Análise comparativa da ResNet-50 em FPGA e CPU

José F. C. Martins[1], João V. R. M. G. Canella[1], Gabriel L. P. Sabino[1], Murilo G. Munhoz[1], Calebe P. Bianchini[12]

[1]Centro Universitário FEI
Av. Humberto A. Castelo Branco, 3972-B – São Bernardo do Campo – Brazil

[2]Faculdade de Computação e Informática - FCI
Universidade Presbiteriana Mackenzie – São Paulo, SP – Brazil

{jfelipem115[1], jhony.canella[1]}@gmail.com

{gabriel.pires7[1], murilogmunhoz[1]}@hotmail.com

calebe@fei.edu.br[1], calebe.bianchini@mackenize.br[2]

Resumo. *Este trabalho apresenta um estudo inicial sobre o desempenho da arquitetura de rede neural convolucional ResNet-50 em FPGAs e CPUs, utilizando o ambiente padronizado do Vitis-AI. Avaliou-se a acurácia e o tempo de execução da ResNet-50 no conjunto de dados ImageNet. Neste experimento, os resultados observados no ambiente com o FPGA mostrou vantagem em relação ao tempo de execução, enquanto a CPU teve acurácia ligeiramente superior.*

1. Introdução

As *Convolutional Neural networks* (CNNs) emergiram como uma ferramenta essencial na computação moderna, desempenhando um papel vital em diversas aplicações de visão computacional. Nesse âmbito, as CNNs são especialmente desenvolvidas para processar dados espaciais, capturando padrões e características relevantes. No contexto atual da Inteligência Artificial (IA), a CNN é um pilar crucial, contribuindo significativamente para as previsões de impacto econômico e social da IA [de Couto Júnior 2021].

Como cada vez mais uso de redes neurais se intensifica, o aumento do uso de recursos computacionais - tais como processamento, memória e energia - também foi perceptível. Com o uso de novas tecnologias, tais como as GPUs *Graphics Processing Units* e FPGAs (*Field-Programmable Gate Array*), foi possível viabilizar melhores soluções para o uso de redes neurais [SAS 2023].

A grande vantagem do FPGA é que, ao contrário de outras tecnologias, o circuito dentro do dispositivo pode ser reprogramado quantas vezes for necessário. Os FPGAs também se destacam pelo desempenho, custo-efetividade e menor consumo de energia. Sua flexibilidade e capacidade de reconfiguração após a implementação são essenciais para a adaptabilidade em diversas aplicações [Silva Júnior 2021].

Assim, o objetivo deste trabalho é apresentar o estudo de uma solução de CNN em FPGA, analisando os resultados obtidos e comparando-os com uma solução executada em CPU tradicional, observando aspectos como velocidade de processamento e acurácia. Originalmente, eles foram detalhados no trabalho [Martins et al. 2024].

Este trabalho está estruturado da seguinte forma: a seção 2 apresenta alguns trabalhos relacionados ao tema desenvolvido neste trabalho; a seção 3 apresenta o desenvolvimento e os resultados obtidos e a seção 4 apresenta algumas conclusões.

2. Trabalhos relacionados

Uma implementação de uma *Convolutional Neural Network* (CNN) em FPGA foi apresentada por [Lera and da Costa Bianchi 2019]. Seu objetivo foi de implementar uma CNN no mais alto nível possível e depois transformá-la por processos em uma rede implementada em FPGA. A rede usada reconhece dígitos manuscritos e possui três camadas. Foi desenvolvida na biblioteca *Keras*, em *Python*. O estudo conclui que alguns dos programas utilizados - *hls4ml* e *Vivado High Level Synthesis* - ainda devem ser refinados. Os autores não divulgaram a acurácia da rede, mostrando a viabilidade e o avanço na implementação de *Convolutional Neural Network* em FPGAs.

O estudo realizado por [de Sousa 2019] foi mais profundo em questões de *hardware*, trazendo os conceitos de paralelismo, balanceamento de *pipeline*, armazenamento dos *kernels* e *buffers* tubulares. O objetivo era implementar uma CNN em um FPGA. Os resultados obtidos mostraram que é possível fazer tal implementação. Os autores descreveram os resultados observando as métricas de velocidade e *hardware*. Eles apresentaram o paralelismo nas camadas convolucionais, bem como o paralelismo das convoluções executadas em cada camada. Também apresentaram um *pipeline* ao longo das camadas, que geravam dados para as próximas camadas. Neste caso, 95% do custo computacional estava associado à execução das camadas convolucionais em uma rede AlexNet, ou seja, o paralelismo é uma alternativa viável para reduzir o tempo de execução.

