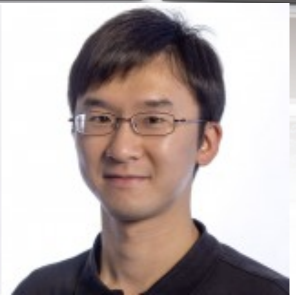
Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision

Amr Suleiman*, Yu-Hsin Chen*, Joel Emer, Vivienne Sze

Massachusetts Institute of Technology

Contact Info
email: sze@mit.edu
website: www.rle.mit.edu/eems

Amr Suleiman

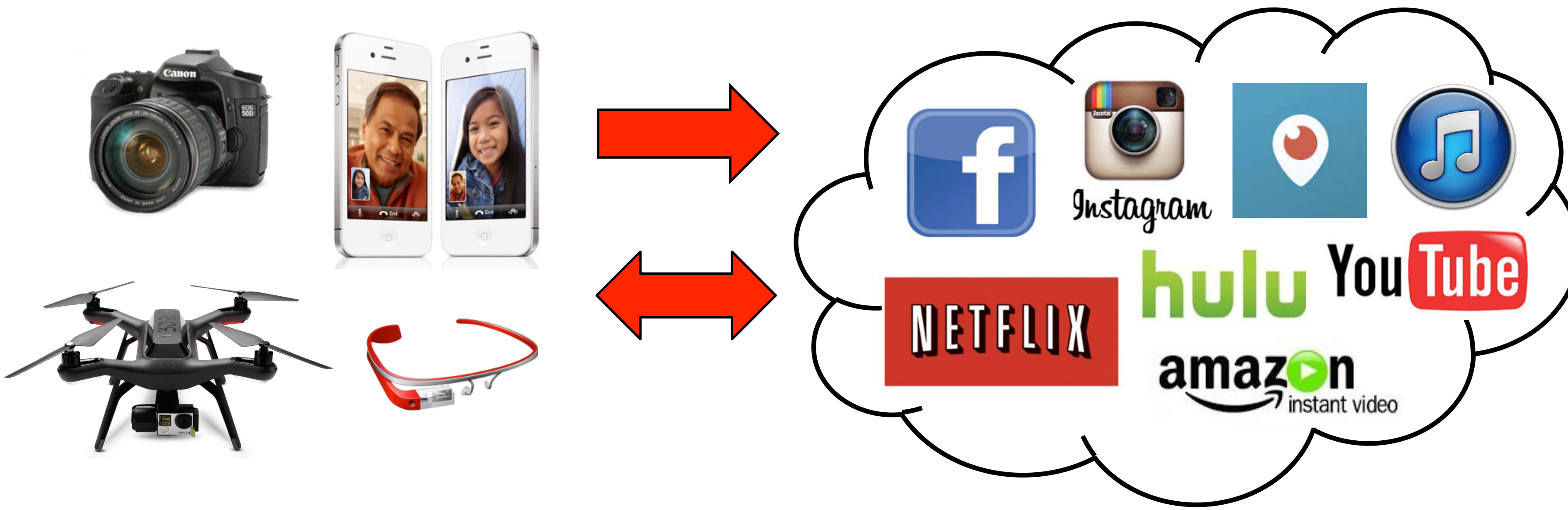


Yu-Hsin Chen

MIT
RESEARCH LABORATORY OF ELECTRONICS AT MIT
microsystems technology laboratories massachusetts institute of technology

Video is the Biggest Big Data

Over 70% of today's Internet traffic is video
Over 300 hours of video uploaded to YouTube **every minute**
Over 500 million hours of video surveillance collected **every day**



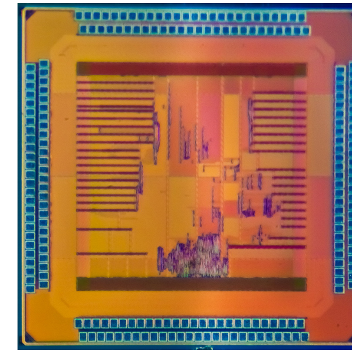
Energy limited due to battery capacity

Power limited due to heat dissipation

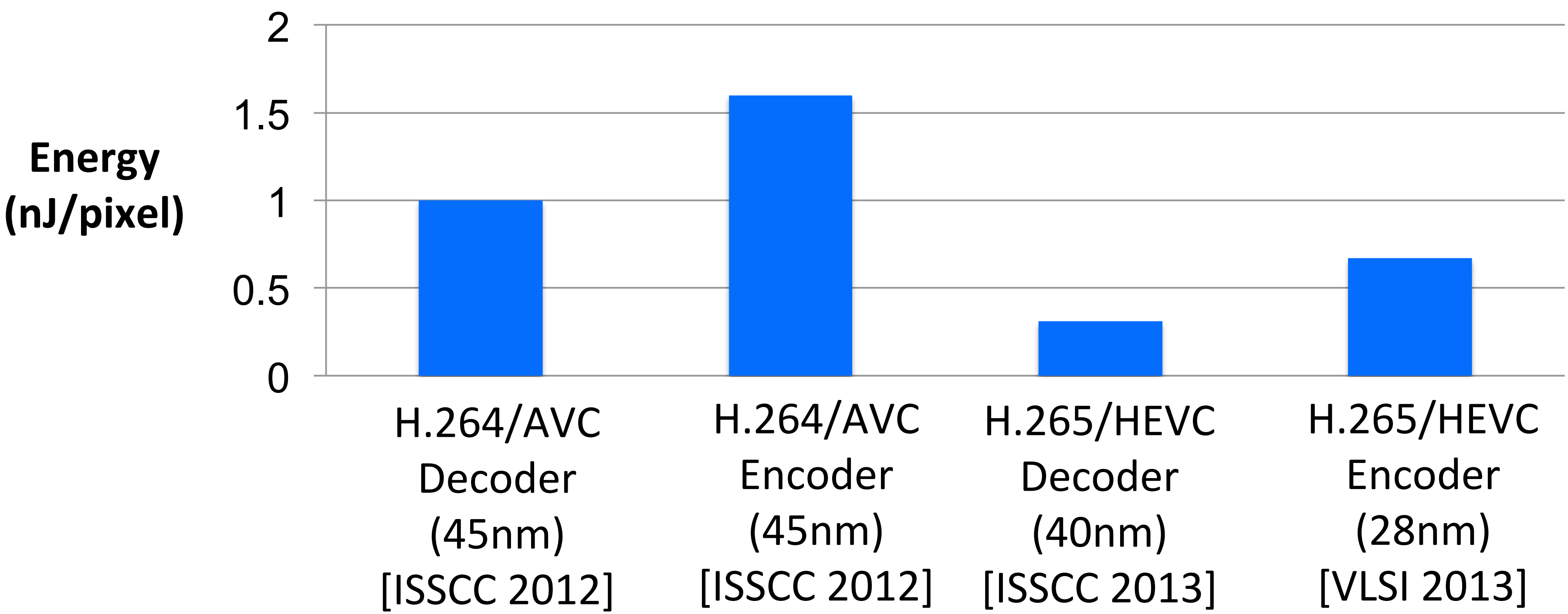
Need energy-efficient pixel processing!

Typical Constraints on Video Compression

- **Area cost:** Memory Size of 100-500kB, ~1000kgates
- **Power budget:** < 1W for smartphones
- **Throughput:** Real-time 30 fps
- **Energy:** ~1nJ/pixel



[ISSCC 2014]

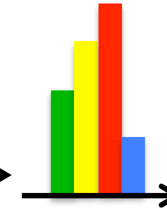


Object Detection/Classification Pipeline

Image

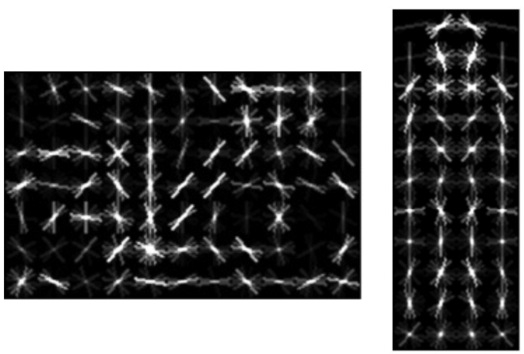
pixels → **Feature Extraction** → Features (**x**) → **Classification** ***($w^T x$)*** → Scores

Trained weights (**w**) → **Classification**



Scores per class
(select class based on max or threshold)

Handcrafted Features (e.g. HOG)

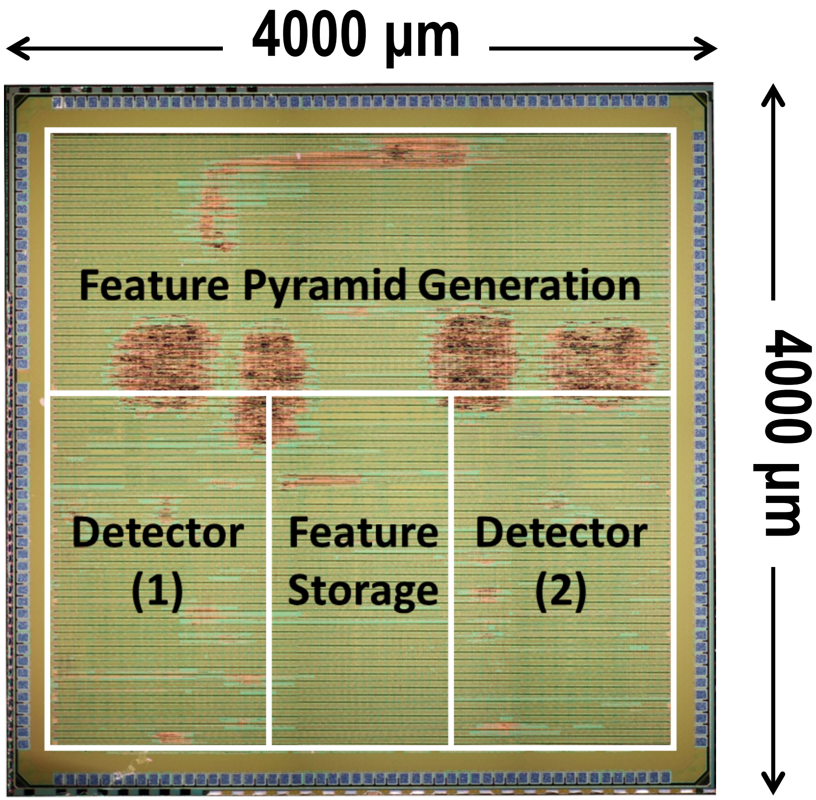


Learned Features (e.g. CNN)

