Deep Learning with Low Precision Hardware

Challenges and Opportunities for Logic Synthesis

Luca Benini

ETHZ & UNIBO

PULP
Parallel Ultra Low Power

http://www.pulp-platform.org

European Research Council

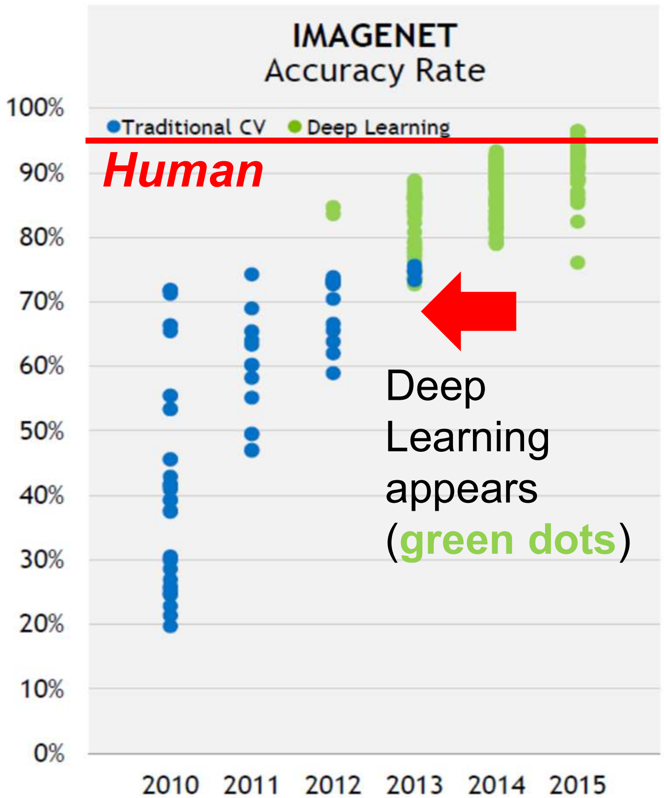
European Commission | Horizon 2020 European Union funding for Research & Innovation

nano-tera.ch

Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

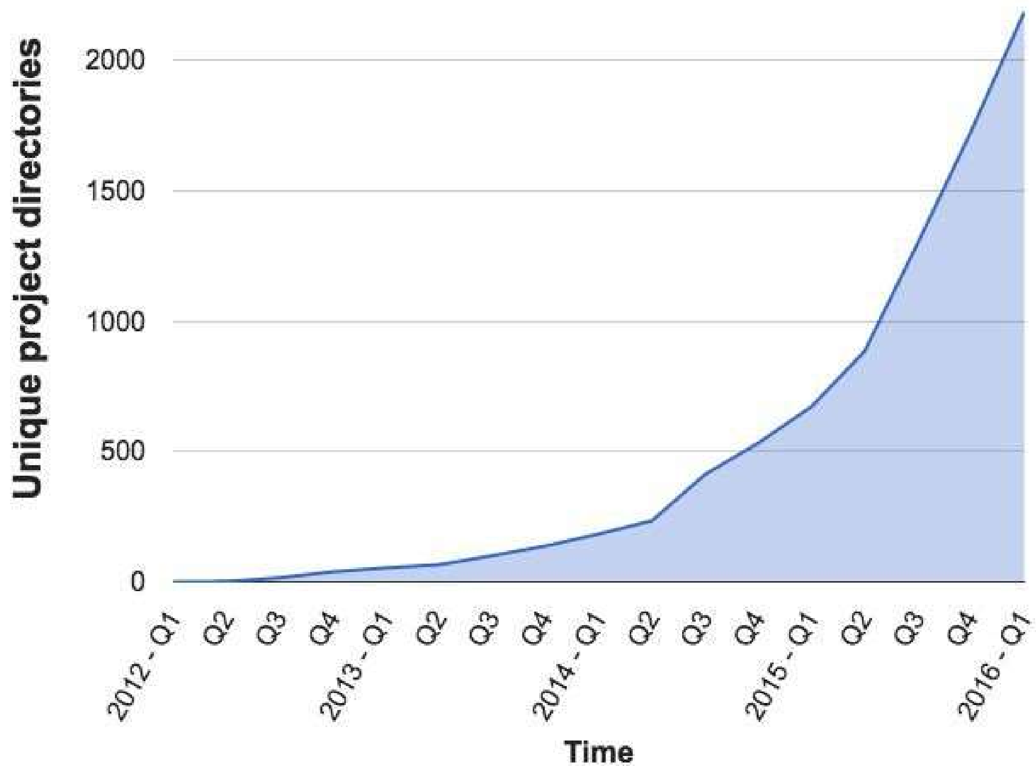
Deep Learning: Why?

First, it was machine vision…

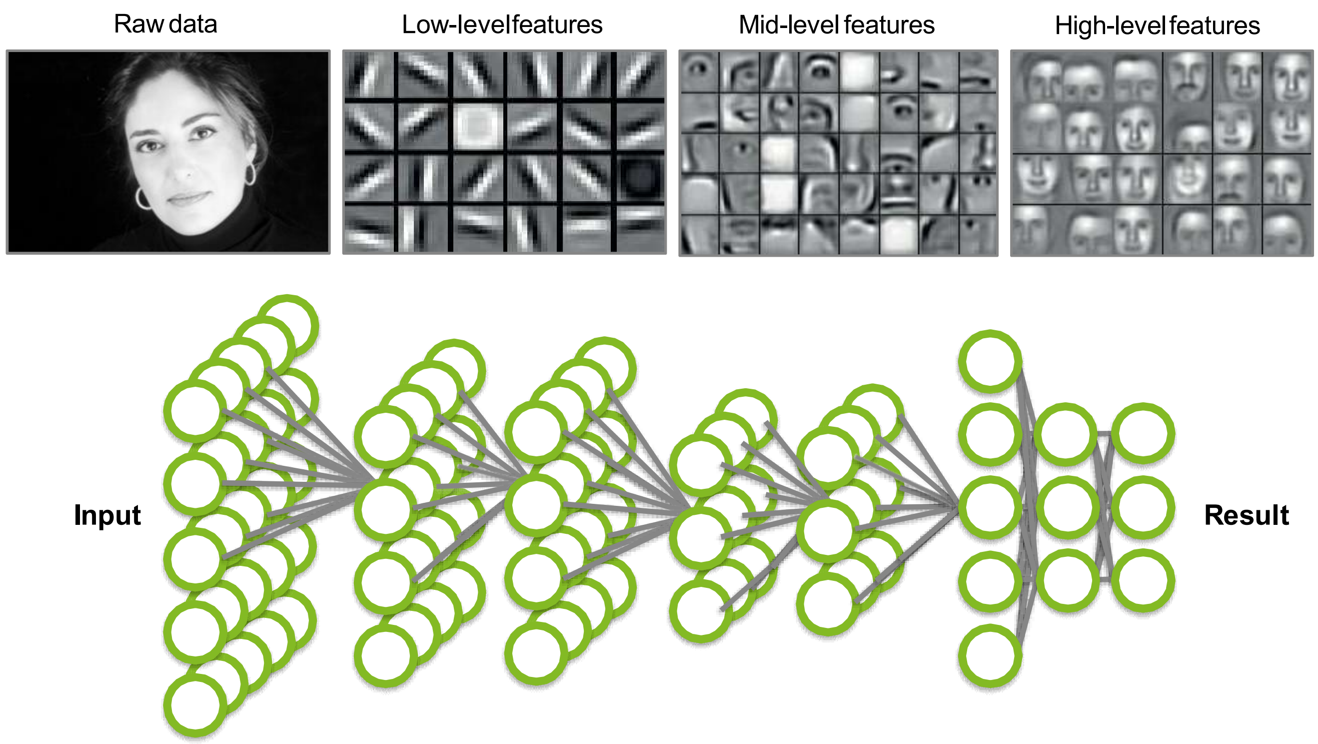


Now it's everywhere!

Growing Use of Deep Learning at Google [J. Dean]



Deep neural networks (DNNs)

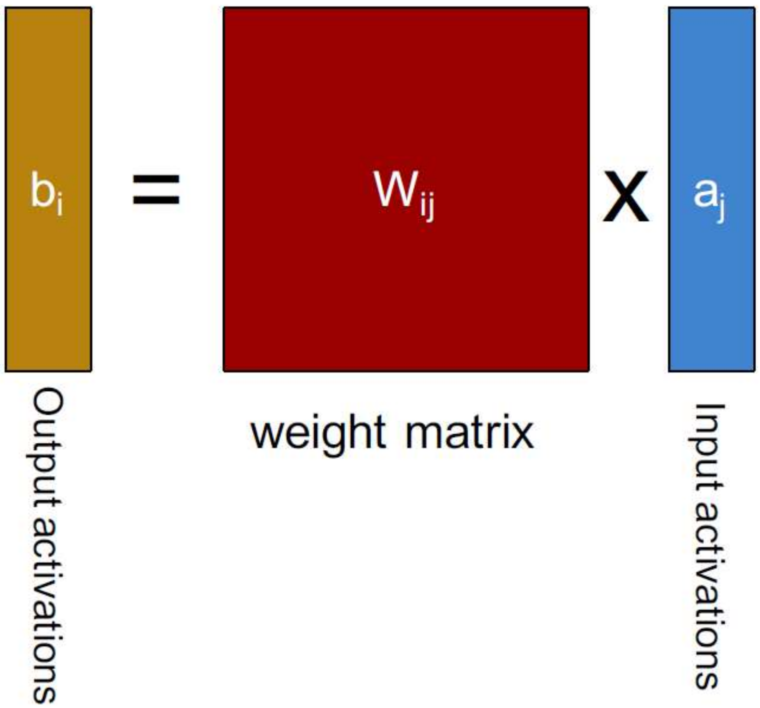


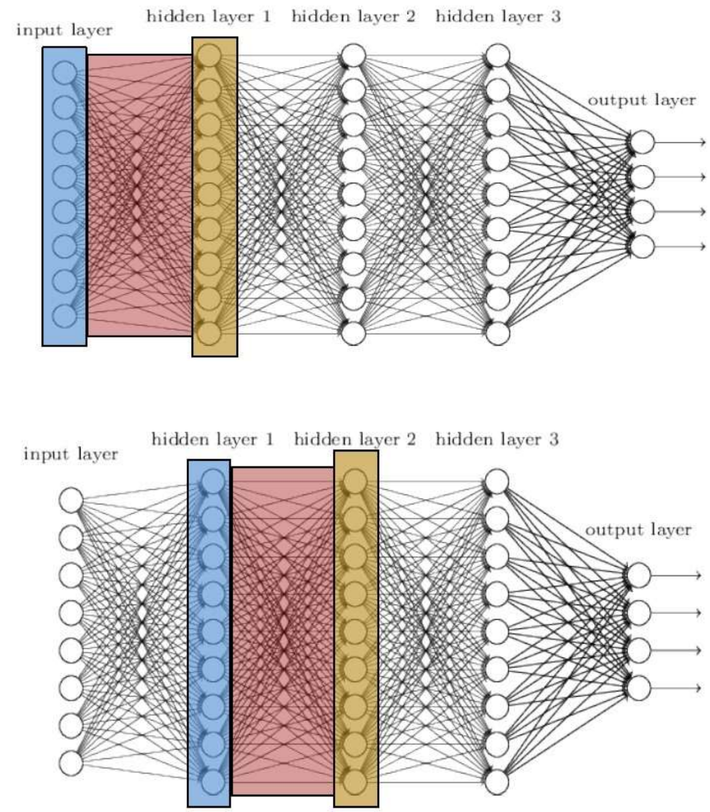
Application components:

