

DETECÇÃO E MEDIÇÃO DO CRESCIMENTO DE PLANTAÇÃO POR PROCESSAMENTO DE IMAGENS - UMA APLICAÇÃO INTEGRADA AO SMART CAMPUS E À HORTA AUTOMATIZADA DO IMT

Ricardo Banci ¹; Fernando de Almeida Martins ²; Rogério Cassares Pires³; Alessandra Dutra Coelho ²; Wânderson de Oliveira Assis ²

¹ Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);

² Professor do Instituto Mauá de Tecnologia (IMT);

³ Engenheiro do Centro de Pesquisas (IMT)

Resumo. *A pesquisa propôs a implementação da visão computacional para monitorar e apresentar dados de altura e taxa de crescimento de plantas no dashboard do SmartCampus da Mauá. A análise do crescimento das plantas visa possibilitar, em estudos futuros, a correlação da taxa de crescimento com variáveis ambientais como umidade, temperatura, aplicação de fertilizantes e época do ano. O estudo incluiu a familiarização com o YOLOv8, treinamento com conjuntos de dados personalizados e a verificação da viabilidade dessa ferramenta na parte da visão computacional da agricultura de precisão quando implementado com IoT. Foram identificados desafios e suas respectivas soluções com o uso do YOLOv8, como a dificuldade do modelo em identificar plantas sem alguma especificação e em situações de sobreposição. Essas descobertas contribuem para aprimorar a aplicação prática dessa tecnologia na monitorização eficiente do crescimento das plantas de forma simplificada.*

Introdução

Com os avanços contínuos no processamento computacional e a crescente relevância da inteligência artificial, a aplicação da visão computacional na agricultura de precisão torna-se cada vez mais viável e acessível. A capacidade da visão computacional em delimitar objetos específicos, como plantas, possibilita uma análise detalhada do crescimento vegetal, tornando-se uma ferramenta de aplicação simples e eficaz para otimização agrícola. Este estudo visa apresentar, por meio do *dashboard* do *SmartCampus* da Mauá a taxa de crescimento e a altura das plantas, permitindo uma análise abrangente do seu desenvolvimento e possibilita, em futuros estudos, comparar a taxa de crescimento com variáveis ambientais, como umidade, temperatura e época do ano, para compreender as interações complexas que influenciam o crescimento e saúde das plantas.

Ao longo de mais de uma década, a agricultura de precisão tem se destacado de maneira significativa na comunidade agrícola. Embora muitos agricultores ainda não adotem completamente esse novo conceito, observa-se um empenho considerável em pesquisa, uma disposição entusiasmada por parte dos consultores em aprender, e o surgimento contínuo de equipamentos cada vez mais sofisticados por parte de empresas de engenharia agrícola e outras, com o objetivo de facilitar a prática da agricultura de precisão (Srinivasan, 2006). Nesta Iniciação, aborda-se a utilização do método de visão computacional YOLO em sua versão 8, e mesmo que adotada de maneira simplificada, demonstra ser de fácil aplicação, possibilitando que pequenos agricultores e entusiastas integrem a visão computacional em suas plantações, ampliando o acesso a tecnologias avançadas e promovendo uma abordagem mais eficiente na gestão agrícola.

Material e Métodos

O objetivo principal do trabalho consiste em capturar imagens da horta por meio do protocolo RTSP (*Real-Time Streaming*), processá-las utilizando o YOLOv8 para extrair as coordenadas do retângulo delimitador e disponibilizar os dados da detecção via API (*Application Programming Interface*) para gerar soluções inteligentes de integração de baixo custo.

Captura das imagens

O projeto *SmartCampus* é uma iniciativa do IMT que busca conectar o *campus* empregando soluções de Internet das Coisas como LoRaWAN (*Long Range Wide Area Network*), Wi-Fi, Ethernet, 3G/4G entre outras que se adequem ao contexto de *Smart Cities*.

Para realizar o trabalho foram obtidas as imagens uma câmera de segurança do *SmartCampus* que transmite, em tempo real e de forma contínua, as imagens da horta. Para capturar as imagens dessa câmera, empregou-se uma biblioteca capaz trabalhar com o protocolo RTSP (Amaral, 2022), viabilizando uma abordagem dinâmica e interativa, essencial para o escopo deste projeto de iniciação científica. A partir das imagens disponíveis, deve-se processá-las com YOLO.

