Live Demonstration: End-to-end Event-driven Sensing and Computing for Body Gesture Recognition on an Ultra Low Power Neuromorphic SoC

Yannan Xing, Yuhang Liu, Nogay Kuepelioglu, Felix Bauer, Kai Wang, Yalun Hu, Gregor Lenz, Yudi Ren, Carsten Nielsen, Tugba Demirci, Ning Qiao, Sadique Sheik
SynSense AG, Co Ltd.
{yannan.xing, sadique.sheik}@synsense.ai

Abstract

SPECK is the world's first neuromorphic system-on-chip that combines an event camera with a neuromorphic SNN processor as a single ASIC. We demonstrate a fully event-driven gesture recognition application using SPECK to control a game, using a dynamic power budget of only 5.5 mW. We also show how our user-friendly software toolchain enables the efficient development, simulation, and deployment of spiking convolutional neural networks on SPECK.

1. Introduction

As machine learning models become more capable, there is an increasing demand to deploy them to power-constrained edge devices such as smart sensors that can operate in real-time. Event-driven sensors such as event cameras have emerged as a promising low-power technology to capture visual data in an efficient manner. They exploit asynchronous circuits to capture change in illumination rather than absolute grey-level values. This technology is a natural fit for battery-powered, always-on applications because the generated amount of data remains close to zero when nothing is happening and only increases when there is activity in the visual scene. We pair this sensor technology with spiking convolutional neural networks on SPECK. Convolutional neural networks (CNN) have demonstrated human-level performance in many computer vision tasks. In the context of event event-based sensing, we combine the convolutional connectivity between layers with a spiking neuron model that can integrate information over time, such that the events that signify changes in illumination can be accumulated for a prediction. Each neuron's output is zero or one and on average has a high temporal and spatial sparsity. We call these networks spiking convolutional networks (SCNN).

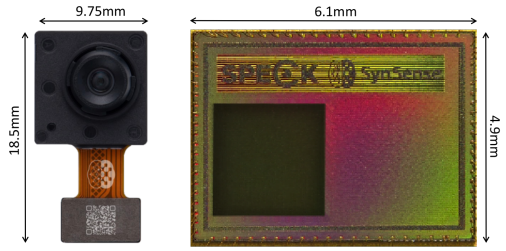

Figure 1. SPECK SoC. Left: SPECK module; Right: SPECK chip die with event camera sensor array

2. SPECK system-on-chip

Figure 1 shows the SPECK system-on-chip (SoC) and a detail of the die, which integrates an event camera and an SCNN ASIC. The former has a spatial resolution of 128 by 128 pixels and the latter uses digital asynchronous technology to couple processing of events and therefore power consumption directly to event camera output. It has 9 neuromorphic cores that each combine convolutional connectivity, spiking neurons and a pooling layer. The total memory of 592 KB is distributed in such as way that some layers account for more neurons, whereas other cores can have higher connectivity. Models can be deployed freely across any number of cores.

3. Software toolchain

Event data is radically different from traditional frame-based image or video formats. In particular the high temporal resolution requires specialized data wrangling tools to produce representations that can be processed efficiently on conventional machine learning hardware. To ease the transition from traditional machine learning to SCNN models tailored for SPECK, we developed a software toolchain which is detailed in Fig. 2. It encompasses data wrangling (Tonic [2]), model training (Sinabs) and hardware deployment (Samna). Sinabs is a deep learning framework based on PyTorch that focuses on simplicity and speed and en-

ables quick development iterations and fast training of spiking neural network models. It features the EXODUS algorithm [1], which is an efficient, GPU-accelerated implementation of backpropagation-through-time. Samna provides a medium-level Python API to program SPECK and deploy models. It supports configuration of the event camera, custom low-level filters, monitoring functionalities as well as the construction of arbitrary computational graphs for on-chip inference.

