HyperCam: Low-Power Onboard Computer Vision for IoT Cameras

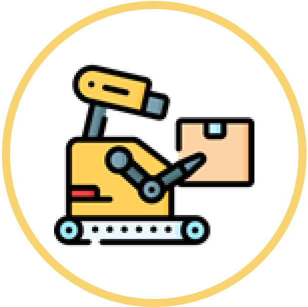
Chae Young Lee[1], Luke (Pu) Yi[1], Maxwell Fite[2],
Tejus Rao[2], Sara Achour[1,2], Zerina Kapetanovic[2]

1 Department of Computer Science, Stanford University
2 Department of Electrical Engineering, Stanford University

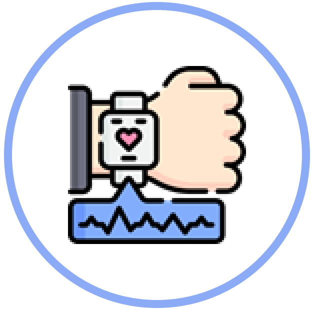
Stanford
SMART SENSING
SYSTEMS LAB

Today’s sensing systems

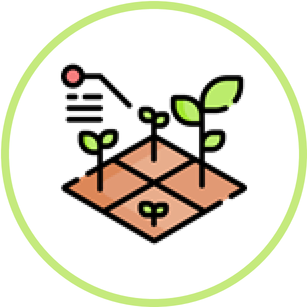
Supply Chain



Healthcare



Agriculture



These sensing systems rely on huge amounts of data to perform tasks and deliver insights

Images

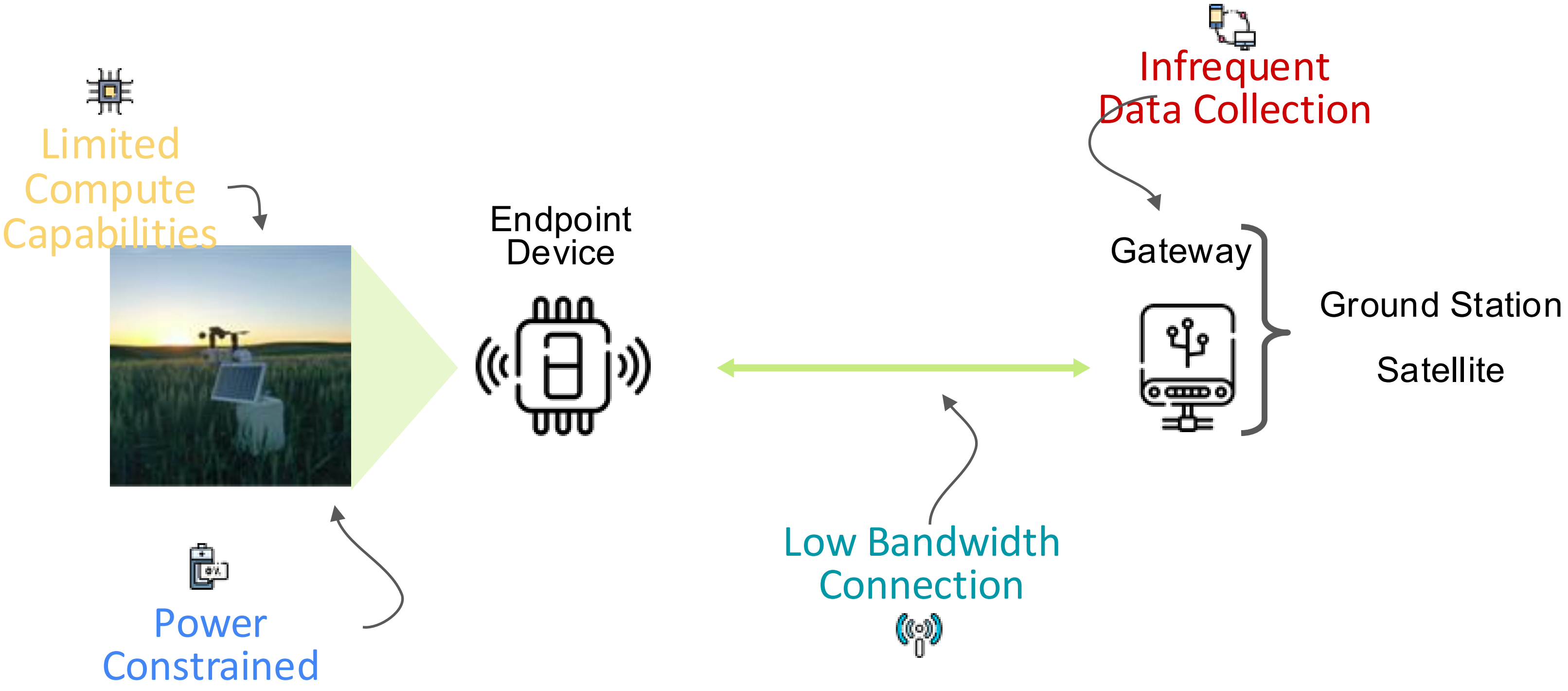
RF

Audio

Temperature

Moisture

Offloading data can be challenging

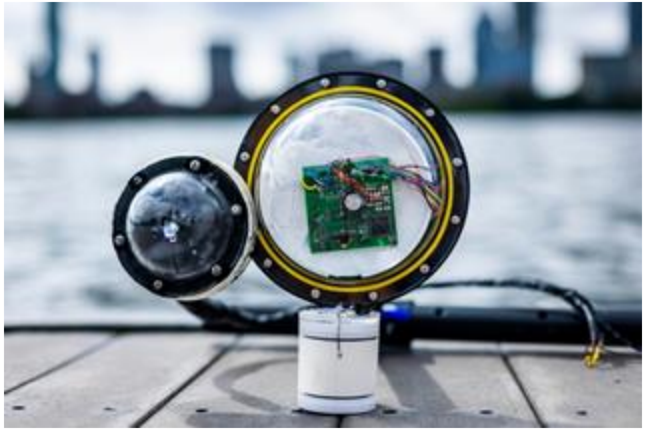


Extremely energy-efficient devices

BeetleCam[1]

