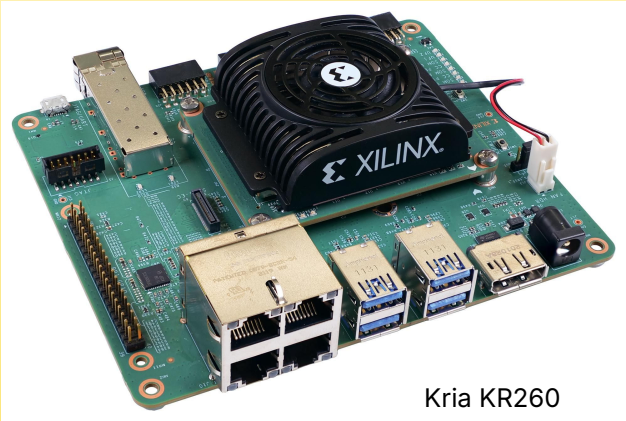
Deep Learning Reconstruction of Optoacoustic images on FPGA



Kria KR260

Name: **Habib Ben Abda**

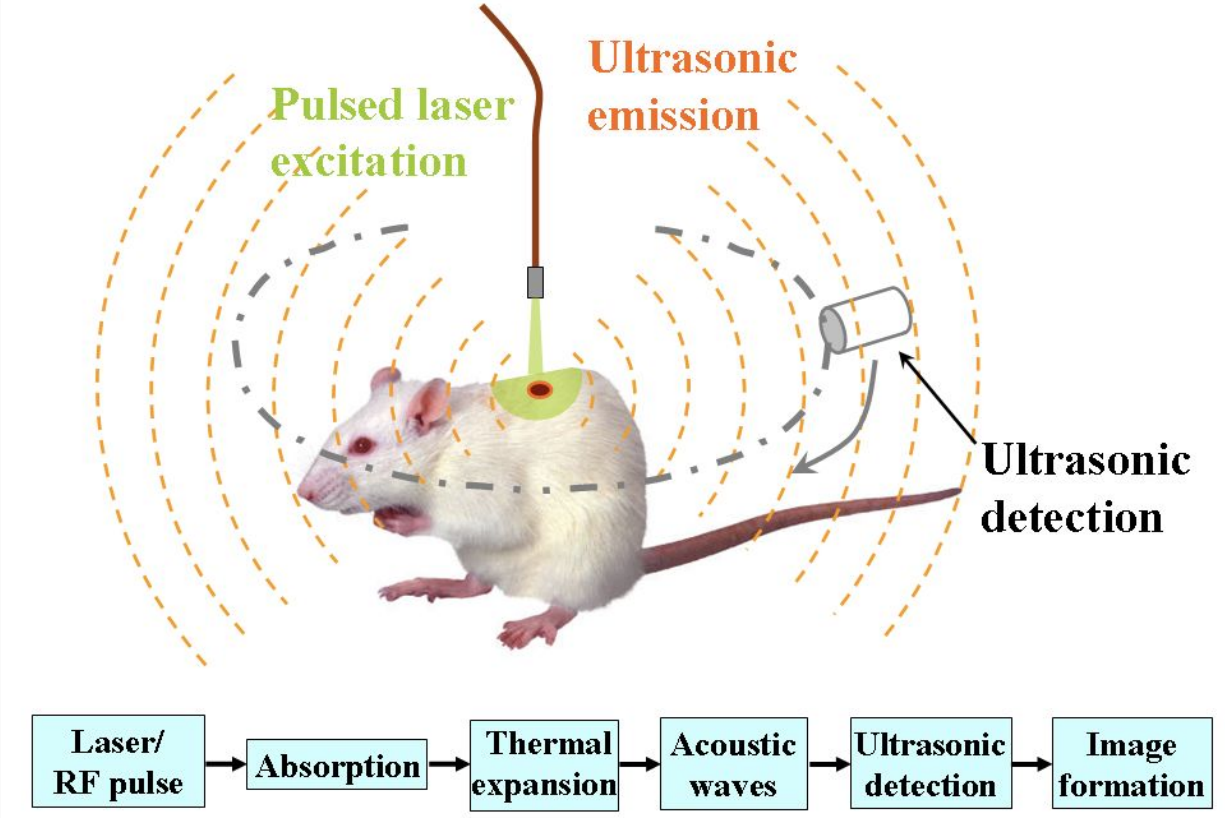
Exam: **System on Chip for Data Analytics and Machine Learning**

