

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

MATHEUS SANDIM GONÇALVES

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO COMPUTACIONAL
PARA DETECÇÃO DO PONTO DE COLHEITA DE BANANAS**

FLORIANÓPOLIS, 2025.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA – CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

MATHEUS SANDIM GONÇALVES

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO COMPUTACIONAL
PARA DETECÇÃO DO PONTO DE COLHEITA DE BANANAS**

Trabalho de conclusão de curso submetido
ao Instituto Federal de Educação, Ciência e
Tecnologia de Santa Catarina como parte
dos requisitos para obtenção do título de
Engenheiro Eletrônico.

Orientador:
Prof. Robinson Pizzio, Dr.

FLORIANÓPOLIS, 2025.

Ficha de identificação da obra elaborada pelo autor.

Gonçalves, Matheus Sandim

Desenvolvimento de um sistema de visão computacional
para detecção do ponto de colheita de bananas / Matheus Sandim
Gonçalves; orientação de Robinson Pizzio. -
Florianópolis, SC, 2025.

36 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Eletrônica. Departamento Acadêmico
de Eletrônica.

Inclui Referências.

1. Visão computacional . 2. Inteligência Artificial
. 3. YOLOv8. 4. Maturação de bananas. 5. CV2. I. Pizzio,
Robinson. II. Instituto Federal de Santa Catarina.
III. Desenvolvimento de um sistema de visão computacional
para detecção do ponto de colheita de bananas.

DESENVOLVIMENTO DE UM SISTEMA DE VISÃO COMPUTACIONAL PARA DETECÇÃO DO PONTO DE COLHEITA DE BANANAS

MATHEUS SANDIM GONÇALVES

Este trabalho foi julgado adequado para obtenção do título de Engenheiro Eletrônico e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 25 de março, 2025.

Banca Examinadora:

Documento assinado digitalmente
 **ROBINSON PIZZIO**
Data: 31/03/2025 19:06:34-0300
Verifique em <https://validar.iti.gov.br>

Robinson Pizzo, Dr.

Documento assinado digitalmente
 **CARLOS GONTARSKI SPERANZA**
Data: 01/04/2025 08:49:17-0300
Verifique em <https://validar.iti.gov.br>

Carlos Gontarski Speranza, Dr.

Documento assinado digitalmente
 **DANIEL LOHMANN**
Data: 02/04/2025 09:04:43-0300
Verifique em <https://validar.iti.gov.br>

Daniel Lohmann, M.Eng.

Este trabalho é dedicado à minha família,
enfim eu terminei ele, não?

RESUMO

Este trabalho desenvolve um sistema de visão computacional para identificar o ponto ideal de colheita de bananas, visando garantir a qualidade e o bom aspecto do fruto para o consumidor. As bananas precisam ser colhidas antes de atingirem a coloração amarela, permitindo que cheguem ao destino sem perda de qualidade visual e nutricional. Para isso, o sistema proposto utiliza o modelo YOLOv8 para detectar cachos e bananas ainda na bananeira, além de analisar a cor do fruto como uma aproximação para identificar o estágio de maturação ideal para a colheita. O método escolhido envolveu o treinamento do modelo YOLOv8 em um conjunto de dados que permite o reconhecimento das bananas e de seus respectivos cachos, correlacionando a coloração do fruto com o ponto ideal de maturação. Os resultados indicaram que o sistema apresenta boa precisão na detecção dos cachos e das bananas, bem como na análise da coloração do fruto, oferecendo uma estimativa confiável para o ponto de colheita. Embora não seja totalmente exato, o método se mostra útil como uma ferramenta de apoio para produtores, promovendo eficiência na cadeia de suprimento de bananas. Assim, o sistema de visão computacional apresentado se destaca como uma solução promissora para o manejo agrícola, contribuindo para a tomada de decisão no momento da colheita e impactando positivamente na qualidade dos produtos oferecidos ao mercado.

Palavras-chave: Visão computacional. Inteligência Artificial. YOLOv8. Maturação de bananas. CV2.

ABSTRACT

This work develops a computer vision system to identify the ideal harvest time for bananas, aiming to ensure the quality and appearance of the fruit for consumers. Bananas must be harvested before they reach a yellow coloration to arrive at their destination without losing visual or nutritional quality. For this purpose, the proposed system utilizes the YOLOv8 model to detect banana bunches and individual bananas still on the plant, as well as to analyze the fruit's color as an approximation for identifying the optimal maturation stage for harvest. The chosen method involved training the YOLOv8 model on a dataset capable of recognizing bananas and their respective bunches, correlating the fruit's coloration with the ideal maturation point. The results indicated that the system shows good accuracy in detecting bunches and bananas, as well as in analyzing the fruit's color, providing a reliable estimate for the harvest point. Although it is not completely precise, the method proves useful as a decision-support tool for producers, promoting efficiency in the banana supply chain. Thus, the computer vision system presented stands out as a promising solution for agricultural management, contributing to informed harvesting decisions and positively impacting the quality of products offered to the market.

Keywords: Computer vision. Artificial Intelligence. YOLOv8. Banana ripeness. CV2.

LISTA DE ABREVIATURAS E SIGLAS

IFSC	Instituto Federal de Santa Catarina
YOLOv8	You Only Look Once version 8
IA	Inteligência Artificial
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
HSV	Hue Saturation Value
IoU	Intersection over Union
mAP	mean Average Precision

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 Justificativa.....	10
1.2 Definição do Problema.....	11
1.3 Objetivo Geral.....	11
1.4 Objetivos Específicos.....	11
2 FUNDAMENTAÇÃO TEÓRICA.....	11
2.1 Rede Neural Convolucional.....	12
2.2 Rede Neural Convolucional baseada em Regiões (R-CNN).....	13
2.3 YOLO e YOLOv8.....	14
2.4 HSV (Hue, Saturation, Value).....	17
3 DESENVOLVIMENTO.....	18
3.1 Treinamento.....	18
3.2 Maturação da banana.....	26
3.3 Aplicativo para celular.....	28
4 CONSIDERAÇÕES FINAIS.....	31
4.1 Sugestões para trabalhos futuros.....	31

1 INTRODUÇÃO

Segundo levantamento realizado pelas Nações Unidas (NATIONS, 2025), a população mundial está estimada em 8,16 bilhões de pessoas. Para o ano de 2061, a previsão é de que esse número ultrapasse os 10 bilhões. Os desafios de alimentar uma população tão grande são diversos, incluindo questões relacionadas à terra, água e distribuição.

Cerca de 2,33 bilhões de pessoas no mundo enfrentam insegurança alimentar moderada ou grave em 2023” (GOV, 2024). Segundo a coordenadora do programa de agricultura familiar da Embrapa, Daniela Bittencourt, “estima-se que cerca de 70% da comida que chega às mesas das nossas casas é proveniente da agricultura familiar. Essa modalidade de agricultura tem relação direta com a segurança alimentar e nutricional da população brasileira.(BITTENCOURT, 2018)

O uso de novas ferramentas e tecnologias é uma forma fundamental de impulsionar a eficiência e sustentabilidade da agricultura familiar no Brasil. A modernização permite um melhor gerenciamento dos insumos, como sementes, fertilizantes e defensivos agrícolas, otimizando sua aplicação e reduzindo desperdícios. Sistemas de irrigação inteligentes, como a irrigação por gotejamento automatizada, garantem o uso eficiente da água. “O uso de drones, que podem ser usados para mapear áreas de plantio, monitorar o crescimento das plantas e identificar problemas como erosão e desmatamento.” (SIAGRI, 2023).

O Brasil é o quarto maior produtor de bananas no mundo (EMBRAPA, 2021). Segundo o IBGE a estimativa de produção em 2021 foi de 7 milhões de toneladas, 99% dessa produção são destinadas para o consumo interno (CNA, 2021). Esse dado reflete a importância da banana na dieta dos brasileiros e no papel da agricultura familiar, responsável por grande parte dessa produção.