O trabalho apresentado por [Peres 2018] executa uma CNN em um FPGA utilizando técnicas de compressão, tais como o método de *Low-Rank*, o método de *Pruning* e o método de Codificação de Huffman, reduzindo o número de dados e promovendo melhor desempenho na execução. O autor conclui que o *pruning* é uma técnica eficaz, pois reduz muito o tamanho das redes, ao custo de um pequeno impacto na sua precisão. Com a compressão de matrizes esparsas, o resultado obtido foi melhor comparado à codificação de *Huffman*. No caso da técnica de redução de *Low-Rank*, ela obteve menos sucesso nos resultados, porém a redução dos pesos das camadas convolucionais com esta técnica é importante, principalmente para arquiteturas que têm mais pesos nessas camadas.

Os resultados destes e de outros trabalhos[1] auxiliaram a entender as medidas que podem ser usadas para avaliar uma CNN executada em um FPGA. O *throughput*, por exemplo, é expresso em GOP (*Giga Operations Per Second*). Ele oferece um entendimento sobre a eficiência em termos de operações por unidade de tempo. O tempo total, medido em milissegundos (ms), auxilia no entendimento da velocidade de execução de operações no FPGA. A memória, que pode ser quantificada em *megabytes* (MB), mostra o tamanho de ocupação da rede neural. E, por fim, a energia utilizada, mensurada em *watts* (W), reflete o consumo energético, para cada experimento estudado.

3. Desenvolvimento e Resultados

O primeiro passo do desenvolvimento deste trabalho foi escolher um par ideal de tecnologias que permitira experimentar a rede neural convolucional ResNet-50. Dentre os

[1]Por uma questão de espaço, eles foram omitidos, mas estão detalhados em [Martins et al. 2024].

frameworks, destacam-se o Vitis-AI e o OpenVINO. Quanto aos FPGAs que foram estudadas para este experimento estão o FPGA a Xilinx U55C e o FPGA Altera DE2-115.

Ao longo do estudo dessas tecnologias, percebeu-se que a integração entre o *framework* Vitis-AI era nativa com o FPGA Xilinx U55C em comparação ao *framework* OpenVINO e o FPGA Altera DE2-115. Destes últimos, a integração deveria necessariamente ser realizada manualmente, com adaptação dos códigos em VHDL que poderia ser gerados. Por esse motivo, foi escolhido para este experimento o par Vitis-AI e Xilinx U55C.

Para os experimentos na CPU, foram utilizados *frameworks* conhecidos, como *TorchVision* e *PyTorch*. Eles foram feitos em uma CPU Intel(R) Xeon(R) Silver 4210R @ 2.40GHz com 160 GB de RAM DDR4.

Na intenção de melhorar os mecanismos de comparação, redes *ResNet-50* com pesos pré-treinados do *ImageNet* foram utilizadas em todos os experimentos, aproveitando o melhor que cada uma das bibliotecas oferecia. Também foi unificada a forma de coleta das métricas dos experimentos. Neste trabalho, estão abordados somente acurácia e tempo de execução.

Realizaram-se 19 testes, cada um com 1000 imagens, registrando o resultado da classificação de cada imagem. Este *score* indica a porcentagem de certeza com relação à base de treinamento. A acurácia desta rede foi avaliada comparando-as com as etiquetas verdadeiras das imagens. Observou-se também o tempo total de execução para classificar todas as imagens.

A Tabela 1 apresenta os resultados das medidas observadas no FPGA e na CPU. Apesar de utilizar uma rede ResNet-50 pré-treinada a partir da *Imagenet*, entendeu-se que o uso nativo das redes das diferentes bibliotecas (Vitis-AI e *TorchVision*) causou essa diferença. A Figura 1 apresenta essa diferença.

Tabela 1. Estatísticas das execuções em FPGA e em CPU.

	Média	Mediana	Desvio Padrão	Tempo de execução
FPGA	87,4%	89%	0,06	1.750,513s
CPU	93,9%	95%	0,04	6.687,348s

O teste de tempo de execução mediu o experimento usando 1000 imagens previamente escolhidas. Tanto o FPGA quanto a CPU foram avaliadas com as mesmas imagens e o tempo total de execução de cada uma foi, respectivamente, 1.750,513s e 6.687,348s. Neste experimento, o FPGA foi 9,5 mais rápido que a CPU.