WISPCam[2]

Underwater Wireless Camera[3]

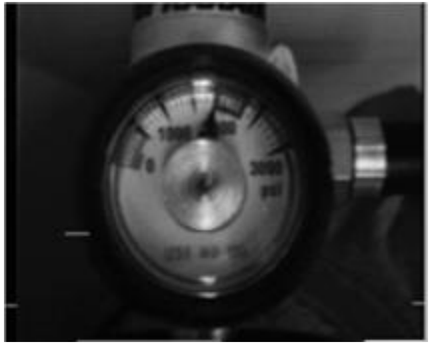


[1] Iyer, Vikram, et al. "Wireless steerable vision for live insects and insect-scale robots." *Science robotics* 5.44 (2020): eabb0839.
[2] Naderiparizi, Saman, et al. "WISPCam: A battery-free RFID camera." *2015 IEEE International Conference on RFID (RFID)*. IEEE, 2015.
[3] Afzal, Sayed Saad, et al. "Battery-free wireless imaging of underwater environments." *Nature communications* 13.1 (2022): 5546.

Some challenges

Battery-free
Wireless Camera

Image Transfer
Delays



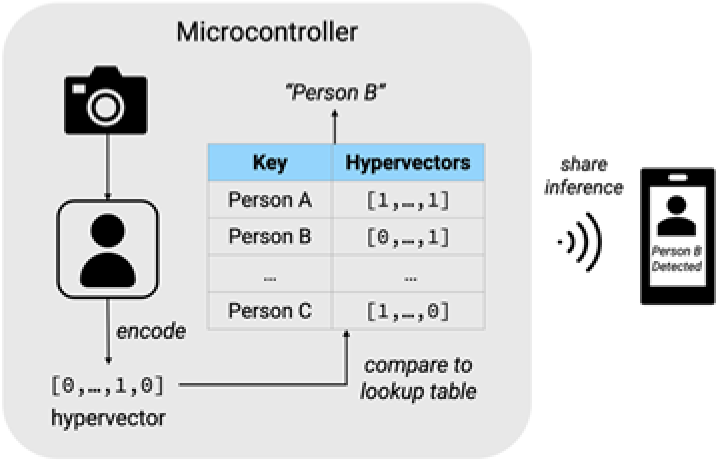
Missing Pixels

Can we enable onboard computer vision on **resource-constrained** IoT cameras?

HyperCam in a nutshell

Onboard ML classifier built on hyperdimensional computing principles.

- Comparable accuracy as DNNs but **55-398x faster** and **12-33x more lightweight** than DNNs.
- Proposes a novel hyperdimensional encoder that optimizes memory and time usage.



Lee et al., "HyperCam: Low-Power Onboard Computer Vision for IoT Cameras," MobiCom, 2025.

Hyperdimensional Computing (HDC)

Motivated by the observation that human brain operates on **high-dimensional** representation of data.

A paradigm of computing in **hyperspace** using **hypervectors**.