Com uma demanda tão significativa, o uso de tecnologias que auxiliem no manejo e colheita das bananas torna-se essencial. Sistemas capazes de identificar o ponto ideal de colheita, por exemplo, garantem maior qualidade do produto, reduzindo perdas e aumentando a rentabilidade para os agricultores.

O momento indicado para a colheita de bananas varia de acordo com diversos fatores, como:

[...] número de dias que transcorrerá no seu transporte da zona produtora para o mercado consumidor, da estação do ano, normas do mercado comprador, tipo de embalagem e utilização dos frutos para consumo local, exportação ou industrialização. Quanto maior for o tempo de transporte desde o bananal até o mercado consumidor e quanto mais quente for a época do ano, mais verdes e com menor diâmetro devem ser colhidas as bananas. (LIMA et al., 2021).

A avaliação do ponto de colheita da banana é, em grande parte, realizada de forma visual. Os critérios mais comuns incluem o diâmetro do fruto, a angulosidade das quinas e a coloração da casca, que passa gradualmente de um verde escuro para tons mais claros (PAIVA et al., 2011).

Dada a contextualização acima, o presente trabalho tem como objetivo o desenvolvimento de um aplicativo para a detecção do ponto de colheita de bananas, utilizando inteligência artificial e visão computacional para realizar a classificação, contagem e detecção da cor das bananas em seus cachos, além de auxiliar o usuário se as bananas já estão no ponto de colheita.

1.1 Justificativa

Diversos fatores influenciam na determinação do momento certo para a colheita da banana. Este trabalho tem como objetivo facilitar a vida de agricultores familiares ou até mesmo de pessoas que possuem bananeiras nos quintais, ajudando a identificar o ponto de maturação da fruta, o que pode auxiliar na decisão de quando realizar a colheita.

O uso de visão computacional foi escolhido devido à sua capacidade de analisar de forma precisa e automatizada características visuais da banana, como cor e quantidade. Essa tecnologia possibilita uma expansão futura nas funcionalidades do aplicativo desenvolvido, tornando cada vez mais robusta a avaliação da maturação, saúde e desenvolvimento da fruta.

1.2 Definição do Problema

A variação dos tons de verde da banana é bastante sutil, o que torna a determinação do ponto de colheita desafiadora para os produtores. Estes utilizam, frequentemente, o diâmetro da fruta como referência. Com o uso de algoritmos de visão computacional, auxiliada por IA (inteligência artificial), é possível introduzir um novo parâmetro, o que pode auxiliar o produtor a identificar, de forma mais precisa, o melhor ponto de colheita. Portanto, o desafio tecnológico enfrentado neste TCC é a determinação, através de técnicas de visão computacional e IA, do melhor ponto de colheita para bananas.

1.3 Objetivo Geral

Este trabalho tem como objetivo treinar uma inteligência artificial para detectar bananas e determinar seu ponto de colheita, considerando o espectro de cores como parâmetro de análise, além de criar um aplicativo de celular para facilitar o acesso a esse detector de maturidade.

1.4 Objetivos Específicos

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos :

- a) Adquirir imagens para construção de um banco de imagens de bananas em seus cachos e criação das caixas delimitadoras das bananas e cachos;
- b) treinar a IA utilizando o YOLOv8;
- c) testar o funcionamento da IA: avaliação da precisão de detecção e contagem;
- d) implementar algoritmo para determinar a cor da banana com base nas imagens detectadas pela IA;
- e) criar e testar o aplicativo para celular.

2 FUNDAMENTAÇÃO TEÓRICA

Visão computacional é uma área da ciência da computação (AZURE, 2010) que se concentra em permitir que computadores e sistemas compreendam, interpretem e processem informações visuais do mundo real, da mesma forma que os humanos fazem com os olhos e o cérebro. Ela permite que as máquinas identifiquem objetos, reconheçam padrões, analisem movimentos e realizem tarefas automatizadas com base em imagens ou vídeos.

Além da forma clássica de visão computacional, hoje utiliza-se, também, ferramentas e técnicas de inteligência artificial para contribuir com o melhor aproveitamento e resultados em sistemas de visão computacional. Uma ferramenta de IA que é empregada para sistemas de visão computacional é a YOLO. Neste trabalho fizemos uso da YOLOv8 e, para que possamos compreender seu funcionamento, é essencial abordar a base teórica das Redes Neurais Convolucionais (CNN) e sua evolução para Rede Neural Convolucional Baseada em Regiões (R-CNN), as quais apresentamos nas seções abaixo.

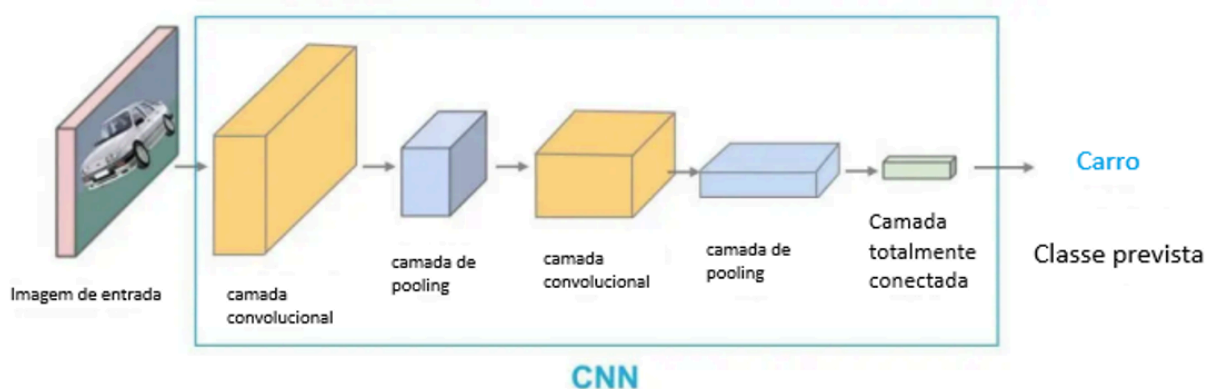
2.1 Rede Neural Convolucional

Rede Neural Convolucional é um tipo de arquitetura de rede neural amplamente utilizada em tarefas de visão computacional. Combinar as camadas de uma CNN ensina a rede neural a identificar o objeto no qual ela está sendo treinada. Estas redes são empregadas na classificação e detecção de objetos em imagens (ODEMAKINDE, 2022).

No funcionamento de uma Rede Neural Convolucional (CNN), ilustrado na Figura 1, as camadas desempenham papéis distintos e complementares. A camada convolucional (representada em amarelo na Figura 1) é responsável por extrair características das imagens, como bordas, texturas e padrões, por meio de operações de convolução que aplicam filtros em diferentes regiões da imagem. A camada de pooling (representada em azul na Figura 1) atua especificamente na detecção de objetos, mapeando as Regiões de Interesse (ROIs) do mapa de características gerado pela camada convolucional para tamanhos fixos. Isso padroniza as dimensões das ROIs e permite que os dados sejam processados

uniformemente. Por fim, a camada totalmente conectada (representada em verde na Figura 1) traduz as características extraídas e padronizadas em classificações ou previsões, conectando todos os neurônios e produzindo o resultado final, como a identificação de objetos em uma imagem. Essas camadas, juntas, transformam imagens brutas em informações úteis de forma hierárquica e estruturada (ODEMAKINDE, 2022).

Figura 1 – Rede Neural Convolucional (CNN)



Fonte: Adaptado de Odemakinde (2022).