Task objective
e.g. Identify face
Training data
10-100M images
Network architecture
~10 layers
1B parameters
Learning algorithm
~30 Exaflops
~30 GPU days

Key operation is dense M x V

Layer-by-layer computation





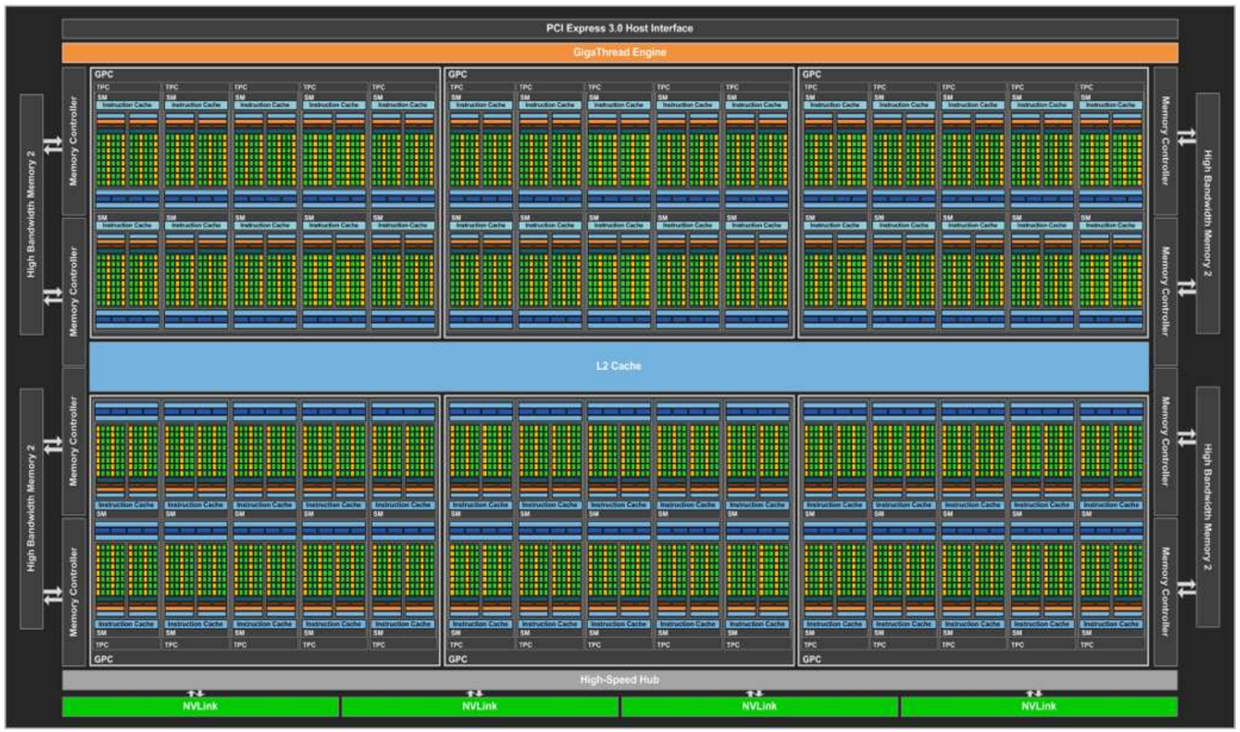
Repeat for each layer

Training by recursive backpropagation of error on fitness function

GPUs are Great for Vanilla CNNs

Why?

Because they are good at matrix multiply → 90% utilization is achievable (on lots of "cores")



28pJ/OP*

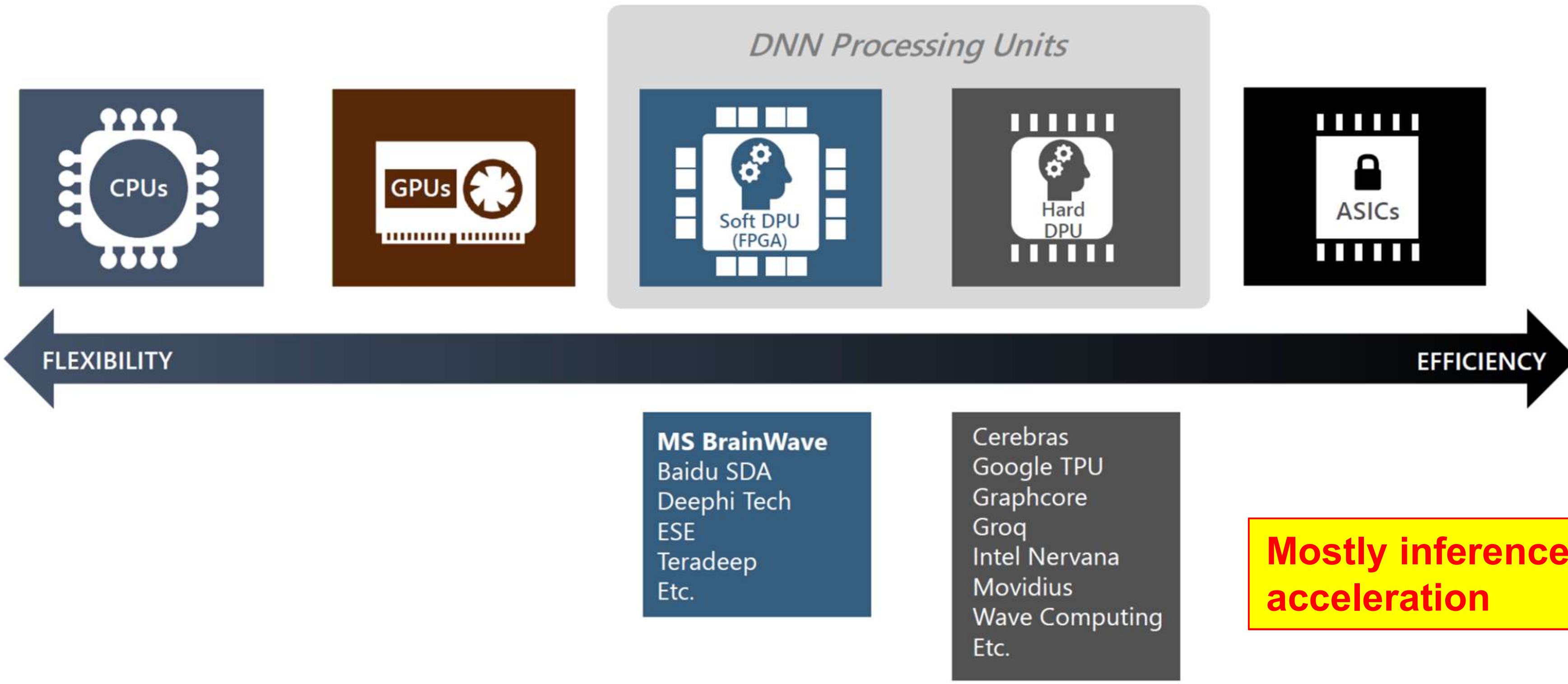
Pascal GP100
3840 "cores"

3840 MAC/cycle
@ 1.4GHz

5.3 TMACS (FP)
@300W

*Volta with tensor engine claims 4x better E

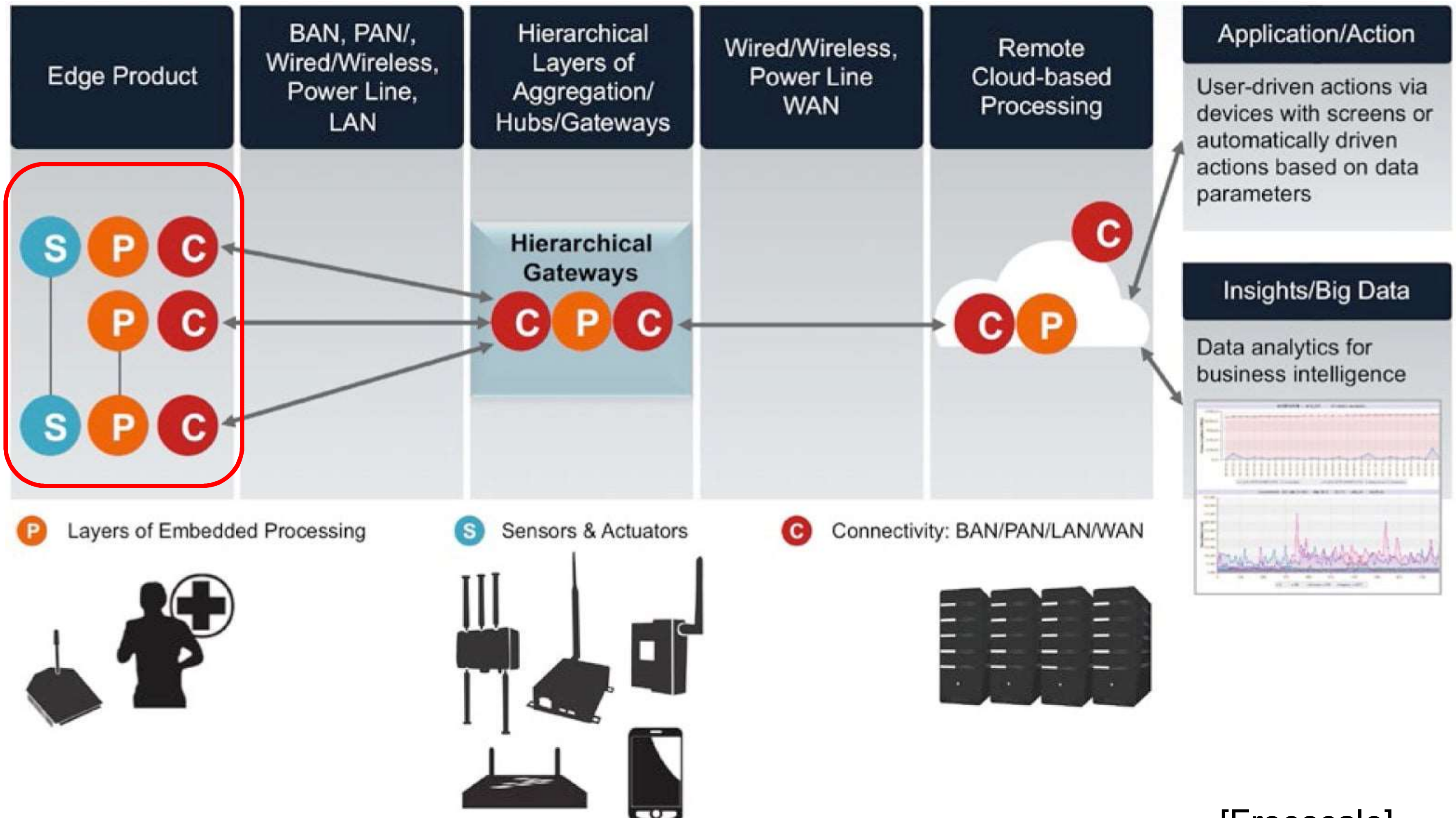
HW for deep Networks: Frenzy



Mostly inference acceleration

Datacenter → High-performance embedded →Mobile

Information Processing in the IoT



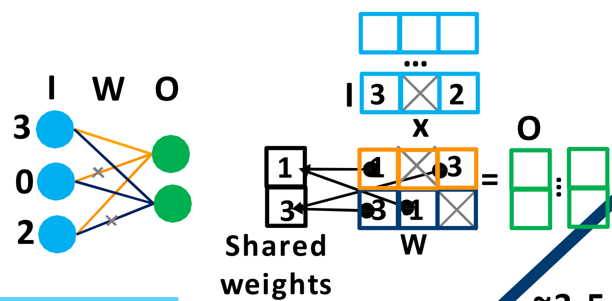
[Freescale]