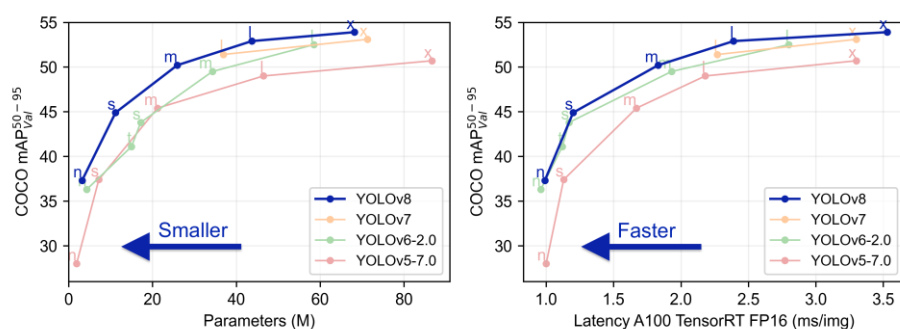
Figura 1 - Funcionamento do RTSP



YOLOv8

Este estudo tem como objetivo a detecção precisa da presença da planta escolhida, bem como a determinação de sua posição e altura utilizando as imagens capturadas. Para atingir esse propósito, adotou-se o YOLOv8 como a principal ferramenta de visão computacional. A escolha pelo YOLOv8 baseou-se em sua eficácia, evidenciada por resultados superiores em comparação com versões anteriores. Além disso, a simplicidade de sua aplicação contribui significativamente para a eficiência e praticidade na implementação do método.

Figura 2 - Comparação entre versões de YOLO



O YOLO, desenvolvido por Joseph Redmon e Ali Farhadi durante seus estudos de doutorado em 2015, representa uma inovação significativa em relação aos algoritmos anteriores

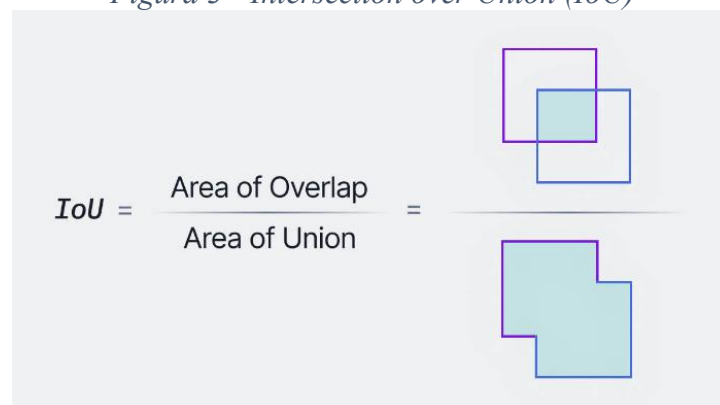
de detecção de objetos, como R-CNN ou *faster* R-CNN. Sua diferença reside na abordagem de "olhar" apenas uma vez pela imagem, o que justifica seu nome, *You Only Look Once* (YOLO). Diferentemente de seus predecessores, o YOLO adota uma estratégia única de dividir a imagem em um *grid* com, normalmente, 19 x 19 células. Se o centro do objeto estiver contido em uma determinada célula, deve ser a responsável por detectar o objeto. Cada célula realiza previsões de caixas delimitadoras e suas respectivas probabilidades. Então, o YOLO prevê múltiplas caixas delimitadoras por célula do *grid*. No entanto, é desejável apenas uma para cada objeto. Para alcançar isso, o YOLO atribui a responsabilidade a um algoritmo que prevê com base na maior Interseção sobre União (IoU) atual com a verdadeira posição do objeto.

Duas das métricas de avaliação mais comuns para modelos de detecção de objetos são a Interseção sobre União (IoU) e as métricas de Precisão Média (AP).

