PIRM Presentation 2

EE / CprE / SE 492 – sddec20-proj01

Machine learning for pilot biometrics

Team Members:

Jianhang Liu, Feng Lin, Xuewen Jiang, Xiuyuan Guo, Sicheng Zeng, Junjie Chen

Email: zengsc@iastate.edu

Sponsor: Rockwell Collins

Project Goals

- Improve existing machine learning algorithm in terms of :
  - Higher Accuracy
    - Increased training dataset
    - Data pre-manipulating
  - Less Latency
    - FPGA(DPU)
    - Pruning
    - Hyper-Parameter
    - Data pre-manipulating
  - Less Memory
    - Pruning
    - Data pre-manipulating
    - quantization

Current Status

Hardware:

- Loaded existing FPGA design (DPU implementation).
- Communication between CPU and DPU
- daughter card design for our edge(ULTRA96v2 development board).（BOM，Schematic, layout)

Hyper-parameter:

- Using grid search to find optimized number for hyper-parameter such as batch size, kernel size , activation function and etc.

Pruning:

- After pruning our model, we can use less memory for the inference.

Data Pre-manipulating:

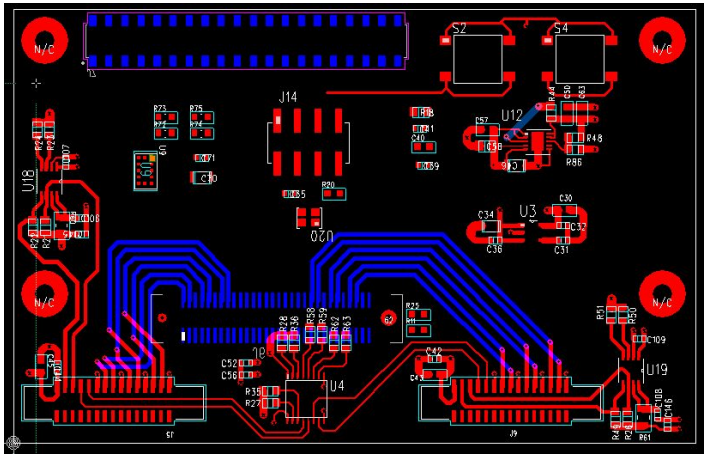
- Over exposure image will improve the result.
- The gabor filter and other types of filters will decrease performance of the model.

Technical Challenges

- Compatibility (versions, model format)
  - Software,system versions
  - Model format such as quantized model or .h5 model
- PCB design
  - Don’t have experience on assemble PCB and testing
- FPGA implementation
  - DPU configuration
  - DPU communication

PCB Design: Schematic & Layout

- Re-routing for better schematic design
- Tools: PCB123
- Schematic: Xuewen Jiang
- Layout: Jianhang Liu and Issac
- BOM, 3 pages of schematic, 4 layer PCB board, 85mmx54mm
- Estimated finishing time: in a week



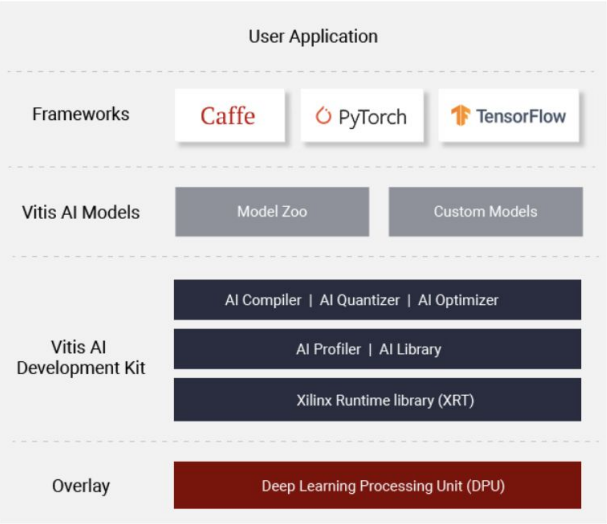
Jianhang (1)

Image processing

- In order to improve system performance (reduce time delay and memory usage, improve accuracy, etc.)
- Many ways to process image (e.g. edge detection, noise reduction, gabor filter)
- Some improve overall performance while others not.
- Need multiple trails to decide best filter parameters.