2.2 Rede Neural Convolucional baseada em Regiões (R-CNN)

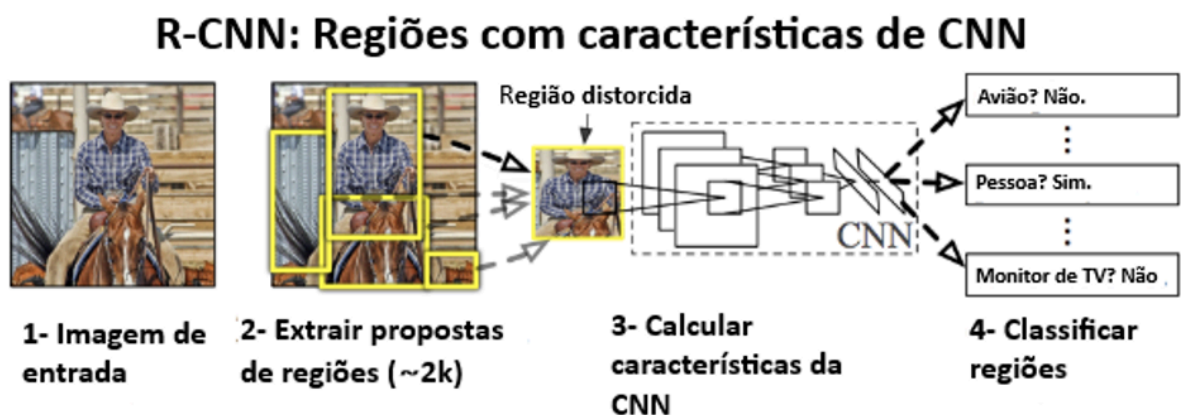
A Figura 2 extraída de (POTRIMBA, 2023), ilustra o funcionamento do algoritmo R-CNN (*Regions with CNN Features*), que realiza a detecção e classificação de objetos em imagens. O processo é dividido em quatro etapas principais:

1. **Imagem de Entrada:** Uma imagem é fornecida como entrada para o modelo, contendo um ou mais objetos de interesse que devem ser identificados.
2. **Propostas de Regiões (~2k):** Nesta etapa, o algoritmo gera cerca de 2.000 propostas de regiões que podem conter objetos. Essas regiões são delimitadas por caixas (*bounding boxes*), destacadas em amarelo na imagem.
3. **Calcular Características com CNN:** As regiões propostas são "deformadas" para se ajustarem a um formato fixo e, em seguida, processadas por uma Rede Neural Convolucional. Essa etapa extrai

características visuais importantes de cada região, como bordas, texturas e padrões.

4. **Classificar Regiões:** As características extraídas são passadas para uma camada totalmente conectada (*fully connected layer*) ou um classificador, que avalia se a região corresponde a uma das classes de objetos pré-definidas, como "pessoa", "avião" ou "monitor de TV".

Figura 2 – Rede Neural Convolucional baseada em Regiões (R-CNN)



Fonte: Adaptado de Potrimba (2023).

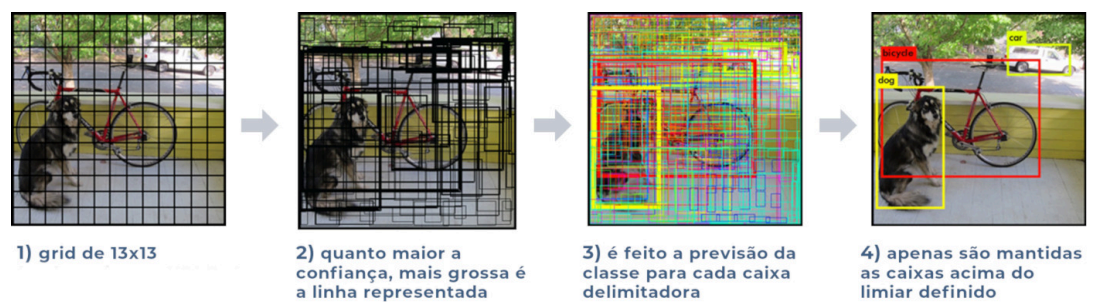
2.3 YOLO e YOLOv8

YOLO (*You Only Look Once*) é um algoritmo de detecção de objetos que revolucionou a área de visão computacional devido à sua eficiência. Ao contrário de abordagens tradicionais como R-CNN, que dividem o processo de detecção em várias etapas (propostas de regiões, extração de características e classificação), YOLO adota uma abordagem de detecção única (*single-shot detection*), processando toda a imagem em uma única etapa.

A Figura 3 apresenta o funcionamento do YOLO, onde a imagem de entrada é dividida em uma grade de 13x13. Cada célula dessa grade é responsável por prever caixas delimitadoras (*bounding boxes*) e determinar a classe dos objetos nelas contidos, juntamente com uma pontuação de confiança (ALVES 2020). Podemos resumir a Figura 3 da seguinte forma:

1. **Divisão em grade:** A imagem é dividida em uma grade onde cada célula observa uma parte da imagem.
2. **Confiança e caixas delimitadoras:** A espessura das linhas indicadas representa a confiança de detecção para cada caixa.
3. **Predição de classes:** Para cada caixa delimitadora, o modelo prevê a classe correspondente (por exemplo, cachorro, bicicleta).
4. **Refinamento de resultados:** Apenas as caixas com confiança acima de um limite pré-definido são mantidas, eliminando redundâncias e sobreposições, como mostrado no resultado final.

Figura 3 – Funcionamento do YOLO



Fonte: ALVES (2020).

A principal vantagem do YOLO é sua rapidez, permitindo aplicações em tempo real. Porém, versões iniciais enfrentavam dificuldades com objetos muito pequenos ou próximos. O YOLOv8 é a oitava versão do YOLO, ele incorpora avanços tanto na arquitetura quanto em desempenho, melhorando a precisão e a eficiência. Ele foi escolhido para este trabalho por ser a versão mais recente e por sua melhora na detecção de objetos menores e com maior densidade (o que é o caso de bananas no cacho).

O YOLOv8 possui cinco variações de modelo, nos quais podem ser observados na Figura 4. O modelo escolhido foi o YOLOv8m (*medium*) por possuir o equilíbrio entre velocidade, precisão e seu custo computacional mediano. Os modelos YOLOv8n (*nano*) e YOLOv8s (*small*) são os mais rápidos e os modelos YOLOv8l (*large*) e YOLOv8x (*extra-large*) são os mais precisos. O Modelo YOLOv8M possui 169 camadas convolucionais e 26,2 milhões de parâmetros, essas informações podem ser encontradas no arquivo yolov8.yaml no repositório da Ultralytics (GITHUB-ULTRALYTICS, 2024).

Figura 4 – Funcionamento do YOLO

Modelo	tamanho (pixéis)	mAPval 50-95	Velocidade CPU ONNX (ms)	Velocidade A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	18.4	142.4	1.21	3.5	10.5
YOLOv8s	640	27.7	183.1	1.40	11.4	29.7
YOLOv8m	640	33.6	408.5	2.26	26.2	80.6
YOLOv8l	640	34.9	596.9	2.43	44.1	167.4
YOLOv8x	640	36.3	860.6	3.56	68.7	260.6

Fonte: ULTRALYTICS (2023).

