ANÁLISE DE TÉCNICAS DE QUANTIZAÇÃO PARA MODELOS DE DETECÇÃO DE PRAGAS EM ARMADILHAS INTELIGENTES

ISADORA V. DIAS[1]; VICTOR R. S. SANTOS[2]; LUCAS FREITAS[3], PAULO R. FERREIRA JR.[3], LISANE BRISOLARA[3], JÚLIO C. B. MATTOS[3]

[1]Curso de Engenharia da Computação/UFPEL – ivdias@inf.ufpel.edu.br
[2]Curso de Engenharia da Computação/UFPEL – vrssantos@inf.ufpel.edu.br
[3]Centro de Desenvolvimento Tecnológico/UFPEL

1. INTRODUÇÃO

A implementação de sistemas inteligentes e autônomos em aplicações agrícolas, como o monitoramento de pragas, exige soluções com alta eficiência energética e baixa latência (PASSIAS et al., 2024). Dispositivos embarcados em armadilhas inteligentes precisam operar continuamente em ambientes remotos, muitas vezes sob restrições de conectividade e energia.

Embora as Redes Neurais Convolucionais (Convolutional Neural Networks – CNNs) tenham se mostrado eficazes na identificação de pragas como Ceratitis capitata e Grapholita molesta, modelos de alta precisão apresentam elevado custo computacional, o que limita sua aplicação prática (FREITAS et al., 2022).

Nesse contexto, entre os modelos pré-treinados disponíveis, a ResNet18 se destaca pelo bom desempenho de classificação, como demonstrado em FREITAS et al. (2022). O PyTorch é uma das bibliotecas Python mais utilizadas para treinamento e implementação de redes neurais, e, aliado à biblioteca Brevitas (PAPPALARDO, 2025), possibilita a aplicação de técnicas de quantização de forma eficiente.

A quantização, que consiste na redução da precisão numérica de pesos e ativações do modelo, é uma técnica essencial para otimizar a inferência em hardware com recursos restritos (UMUROGLU, 2018). Duas estratégias podem ser aplicadas: a post-training quantization (PTQ), na qual a quantização é realizada após o treinamento, e o quantization-aware training (QAT), que incorpora os efeitos da quantização durante o processo de treino, tornando o modelo mais robusto.

O objetivo deste trabalho foi avaliar e comparar o impacto das abordagens PTQ e QAT na acurácia do modelo ResNet18 (utilizando a biblioteca Brevitas), visando à sua futura portabilidade para plataformas embarcadas com Field-Programmable Gate Arrays (FPGAs). Para isso, foi empregado o FINN, um framework especializado em redes neurais quantizadas, com ênfase na geração de arquiteturas em estilo dataflow personalizadas para cada rede.

2. METODOLOGIA

O modelo base utilizado nos experimentos foi a arquitetura ResNet18, que apresentou o melhor desempenho de classificação em estudos anteriores (FREITAS et al., 2022). O conjunto de dados empregado foi previamente coletado e é composto por imagens recortadas (100×100 pixels) de insetos capturados em armadilhas adesivas, categorizadas em três classes: Ceratitis capitata, Grapholita molesta e outros.

Na abordagem PTQ (post-training quantization), o modelo ResNet18 treinado em precisão total foi submetido a quantização estática por meio da

biblioteca Brevitas, utilizando um subconjunto de dados representativo para calibração, sem etapas adicionais de retreinamento. Já na abordagem QAT (quantization-aware training), o modelo foi reajustado (fine-tuned) por 10 épocas, processando todo o conjunto de dados sob restrições de quantização, de modo a permitir que a rede se adaptasse ao ruído introduzido pela redução de precisão.

Em ambos os cenários, foram avaliadas nove configurações de precisão para pesos (W) e ativações (A), variando de W2A2 até W8A8. O desempenho dos modelos foi analisado com base nas métricas de Acurácia, Precisão, Recall e F1-Score no conjunto de testes.

