Comparison of Convolutional and Spiking Neural Networks on MNIST Using the BrainChip Akida Processor

Alex Clunan, Bobby Downey
Department of Electrical and Computer Engineering
ECE 4380 / ECE 6380 — AI Hardware
Fall 2025

***Abstract*—This work presents a comparative evaluation of Convolutional Neural Networks (CNNs) and Spiking Neural Networks (SNNs) on the MNIST handwritten digit classification task. A conventional CNN is executed on a CPU, while a converted SNN is deployed on the BrainChip Akida neuromorphic processor. We evaluate classification accuracy, inference latency, throughput, and energy efficiency across batch sizes of 1 and 256. Results show that while CPU-based CNNs achieve slightly higher accuracy and throughput, the Akida SNN demonstrates orders-of-magnitude improvements in energy efficiency, highlighting the advantages of neuromorphic hardware for low-power edge inference.**

I. Introduction

Convolutional Neural Networks (CNNs) are widely used for image classification due to their strong representational power and high accuracy. However, CNN inference relies on dense, continuous-valued computations that incur significant energy cost, particularly when deployed on general-purpose processors. This energy inefficiency limits their suitability for power-constrained edge devices.

Spiking Neural Networks (SNNs) offer an alternative computation method inspired by biological neural systems. By using event-driven, spike-based computation, SNNs can reduce redundant operations and improve energy efficiency. Neuromorphic processors such as the BrainChip Akida are specifically designed to take advantage of these characteristics.

In this project, we compare CNN and SNN implementations of an MNIST digit classifier. The CNN is executed on a CPU, while the SNN is derived via quantization and conversion and deployed on the Akida Development Kit. We analyze performance trade-offs in accuracy, latency, throughput, and energy consumption.

II. Background

CNNs perform inference using multiply–accumulate operations over dense tensors, resulting in high computational throughput but also high power consumption. In contrast, SNNs encode information as discrete spike events and compute only when spikes occur. This sparse, event-driven behavior allows neuromorphic systems to reduce energy usage substantially.

The BrainChip Akida platform supports deployment of SNNs converted from trained CNNs through quantization and mapping tools. Akida executes inference asynchronously, enabling efficient processing for edge AI workloads.

TABLE I: Performance comparison between CPU CNN and Akida SNN.

Platform	Batch	Acc. (%)	Thrpt. (img/s)	Energy / Img
CPU	256	98.42	56,824	1.76 mJ
CPU	1	98.42	2,651	13.98 mJ
Akida	256	96.01	4,455	0.24 mJ
Akida	1	96.01	1,489	0.44 mJ

III. Methodology

A. Dataset

The MNIST dataset consists of 60,000 training images and 10,000 test images of handwritten digits. Each image is grayscale with a resolution of 28×28 pixels.

B. CNN Model and Conversion Process

The CNN was implemented in TensorFlow/Keras with three convolutional layers followed by a fully connected output layer. The training process used the Adam optimizer for five epochs. To deploy on the Akida hardware, the trained CNN underwent the following stages:

1) **Quantization:** The CNN weights and activations were quantized to 4-bit precision to reduce model size and prepare it for conversion to an SNN.
2) **Conversion to SNN:** Using the BrainChip Akida SDK, the quantized CNN was converted into a spike-based network compatible with Akida's neuromorphic architecture.
3) **Mapping and Deployment:** The converted SNN was mapped to the Akida Development Kit, enabling inference with on-board power monitoring.
4) **Benchmarking:** Inference performance was measured for batch sizes of 1 and 256, including throughput, per-image latency, classification accuracy, and energy consumption.

IV. Experimental Results

A. Overall Performance

Table I summarizes the primary performance metrics for CPU and Akida inference.

