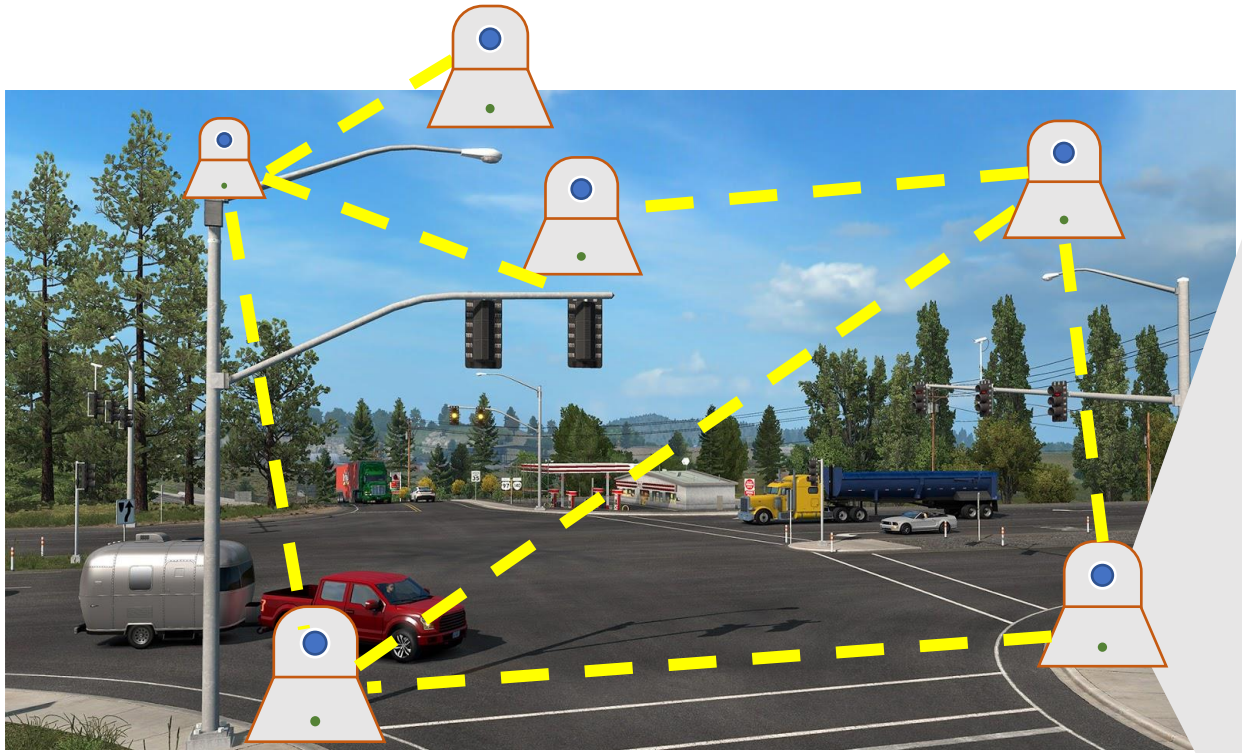
Towards Adaptive and Robust Tiny Machine Learning on Multi-Core Embedded RISC-V MCUs

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Manuele Rusci [1,2] - manuele.rusci@unibo.it
Francesco Conti [1] - f.conti@unibo.it
Luca Benini [1,3] – luca.benini@unibo.it

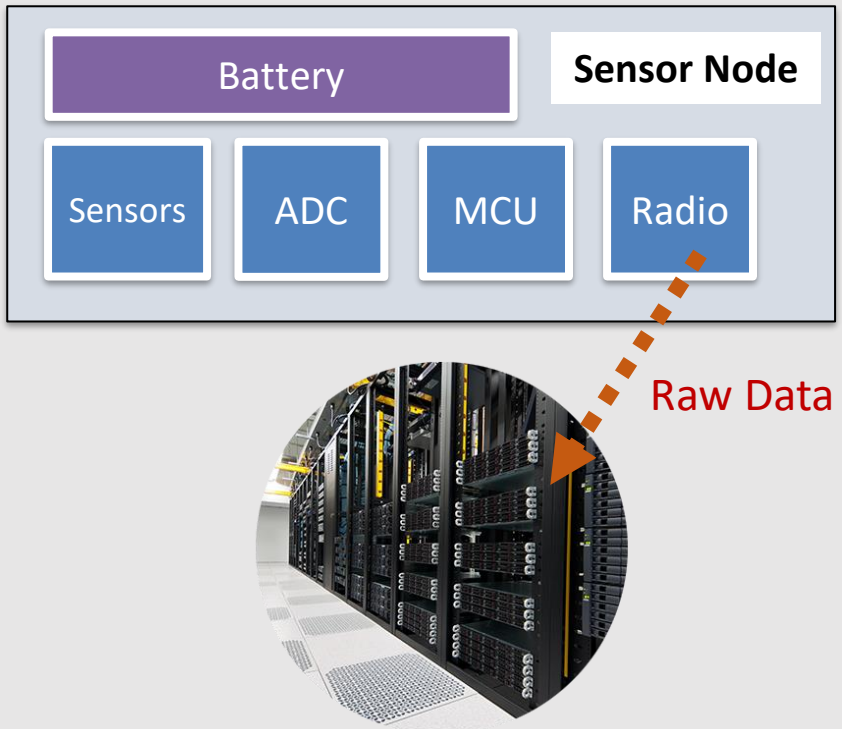
[1] Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI) – Università di **Bologna**
[2] GreenWaves Technologies – Bologna
[3] Integrated Systems Laboratory – ETH Zurich