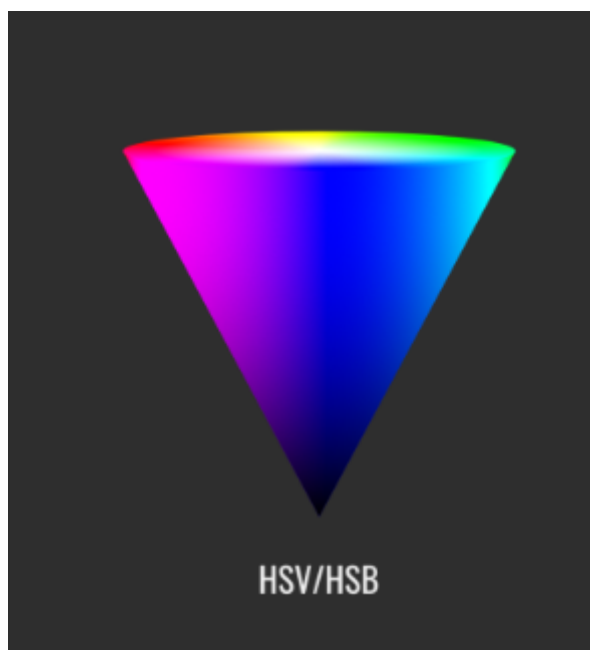
A seguir um breve resumo de cada coluna presente na figura 4:

- 1. Tamanho (pixels):** Resolução da imagem utilizada no modelo 640x640.
- 2. mAP@50-95:** Precisão do modelo; valores maiores indicam melhor desempenho.
- 3. Velocidade CPU ONNX (ms):** Tempo que o modelo leva para processar uma imagem quando rodado em uma CPU usando formato ONNX (milissegundos) ; menor é melhor.
- 4. Velocidade A100 TensorRT (ms):** Tempo de inferência (em milissegundos) quando rodado em uma GPU NVIDIA A100 usando TensorRT, que é uma tecnologia de otimização de inferência; menor é melhor.
- 5. Params (M):** Número de parâmetros do modelo em milhões; mais parâmetros aumenta a precisão, mas deixam o modelo mais pesado.
- 6. FLOPS (B):** Número de operações de ponto flutuante em bilhões; indica o custo computacional do modelo.

2.4 HSV (Hue, Saturation, Value)

É uma forma de representar as cores com base em três dimensões: Matiz (Hue) que representa a cor e é medido em graus, saturação (Saturation) que mede a intensidade da cor que varia em 0 a 100% e o brilho (Value) que varia de 0 a 100% e apresenta o valor da luminosidade da cor. A Figura 5 apresenta uma representação do HSV em três dimensões (MONTANARI, 2023).

Figura 5 – Representação 3D do HSV



Fonte: Adaptado de Montanari (2023).

3 DESENVOLVIMENTO

Neste capítulo, serão descritas as etapas do treinamento da inteligência artificial, incluindo: a criação de um banco de imagens de bananas; o processo de rotulação das imagens com caixas delimitadoras; os procedimentos de ampliação da base de dados através da edição das imagens originais, como a adição de ruídos, ajustes de rotação e brilho; o treinamento propriamente dito e os resultados obtidos. Além disso, será apresentada a análise da maturação das bananas por meio da sua coloração, bem como o desenvolvimento e funcionamento do aplicativo para celular.

3.1 Treinamento

Para o treinamento, foram selecionadas imagens de bananas em seus cachos, obtidas tanto na internet quanto na vizinhança do autor. Como o objetivo do trabalho é detectar o ponto de colheita da banana, as imagens utilizadas representam, majoritariamente, bananas verdes. O banco de imagens é composto por 800 imagens, das quais 665 foram destinadas ao treinamento, 100 à validação e o restante aos testes. A Figura 6 apresenta uma amostra das fotos utilizadas no treinamento.

Figura 6 – Amostra das imagens do treinamento



Para rotular as imagens, foram definidos dois rótulos: um para os cachos e outro para as bananas. Essa escolha foi estratégica, visando uma possível futura contagem de cachos em plantações, com o treinamento já adaptado para essa aplicação. Para facilitar o processo de rotulagem, utilizou-se o site [CVAT](#) (CVAT, 2025), que oferece atalhos para marcação de caixas delimitadoras e a opção de exportar os rótulos no formato compatível com o YOLOv8.

Há uma grande diversidade de tipos de bananas, e os números podem variar. Segundo Ullmann (2002), um cacho de banana nanica pode conter de 10 a 12 pencas, sendo que a primeira penca pode apresentar até 36 bananas, enquanto a última varia entre 6 e 8. Com base nisso, durante o processo de rotulagem das 665 imagens de treinamento, nas quais frequentemente apareciam mais de um cacho, foram realizadas aproximadamente 43 mil marcações, trabalho que exigiu vários dias de dedicação. A banana possui marcação de número “0” e o cacho marcação de número “1”.

Na Figura 7, é apresentada uma amostra das imagens com as marcações realizadas, mostrando as caixas delimitadoras para os cachos e as bananas.

Figura 7 – Caixas delimitadoras do treinamento



Já na Figura 8, ilustramos o pior caso de marcação para o treinamento, pois a quantidade de rótulos necessários foi da ordem de 500 devido à grande quantidade de bananas e cachos presentes na imagem. Essa imagem demonstra a complexidade de algumas etapas do processo de rotulagem, especialmente em cenários com alta densidade de objetos.

Figura 8 – Maior número de caixas delimitadoras em uma imagem



Fonte: Elaboração própria (2025).

O treinamento do modelo tornou-se muito mais simples ao utilizar o YOLOv8, pois a maioria dos parâmetros já vêm pré-configurados. Essas configurações podem ser encontradas na documentação oficial do Ultralytics (ULTRALYTICS, 2024). Foi necessário apenas definir o caminho das imagens, que correspondem àquelas previamente selecionadas, rotuladas e armazenadas no repositório do [Google Drive](#) (DRIVE, 2025), e o número de iterações, ajustado para 100 épocas (epochs=100) e o tamanho da imagem limitada em 640x640, padronizando assim as dimensões das imagens e facilitando na hora do treinamento. Por não termos à disposição um computador com boa capacidade computacional, recorreu-se à versão paga do Google Colab, que oferece suporte a GPUs mais potentes, acelerando significativamente o processo de treinamento. Utilizando de GPUs A100 o treinamento levou 8 horas para ser concluído. O código

completo utilizado para o treinamento está disponível no repositório do projeto no [Github](#) (GITHUB, 2025), no arquivo *TCC_colab_treinamento*.

Apesar da grande quantidade de imagens e rótulos utilizados no treinamento, o resultado inicial não atingiu as expectativas. Diversas bananas e cachos não foram corretamente reconhecidos pelo modelo, quando elas estão a uma certa distância da câmera, evidenciando a necessidade de ajustes e refinamentos no processo. Na Figura 9, é apresentada uma das predições realizadas após esse primeiro treinamento. É possível observar as limitações enfrentadas, com a banana próxima na foto elas são bem reconhecidas e a distância apenas o cacho é reconhecido. Os números acima da banana são referentes ao label “0” e a % de certeza de ter sido reconhecida 0.X, o mesmo acontece para os cachos, mas com o label de “1”

Figura 9 – Resultado do primeiro treinamento



Fonte: Elaboração própria (2025).

Para o segundo treinamento, o banco de imagens foi ampliado com ajustes realizados no acervo original por meio do site [Roboflow](#) (ROBOFLOW,2025). As imagens foram editadas com diversas alterações, incluindo: modificação da saturação entre -30% e +30%; rotação em sentido horário, anti-horário, para cima e para baixo; variação da cor entre -20% e +20%; ajuste do brilho entre -20% e +20%; e adição de efeito de desfoque de 2.5px. Esses valores foram sugeridos pela própria plataforma Roboflow, com essas alterações o banco de imagens foi ampliado em cerca de quatro vezes, totalizando agora 2.809 imagens. Um dos benefícios oferecidos pelo Roboflow foi a adaptação automática dos rótulos para as novas imagens geradas, o que facilitou significativamente a expansão do acervo e o preparo para um treinamento mais robusto.

O resultado do segundo treinamento apresentou melhorias significativas, como mostrado na Figura 10. Apesar de não ter sido configurado um limite para a porcentagem mínima de confiança na detecção de bananas e cachos, pois essa imagem foi gerada pela própria YOLO ao final do treinamento, o modelo identificou uma maior quantidade de bananas mesmo aquelas a uma distância razoável da câmera, evidenciando assim uma melhora considerável entre os dois treinamentos .