Date: **21/08/2025**

Optoacoustic Imaging

1. Laser pulse generates thermal expansion

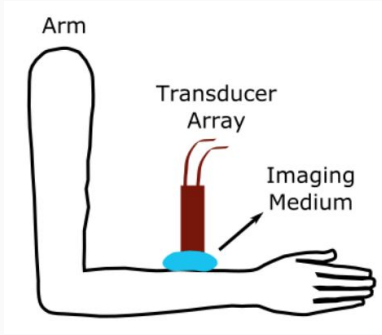
2. Pressure wave from expansion detected by ultrasonic receiver



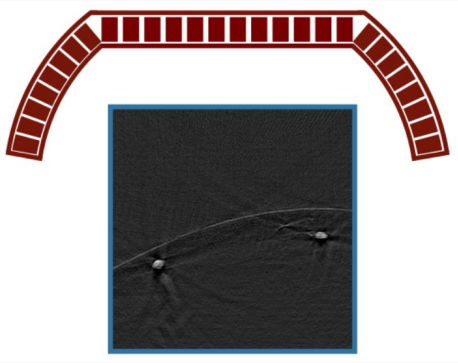
https://en.wikipedia.org/wiki/Photoacoustic_imaging

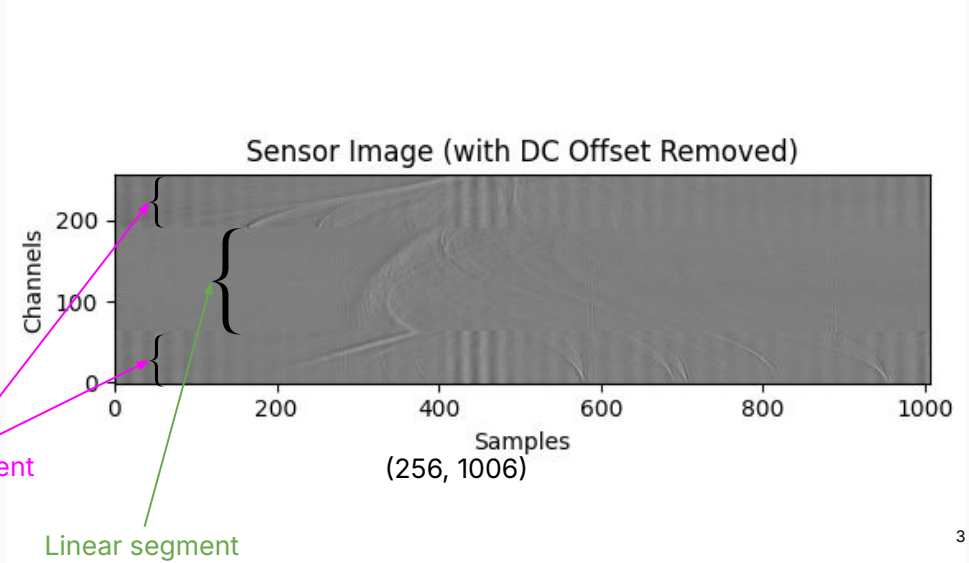
Dataset

- Prof. Dr. Daniel Razansky's Lab
- 4 recordings:
  - Right arm normal
  - Right arm parallel
  - Left arm normal
  - Left arm parallel
- ~ 5500 images
- Float32
- 256 channels, 1006 time samples



Multisegment array



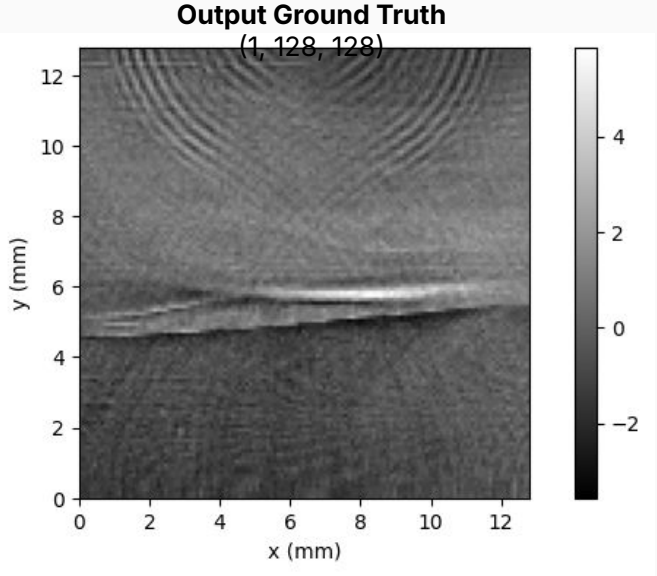


F. Ozdemir, et al., "OADAT: Experimental and Synthetic Clinical Optoacoustic Data for Standardized Image Processing," May 03, 2023, arXiv

