



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE
AGH UNIVERSITY OF KRAKOW

Klasyfikacja obrazów na FPGA z użyciem Vitis AI

Problem

- Klasifikacja obrazów z datasetu CIFAR-10
 - 60000 obrazków 32x32 piksele
 - 10 klas – Samochody, samoloty itd.
- Celem jest sprawdzenie jak kwantyzacja i implementacja na FPGA wpływa na czas inferencji i dokładność modelu.

airplane



dog



automobile



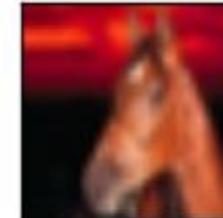
frog



bird



horse



cat



ship



deer

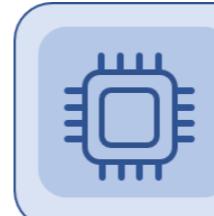


truck



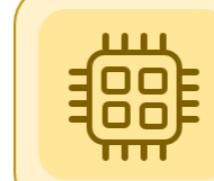
Zalety Machine Learningu na FPGA

- Niskie opóźnienia
- Bardzo dobra wydajność
- Dobry stosunek cena/jakość
- Niskie zużycie energii



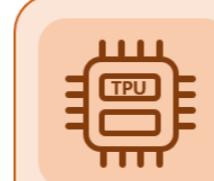
CPU

- Small models
- Small datasets
- Useful for design space exploration



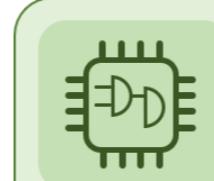
GPU

- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL



TPU

- Matrix computations
- Dense vector processing
- No custom TensorFlow operations

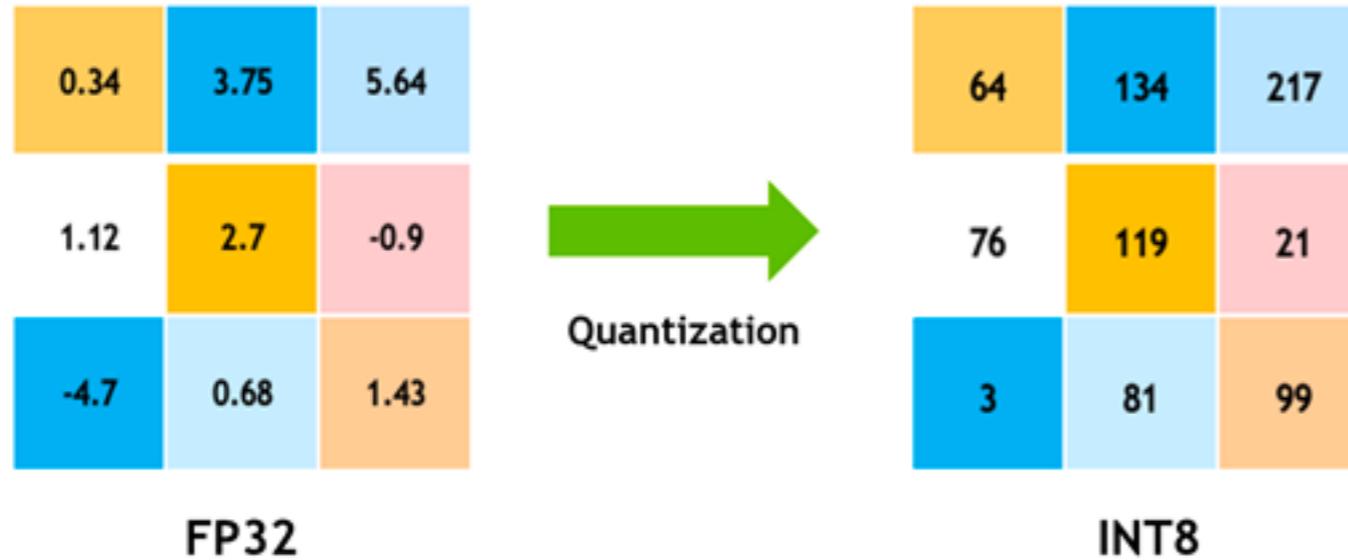


FPGA

- Large datasets, models
- Compute intensive applications
- High performance, high perf./cost ratio