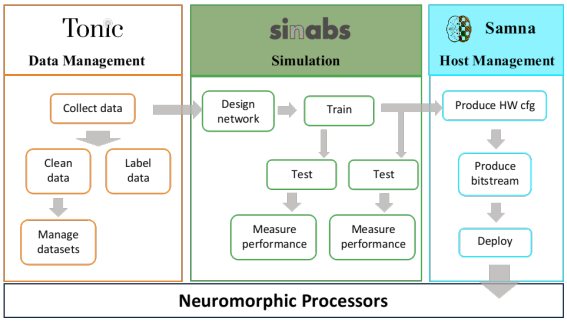

Figure 2. SPECK toolchain: Including tonic for data management, sinabs for SNN development and samna for deployment

4. Visitor experience: Gesture controlled game

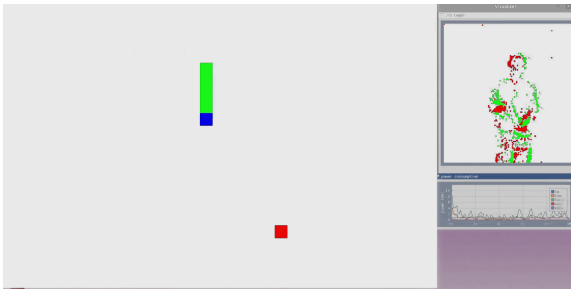

Figure 3. Snake game graphical user interface

For the live demonstration we chose an interactive task where the user is able to control the directions of a snake in a game using their hand and body gestures. The goal is to control the snake to eat food that randomly appears on the screen, which gets harder over time as the snake grows with every bite of food. The graphical user interface (GUI) shown in Figure 3 combines the actual game display, a live rendering of event camera output as well as a graph of dynamic and static power consumption metrics over time. The control gestures include start, pause, turn left, turn right, accelerate or slow down. Using a total of approximately 1 hour worth of training data, we train a 27.4K parameter SCNN model and achieve 98.6% test accuracy for the gesture classification task (See Table 1.). 2 out of 10 recorded subjects are reserved for the test set. The model occupies

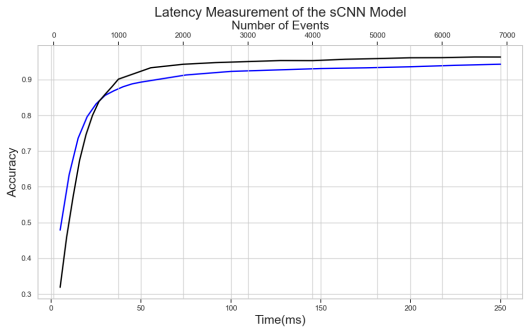

Figure 4. The accuracy dynamics for gesture recognition along with the number of input events and time to SPECK

4 out of 9 cores on SPECK. For power measurements, we sample the power rail with 100 Hz to obtain estimates for dynamic and static power. While performing the gestures, we observe a mean static power consumption of 1.67 mW and mean dynamic power consumption of 5.55 mW. The latency is measured in number of events that need to be fed or the time it takes to trigger a prediction. We show the trade-off curve in Figure 4 and achieve the reported accuracy at a latency of about 1.5k events or 50 ms. It is worth noting that SPECK actually processes individual event in sub-μs latency, but for a complete prediction, a reasonable accumulation time is needed for sufficient data to be presented. Lastly, we would also like to highlight that both sensing and processing happen exclusively on chip and do not need any post-processing on the PC that displays SPECK output inside the GUI and controls the game mechanics.

Our demo shows that it has become very straightforward and efficient to train a spiking neural network and deploy it on neuromorphic hardware from scratch using our software and hardware stack within a span of a few hours.

Metric	Value
Accuracy	98.6%
Categories	8
Static Power(mW)	$\sim$ 1.671
Dynamic Power(mW)	$\sim$ 5.553
Latency(N_{events})	$\sim$ 1.5k / 50 ms

Table 1. Performance of Gesture Recognition running on SPECK

References

[1] Felix C. Bauer, Gregor Lenz, Saeid Haghighatshoar, and Sadique Sheik. Exodus: Stable and efficient training of spiking neural networks. *Frontiers in Neuroscience*, 17, 2023.

[2] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., July 2021. Documentation available under https://tonic.readthedocs.io.