Intersection over Union (IoU)

A IoU é uma métrica popular para medir a precisão de localização e calcular erros de localização em modelos de detecção de objetos. Para calcular a IoU entre as caixas delimitadoras previstas e as caixas delimitadoras verdadeiras, primeiramente deve ser determinada a área de interseção entre as duas caixas correspondentes para o mesmo objeto. Em seguida, calcula-se a área total coberta pelas duas caixas delimitadoras, conhecida como "União", e a área de sobreposição entre elas chamada de "Interseção", como ilustrado na figura 3. A interseção dividida pela União fornece a razão da sobreposição para a área total, oferecendo uma boa estimativa de quão próxima a caixa delimitadora prevista está da caixa delimitadora original. A explicação do funcionamento do YOLO e do IoU foi retirada do blog (Kundu,2023).

Figura 3 - Intersection over Union (IoU)



Precisão e Recall

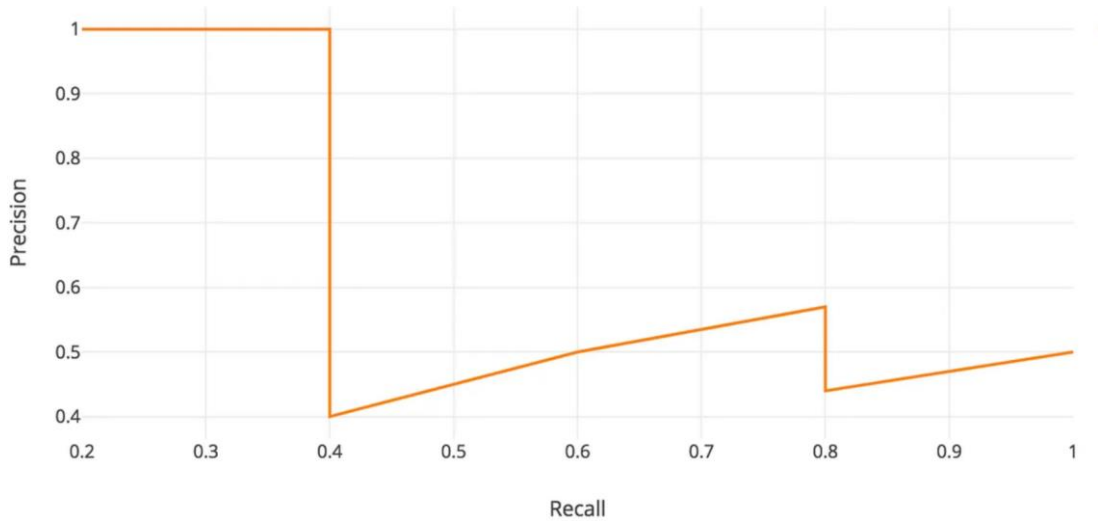
A Precisão mede quão precisa são as previsões, ou seja, a porcentagem de suas previsões que estão corretas.

O Recall mede quão bem se encontra todos os casos positivos, ou seja, a porcentagem das imagens que foram classificadas em uma categoria em comparação com todas as imagens que deveriam ser classificadas nessa categoria.

Average Precision (AP)

A Acurácia Média (AP) é calculada como a área sob a curva de precisão *versus recall* para um conjunto de previsões. O *recall* e a precisão oferecem uma compensação representada graficamente em uma curva ao variar o limiar de classificação. A área sob esta curva de precisão *versus recall* fornece a Acurácia Média por classe para o modelo. A média desse valor, ao longo de todas as classes, é chamada de Precisão Média Global (mAP).

Figura 4 - Exemplo de gráfico precisão x Recall



A acurácia média é a área abaixo da curva representada na figura 4, ou seja, sua integral:

