Machine Learning for Data Reduction at LCLS-II: A Path Toward 1MHz Detection

Ryan Coffee (PI), Amedeo Perazzo (co-PI), Kunle Olukotun (co-PI), Omar Quijano, Ryan Herbst, Audrey Corbeil Therrien, Abdullah Rashed Ahmed, Tim Aiken, Matt Feldman, and Katherine Fotion

Introduction

The goals of this project were three-fold
- FPGA & GPU-based inference node design
- Standardized interactive development environment
- Optimizing compiler that targets FPGA deployment

An ancillary benefit of this project was that SLAC is now taking a leadership position related to EdgeAI, the detector-side of machine learning application for ultrafast autonomous data compression and machine control. SLAC's use cases are emerging as exemplar cases that are inspiring industry partners in the Edge AI space.

Node Design

We have one of each flavor of node: master, docker build, train/test, camera inference, and waveform inference. **Master** – launching kubernetes pods and services, **Build** – Docker continer images, **EdgeTrain** – V100 training and inference GPU + P40 inference GPU + Xilinx KCU1500 FPGA, **EdgeImage** – 4x quad CXP12 image capture cards + V100 training and inference GPU + 1 dual Abaco digitizer card + Xilinx KCU1500 FPGA, **EdgeSpect** – 4x dual Abaco digitizer cards + Xilinx KCU1500 FPGA + P40 inference GPU

Each of the three Edge nodes have 2TB of local NVMe storage for model updates and eventual re-training.

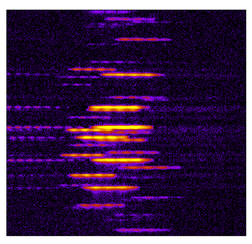


Fig. 1. Edge Imaging (left) and image to be processed (right)

Development & Environment

As per Fig. 2, we foresee FPGA-based pre-processing. Figure 3 shows that dimensional reduction that condenses the information, edge location in this 2D-TimeTool case, makes for more accurate and robust inference. Information dense features also reduce the resource usage for downstream networks of decision trees.

Inference models were developed in a Kubernetes managed cluster. We note that Singularity supports Docker build recipes for use in HPC environments, facilitating simulation-based training set generation.

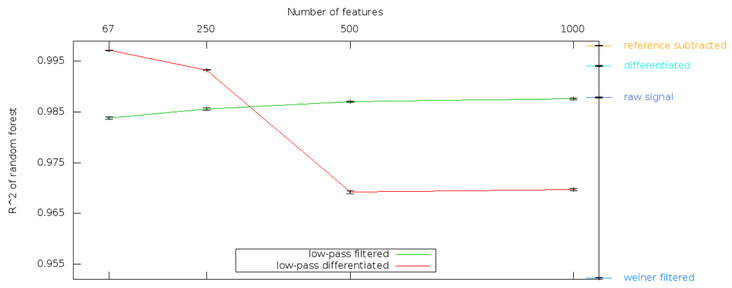


Fig. 3. EdgeImage random forest results for training and accuracy. Curves indicate importance of dimensional reduction via initial pattern matching convolutions in the FPGA. The resulting parameters are inputs to the trained models rather than the raw image input.

Model Deployment

FPGA deployment is made by training models in the traditional python environment with TensorFlow. The fitted model is then compiled to FPGA byte-code via the Spatial language [1], a SCALA-like functional programming language used to define pre-analysis and for model interactions. The deployment results for the EdgeImage case are shown in Fig. 4 and demonstrate that Gradient Boosted Decision Trees minimize the FPGA resource usage (green patch) while maintaining comparably accurate predictions to fully connected neural networks that actually over-run the existing FPGA resources (red patch).

Figure 5 shows the example of a fully connected 3-layer neural network for the EdgeSpect case. Here the prediction of SASE spike number is needed at the first FPGA on the waveform digitizer [2].