Algorithmic Opportunities

DNNs Are Evolving Rapidly

Many efforts to improve efficiency

- Batching
- Reduce bitwidth
- Sparse weights
- Sparse activations
- Compression
- Compact network



All applicable for inference
Some for training

SqueezeNet + DeepCompression:
6-bit, 20-50% sparse
AlexNet accuracy, ~500x smaller (0.5MB)

XNORnet (1-bit) → ~2% AlexNet

TernaryNet (2-bit, 50% sparse) → ~1% ResNet

~3.5 years

~1 year

Spatially SparseCNN [CIFAR-10 winner '14]

BinaryConnect [NIPS'15]
SparseCNN [CVPR'15]
Pruning [NIPS'15]
HashedNets [ICML'15]

XNORNet
TernaryConnect [ICLR'16]
DeepComp [ICLR'16]
SqueezeNet

	Before 2000	2012	2014	2014	2015
	LeNet5	AlexNet (~80% Top5)	VGG (~89% top5)	GoogLeNet (~89% top5)	ResNet (~94% top5)
Deeper	5 layers	8 layers	19 layers	22 layers	152 layers
More params?	Params: 1M	Params: 60M	Params: 140M	Params: 6M	Params: 60M
Larger model?	Model: 4MB	Model: 240MB	Model: 500MB	Model: 24MB	Model: 240MB

Before 2000 — 2012 — 2013 — 2014 — 2015 — 2016

[Intel FPGAS 2017]

Orders of magnitude compute effort an memory reduction with no loss in accuracy

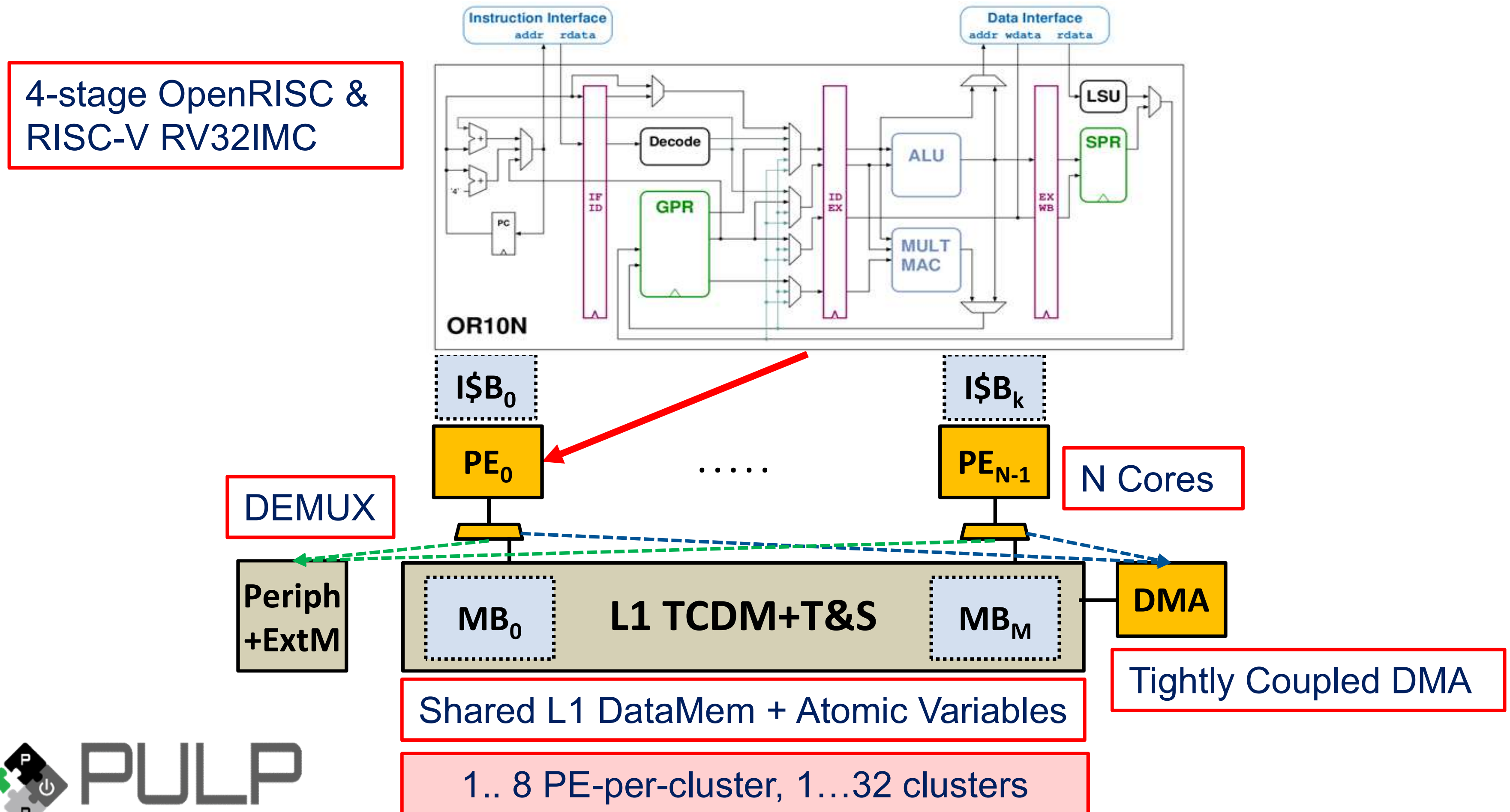
Toward Micropower CNN HW

Outline

- **Near Threshold Multiprocessing**
- Non-Von Neumann Accelerators
- Aggressive Approximation
- From Frame-based to Event-based Processing
- Outlook and Conclusion

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Near-Threshold Multiprocessing



PULP
Parallel Ultra Low Power

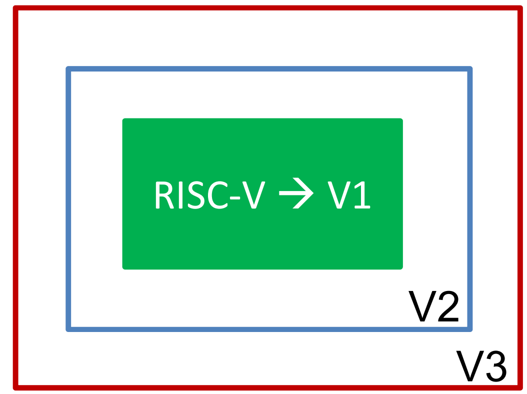
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Extending RISC-V for CNNs

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

<32-bit precision → SIMD2/4 opportunity

1. Dot product between SIMD vectors
2. Shuffle operations for vectors
3. Packed-SIMD ALU operations
4. Bit manipulations
5. Rounding and Normalizazion



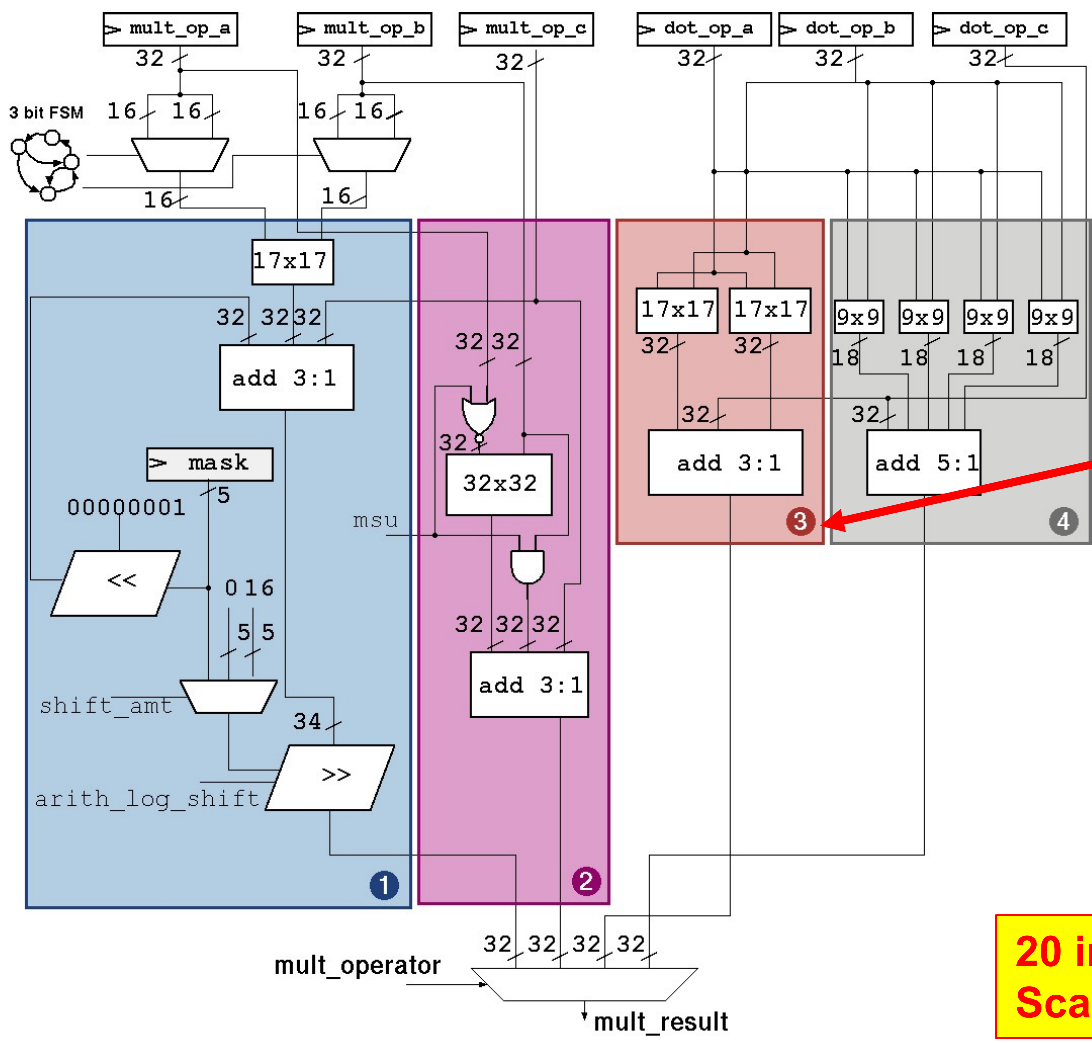
V1 Baseline RISC-V RV32IMC

V2 HW loops
Post modified Load/Store
Mac

V3 SIMD 2/4 + DotProduct + Shuffling
Bit manipulation unit
Lightweight fixed point