Tiny Machine Learning for the IoT ecosystem



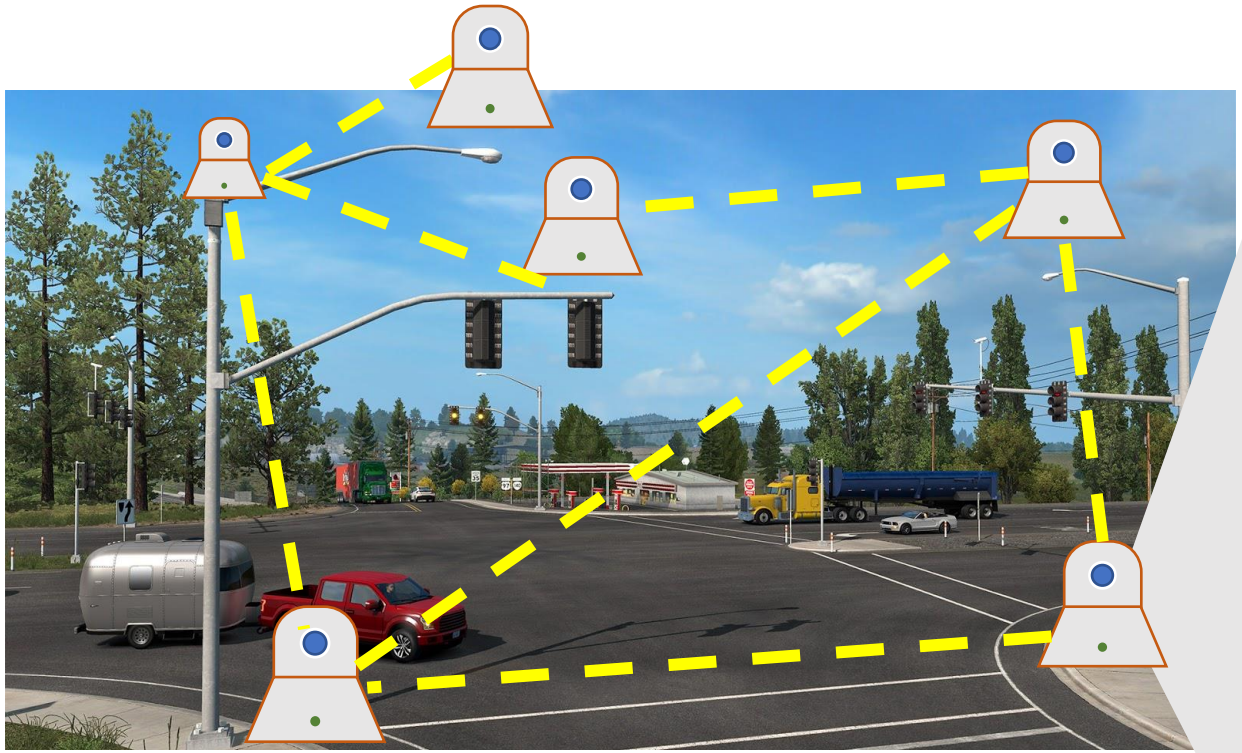
[Credits: Unknown Author]

Miniaturized **low-power** and **low-cost** sensor nodes for cyber-physical systems



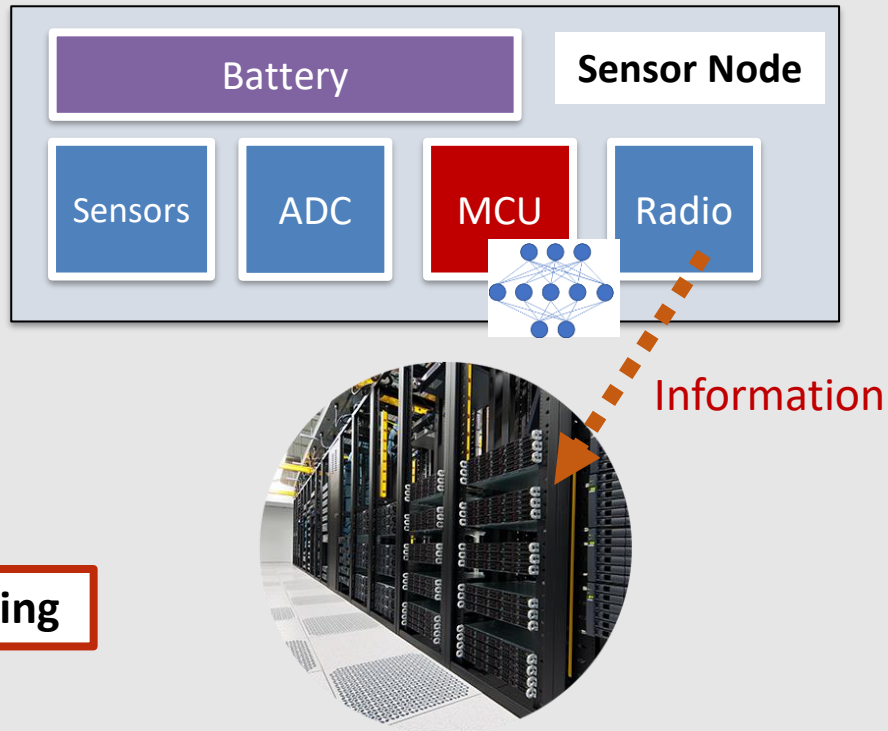
- ✓ simple design
- ✓ server computational power
- ✗ Network scalability
- ✗ Privacy of personal data
- ✗ TX Power consumption (100s mWs)

