Ultra-Lightweight Edge Intelligence Using Vector Symbolic Architecture

Chae Young Lee

Stanford University
Stanford, USA
chae@stanford.edu

ABSTRACT

Edge intelligence greatly benefits Internet-of-Things devices by providing insights at the sensor node without transmitting the data to and from the cloud. Although recent work has scaled deep neural networks down to microcontrollers, these solutions remain impractical for devices operating under strict energy budgets—such as battery-powered camera traps in remote forests or micro-robots in disaster zones. My work addresses that gap by exploring a brain-inspired Vector Symbolic Architecture (VSA) over conventional neural networks. I introduce *HyperCam* for low-power image classification, *NavHD* for micro-robot navigation, and *Omen* for dynamic optimization during inference. These methods each optimize VSA's encoding and training algorithms to achieve meaningful speedups and energy savings. Ongoing work seeks to demonstrate these advances as real-world systems, where VSA can extend the lifespan of intelligent systems in remote or energy-constrained environments.

ACM Reference Format:
Chae Young Lee. 2025. Ultra-Lightweight Edge Intelligence Using Vector Symbolic Architecture. In *The 23rd Annual International Conference on Mobile Systems, Applications and Services (MobiSys' 25), June 23–27, 2025, Anaheim, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3711875.3736678

1 INTRODUCTION

TinyML is emerging as a means to run Machine Learning (ML) models directly on embedded devices, such as microcontrollers (MCUs). By processing data on-device rather than sending it to a gateway or cloud server, TinyML systems conserve time, bandwidth, energy, and privacy. Recent efforts have shown that deep neural networks (DNNs) can be scaled down to mobile devices and smaller MCUs [5]. Yet, these solutions often remain impractical for highly resource-limited systems such as those powered by batteries or deployed in remote areas. Even if more memory or faster clock speeds are available, leveraging them to boost DNN performance can quickly deplete limited energy reserves.

Consider, for instance, a fleet of wireless camera traps deployed in remote forests. Onboard intelligence can discard non-wildlife images and thereby greatly conserve transmission energy. Meanwhile, micro-robots rely on ML for real-time navigation. Both systems must carefully manage power: camera traps use most of their energy on image transmission, while micro-robots expend it on locomotion. As a result, they require an efficient, low-power ML solution that goes beyond typical TinyML constraints and addresses the strict energy budgets of these challenging scenarios.

My work addresses this gap by developing ultra-lightweight intelligence for resource-limited hardware. Specifically, it explores a neuromorphic learning paradigm called Vector Symbolic Architecture (VSA), which shows promising performance and resource efficiency. VSA represents information as vectors and processes it using simple operations (e.g., bitwise exclusive-or or addition), performing classification through vector-distance comparisons. This inherent simplicity and hardware-friendliness enable highly efficient resource usage. Recent studies have shown that VSA methods can match DNN-level accuracy on tasks such as behavioral classification and simple image recognition while drastically reducing computational overhead [1, 2].

Building on these strengths, my research adapts VSA for various ML tasks on severely resource-constrained MCUs. For instance, *HyperCam* [4] enables low-power image classification of 120x160 grayscale images using about 50 KB of flash memory and 20 KB of RAM. *NavHD* [3] achieves VSA-based off-policy reinforcement learning for micro-robot navigation using only 10 KB of flash memory. Additionally, both models' on-device inference time can be significantly reduced with *Omen* [6], which supports early termination of vector encoding. Moving forward, I plan to apply these techniques in real-world settings and demonstrate long-term deployments of inexpensive, intelligent IoT platforms at scale.

MobiSys' 25, June 23–27, 2025, Anaheim, CA, USA

ACM ISBN 979-8-4007-1453-5/25/06
https://doi.org/10.1145/3711875.3736678

2 VSA-BASED EDGE ML SYSTEMS

VSA offers a lightweight, hardware-friendly approach to ML. However, naive VSA implementations can still demand significant time and memory for encoding input data into vectors. To address this challenge, I developed novel encoding and training algorithms tailored for resource-constrained hardware. These optimizations substantially reduce the overhead associated with VSA-based methods and are validated through real-world systems running on low-power MCUs.