$$\mathbf{h}_a = [0,1,1,0,1,\ldots\ldots\ldots\ldots,1,0,1]$$

$$\mathbf{h}_b = [1,1,0,0,1,\ldots\ldots\ldots\ldots,1,1,1]$$

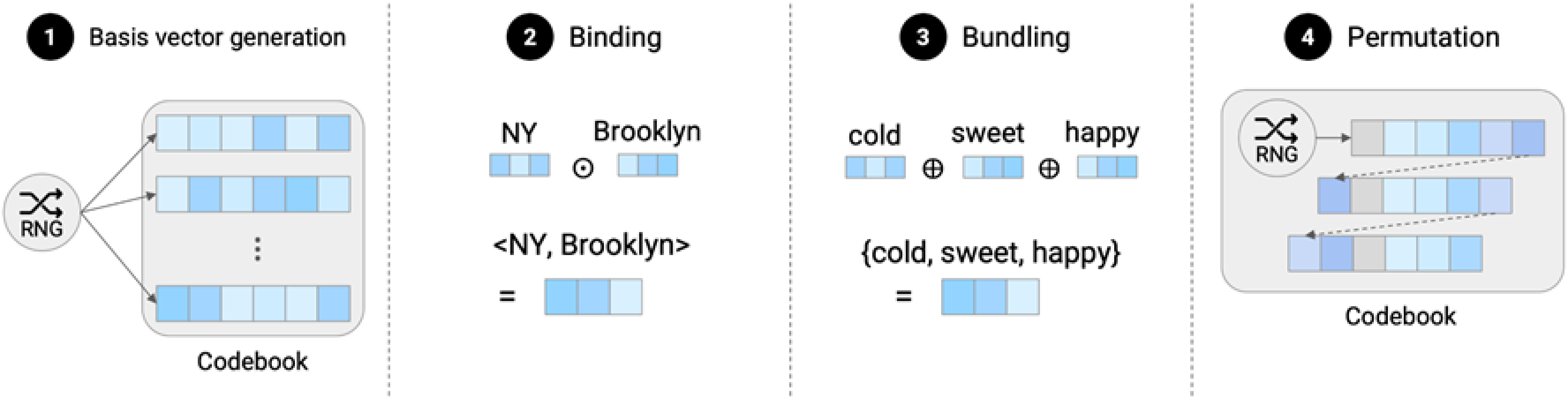
$$\vdots$$

$$\mathbf{h}_z = [0,0,1,1,0,\ldots\ldots\ldots\ldots,1,0,0]$$

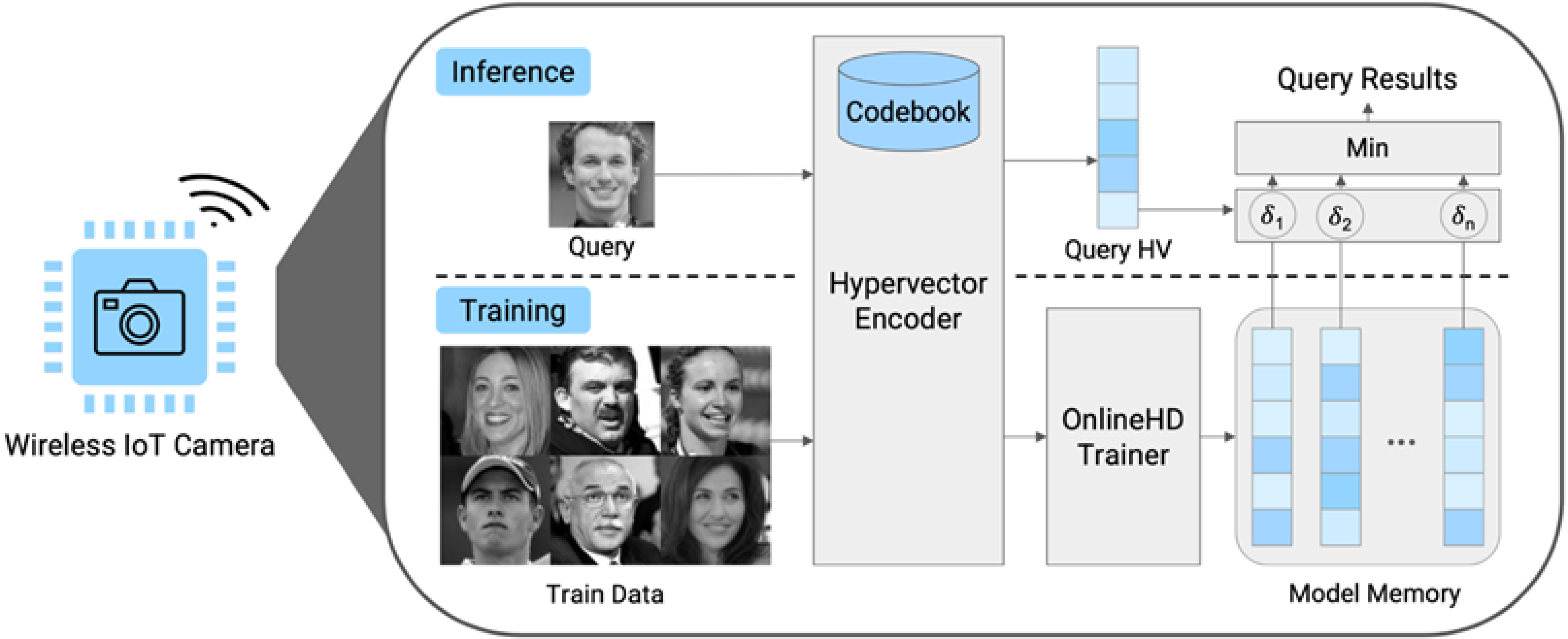
Pentti Kanerva

P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors," Cognitive Computation, vol. 1, no. 2, pp. 139-159, 2009.

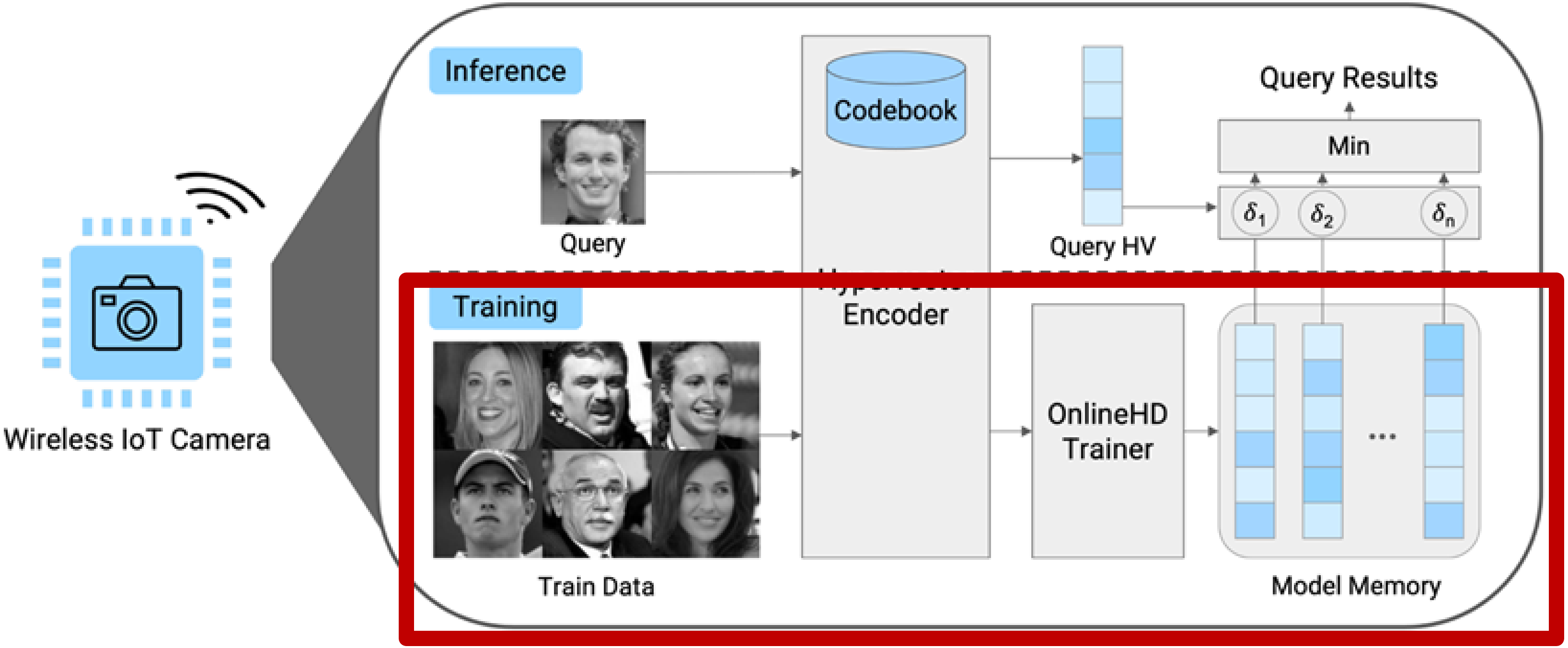
Hyperdimensional Computing (HDC)



With HDC, we can build machine learning classifiers that can work with images on the edge.