Tiny Machine Learning for the IoT ecosystem



[Credits: Unknown Author]

Miniaturized **low-power** and **low-cost** sensor nodes for cyber-physical systems



Bringing Machine (Deep) Learning to Tiny Devices → **Tiny Machine Learning**

Every **Smart Sensor** plays *data-to-information* conversion and transmits only low-bandwidth data to the server

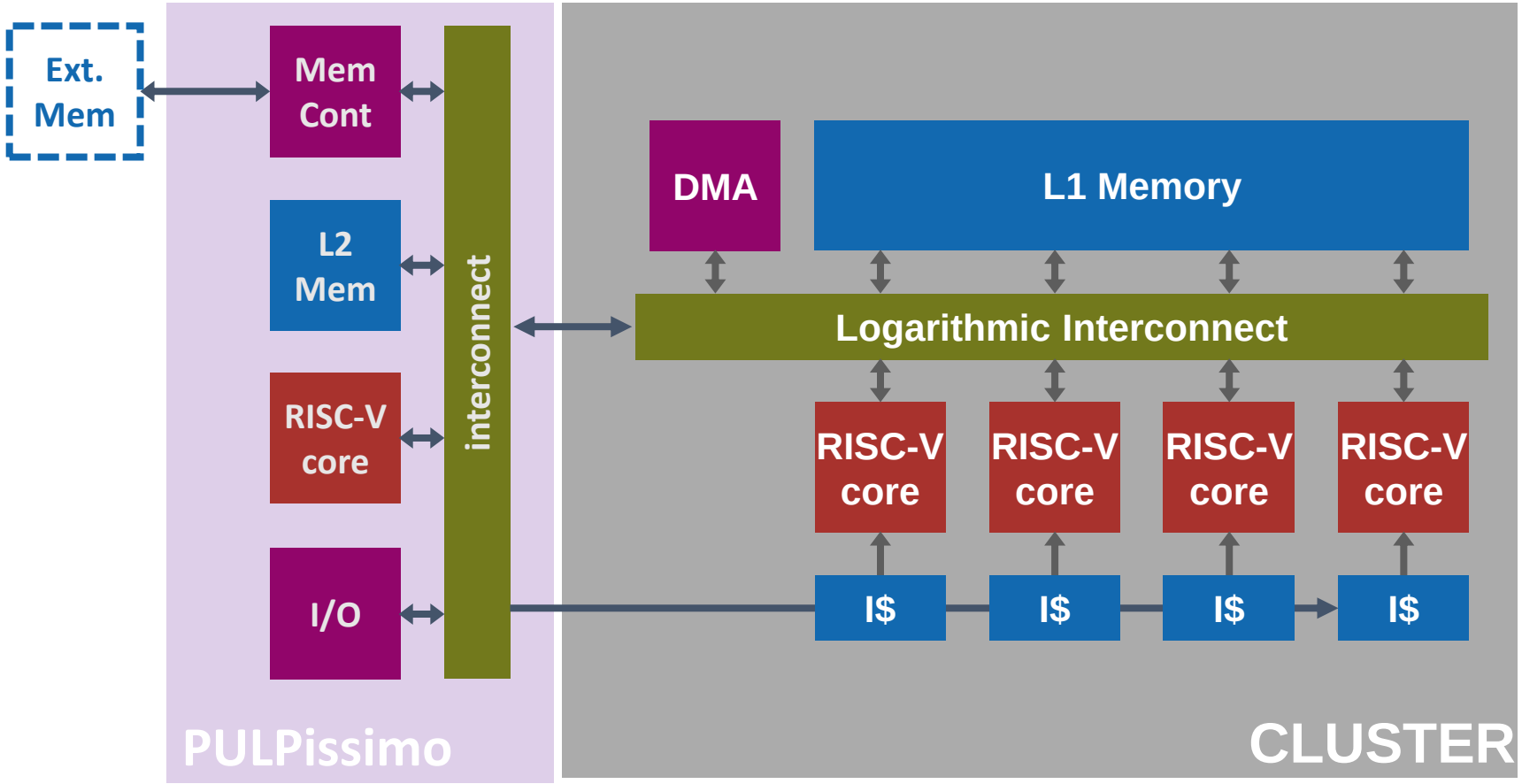
✓ Privacy by design

✓ TX Bandwidth and Power Reduction

→ Need of Energy-Efficient TinyML devices, tools and design methods!

An (Open) HW perspective: the PULP platform

A **RISC-V based multi-core MCU-like platform**, with extended RISC-V ISA to efficiently support ML workloads, e.g. vectorized 8-bit MAC single-cycle instructions

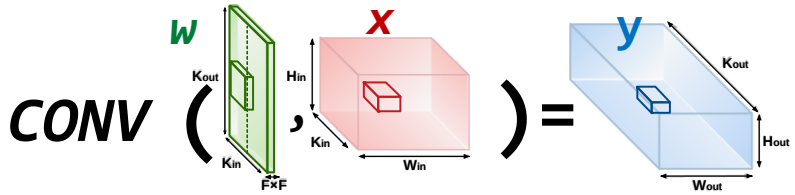


Open sourced since 2017:
github.com/pulp-platform/pulp

Greenwaves Technologies GAP-8:
1+8 cores @ 175MHz, <100mW

More info at: https://www.pulp-platform.org/

PULP-NN: Accelerating DNN Inference on the PULP platform




```
parallel for i,j in range(0, H_out, W_out):
  for m in range(0, K_out):
    for ui in range(0, F):
      for uj in range(0, F):
        for n in range(0, K_in):
          im2col[ui*F*K_in + uj*K_in + n]=
                    x[i+ui,j+uj,n]
    psum = 0
    for idx in range(0, F*F*K_in):
      psum += w[m,idx] * im2col[idx]
    y[i,j,m] = act(psum)
```