Small Power and Area overhead → Energy reduction in NT >3x

PULP-CNN ISA-Extensions



- The Dot-Product instruction:
- Hardware-loops eliminate loop overhead

Sum-of-dot-product units:

8b version

16b version

1 cycle execution

- 7 Sum-of-dot-product
- 4 move
- 1 shuffle
- 3 lw/sw
- ~ 5 control instructions

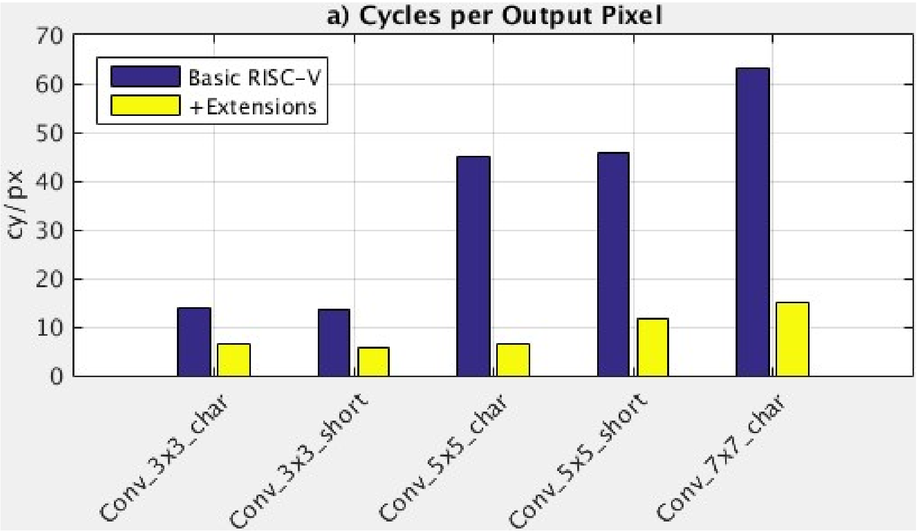
20 instr. / output pixel
Scalar version >100 instr. / output pixel

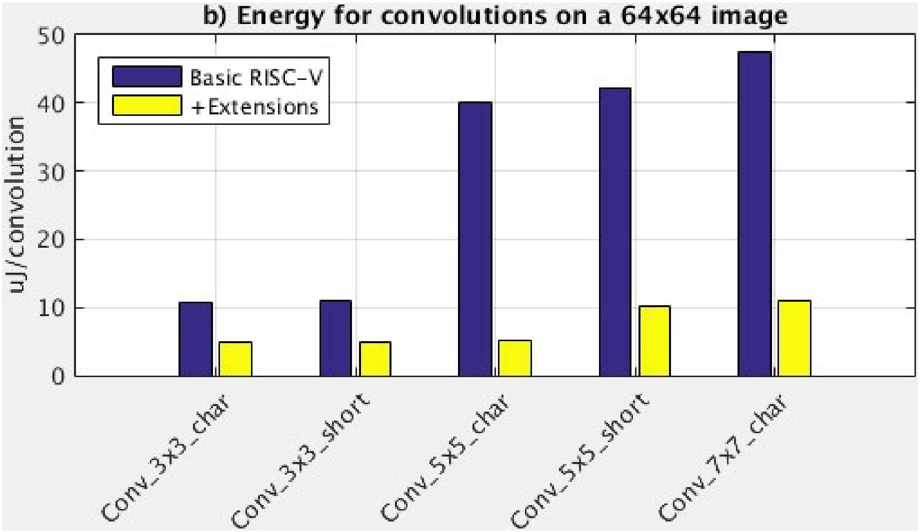
PULP-CNN ISA-Extensions

Convolution Performance on PULP with 4 cores

speedup: 3.9x

Energy gains: 4.2x



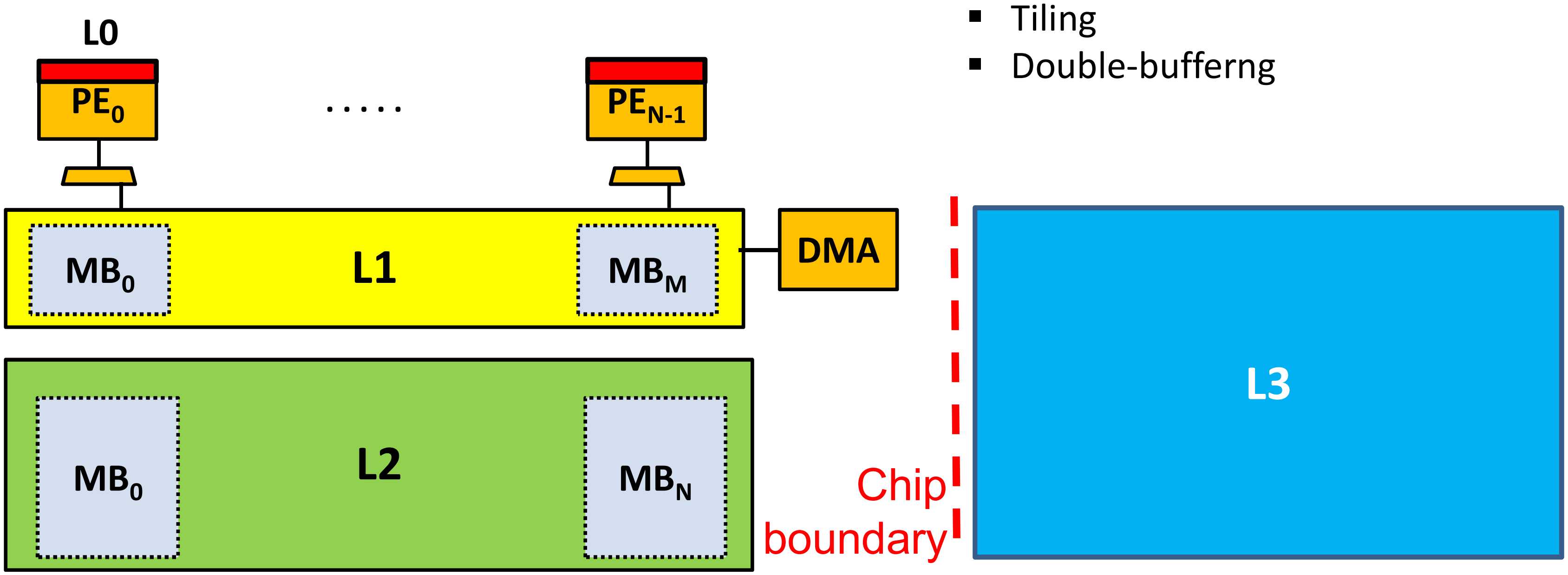


5x5 convolution in only 6.6 cycles/pixel

The Memory Optimizazion Challenge

```
for (c = 0; c < C; c++) {                        // L4 : C iterations
  for (m = 0; m < M; m++) {                      // L3 : M iterations
    for (i = 0; i < H; i += S) {                 // L2 : E iterations
      for (j = 0; j < H; j += S) {               // L1 : E iterations
        for (k = 0; k < R; k++) {                // L0 : R iterations
          for (l = 0; l < R; l++) {
            p = I[c][i+k-R/2][j+k-R/2]
            w = W[m][c][k][l]
            O[m][i][j] += p*w
          }
        }
      }
    }
  }
}
```

The canonical CNN 6-loop nest



- Memory hierarchy optimization
  - Sizing
  - Banking factors
- Data Movement engines
  - Shuffle instructions
  - DMA, prefetchers
- Loop optimizations
  - Reordeding
  - Tiling
  - Double-bufferng

Outline

- Near Threshold Multiprocessing
- **Non-Von Neumann Accelerators**
- Aggressive Approximation
- From Frame-based to Event-based Processing
- Outlook and Conclusion

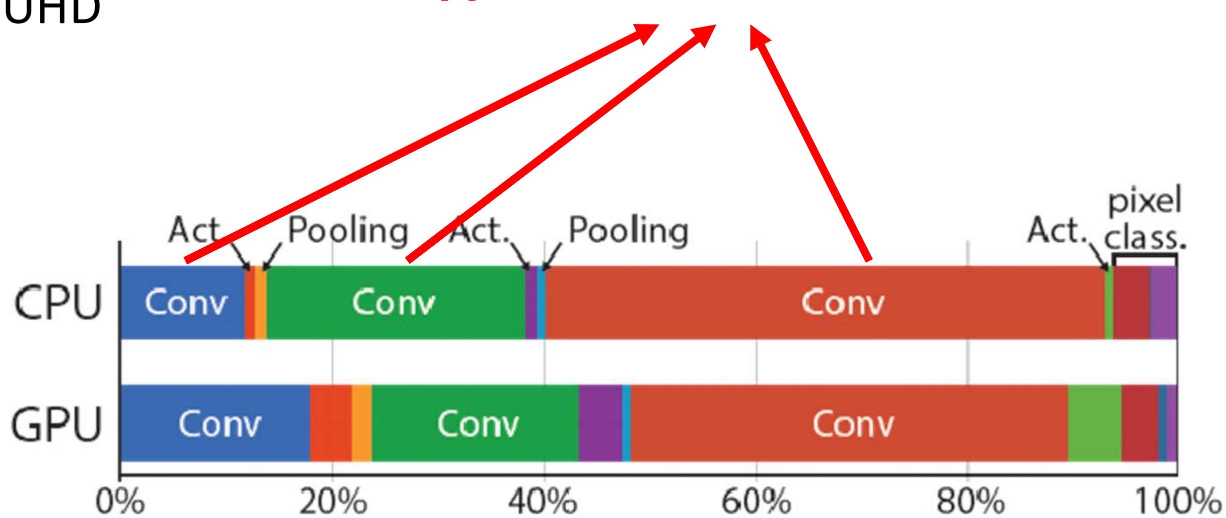
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Computational Effort

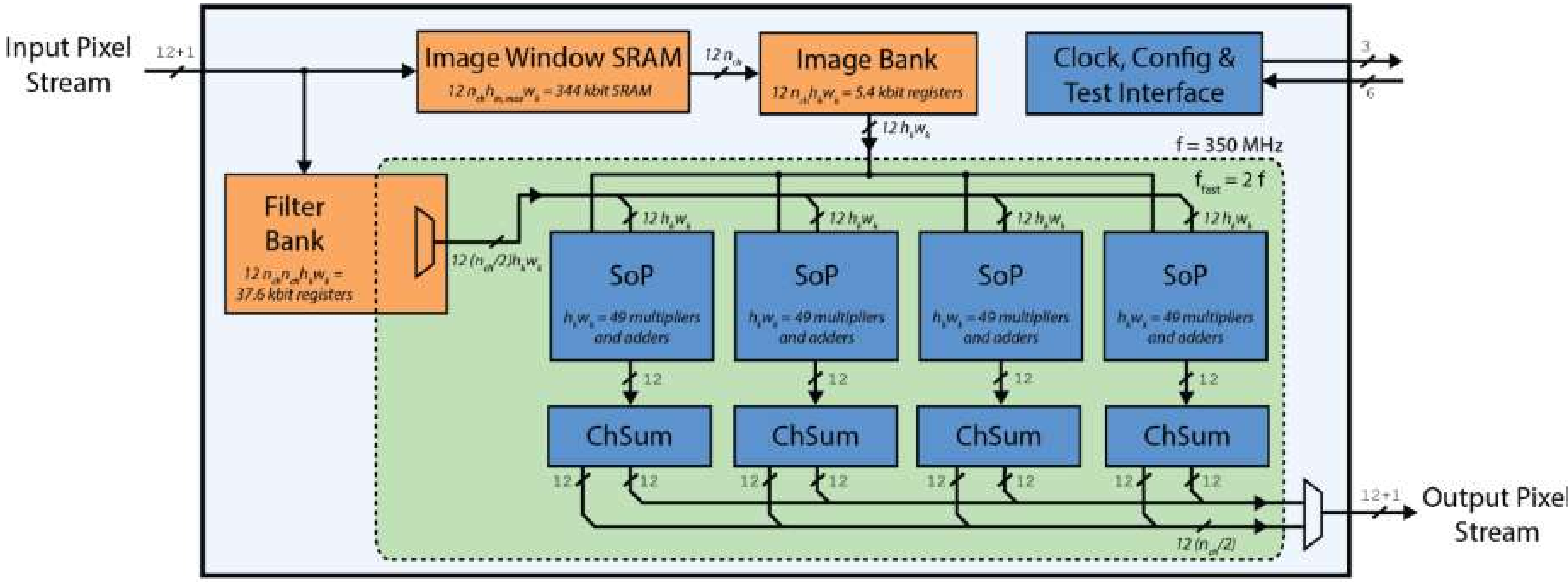
- Computational effort
  - 10-class scene labeling on Stanford-BG
  - 7.5 GOp/frame for 320x240 image (#Op=2 × #MAC)
  - 260 GOp/frame for FHD
  - 1050 GOp/frame for 4k UHD

