

Live Demonstration: Face Recognition on an Ultra-low Power Event-driven Convolutional Neural Network ASIC

Qian Liu¹, Ole Richter¹, Carsten Nielsen^{1,2}, Sadique Sheik¹, Giacomo Indiveri^{1,2}, and Ning Qiao^{1,2}

¹aiCTX AG, Thurgauerstrasse 40, CH-8050 Zurich, Switzerland

²Univeristy of Zurich, Zurich, Switzerland

Abstract

We demonstrate an event-driven Deep Learning (DL) hardware software ecosystem. The user-friendly software tools port models from Keras (popular machine learning libraries), automatically convert DL models to Spiking equivalents, i.e. Spiking Convolutional Neural Networks (SCNNs) and run spiking simulations of the converted models on the hardware emulator for testing and prototyping. More importantly, the software ports the converted models onto a novel, ultra-low power, real-time, event-driven ASIC SCNN Chip: DynapCNN. An interactive demonstration of a real-time face recognition system built using the above pipeline is shown as an example.

1. Introduction

There has been a tremendous growth in our understanding and use of SCNNs in the recent past [3, 4]. This has enabled us to make a significant leap in our computational abilities with Spiking Neural Networks (SNNs). SNNs also enable us to harness the power efficiency of neuromorphic engineering to build ultra-low power neural network systems. Integrating these two aspects, we demonstrate a novel real-time neuromorphic hardware software eco-system for simulating and emulating SCNNs.

We demonstrate various components of this eco-system, including the user friendly software framework - SINABS for developing and simulating SCNNs. As an application demonstration we will show how a simple 6 layer Convolutional Neural Network (CNN) trained to recognize faces of several people can be converted to its spiking equivalent SCNN. We will then show how this model processes input from an onsite Dynamic Vision Sensor (DVS) [1] and predict the identity of the person in front of the camera. We will also port the same model onto our DynapCNN and show the output and power consumption of the chip in real-time.

2. SINABS: Open-source Python library

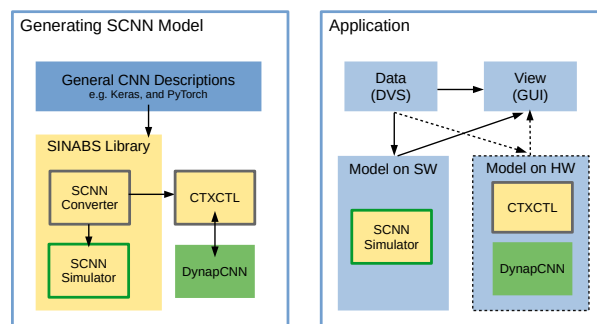


Figure 1. The architecture of the HW-SW ecosystem. Left: the process of generating an SCNN model onto a HW/SW simulator; Right: the components to run any SCNN-based applications.

SINABS library provides a smooth transition from Deep Neural Networks (DNNs) to SCNNs implementation. It allows simulation of multilayer-SCNNs. The SCNNs simulations allow emulation and validation of models in inference mode. The library also enables simulations at arbitrary fixed precision. The library is based on PyTorch and all the layers are fully compatible with all PyTorch modules. So all PyTorch functionality such as simulating on GPUs is retained. In addition, one of the core functionality of this library is to enable porting of models defined and developed in Keras into equivalent SCNNs.

Finally and most crucially, CTXCTL is a software framework aimed at facilitating the control and execution of experiments using neuromorphic hardware communicating through Address-Event Representation (AER). CTXCTL also provides visualization tools for efficient debugging and network design. CTXCTL driver module enables porting the developed models from Python onto an ultra-low power and latency DynapCNN chip. Figure 1 shows a block diagram of the various modules of the hardware software ecosystem and how they interact with each other.

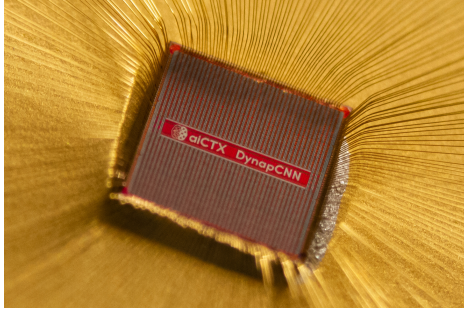


Figure 2. Photograph of the DynapCNN

3. DynapCNN: Ultra-low power event-driven CNN Processor

DynapCNN (Figure 2) is a scalable, fully-configurable digital event-driven neuromorphic processor with 1M spiking Integrate and Fire neurons per chip. The chip convolves the visual event stream in a sparse fashion by heavily utilizing in-memory computation for implementing SCNNs. It provides a dedicated interface for several DVSS; and allows direct input of event streams of DVS, DAVIS [1] and ATIS [2] camera series, enabling low latency, direct input with a data preprocessing pipeline. The chip supports various types of CNN layers (like ReLU, Cropping, Padding and Pooling) and network models (like LeNet, ResNet and Inception). The chip itself supports infinitely deep networks via multiple daisy chained devices.

4. Demonstration Setup

Figure 1 (right) illustrates the components required to run an event-based vision application on our proposed system. For the face recognition demonstration, the GUI (see Figures 3 and 4) displays the raw input from a DVS on the top-left panel, the low pass-filtered output spike count in time at the bottom-left. On the top-right panel, a picture of the recognized person is presented. Moreover, the power estimation of running the same SCNN model on a DynapCNN in real time is illustrated on the bottom-right panel. For software simulations, the power is estimated by monitoring the Synaptic Operations per second (SOps) during the demonstration. For software simulations, the setup consists of a DVS to a laptop via USB (Figure 3); while for running on a DynapCNN (Figure 4), a DVS directly connects to a PCB of DynapCNNs, and the CTXCTL and a demo UI running on a laptop reads the output spikes and displays on the screen. The face recognition application serves as a proof-of-concept to demonstrate a DVS-based vision model running on our hardware-software ecosystem. The network consists of three convolutional and two dense layers, having 40K parameters in total. The peak SOps of the network is around 200M, thus running at an estimated average power



Figure 3. The demonstration setup involving a DVS sensor providing input to the SINABS to simulate a multi-layer SCNN.



Figure 4. In this demonstration setup, a DynapCNN receives input directly from a DVS.

in sub mW range on the DynapCNN. The recognition performance of 98% was measured on a recorded DVS dataset and is validated with SCNN software simulations.

5. Visitor Experience

The visitors can interact with the demonstration through the DVS sensor. A monitor will display the sensor activity, where they can see themselves and the response of the network to their presence. They can also experience and test the network under varying light conditions.

References

- [1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 db $3 \mu s$ latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.
- [2] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. An asynchronous time-based image sensor. In *2008 IEEE International Symposium on Circuits and Systems*, pages 2130–2133. IEEE, 2008.
- [3] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.
- [4] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13, 2019.