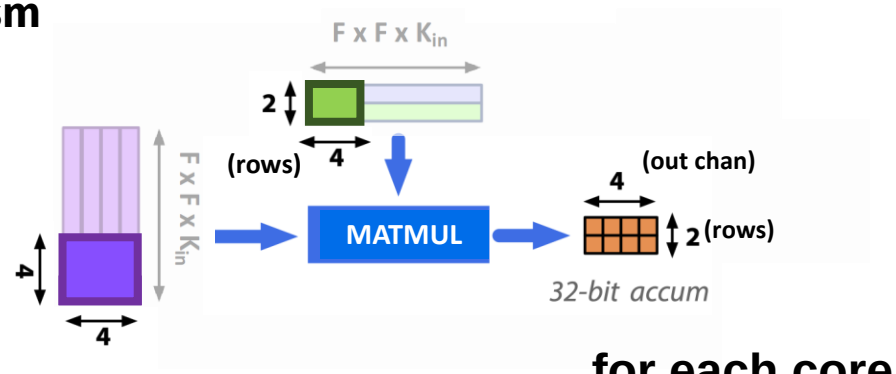
Parallel loop

im2col

GEMM

- Peak Performance of **15.5 MAC /clk** on **8** cores
- **19.6x** and **30x** faster than CM7 and CM4 based MCUs using CMSIS-NN for CIFAR10 inference

Optimized GEMM on PULP by 1) maximizing **data reuse in register file** 2) exploiting **SIMD MAC isntructions** and 3) **parallelism**




```
lp.setup
  p.lw  w0, 4(W0!)
  p.lw  w1, 4(W1!)
  p.lw  w2, 4(W2!)
  p.lw  w3, 4(W3!)
  p.lw  x1, 4(X0!)
  p.lw  x2, 4(X1!)
  pv.sdotsp.b    acc1, w0, x0
  pv.sdotsp.b    acc2, w0, x1
  pv.sdotsp.b    acc3, w1, x0
  pv.sdotsp.b    acc4, w1, x1
  pv.sdotsp.b    acc5, w2, x0
  pv.sdotsp.b    acc6, w2, x1
  pv.sdotsp.b    acc7, w3, x0
  pv.sdotsp.b    acc8, w3, x1
end
```

for each core

Load 16 weights (8-bit)
4 out chan, 4 in chan
address post-increment

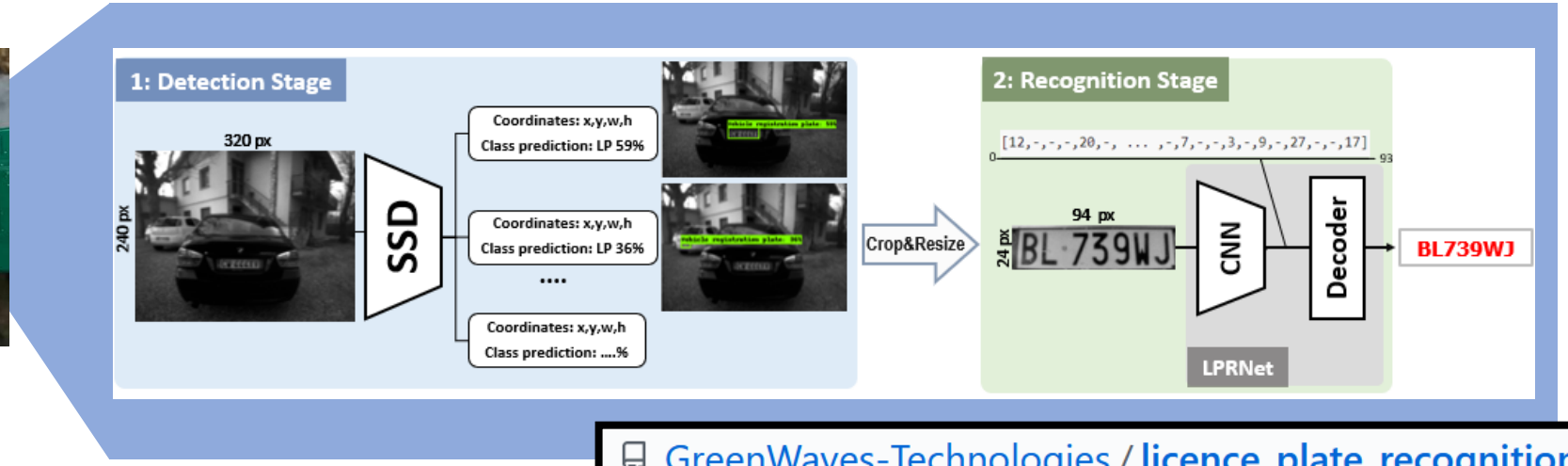
Load 8 pixels
2 rows, 4 in chan
address post-increment

Compute 32 MAC over 8 accumulators using **vectorized** dot-product instructions

Garofalo A. "PULP-NN: accelerating quantized neural networks on parallel ultra-low-power RISC-V processors." *Philosophical Transactions of the Royal Society A* 2020.