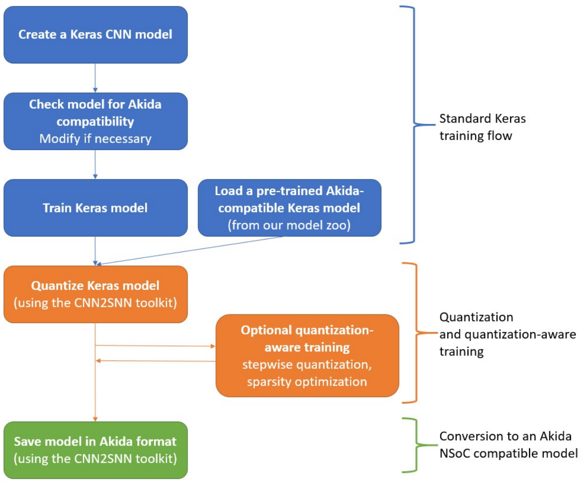
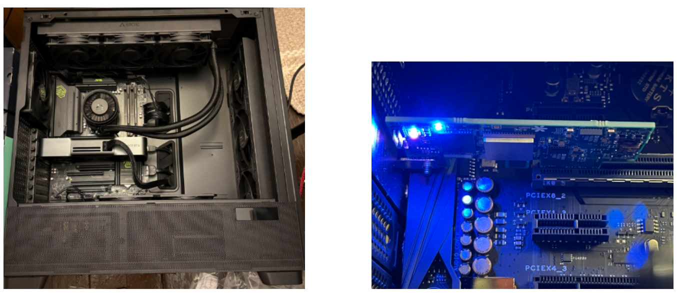


Fig. 1: CNN-to-SNN conversion pipeline using the Akida SDK (scaled to fit one column).



(a) Desktop system

(b) Akida PCIe board

Fig. 2: Hardware used for Akida deployment and benchmarking.

B. CPU vs. Akida Ratios

For batch size 256, the CPU provides a 12.75× throughput advantage, while Akida achieves a 7.34× improvement in energy efficiency per image. For batch size 1, Akida's energy advantage increases to over 31×, demonstrating the benefits of event-driven computation for low-latency inference.

V. Hardware Setup Challenges

Deploying the Akida Development Board posed several challenges. Initial attempts using a modern desktop system with an AMD 9950X3D CPU failed due to Linux kernel incompatibilities, as Akida drivers support only Linux kernel version 5. Virtualization approaches using Proxmox and Ubuntu 20.04 suffered from unstable PCIe passthrough.

An older Intel Core i7-8700K system in the Rice 240 CPE lab was ultimately successfully used with a Proxmox-hosted Ubuntu 20.04 virtual machine.

A. A. Latency Percentiles

These plots show the distribution of inference latency. P99 indicates worst-case latency, P90 shows near-worst-case, and P50 shows the median. Akida maintains relatively low worst-case latency due to event-driven computation.

B. B. Power and Energy

Akida demonstrates significant reductions in energy per image compared to CPU, especially at low batch sizes, while CPU consumes higher average power to achieve higher throughput.

C. C. Latency and Throughput

Increasing the batch size reduces latency because the fixed overhead is spread across more inputs. On CPUs, larger batches also greatly increase throughput, whereas the Akida processor keeps latency low even with moderate throughput.

D. D. Pareto Trade-Off

We compare throughput versus energy per image to illustrate the trade-offs between performance and efficiency. Each point represents a batch size and platform.

Akida points lie in the lower-left region of the Pareto plot, showing substantial energy efficiency at moderate throughput. CPU points achieve higher throughput but with much higher energy costs.

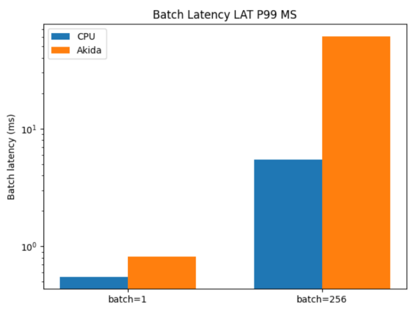
VI. Conclusion

SNN inference on the BrainChip Akida processor achieves substantial energy efficiency gains compared to CPU-based CNN inference, while maintaining reasonable classification accuracy on MNIST. CPUs provide higher throughput and slightly better accuracy, but their energy consumption is orders of magnitude higher.