Standard Reconstruction

Delay and Sum:

```
for each pixel (x,y):
    for each channel:
        d = dist(receiver, pixel)
        delay = d / speed_of_sound
        pixel_value += time_signal(delay)
```



- Inverse problem with high computational load
- Data dependency: each pixel involves signals from many sensors
- Inefficient GPU utilization (non-trivial indexing and irregular memory access)

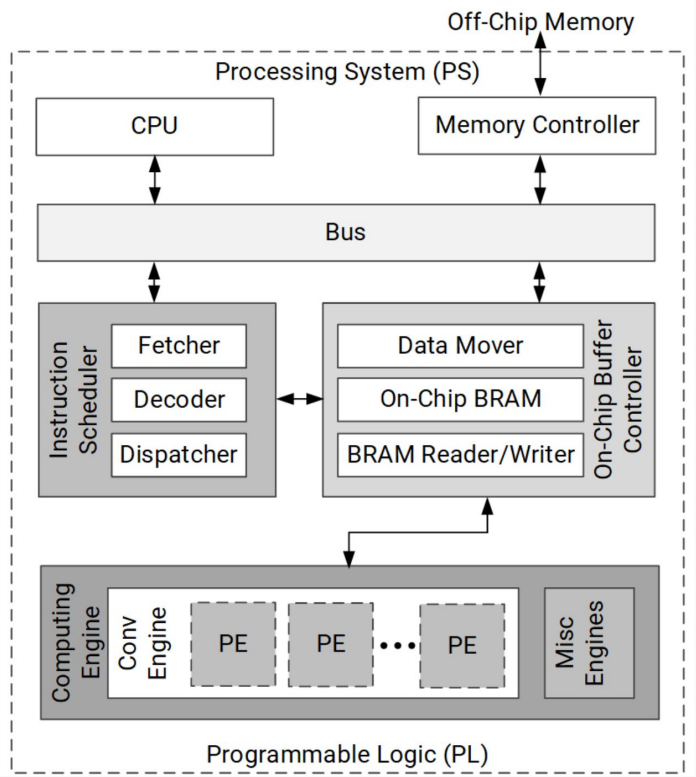
Objective:

→ Real time implementation on embedded device

→ Use Deep Learning interesting for fast inference

Target Device: AMD Kria™ K26 SOM

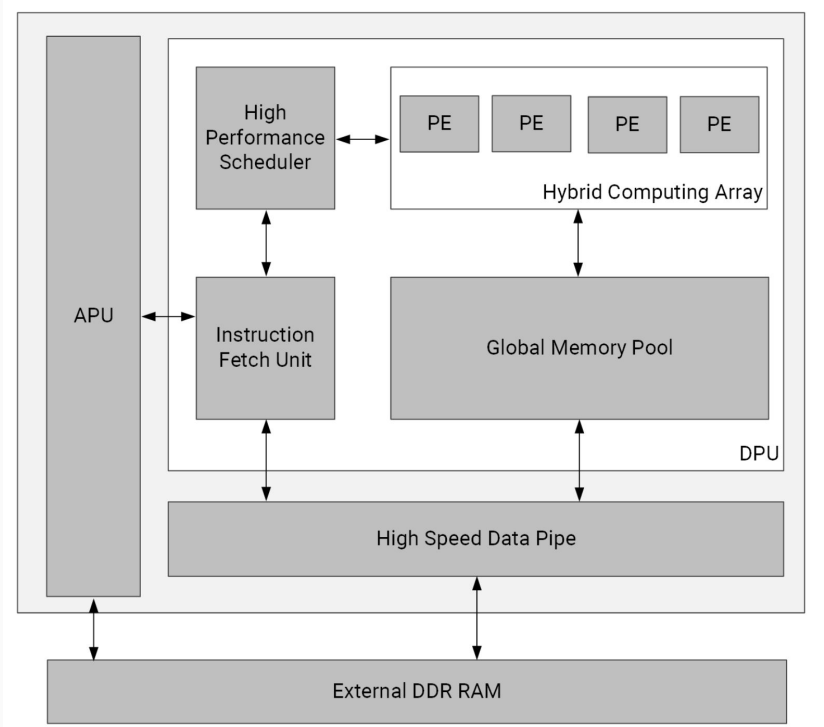
FPGA Fabric	Xilinx Zynq UltraScale+™ MPSoC EV (XCK26)
Deep Learning Processing Unit (DPU)	Up to 1.4 TOPS (peak performance)
CPU	Quad-core Arm® Cortex®-A53 MPCore™ up to 1.5 GHz
Memory	4 GB DDR4 RAM