Image classification. A naive VSA encoder takes several minutes to encode one 120x160 grayscale image on an STM32-U585AI MCU. It also uses about a megabyte of flash memory. HyperCam [4] reduces the encoding time 72x by leveraging the sparsity of vectors. Using ideas from Bloom Filter and Count Sketch, HyperCam approximates the bitwise majority voting of a large number of hypervectors, which consumes the most time in encoding. Furthermore, HyperCam drastically reduces the encoding memory requirement 30x by dynamically computing the mapping instead of storing the precomputed ones. As a result, HyperCam not only outperforms existing neural networks but also lightweight ML models and existing HD classifiers in resource efficiency while achieving above 90% accuracy for MNIST, Fashion MNIST, and face detection on 120x160 grayscale images.

Model	Accuracy	Memory	Latency
`MobileNet V3`	88.18%	1640.00 KB	18.53 s
`MCUNet V3`	**99.88%**	1190.00 KB	6.70 s
Ours (Count Sketch)	92.98%	53.83 KB	0.27 s
Ours (Bloom filter)	92.73%	**42.91 KB**	**0.12 s**

Table 1: Performance on face detection.

Reinforcement Learning. NavHD [3] achieves off-policy Reinforcement Learning for motor controls of micro-robots using only 10.2 KB of flash memory. Per inference, it consumes 5.6 μs and 1.1 mJ of energy. NavHD uses a novel VSA Q-value function and an adaptive encoder that learns the spatial information from arbitrary sensor positions. NavHD is both fully differentiable and exchangeable, and therefore is subject to both learning-based updates and dynamic optimization. With these features, NavHD outperforms DNN and existing VSA methods in obstacle avoidance by more than 2x in learning quality in imitation learning and zero-shot real-world experiments. Moreover, it outperforms baseline by 2-26x in resource efficiency, as shown in Table 2.

Model	Memory (kB)	Clock Cycle (k)	Energy (mJ)
DNN	263.8	2.9	3.4
VSA	19.5	8.4	9.9
Ours	**10.2**	**0.9**	**1.1**

Table 2: Resource usage on the target hardware.

On-Device Dynamic Optimization Further optimization methods were studied to reduce computation time. Omen [6] uses inferential statistics to early-terminate VSA encoding. Omen proves that existing VSA models satisfy exchangeability, which stems from equal distributedness of information across vectors. Using it, Omen reframes VSA computations as statistical sampling tasks and uses statistical tests to early terminate encoding of long vectors. As a result, Omen achieves 7-12x speedup while causing only 0.0-0.7% accuracy drop in 19 benchmarks. Furthermore, the statistical tests in Omen can be optimized using pre-computed distance lookup, and Omen can be easily deployed on embedded hardware to further optimize existing work [3, 4].

3 FUTURE WORK

One immediate direction is deploying a self-sustaining wireless camera trap in remote forests. By integrating *HyperCam* [4] to identify wildlife and transmit only critical images via satellite backhaul, the platform can operate continuously by harvesting ambient energy. Such a setup aims to run for months with minimal human intervention.

Another research thrust involves on-device training. Unlike deep neural networks, which typically consume significantly more resources during training than inference, VSA models employ the same mechanisms for both. Consequently, a VSA model that performs on-device inference can train on-device without additional overhead. I plan to explore this feature in realistic contexts, such as adapting to terrain changes for navigation or adding new classes in classification.

REFERENCES

[1] Shijin Duan, Yejia Liu, Shaolei Ren, and Xiaolin Xu. 2022. LeHDC: learning-based hyperdimensional computing classifier. In *Proceedings of the 59th ACM/IEEE Design Automation Conference* (San Francisco, California) *(DAC '22)*. Association for Computing Machinery, New York, NY, USA, 1111–1116. https://doi.org/10.1145/3489517.3530593

[2] Shijin Duan, Xiaolin Xu, and Shaolei Ren. 2022. A Brain-Inspired Low-Dimensional Computing Classifier for Inference on Tiny Devices. arXiv:2203.04894 [cs.LG] https://arxiv.org/abs/2203.04894

[3] Chae Young Lee, Sara Achour, and Zerina Kapetanovic. 2025. Low-Power Neuromorphic Learning for Micro-Robotic Controls in the Wild. Under Review.

[4] Chae Young Lee, Pu (Luke) Yi, Maxwell Fite, Tejus Rao, Sara Achour, and Zerina Kapetanovic. 2025. HyperCam: Low-Power Onboard Computer Vision for IoT Cameras. In *Proceedings of the 31st Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, Hong Kong.

[5] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. 2020. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems* 33 (2020), 11711–11722.

[6] Pu (Luke) Yi, Yifan Yang, Chae Young Lee, and Sara Achour. 2025. Early Termination for Hyperdimensional Computing Using Inferential Statistics. In *Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM, Rotterdam, The Netherlands, 342–360.