

Uczenie maszynowe (sieci neuronowe) z zastosowaniem FPGA

KAJETAN KRUCZKOWSKI

Uczenie maszynowe, a sieci neuronowe

Jest to praktyka pozwalająca na wykrywanie przez komputer ukrytych wzorców oraz regularności w dostarczonych danych.

►Zastosowanie:

- problemy wymagające dużej ilości zasad
 - Skomplikowane problemy (brak rozwiązań przy tradycyjnym podejściu)
 - Niestabilne środowisko
-
- Sieć neuronową można opisać jako matematyczny model do przetwarzania informacji.
 - Zastosowania:
 - -autonomiczne auta
 - -rozpoznawanie mowy
 - -rozpoznawanie obrazów
 - -giełda
 - -marketing – profilowane reklamy
 - -medycyna
 - -robotyka/automatyzacja

Dlaczego FPGA?

- ▶ FPGA oferuje znacznie mniejsze opóźnienia, w przeciwieństwie do GPU, które wymaga komunikacji z CPU (architektura bazująca na instrukcjach).
- ▶ Mniejsze zużycie energii, doskonałe możliwości zrównoleglenia obliczeń.
- ▶ FPGA może wykorzystać CPU w celu akceleracji.

Serwery Amazon oraz Microsoft działają na FPGA.

Dlaczego FPGA?

	NVIDIA JETSON TX2	XILINX ZYNQ ULTRASCALE + MPSOC
Frequency	1.3 GHz	0.3 GHz
Power	7 W	4.5 W
Speed	1.5 FPS	35 FPS
Energy efficiency	0.2 FPS/Watt	7.9 FPS/Watt

Benchmarking YOLOv2 algorithm in GPU and FPGA

Dlaczego FPGA?

Domain	Algorithm	Winner	Factor
Physics Simulation	HotSpot	GPU	0.59
Fluid Dynamics	Flux	GPU	0.35
Data Mining	KNN	GPU	0.32
Image Processing	SRAD	FPGA	4.52
Medical Imaging	GICOV	FPGA	7.76
Bioinformatics	HW	FPGA	19.79

DPU

- ▶ Xilinx® Deep Learning Processor Unit (DPU) jest programowalnym silnikiem dedykowanym dla konwolucyjnych sieci neuronowych. Jednostka zawiera moduł konfiguracji rejestru, moduł kontrolera danych oraz moduł obliczeń konwolucji (splotów). Do DPU dołączony jest specjalistyczny zestaw instrukcji, który umożliwia wydajną pracę DPU dla wielu sieci CNN. DPU obejmuje VGG, ResNet, GoogLeNet, YOLO, SSD, MobileNet, FPN, itp.
- ▶ Do komunikacji processor <-> DPU wykorzystywana jest magistrala AXI.

Xilinx – Edge AI Platform Tutorial

- ▶ Utworzony na bazie danych CIFAR-10 – 60k zdjęć 32x32x8.
- ▶ Wymaga:
 - ▶ 1. Hosta Ubuntu
 - ▶ 2. Docker'a (bazującego na nvidia)
 - ▶ 3. Putty
 - ▶ 4. Płytki prototypowej na bazie ZCU102



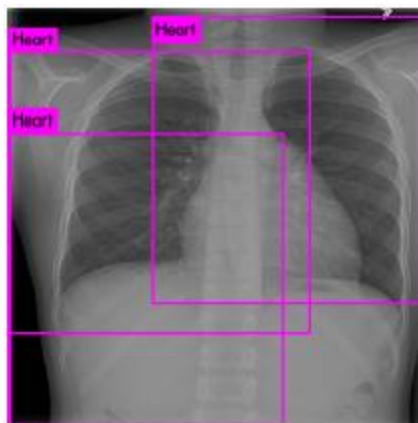
Na oficjalnym githubie Xilinx'a znajdują się skrypty potrzebne do Przeprowadzenia zadania.