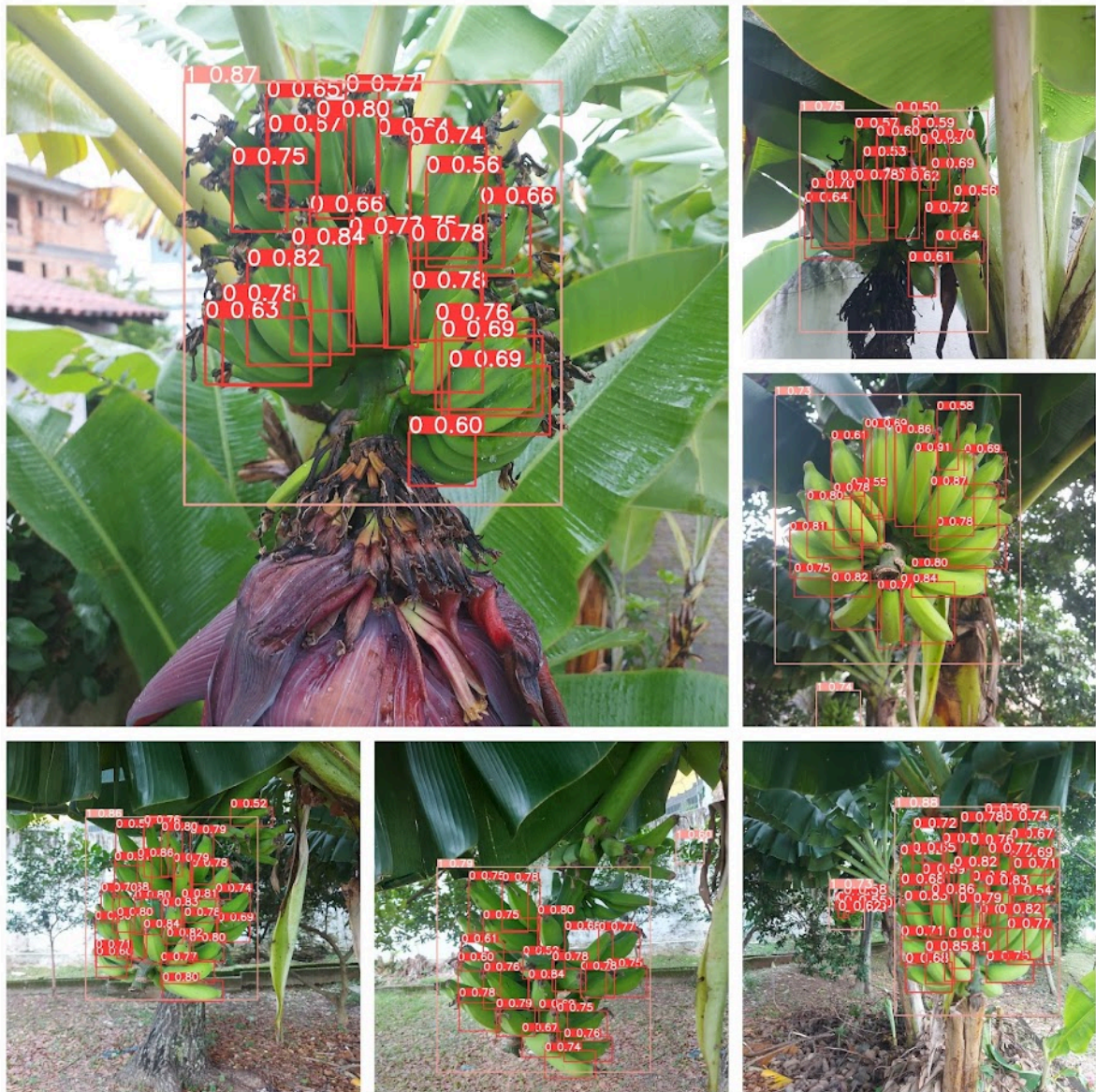
Figura 10 – Resultado do segundo treinamento



Fonte: Elaboração própria (2025).

A Figura 11 apresenta mais testes realizados para comprovar a eficiência do treinamento em detectar as bananas e os cachos. Para facilitar a visualização foi retirado os nomes das etiquetas, permanecendo apenas a numeração “0” para banana e “1” para os cachos. Observa-se que boa parte das bananas/cachos foram detectadas inclusive em casos nos quais elas estavam a uma certa distância, reforçando assim a eficiência do treinamento.

Figura 11 – Matriz de confusão normalizada



Fonte: Elaboração própria (2025).

A média obtida pela curva de Precisão x Recall (Figura 12) demonstra quão bem o modelo detecta e classifica os objetos nas imagens, onde a Precisão indica quantos objetos foram detectados de maneira correta no qual a fórmula é apresentada na Equação 1. A Equação 2 apresenta a fórmula do Recall, seu valor representa a quantidade de objetos reais detectados corretamente.

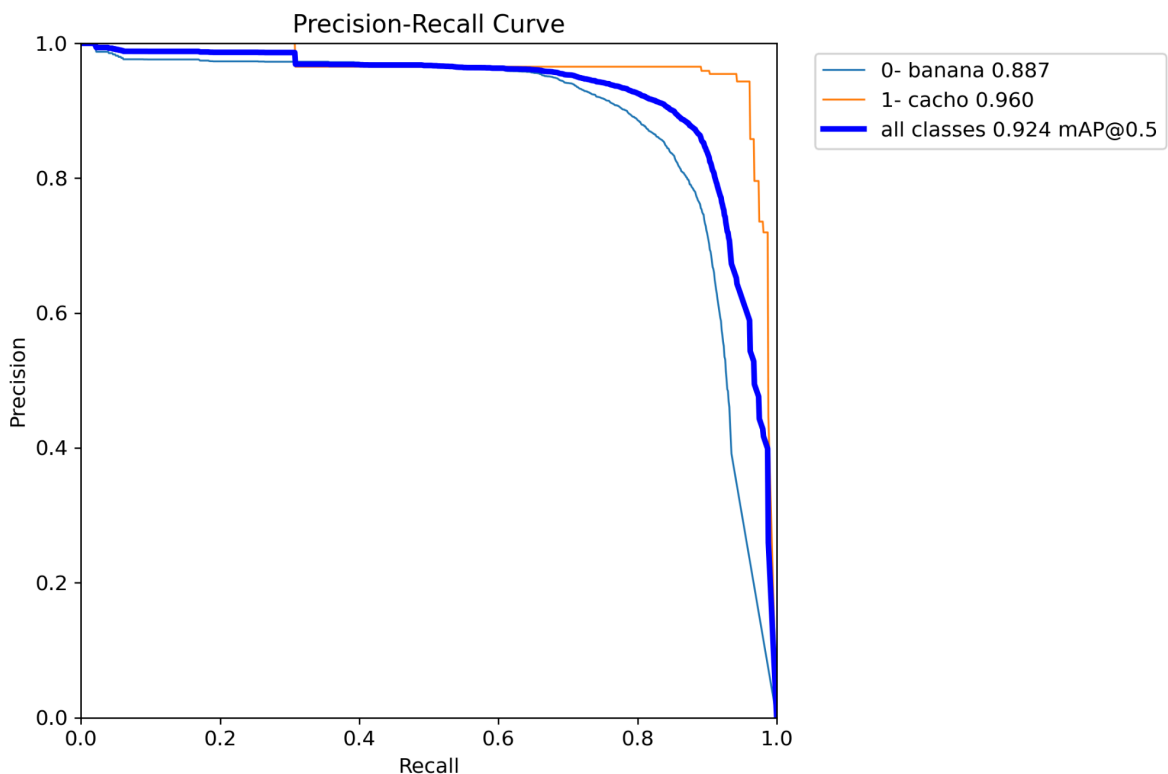
$$Precisão = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Positivos\ (FP)} \quad (1)$$

$$Recall = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Negativos\ (FN)} \quad (2)$$

Fonte: FILHO (2023).