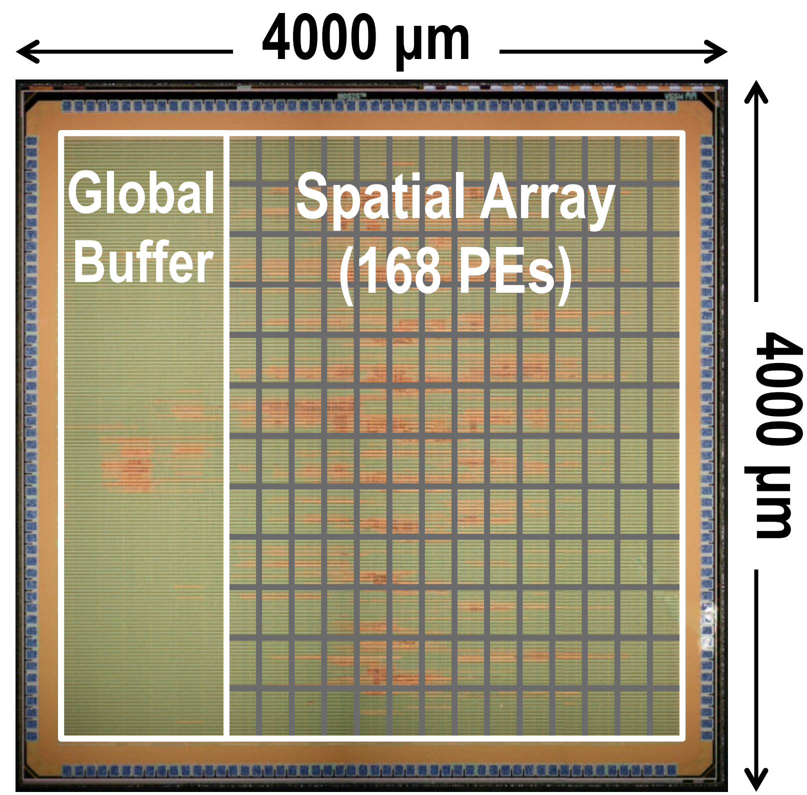
This talk will focus on the **Feature Extraction** cost

Compare HOG vs. CNN

Compare using measured results from test chips (65 nm)



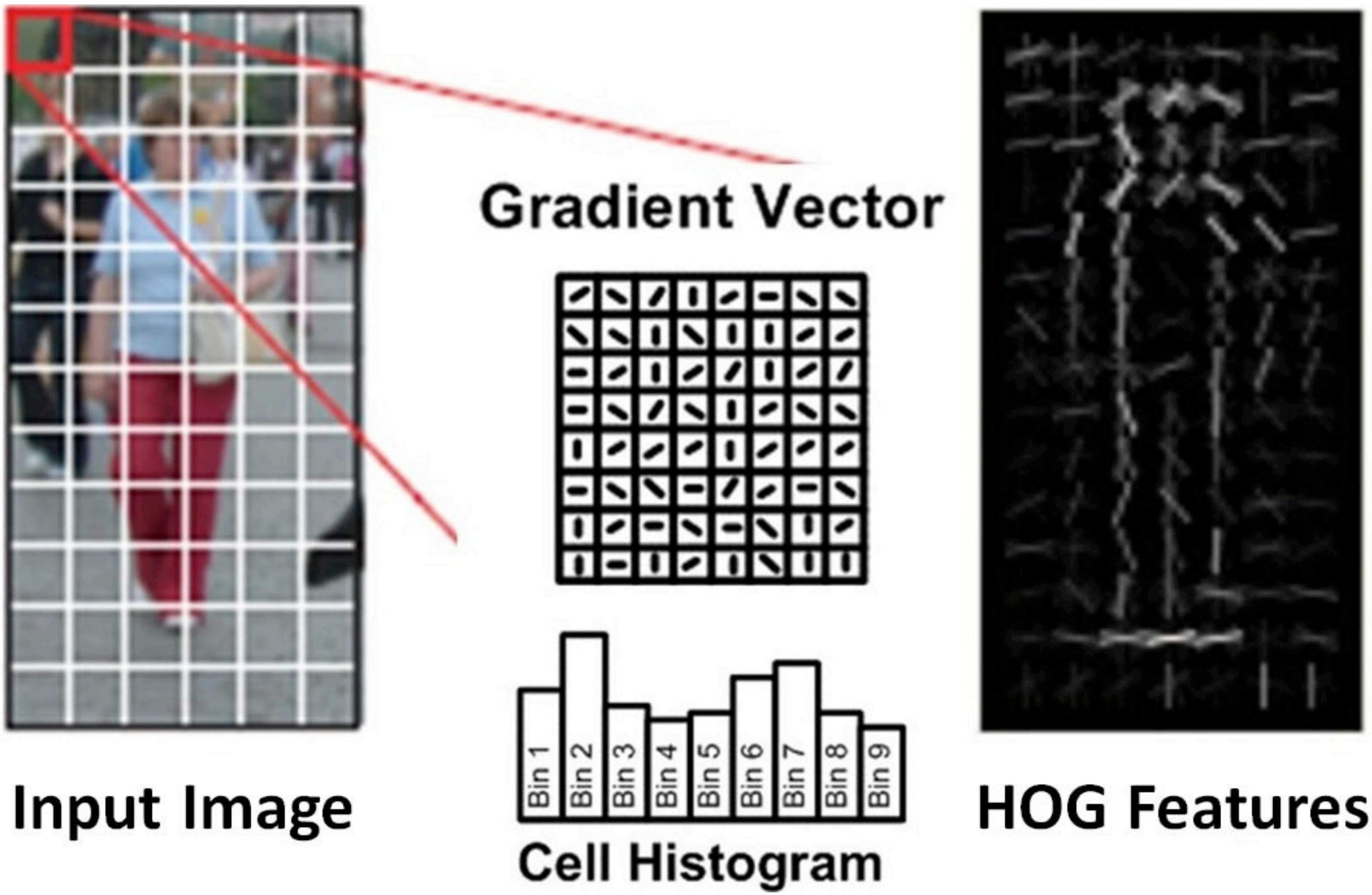
Object Detection using **HOG features** and Deformable Parts Models [VLSI 2016]



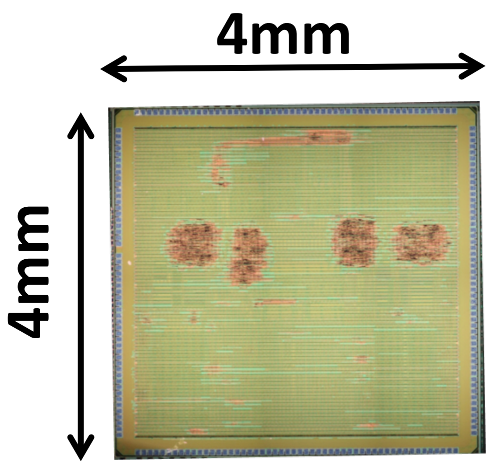
Eyeriss: **Convolutional Neural Networks** [ISSCC 2016, ISCA 2016]

Hand-crafted Features (HOG)