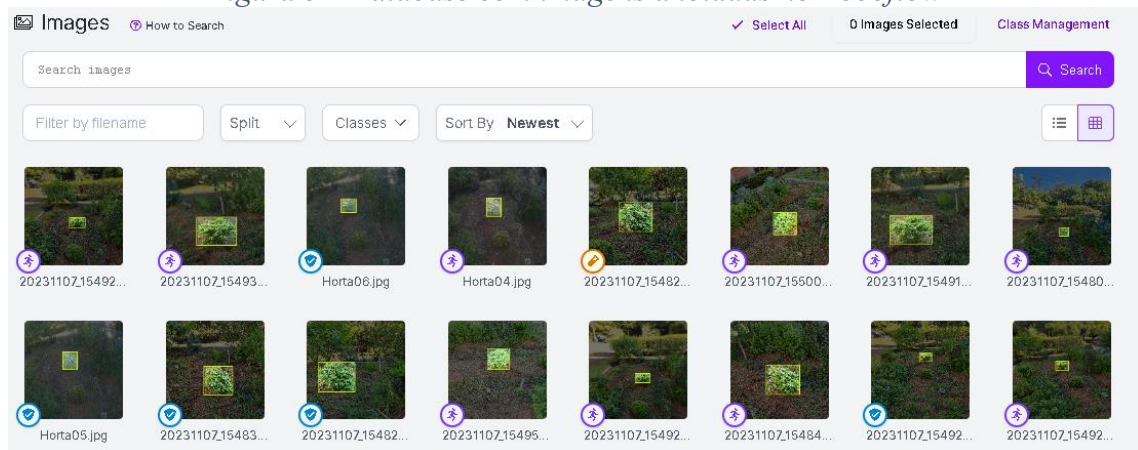
$$AP = \int_0^1 p(r)dr$$

Em detecção de objetos, precisão e *recall* não são usadas como previsões de classe. Em vez disso, elas servem como previsões de caixas delimitadoras para medir o desempenho das decisões. Um valor de IoU maior que 0,5 é considerado uma previsão positiva, enquanto um valor de IoU menor que 0,5 é considerado uma previsão negativa. As informações sobre as métricas utilizadas pelo YOLO foram obtidos do site (Hui, 2018).

Treinamento YOLOv8

O processo de treinamento dos modelos utilizando YOLOv8 inicia-se pela criação do *dataset*, que consiste em imagens e suas anotações. Para facilitar esse processo, foram utilizadas as ferramentas fornecidas pelo Roboflow, um site que disponibiliza *datasets* feitos pela comunidade e oferece uma interface simples para realizar as anotações. A figura 5 mostra algumas imagens do *dataset* utilizado no trabalho.

Figura 5 - Database com imagens anotadas no Roboflow



Posteriormente, para realizar o treinamento dos modelos, optou-se por utilizar o ambiente de desenvolvimento Google Colab. Este editor de código proporciona acesso gratuito à GPU T4.

Figura 6 - Treinamento do modelo

```
GPU T4

[ ] !nvidia-smi

Install YOLOv8 from ultralytics

[ ] !pip install ultralytics

[ ] from ultralytics import YOLO
import os
from IPython.display import display, Image
from IPython import display
display.clear_output()
!yolo mode=checks

Train YOLOv8 Model on Custom Dataset

[ ] !pip install roboflow

[ ] from roboflow import Roboflow
rf = Roboflow(api_key=[REDACTED])
project = rf.workspace("inicio-cientifica").project("pimenteira-camera-cell-detection")
dataset = project.version(2).download("yolov8")

[ ] !yolo task=detect mode=train model=yolov8m.pt data={dataset.location}/data.yaml epochs=100 imgsz=640

[ ] Image(filename=f'/content/runs/detect/train2/confusion_matrix.png', width=600)

[ ] Image(filename=f'/content/runs/detect/train2/results.png', width=600)
```

A primeira etapa foi verificar se o T4 GPU fornecido pela Google já foi configurado como o ambiente de execução. Em seguida, foi realizada a instalação da biblioteca Ultralytics, que é responsável por baixar todas as dependências necessárias para a utilização do YOLOv8. Vale ressaltar que este processo é significativamente simplificado em comparação com versões

anteriores, que exigiam que o usuário clonasse o repositório no Git. Além disso, foi importada a biblioteca Roboflow, que simplifica o processo de utilização do *dataset* previamente criado.

Com o ambiente de execução devidamente configurado, as bibliotecas instaladas, a etapa seguinte envolve a importação do *dataset* e a execução do treinamento dos dados, sendo utilizado um total de 100 *epochs*, que é a quantidade de treinamentos que a inteligência artificial fará. Posteriormente, os resultados do treinamento podem ser visualizados nas linhas de código subsequentes, e o modelo treinado juntamente com o *dataset* podem ser baixados para análises posteriores.