Leveraging the PULP technology for real world applications

Automatic License Plate Detection and Recognition on GAP8



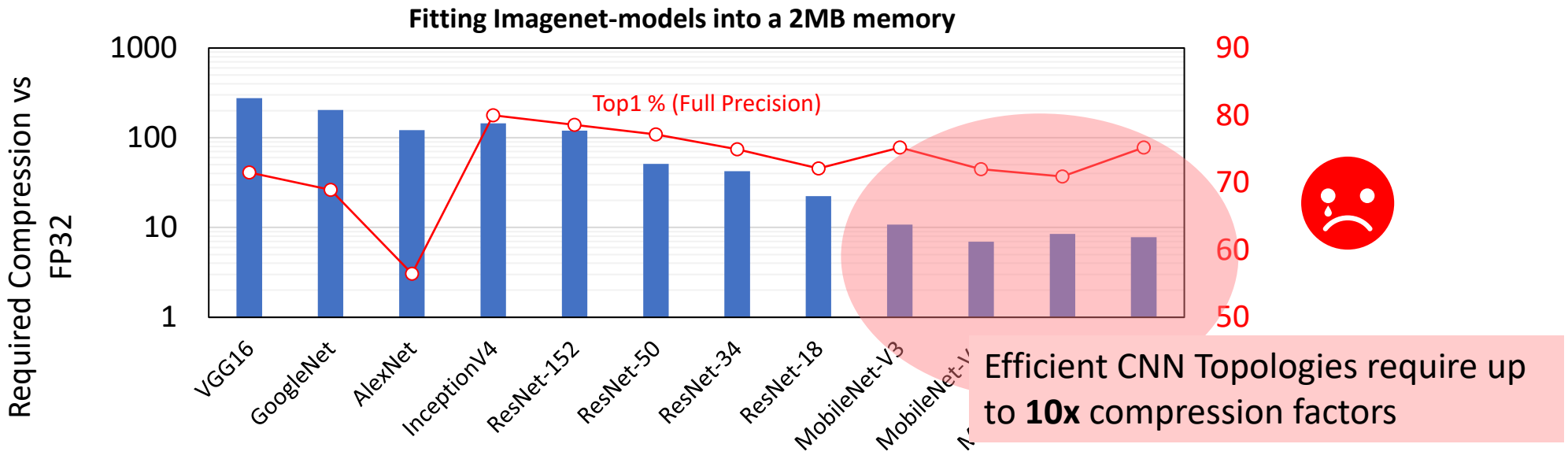
GreenWaves-Technologies / licence_plate_recognition

- **1.1 FPS** inference @ 175MHz, performing **687M MAC**.
  highest complexity on MCUs
- **4.1 MB** memory footprint (after 8-bit quantization)
  - using 8MB off-chip memory
- **Accuracy:** 39% mAP for LP det. & > 99.13% for LP rec.
- **Max recognition distance:** 4m for detection and 2m for recognition
- **117mW** power envelope**, 108 mJ** per inference.
  73x less energy w.r.t. previous ALPR system.

L. Lamberti et al. “Low-Power License Plate Detection and Recognition on a RISC-V Multi-Core MCU-based Vision” ISCAS 2021

Lossless compression of DNNs under severe memory constraints

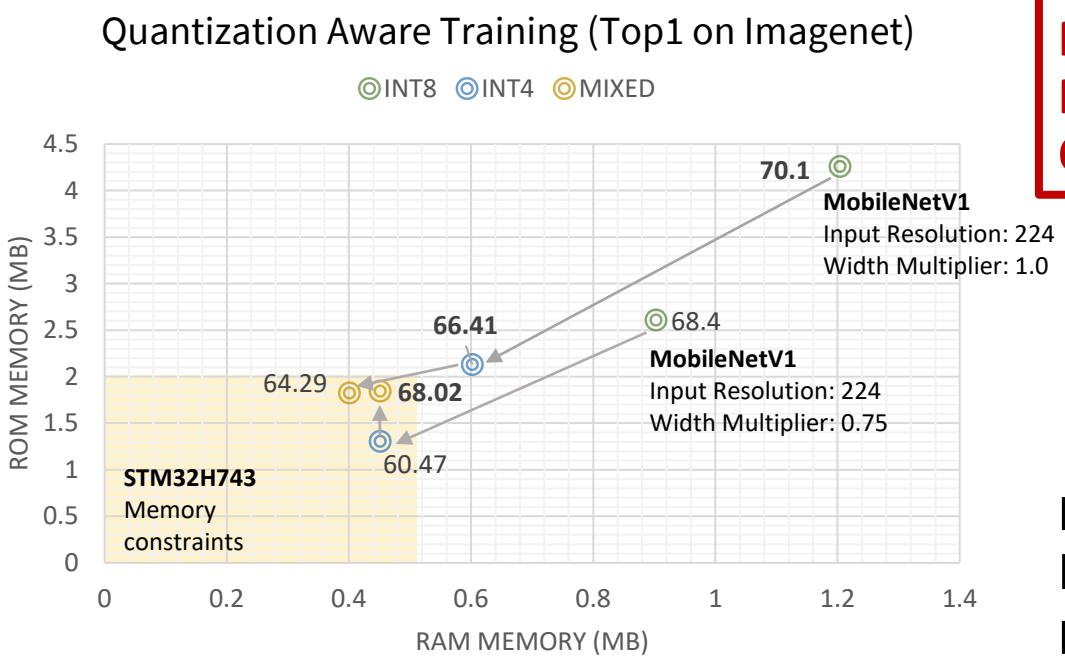
On-chip memories are highly constrained (< 2MB) and off-chip memories present high costs (energy, $).