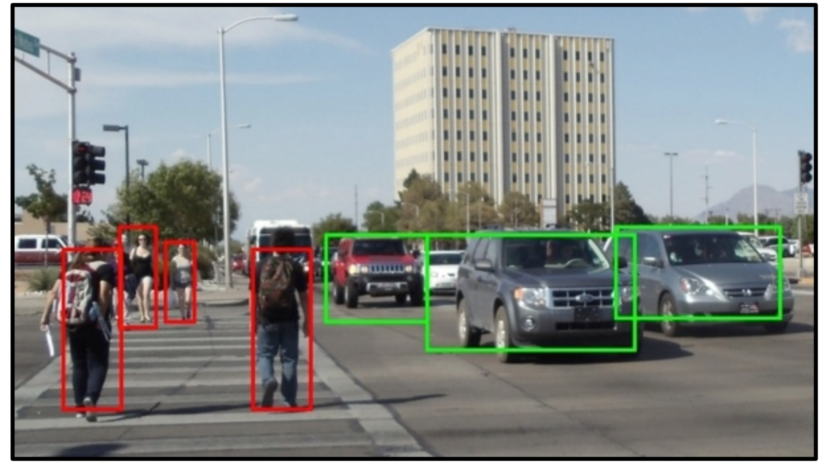
HOG = Histogram of Oriented Gradients



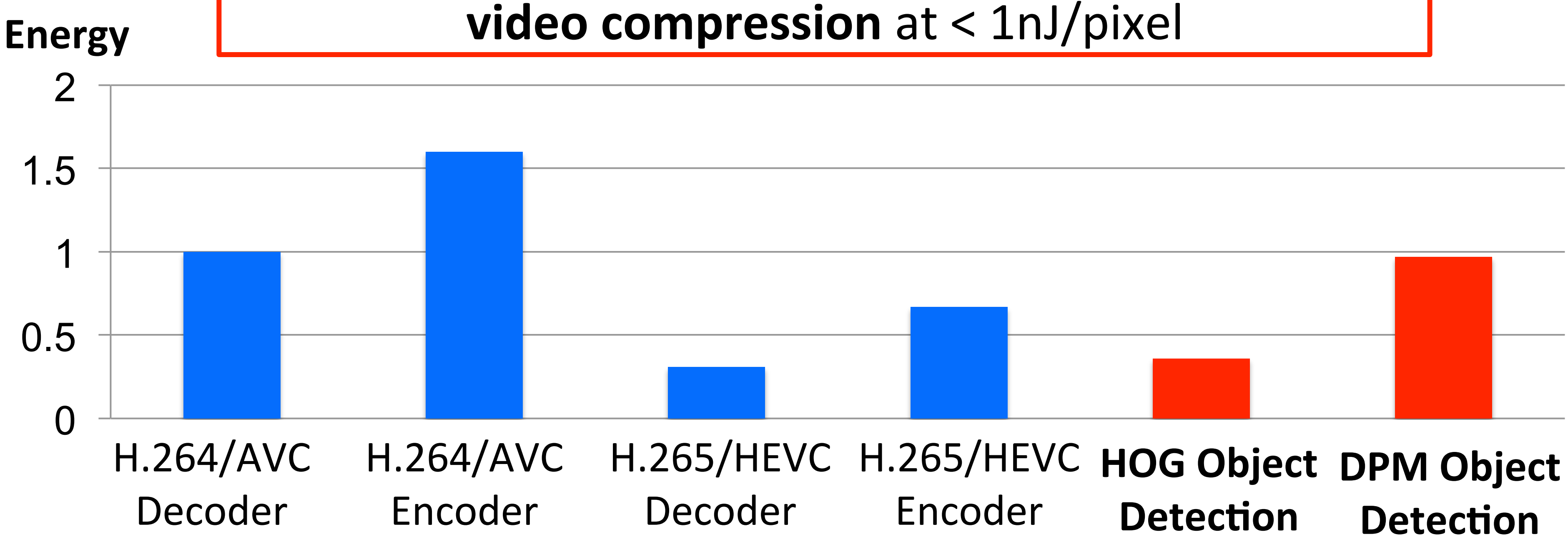
Energy-Efficient Object Detection



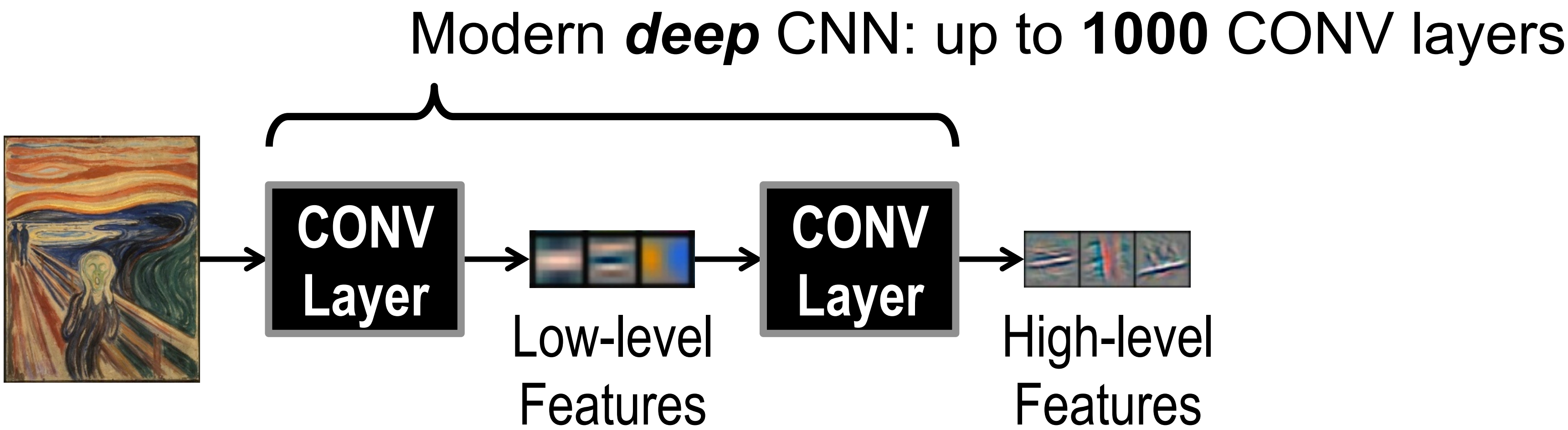
MIT Object Detection Chip [VLSI 2016]



Enable **object detection** to be as **energy-efficient** as **video compression** at < 1nJ/pixel

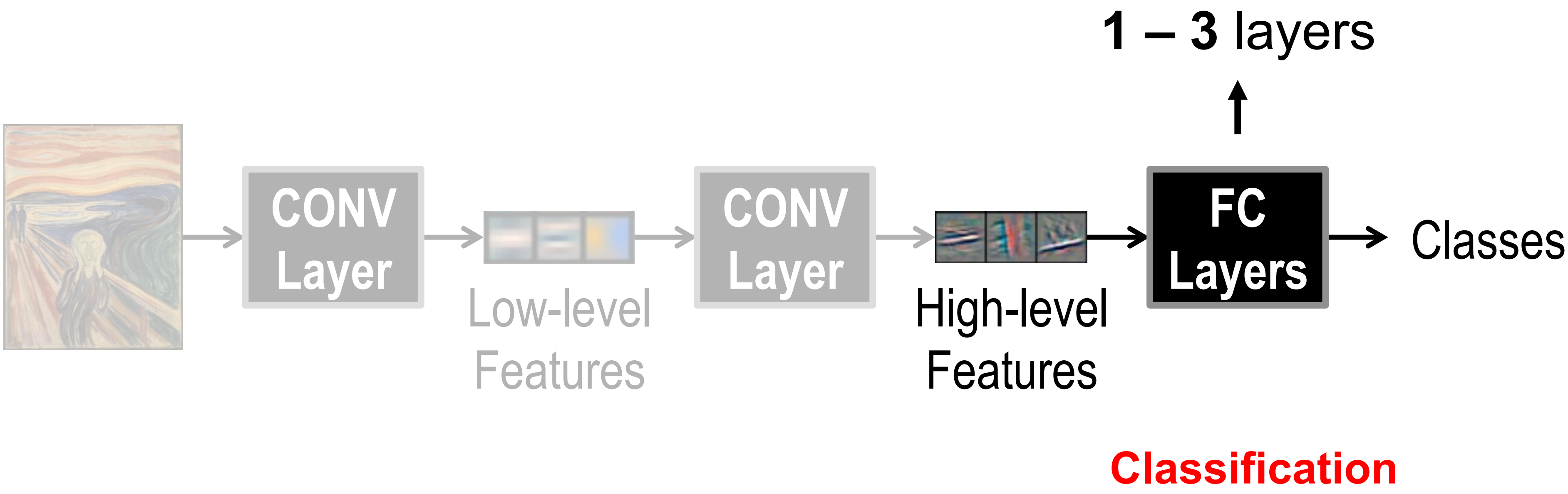


Deep Convolutional Neural Networks

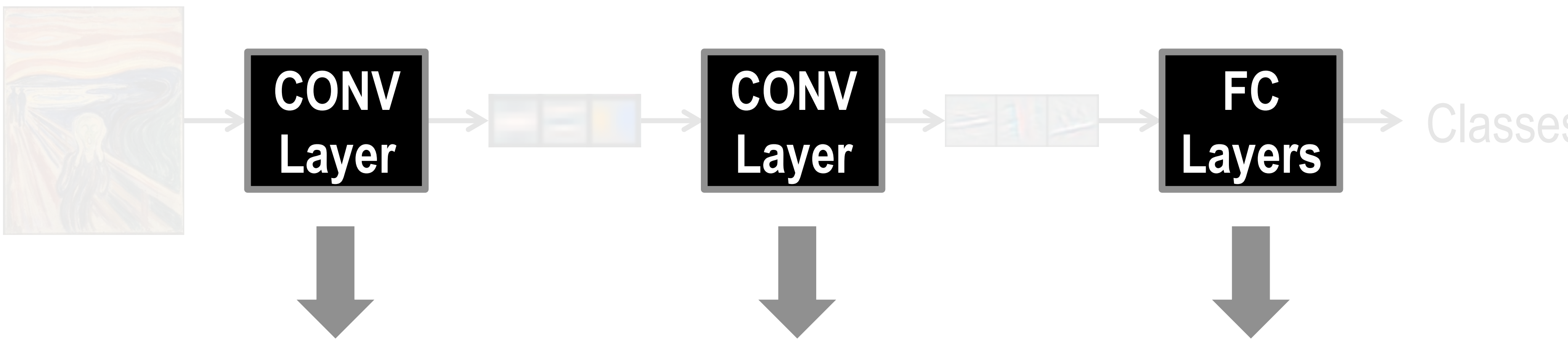


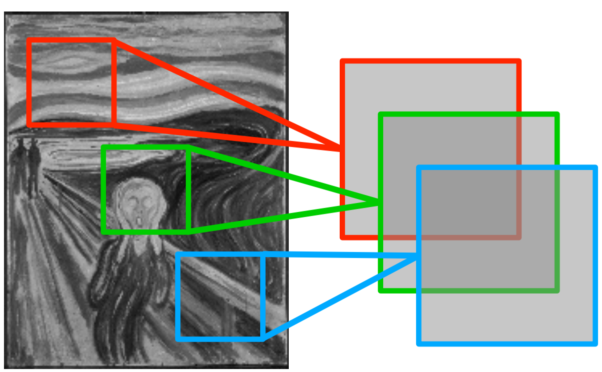
Feature Extraction

Deep Convolutional Neural Networks



Deep Convolutional Neural Networks





Convolutions account for more than 90% of overall computation, dominating **runtime** and **energy consumption**