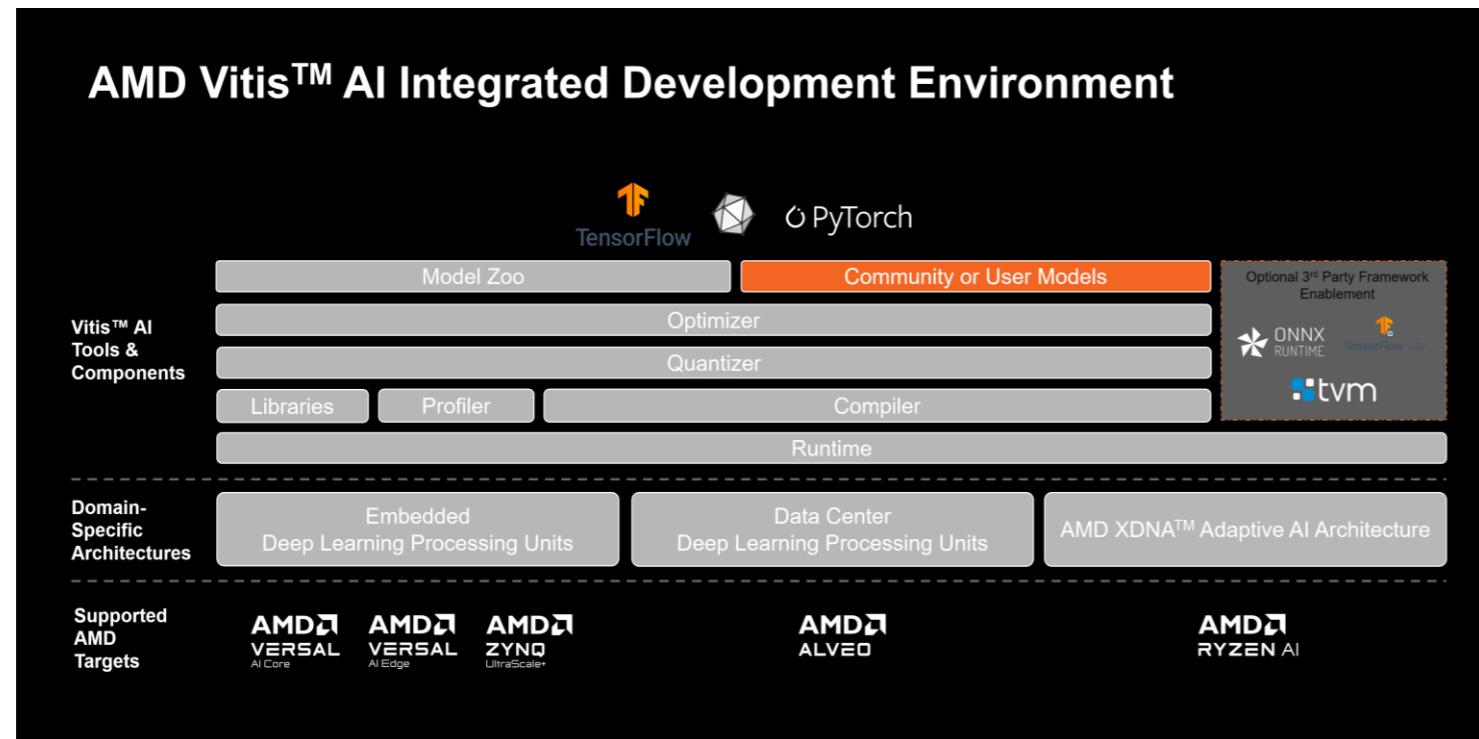
Kwantyzacja

- Kwantyzacja pozwala na znaczne zmniejszenie modelu i jego przyśpieszenie kosztem minimalnego spadku na dokładności.



Vitis AI

- Zintegrowane środowisko do rozwoju aplikacji AI na platformach AMD
- Zawiera narzędzia, biblioteki oraz gotowe modele i przykłady



Etapy pracy

- Trenowanie modelu w PyTorch
 - MobileNetV2 na zbiorze CIFAR-10
- Eksport do ONNX
 - `torch.onnx.export(model, input, "model.onnx")`
- Kwantyzacja
 - Użycie Vitis AI Quantizer (`vai_q_pytorch`) do konwersji na INT8
- Kompilacja
 - Kompilacja modelem w Vitis AI Compiler pod konkretną platformę FPGA
- Uruchomienie inferencji
 - Wykorzystanie Vitis AI Runtime (VART) do wykonania na FPGA

Wyniki

- Trenowanie modelu w PyTorch
 - MobileNetV2 na zbiorze CIFAR-10

```
... Epoka 1/5
    Train: loss=0.4381, acc=84.98%
    Test : loss=0.3200, acc=88.84%
Epoka 2/5
    Train: loss=0.2711, acc=90.70%
    Test : loss=0.3129, acc=89.58%
Epoka 3/5
    Train: loss=0.2248, acc=92.38%
    Test : loss=0.2680, acc=90.97%
Epoka 4/5
    Train: loss=0.1938, acc=93.36%
    Test : loss=0.2668, acc=91.08%
Epoka 5/5
    Train: loss=0.1699, acc=94.09%
    Test : loss=0.2888, acc=90.81%
```

Wyniki

- Eksport do ONNX

- ~~torch.onnx.export(model, input, "model.onnx")~~

```
[torch.onnx] Obtain model graph for `MobileNetV2([...])` with `torch.export.export(..., strict=False)`...
[torch.onnx] Obtain model graph for `MobileNetV2([...])` with `torch.export.export(..., strict=False)`... ✓
[torch.onnx] Run decomposition...
[torch.onnx] Run decomposition... ✓
[torch.onnx] Translate the graph into ONNX...
[torch.onnx] Translate the graph into ONNX... ✓
Applied 104 of general pattern rewrite rules.
Model wyeksportowany do mobilenetv2_cifar10.onnx
```

- Eksport pyTorch (plik *.pt)

 mobilenetv2_cifar10.pt	17.12.2025 ...	Plik PT	9 322 KB
 mobilenetv2_cifar10.onnx	17.12.2025 ...	Plik ONNX	246 KB
 mobilenetv2_cifar10.onnx.data	17.12.2025 ...	Plik DATA	8 704 KB



Wyniki

- Vitis AI 😞

Źródła

- <https://my.avnet.com/silica/resources/article/fpga-vs-gpu-vs-cpu-hardware-options-for-ai-applications/>
- <https://xilinx.github.io/Vitis-AI/3.5/html/index.html>