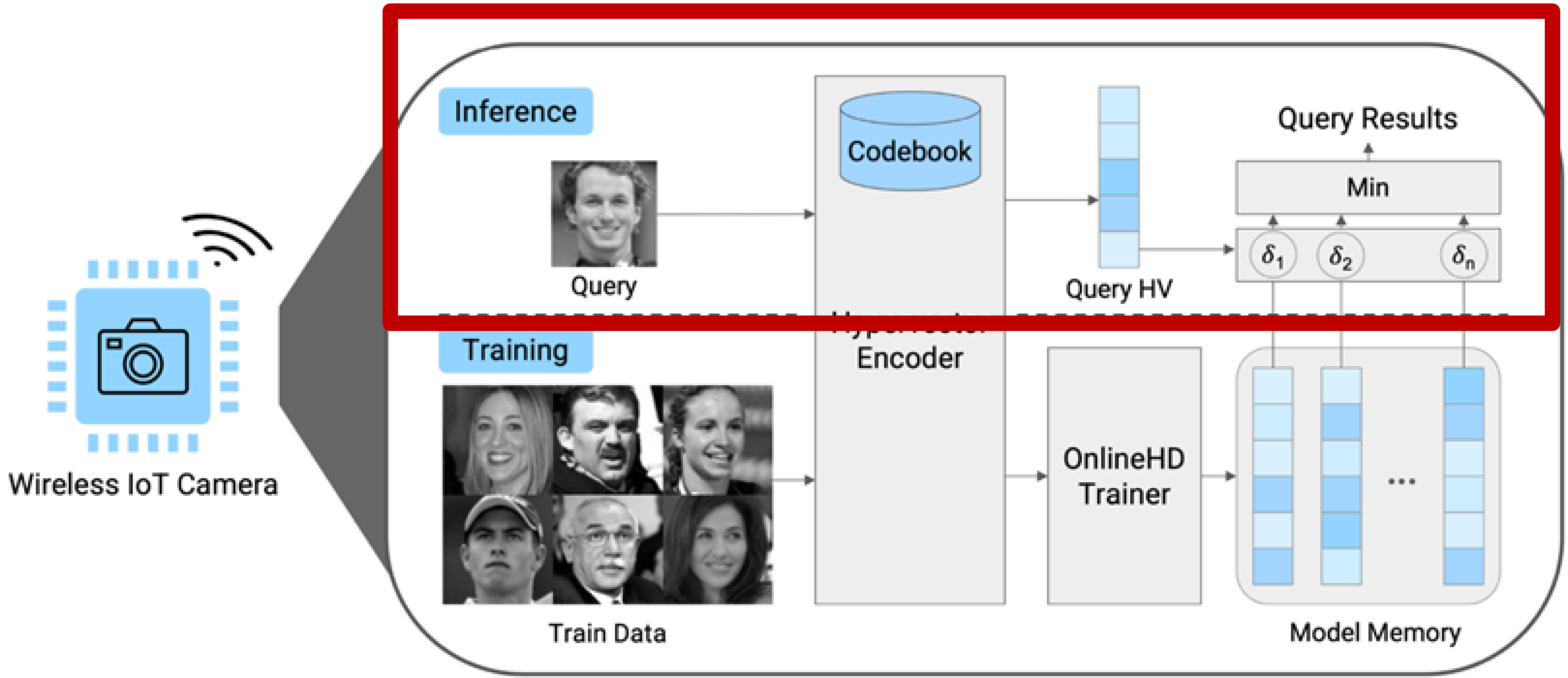


Training



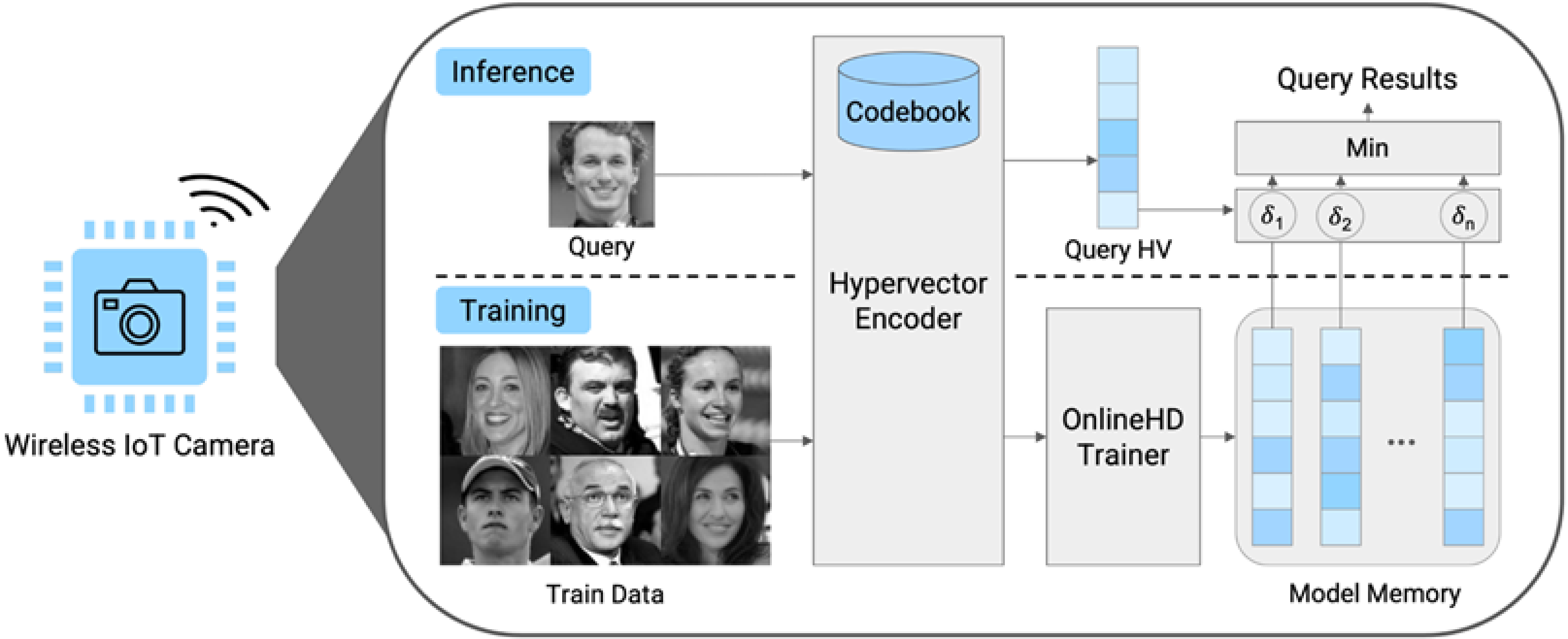
executed on commodity hardware

Inference

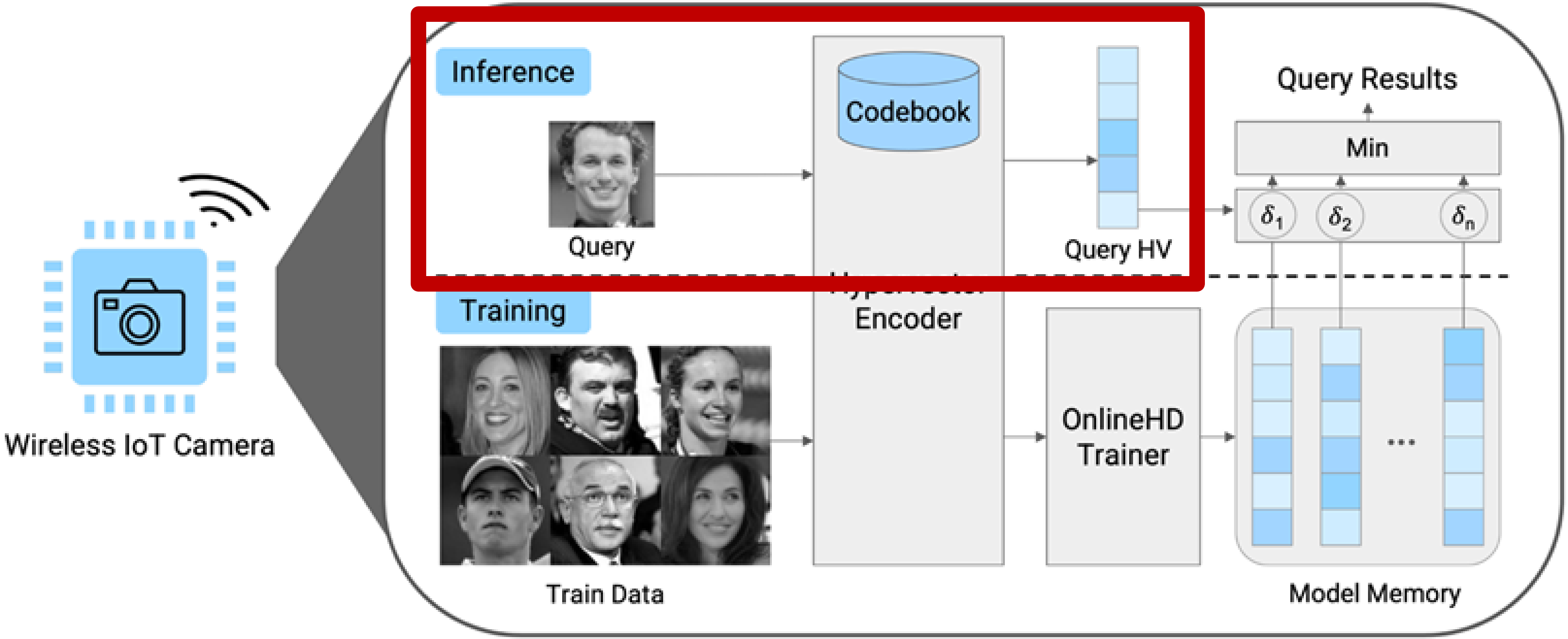


executed on ultra-low power microcontroller

Where is the performance bottleneck?



The Encoder


Wireless IoT Camera
Inference
Query
Codebook
Query HV
Query Results
Min
Encoder
Training
OnlineHD
Trainer
...
Train Data
Model Memory

The challenge is with **encoding**

Image encoding scales at O(whn), where inference only at O(n).

On STM32U585AI MCU, existing image encoding method takes **~1 minute** to encode 120x160 grayscale image.

Our innovation reduces this to **0.05s**.

HyperCam Image Encoding

HyperCam employs HD expression rewrites to reduce resource usage of encoding operation.

$$hv_{img} = \sum_{i=1}^{h} \sum_{i=1}^{w} R(i) \odot C(j) \odot V(Img[i \cdot w + j])$$

Rewrite 1 $$hv_{img} = \sum_{i=1}^{h} \sum_{j=1}^{w} p^{i}[R(0)] \odot p^{i}[C(0)] \odot V(Img[i \cdot w + j])$$

Rewrite 2 $$hv_{img} = \sum_{i=1}^{h} \sum_{j=1}^{w} p^{i \cdot w + j}[X(0)] \odot V(Img[i \cdot w + j])$$

semantics-preserving

Rewrite 3 $$hv_{img} = \sum_{z=0}^{255} |Pix(z)| \cdot V(z) \odot \left[\sum_{i,j \in Pix(z)} p^{i \cdot w + j}[X(0)] \right]$$

HyperCam Image Encoding

HyperCam deploys a novel **sparse bundling operation** that approximates bundling of independent vectors. **Sparse bundling is 500x faster than normal bundling.**

$$hv_{img} = \sum_{z=0}^{255} |Pix(z)| \cdot V(z) \odot \boxed{SparseBundle(Pix(z))}$$

Each bundling operation reduced to ~20 (instead of 10,000) vector updates.