Efficient CNN Topologies require up to **10x** compression factors

Issue: 8-bit quantization (4x compression) is **not** sufficient to bring 'complex' models on MCUs

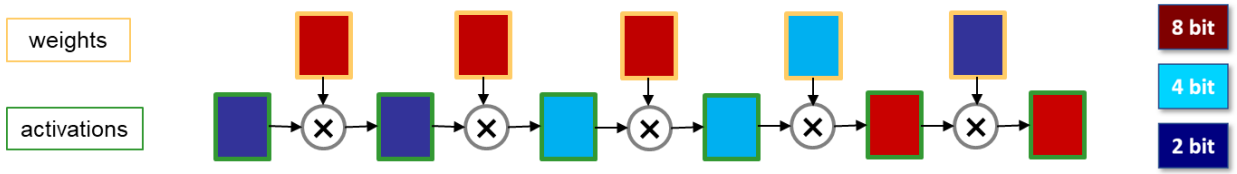
DNN Inference with Mixed-Low-Precision Quantization



- -2% wrt most accurate INT8 mobilenet (224_1.0)
- **+8% wrt most accurate INT8 mobilenet fitting the memory (192_0.5)**
- +2% wrt most accurate INT4 mobilenet (224_1.0)

Mixed Low Precision Quantization

apply *tensor-wise quantization* to minimize accuracy degradation wrt full-precision models



Rule-based (Rusci, Mlsys 2020): rule-based bit reduction policy (from 8-bit to 2-bit) up to fit the memory constraints

Reinforcement-Learning (Rusci, ITEM 2020): same accuracy levels – the RL agent maximizes memory utilization

Model	#Params (M)	FP32 Top1 Accuracy	Mixed Prec. Top1 Accuracy	Mixed Prec. Weight Compression	Mixed Prec. Activ. (average) Compression
MobileNetV1 (224_1.0)	4.2	70.6%	**68.4%**	9x	5.1x
MobileNetV1 (224_0.75)	2.6	68.4%	68.0%	5.4x	4.6x
~~MobileNetV2 (224_1.0)~~	~~3.4~~	~~72.04%~~	~~55.1%~~	~~7.6x~~	~~5.1x~~
MobileNetV2 (192_1.0)	3.4	70.7%	58.0%	7.4x	5.2x
MobileNetV2 (160_1.0)	3.4	68.8%	67.5%	7.4x	4.4x
MNasNet-A1 (224_1.0)	3.9	**75.2%**	62.6%	9.4x	4.7x
MNasNet-A1 (224_0.35)	1.7	64.1%	64.1%*	4x (8-bit)	4x (8-bit)

* optimistic score: we assume a lossless 8-bit compression w.r.t. FP32

(Rusci, Mlsys, 2020) Rusci et al, "Memory-driven mixed low precision quantization for enabling deep network inference on microcontrollers." *Mlsys* (2020).
(Rusci, ITEM 2020) Rusci, M. et al. "Leveraging Automated Mixed-Low-Precision Quantization for tiny edge microcontrollers." ITEM 2020.

