

# Análise Modelos YOLOv11 Aplicados à Detecção de Objetos

Beatriz Almeida Felício

## 1. Introdução

Esse relatório apresenta uma análise comparativa de três modelos de detecção de objetos treinados com o framework **YOLOv11**, utilizando a biblioteca **Ultralytics**. O objetivo principal foi avaliar o desempenho dos modelos com diferentes tamanhos de entrada de imagem (imgsz: 256, 512 e 640 pixels), mantendo os demais hiperparâmetros constantes. O foco está na capacidade analítica e de interpretação de métricas de desempenho, conforme proposto no desafio técnico.

O dataset fornecido foi anonimizado e dividido em 80% para treino e 20% para validação. Todos os modelos foram treinados por 50 épocas, devido a limitações de uso de GPU no Google Colab, onde todo o projeto foi executado e armazenado utilizando Google Drive.

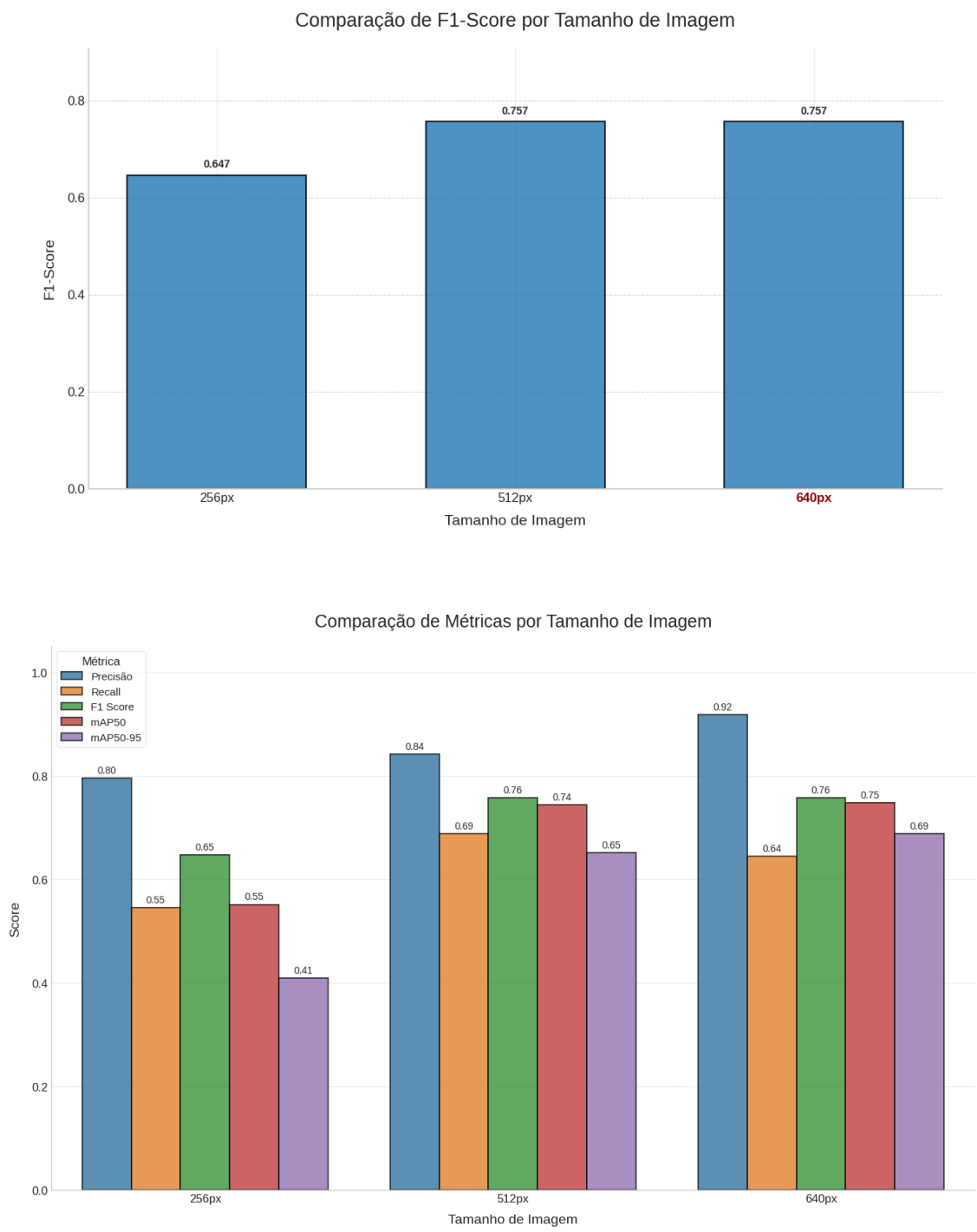
### Configurações utilizadas:

