Quantização de Modelos de Deep Learning para Detecção em Tempo Real de Pólipos Colorretais em Diferentes Plataformas de Hardware

Rian de Souza Santos[1], Gustavo Novack Viana Lima[2], Carlos Eduardo Gonçalves de Oliveira[3], Davi de Jesus Teixeira[3], Ricardo Augusto Pereira Franco[3]

[1]Centro de Excelência em Inteligência Artificial (CEIA)
Universidade Federal de Goiás (UFG)
Goiânia, Goiás, Brasil

[2]Escola de Engenharia Elétrica, Mecânica e de Computação (EMC)
Universidade Federal de Goiás (UFG)
Goiânia, Goiás, Brasil

[3]Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia, Goiás, Brasil

rian.souza@discente.ufg.br, novack@discente.ufg.br,

carlosedgonc@gmail.com, dvjteixeira@gmail.com,

ricardofranco@ufg.br

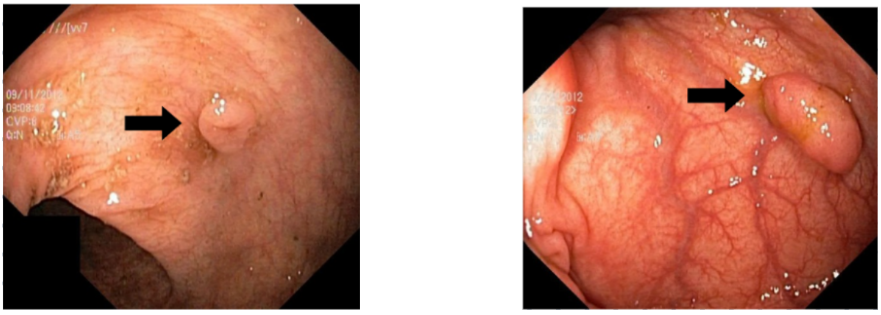
Abstract. *This study evaluates real-time colorectal polyp detection using YOLOv8l, YOLOv9c, and YOLOv11l across three GPUs (RTX 4090, 3070, 2060) and formats (FP32, FP16, INT8). Trained on 1,808 images and optimized via TensorRT, results indicate FP16 enables real-time inference ($\geq$60 FPS) on all hardware with negligible diagnostic loss, whereas INT8 slightly degrades accuracy. YOLOv8l FP16 was the optimal configuration, achieving 0.948 recall, 0.939 mAP@0.5, and 91.5–329.3 FPS. Delivering speedups up to 3.10$\times$ on the RTX 2060, FP16 proves highly viable for resource-constrained clinical environments.*

Resumo. *Este estudo avalia a detecção de pólipos em tempo real usando YOLOv8l, YOLOv9c e YOLOv11l em três GPUs (RTX 4090, 3070, 2060) e formatos (FP32, FP16, INT8). Treinados com 1.808 imagens e otimizados via TensorRT, os resultados mostram que a quantização FP16 viabiliza inferência em tempo real ($\geq$60 FPS) em todos os hardwares com impacto diagnóstico irrelevante, enquanto INT8 degrada levemente a acurácia. O YOLOv8l FP16 obteve o melhor balanço (recall 0,948, mAP@0.5 0,939 e 91,5–329,3 FPS). Com speedups de até 3,10$\times$ na RTX 2060, o FP16 prova-se altamente viável para clínicas com restrição de hardware.*

1. Introdução

O câncer colorretal é uma das principais causas de mortalidade por câncer em todo o mundo [Bray et al. 2024, Corley et al. 2014]. A Figura 1 ilustra exemplos de pólipos

identificados durante o exame de colonoscopia, cuja detecção em estágios iniciais é determinante para um prognóstico favorável. É amplamente reconhecido que a detecção e remoção precoce de pólipos durante o exame de colonoscopia reduzem significativamente tanto a incidência quanto a mortalidade associadas à doença [Bray et al. 2024, Corley et al. 2014]. Apesar de ser o exame padrão-ouro para rastreamento, a colonoscopia apresenta limitações inerentes, tais como variabilidade interobservador, fadiga do examinador e a natureza altamente dinâmica do procedimento, fatores que contribuem para a perda de lesões, especialmente pólipos pequenos ou planos [Rex et al. 2024].



(a) (b)

Figura 1. Exemplos de pólipos colorretais identificados durante exame de colonoscopia; as setas indicam a localização das lesões. Fonte: HyperKvasir [Borgli et al. 2020].

Nos últimos anos, sistemas baseados em Inteligência Artificial (IA), em especial aqueles fundamentados em redes neurais profundas aplicadas à visão computacional, têm demonstrado ganhos significativos na taxa de detecção de adenomas (*Adenoma Detection Rate* - ADR), atuando como ferramentas de suporte ao endoscopista durante o exame [Wang et al. 2019, Makar et al. 2025]. Entre essas abordagens, a detecção de pólipos *frame a frame* em fluxos contínuos de vídeo destaca-se por possibilitar a sinalização visual imediata de lesões suspeitas, desde que a latência do sistema seja compatível com o ritmo operacional do procedimento clínico [Jha et al. 2021].