Computational Savings

	Naive	Rewrite 1	Rewrite 2	Rewrite 3	HyperCam
Codebook	536	258	258	258	258
Bind	38400	38400	19200	19200	19200
Bundle	19200	19200	19200	19456	256
SparseBundle	0	0	0	0	19200

Table 1: Comparison of encoding methods based on the size of the codebook and the number of bind, bundle, and sparse bundle operations. Each bundling operation involves 10000 bit-wise addition, whereas each sparse bundling operation involves 20.

HyperCam gains huge computational and memory savings at the expense of degraded accuracy

Evaluation

4 Datasets

- MNIST, Fashion MNIST: 28x28 60,000 images in 10 classes
- Face Detection & Identification: 120x160 5000 images in 7 person classes and 1 non-person class

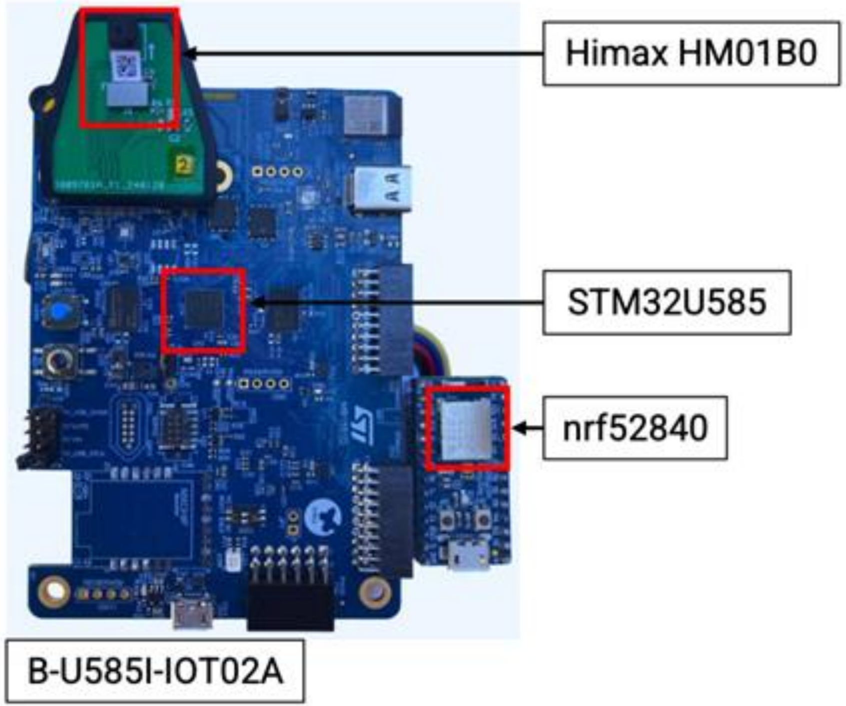
3 Types of Models

- HDC: Vanilla HDC, OnlineHD (SOTA trainer at the time), Rewrite 2
- Traditional ML Models: SVM, xgBoost Tree
- Tiny ML: MobileNet, MicroNet, MCUNet-Small, MCUNet-Large