DPU: microcoded compute engine, optimized instruction set, can support most CNN networks

Target Device: AMD Kria™ K26 SOM

FPGA Fabric	Xilinx Zynq UltraScale+™ MPSoC EV (XCK26)
Deep Learning Processing Unit (DPU)	Up to **1.4 TOPS** (peak performance)
CPU	Quad-core Arm® Cortex®-A53 MPCore™ up to 1.5 GHz
Memory	4 GB DDR4 RAM



DPU: microcoded compute engine, optimized instruction set, can support most CNN networks

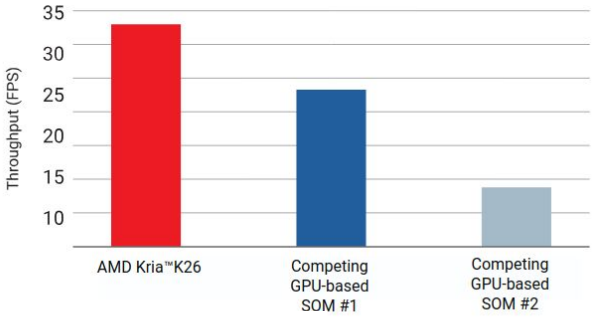
https://docs.amd.com/r/en-US/pg338-dpu/Core-Overview

Target Device: AMD Kria™ K26 SOM

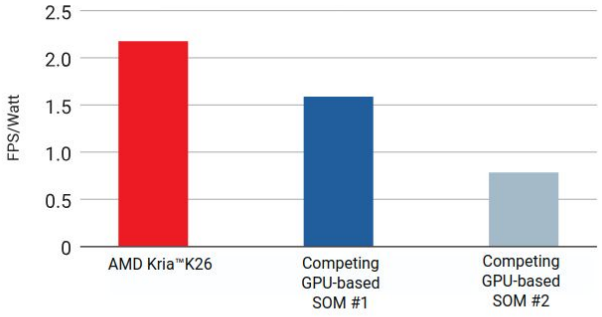
BENCHMARKS

Video Pipeline with AI

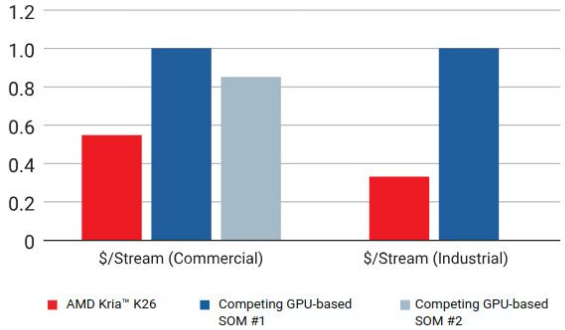
3X Performance Advantage



2.2X Performance-per-Watt



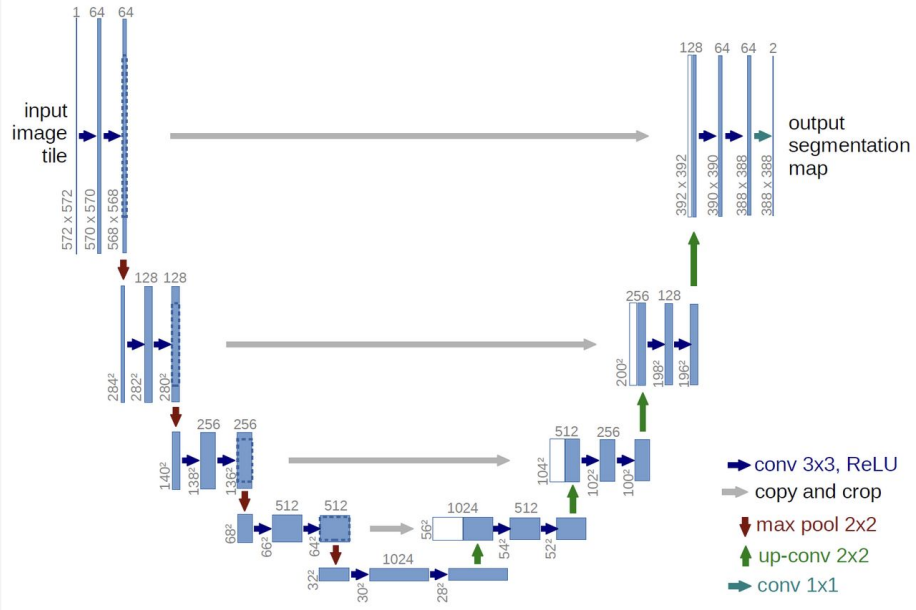
Half the Cost per Video Stream



https://www.xilinx.com/publications/product-briefs/xilinx-k26-product-brief.pdf

Model choice: U-Net

- One of the best networks for biomedical image segmentation/reconstruction since 2015
- Fast & precise
- CNN based