Nesse contexto, a noção de *tempo real* torna-se central. Tradicionalmente, aplicações médicas de detecção de pólipos em colonoscopia consideram taxas de processamento na faixa de 25 a 30 quadros por segundo (FPS) como suficientes para caracterizar operação em tempo real, uma vez que essa faixa corresponde à taxa de captura padrão de muitos equipamentos clínicos [Rufo et al. 2023]. De fato, diversos trabalhos na área de detecção de objetos adotam o patamar de 30 FPS como limiar mínimo para definir um detector como operando em tempo real [Wang et al. 2023]. No entanto, avanços recentes em hardwares, como as Unidades de Processamento Gráfico (*Graphic Processing Units* - GPU), e em técnicas de otimização de inferência, têm motivado uma reavaliação desses limites, especialmente em sistemas de visão computacional interativos e de alta responsividade.

Do ponto de vista perceptual, a taxa de atualização padrão da grande maioria dos monitores e dispositivos de exibição contemporâneos é de 60 Hz, o que estabelece um limite natural para a fluidez visual percebida pelo operador humano [Kiani Galoogahi et al. 2017]. Taxas de processamento inferiores a esse patamar podem

introduzir aliasing temporal e aumento da latência percebida, fatores particularmente relevantes em sistemas interativos nos quais o ciclo de percepção–ação do examinador depende de retroalimentação visual contínua.

Nesse contexto, modelos da família YOLO (*You Only Look Once*) destacam-se pelo equilíbrio entre acurácia e velocidade de inferência [Redmon et al. 2016], enquanto técnicas de quantização numérica, como FP16 e INT8, oferecem caminhos promissores para a redução da latência computacional [Jacob et al. 2018, Gholami et al. 2021]. Contudo, a viabilidade desses modelos em cenários clínicos reais depende fortemente da combinação entre arquitetura, hardware disponível e nível de otimização empregado.

Diante desse cenário, o objetivo deste trabalho é testar e avaliar a viabilidade da realização da tarefa de detecção de pólipos em tempo real em GPUs distintas por meio de técnicas de quantização. Para isso, serão analisadas as variantes *large* dos modelos YOLOv8, YOLOv9 e YOLOv11 com o intuito de avaliar o impacto da quantização em modelos que possuam grande quantidade de parâmetros (ex.: variante *large*). Os modelos serão avaliados em plataformas de hardware distintas e sob duas formas de quantização (FP16 e INT8). Portanto, este trabalho visa avaliar a execução de modelos em tempo real para identificar quais combinações (modelo–hardware–quantização) obtém as melhores métricas de desempenho e atingem ou superam o limiar de 60 FPS, caracterizando a operação de detecção de pólipos em tempo real.

Nesse sentido, este trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados à detecção de pólipos colorretais com modelos de visão computacional; na Seção 3 é descrita a metodologia experimental proposta; os resultados da comparação entre as variantes dos modelos YOLO com e sem quantização em hardwares distintos são apresentados e discutidos na Seção 4; por fim, a Seção 5 apresenta as conclusões obtidas.

2. Trabalhos Relacionados

A detecção automática de pólipos colorretais tem sido impulsionada pela contínua evolução das arquiteturas de redes neurais profundas. Estudos pioneiros como o de Jha et al. [Jha et al. 2021] destacaram o potencial de modelos baseados em aprendizado profundo para a detecção em tempo real durante colonoscopias. Mais recentemente, Yang et al. [Yang et al. 2025] introduziram o YOLO-OB, modelo *anchor-free* que alcançou *recall* de 98,23% no dataset SUN, porém com velocidade de apenas 39 FPS em uma GPU RTX 3090.

Para além da performance diagnóstica, a adequação desses modelos à prática clínica requer inferência em tempo real utilizando hardware com recursos limitados. Nesse contexto, a quantização de modelos neurais tem se consolidado como estratégia fundamental para a redução da latência [Jacob et al. 2018, Gholami et al. 2021]. Em [Carrinho and Falcao 2023] foi demonstrado que o uso do framework NVIDIA TensorRT para quantizar o YOLOv4 é capaz de aumentar de forma relevante a velocidade de inferência sem incorrer em decréscimo significativo na acurácia (mAP@0.5). Contudo, os estudos nesta área avalia os modelos em uma única GPU, sem explorar sistematicamente a interação entre arquitetura, nível de quantização e plataforma de hardware.