Class index	Class name
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

Przykład przed zmianami Xilinx'a

- ▶ Przygotowanie danych oraz trening sieci na Darknet yolov3 przy pomocy środowiska stworzonego na Dockerze.
- ▶ Po przeprowadzeniu uczenia wymagane jest przekonwertowanie sieci oraz przystosowanie do pracy na płycie z DPU.

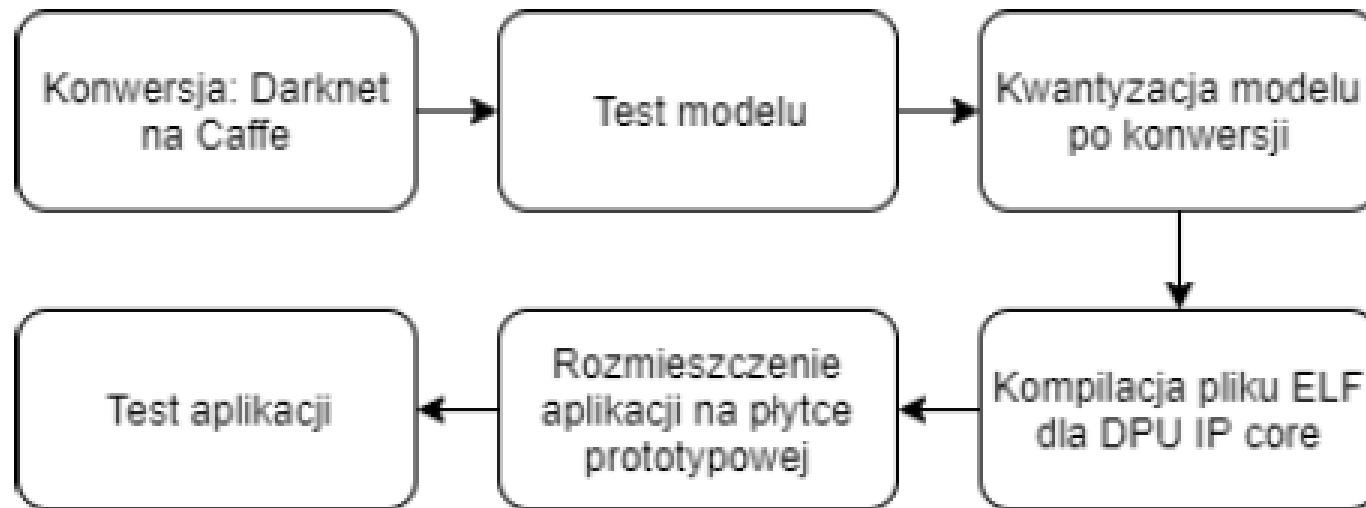
```
classes= 2  
train  = data/train.txt  
valid  = data/test.txt  
names  = data/obj.names  
backup = backup/
```