Caso a precisão seja alta e o recall baixo, o modelo detecta poucos objetos, mas quase sem erros. Agora se a precisão é baixa e o recall possui valor alto, o modelo detecta muitos objetos mas com muitos erros, o ideal é encontrar um equilíbrio entre os dois valores. Para avaliar a precisão da detecção de objetos é utilizada a métrica do IoU (*Intersection over Union*), onde ele mede o quanto a caixa predita pelo modelo está sobreposta a caixa real delimitadas pela pessoa que está fazendo o treinamento, ou seja um IoU de 50% significa que a interseção entre a caixa predita e a real ocupa metade da união das duas caixas.

Figura 12 – Curva de Precisão x Recall



Fonte: Elaboração própria (2025).

O valor de mAP (*Mean Average Precision*) é baseado na Precisão x Recall e no IoU. O valor de mAP@0.5 presente na Figura 12, significa que para valores de IoU de 50% o modelo detecta e classifica os objetos com uma precisão de 92,4%.

3.2 Maturação da banana

Como um dos objetivos deste trabalho é a recomendação do ponto de maturação das bananas para colheita, inicialmente buscou-se verificar o nível de maturação das bananas utilizando máscaras para comparação com uma banana madura, ou ainda por meio da medição do diâmetro da fruta. No entanto, essas abordagens não se mostraram adequadas, pois não proporcionaram uma detecção genérica que pudesse ser aplicada a diferentes espécies de banana. Isso ocorre porque métodos tradicionais, como a verificação do diâmetro ou desaparecimento da angulosidade dos frutos, podem não ser eficientes para todas as variedades. Como destaca Chitarra (CHITARRA et al., 1998), tais métodos não são apropriados para a banana 'Prata' (Musa AAB), pois essa cultivar possui porte alto e frutos irregulares em tamanho e forma, dificultando a padronização da colheita com base nessas características.

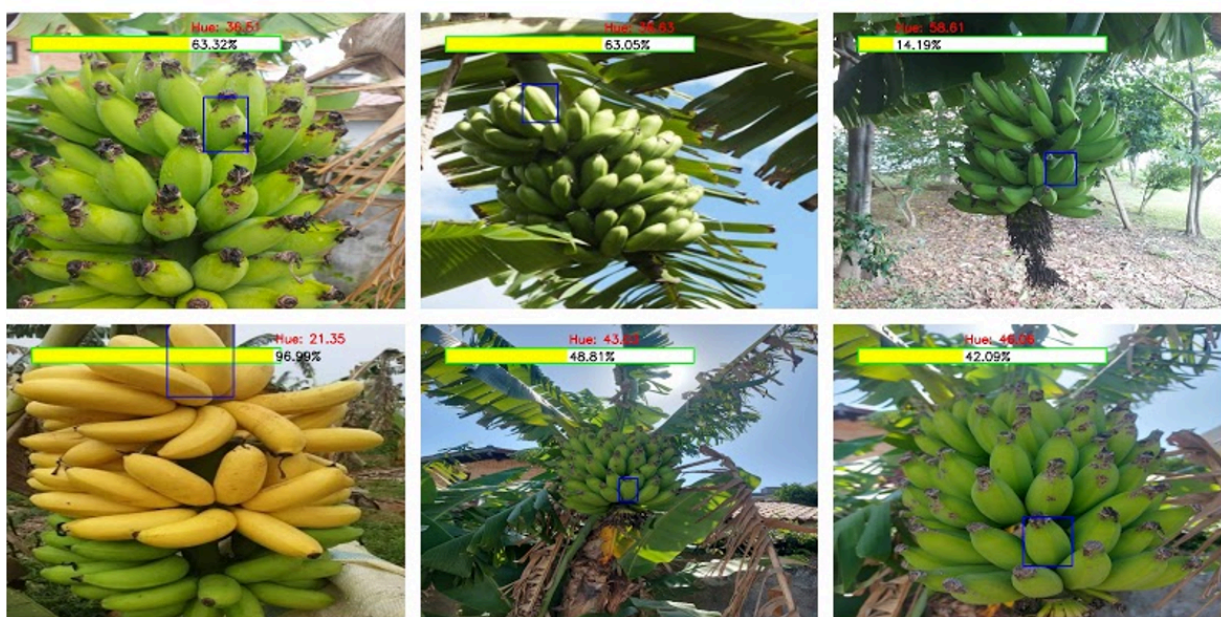
A solução encontrada foi analisar o nível de maturação das bananas por meio da coloração, utilizando o espaço de cores HSV (*Hue, Saturation, Value*). Nesse modelo, a matiz (Hue) representa a tonalidade da cor, medido em graus (0° a 360°); a Saturação (S) indica a pureza da cor, com valores mais baixos representando tons desbotados e mais altos, cores vibrantes; e o Value (V) refere-se ao brilho ou intensidade da cor (MOTA, 2020). Apenas o valor da matiz de cor (H) será analisado, enquanto os parâmetros (S) e (V) não terão restrição, permitindo que as bananas e os cachos sejam analisados tanto na sombra quanto sob o sol.

Para determinar a maturação, o valor de matiz foi definido como 65° para o verde (totalmente verde) e 20° para o amarelo (totalmente maduro). O trecho do código ([MATURAÇÃO](#), 2025) utiliza o valor médio de matiz (hue_mean) para calcular o percentual de maturação das bananas, com a fórmula apresentada na Equação 3, onde 65° corresponde a 0% de maturação e 20° a 100%.

$$maturação = \frac{65 - hue_mean}{65 - 20} \times 100 \quad (3)$$

Outra parte do código que merece destaque é a função *def draw_maturity_bar* (MATURAÇÃO, 2025). Este tem como objetivo desenhar uma barra de maturação diretamente na imagem analisada. Essa barra oferece uma representação visual intuitiva do nível de maturação das bananas, indicando a proximidade da coloração verde para o amarelo e, conseqüentemente, o percentual de maturação. Esse recurso facilita a interpretação dos resultados, permitindo ao usuário identificar rapidamente o estado de maturação da fruta de forma clara e prática. Na Figura 13, é possível observar alguns exemplos dessa barra aplicada às imagens analisadas pelo código em questão.

Figura 13 – Barra de maturação da banana



Fonte: Elaboração própria (2025).

Para facilitar a visualização e também a eficiência do código foi escolhido analisar apenas a banana, na qual a IA tem maior confiança, por isso que na Figura 13, apenas uma banana por foto está destacada.