Example of edge detection,
Image from BogoToBogo

Port the Algorithm to FPGA

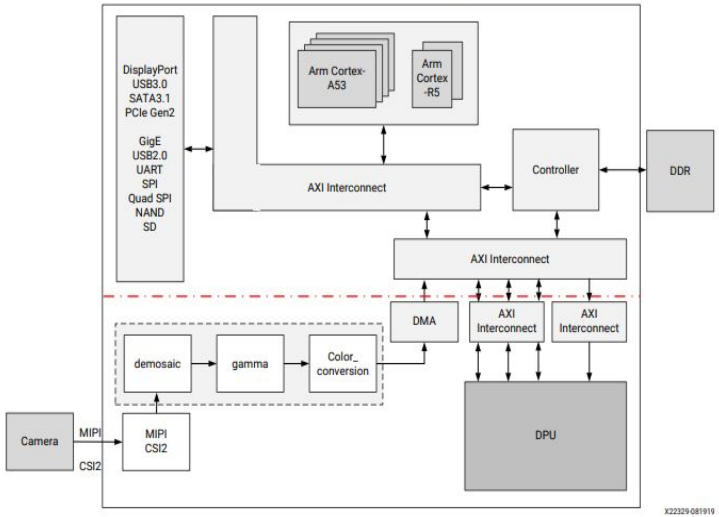


- Traditionality done with vivado, HDL
- ‘Vitis AI’ stack for fast scripting
- pre-built DPU support conv,pool.. operation
- feed in the model->run the scripts->mapping of network on FPGA

FPGA for our project

- In our project we are using Xilinx DPU(Deep learning process unit) technology to approach FPGA acceleration goal.
- DPU could accelerate inference speed for our model
  - Multi-thread(Pthread)
  - Multiple layers speed up implementation
- The workflow will be:
  - Capture data from camera send them to the DPU so that DPU could calculate weights, bais and activation function for the CPU(in this case our CPU will be Arm Cortex A53)
  - Then output the result from DPU to CPU
  - We could connect a display device to show what final output from CPU.

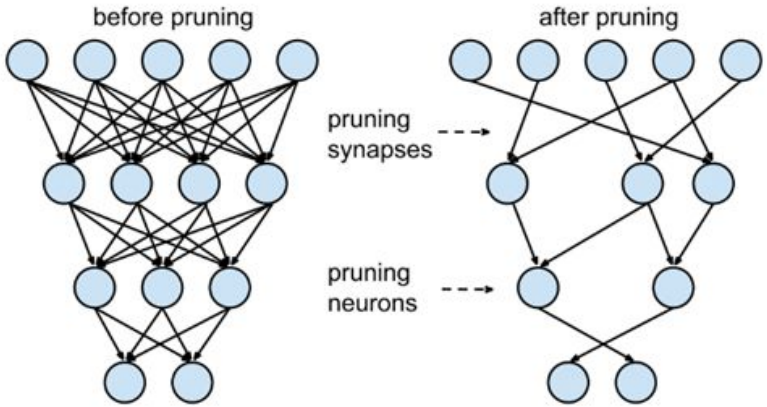
Figure 3: **Example System with Integrated DPU**



Reference:https://www.xilinx.com/support/documentation/ip_documentation/dpu/v3_2/pg338-dpu.pdf

Prune-Make model have less memory

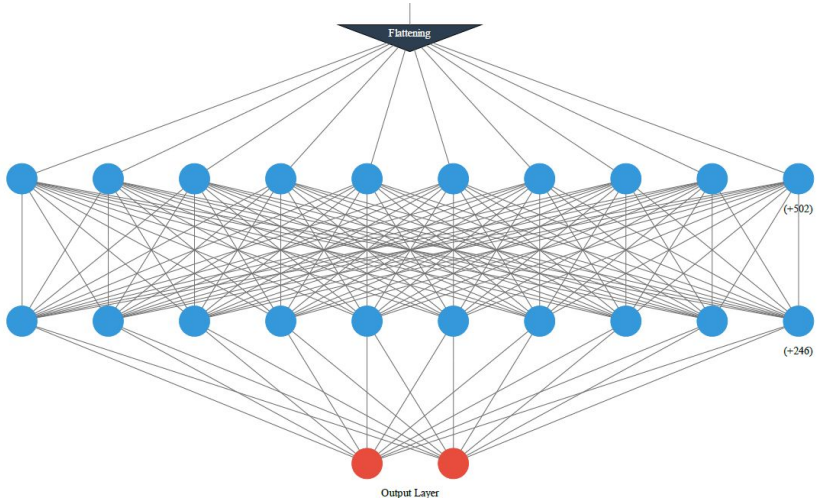
Pruning means eliminating unnecessary values in the weight tensors.



Model visualization

Make model visualization can help us to improve accuracy. We need to learn how to prune in a way that doesn't impact the accuracy this much. This is why we want to visualize the data so we prune in a less impactful way.

ANN visualizer



Improved model by Hyperparameter tuning

- Decreased memory usage and latency by reduce the hidden layer.
- Improve the later on accuracy by finding the best epoch for our model using early stopping
- Improve the later on accuracy by find the best learning rate so far by use the grid search on the model
- Combine the data preprocess data to reduce the filter layer and pool layer that make the latency decrease
- Used grid search to find the optimizer, batch size, kernel size, strip size, activation function and node number of our model.

The Result come out with original latency of 23ms per data to 19ms per data, And current accuracy is 96.73%