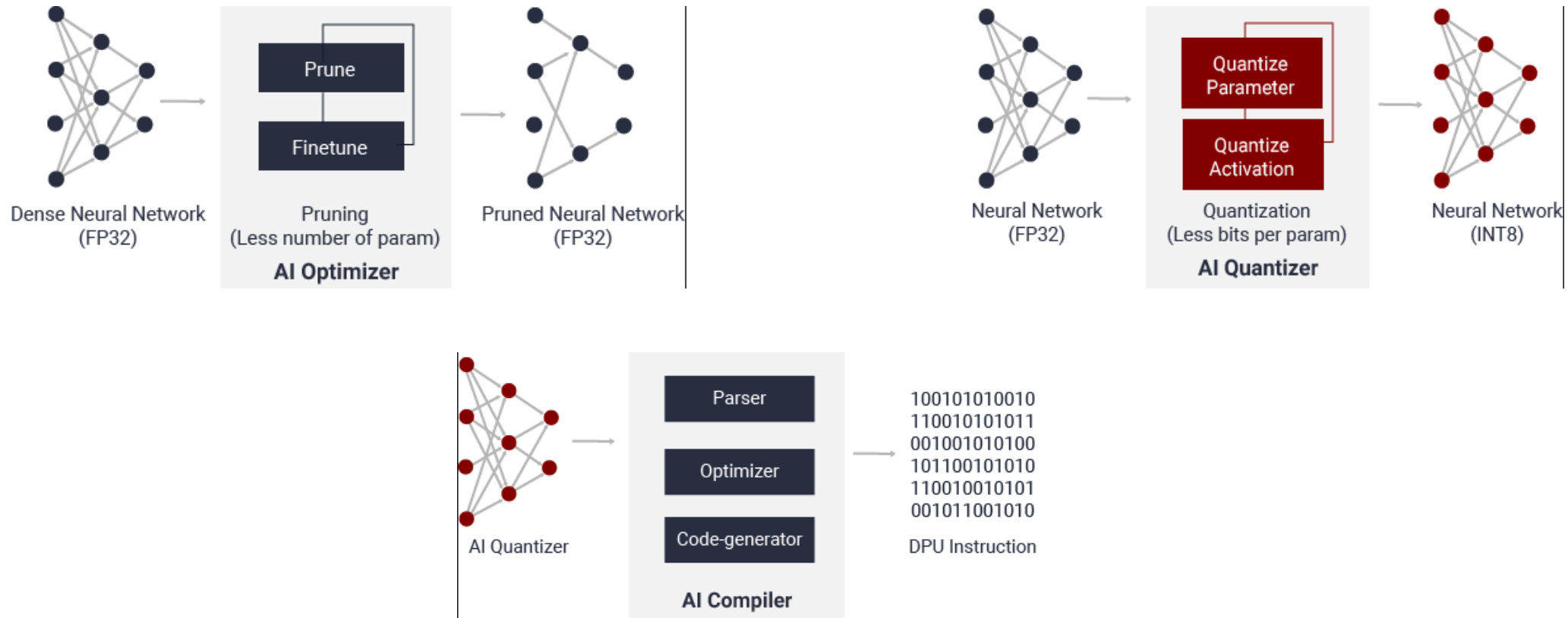
Aldec TySOM-3 (Xilinx
Zynq UltraScale+
ZU7EV)



Przykład przed zmianami Xilinx'a - część programowa



Przykład przed zmianami Xilinx'a - część programowa



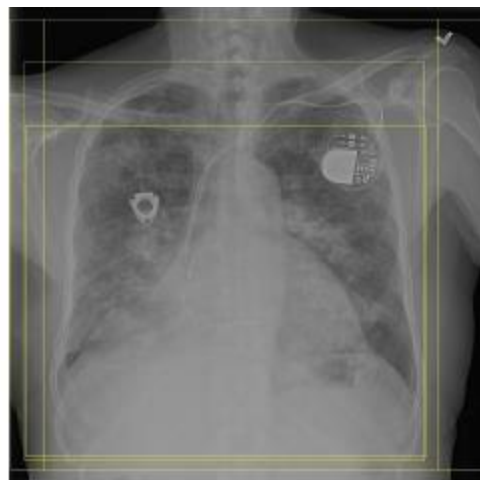
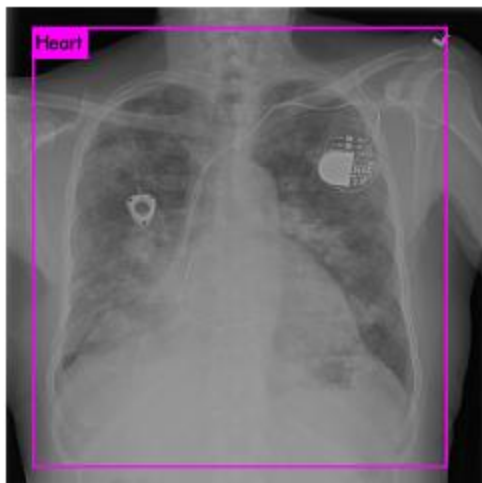
Przykład przed zmianami Xilinx'a - część sprzętowa