High-Dimensional CNN Convolution

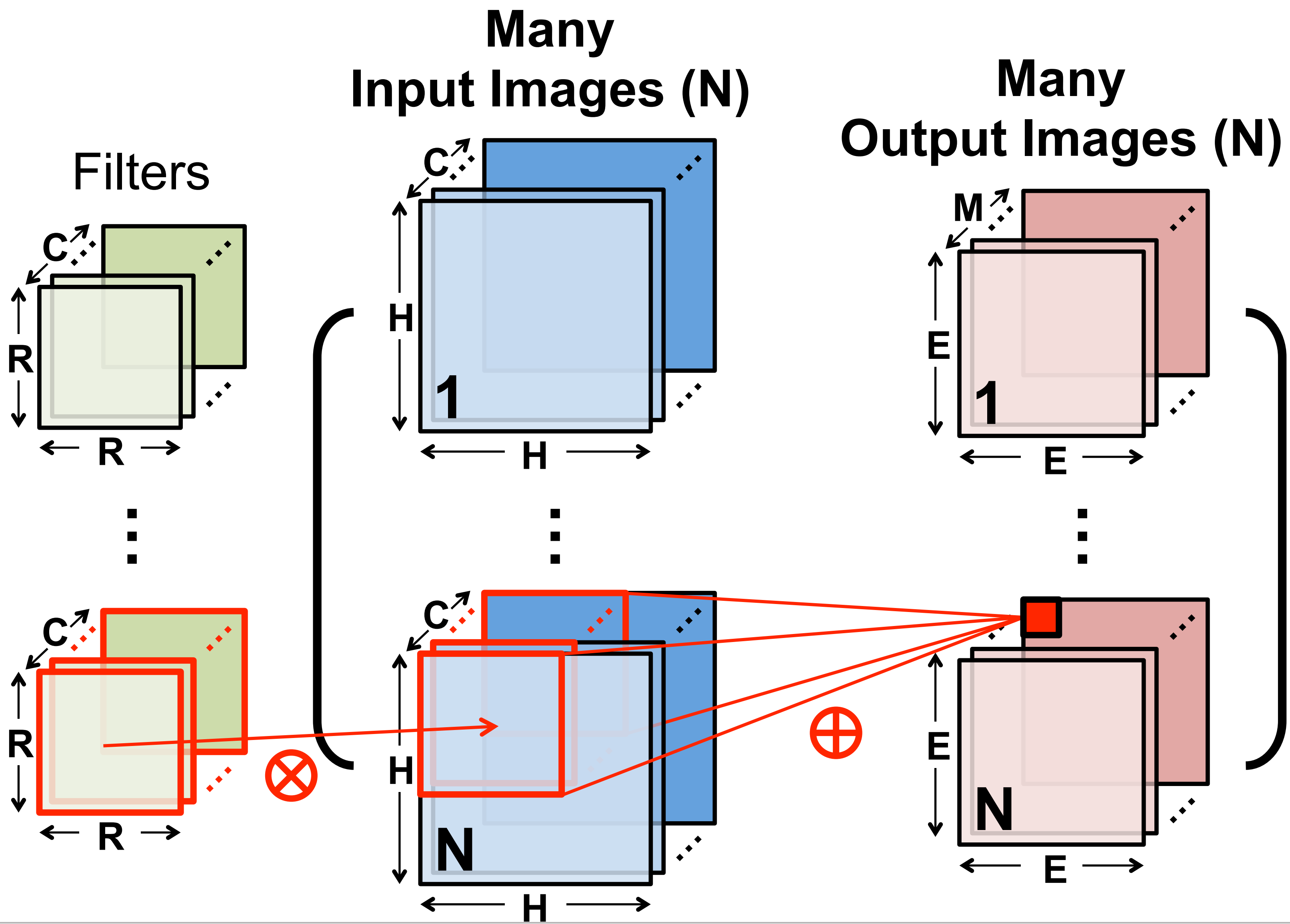
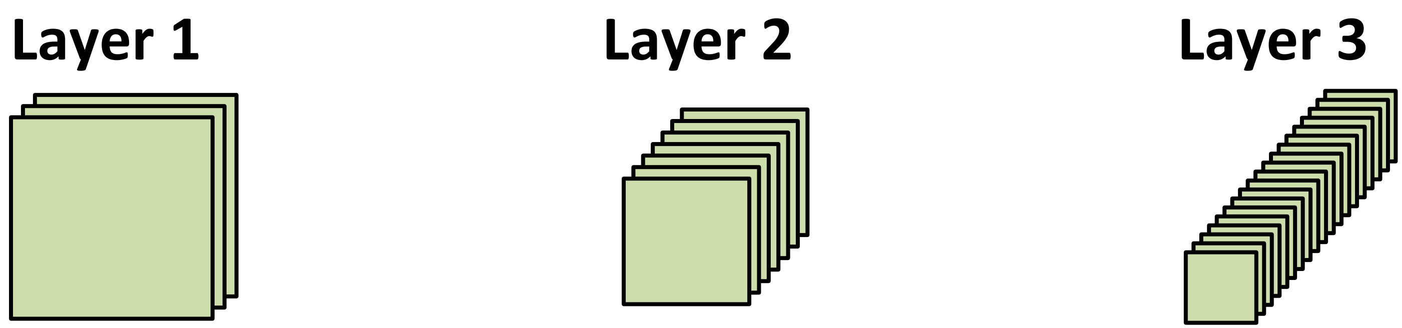


Image batch size: 1 – 256 (N)

Large Sizes with Varying Shapes

AlexNet[1] Convolutional Layer Configurations

Layer	Filter Size (R)	# Filters (M)	# Channels (C)	Stride
1	11x11	96	3	4
2	5x5	256	48	1
3	3x3	384	256	1
4	3x3	384	192	1
5	3x3	256	192	1



34k Params / **105M MACs**

307k Params / **224M MACs**

885k Params / **150M MACs**

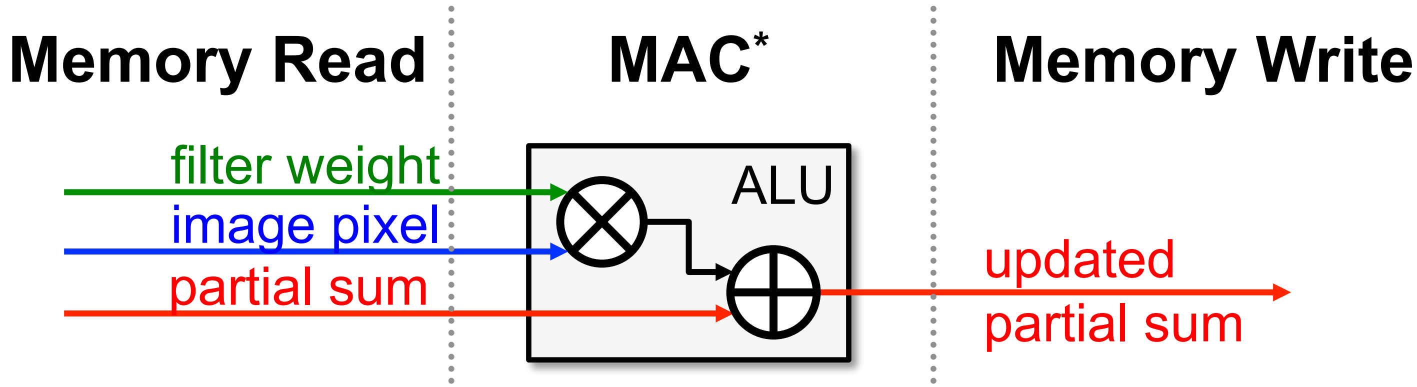
Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible

Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible

- Memory Access is the Bottleneck

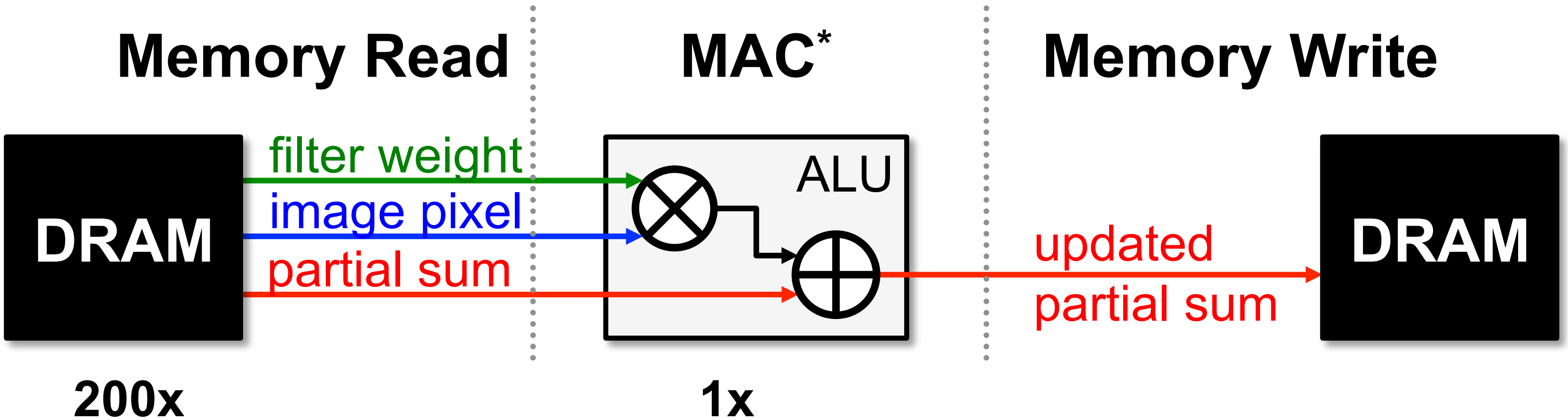


* multiply-and-accumulate

Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible

- Memory Access is the Bottleneck

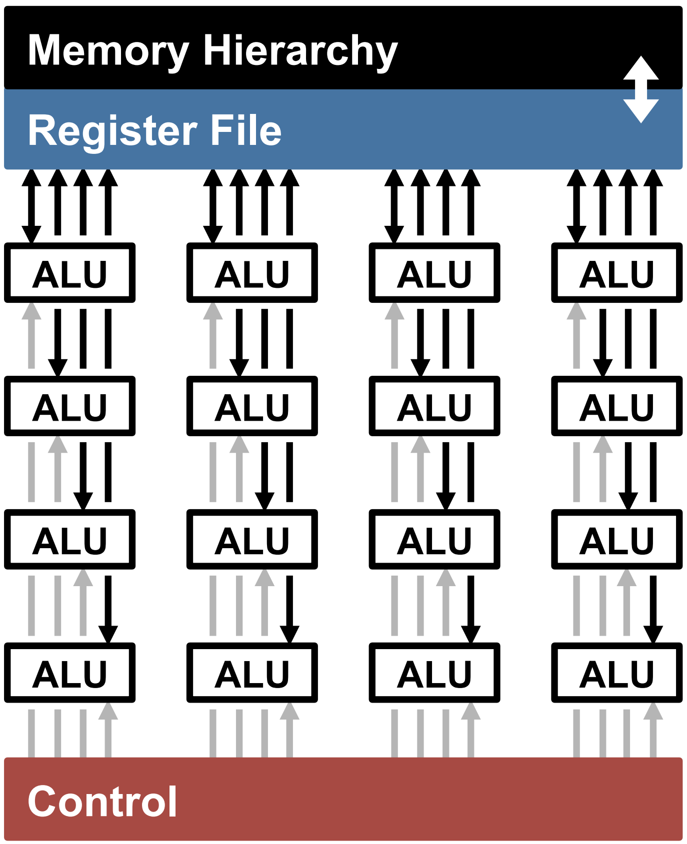


<u>Worst Case</u>: all memory R/W are **DRAM** accesses

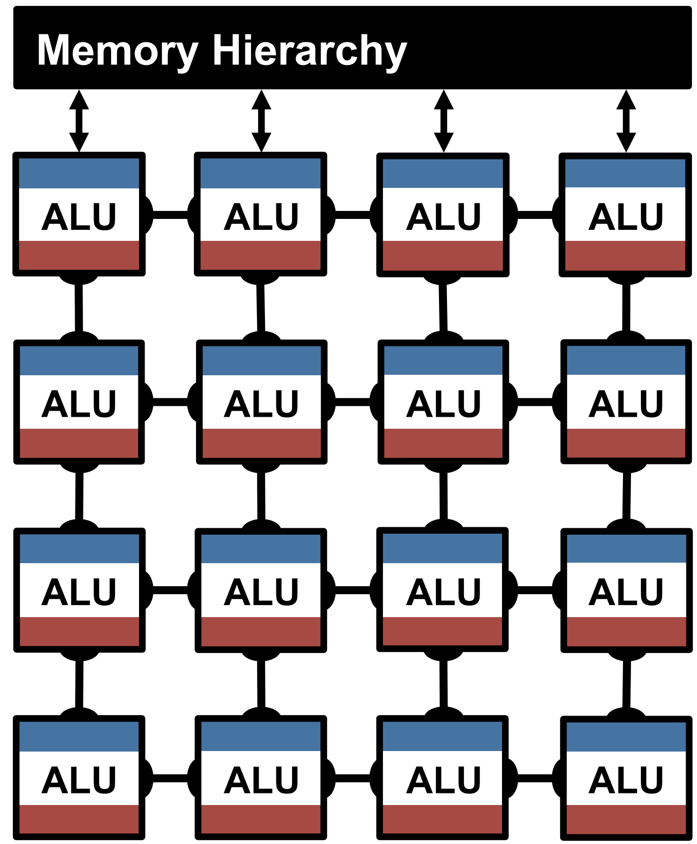
- Example: AlexNet [NIPS 2012] has **724M** MACs
  → **2896M** DRAM accesses required

Highly-Parallel Compute Paradigms

Temporal Architecture (SIMD/SIMT)



Spatial Architecture (Dataflow Processing)



Advantages of Spatial Architecture

Temporal Architecture (SIMD/SIMT)

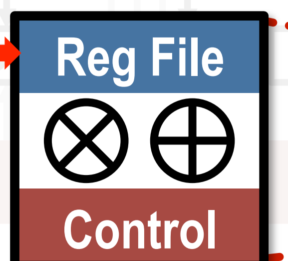
Spatial Architecture (Dataflow Processing)

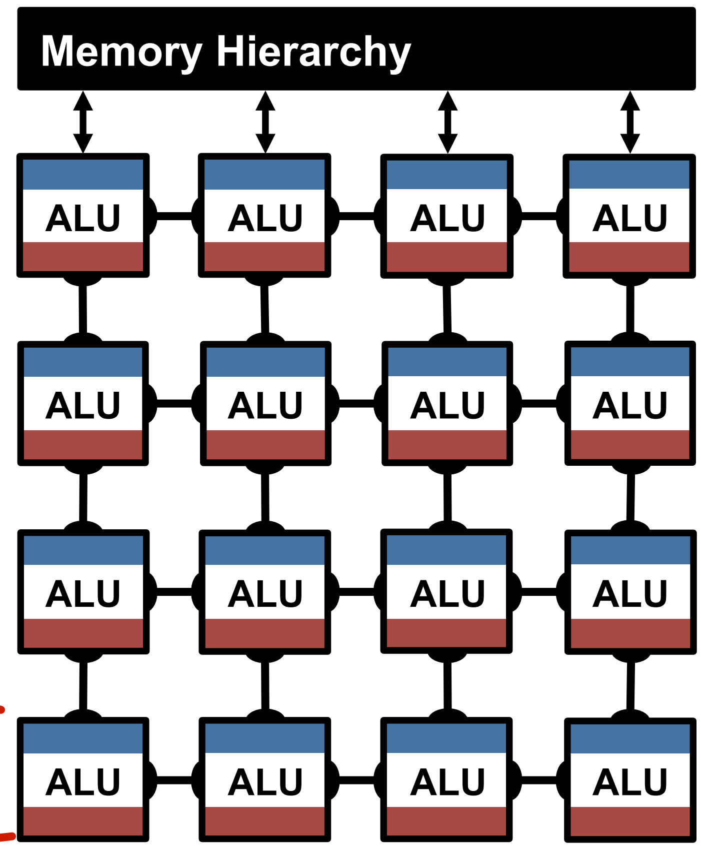
Efficient Data Reuse
Distributed local storage (RF)

Inter-PE Communication
Sharing among regions of PEs

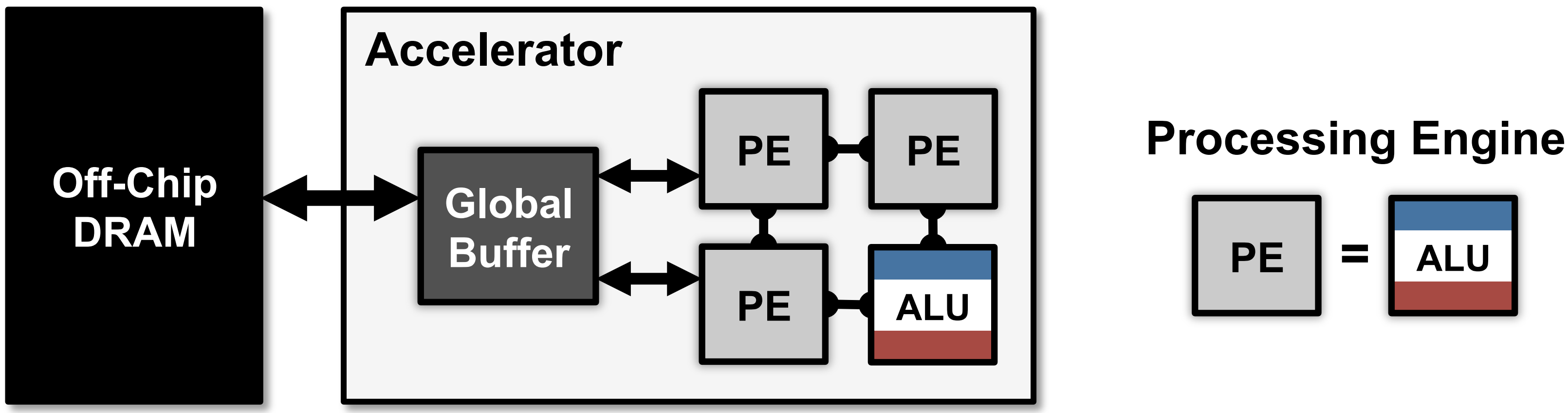
Processing Element (PE)

