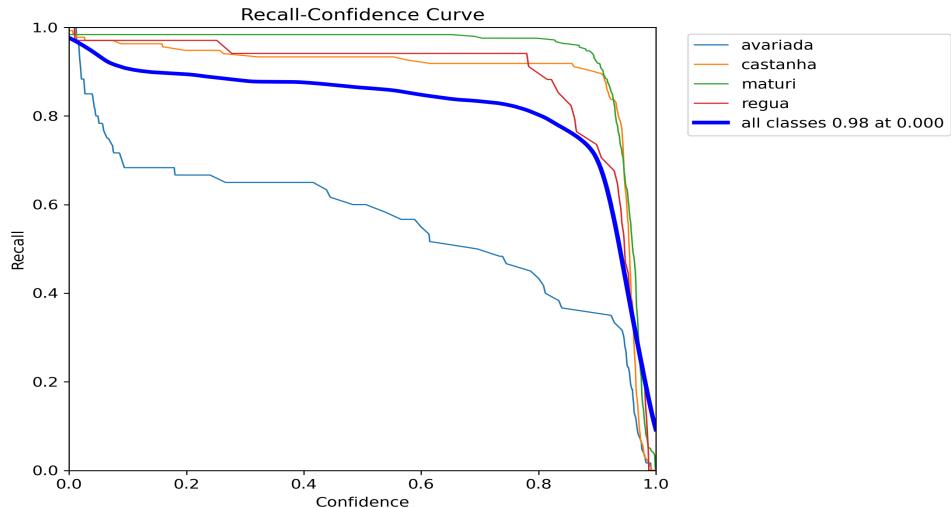
A Figura 15 apresenta o resultado da classificação das castanhas e da detecção da régua pelo modelo. O cálculo do tamanho das castanhas, em centímetros, é realizado em duas etapas.

Na primeira etapa, com base no tamanho real da régua (TR), previamente fornecido ao sistema, é feita a conversão das dimensões da imagem de pixels para centímetros. Para isso, utiliza-se o maior lado da *bounding box* da régua (LM), aplicando-se a seguinte razão:

$$\text{Fator de conversão} = \frac{LM}{TR} \quad (3)$$

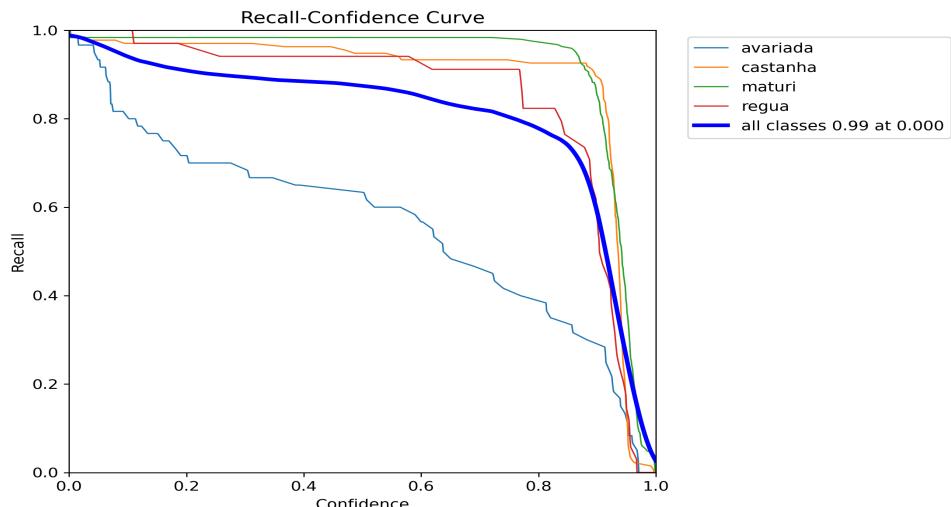
Na segunda etapa, o maior lado da *bounding box* de cada castanha é dividido pelo fator de conversão obtido na Equação 3. Dessa forma, o sistema estima o tamanho das castanhas em centímetros e gera uma tabela em formato PDF, contendo a quantidade de castanhas detectadas,

Figura 13 – Curva de *Recall-Confiaça* do Modelo na primeira etapa



Fonte: Elaborado pelo autor (2025)