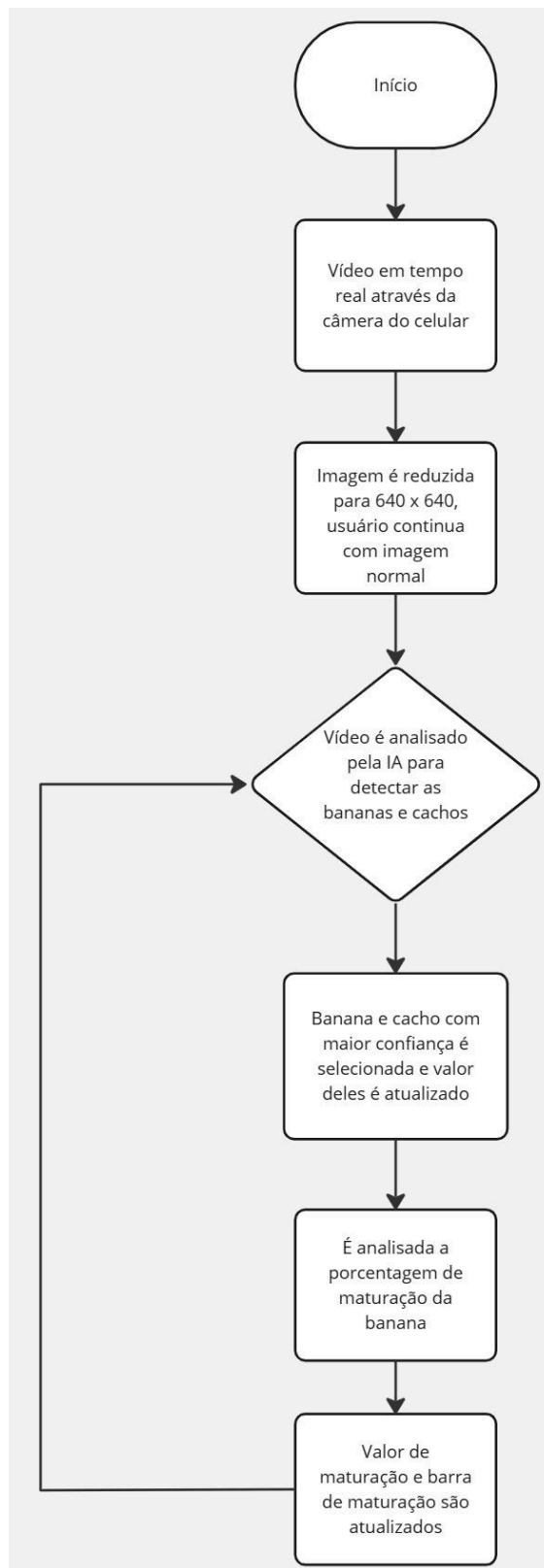
3.3 Aplicativo para celular

A ideia de criar um aplicativo para celular (app) surgiu devido ao fato de que a grande maioria da população mundial possui um *smartphone* nos dias atuais, ou ao menos o número de assinaturas móveis é de 8,58 bilhões enquanto a população mundial é de 7,95 bilhões em 2022 (RICHTER, 2023). Um app facilitaria o acesso a essa ferramenta de detecção do ponto de maturação da banana. O primeiro passo foi converter o arquivo *best.pt*, resultado do treinamento anterior, para um arquivo compatível com dispositivos móveis, o *best.tflite* que se encontra no repositório do [DRIVE2](#) (DRIVE2, 2025). Essa etapa de conversão e o desenvolvimento do aplicativo foram baseados no projeto do GitHub de Aarohi Singla (SINGLA, 2024).

Após a adaptação do código para a linguagem de desenvolvimento Android Kotlin, foram realizados testes na detecção do ponto de maturação da banana. Durante esses testes, verificou-se que a execução do modelo apresentava um alto consumo de processamento, tornando a detecção em vídeos inviável devido ao atraso de quase cinco segundos na atualização das caixas delimitadoras das bananas detectadas.

Inicialmente, considerou-se a implementação de uma opção de detecção por meio de imagens provenientes da galeria do celular. No entanto, para otimizar o código foi necessário reduzir a imagem analisada para 640x640, igual ao treinamento e selecionar através da IA a banana com maior precisão, no lugar de analisar todas as bananas encontradas a partir de um certo valor de precisão, por exemplo 60%. Com esses ajustes o tempo de resposta tornou-se aceitável para uso em tempo real. O vídeo demonstrando o funcionamento do aplicativo pode ser acessado por meio do repositório do [Google Drive](#) (Drive, 2025), junto também com os arquivos do [aplicativo](#) (APLICATIVO, 2025). O funcionamento do aplicativo pode ser representado no fluxograma da Figura 14. Nele, o modelo de detecção é executado diretamente no celular para identificar cachos e bananas na cena. Em seguida, a banana com a maior confiança na detecção é selecionada, e o sistema inicia o processo de análise da maturação, determinando sua porcentagem de amadurecimento.

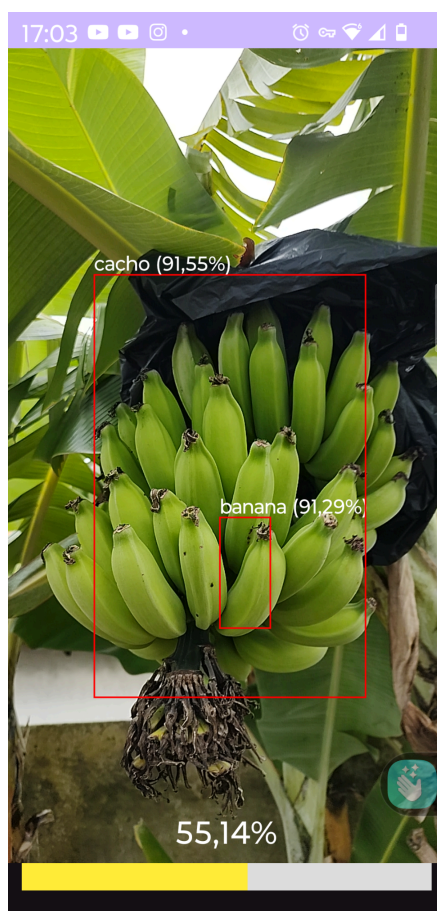
Figura 14– Fluxograma do aplicativo



Fonte: Elaboração própria (2025).

A Figura 15 apresenta a captura de tela do aplicativo, na qual os números acima das caixas delimitadoras representam a confiança da IA na detecção do cacho e da banana. Neste caso específico, optou-se por exibir apenas uma banana, a fim de reduzir a poluição visual e otimizar o tempo de resposta do aplicativo. Além disso, o número acima da barra de maturação indica o quão próximo o tom de verde da banana detectada está do amarelo, permitindo estimar seu nível de maturação.

Figura 15 – Funcionamento do aplicativo



Fonte: Elaboração própria (2025).

4 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou o desenvolvimento de um sistema para a detecção do ponto de maturação da banana por meio de sua coloração. Para isso, foi realizado o treinamento de uma inteligência artificial utilizando o modelo YOLOv8, permitindo a detecção da localização das bananas e dos cachos em imagens. Além disso, foram exploradas técnicas de processamento de imagens no espaço de cores HSV para estimar o nível de maturação do fruto com base no valor de Hue.

Durante o desenvolvimento, verificou-se que o treinamento inicial não foi satisfatório, exigindo a ampliação do banco de imagens e a aplicação de técnicas de aumento de dados para melhorar a robustez do modelo. Após esses ajustes, os resultados obtidos demonstraram uma precisão satisfatória na detecção dos cachos e das bananas. A análise da maturação, embora não seja uma medida exata, apresentou-se como uma solução viável para auxiliar produtores na tomada de decisão sobre o momento ideal para a colheita.

Além disso, a implementação do modelo em um aplicativo para dispositivos móveis possibilitou o uso prático da ferramenta, tornando-a acessível a um público mais amplo. No entanto, alguns desafios foram encontrados, principalmente relacionados ao alto consumo computacional para a detecção em tempo real via vídeo. Para contornar essa limitação, foram realizadas otimizações no processamento, incluindo a redução da resolução das imagens para 640x640 pixels e a análise da banana com maior grau de confiança, no lugar de analisar todas as bananas encontradas. Essa adaptação permitiu um processamento mais eficiente, tornando a análise das imagens mais ágil e melhorando a experiência do usuário sem comprometer significativamente a precisão da detecção.

4.1 Sugestões para trabalhos futuros

Apesar de ser um tema específico, o presente estudo pode ser expandido para diversas áreas relacionadas ao desenvolvimento e à colheita da banana. Uma possível melhoria seria a realização de um novo treinamento da IA para detectar doenças que possam afetar a bananeira, tornando o aplicativo mais robusto. Além disso, a implementação de um banco de dados permitiria acompanhar o

desenvolvimento de cada cacho individualmente, utilizando identificação por GPS ou nome, facilitando o monitoramento do tempo transcorrido desde a floração e agregando mais uma variável à análise do ponto de maturação.