4. Considerações Finais

Este trabalho analisou o resultado de uma solução de uma CNN *ResNet-50* em FPGA, comparando-a com uma solução em CPU. Os experimentos realizados demonstraram que é viável a implementação de CNN em FPGAs para acelerar o reconhecimento de imagens.

Apesar dessa viabilidade, os experimentos mostraram acurácia diferentes devido ao uso de redes pré-treinadas diferentes derivadas das bibliotecas utilizadas. Os resultados também mostraram que o tempo de execução total de um experimento pode ser consideravelmente reduzido ao utilizar um FPGA.

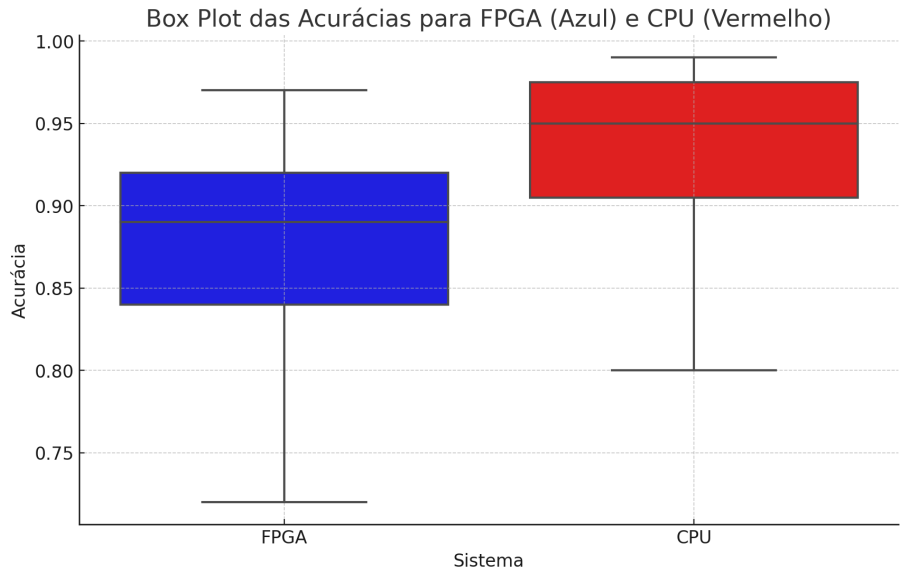

Figura 1. Gráfico de acurácia das execuções em FPGA e em CPU.

Como trabalhos futuros, podem ser citados (i) a utilização de uma mesma rede pré-treinada nas futuras comparações; (ii) incluir comparações com GPUs; (iii) desenvolver um único arcabouço para facilitar os testes e os *benchmarks*; (iv) escolher outras arquiteturas de CNNs; (v) implementar outras técnicas de otimização no FPGA.

Os autores agradecem ao MackCloud[2], Laboratório Multidisciplinar de Computação Científica e Nuvem e ao projeto SPRACE - Processo nº 2018/25225-9, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

de Couto Júnior, H. R. V. (2021). Mapas de textura como entrada em cnns 3d aplicados para classificar nódulos em imagens de tc.

de Sousa, M. C. F. (2019). Método para execução de redes neurais convolucionais em fpga. Master's thesis, Escola Politécnica da Universidade de São Paulo.

Lera, R. D. C. and da Costa Bianchi, R. A. (2019). Aplicação de uma rede neural convolucional em tecnologia fpga.

Martins, J. F. d. C., Sabino, G. L. P., Canella, J. V. R. M. G., and Munhoz, M. G. (2024). Análise comparativa do desempenho da resnet-50 em plataformas fpga e cpu. Trabalho de Conclusão de Curso (Ciência da Computação), Centro Universitário FEI.

Peres, T. A. M. (2018). Otimização de redes neuronais convolucionais em fpga utilizando técnicas de compressão. Master's thesis, Instituto Superior de Engenharia de Lisboa.

SAS (2023). Redes neurais - o que são e qual sua importância? Acesso em: 07 de Janeiro de 2024.

Silva Júnior, J. T. d. (2021). piflowmr - uma nova arquitetura a fluxo de dados dinâmico, escalável e com múltiplos anéis, implementada em um cluster de fpgas de baixo custo. Master's thesis, Universidade de São Paulo.

[2]https://mackcloud.mackenzie.br