Figura 14 – Curva de *Recall-Confiaça* do Modelo na segunda etapa

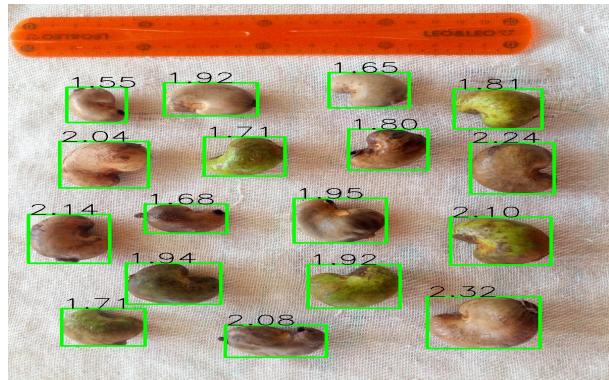


Fonte: Elaborado pelo autor (2025)

organizadas por classe e faixa de tamanho, conforme ilustrado na Figura 16. Os tamanhos seguem os critérios apresentados no trabalho de (LIMA A. C.; VIDAL NETO, 2022), que define as categorias da seguinte forma:

- **Grande** – Retida na peneira de 23 mm (90 frutos por quilo);
- **Média** – Passa na peneira de 23 mm e é retida na de 21 mm (91 a 140 frutos por quilo);
- **Pequena** – Passa na peneira de 21 mm e é retida na de 19 mm (141 a 220 frutos por quilo);
- **Miúda** – Passa na peneira de 19 mm e é retida na de 15 mm (221 a 300 frutos por quilo);
- **Cajuí** – Passa na peneira de 15 mm (acima de 300 frutos por quilo).

Figura 15 – Demonstração da inferência de tamanho das castanhas



Fonte: Elaborado pelo autor (2025)

Figura 16 – Exemplo da tabela

Classe	Cajuí	Miúda	Pequena	Média	Grande
Avariada	0	2	3	0	0
Castanha	0	1	3	2	1
Marturi	0	3	1	1	0

Fonte: Elaborado pelo autor (2025)

6 CONCLUSÃO

Em síntese, este trabalho apresentou uma abordagem abrangente para a construção de um *dataset* específico de castanhas de caju e o treinamento de um modelo de detecção de castanha de caju, explorando tanto o desenvolvimento *from scratch* quanto o uso de *transfer learning*. O processo de treinamento foi dividido em duas etapas, totalizando 60 épocas (25 e 35, respectivamente).

O modelo desenvolvido apresenta alta precisão e boa capacidade de generalização. Apesar de demonstrar alguma confusão entre as classes “avariadas” e “castanha”, em que 12,82% das predições da classe “avariada” feitas pelo modelo na segunda etapa são da classe “castanha”, essa confusão ocorre possivelmente pela semelhança entre algumas instâncias dessas classes. Ainda assim, o modelo é capaz de distinguir adequadamente as demais classes. Além disso, a detecção da classe “régua” permite a inferência do tamanho das castanhas.

Entretanto, a partir do limiar de confiança de 70%, em que o modelo alcança precisões superiores a 80%, observa-se um *recall* que varia de 0% a 90%, o que afeta diretamente a quantidade de detecções reais que deixam de ser realizadas. Isso o torna inviável para a tarefa de contagem, uma vez que suas detecções não correspondem fielmente à quantidade real de objetos presentes.

Dessa forma, este estudo demonstra sua viabilidade prática na identificação e classificação de castanhas de caju *in natura*, oferecendo benefícios significativos para a cadeia produtiva.

Como perspectivas para trabalhos futuros, destaca-se o aprimoramento do *dataset* inicial,

com foco no aumento da quantidade de imagens contendo múltiplas instâncias de objetos, visando elevar o *recall* em níveis mais altos de confiança, mantendo uma boa precisão mesmo em limiares mais baixos. Busca-se, assim, atingir valores superiores a 98% de *recall* sem comprometer a precisão. Outras melhorias relevantes incluem o balanceamento da quantidade de instâncias por classe, a fim de mitigar problemas de confusão, além da ampliação das categorias de classificação.