Portanto, este trabalho se diferencia dos demais ao investigar a viabilidade das variantes *large* dos modelos YOLOv8, YOLOv9 e YOLOv11 sob o rigoroso limiar de

60 FPS, focando na otimização via quantização FP16 e INT8. Além disso, a metodologia utilizada pode ser facilmente replicada para as demais variantes YOLO assim como para outros modelos neurais. Os resultados obtidos fornecem diretrizes atualizadas para a implementação de IA em exames de colonoscopia em tempo real, considerando desde GPUs de alto desempenho até hardware de geração anterior com restrição computacional.

3. Metodologia

A metodologia proposta é estruturada de forma a permitir a análise reprodutível do impacto da arquitetura do modelo, da quantização e da plataforma computacional sobre o desempenho de inferência e a eficácia diagnóstica. O pipeline da metodologia proposta pode ser visualizado na Figura 2.

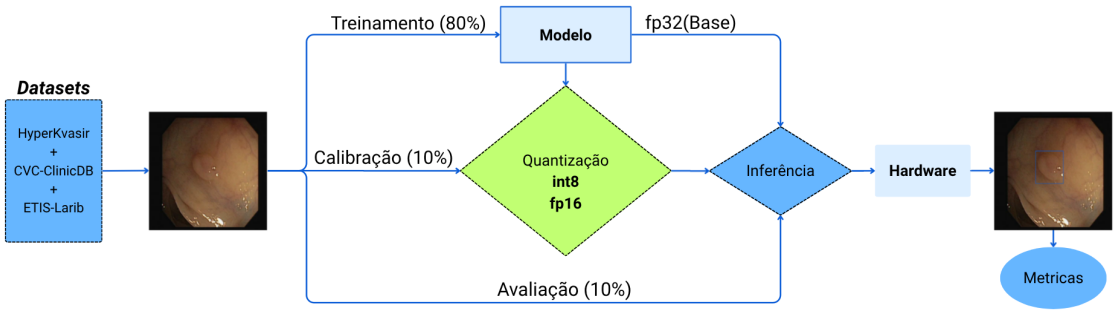


Figura 2. Pipeline experimental proposto.

O fluxo experimental adotado neste trabalho compreende as seguintes etapas principais: (i) seleção e preparação do conjunto de dados; (ii) treinamento dos modelos de detecção; (iii) quantização dos modelos treinados; (iv) inferência em múltiplos hardwares; e (v) análise comparativa dos resultados obtidos. A seguir, serão descritas as etapas do pipeline.

3.1. Conjunto de Dados

O conjunto de dados (*dataset*) foi construido a partir da combinação de outros três datasets amplamente utilizados na literatura: HyperKvasir[Borgli et al. 2020], CVC-ClinicDB [Bernal et al. 2015] e ETIS-Larib Polyp DB [Huang et al. 2024]. Tais bases de dados possuem origens geográficas distintas — Noruega (HyperKvasir), Espanha (CVC-ClinicDB) e França (ETIS-Larib), o que contribui com a generalização do modelo; contudo, visando a aplicabilidade do modelo no Brasil, se torna importante o desenvolvimento de um *dataset* também contemplando a população brasileira

Após a junção dos *datasets*, obteve-se um conjunto final com um total de 1.808 imagens, particionado aleatoriamente em 80% para treinamento, 10% para validação e 10% para teste. Após a junção dos *datasets*, obteve-se um conjunto final com um total de 1.808 imagens, particionado aleatoriamente em 80% para treinamento, 10% para validação e 10% para teste. .

3.2. Modelos de Detecção de Objetos

Dentre os modelos selecionados para análise, optou-se pela variante *large* dos modelos YOLOv8, YOLOv9 e YOLOv11 [Jocher et al. 2023]. A escolha pela variante *large* foi

motivada pelo interesse em analisar arquiteturas de maior capacidade representacional, tipicamente associadas a um maior custo computacional, e investigar sua viabilidade em cenários de tempo real quando combinadas com técnicas de quantização.

Todos os modelos foram inicializados a partir de pesos pré-treinados no *dataset* COCO e, posteriormente, submetidos a um processo de *fine-tuning* com o conjunto de dados de treinamento, de modo a especializá-los para a tarefa de detecção de pólipos colorretais.

3.3. Ambiente Experimental e Treinamento

O treinamento dos modelos foi realizado em uma estação de trabalho equipada com uma GPU NVIDIA RTX 4090, um processador Intel Core i9-13900K e 64 GB de memória RAM.

Durante o treinamento, todas as imagens de entrada foram redimensionadas para 640×640 pixels. O processo foi conduzido por 100 épocas, utilizando *batch size* igual a 16. Foi utilizado o otimizador Adam, com *learning rate* inicial de 1×10^{-3}, aliado a um *cosine scheduler* para o decaimento da taxa de aprendizado ao longo das épocas.