~90% workload is Conv

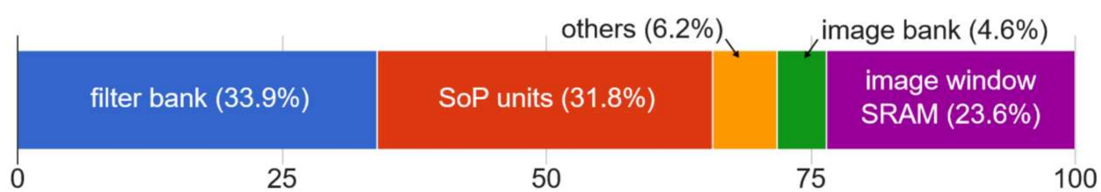


Origami CNN ASIC

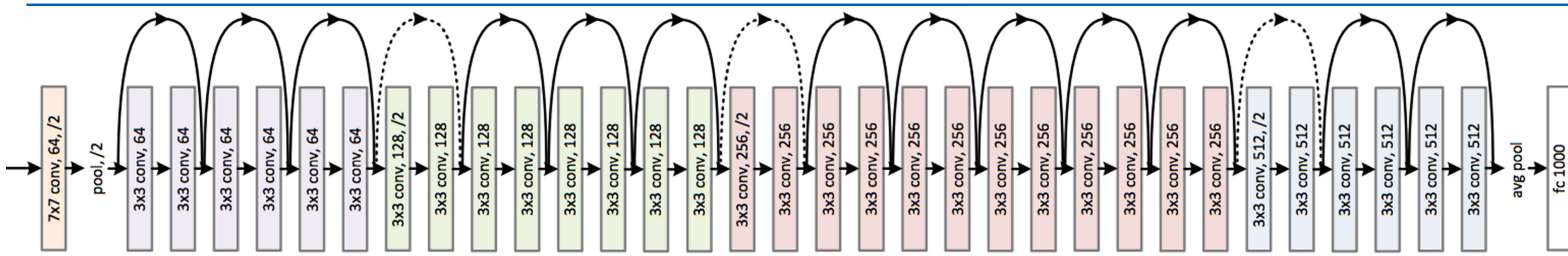
Origami: A CNN Accelerator



- **FP not needed**: 12-bit signals sufficient
- Input to classification double-vs-12-bit accuracy loss $< 0.5\%$ (80.6% to 80.1%)



CNNs: typical workload



Example: **ResNet-34**

- classifies 224x224 images into 1000 classes
- ~ trained human-level performance
- ~ **21M** parameters
- ~ **3.6G MAC** operations

Scaling Origami to 28nm FDSOI

Performance for 10 fps: **~73 GOPS/s**

Energy efficiency: **~2300 GOPS/W** efficiency

0.4pj/OP

Origami core in 28nm FDSOI → 10 fps ResNet-34 with ~32mW

Outline

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- **Aggressive Approximation**
- From Frame-based to Event-based Processing
- Outlook and Conclusion

Pushing Further: YodaNN[1]

- Approximation at the algorithmic side → Binary weights
- BinaryConnect [Courbariaux, NIPS15], XOR NET [Rastegari, arXiv16]
  - Reduce weights to a binary value -1/+1
  - Stochastic Gradient Descent with Binarization in the Forward Path

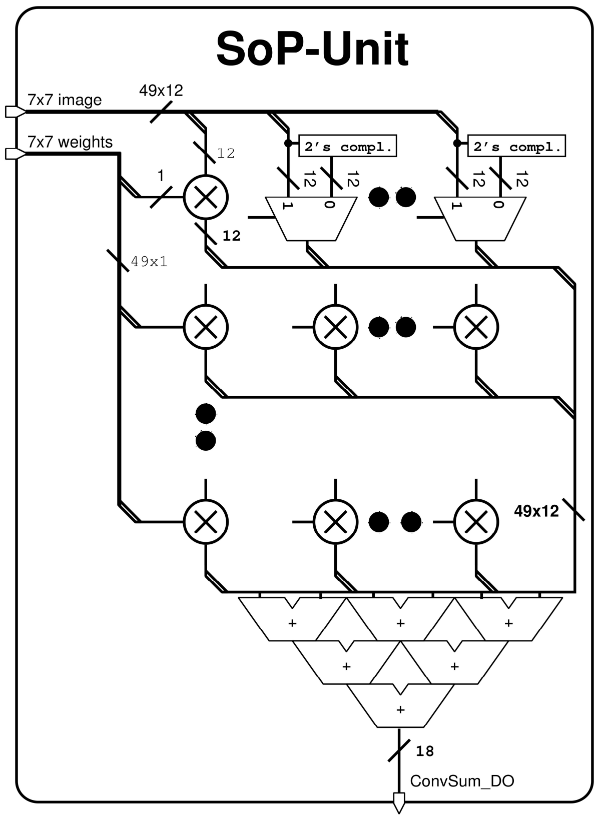
$$w_{b,stoch} = \begin{cases} -1 & p_{-1} = \sigma(w) \\ 1 & p_1 = 1 - p_{-1} \end{cases} \qquad w_{b,det} = \begin{cases} -1 & w < 0 \\ 1 & w > 0 \end{cases}$$

  - Learning large networks is still challenging, but starts to become feasible:

    ResNet-18 on ImageNet with **83.0%** (binary-weight) vs. **89.2%** (single-precision) top-5 accuracy; and **60.8%** vs. **69.3%** top-1 accuracy
- Ultra-optimized HW is possible!
  - Major arithmetic density improvements: MAC → 2s compl. & Accum.
    - Area can be used for more energy-efficient weight storage
  - Storage reduction → SCM memories for lower voltage → E goes with $1/V^2$

[1]After the Yedi Master from Star Wars - "Small in size but wise and powerful" cit. www.starwars.com

SoP-Unit Optimization



Equivalent for 7x7 SoP

ImageBank

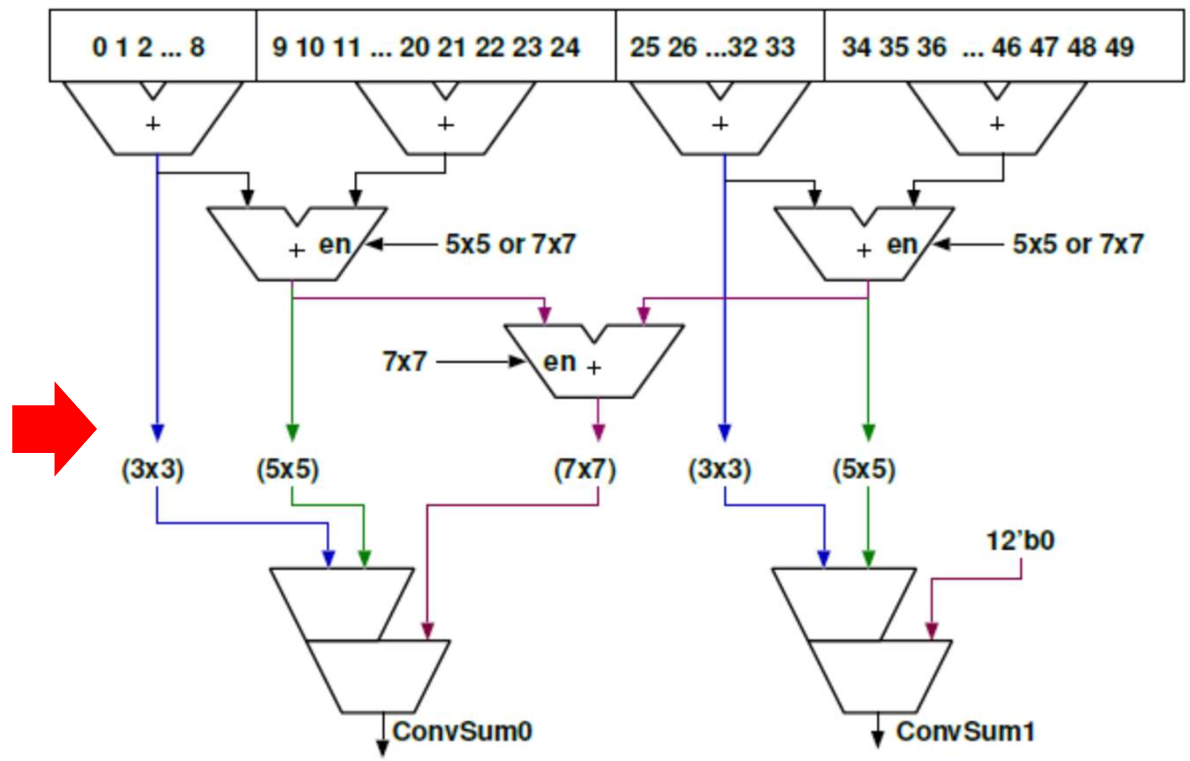
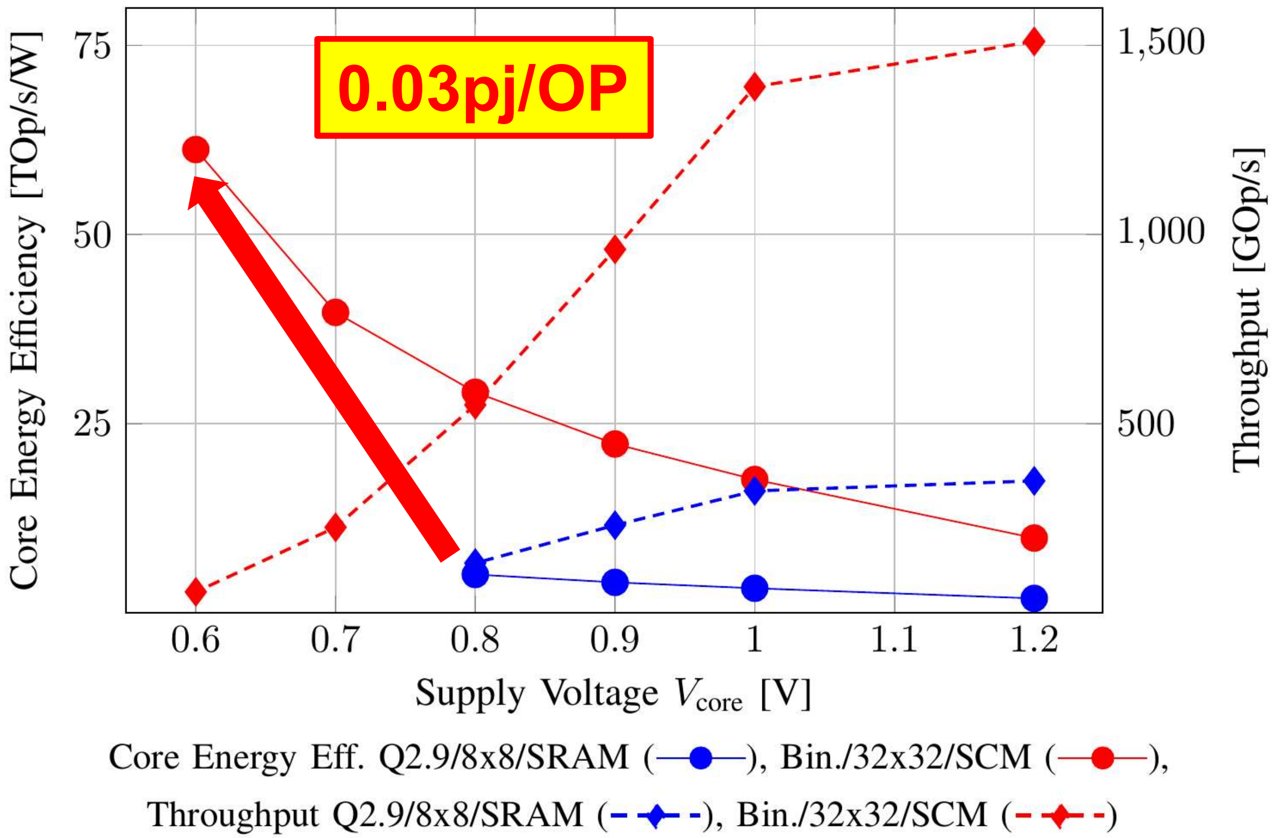