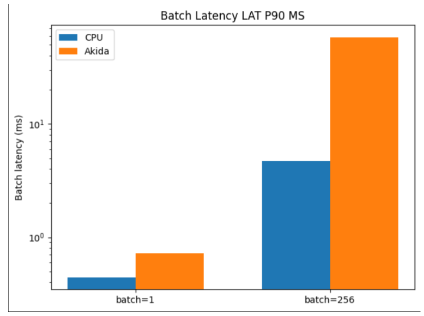
These results indicate that neuromorphic hardware is well-suited for low-power edge AI applications. Deployment remains sensitive to software and hardware compatibility constraints. Future work includes exploring temporal encoding, Akida Edge Learning, and scaling to more complex datasets.

References

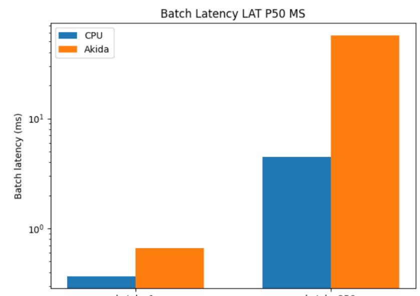
[1] BrainChip, "Akida Developer Documentation," https://brainchip.com/developer.
[2] P. U. Diehl and M. Cook, "Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity," *Frontiers in Computational Neuroscience*, 2015.
[3] BrainChip, "AKD1500 Product Brief v2.5," Nov. 2025.



(a) P99

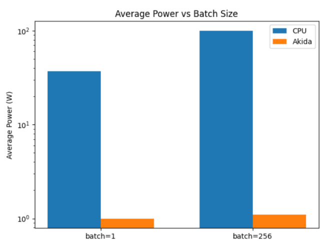


(b) P90

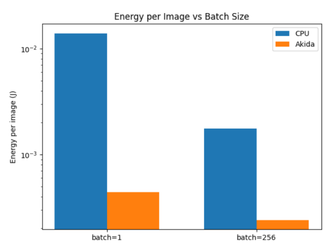


(c) P50

Fig. 3: Latency percentiles for CPU and Akida inference.

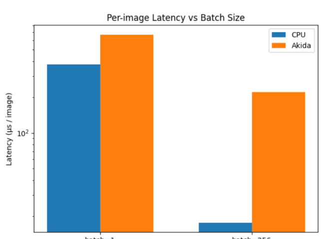


(a) Average power

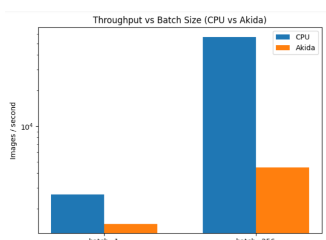


(b) Energy per image

Fig. 4: Power and energy comparison across batch sizes.



(a) Per-image latency



(b) Throughput

Fig. 5: Latency and throughput versus batch size.

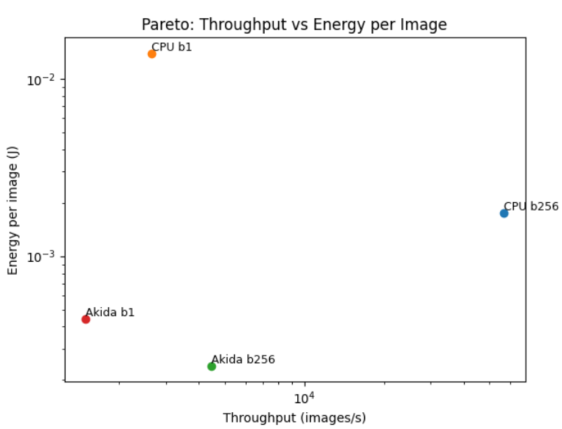


Fig. 6: Pareto trade-off between throughput and energy per image.