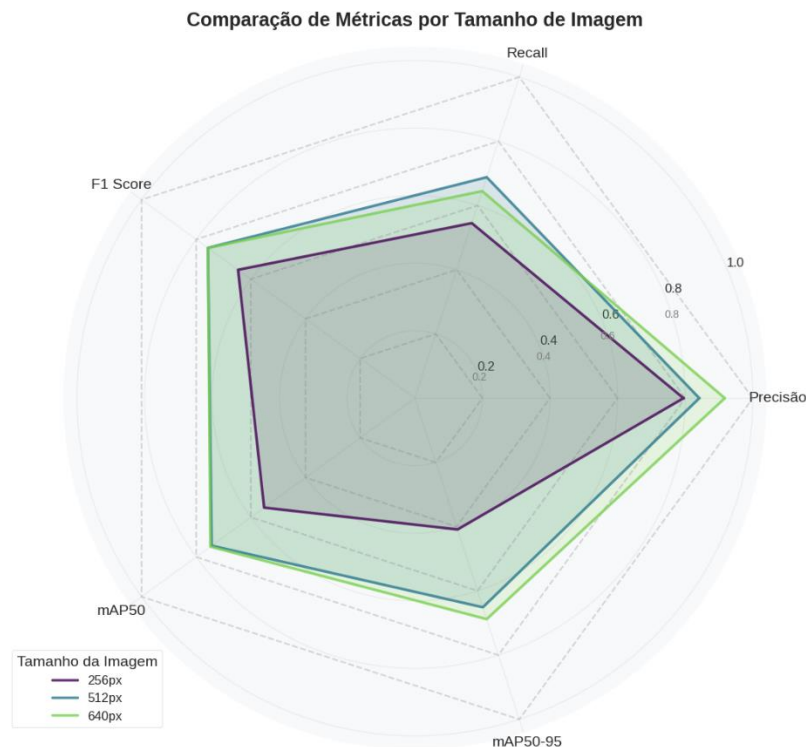
- **Framework:** Ultralytics YOLOv11
- **Épocas:** 50
- **Parâmetro variável:** imgsz (256, 512, 640)
- **Dataset:** Fornecido pelo desafio (divisão 80/20)
- **Demais hiperparâmetros:** constantes (batch size, otimizador, etc.)

## 2. Métricas Utilizadas e Explicação

- **F1-score:** Média harmônica entre precisão e revocação. Útil para avaliar o equilíbrio entre falsos positivos e falsos negativos.
- **Precisão (Precision):** Proporção de verdadeiros positivos (VP) entre todos os positivos preditos. Quanto maior a precisão, menos falsos positivos (FP).
- **Revocação (Recall):** Proporção de verdadeiros positivos (VP) entre todos os positivos reais. Quanto maior a revocação, menos falsos negativos (FN).
- **mAP@50:** Média das precisões para todas as classes considerando que uma detecção é correta se o IoU (Intersection over Union)  $\geq 0.5$ . É uma métrica de avaliação comum em tarefas de detecção de objetos, indicando o quão bem o modelo localiza corretamente os objetos, mesmo com um critério mais permissivo.
- **mAP@50-95:** Média das precisões calculadas em múltiplos níveis de IoU, de 0.50 a 0.95 em passos de 0.05. Essa métrica é mais rigorosa, pois avalia o desempenho do modelo com diferentes níveis de exigência quanto à sobreposição entre a predição e o objeto real.