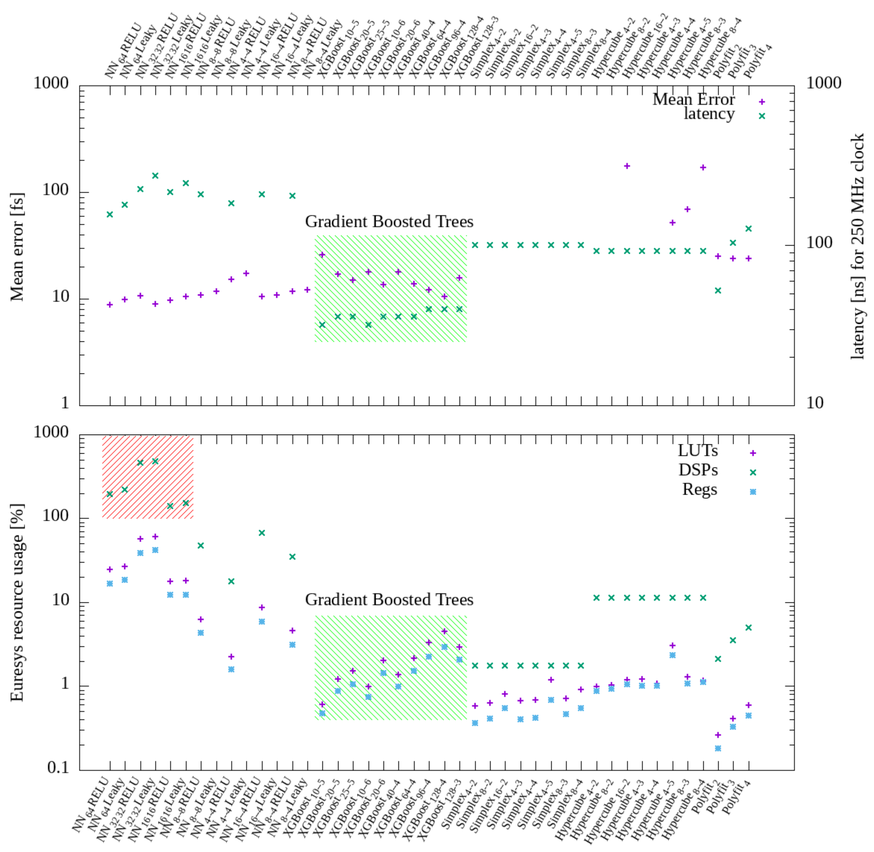


Fig. 4. EdgeImage error and latency (top) and FPGA resource usage relative to native Quand-CXP12 image capture card FPGA (bottom) for compiled models [1]. Inputs to the models are results of the initial convolution against the "differentiation filter" referenced in Fig. 3.

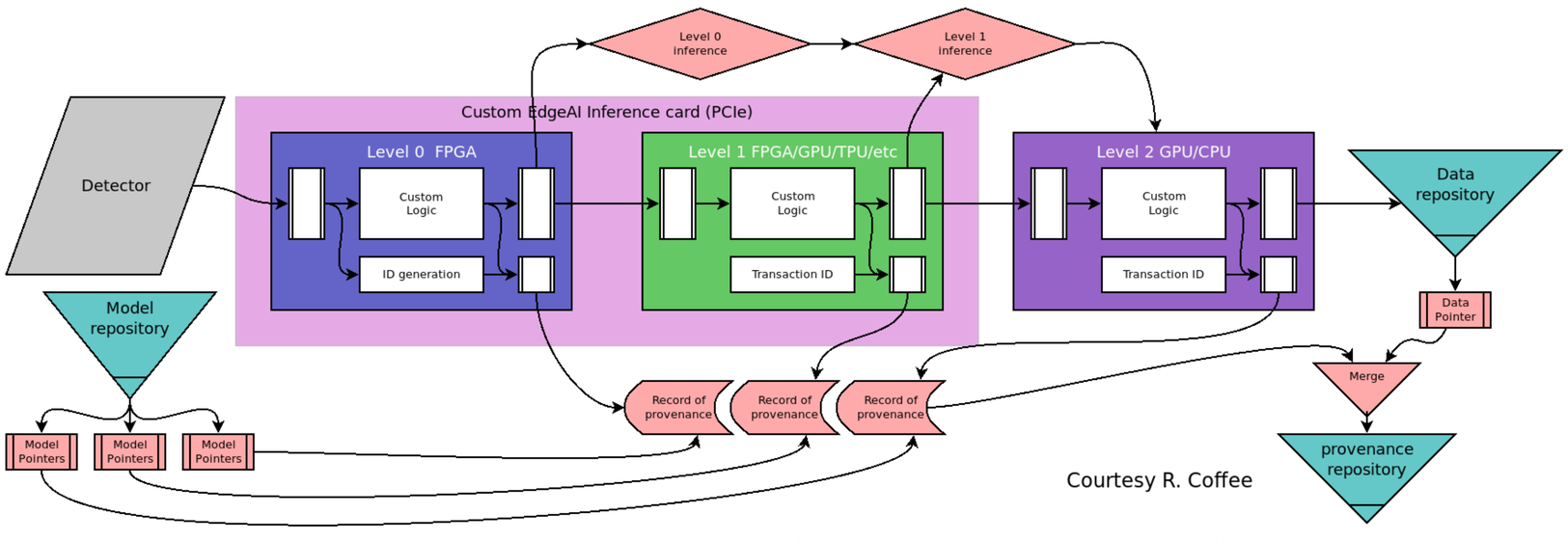
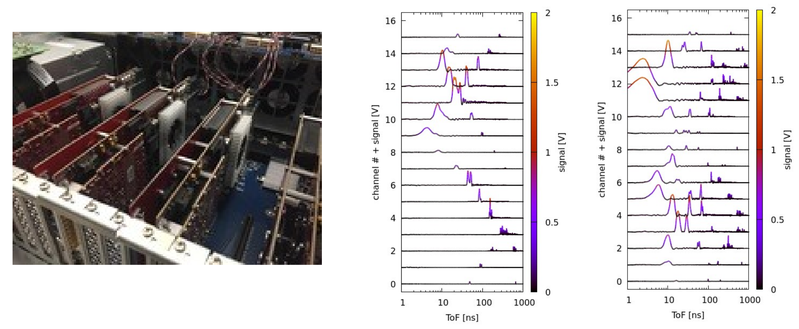