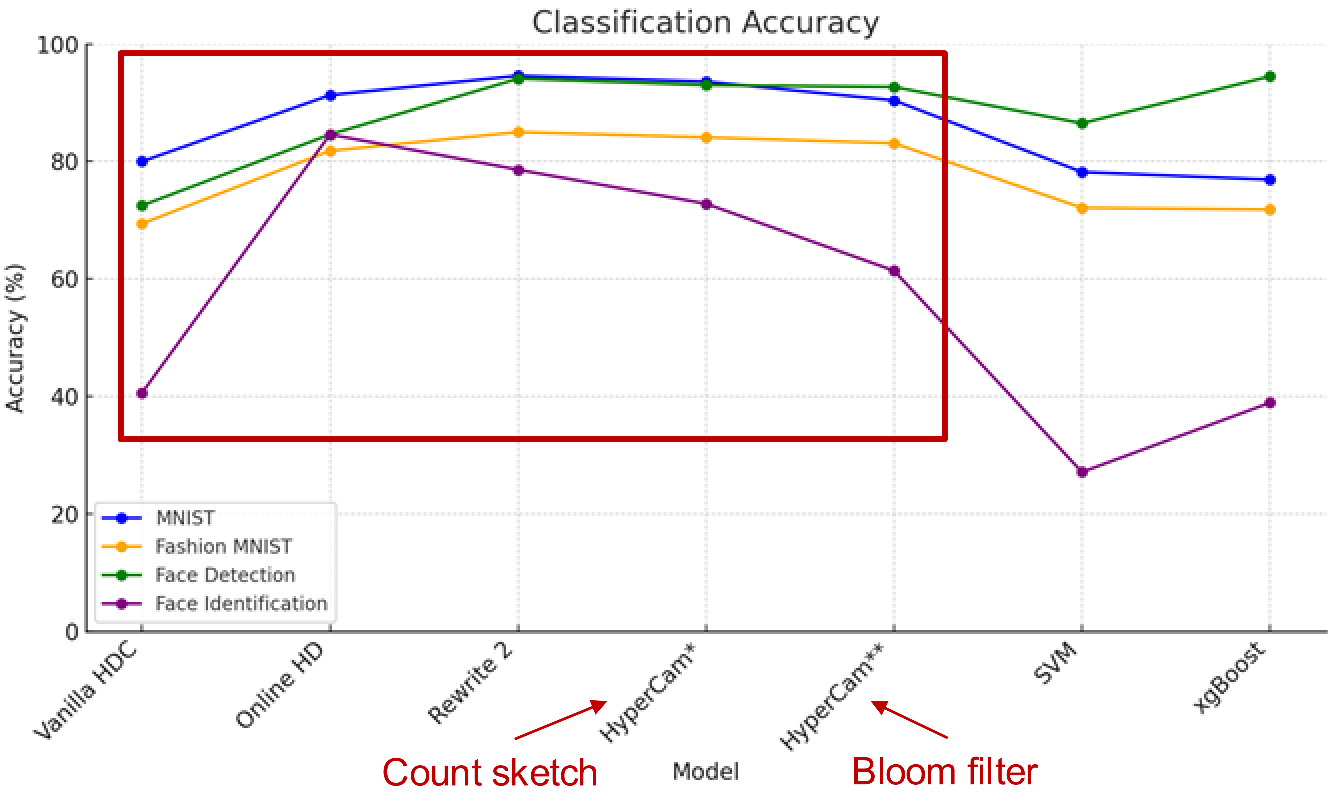
Evaluation

Test accuracy evaluated on software

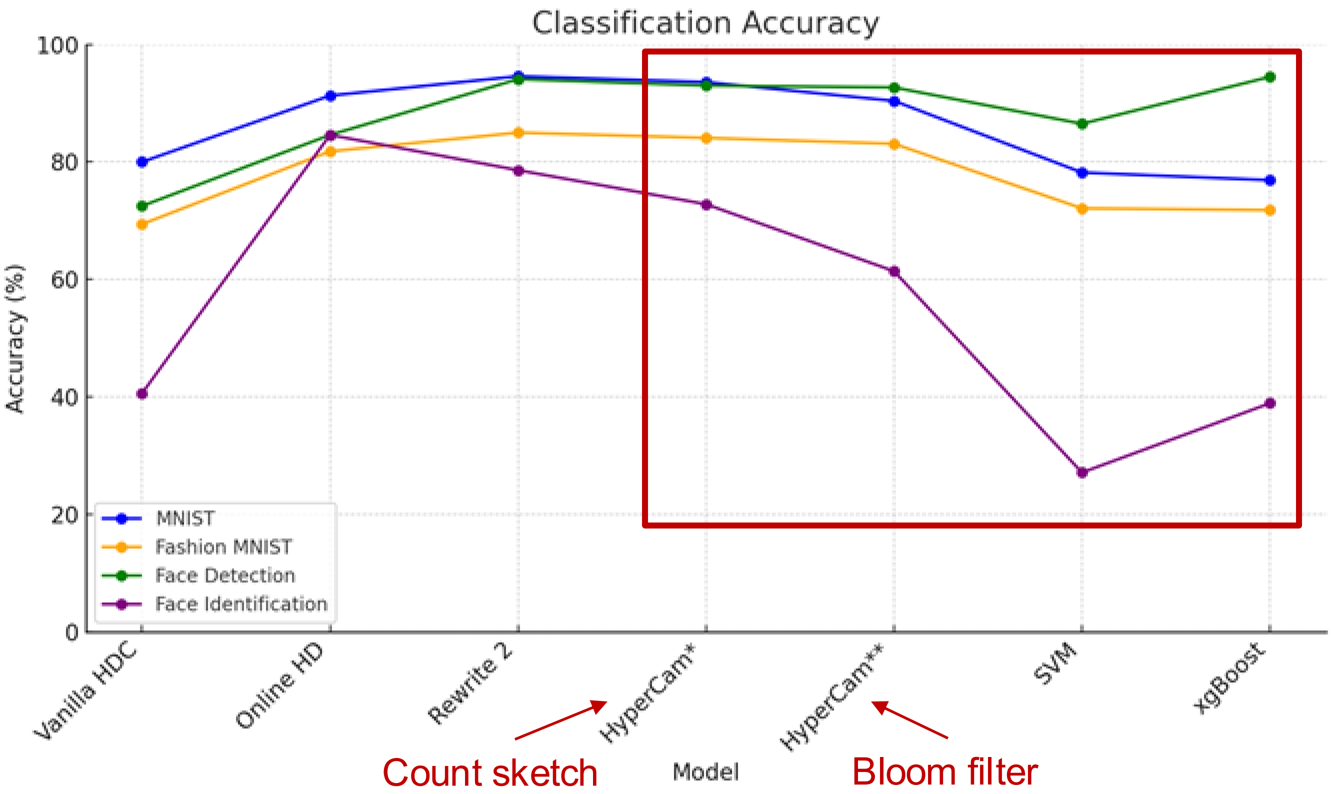
Resource usage evaluated on hardware



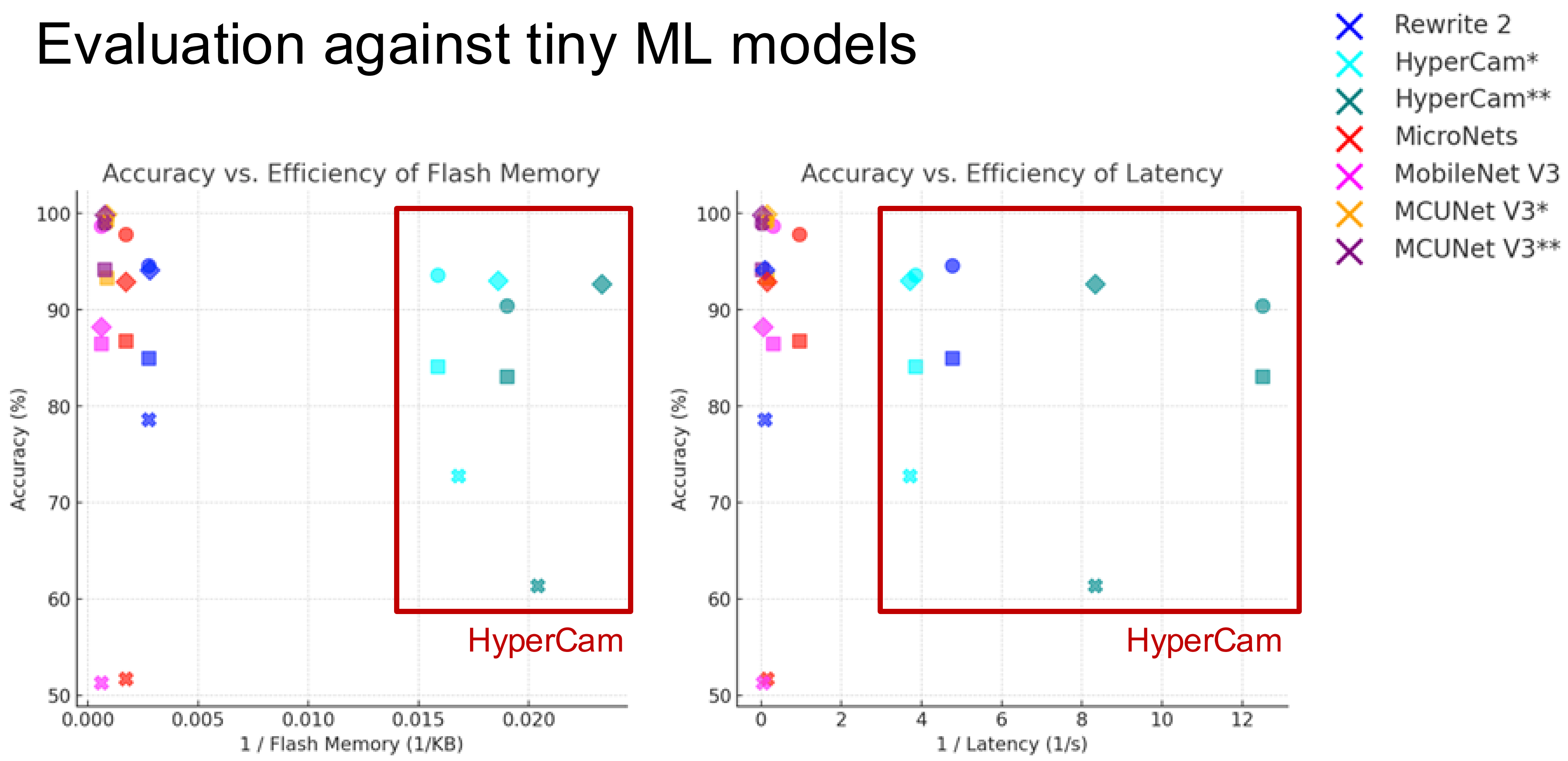
Evaluation against HDC and traditional ML models