3.4. Quantização e Otimização dos Modelos

Após o treinamento, os modelos foram avaliados em três formatos numéricos distintos: FP32 (modelo base), FP16 e INT8. Optou-se por FP16 e INT8 por serem os formatos de baixa precisão acelerados nativamente nas três GPUs; o FP8 foi desconsiderado por exigir *Tensor Cores* de quarta geração (*compute capability* 8.9), presentes apenas GPUs mais recentes (ex.: RTX 4090) e ausentes nas GPUs com arquitetura Ampere e Turing, que estavam disponíveis para uso neste trabalho.

O processo de quantização deve ser executado no hardwarealvo e este trabalho utilizou o *framework* NVIDIA TensorRT [NVIDIA 2024], que possibilita a fusão de camadas e a otimização de *kernels* considerando o hardware alvo. Para a quantização INT8, foi utilizado um subconjunto representativo do conjunto de validação como conjunto de dados de calibração, visando minimizar impactos na acurácia do modelo.

Essa diversidade de plataformas possibilitou avaliar a escalabilidade dos modelos e o impacto da quantização em cenários realistas de implantação em tempo real.

3.5. Métricas de Avaliação

A qualidade das detecções foi avaliada com base na métrica de *Intersection over Union* (IoU), que mede o grau de sobreposição espacial entre a caixa delimitadora inferida pelo modelo (B_{inf}) e a anotação de referência (B_{ref}), sendo definida pela Equação 1:

$$IoU = \frac{\text{Área}(B_{inf} \cap B_{ref})}{\text{Área}(B_{inf} \cup B_{ref})} \tag{1}$$

Considerou-se o limiar de $IoU \geq 0.5$ para categorizar cada predição como Verdadeiro Positivo (TP), sendo as demais detecções classificadas como Falso Positivo (FP) quando não atingem esse limiar, e as lesões anotadas sem correspondência válida contabilizadas como Falso Negativo (FN).

Com base nessa categorização, duas métricas complementares foram empregadas para caracterizar o comportamento do detector. A precisão (Equação 2) expressa a proporção de detecções corretas dentre todas as predições emitidas, enquanto o *recall* (Equação 3) quantifica a fração de lesões efetivamente identificadas em relação ao total presente nas imagens — aspecto particularmente relevante no contexto clínico, dada a necessidade de minimizar a ocorrência de pólipos não detectados durante o exame.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{2}$$

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{3}$$

A avaliação consolidada do desempenho dos modelos foi realizada por meio do *mean Average Precision* a um limiar de 0.5 (mAP@0.5), métrica padrão na literatura de detecção de objetos que corresponde à área sob a curva precisão-*recall*.

O desempenho computacional foi mensurado pela taxa de quadros por segundo (FPS), definida como a razão entre o número total de imagens processadas e o tempo total de inferência.

4. Resultados e Discussão

Os resultados obtidos, apresentados na Tabela 1 e na Figura 3, revelam um impacto expressivo da quantização sobre o desempenho computacional dos modelos avaliados, com comportamento fortemente dependente da plataforma de hardware utilizada.

As medições de inferência foram realizadas em três plataformas distintas: NVIDIA RTX 4090 (*desktop*), representando hardware de alto desempenho; NVIDIA RTX 3070 (notebook), representando hardware intermediário; e NVIDIA RTX 2060 (*desktop*), representando hardware de geração anterior e maior restrição computacional.

De maneira geral, conforme apresentado na Tabela 1, a quantização FP16 viabiliza a operação em tempo real ($\geq$ 60 FPS) em todas as configurações avaliadas, com impacto mínimo ou nulo sobre as métricas diagnósticas. Já a quantização INT8 proporciona ganhos adicionais de velocidade, contudo, há degradação significativa na acurácia em diversas configurações analisadas.

Na estação de trabalho que contém a GPU RTX 2060, nenhum modelo atingiu o requisito de 60 FPS considerando o cenário em FP32. No entanto, os resultados com FP16 mostram observações relevantes, isto é, o YOLOv11l alcança 126,3 FPS, o YOLOv9c 115,2 FPS e o YOLOv8l 106,5 FPS. A quantização FP16 preserva integralmente as métricas diagnósticas do YOLOv8l (mAP@0.5 = 0,939), do YOLOv9c (mAP@0.5 = 0,900) e do YOLOv11l (mAP@0.5 = 0,886), sem qualquer degradação mensurável em relação ao FP32. A quantização INT8 eleva a taxa de FPS, porém apresenta degradação diagnóstica: o *recall* do YOLOv8l reduz para 0,806 e o do YOLOv9c para 0,864.

Já o cenário de análise com a GPU RTX 3070 (notebook), nenhum dos três modelos atinge o limiar de 60 FPS na configuração base FP32: o YOLOv8l opera a 50,7 FPS, o YOLOv11l a 47,2 FPS e o YOLOv9c a apenas 40,6 FPS. A quantização FP16 é suficiente para viabilizar a operação em tempo real dos três modelos, elevando-os para 91,5 FPS

Tabela 1. Desempenho de inferência dos modelos em diferentes GPUs e níveis de quantização para detecção de pólipos colorretais.