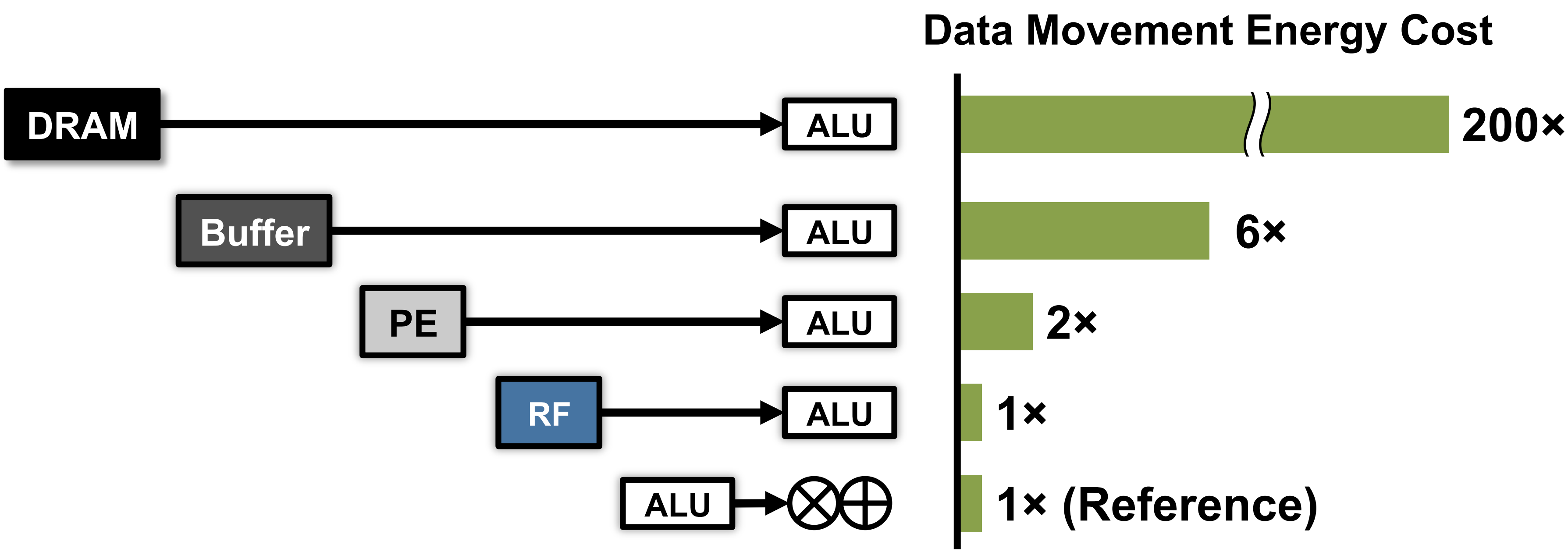
0.5 – 1.0 kB





Data Movement is Expensive



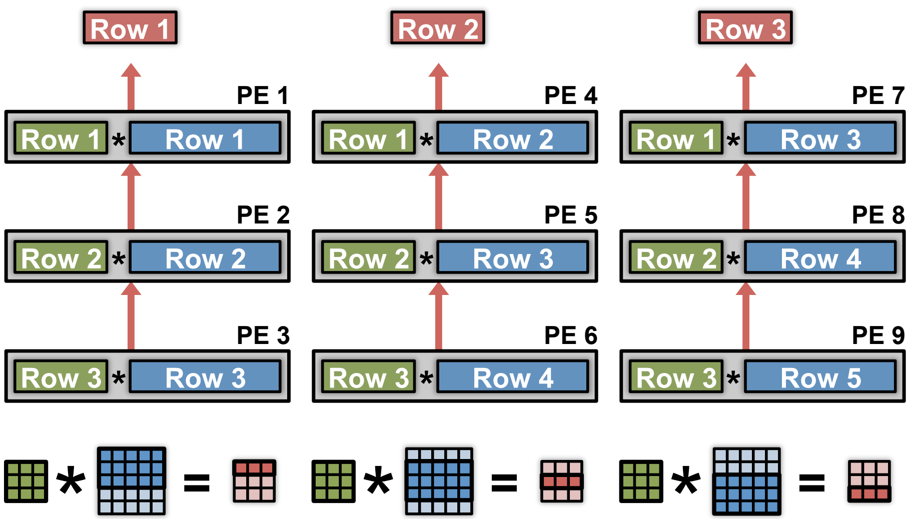


Maximize data reuse at lower levels of hierarchy

Optimization to Reduce Data Movement

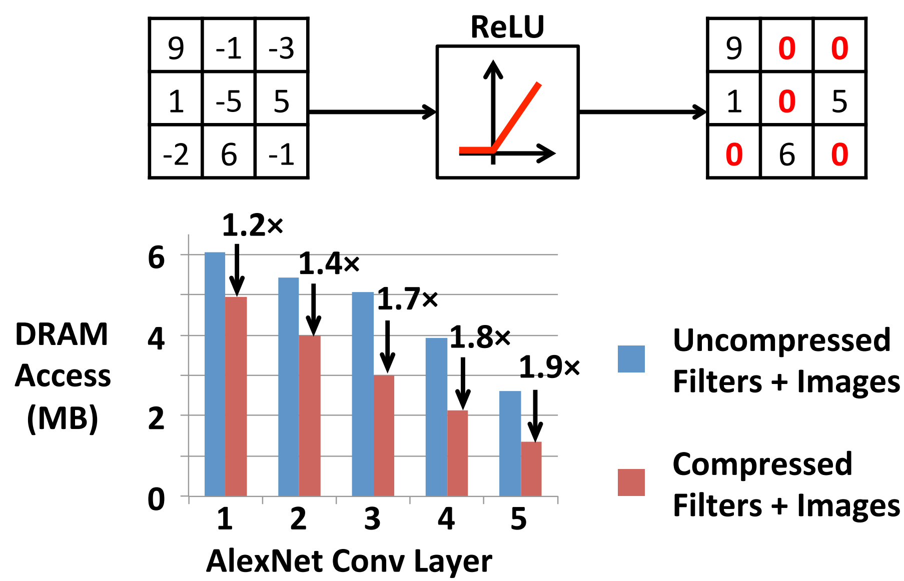
- Energy-efficient **dataflow to reduce data movement**
- **Exploit data statistics** for high energy efficiency