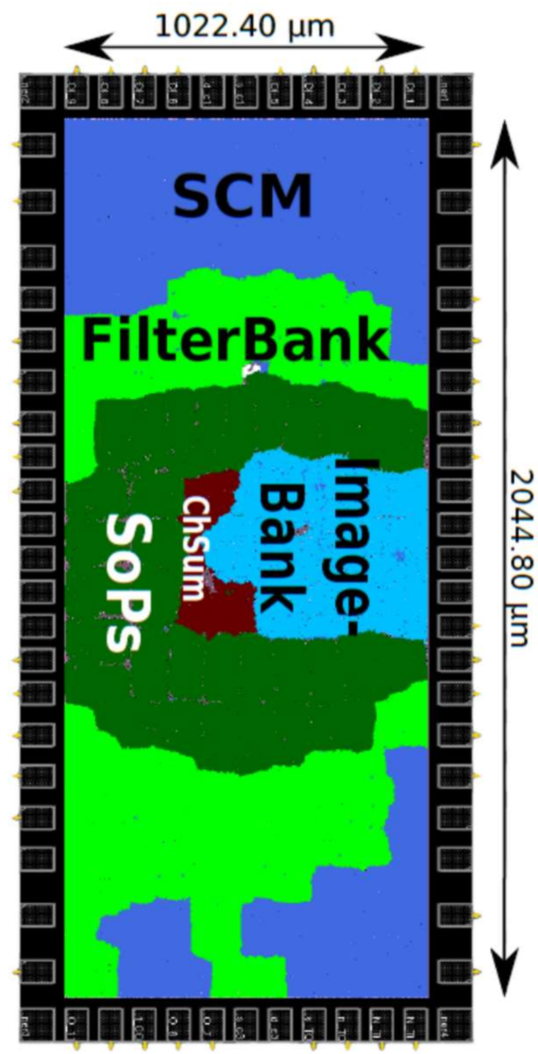
Image Mapping (3x3, 5x5, 7x7)

1 MAC Op = 2 Op (1 Op for the “sign-reverse”, 1 Op for the add).

YODANN Energy Efficiency

Same area 8→32 SoP units + all SCM





12x Boost in core energy efficiency (single layer)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

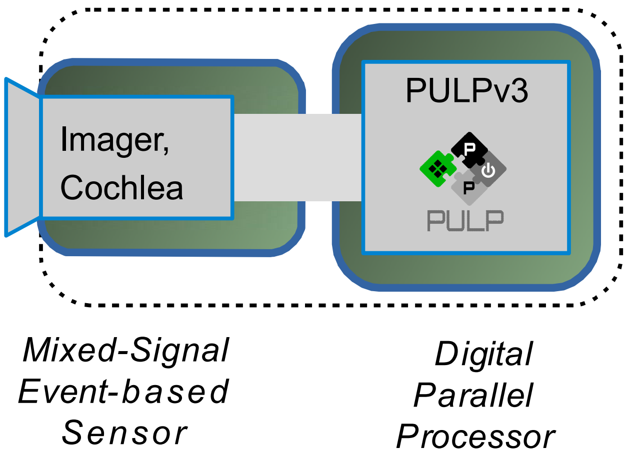
Outline

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- Aggressive Approximation
- **From Frame-based to Event-based Processing**
- Outlook and Conclusion

Back to System-Level

Smart Visual Sensor→ idle most of the time (nothing interesting to see)



- **Event-Driven Computation**, which occurs only when relevant events are detected by the sensor
- **Event-based sensor interface** to minimize IO energy (vs. Frame-based)
- **Mixed-signal event triggering** with an ULP imager, cochlea with internal processing AMS capability

A Neuromorphic Approach for doing *nothing* VERY well

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

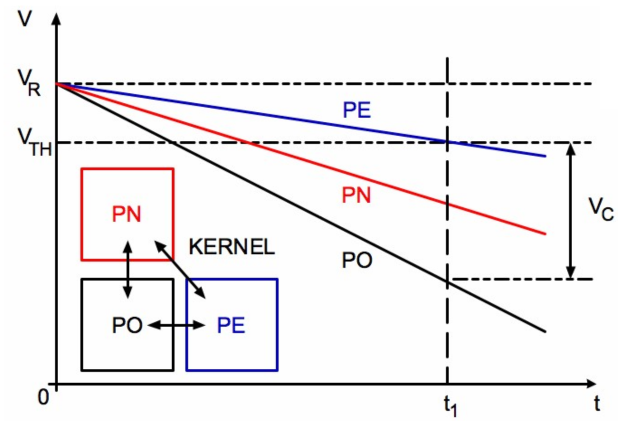
GrainCam Imager

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

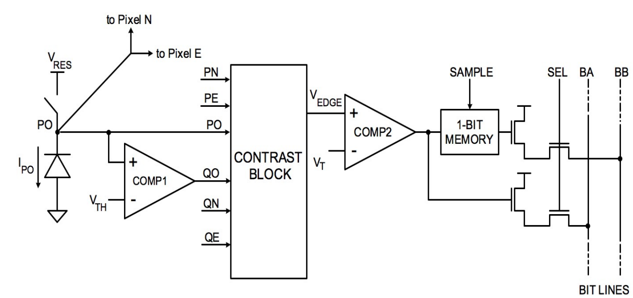
FONDAZIONE BRUNO KESSLER

[Gottardi, JSSC09]

Pixel-level spatial-contrast extraction



$$V_C = V_{PE}(t_1) - V_{PO}(t_1) = (V_R - V_{TH})\left(\frac{I_{PO} - I_{PE}}{I_{PE}}\right)$$



Analog internal image processing

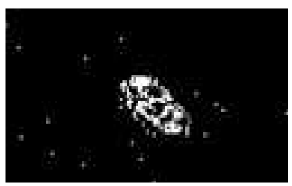
- Contrast Extraction
- Motion Extraction, differencing two successive frames
- Background Subtraction with the reference image stored in pixel memory

Graincam Readout

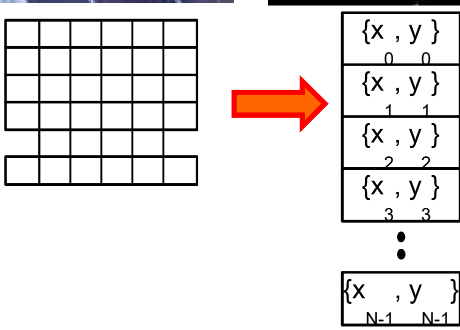
Readout modes:

- IDLE: readout the counter of asserted pixels
- ACTIVE: sending out the addresses of asserted pixels (address-coded representation), according raster scan order

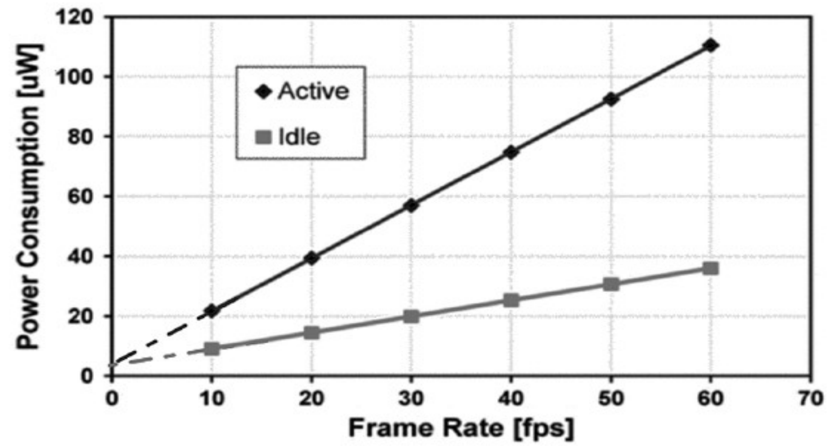
Event-based sensing: output frame data bandwidth depends on the external context-activity



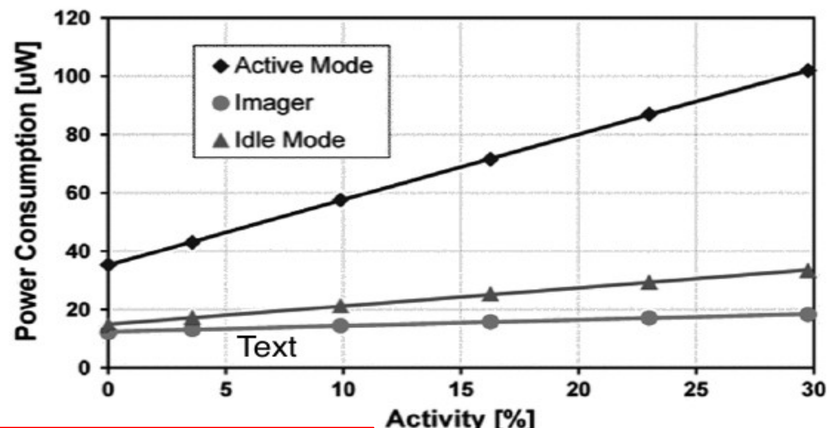
Frame-based



Event-based



(a)



(b)

Ultra Low Power Consumption e.g. 10-20uW @10fps

Even-driven CNNs? Yes!