Fig. 2. Current concept for EdgeAI information extraction with provenance record. Currently seeking funding.



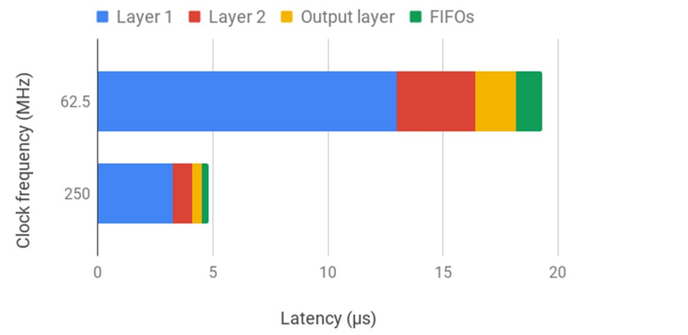


Fig. 5. EdgeSpect (top-left) predicting SASE sub-spikes based on 16 electron Time-of-Flight spectrometers (top-right) with latency for fully connected neural network of 3 layers with RELU activation (bottom) [2].

Conclusions

Three flavors of EdgeAI node. EdgeImage to demo a 100 kFps 2D visible camera in early March. EdgeSpect will demo 100 kFps shortly thereafter. Spread Kubernetes cluster management to LCLS and integrating with ePix family x-ray detector development in TID-AIR.

Project Exposure

A.C.T. – IEEE NSSMIC 2019 [submission], DESY Instrumentation Seminar, BNL-IFDEPS, IEEE Real Time. **R.N.C.** – Global Innovation Forum 2019, Whitehouse OSTP workshop for data handling, NNSS data analysis for nuclear security science, Interview at Next AI Platform 2020, Nvidia GTC2020.

Industry engagement as a result of LDRD Project:
- AI Chip startups – Cornami, SambaNova, Groc – plan to benchmark against our trained models.
- Nvidia Edge R&D project planning to make demonstration case of either the 2dTimeTool or CookieBox example
- VisionResearch (Visible imaging cameras), Euresys (capture cards), Abaco (digitizers) are in discussion about custom design and/or custom FPGA logic.
- TID-AIR ePix family of detectors reaching Mfps based on Edge AI handling at the sensor.

Personnel Development

Omar Quijano – LCLS IT/Networking department head, **Audrey Corbeil Therrien** – Faculty appointment, Univ. of Sherbrooke, Canada, **Matt Feldman** – PhD Candidate Stanford CS/EE, **Katie Fotion** – ML Engineer at Cruise Autonomous Vehicles, **Abdullah Rashed Ahmed** – PhD Candidate at Brown in Computational Neuroscience, **Tim Aiken** – Developer at Gbanga Millform Inc.

Acknowledgments

This LDRD project supported all computing hardware except the CookieBox digitizers (DOE-BES FWP 100498) and partially supported R.N.C., A.R.A., K.F., O.Q., and T.A. A.C.T. was principally supported through a Banting Fellowship, M.F was supported via Stanford CS Graduate Fellowship. R.H is supported through TID-AIR.

[1] Feldman PhD Thesis, in progress
[2] Corbeil Therrien *et al.,* "Machine Learning at the Edge for Ultra High Rate Detectors" IEEE Nucl. Sci. Sym. and Med. Imag. Conf. (2019)