Pierwszym krokiem jest generacja plików do partycji BOOT, w celu uruchomienia systemu (program SDx). W wyniku otrzymujemy pliki generacyjne partycji BOOT: BOOT.BIN, devicetree.dtb, uramdisk.image.gz.. Do komunikacji, pomiędzy użytkownikiem a systemem plików służy system Petalinux.

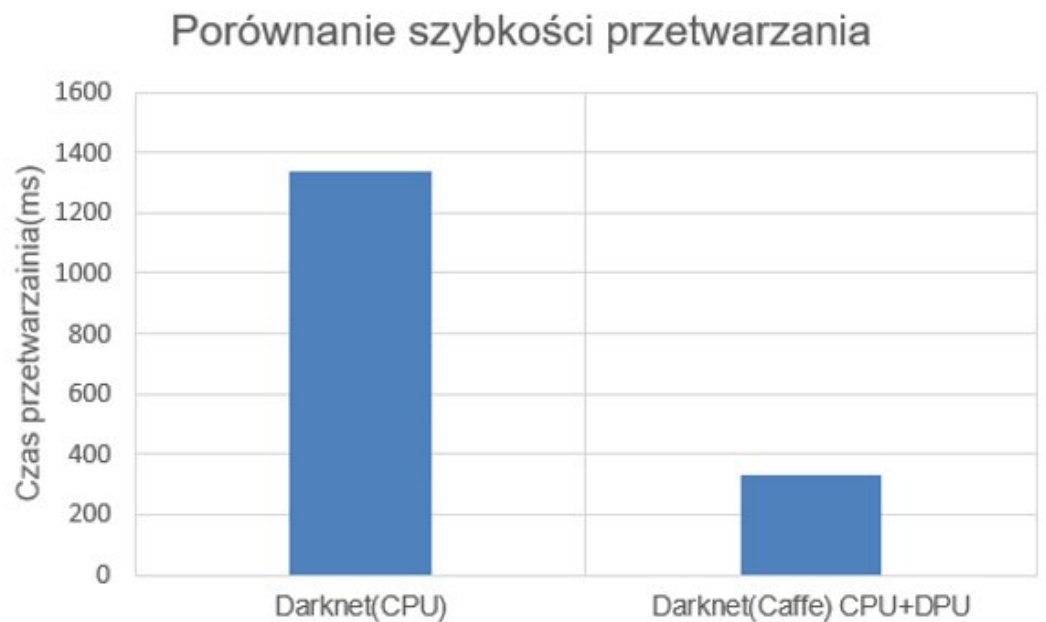
Kolejnym krokiem jest dokonanie kompilacji hybrydowej, mającej na celu połączenie jednostki DPU, z architekturą 64-bitową procesora ARM Cortex A53. W wyniku kompilacji hybrydowej otrzymujemy plik wykonywalny, przez system procesorowy o nazwie yolo.elf.

Przykład przed zmianami Xilinx'a

Wyniki predykcji na komputerze oraz płytce prototypowej.



Przykład przed zmianami Xilinx'a



Typ	Darknet(CPU)	Caffe(Darknet) CPU+DPU
czas przetwarzania(ms)	1336	332,761

Źródła

- ▶ Fang, J., Mulder, Y.T.B., Hidders, J. *et al.* In-memory database acceleration on FPGAs: a survey. *The VLDB Journal* **29**, 33–59 (2020).
<https://doi.org/10.1007/s00778-019-00581-w>
- ▶ <https://haltian.com/news/fpga-vs-gpu/> (benchmarks GPU vs FPGA)
- ▶ <https://github.com/Xilinx/Vitis-AI>