

24 de Novembro de 2025

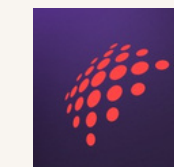
Quantization Strategies for Low-Resource Hardware in Deep Learning Applications

Orientador

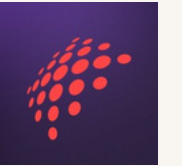
Rodrigo Mangoni Nicola

Alunos

Caio Martins de Abreu
Filipi Enzo Siqueira Kikuchi
Pablo Ruan Lana Viana

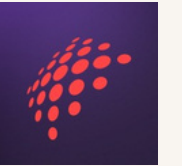


| | | |
|-----|-------------|----|
| I | Introdução | 3 |
| II | Objetivos | 4 |
| III | Hipótese | 5 |
| IV | Metodologia | 6 |
| V | Resultados | 11 |
| VI | Conclusão | 15 |



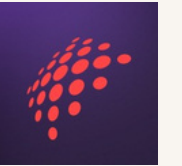
Introdução

As **CNNs** são essenciais na visão computacional, porém exigem muita **memória** e **processamento**. Em sistemas embarcados, essas redes em precisão total tornam-se difíceis de executar em tempo real.



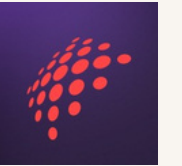
Objetivos

Este trabalho avalia técnicas de quantização pós-treinamento (**PTQ**) em **CNNs**, examinando impacto em **latência**, **CPU**, **memória** e desempenho no hardware embarcado, demonstrando a viabilidade e importância da otimização para IA eficiente.



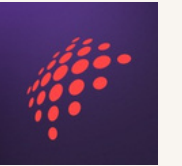
Hipótese

Modelos quantizados mantêm desempenho próximo ao **FP32**, enquanto reduzem drasticamente o custo computacional, tornando-se mais adequados para dispositivos embarcados.



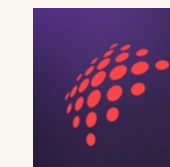
Quantização Estática

A quantização estática usa **calibração** para converter **pesos** e **ativações** em **INT8**, reduzindo memória e latência, preservando estabilidade e acurácia em hardware embarcado.



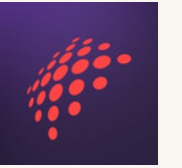
Quantização Dinâmica

A quantização dinâmica **converte pesos** para **INT8** em **tempo de execução**, ajustando ativações dinamicamente, agilizando inferência sem calibração e mantendo desempenho eficiente em dispositivos de baixo recurso.



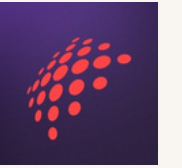
Modelo Selecionado

YOLOv8n foi escolhido por combinar alta acurácia, baixa latência e eficiência computacional. Sua arquitetura **anchor-free** e modular facilita o treino e a aplicação de **PTQ**, tornando-o ideal para hardware embarcado limitado.



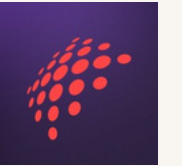
Conjunto de Dados

O conjunto de dados contem **3.737** imagens variadas, anotadas manualmente e padronizadas no **Roboflow** para 640×640 . Dividido em treino/validação/teste, inclui múltiplas pessoas por cena e resolução média de **0,92 MP**.



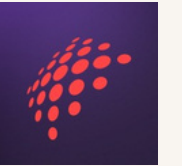
Hardware

O **Raspberry Pi 5** foi utilizado como plataforma de testes de baixo recurso para avaliar os efeitos da **PTQ** em modelos **YOLO**, medindo latência de inferência, uso de memória e acurácia, fornecendo um ambiente realista para implantação de IA embarcada.