### 3. Resultados Obtidos





➤ **Análise Geral:**

- Modelo A (256px) apresentou o desempenho mais fraco. A baixa resolução prejudicou a identificação de detalhes, especialmente em objetos menores, resultando em menor recall e mAP.
- Modelo B (512px) mostrou uma melhoria substancial em todas as métricas, com um bom equilíbrio entre custo computacional e desempenho. Obteve recall superior ao modelo de 640px e desempenho consistente em mAP.
- Modelo C (640px) alcançou a melhor mAP@50-95, o que indica melhor capacidade de generalização em diferentes níveis de exigência de sobreposição (IoU). A alta precisão evidencia uma redução nos falsos positivos, embora tenha perdido ligeiramente em recall em comparação ao modelo de 512px.

#### 4. Considerações e Limitações

- **Limitações de Infraestrutura:** O treinamento foi realizado integralmente no Google Colab, o que impôs restrições de tempo de uso e poder computacional. Por isso, os modelos foram treinados por apenas 50 épocas.
- **Adaptação de Ambiente:** Devido à indisponibilidade de GPU local, todo o projeto foi adaptado para uso com Google Drive, exigindo ajustes no gerenciamento de arquivos e paths.

## 5. Conclusões Finais

Com base nos experimentos realizados foi possível identificar os seguintes pontos:

### ➤ Impacto do Tamanho da Imagem

- Imagens maiores (640px): Melhor performance na detecção de objetos pequenos, devido à riqueza de detalhes.
- Imagens menores (256px): Processamento mais rápido, porém com perda significativa de precisão.
- 512px: Representa um bom compromisso entre desempenho e velocidade.

### ➤ Considerações de Generalização

- O modelo de 640px apresentou melhor generalização com maior mAP.
- O modelo de 512px é o mais indicado para aplicações em tempo real, com boa performance e menor exigência computacional.
- A escolha ideal do modelo depende diretamente do contexto da aplicação.

### ➤ Recomendações para Produção

- Para aplicações que exigem alta precisão e detecção detalhada: Modelo de 640px.
- Para cenários com foco em veículos grandes e menor carga computacional: Modelo de 512px.
- Para dispositivos com recursos extremamente limitados ou que priorizam rapidez: Modelo de 256px.

## 6. Melhorias Futuras

- Aumentar o número de épocas porque treinamentos com mais de 50 épocas podem elevar o desempenho e a generalização.
- Aplicar técnicas de data augmentation para compensar a qualidade limitada do dataset.
- Testar em novos conjuntos de dados para validar a robustez dos modelos em cenários mais variados.

## 7. Referências Bibliográficas

**KUKIL. Mean Average Precision (mAP) in object detection.** Disponível em: <<https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/>>.

**MERT. How to use YOLOv11 for Object Detection - Mert.** Disponível em: <<https://medium.com/@Mert.A/how-to-use-yolov11-for-object-detection-924aa18ac86f>>.

**NIKHILESWARA RAO, S. YOLOv11 architecture explained: Next-level object detection with enhanced speed and accuracy.** Disponível em: <<https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71>>.

**PÁDUA, M. Machine Learning -Métricas de avaliação: Acurácia, Precisão e Recall, F1-score.** Disponível em: <<https://medium.com/@mateuspdua/machine-learning-m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-e-recall-d44c72307959>>.

**SADANGI, S. How to deal with files in Google Colab: Everything you need to know.** Disponível em: <<https://neptune.ai/blog/google-colab-dealing-with-files>>.

**ULTRALYTICS. YOLO Métricas de desempenho.** Disponível em: <<https://docs.ultralytics.com/pt/guides/yolo-performance-metrics/>>.

**ULTRALYTICS. Ultralytics YOLO11.** Disponível em: <<https://docs.ultralytics.com/pt/models/yolo11/>>.