Hardware	Model	Quantization	FPS	precision	recall	mAP@0.5
RTX 2060	YOLOv8l	Base (FP32)	42.0	0.924	**0.948**	**0.939**
		FP16	106.5	0.924	**0.948**	**0.939**
		INT8	144.2	**0.945**	0.806	0.797
	YOLOv9c†	Base (FP32)	37.2	0.888	0.911	0.900
		FP16	115.2	0.888	0.911	0.900
		INT8	146.5	0.892	0.864	0.849
	YOLOv11l	Base (FP32)	51.2	0.895	0.895	0.886
		FP16	126.3	0.895	0.895	0.886
		INT8	**132.1**	0.892	0.864	0.883
RTX 3070	YOLOv8l	Base (FP32)	50.7	0.924	**0.948**	**0.939**
		FP16	**91.5**	0.918	0.937	0.929
		INT8	**91.5**	0.946	0.822	0.812
	YOLOv9c†	Base (FP32)	40.6	0.888	0.911	0.900
		FP16	75.6	0.869	0.906	0.894
		INT8	81.9	**0.948**	0.759	0.751
	YOLOv11l	Base (FP32)	47.2	0.895	0.895	0.886
		FP16	75.9	0.905	0.901	0.892
		INT8	75.9	0.928	0.812	0.797
RTX 4090	YOLOv8l	Base (FP32)	170.1	0.924	**0.948**	**0.939**
		FP16	329.3	0.924	**0.948**	**0.939**
		INT8	**375.5**	**0.947**	0.838	0.826
	YOLOv9c†	Base (FP32)	166.2	0.888	0.911	0.900
		FP16	338.9	0.888	0.911	0.900
		INT8	343.2	0.896	0.864	0.849
	YOLOv11l	Base (FP32)	187.2	0.895	0.895	0.886
		FP16	336.1	0.895	0.895	0.886
		INT8	373.8	0.882	0.867	0.894

† O YOLOv9 não possui variante *large* oficial; utilizou-se a variante *compact* (YOLOv9c) como representante de capacidade equivalente desta arquitetura.

(YOLOv8l), 75,9 FPS (YOLOv11l) e 75,6 FPS (YOLOv9c). Em termos de acurácia, a maior redução absoluta de mAP@0.5 com FP16 foi de 0,010 no YOLOv8l (de 0,939 para 0,929). Pode-se observar que o YOLOv11l FP16 apresenta leve melhora no mAP@0.5 (de 0,886 para 0,892), comportamento que pode ser atribuído ao processo de otimização de *kernels* realizado pelo TensorRT que, ao reestruturar operações para o hardware específico, pode introduzir efeitos regularizadores que beneficiam a generalização do modelo [NVIDIA 2024]. Nota-se que a quantização INT8 nesta plataforma compromete severamente a acurácia.

No hardware contendo a GPU RTX 4090, todos os modelos superam amplamente o limiar de 60 FPS na configuração base FP32, atingindo 170,1 FPS (YOLOv8l), 166,2 FPS (YOLOv9c) e 187,2 FPS (YOLOv11l). A quantização em FP16 aproximadamente duplica esses valores, com os três modelos ultrapassando valores de 329 FPS,