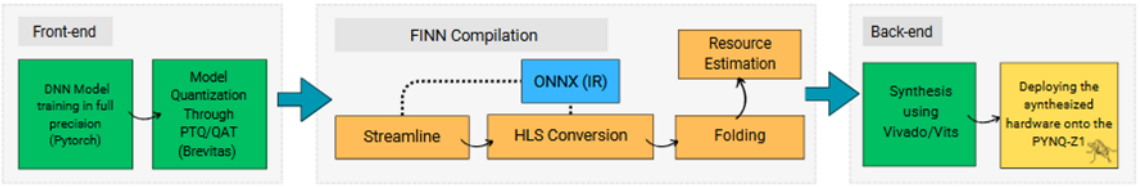

Figura 1: Fluxo de ponta a ponta para implantação de redes neurais quantizadas em FPGA utilizando FINN. Fonte: do autor.

Após a avaliação experimental, as configurações de melhor desempenho foram selecionadas para exportação na plataforma FINN, que possibilita a implementação das redes quantizadas em FPGAs. A Figura 1 ilustra o fluxo de etapas do FINN.

No Front-end, parte inicial do fluxo, utiliza-se um modelo de alta precisão treinado no PyTorch, posteriormente quantizado via Brevitas, de forma a gerar uma aplicação leve para hardware. Esse modelo é então exportado no formato ONNX. Na etapa de Compilação, realizam-se transformações voltadas à compatibilidade com o hardware, incluindo: Streamline: ajuste e simplificação de operações; HLS Conversion: transformação em código C++ compatível com síntese; Folding: definição do grau de paralelismo e reutilização de componentes; e Resource Estimation: estimativa do consumo de recursos pela aplicação. Por fim, o Back-end consiste na síntese do código gerado na etapa de compilação por meio do Vivado/Vitis, direcionada a uma FPGA alvo, como a PYNQ-Z1.

3. RESULTADOS E DISCUSSÃO

As Tabelas 1 e 2 apresentam os resultados de precisão, recall, F1-Score e acurácia para configurações de quantização selecionadas PTQ e QAT. Os resultados apresentados da abordagem PTQ indicam que quantização moderada conseguem manter um bom desempenho. A configuração W^8A^8 atingiu 85,57% de acurácia, mas a performance se degradou drasticamente em cenários de quantização mais agressivos, como o W^8A^2 que apresentou uma acentuada queda de acurácia com 53,51%.

Por outro lado, a abordagem QAT demonstrou uma robustez significativamente superior. A configuração W^8A^4 (pesos de 8 bits e ativações de 4 bits) alcançou a maior acurácia geral, de 87,47%, superando todas as outras configurações. Mesmo modelos com quantização agressiva, como o W^4A^4, mantiveram uma acurácia elevada de 85,12%.

A comparação direta entre as duas técnicas evidencia que o QAT consistentemente supera o PTQ, especialmente em configurações de baixa

precisão como W^8A^2 e W^4A^2, onde o QAT foi capaz de recuperar uma parcela substancial da acurácia que seria perdida com o PTQ.

É notável que modelos como W^8A^4 e W^4A^8 treinados com QAT obtiveram acurácia superior até mesmo ao modelo W^8A^8 com PTQ, ressaltando a capacidade do QAT de não apenas preservar, mas por vezes melhorar o desempenho em regimes de precisão reduzida.

Tabela 1: Precisão, Recall, F1-Score e Acurácia para configurações de Quantização selecionadas (PTQ). Fonte: do autor.

Config	Classe	Precisão	Recall	F1-Score	Acurácia
W^8A^8	ceratitis	0.8125	0.9841	0.8901	0.8557
	grapholita	0.8817	0.9090	0.8951	
	others	0.9322	0.4417	0.5994	
W^8A^4	ceratitis	0.8009	0.9824	0.8824	0.8475
	grapholita	0.8890	0.8689	0.8789	
	others	0.8623	0.5071	0.6387	
W^4A^4	ceratitis	0.7634	0.9824	0.8591	0.8246
	grapholita	0.8928	0.8162	0.8528	
	others	0.8161	0.5179	0.6336	
W^4A^8	ceratitis	0.6976	0.9886	0.8180	0.8038
	grapholita	0.9137	0.7792	0.8411	
	others	0.9146	0.4845	0.6335	
W^8A^2	ceratitis	0.7209	0.8879	0.7957	0.5351
	grapholita	0.9326	0.1507	0.2595	
	others	0.3020	0.8476	0.4453	

Tabela 2: Precisão, Recall, F1-Score e Acurácia para configurações de Quantização selecionadas (QAT). Fonte: do autor.

