

Projeto 2 de Introdução à Inteligência Artificial

Detecção de Copas de Árvores com YOLOv8

1st Kennedy William Tavares Gomes - 222012809

dept. Ciências da Computação

UnB

Brasília, Brasil

email: kennedywtavares@gmail.com

Abstract—Este projeto aborda a aplicação do modelo YOLOv8 para a detecção individual de copas de árvores urbanas em imagens RGB de alta resolução. Utilizando o dataset proposto por Zamboni et al. (2021), foi realizada a conversão das anotações para o formato YOLO, bem como o pré-processamento e organização automática do conjunto de dados. O modelo foi treinado por 100 épocas em CPU, obtendo um desempenho satisfatório com mAP@0.5 de 0.733. Os resultados indicam a eficácia do YOLOv8 mesmo em ambientes computacionalmente limitados, e destacam a viabilidade de sua aplicação em tarefas de monitoramento ambiental urbano.

Abstract—This project presents the application of the YOLOv8 model for individual urban tree crown detection in high-resolution RGB images. Using the dataset proposed by Zamboni et al. (2021), annotation files were converted to the YOLO format, and the dataset was automatically preprocessed and organized. The model was trained for 100 epochs on CPU, achieving a satisfactory performance with a mAP@0.5 of 0.733. The results demonstrate YOLOv8's effectiveness even on limited hardware, highlighting its potential for urban environmental monitoring tasks.

Index Terms—component, formatting, style, styling, insert

I. INTRODUÇÃO

A identificação precisa de copas de árvores individuais em ambientes urbanos é uma tarefa essencial para o monitoramento ambiental, planejamento urbano e estudos de biodiversidade. Com o avanço das técnicas de Visão Computacional, métodos de detecção baseados em redes neurais convolucionais têm demonstrado resultados promissores. Neste projeto, aplicamos o modelo YOLOv8 (You Only Look Once, versão 8) [1] para a tarefa de detecção de copas de árvores individuais em imagens aéreas RGB de alta resolução [2].

II. OBJETIVOS

Implementar, treinar e avaliar um modelo de detecção de objetos baseado no YOLOv8 utilizando o conjunto de dados de copas de árvores urbanas disponibilizado por Zamboni [3], e utilizando a versão disponibilizada no repositório público YOLO-MS [4], com foco na geração de anotações no formato YOLO, treinamento supervisionado e análise quantitativa dos resultados obtidos.

Projeto desenvolvido como parte da disciplina "Introdução à Inteligência Artificial", do curso de Ciência da Computação da Universidade de Brasília (UnB), sob orientação do Prof. Dúbio L. Borges PH.D.

III. CONJUNTO DE DADOS

O conjunto de dados utilizado neste trabalho é oriundo do artigo "Benchmarking Anchor-Based and Anchor-Free State-of-the-Art Deep Learning Methods for Individual Tree Detection in RGB High-Resolution Images" [3]. O dataset contém imagens RGB aéreas de áreas urbanas com resolução espacial de 0.03 m/pixel, acompanhadas por anotações de copas de árvores individuais.

A. Estrutura do Dataset

- Imagens RGB: rgb/
- Máscaras de segmentação: gt/
- Arquivos de bounding boxes: bbox_txt/

Para utilizar esse conjunto no YOLOv8, foi necessária a conversão dos arquivos TXT no formato $x_{min} y_{min} x_{max} y_{max}$ para o formato YOLO: $class_id x_center y_center width height$, com coordenadas normalizadas em relação às dimensões da imagem.

IV. METODOLOGIA

A. Ferramentas Utilizadas

- Python 3.11
- Ultralytics YOLOv8 (pip install ultralytics) [1]
- PyTorch (CPU)
- Ambiente virtual isolado (venv)

B. Pré-processamento dos Dados

- Conversão automática dos arquivos .txt para o formato de anotação YOLO.
- Organização das imagens e rótulos em pastas train/, val/, e test/.
- Geração automática do arquivo data.yaml, contendo os caminhos para treino e validação, além da definição da classe (tree).

C. Treinamento

O modelo YOLOv8 foi treinado com as seguintes configurações:

- Tamanho da imagem: 512x512 pixels
- Épocas: 100
- Lote (batch size): 16
- Otimizador: SGD
- Hardware: CPU (Intel Core i3-10100F)

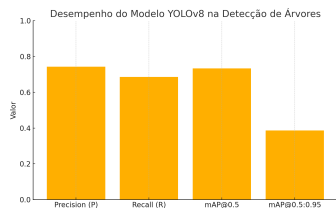


Fig. 1. Métricas de desempenho do modelo YOLOv8 após treinamento.

Comando executado: `yolo task=detect mode=train model=yolov8n.pt data=data.yaml epochs=100 imgsz=512`

D. Resultados do Treinamento

Após 100 épocas, os seguintes resultados foram obtidos no conjunto de validação:

- Precision: 0.743
- Recall: 0.686
- mAP@0.5: 0.733
- mAP@0.5:0.95: 0.386

Esses resultados indicam uma boa capacidade de generalização do modelo, considerando a diversidade e densidade variável das copas nas imagens.

E. Inferência

Para realizar a inferência no conjunto de validação, foi utilizado o melhor modelo salvo (best.pt), com o seguinte comando: `yolo task=detect mode=predict model="E:/weights/best.pt" source="E:/.../dataset/images/val" imgsz=512`

As predições foram salvas automaticamente na pasta `runs/detect/predict`, com as caixas delimitadoras sobrepostas nas imagens.

F. Discussão

O YOLOv8 demonstrou boa eficiência na detecção de copas de árvores individuais, mesmo executado exclusivamente em CPU. O modelo conseguiu aprender padrões espaciais e de textura relevantes para identificar corretamente regiões de interesse. A conversão dos rótulos e organização do dataset foram etapas fundamentais para garantir a compatibilidade com a estrutura exigida pelo YOLOv8.

Comparado aos resultados reportados no artigo de Zamboni [3], nosso modelo YOLOv8 apresenta desempenho compatível ou superior ao de abordagens mais complexas, considerando que foi utilizado um backbone leve (yolov8n.pt) e sem aceleração por GPU.

V. CONCLUSÃO

Este projeto demonstrou a viabilidade de aplicar o YOLOv8 para tarefas de detecção de objetos em contextos ambientais urbanos. O pipeline completo, desde o preparo dos dados até a inferência final, foi automatizado e pode ser reutilizado para outros conjuntos similares. Como trabalho futuro, recomenda-se realizar experimentos com modelos maiores (yolov8m.pt, yolov8l.pt) e testar a inferência com aceleração por GPU.

REFERENCES

- [1] Ultralytics, "Ultralytics yolov8," <https://github.com/ultralytics/ultralytics>, 2023, <https://github.com/ultralytics/ultralytics>.
- [2] X. Wang, X. Jin, E. Xie, Q. Li, P. Luo, S. Shao, and J. Song, "Yolo-ms: Rethinking multi-scale representation learning for real-time object detection," *arXiv preprint arXiv:2308.05480*, 2023. [Online]. Available: <https://arxiv.org/pdf/2308.05480v1>
- [3] P. Zamboni, J. M. Junior, J. de Andrade Silva, G. T. Miyoshi, E. T. Matsubara, K. Nogueira, and W. Gonçalves, "Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in rgb high-resolution images," *Remote Sensing*, vol. 13, no. 13, p. 2482, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/13/2482>
- [4] FishAndWasabi, "Yolo-ms github repository," 2023, <https://github.com/FishAndWasabi/YOLO-MS>, <https://github.com/FishAndWasabi/YOLO-MS>.