Row Stationary Dataflow

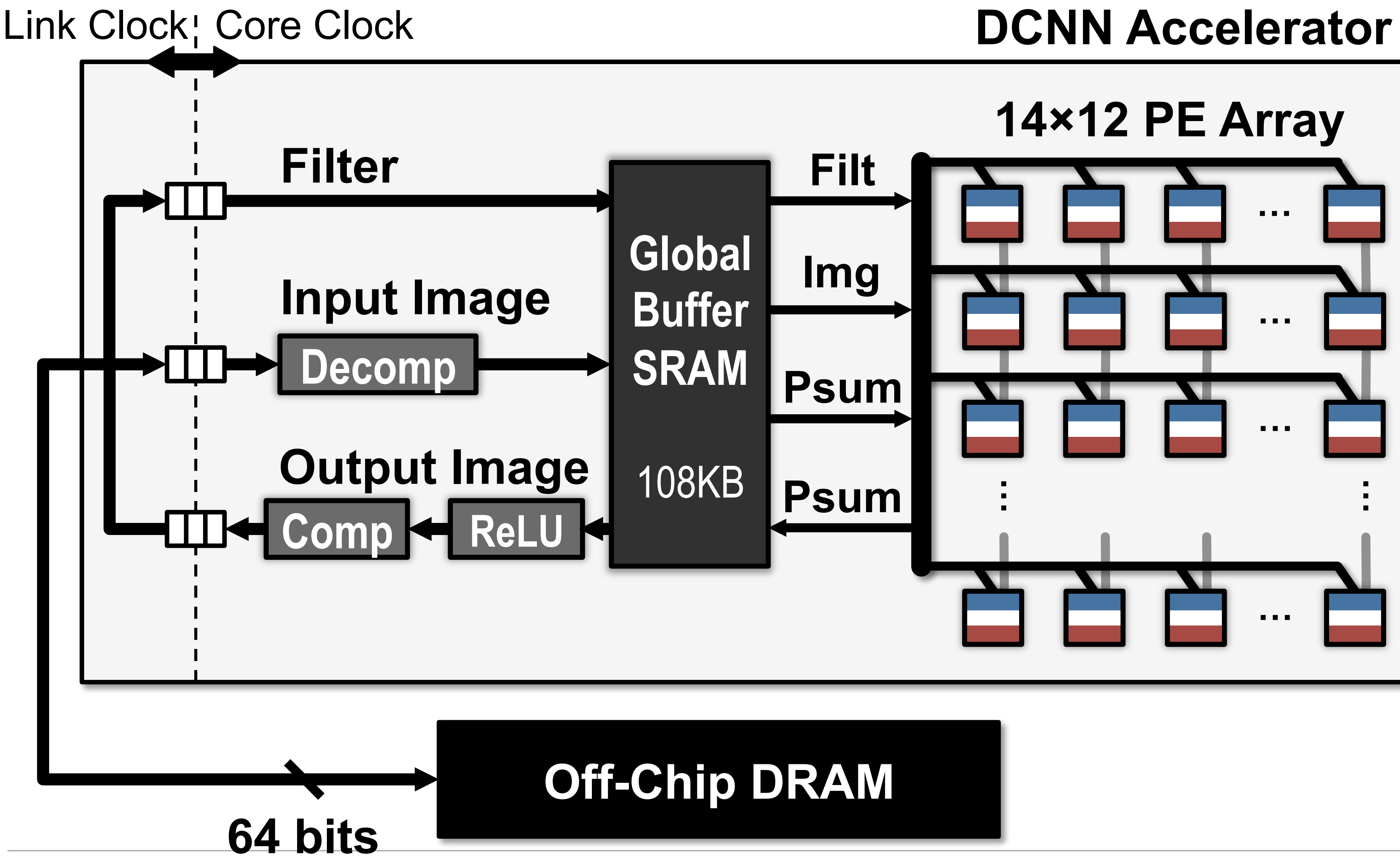


Sparsity in Activations

Apply Non-Linearity (**ReLU**) on Filtered Image Data

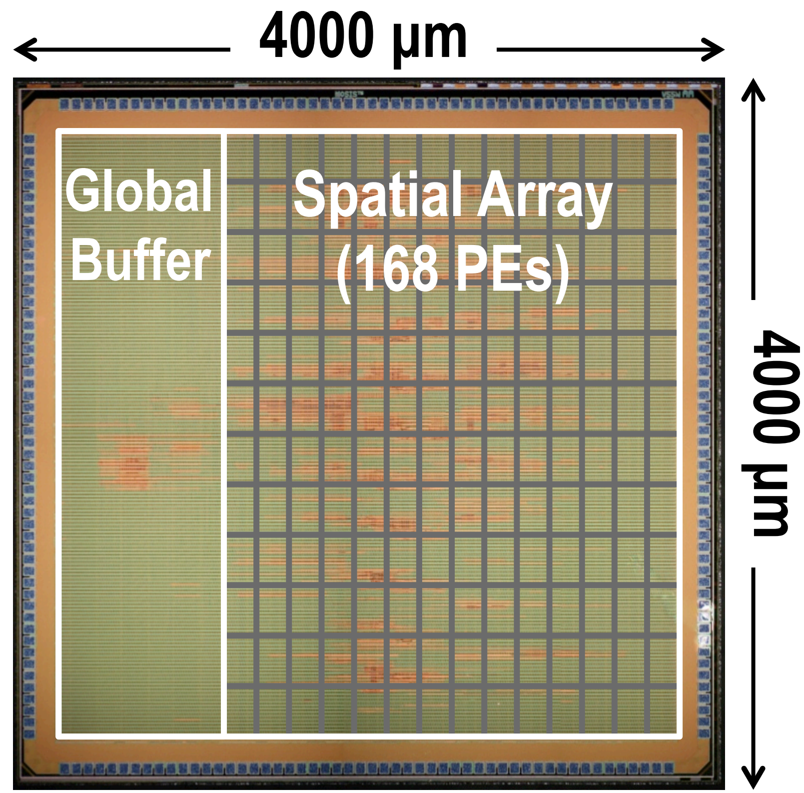


Eyeriss Deep CNN Accelerator



Eyeriss Chip Spec & Measurement Results

Technology	TSMC 65nm LP 1P9M
On-Chip Buffer	108 KB
# of PEs	168
Scratch Pad / PE	0.5 KB
Core Frequency	100 – 250 MHz
Peak Performance	33.6 – 84.0 GOPS
Word Bit-width	16-bit Fixed-Point
Natively Supported CNN Shapes	Filter Width: 1 – 32 Filter Height: 1 – 12 Num. Filters: 1 – 1024 Num. Channels: 1 – 1024 Horz. Stride: 1–12 Vert. Stride: 1, 2, 4

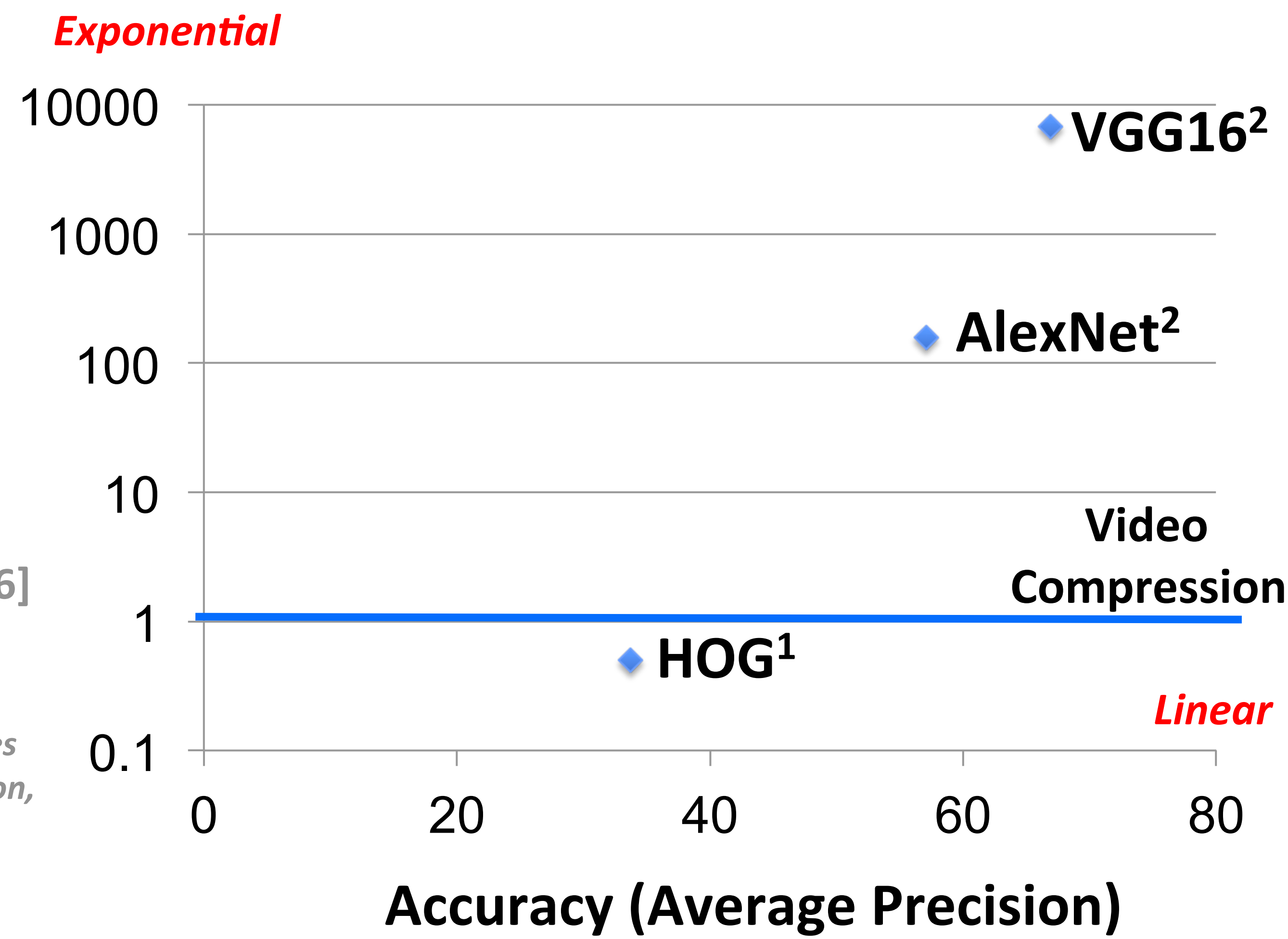


Over **10x more energy efficient** than a mobile GPU (Nvidia TK1)

[Chen et al., ISSCC 2016]

Features: Energy vs. Accuracy

Energy/
Pixel (nJ)



*Measured in 65nm**

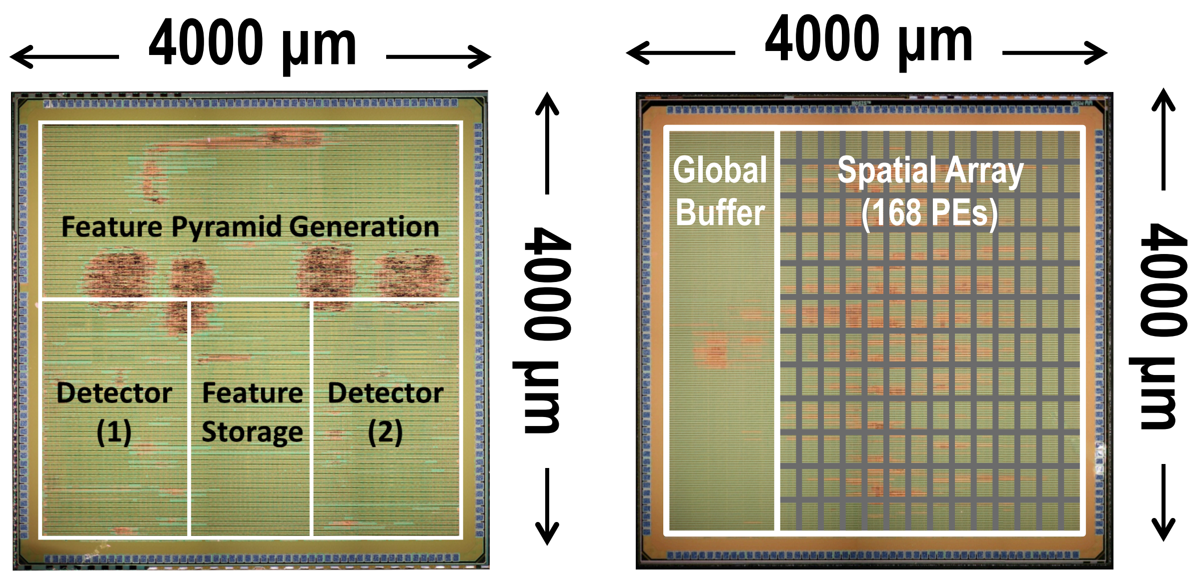
1. [Suleiman, VLSI 2016]
2. [Chen, ISSCC 2016]

** Only feature extraction. Does not include data, augmentation, ensemble and classification energy, etc.*

Measured in on VOC 2007 Dataset

1. DPM v5 [Girshick, 2012]
2. Fast R-CNN [Girshick, CVPR 2015]

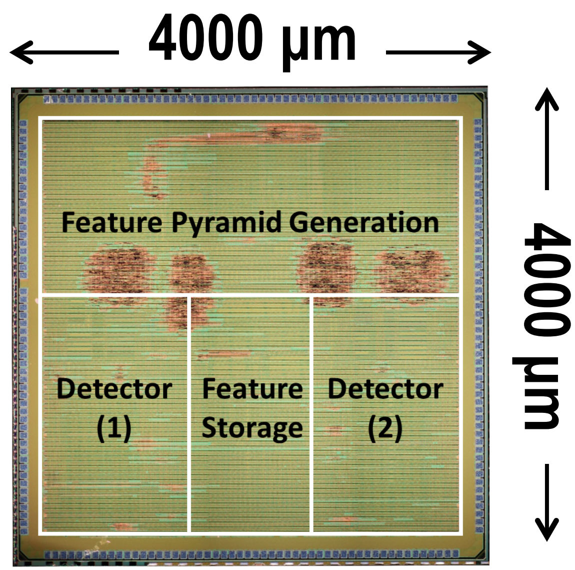
HOG vs. CNN: Hardware Cost

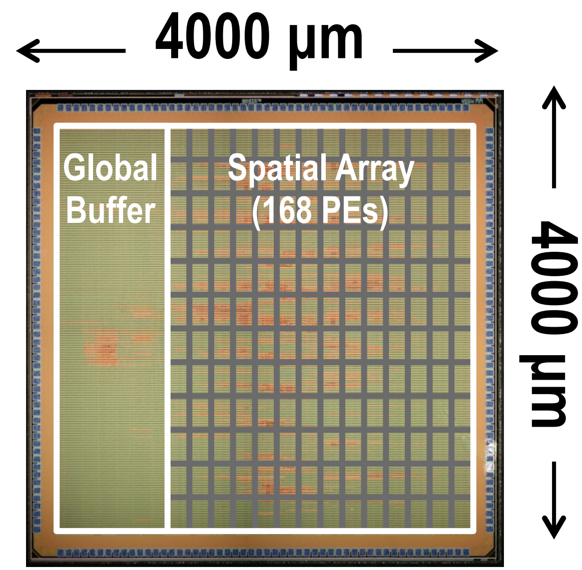


	HOG [VLSI 2016]	CNN [ISSCC 2016]
Technology	TSMC LP 65nm	TSMC LP 65m
Gate Count (kgates)	893	1176
Memory (kB)	159	181.5