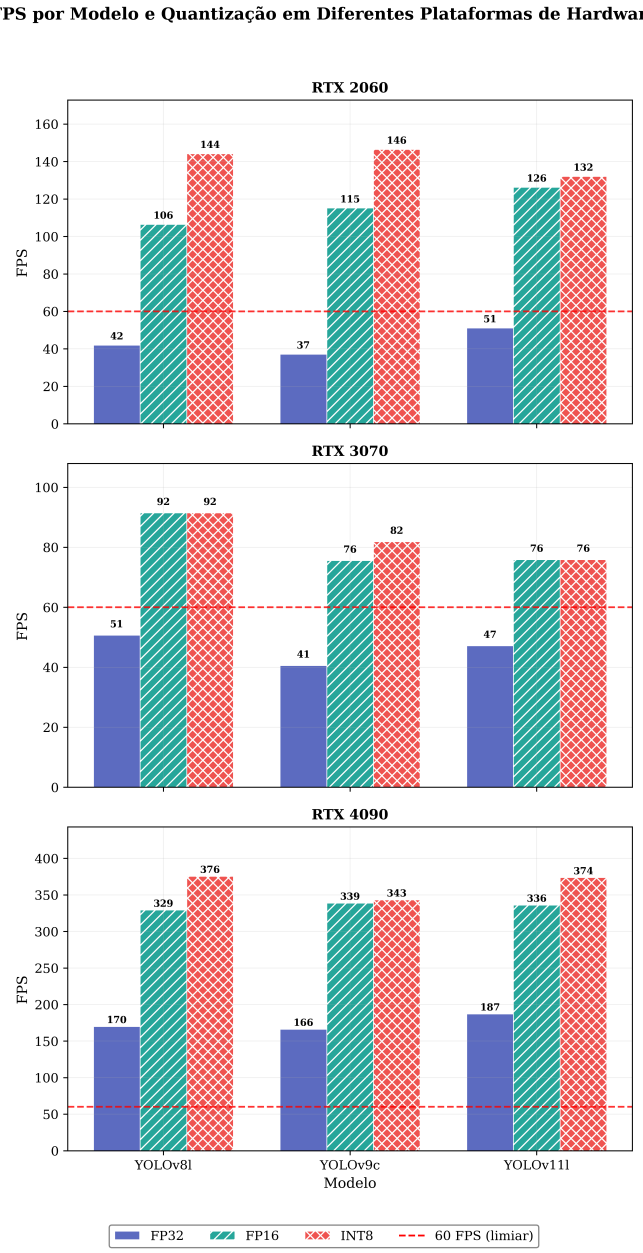


Figura 3. Taxas de inferência (FPS) por modelo e nível de quantização em cada GPU.

enquanto as métricas diagnósticas permanecem idênticas às obtidas em FP32 em vários casos. A quantização INT8, em geral, proporciona ganhos adicionais de velocidade, contudo, ocorre redução expressiva nas métricas obtidas.

A Figura 4 apresenta o *speedup* relativo proporcionado pela quantização enter a configuração base (FP32) e as configurações quantizadas (FP16 e INT8). Observa-se um padrão consistente, ou seja, o ganho relativo é inversamente proporcional à capacidade computacional da plataforma. Na RTX 2060, os maiores ganhos observados de *speedup* foram: FP16 de 2,47 (YOLOv11l) a 3,10 (YOLOv9c) e *speedup* INT8 de até 3,94×. Na RTX 3070, nota-se valores são mais inferiores que a RTX 2060 (*speedup* de 1,61 a 1,86), possivelmente limitados pela largura de banda de memória e pelo *thermal throttling* de

computadores móveis. Na RTX 4090, os *speedups* FP16 situam-se entre 1,79 e 2,04. Esses resultados indicam que a quantização é particularmente benéfica em computadores pessoais de uso fixo com maior restrição computacional, na qual as otimizações do TensorRT exploram de forma mais eficiente os ganhos de *throughput* proporcionados pela redução da precisão numérica [Gholami et al. 2021, Jacob et al. 2018].