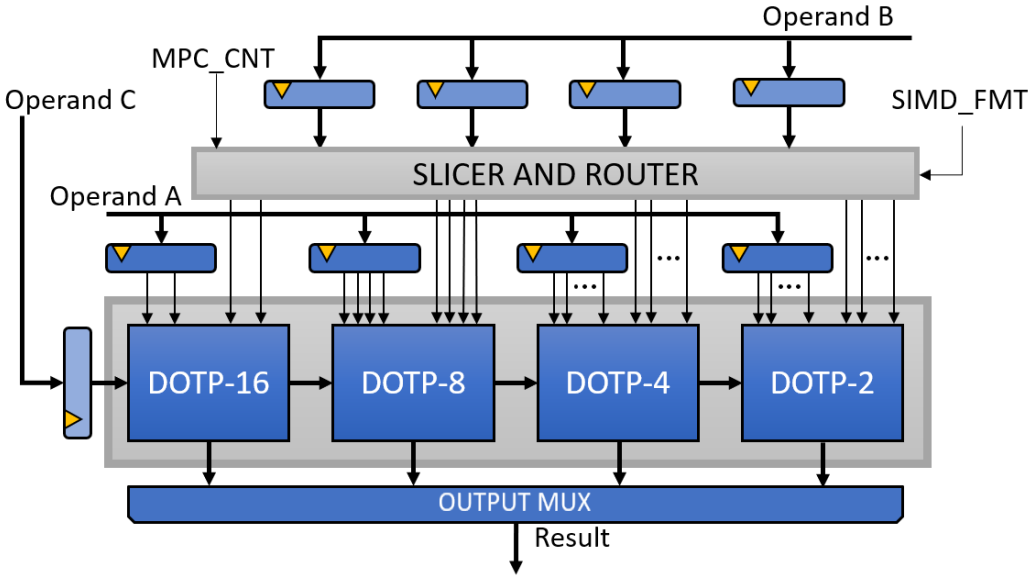
Mixed-Precision Inference for MCUs

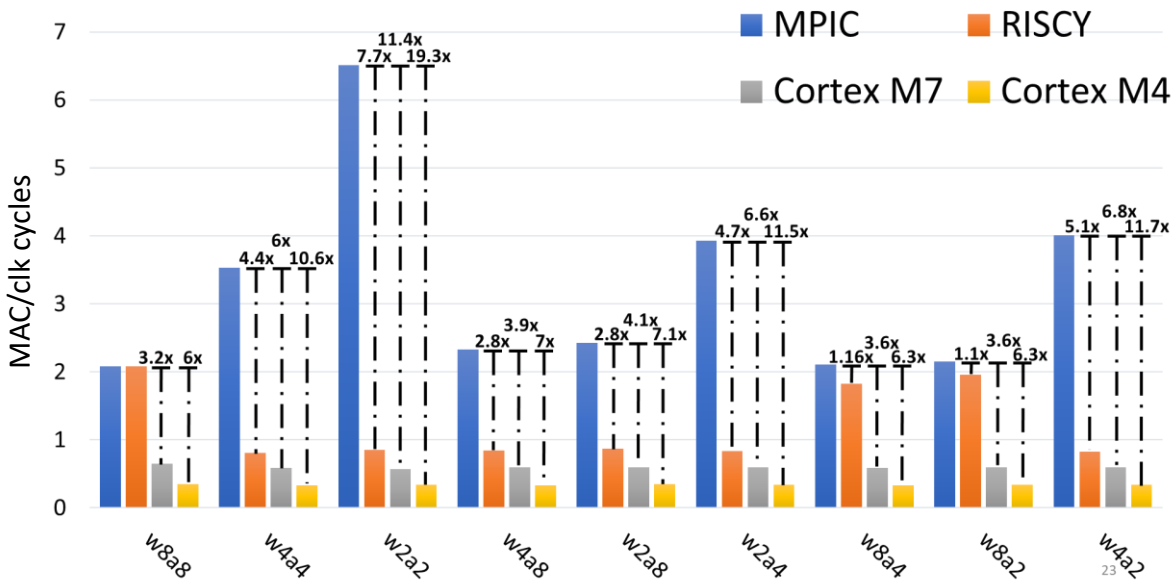
Low-bitwidth multi-precision is not natively supported by RISC-V ISA.

- ➢ SW-emulation of 2-4-bit Mixed precision MAC adds overhead mixed-precision inference results slower than 8-bit inference

The **Mixed-Precision Inference Core (MPIC)** (Ottavi, 2020) added support for 4-, 2-bits and mixed precision operations from 16 down to 2 bits

- ➢ Novel SIMD DOT-Product unit





Quantized DNN kernels mixed precision:

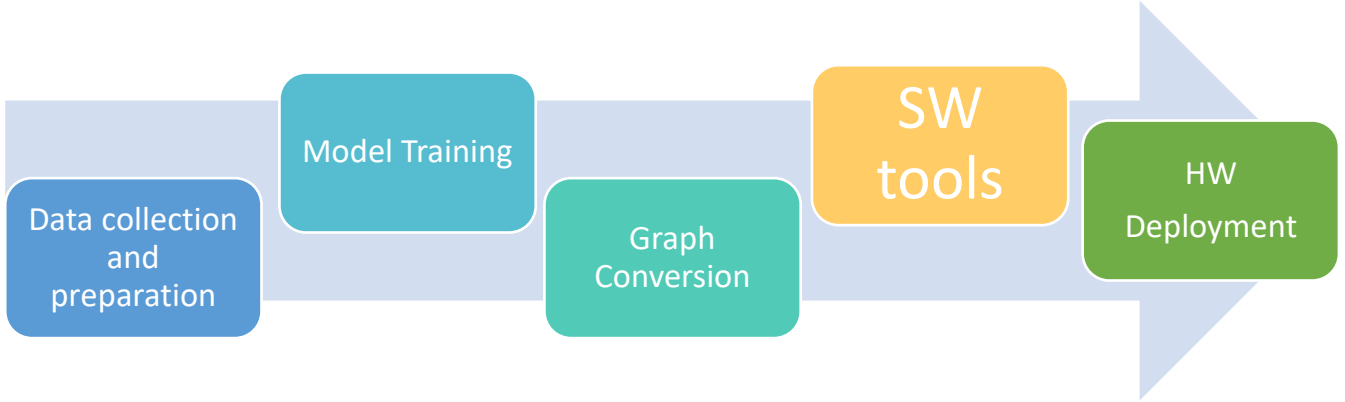
- Up to 4.9x better performance compared to RISCY
- Up to 11.7x better performance compared to Cortex M4

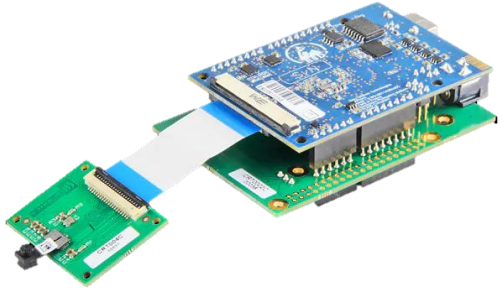
Demand for end-to-end tools…

(Ottavi 2020) Ottavi G. et al "A mixed-precision RISC-V processor for extreme-edge DNN inference." ISVLSI 2020.

But real-world is challenging!