Binary Neural Networks reduce precision of weights and post-activation neurons to 1-bit precision while leading to a limited performance drop

Benefits:

Reduced storage costs by **32x** either for synaptic weights and temporary input/output data

Reduced computation complexity

weight filter

1	0	0
0	1	1
0	1	0

Convolution reduce to:

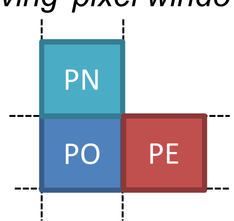
C = popcount (NOT (***weights*** XOR ***image***))

Binarization:

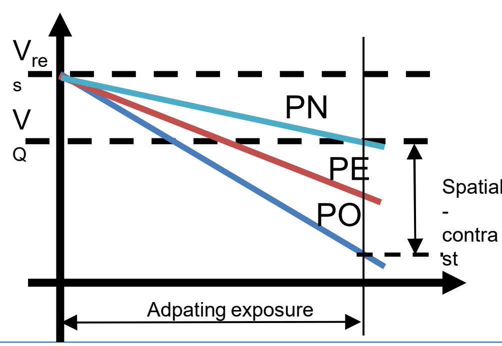
output = **C >0** ? **1** else **0**

Performing spatial filtering and binarization on the sensor die through mixed-signal sensing! → in-sensor first stage of the binary NN!!

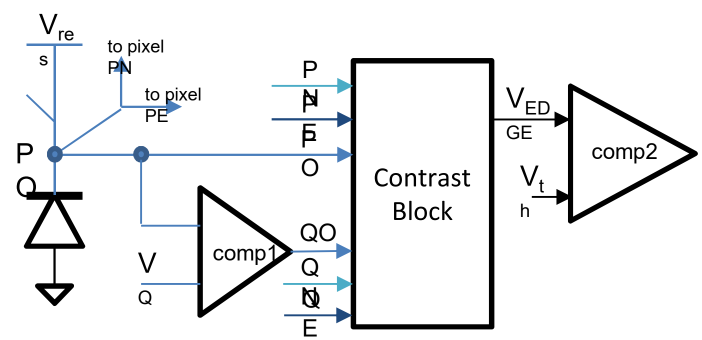
'Moving' pixel window



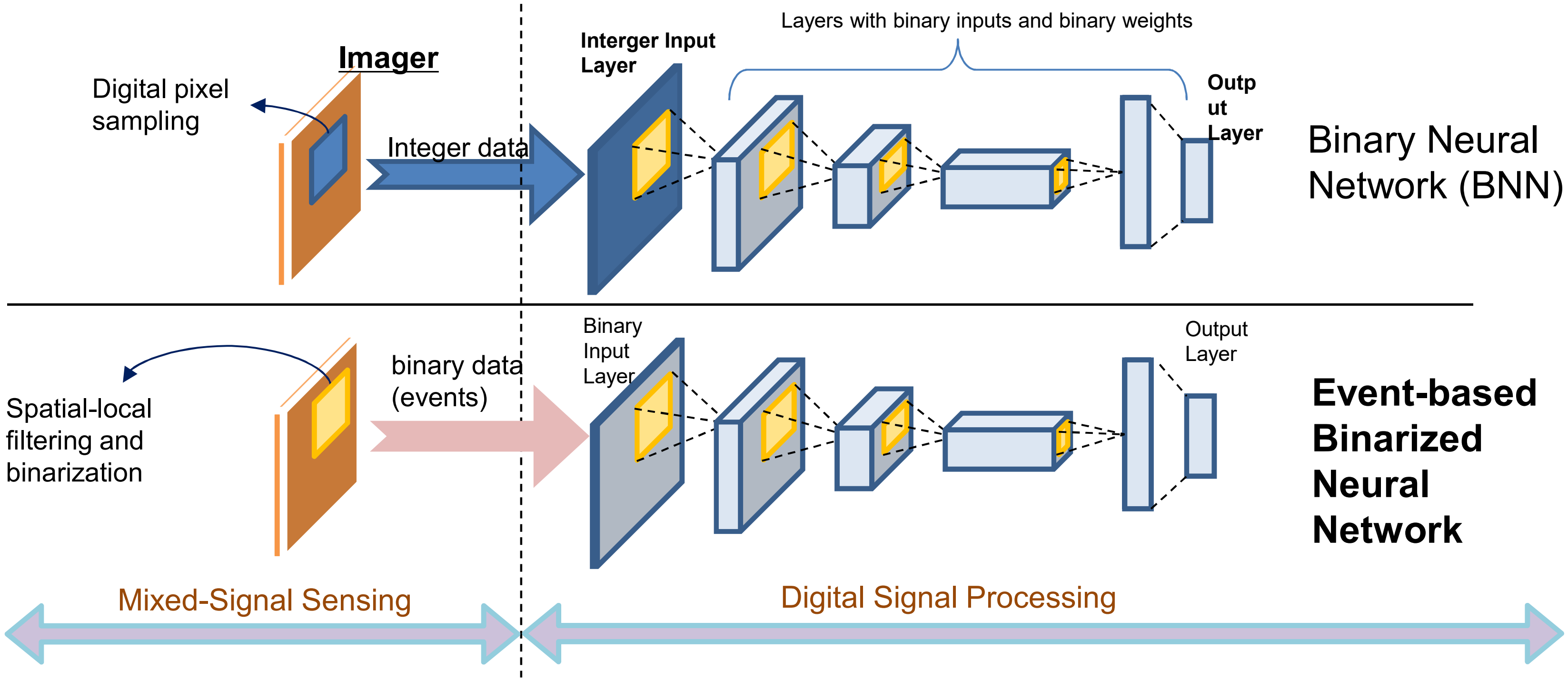
Gradient extraction



Per-pixel circitut for filtering and binarization



Event-Driven Binary Deep Network



Training challenge

Training Event-based Binarized Neural Network:

[ISSUE] Absence of huge amount of data for training

Modelling the ''graincam filter'' as a digital filter

Contrast Value $V_C \sim \frac{\max(|p_E - p_O|, |p_N - p_O|)}{\max(p_E, p_O, p_N)}$ Binary Output $V_O = sgn(V_C - V_{th})$


Original RGB image


Synthetic image


Graincam image

Evaluation on **CIFAR-10** (10 classes, 45k training, 5k valid, 10k testing)

Baseline with RGB input	92%
BNN with RGB input	86%
Baseline with binary input	72%
BNN with binary input	68%

Model VGG-like with 12 Convolutional laters and 3 Fully Connected Layers

18% performance drop because of input representation but still converges

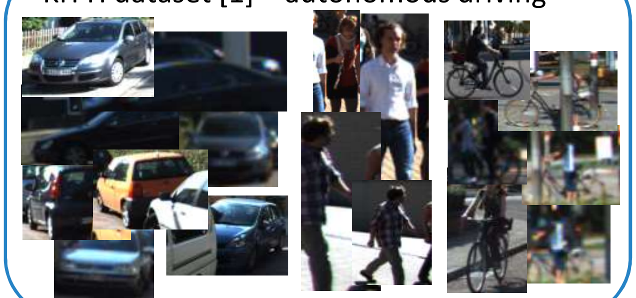
Results

TRAINING DATASETS

Training:
60k samples

Validation:
900 samples

KITTI dataset [1] – autonomous driving



MIO-TCD dataset [2] – surveillance system

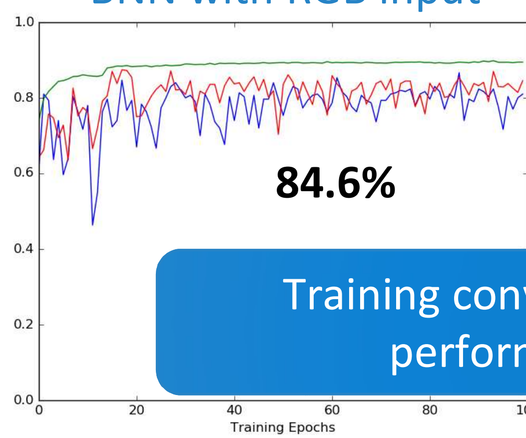
ACCURACY

Model: VGG-like architecture

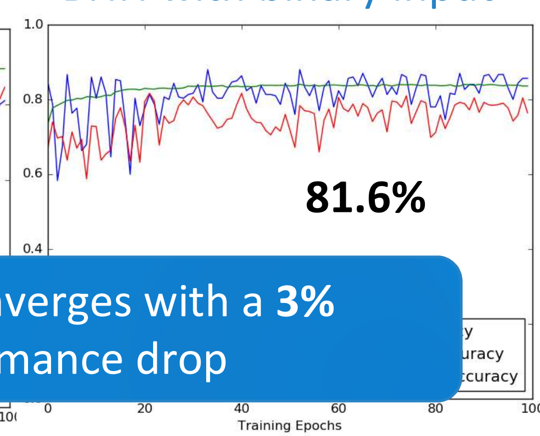
<20kB "binary weight program"

CONV3x3(#c,16) + POOLING
CONV3x3(16,32) + POOLING
CONV3x3(32,48) + POOLING
CONV3x3(48,64) + POOLING
CONV3x3(64,96) + POOLING
FULLY-CONNECTED (384,64)
FULLY-CONNECTED (64,3)

BNN with RGB input



BNN with binary input



Training converges with a **3%** performance drop

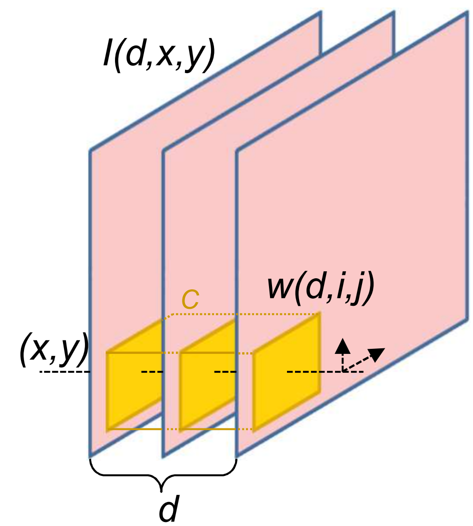
[1] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d
[2] http://podoce.dinf.usherbrooke.ca/

BNN implementation on PULP

Basic **Convolution Operation**:

$$\rho(x,y) = \sum_{(d,i,j)\in C} w(d,i,j) * I(d,i,j,x,y)$$

$I, w \in \{0,1\}$



bdot_3d

$$\boxed{\rho(x,y) = \boldsymbol{popcount}\{\, \boldsymbol{NOT}\ (w\ \boldsymbol{XOR}\ I(x,y))\,\}}$$

Batch **Normalization** and **binarization**:

$$o_z(x,y) = \frac{\rho(x,y) + b - \mu}{\sigma}\gamma + \beta \geq 0$$

$\rho(x,y) \in N$

if $\gamma \geq 0$ **then** $o_z(x,y) = \rho(x,y) \leq \left\lfloor \mu - b - \frac{\beta * \sigma}{\gamma} \right\rfloor$ **else** $o_z(x,y) = \rho(x,y) \geq \left\lceil \mu - b - \frac{\beta * \sigma}{\gamma} \right\rceil$

just logic operation and integer comparison!