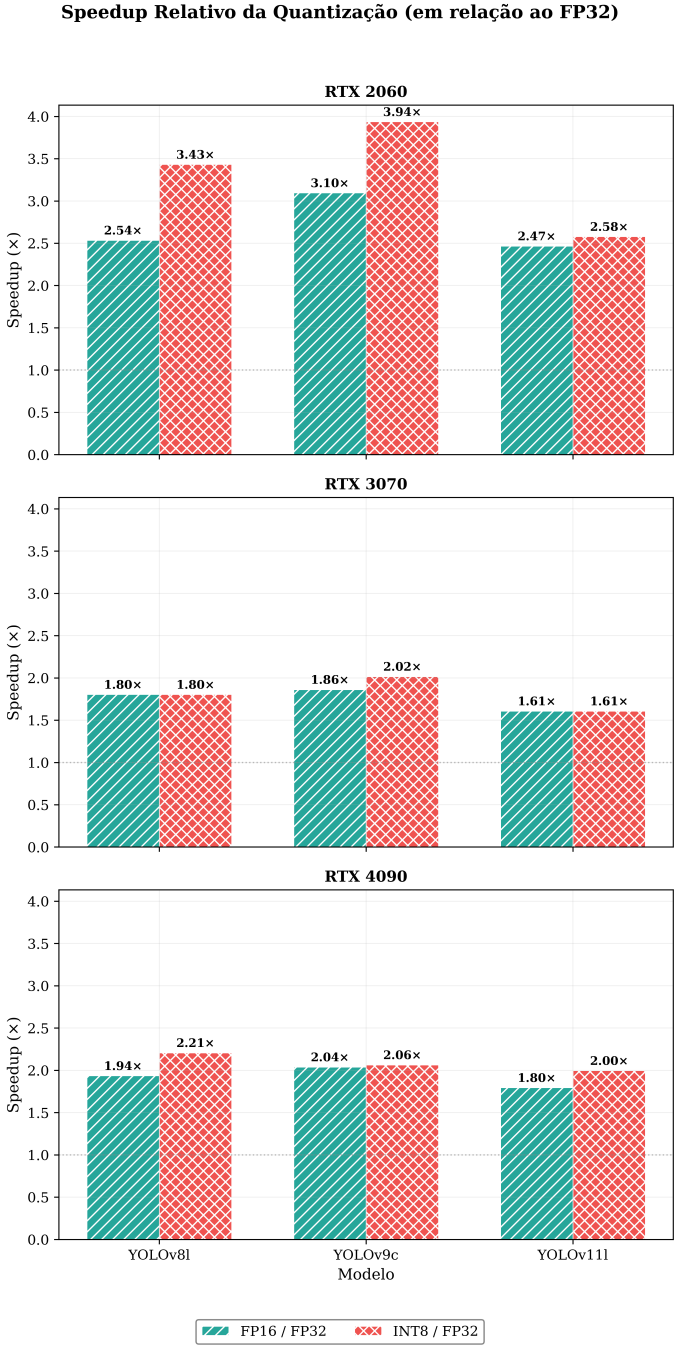


Figura 4. *Speedup* relativo da quantização FP16 e INT8 em relação ao FP32 para cada modelo e GPU.

A análise comparativa entre os modelos revela perfis de desempenho distintos. O YOLOv8l apresenta a maior acurácia diagnóstica, com mAP@0.5 de 0,939 e *re-*

call de 0,948 na configuração FP32 — os maiores valores absolutos de todo o experimento, métricas que são integralmente preservadas com FP16 nas plataformas RTX 4090 e RTX 2060. Esse perfil posiciona o YOLOv8l como a escolha mais conservadora do ponto de vista da segurança diagnóstica. O YOLOv11l destaca-se pelo melhor desempenho computacional em FP32 (187,2 FPS na RTX 4090). Suas métricas em FP16 são iguais às métricas em FP32 (mAP@0.5 entre 0,886 e 0,892), embora inferiores às do YOLOv8l.

Do ponto de vista clínico, o *recall* merece atenção especial, pois está diretamente associado à taxa de detecção de adenomas (ADR). Falsos negativos — pólipos não detectados, podem resultar em lesões pré-cancerosas não identificadas, com potencial impacto na mortalidade por câncer colorretal [Corley et al. 2014]. Nesse sentido, o YOLOv8l FP16 emerge como a configuração mais adequada para implantação clínica, por combinar o maior *recall* (0,937 a 0,948) e mAP@0.5 (0,929 a 0,939) com velocidades entre 91,5 e 329,3 FPS. O YOLOv11l FP16 posiciona-se como alternativa atrativa, oferecendo maior eficiência computacional com acurácia competitiva. A escolha entre maximizar *recall* ou precisão deve ser guiada pelo contexto clínico específico e pelas preferências institucionais. A Figura 5 apresenta exemplos visuais de inferências do modelo YOLOv8 *large* na detecção de pólipos.

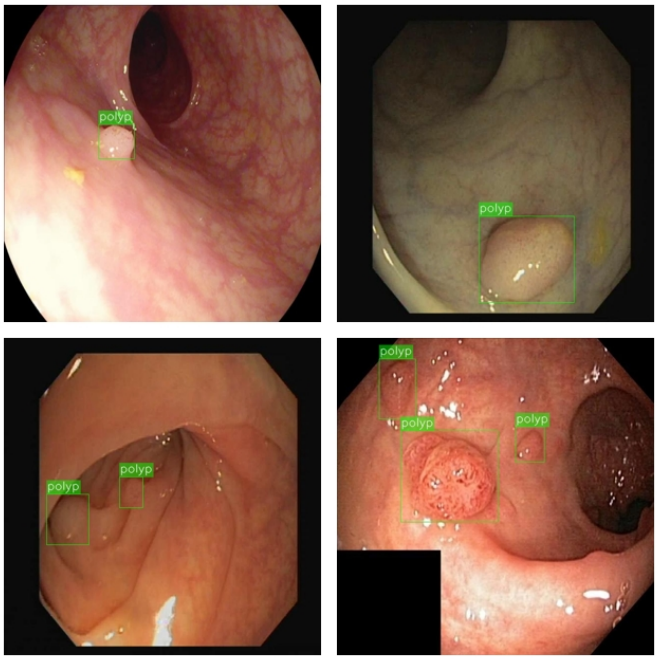


Figura 5. Exemplos de inferência do modelo YOLOv8l.

Por fim, destaca-se que a viabilidade em tempo real é demonstrada mesmo na RTX 2060, GPU de geração anterior e custo acessível em comparação às demais GPUs mais recentes. Os *speedups* elevados nesta plataforma (até 3,10 com FP16) reforçam que a quantização via TensorRT é uma estratégia relevante para democratizar o acesso a sistemas de IA em tempo real em ambientes clínicos com restrições de hardware.

5. Conclusões

Este trabalho propõe uma metodologia que avalia a viabilidade da detecção de pólipos colorretais em tempo real utilizando as variantes *large* dos modelos YOLOv8, YOLOv9 e YOLOv11, avaliadas em três plataformas de hardware (RTX 4090, RTX 3070 notebook e RTX 2060) e três formatos numéricos (FP32, FP16 e INT8), totalizando 27 configurações experimentais.