Can’t use directly. Need some adjustments

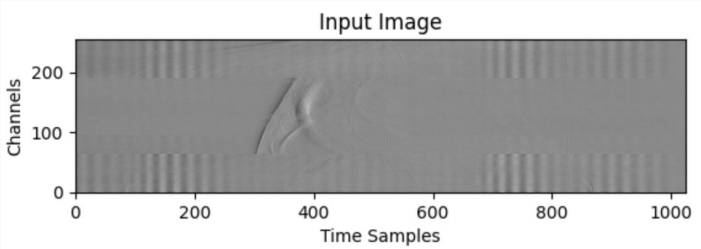
- Different size input and output
- Different number of channels

O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015

Preprocessing Pipeline

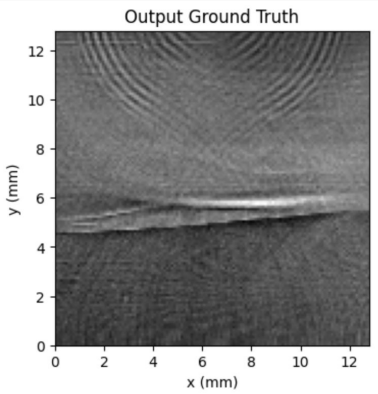
Input:

- Remove DC offset
- Median padding time signal (256,1006) → (256,1024)
- Normalizing (zero mean, unit variance)
- Removing outliers (first ~150 images of each recording)



Generated Labels:

- (C,H,W) = (1, 128, 128)
- Corresponds to 250µm resolution
- Normalization (zero mean, unit variance)



Vitis-AI

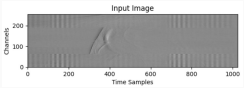
IDE to deploy AI models and accelerate AI inference on AMD hardware platforms

Vitis-AI Inspector:

Check if model is supported by DPU

- Can’t use adaptive pooling
- Can’t use resizing
- Can’t use cropping function or any other custom functions
- Can’t use random sized kernels and strides

Custom-Unet-256CH



(1, 256×1024) → (32, 256×1024)

(32, 128×512) → (64, 128×512)

(64, 64×256) → (128, 64×256)

(128, 32×128) → (256, 32×128)

(256, 16×64) → (512, 16×64)

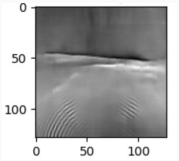
(512, 16×16) → (256, 16×16)

(256, 32×32) → (512, 32×32)

(128, 64×64) → (256, 64×64)

(64, 128×128) → (128, 128×128)

(32, 128×128) → (1, 128×128)