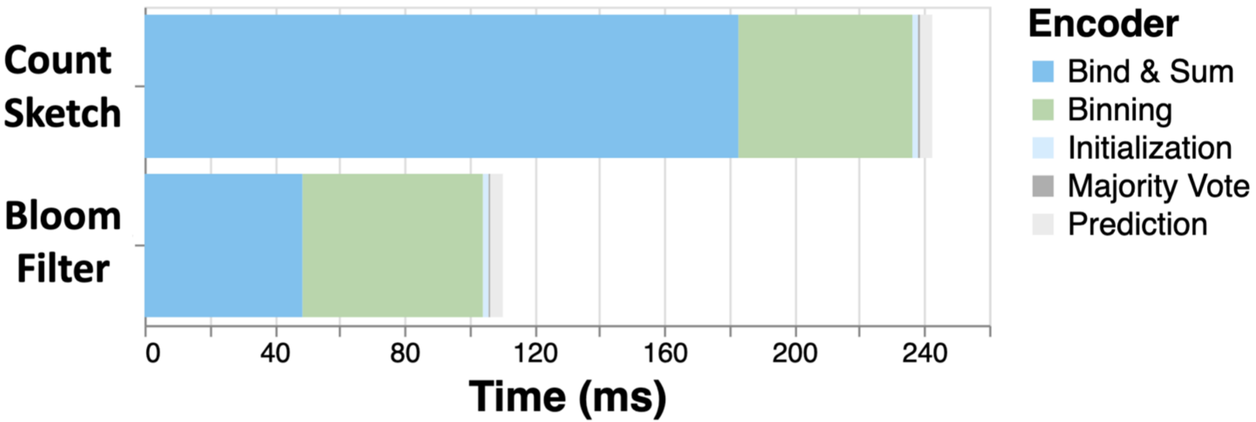
Evaluation against HDC and traditional ML models



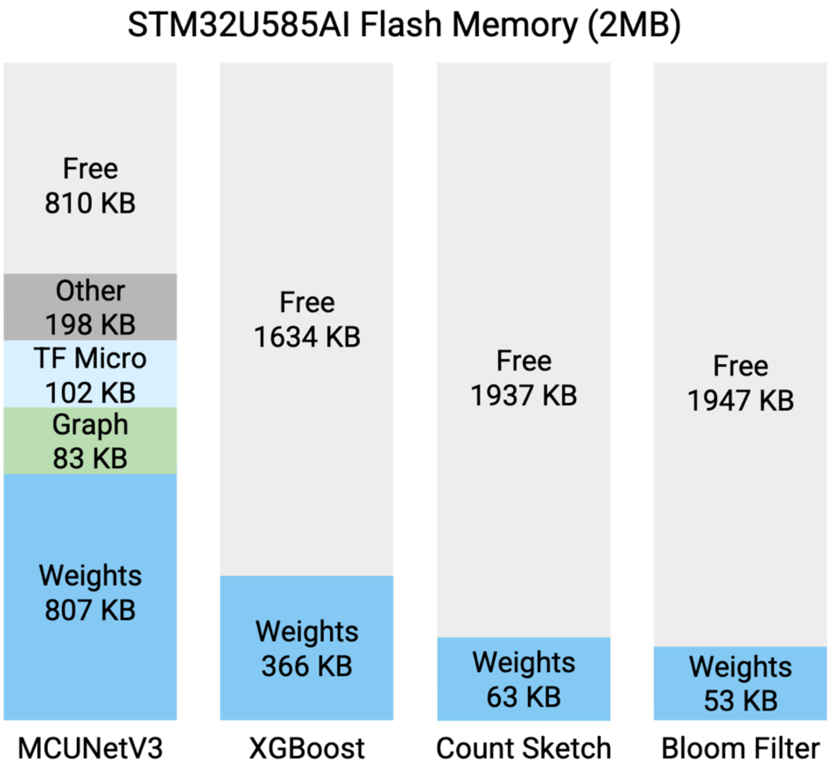
Evaluation against tiny ML models



Resource Usage



Resource Usage



Conclusion

- We propose HyperCam, a fast and lightweight ML classifier based on hyperdimensional computing.
- HyperCam optimizes HD encoding for images using sparse bundling, resulting in less than 2% accuracy drop.
- HyperCam delivers latency of 0.08-0.27s while using 42.91-63.00KB flash memory and 22.25KB RAM at peak.

Teaser: learned bloom filter beats DNNs in accuracy.