Resultados e Discussões

Treinamento do modelo YOLOv8 para detecção eficiente de plantas na Horta

A ferramenta YOLOv8 já possui *datasets* disponíveis para uso, como o COCO (*Common Objects in Context*), que detectam várias classes diferentes. Porém, para o caso proposto, os resultados apresentados não são suficientes. A detecção foi feita com uma boa precisão, como mostrada na figura 7, delimitando corretamente os limites das plantas, porém apenas algumas das plantas são identificadas, aproximadamente 23%, sendo classificadas como *potted plant*.

Figura 7 - Resultado com dataset COCO



Então foi realizado novos treinamento para desenvolver um modelo personalizado para a utilização com a arquitetura YOLOv8. O primeiro consistiu na criação de um conjunto de dados composto por 56 imagens capturadas por dispositivos móveis. Todas as plantas presentes na horta foram anotadas de forma genérica como 'Planta'.

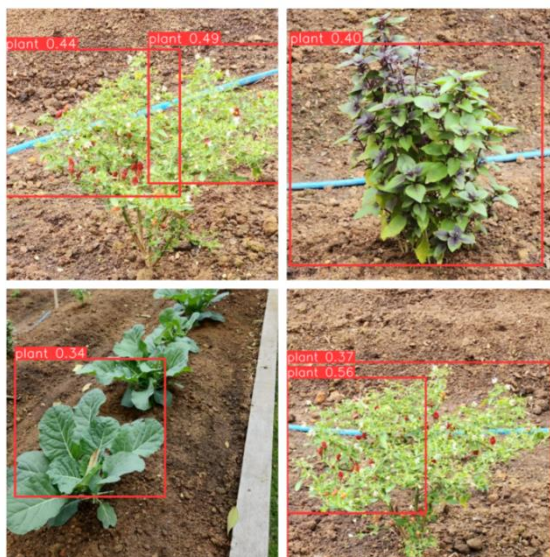
Entretanto, a detecção inicial apresentou baixa precisão, como mostra a figura 8, evidenciada por delimitações que não comportava as dimensões reais das plantas, ou que identificava plantas onde não havia. Observou-se ainda a ocorrência de sobreposições inadequadas, como a identificação de múltiplas plantas em uma única. Adicionalmente, constatou-se que diversas plantas não foram identificadas pelo modelo.

Figura 8 - Resultados com dataset personalizado



Em uma segunda etapa, realizou-se um treinamento adicional utilizando um conjunto de dados obtido do site Roboflow, composto por 6.000 imagens que abrangiam uma variedade de tipos de plantas, todas devidamente anotadas. Contudo, os resultados obtidos neste treinamento também se revelaram insatisfatórios, como mostra a figura 9, com os mesmos problemas apresentados pelo anterior.

Figura 9 - Resultados com dataset do Roboflow



Diante das limitações observadas nos treinamentos anteriores, algumas conclusões foram alcançadas. Ficou evidente que a sobreposição de plantas e a generalização na identificação de diferentes tipos de plantas por parte dos modelos resultaram em desafios significativos para a Inteligência Artificial.

Como uma medida corretiva, ajustamos a posição da câmera responsável por capturar as imagens da horta, reposicionando-a ligeiramente mais alta e direcionando-a para baixo. Essa mudança visa evitar a sobreposição das plantas na imagem, proporcionando um ambiente mais favorável para a detecção.

Adicionalmente, foi realizado um novo treinamento com um modelo específico para identificação de um único tipo de planta, neste caso, a pimenteira. Essa abordagem visou

simplificar a tarefa da rede neural ao focar na detecção de uma espécie específica, buscando melhorar a precisão e a eficácia da detecção.