Conv 2D
BatchNorm
ReLu

Conv 2D

Skip
connection

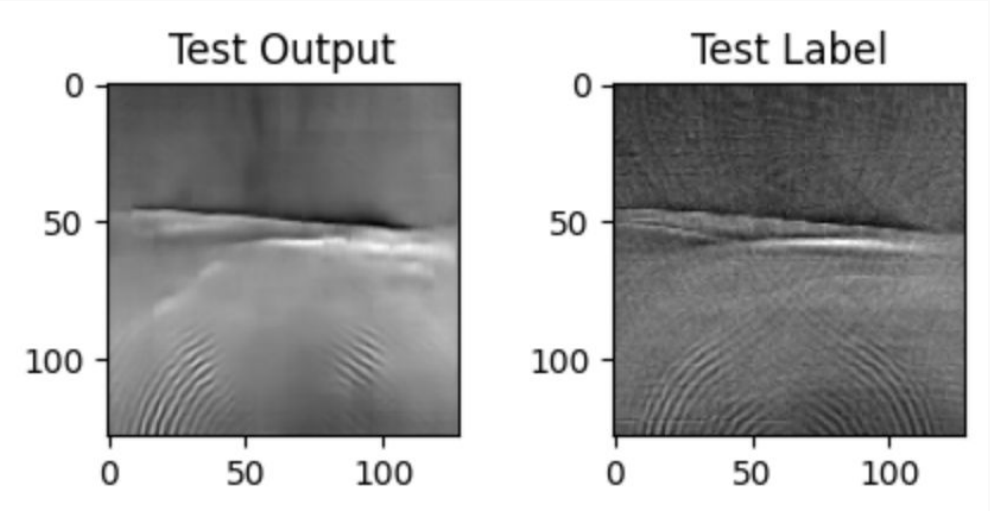
Max Pooling

Upsampling

Result

Custom-Unet-256CH

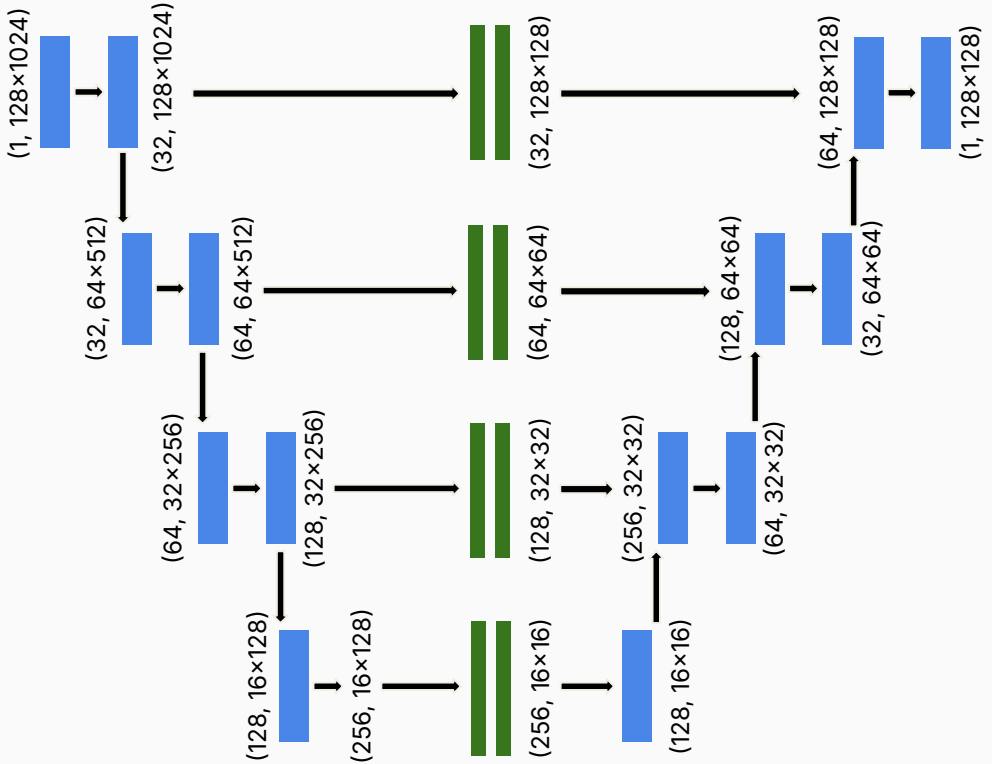
MSE on test set	0.5672
Parameter size	28.8 MB
Estimated **Total size**	617 MB
Inference time (on laptop)	166 ms → 6 fps

