

Ultralytics YOLO: Visão Computacional para Engenharia e Agricultura de Precisão

Aplicações e Desempenho

Leonardo de Andrade Santos

*Universidade Federal do Paraná
Curitiba*

3 de julho de 2025

Sumário

1 Introdução

2 Metodologia

3 Resultados

4 Discussão

5 Conclusão

Introdução

Contexto da Visão Computacional

- Visão computacional otimiza processos em engenharia e agricultura.
- Foco em agricultura de precisão: sustentabilidade e eficiência de recursos.
- Monitoramento de infraestrutura com análise visual avançada.
- Desafios: qualidade de imagens, sobreposição de plantas em VANTs.

Objetivo do Estudo

- Apresentar a biblioteca Ultralytics YOLO como solução para:
 - Detecção e segmentação de objetos em tempo real.
 - Aplicações em agricultura (ex.: análise de plantações de soja).
 - Monitoramento de infraestrutura (ex.: chaves seccionadoras).
- Demonstrar desempenho e flexibilidade do YOLOv11.

Metodologia

O Algoritmo YOLO

- **You Only Look Once (YOLO):** abordagem unificada para detecção de objetos.
- Reformula detecção como problema de regressão via CNN.
- Processo:
 - 1 Divide imagem em grade $S \times S$.
 - 2 Cada célula prevê B caixas delimitadoras e probabilidades de classe.
 - 3 Pontuação de confiança:
$$\Pr(\text{Objeto}) \times \text{IoU}_{\text{truth}}^{\text{pred}}$$

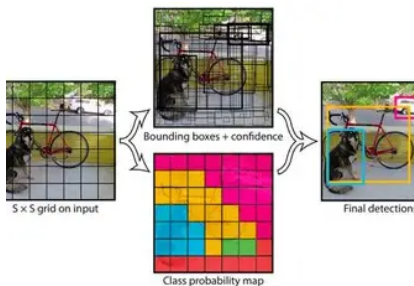


Figura: Exemplo de detecção de objetos com YOLO, mostrando caixas delimitadoras em uma imagem.

Funcionamento do YOLO

- **Parâmetros por caixa:** $(x, y, w, h, \text{confiança})$.
- **Probabilidades de classe:** $\Pr(\text{Class}_i | \text{Object})$.
- **Pontuação final:**

$$\Pr(\text{Class}_i | \text{Object}) \times \Pr(\text{Object}) \times \text{IoU}.$$

- **Vantagens:**
 - Velocidade: 45–155 fps.
 - Raciocínio global: analisa imagem completa.
 - Generalização: robusto em novos domínios.

Ultralytics YOLO

- Versão avançada do YOLO, com suporte a:
 - Detecção, segmentação, classificação, estimativa de pose, rastreamento.

- **Instalação:**

```
pip install ultralytics
```

- **Treinamento:** Suporta fine-tuning com datasets personalizados.

- **Exemplo de código:**

```
from ultralytics import YOLO
model = YOLO("yolov11n.pt")
model.train(data="dataset.yaml", epochs=10, imgsz=640)
```

Resultados

Experimento: Identificação de Veículos

- **Conjunto de dados:** Imagens urbanas com veículos.
- **Modelo:** YOLOv11 pré-treinado.
- **Métricas:**
 - Alta precisão média (mAP) em boas condições de iluminação.
 - Detecção robusta de caixas delimitadoras.
- **Resultado visual:** Caixas delimitadoras com pontuações de confiança.

Desempenho do YOLOv11

Métrica	Valor
Velocidade (fps)	45–155
Precisão Média (mAP)	Alta (boas condições)
Latência	< 25 ms

Tabela: Desempenho do YOLOv11 em experimentos.

Discussão

Vantagens e Limitações

■ Vantagens:

- Alta velocidade e precisão em tempo real.
- Flexibilidade para múltiplas tarefas de visão computacional.
- Aplicações em agricultura e monitoramento de infraestrutura.

■ Limitações:

- Restrições espaciais: detecção limitada de objetos pequenos/próximos.
- Generalização: falhas em proporções incomuns.
- Função de perda: impacto em caixas pequenas.

Conclusão

Conclusões

- Ultralytics YOLO: solução robusta para visão computacional.
- Aplicações práticas em agricultura de precisão e monitoramento.
- YOLOv11: alta precisão e velocidade, ideal para tempo real.
- Melhorias em YOLOv8+: detecção sem âncoras, funções de perda avançadas.
- Contribuição: sustentabilidade e eficiência em engenharia e agricultura.