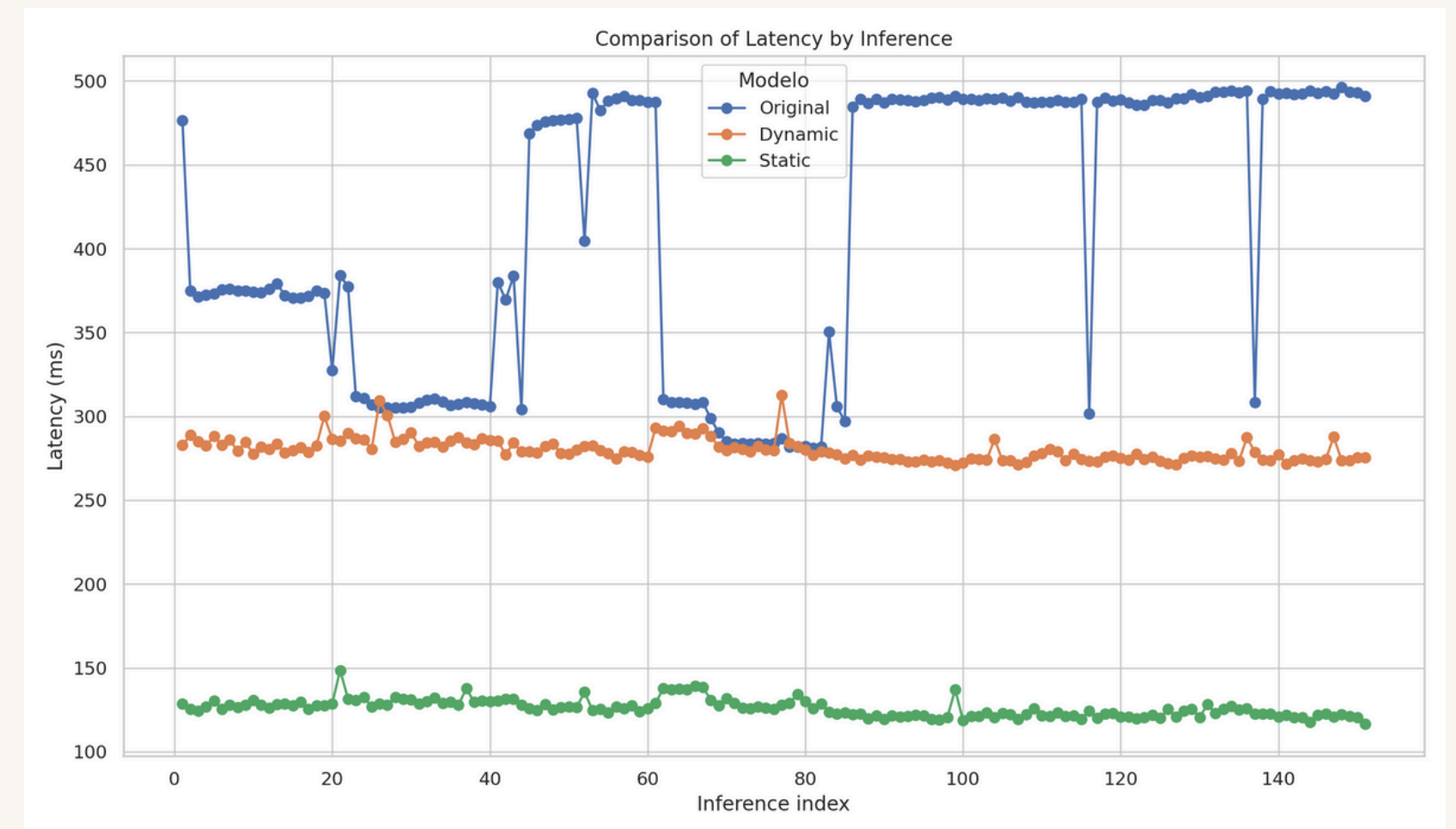
Resultados

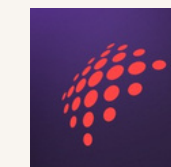
Quantização reduz **latência**, melhora **throughput** e diminui **memória**, com pequeno impacto na acurácia. A **quantização estática** oferece maior aceleração enquanto a **dinâmica** mantém melhor fidelidade, preservando distribuição de confiança e desempenho consistente