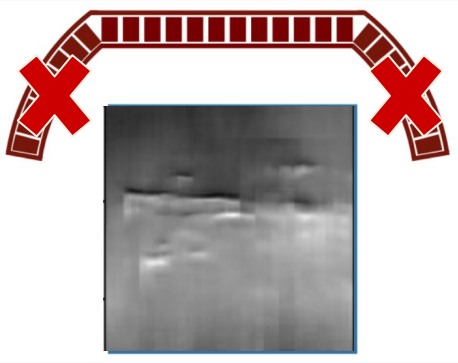


With DPU acceleration, very realistic to reach real-time imaging

Custom-Unet-128CH

Taking only linear segment





Conv 2D
BatchNorm
ReLu

Conv 2D

Skip
connection

Max Pooling

Upsampling

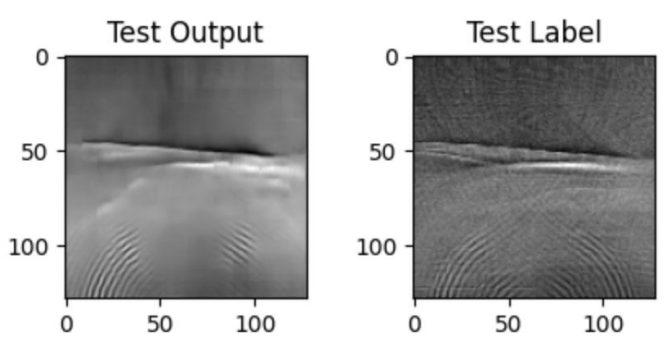
Result

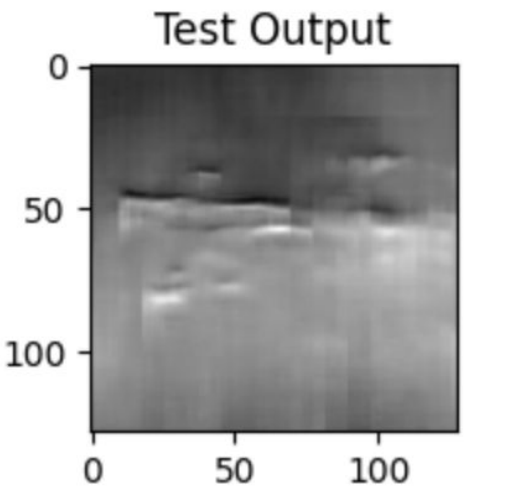
Custom-Unet-256CH

MSE on test set	0.5672
Parameter size	28.8 MB
Estimated **Total size**	617 MB
Inference time (on laptop)	166 ms → 6 fps

Custom-Unet-128CH

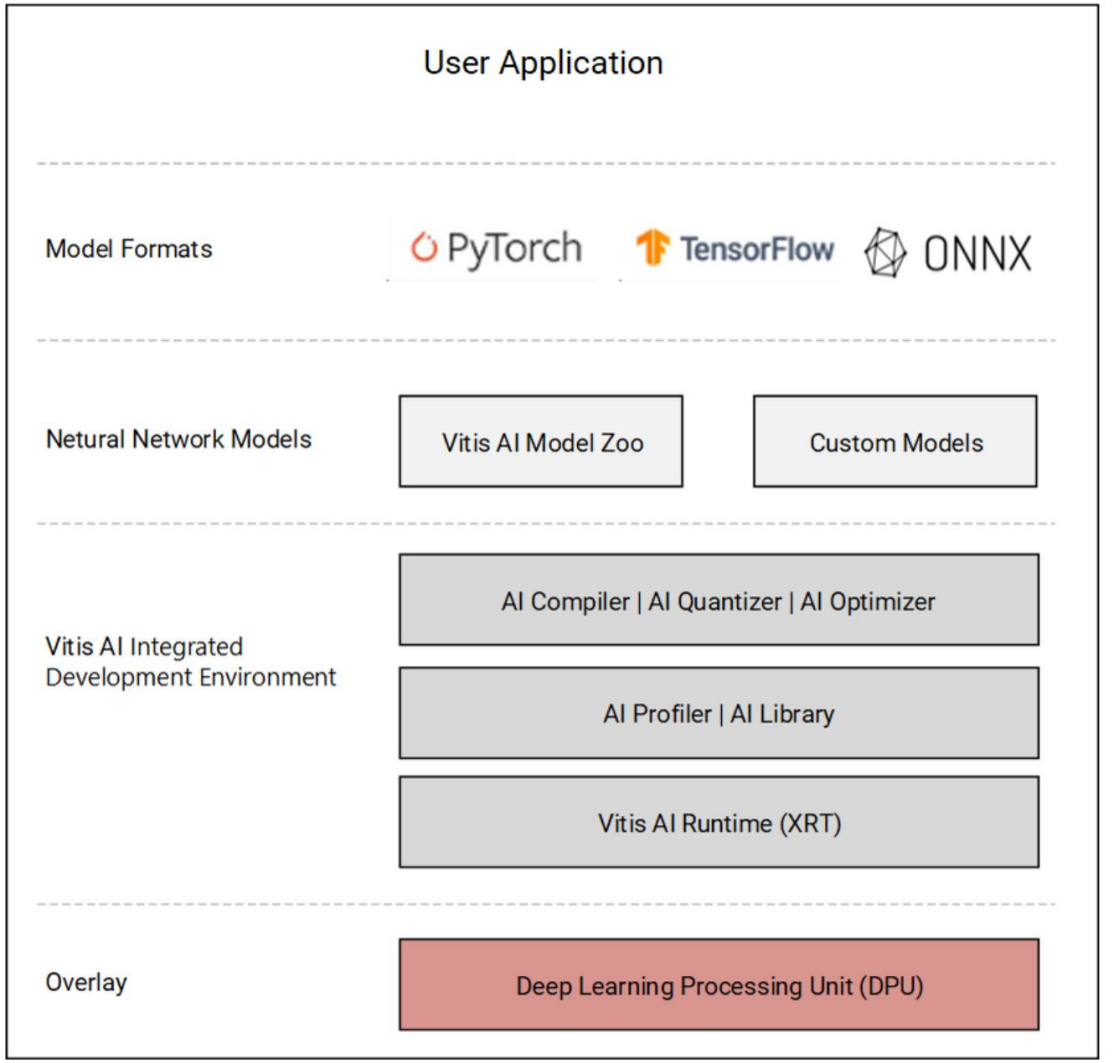
MSE on test set	0.7034
Parameter size	7.87 MB
Estimated **Total size**	301 MB
Inference time (on laptop)	87ms → 11.5fps