Config	Classe	Precisão	Recall	F1-Score	Acurácia
W^8A^4	ceratitis	0.9097	0.9516	0.9302	0.8747
	grapholita	0.8780	0.9307	0.9036	
	others	0.7585	0.5607	0.6448	
W^4A^4	ceratitis	0.8887	0.9408	0.9140	0.8512
	grapholita	0.8832	0.8628	0.8729	
	others	0.6722	0.6321	0.6515	
W^8A^8	ceratitis	0.8467	0.9556	0.8979	0.8494
	grapholita	0.9154	0.8484	0.8807	
	others	0.6765	0.6298	0.6523	
W^4A^8	ceratitis	0.8628	0.9738	0.9150	0.8477
	grapholita	0.8872	0.8497	0.8681	
	others	0.6835	0.5786	0.6267	
W^2A^8	ceratitis	0.8627	0.9300	0.8951	0.8249
	grapholita	0.8443	0.8571	0.8507	
	others	0.6497	0.5167	0.5756	

A Figura 2 estabelece a comparação da acurácia entre as duas formas de quantização abordadas, PTQ e QAT. Inicialmente, destaca-se a QAT, de forma geral, apresenta melhor desempenho nessa métrica, em particular no cenário de pesos reduzidos.

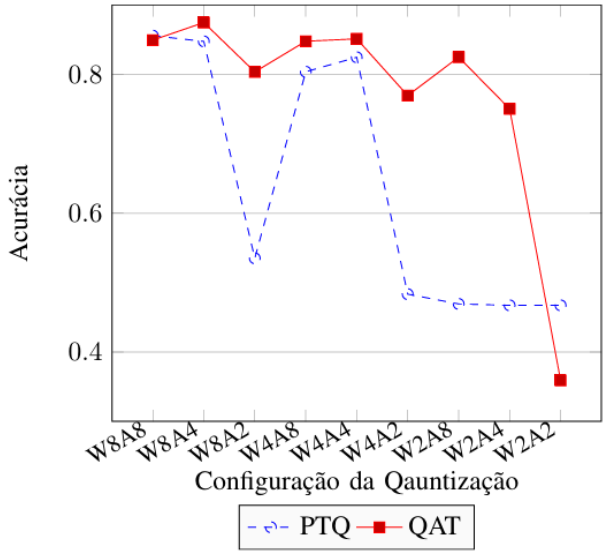

Figura 2: Comparação da acurácia entre PTQ e QAT. Fonte: do autor.

4. CONCLUSÕES

Este trabalho avaliou e comparou as abordagens PTQ e QAT aplicadas ao modelo ResNet18. Os resultados demonstraram que o QAT é mais eficaz do que o PTQ na preservação da acurácia de modelos de aprendizado profundo destinados à classificação de pragas em ambientes com restrições de hardware.

A principal contribuição deste estudo está na demonstração de que o QAT permite ao modelo aprender parâmetros mais robustos à perda de precisão numérica, característica essencial para a implementação eficiente em FPGAs.

Como desdobramento, o próximo passo consiste na utilização do framework FINN para a síntese de uma arquitetura personalizada, que combine alta precisão e baixo consumo energético, visando aplicações em ambientes agrícolas e remotos, onde a disponibilidade de recursos computacionais e energéticos é limitada.

5. REFERÊNCIAS BIBLIOGRÁFICAS

FREITAS, L.C.; MARTINS, V.; DE AGUIAR, M.; DE BRISOLARA, L.B.; FERREIRA JR., P.R.; Deep learning embedded into smart traps for fruit insect pests detection. **ACM Trans. Intell. Syst. Technol**, v. 14, n. 1, 2022.

PASSIAS, A.; TSAKALOS, K.-A.; RIGOGIANNIS, N.; VOGLITSIS, D.; PAPANIKOLAOU, N.; MICHALOPOULOU, M.; BROUFAS, G.; SIRAKOULIS, G. C. Insect pest trap development and dl-based pest detection: A comprehensive review. **IEEE Transactions on AgriFood Electronics**, v. 2, n. 2, p. 323-334, 2024.

UMUROGLU, Y.; JAHRE, M. Streamlined deployment for quantized neural networks. 2018. Online. Disponível em: https://arxiv.org/abs/1709.04060

PAPPALARDO, A.; FRANCO, G.; FRASER, N. J. Xilinx/brevitas: Release v0.12.1. 2025. Online. Disponível em: https://doi.org/10.5281/zenodo.3333552