Análise de Latência

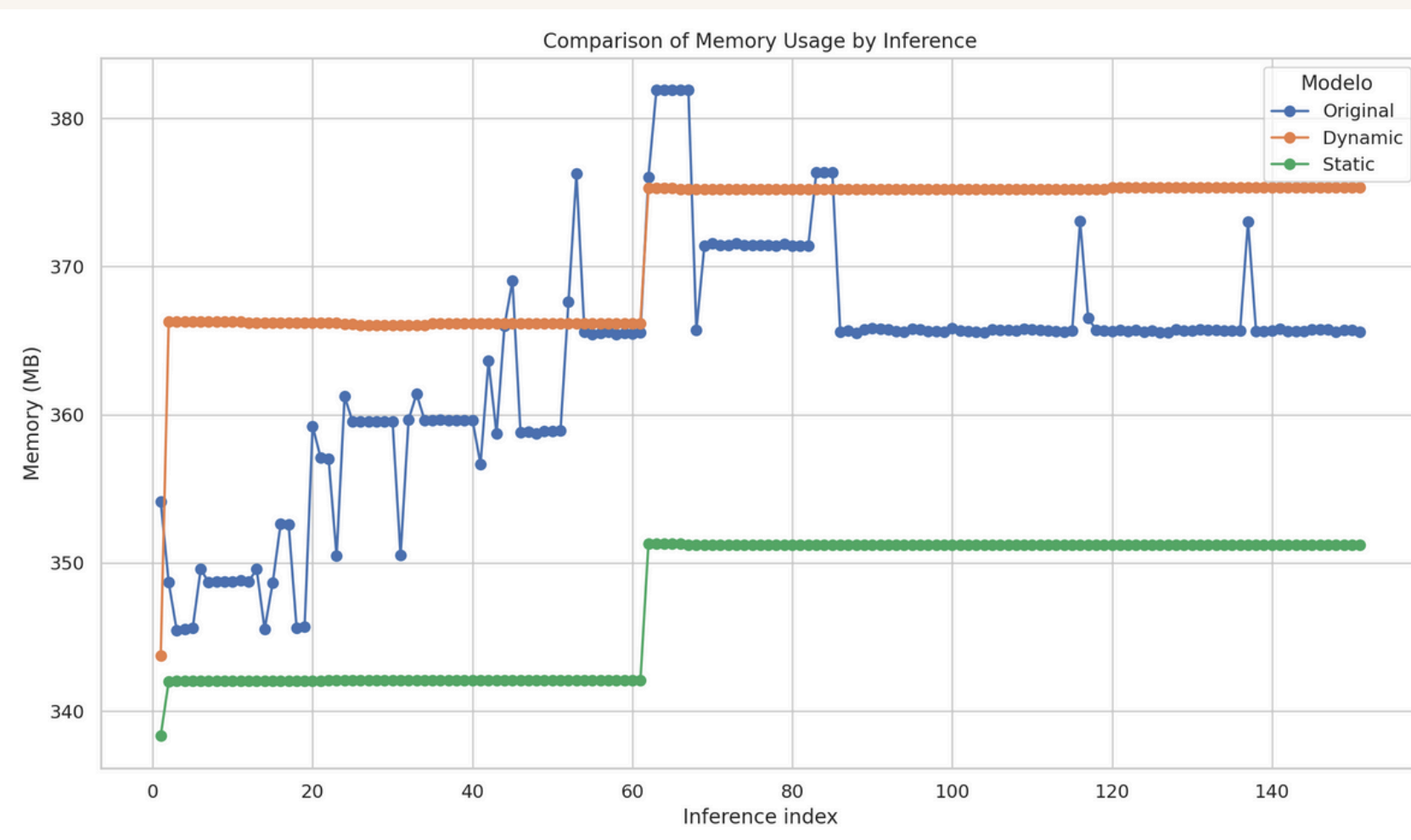
A quantização estática **reduziu** significativamente a **latência** e **aumentou** o **throughput**, embora com maior uso de **CPU**. A dinâmica ofereceu ganhos moderados e maior **adaptabilidade**, mantendo **eficiência** e consumo de **memória** estáveis.

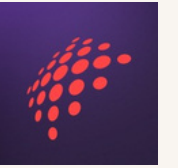




Análise de Memória

A quantização estática **reduziu** cerca de **9%** da memória e mostrou uso mais estável. A dinâmica trouxe leve redução, mas **maior flexibilidade**. Ambas melhoram a eficiência em sistemas embarcados.