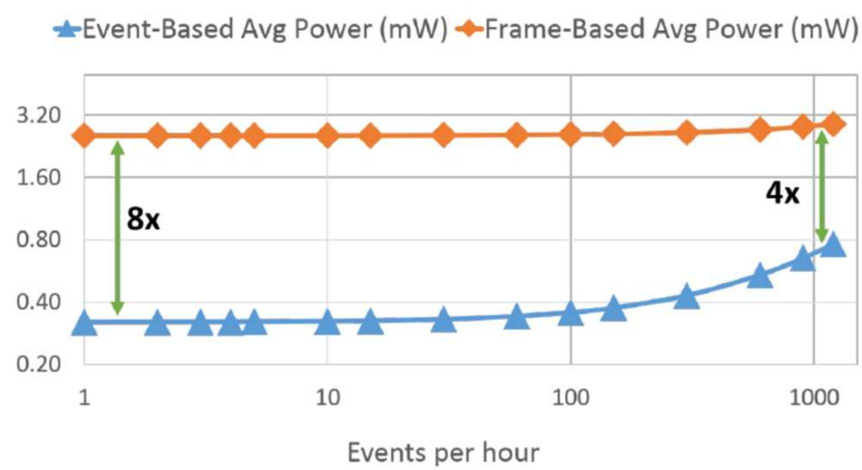
Major opportunity for HW acceleration!

Preliminary Results

Scenario	**BNN with RGB input**	**Event-based BNN**
Image Sensor Power Consumption	1.1mW @30fps	100μW @50fps
Image Size	632446 bits	8192 bits
Image Sensor Energy for frame capture	66.7 μJ	2 μJ
Transfer Time (4bit SPI @50MHz)	3.1 msec	0.04 msec
Transfer Energy (8.9mW @0.7V)	28 μJ	2 μJ
BNN Execution Time (168MHz)	81.3 msec	75.3 msec
BNN Energy consumption (8.9mW @0.7V)	725 μJ	671 μJ
Total System Energy for Classification	820 μJ	674 μJ

Statistics per frame	**Frame-Based**	**Event-based**
Idle (no motion)		
Sensor Power	1.1mW	20μW
Avg Sensor Data	19764 Bytes	-
Transfer Time	790μsec	-
Processing Time	3.02 msec	-
Avg Processor Power	1.45mW	0.3mW (sleep)
Detection		
Sensor Power	1.1mW	60μW
Avg Sensor Data	19764 Bytes	~536 Bytes
Transfer Time	790μsec	21.4μsec
Processing Time	3.47 msec	187.6μsec
Avg Processor Power	1.57mW	0.511mW
Classification		
Sensor Power	2mW	60μW
Avg Sensor Data	79056 Bytes	1024 Bytes
Transfer Time	3.16 msec	41μsec
Processing Time	81.3 msec	75.3 msec
Processor Energy	760 μJ	677 μJ

84.6% vs. 81.6% Accuracy



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Outline

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

- Near Threshold Multiprocessing
- Non-Von Neumann Accelerators
- Aggressive Approximation
- From Frame-based to Event-based Processing
- **Outlook and Conclusion**

Conclusions

- Near-sensor processing for the IoT
- CNNs can be taken into the ULP (mW power envelope) space
  - Non-von-Neumann acceleration
  - Very robust to low precision computations (deterministic and statistical)
  - fJ/OP is in sight!
- Major synthesis challenges
  - Memory optimizatiom: automatic exploration of Archi+Loop
  - Automatic precision tuning of datapath
  - Boolean training
- Open Source HW & SW approach →innovation ecosystem

Morale:
Plenty of room at the bottom

Thanks!!!

PULP
Parallel Ultra Low Power

www.pulp-platform.org
www-micrel.deis.unibo.it/pulp-project
iis-projects.ee.ethz.ch/index.php/PULP

Origami, YodaNN vs. Human

The «energy-efficient AI» challenge (e.g. Human vs. IBM Watson)

Type	Analog (bio)	Q2.9 Precision	Q2.9 Precision	Binary-Weight
Network	human	ResNet-34	ResNet-18	ResNet-18
Top-1 error [%]		21.53	30.7	39.2
Top-5 error [%]	**5.1**	5.6	10.8	17.0
Hardware	Brain	Origami	Origami	YodaNN
Energy-eff. [uJ/img]	100.000(*)	1086	543	**31**

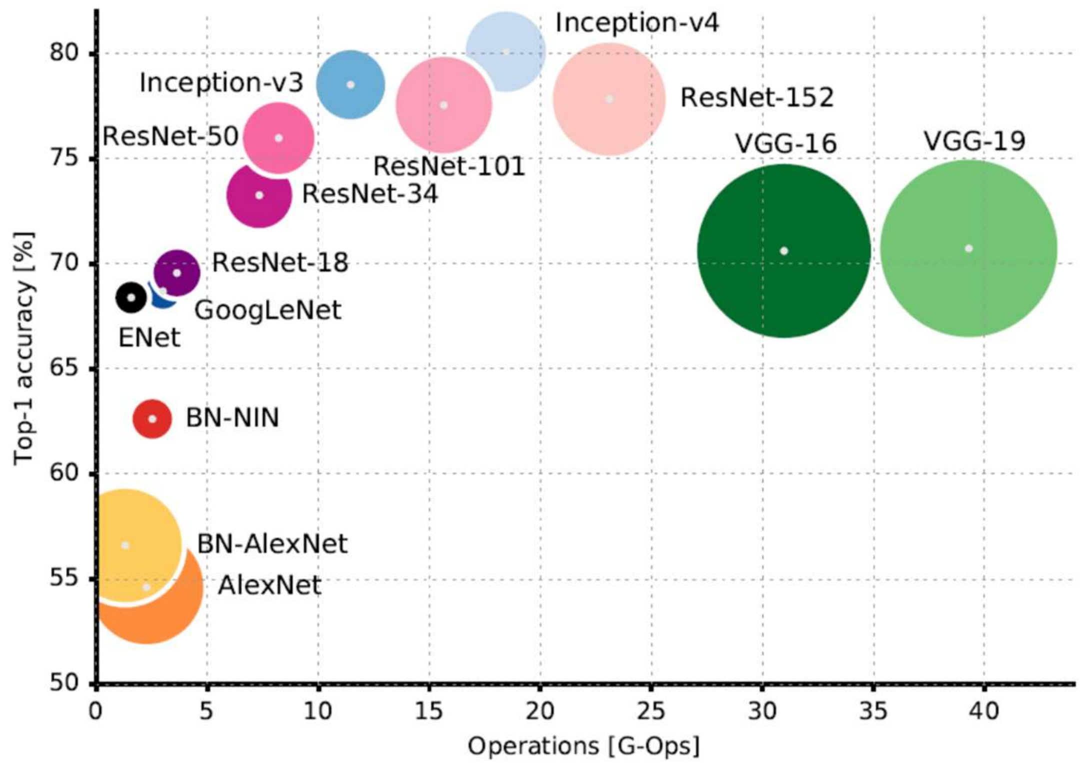
*P_{brain} = 10W, 10% of the brain used for vision, trained human working at 10img/sec

- Game over for humans also in energy-efficient vision?
- .... Not yet! (object recognition is a super-simple task)

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CNN Workloads

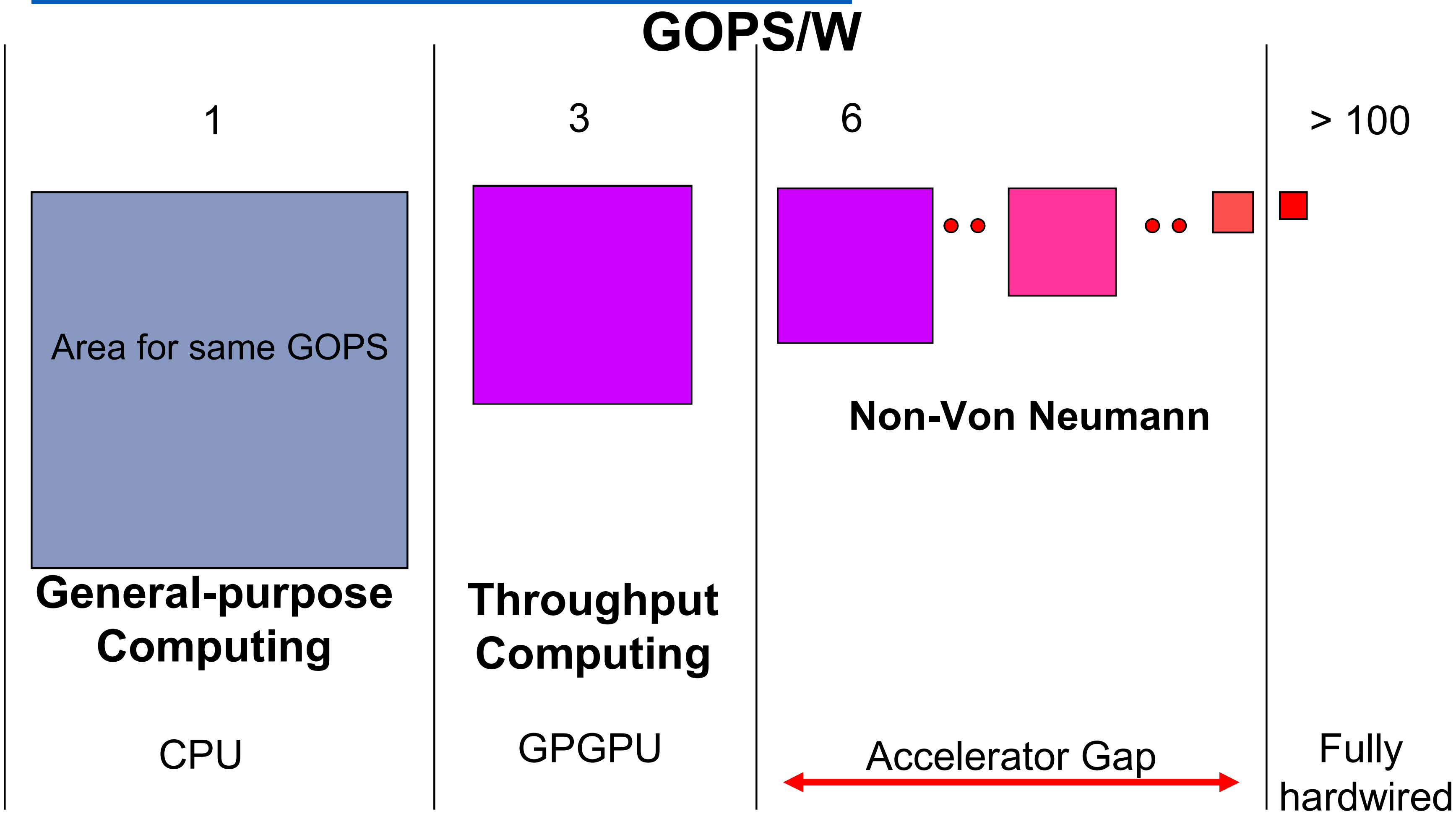
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



[Culurciello16]

Better networks are not necessarily more complex

Recovering silicon efficiency



Closing The Accelerator Efficiency Gap with Agile Customization