"Train-Once-Deploy-Everywhere" approach





- ❑ MCU's constraints prevent the deployment of robust "Big" models w/ high generalization capacity
- ❑ Training data differs from *real-world* sensor data
- ❑ More use-case specific data is always needed (e.g. custom keyword, different voice accent)
  - ❑ New dataset collection and model retraining (from scratch)
- ➢ Takes 10's months for application development

  Not acceptable!

Class: "*bottle*"

Google Open Images

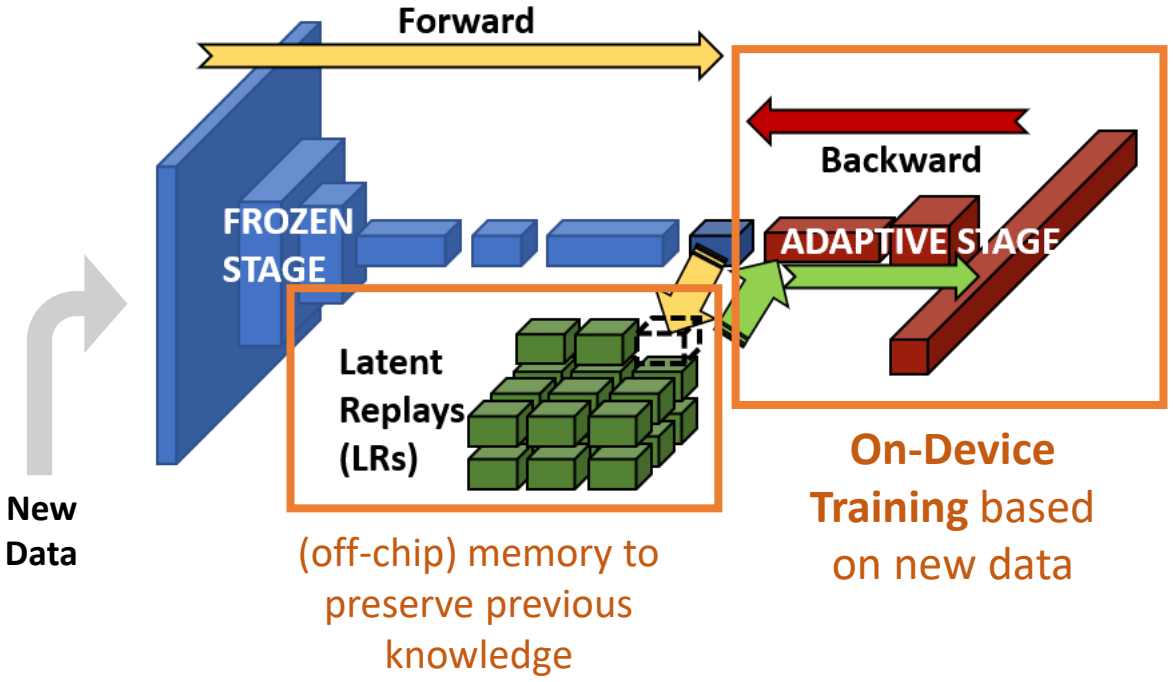
Low-Power Camera

Presently, the <u>major issue</u> towards the diffusion of smart sensor nodes relates to **the lack of adaptation over time**.

Continual Learning as the next paradigm for smart sensors: Adaptive TinyML

Continual Learning to train incrementally DL models **on device** starting from initial knowledge

- **Latent replays** to deal with catastrophic forgetting [Pellegrini 2019]



- Quantize the latent replays (and the Frozen stage) to gain **>4x** compression at **1-3%** accuracy loss (8-bit LR)

LR layer	FP32 baseline	FP32 + UINT-8	UINT-8 + UINT-8	FP32 + UINT-7	UINT-8 + UINT-7
27	**72.7** ± 0.34	**70.1** ± 0.54	**69.2** ± 0.48	**68.0** ± 0.63	**67.8** ± 1.14
25	**73.3** ± 0.58	**70.9** ± 0.65	**70.2** ± 0.67	**66.2** ± 0.75	**66.1** ± 0.94
23	**75.0** ± 0.83	**73.2** ± 0.46	**73.4** ± 0.66	**71.1** ± 0.63	**69.9** ± 1.25
21	**76.5** ± 0.63	**74.9** ± 0.51	**73.9** ± 1.67	**72.7** ± 0.74	**72.6** ± 1.30
19	**77.7** ± 0.73	**76.5** ± 0.48	**76.0** ± 0.80	**74.0** ± 0.57	**75.2** ± 1.10

Accuracy on Core50 with 1500 MRs

- Running FP32 Backpropagation on a PULP-based system, up to **0.62 FMAC/clk** with **8** cores and 512kB L1 memory
- **65x** faster than CM4-based device thanks to the higher frequency (22nm tech), 7.2x parallel speed-up and optimized ISA micro-architecture.

Ravaglia et al. "A TinyML Platform for On-Device Continual Learning with Quantized Latent Replays." JETCAS 2021

[Pellegrini 2019] Pellegrini, Lorenzo, et al. "Latent replay for real-time continual learning." IROS, 2020

Conclusion

Robust Deep Learning on smart tiny device is getting real, thanks to a combination of factors:

- Energy-efficient but flexible HW
- SW stack to efficiently exploit the underlying HW
- DL-HW codesign to enable mixed-low precision on low-power processors
    - More tools are needed

Moving first steps into Adaptive TinyML:

- Proof of concept of On-Device Continual Learning on MCU-like platform
- More to be done… (application side, algorithm, SW, tools..)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dr. Manuele Rusci

DEI – Università di Bologna

manuele.rusci@unibo.it