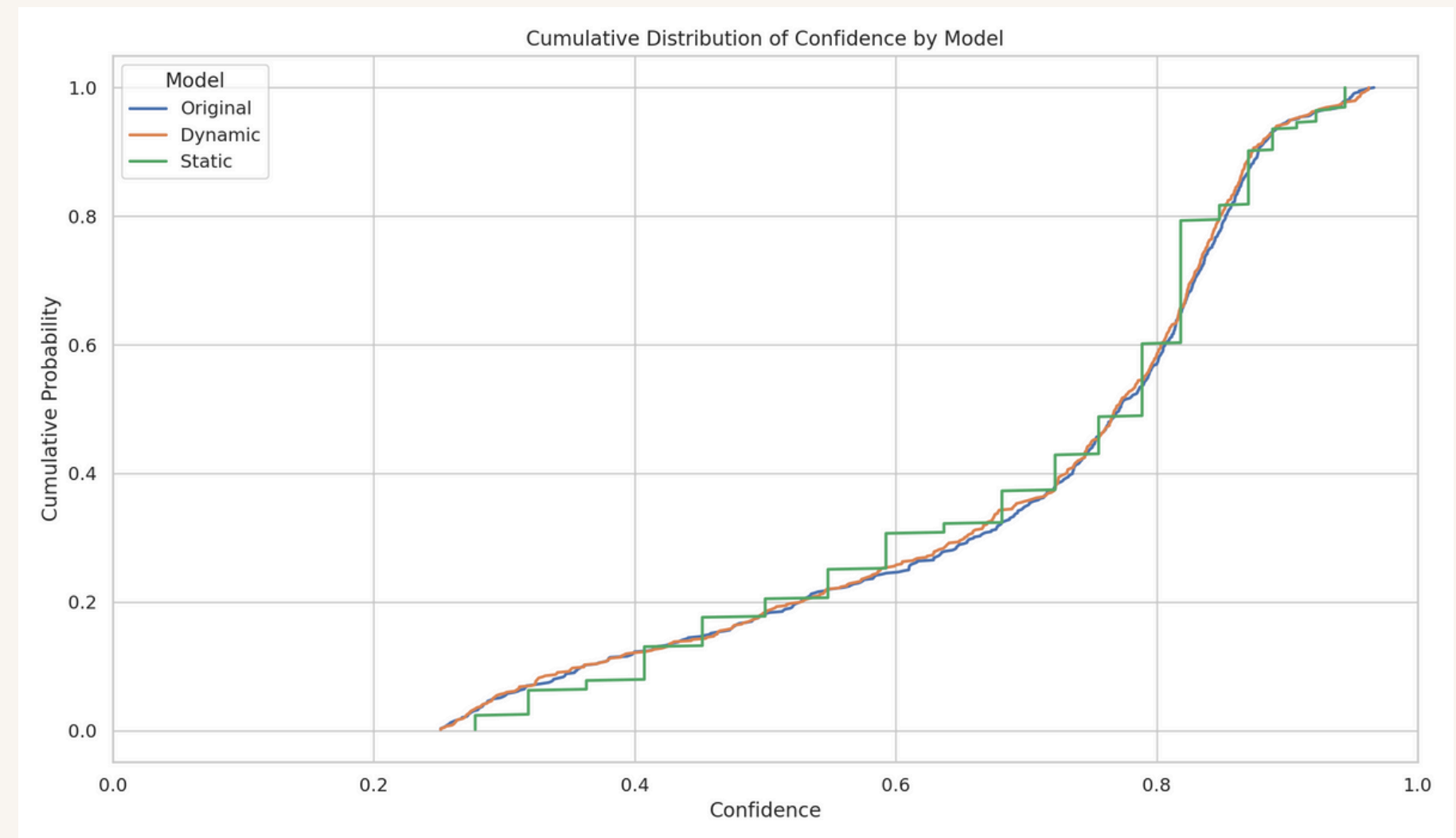


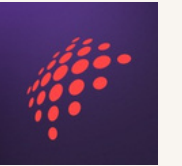


Análise de Acurácia

A análise mostra que a quantização **dinâmica** mantém o comportamento do modelo **original**, enquanto a estática reduz levemente as **confidências**.

Mesmo assim, ambas preservam a **confiabilidade** geral, equilibrando eficiência computacional e fidelidade preditiva





Conclusão

A quantização demonstra-se eficaz para implementar deep learning em hardware limitado. A quantização estática oferece maior desempenho e baixa latência, enquanto a dinâmica traz flexibilidade. Ambas equilibram **acurácia, eficiência e velocidade**, viabilizando visão computacional em dispositivos de borda.