Os resultados demonstram que a quantização FP16 via TensorRT é suficiente para viabilizar a operação em tempo real ($\geq$ 60 FPS) em todas as combinações modelo–hardware avaliadas, com impacto mínimo ou nulo sobre as métricas diagnósticas. Em contraste, a quantização INT8, embora proporcione ganhos adicionais de velocidade, introduz degradação significativa na acurácia da maioria das configurações, comprometendo especialmente o *recall* — métrica diretamente associada à taxa de detecção de adenomas e, portanto, de maior relevância clínica.

Entre os modelos avaliados, o YOLOv8l FP16 destacou-se como a configuração mais adequada para implantação clínica, por reunir o maior *recall* (0,937 a 0,948) e mAP@0.5 (0,929 a 0,939) com velocidades de inferência variando entre 91,5 e 329,3 FPS nas plataformas testadas. O YOLOv11l FP16 posiciona-se como alternativa competitiva, oferecendo maior eficiência computacional com acurácia ligeiramente inferior.

Uma análise relevante refere-se à escalabilidade da quantização em plataformas de menor capacidade computacional. Na RTX 2060 a quantização FP16 proporcionou *speedups* de até 3,10, viabilizando taxas superiores a 106 FPS com preservação integral das métricas diagnósticas. Esse resultado reforça o potencial da quantização como estratégia para democratizar o acesso a sistemas de detecção assistida por IA em ambientes clínicos com restrições de hardware.

Como trabalhos futuros, pode-se citar a avaliação dos modelos em fluxos de vídeo contínuos de colonoscopia e a validação clínica da proposta em cenários reais.

Referências

Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., Gil, D., Rodríguez, C., and Vilariño, F. (2015). Cvc-clinicdb.

Borgli, H., Thambawita, V., Smedsrud, P. H., Hicks, S., Jha, D., Eskeland, S. L., Randel, K. R., Pogorelov, K., Lux, M., Nguyen, D. T. D., Johansen, D., Griwodz, C., Stensland, H. K., Garcia-Ceja, E., Schmidt, P. T., Hammer, H. L., Riegler, M. A., Halvorsen, P., and de Lange, T. (2020). HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Scientific Data*, 7(1):283.

Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., and Jemal, A. (2024). Global cancer statistics 2022: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 74(3):229–263.

Carrinho, P. and Falcao, G. (2023). Highly accurate and fast yolov4-based polyp detection. *Expert Systems with Applications*, 232:120834.

Corley, D. A., Jensen, C. D., Marks, A. R., Zhao, Y., Lee, J. K., Doubeni, C. A., Zauber, A. G., de Boer, J., Fireman, B. H., and Levin, T. R. (2014). Adenoma detection rate

and risk of colorectal cancer and death. *New England Journal of Medicine*, 370:1298–1306.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., and Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

Huang, C.-H., Wu, H.-Y., and Lin, Y.-L. (2024). Etis-larib polyp db.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713.

Jha, D., Ali, S., Tomar, N. K., Johansen, H. D., Johansen, D., Rittscher, J., Riegler, M. A., and Halvorsen, P. (2021). Real-time polyp detection, localization and segmentation in colonoscopy using deep learning. *IEEE Access*, 9:40496–40510.

Jocher, G., Qiu, J., and Chaurasia, A. (2023). Ultralytics yolo.

Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., and Lucey, S. (2017). Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1125–1134.

Makar, J., Abdelmalak, J., Con, D., Hafeez, B., and Garg, M. (2025). Use of artificial intelligence improves colonoscopy performance in adenoma detection: a systematic review and meta-analysis. *Gastrointestinal Endoscopy*, 101(1):68–81.e8.

NVIDIA (2024). *TensorRT Developer Guide, Version 10.0.1*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Rex, D. K., Anderson, J. C., Butterly, L. F., and et al. (2024). Quality indicators for colonoscopy. *Gastrointestinal Endoscopy*, 100(3):352–381.

Rufo, J., Fazal, M. I., García-Moreno, F. M., et al. (2023). A real-time polyp-detection system with clinical application in colonoscopy using deep convolutional neural networks. *Journal of Imaging*, 9(2):49.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475.

Wang, P., Berzin, T. M., Glissen Brown, J. R., and et al. (2019). Real-time automatic detection system increases colonoscopic polyp and adenoma detection rates: a randomized controlled study. *Gut*, 68(10):1813–1819.

Yang, X., Song, E., Ma, G., Zhu, Y., Yu, D., Ding, B., and Wang, X. (2025). Yolo-ob: An improved anchor-free real-time multiscale colon polyp detector in colonoscopy. *Biomedical Signal Processing and Control*, 99:107326.