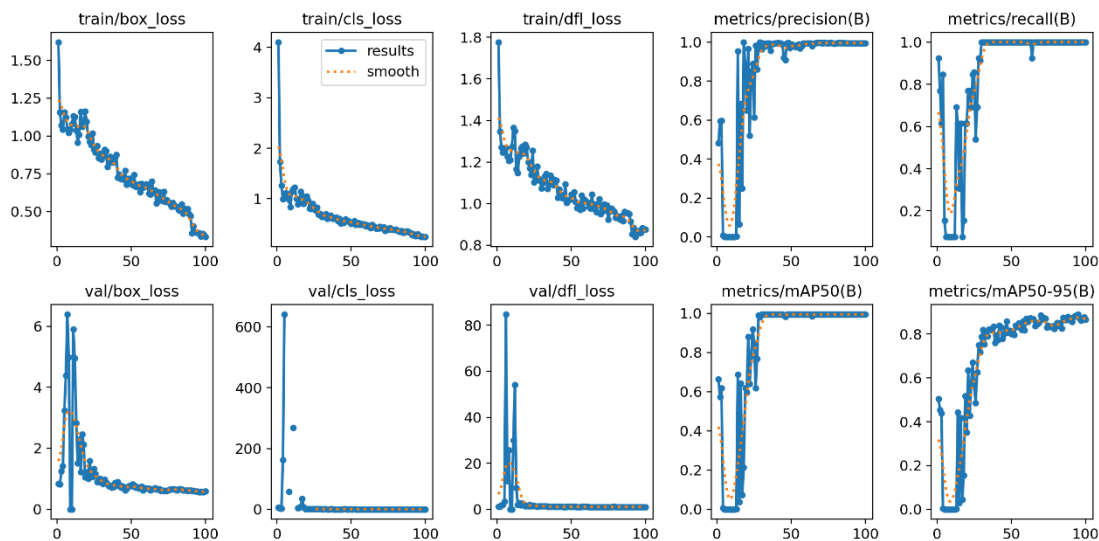
Finalmente, o último treinamento realizado demonstrou resultados promissores, apresentados na figura 10, evidenciando a capacidade do modelo em identificar corretamente a planta-alvo, apresentando uma notável precisão.

Figura 10 – Resultados do último dataset



Os resultados do treinamento do modelo com este conjunto de dados também se mostraram extremamente satisfatórios, conforme evidenciado pelas métricas apresentadas após o treinamento, as quais são exibidas na Figura 11.

Figura 11 - Resultados do treinamento final



A análise gráfica revelou quedas consistentes na *loss* e um crescimento significativo nas métricas de desempenho ao longo das 100 *epochs* de treinamento, destacando a notável capacidade de aprendizado e adaptação do modelo.

Pôde-se notar um desempenho positivo durante o treinamento ao longo dos *epochs*. Os gráficos que representam a *loss* exibem uma tendência decrescente, indicando uma redução nas perdas ao longo do tempo. Além disso, os gráficos de precisão, recall e mAP demonstram uma

tendência de crescimento, sugerindo melhorias nas métricas de desempenho ao longo do treinamento.

Conclusões

Após uma série de treinamentos, foi possível desenvolver um *dataset* otimizado, com foco na anotação da pimenteira, e a prevenção da sobreposição entre as plantas na horta. Este processo resultou em um notável treinamento do YOLOv8, demonstrando impressionantes índices de precisão, *recall* e mAP.

Apesar dos desafios enfrentados, a implementação do YOLOv8 na detecção de plantas foi simples, proporcionando resultados sólidos para a coleta de dados. Ao seguir uma abordagem cuidadosa, acredita-se que desafios como sobreposição e falta de especificidade podem ser eficientemente abordados, permitindo uma aplicação mais ampla na agricultura de precisão.

Para futuras pesquisas, sugere-se explorar como variáveis ambientais podem influenciar o tamanho das plantas. A análise das correlações entre o crescimento da planta e variáveis ambientais oferece uma perspectiva valiosa para estudos futuros, fornecendo conclusões importantes para a agricultura de precisão.

Referências Bibliográficas

- Srinivasan, A. (2006) Precision Agriculture: An Overview. 1st Edition. CRC Press.
- Hui, J. (2018) mAP (mean Average Precision) for Object Detection. <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. Acesso em: 10/11/2023.
- Kundu, R. YOLO: Algorithm for Object Detection Explained [+Examples]. <https://www.v7labs.com/blog/yolo-object-detection>. Acesso em: 10/11/2023.
- Redmon, J., "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
- Martins, C. Protocolo RTSP para câmeras IPs. <https://aprendacftv.com/protocolo-rtsp-para-cameras-ips> Acesso em: 10/11/2023.
- Amaral, C. (2022) O que é o protocolo RTSP e para que ele serve? <https://k2ponto.com.br/blog/o-que-e-o-protocolo-rtsp-e-para-que-ele-serve>. Acesso em: 10/11/2023.