Outra funcionalidade que poderia ser incorporada é a inserção da distância até o mercado consumidor, auxiliando na decisão do momento ideal para a colheita, considerando uma margem adicional para imprevistos no transporte. Adicionalmente, parâmetros físicos como o diâmetro e a curvatura da banana poderiam ser analisados para aprimorar a estimativa do ponto ótimo de colheita, complementando a abordagem baseada exclusivamente na coloração do fruto.

REFERÊNCIAS

UNITED NATIONS. **Total population by sex**. Disponível em: <https://population.un.org/dataportal/data/indicators/49/locations/900/start/1950/end/2100/line/linetimeplotsingle?df=5fec2c9f-98ca-4df4-9a61-43a08982884c> . Acesso em: 10 jan. 2025.

GOV. **Factsheet fome no mundo**. Disponível em: https://www.gov.br/secom/pt-br/arquivos/cria-g20/2411_cria-g20_factsheet-fomenomundo.pdf . Acesso em: 10 jan. 2025.

BITTENCOURT, DANIELA. **Artigo- Agricultura familiar, desafios e oportunidades rumo à inovação**. Disponível em: <https://www.embrapa.br/busca-de-noticias/-/noticia/31505030/artigo---agricultura-familiar-desafios-e-oportunidades-rumo-a-inovacao> . Acesso em: 10 jan. 2025.

SIAGRI. **Os desafios e oportunidades da agricultura familiar no Brasil**. Disponível em: <https://www.sagri.com.br/agricultura-familiar-no-brasil/> . Acesso em: 10 jan. 2025.

EMBRAPA. **Mandioca e fruticultura**. Disponível em: <https://www.embrapa.br/mandioca-e-fruticultura/cultivos/banana> . Acesso em: 10 jan. 2025.

CNA. **Dia da banana - Fruta é cultivada em todos os estados**. Disponível em: <https://www.cnabrazil.org.br/noticias/dia-da-banana-fruta-e-cultivada-em-todos-os-estados#:~:text=Segundo%20o%20BGE%2C%20a%20estimativa,%2C%20sendo%20exportado%20apenas%201%25> . Acesso em: 10 jan. 2025.

LIMA, Marcelo Bezerra; ALVES, Elio José; CARVALHO, José Eduardo Borges de; BORGES, Ana Lúcia. **Banana Colheita**. Disponível em: <https://www.embrapa.br/agencia-de-informacao-tecnologica/cultivos/banana/producao/tratos-culturais/colheita> . Acesso em: 13 jan. 2025.

PAIVA, Lázaro Eurípedes; FLORI, José Egídio; MOREIRA, Wellington Antônio; LIMA, Maria Auxiliadora Coelho de; BASSOI, Luís Henrique; ALENCAR, José Adalberto de. **Fruticultura irrigada**. 2011. Disponível em: <https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/897025/1/Fruticulturalrigada2011p.140155.pdf> . Acesso em: 13 jan. 2025.

AZURE, Microsoft. **What is computer vision**. Disponível em: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision#object-classification> . Acesso em: 15 jan. 2025.

ODEMAKINDE, ELISHA. **Tudo sobre a Mask R-CNN: Um guia para iniciantes**. Disponível em: <https://viso.ai/deep-learning/mask-r-cnn/> . Acesso em: 15 jan. 2025.

POTRIMBA, PETRU. **O que é R-CNN?**. Disponível em: <https://blog.roboflow.com/what-is-r-cnn/> . Acesso em: 15 jan. 2025.

ALVES, GABRIEL. **Deteção de objetos com YOLO - Uma abordagem moderna.** Disponível em: <https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/> . Acesso em: 15 jan. 2025.

GITHUB, ULTRALYTICS. **Yolov8.yaml.** Disponível em: <https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/v8/yolov8.yaml> / . Acesso em: 27 mar. 2025.

MONTANARI, TOBIA. **HSL and HSV Explained: Which Color Model Should You Use?.** Disponível em: <https://www.tobiamontanari.com/hsl-and-hsv-explained-which-color-model-should-you-use/> . Acesso em: 27 mar. 2025.

ULTRALYTICS. **Explorar o Ultralytics YOLOv8.** Disponível em: <https://docs.ultralytics.com/pt/models/yolov8/> . Acesso em: 27 fev. 2025.

CVAT. **Computer vision annotation tool.** Disponível em: <https://app.cvat.ai/> . Acesso em: 27 mar. 2025.

ULLMANN, SAMANTA. **Principais cultivares industriais.** Disponível em: <https://www.ufrgs.br/alimentus1/feira/mpfruta/banana/industi.htm#:~:text=O%20cach,o%20se%20apresenta%20com,com%208%20a%2010%20cm> . Acesso em: 16 jan. 2025.

ULTRALYTICS. **Predefinições** . Disponível em: <https://docs.ultralytics.com/pt/usage/cfg/#train-settings> . Acesso em: 27 mar. 2025.

DRIVE. **Final_completo.** Disponível em: <https://drive.google.com/drive/folders/1iazS2pUUZGhzTqmD1s5UmQeUibFhqxKk?usp=sharing> . Acesso em: 16 jan. 2025.

GITHUB. **Repositório TCC.** Disponível em: <https://github.com/suzuki1994/TCC> . Acesso em: 16 jan. 2025.

ROBOFLOW. **Roboflow** . Disponível em: <https://app.roboflow.com/> . Acesso em: 16 jan. 2025.

MATURAÇÃO. **Repositório TCC.** Disponível em: <https://github.com/suzuki1994/TCC/blob/main/bananacorfinal.py> . Acesso em: 16 jan. 2025.

CHITARRA, Maria Isabel Fernandes; SANTOS, Jorge Eduardo da Silva. **Relação entre a idade do cacho de banana ‘prata’ à colheita e a qualidade dos frutos após a colheita.** Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/45112/1/RELACAO-ENTRE-A-IDADE-DO-CACHO-DE-BANANA.pdf> . Acesso em: 20 fev. 2025.

FILHO, Mario. **Precisão, Recall e F1 Score em Machine learning**. Disponível em: <https://mariofilho.com/precisao-recall-e-f1-score-em-machine-learning/> . Acesso em: 27 mar. 2025.

MOTA, Suzuana. **Como identificar cores no mundo real utilizando visão computacional?**. Disponível em: <https://suzana-svm.medium.com/vis%C3%A3o-computacional-detec%C3%A7%C3%A3o-de-cores-em-tempo-real-utilizando-python-e-opencv-a466444d40e> . Acesso em: 19 jan. 2025.

DRIVE2. **tflite**. Disponível em: <https://drive.google.com/drive/folders/1vI7rM7bxWvHXpcESwJ7-cbyXCtu-CYVc?usp=sharing> . Acesso em: 22 jan. 2025.

SINGLA, AAROHI. **Object detection app usion YOLOv8 and Android**. Disponível em: <https://github.com/AarohiSingla/Object-Detection-Android-App> . Acesso em: 22 jan. 2025.

RICHTER, Félix. **Gráfico: Existem mais telemóveis do que pessoas no mundo**. Disponível em: <https://www.weforum.org/stories/2023/04/charted-there-are-more-phones-than-people-in-the-world/> . Acesso em: 05 fev. 2025.

Drive. **Banana video**. Disponível em: <https://drive.google.com/file/d/14vw7zuCBpsP8tl2IkEazY2YMkP4vqTz1/view?usp=sharing> . Acesso em: 06 fev. 2025.

APLICATIVO. **android_app.zip**. Disponível em: <https://drive.google.com/file/d/1i5L3vzHPXuoqwqYHBQl9gWGHIK3hXefb/view?usp=sharing> . Acesso em: 27 mar. 2025.