REFERÊNCIAS

- ALZUBAIDI, L. et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. **Journal of Big Data**, SpringerOpen, v. 8, n. 1, p. 1–74, 2021. Disponível em: <<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8>>.
- AMARAL, A. A. d. **Monitoramento de animais domésticos com redes neurais convolucionais e transfer learning: uma abordagem utilizando YOLO para cuidar dos meus pets**. 2023. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Campus Aracati. Disponível em: <<https://gestaoaracati.ifce.edu.br/attachments/download/2726/tcc.pdf>>.
- BANDEIRA, P. M. d. C. et al. A inteligência artificial e o setor agrícola nos últimos 20 anos: um estudo. **Natural Resources**, v. 12, n. 2, p. 28–38, 2023. Acesso em: 25 maio 2025. Disponível em: <<https://sustenere.inf.br/index.php/naturalresources/article/view/8036>>.
- BORBA, M. d. C. et al. Gestão no meio agrícola com o apoio da inteligência artificial: uma análise da digitalização da agricultura. **Revista em Agronegócio e Meio Ambiente**, v. 15, n. 3, p. 1–22, 2022. Acesso em: 25 maio 2025. Disponível em: <<https://periodicos.unicesumar.edu.br/index.php/rama/article/view/9337>>.
- FLECK, L. C.; BENEDET, J.; SILVA, J. S. da. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016. Disponível em: <<https://periodicos.utfpr.edu.br/recit/article/view/4330>>.
- GOMES, V. P. **Revisitando o teste de Turing: análises e consequências**. Tese (Doutorado em Filosofia) — Centro de Ciências Humanas, Letras e Artes, Universidade Federal do Rio Grande do Norte, Natal, 2023. Orientador: Dr. Samir Bezerra Gorsky. 109f.
- GOUVEIA, P. R. L. **Redes Neurais Artificiais: Conceitos e Aplicações**. 2012. <<https://cepein.femanet.com.br/BDigital/arqTccs/0911270619.pdf>>. Acessado em: 16 jan. 2025.
- HOSAKI, G. Y.; RIBEIRO, D. F. Deep learning: Ensinando a aprender. **RGE - Revista de Gestão e Estratégia**, v. 1, n. 3, p. 36–50, 2021. Disponível em: <<http://ric-cps.eastus2.cloudapp.azure.com/bitstream/123456789/5060/1/DEEP-LEARNING.pdf>>.
- Instituto Brasileiro de Geografia e Estatística (IBGE). **Estatística da Produção Agrícola: janeiro de 2024**. 2024. <https://biblioteca.ibge.gov.br/visualizacao/periodicos/2415/epag_2024_jan.pdf>. Acesso em: 2 jun. 2025.
- Instituto Brasileiro de Geografia e Estatística (IBGE). **Estatística da Produção Agrícola: abril de 2025**. 2025. <https://biblioteca.ibge.gov.br/visualizacao/periodicos/2415/epag_2025_abr.pdf>. Acesso em: 2 jun. 2025.

- LIMA A. C.; VIDAL NETO, F. d. C. M. C. W. C. P. P. P. F. A. P. P. F. F. d. A. **Recomendações para avaliação rápida da qualidade de castanhas-de-caju destinadas ao beneficiamento industrial.** Fortaleza: Embrapa Agroindústria Tropical., 2022. Distribuído no Evento X. Disponível em: <<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/233193/1/CT-278.pdf>>.
- LIN, Y. et al. Ag-yolo: A rapid citrus fruit detection algorithm with global context fusion. **Agriculture**, v. 14, n. 1, 2024. ISSN 2077-0472. Disponível em: <<https://www.mdpi.com/2077-0472/14/1/114>>.
- OPENAI et al. **Dota 2 with Large Scale Deep Reinforcement Learning.** 2019. Disponível em: <<https://arxiv.org/abs/1912.06680>>.
- PADILLA, R. et al. A comparative analysis of object detection metrics with a companion open-source toolkit. **Electronics**, v. 10, n. 3, 2021. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/10/3/279>>.
- REDMON, J. et al. You only look once: Unified, real-time object detection. **CoRR**, abs/1506.02640, 2015. Disponível em: <<http://arxiv.org/abs/1506.02640>>.
- RUSSELL, S.; NORVIG, P. **Inteligência Artificial: Uma Abordagem Moderna.** 4. ed. São Paulo, Brasil: Pearson, 2021. ISBN 97885582605366.
- Sá, P. et al. Yolov8 para controle de produção pós-colheita e beneficiamento de frutos. **Revista de Engenharia e Pesquisa Aplicada**, v. 9, p. 115–122, 12 2023. Disponível em: <https://www.researchgate.net/publication/377253183_YOLOv8_para_Controle_de_Producao_Pos-colheita_e_Beneficiamento_de_Frutos>.
- ULTRALYTICS. **YOLOv8 – Tarefas e modos suportados.** 2024. <<https://docs.ultralytics.com/pt/models/yolov8/#supported-tasks-and-modes>>. Acessado em 5 de junho de 2025.
- WEN, C. et al. Pest-yolo: A model for large-scale multi-class dense and tiny pest detection and counting. **Frontiers in Plant Science**, Volume 13 - 2022, 2022. ISSN 1664-462X. Disponível em: <<https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.973985>>.
- ZOU, Z. et al. **Object Detection in 20 Years: A Survey.** 2023. Disponível em: <<https://arxiv.org/abs/1905.05055>>.