Vertical resolution lost

Development flow



https://docs.amd.com/r/en-US/pg338-dpu/Core-Overview

Quantization & Deployment

Before Quantization

MSE: 0.5672

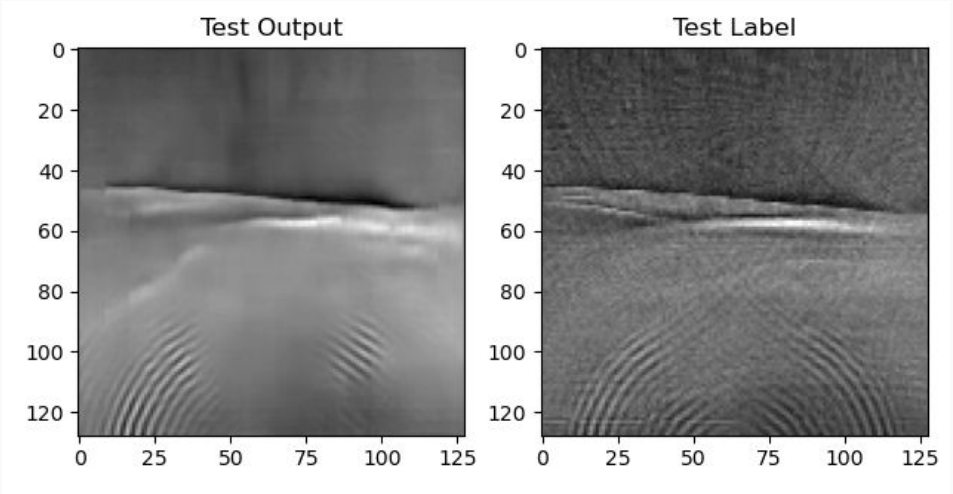
Params size: 28.8 MB

Post training quantization (Float32 → Int8)

MSE: 0.5706

Params size: 7.5 MB

- Smaller than 4Gb DDR4



→ Generated .xmodel file deployable on FPGA

Conclusion

Main tasks:

1. Generated dataset
2. Optimized model for DPU while preserving accuracy
3. Explored subchannel sampling
4. Quantized the model using Vitis-AI tool and generated deployable model

→ Proof of concept: **can implement “real-time” optoacoustic imaging on Kria FPGA using deep learning**

Limitation: Generalization

Thank you!

Q&A

Remarks or less useful

- Don’t have display connector nor ethernet port or router
- ROS on KR260: https://github.com/amd/Kria-RoboticsAI
- Youtuber 1, getting started: https://www.youtube.com/watch?v=N6UBuat8f2U
- Youtuber 1 ai kv26: https://www.youtube.com/watch?v=L_Zd9R4ZeiI
- Mnist github: https://github.com/Xilinx/Vitis-AI-Tutorials/tree/1.4/Design_Tutorials/09-mnist_pyt
- Resnet github: https://github.com/Xilinx/Vitis-AI/blob/3.0/src/vai_quantizer/vai_q_pytorch/example/resnet18_quant.py
-

Useful

- Vitis AI compression flow: https://docs.amd.com/r/en-US/ug1414-vitis-ai/Vitis-AI-Quantizer
- KR260 AMD user guide: https://docs.amd.com/r/en-US/ug1092-kr260-starter-kit/Summary
- Youtube video Useful for presentation: https://www.youtube.com/watch?v=7JmM94AJda0&pp=ygUZdHJhaW4gbW9kZWwgd2l0aCB2aXRpcyBhaQ%3D%3D
- Kr26 doc page: https://xilinx.github.io/kria-apps-docs/kr260/build/html/docs/10gige_vision_camera/10gige_vision_camera_landing.html#tutorials
-

Useful

- Pytorch pruning: https://docs.pytorch.org/tutorials/intermediate/pruning_tutorial.html
- Mouser good doc kr26: https://www.mouser.com/pdfDocs/wp529-som-benchmarks.pdf

Exam structure:

- **Suggested Structure**:
  - **Goal** of the project
  - **Methodology** used
  - **Results** achieved
  - **Comparison** to state-of-the-art
- **Be Prepared** to answer detailed follow-up questions.

Quantization vitis-ai:
https://docs.amd.com/r/en-US/ug1414-vitis-ai/Running-Quantization-and-Getting-the-Result