Similar Hardware Cost (comparable with Video Compression)

HOG vs. CNN: Throughput

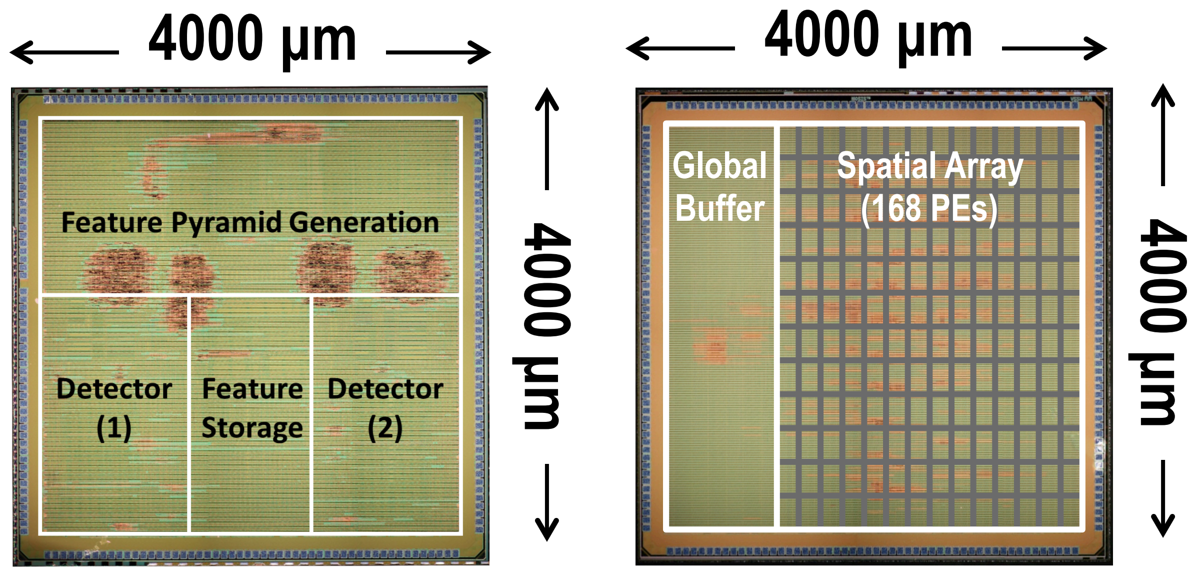




	HOG	CNN (AlexNet)	CNN (VGG-16)
Throughput (Mpixels/s)	62.5	1.8	0.04
GOP/Mpixel	0.7	25.8	610.3
Throughput (GOPS)	46.0	46.2	21.4

Throughput gap explained by GOP/Mpixel gap

HOG vs. CNN: Energy and DRAM Access



	HOG	CNN (AlexNet)	CNN (VGG-16)
Energy (nJ/pixel)	0.5	155.5	6742.9
GOP/Mpixel	0.7	25.8	610.3
Energy (GOPS/W)	1570	166.2	90.7
DRAM (B/pixel)	1.0	74.7	2128.6

Energy gap larger than GOPS/W gap

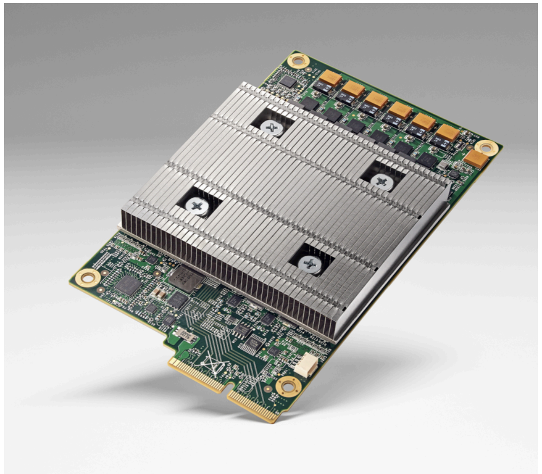
Energy Gap between CNN and HOG

- **CNNs require more operations per pixel**
  - AlexNet vs. HOG = **37x**
  - VGG-16 vs. HOG = **872x**
- **CNN requires a programmable architecture**
  - **Example:** AlexNet CONV layers have **2.3M weights** (assume 8-bits per weight); Area budget of HOG chip is ~**1000 kgates, 150kB**
  - **Design A: Hard-wired weights**
    - Only have 10k multipliers with fixed weights **(>100x increase in area)**
  - **Design B: Store all weights on-chip**
    - Only store 150k weights on chip **(>10x increase in storage)**
  - **Support different shapes per layer and different weights**

Closing the Energy Gap

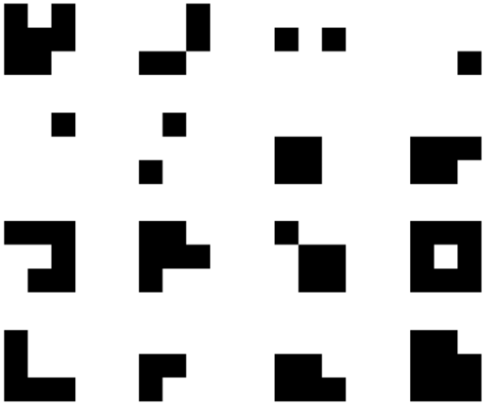
Methods to Reduce Energy of CNNs

- **Reduce Precision**
  - [Google TPU, ISCA 2017], [XNOR-Net, ECCV 2016], [BinaryNets, arXiv 2016]
- **Sparsity by Pruning**
- **Data Compression**
  - [Chen, ISSCC 2016], [Han, ISCA 2016], [Moons, VLSI 2016]
- **Energy Optimized Dataflow**
  - [Chen, ISCA 2016]



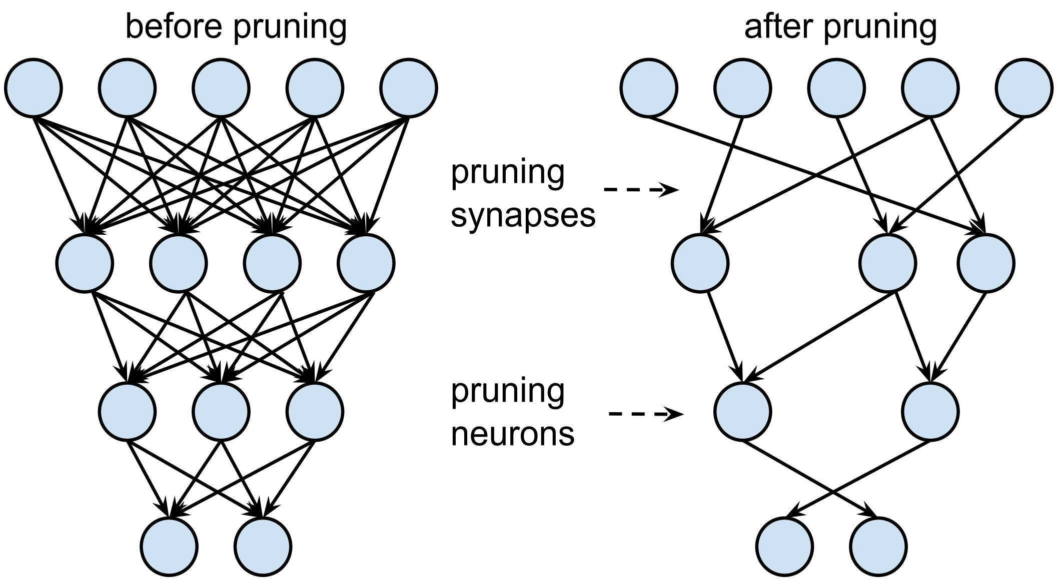
Google's TPU (8-bits)

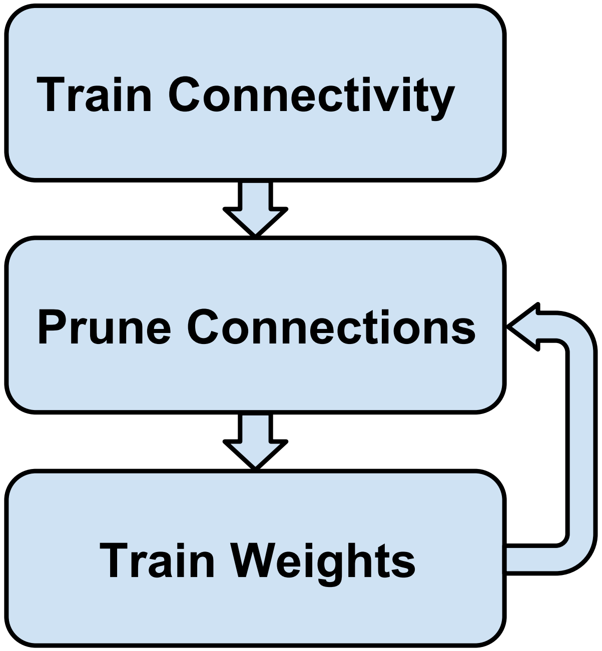
Binary Filters



Pruning – Make Weights Sparse

Prune based on ***magnitude*** of weights





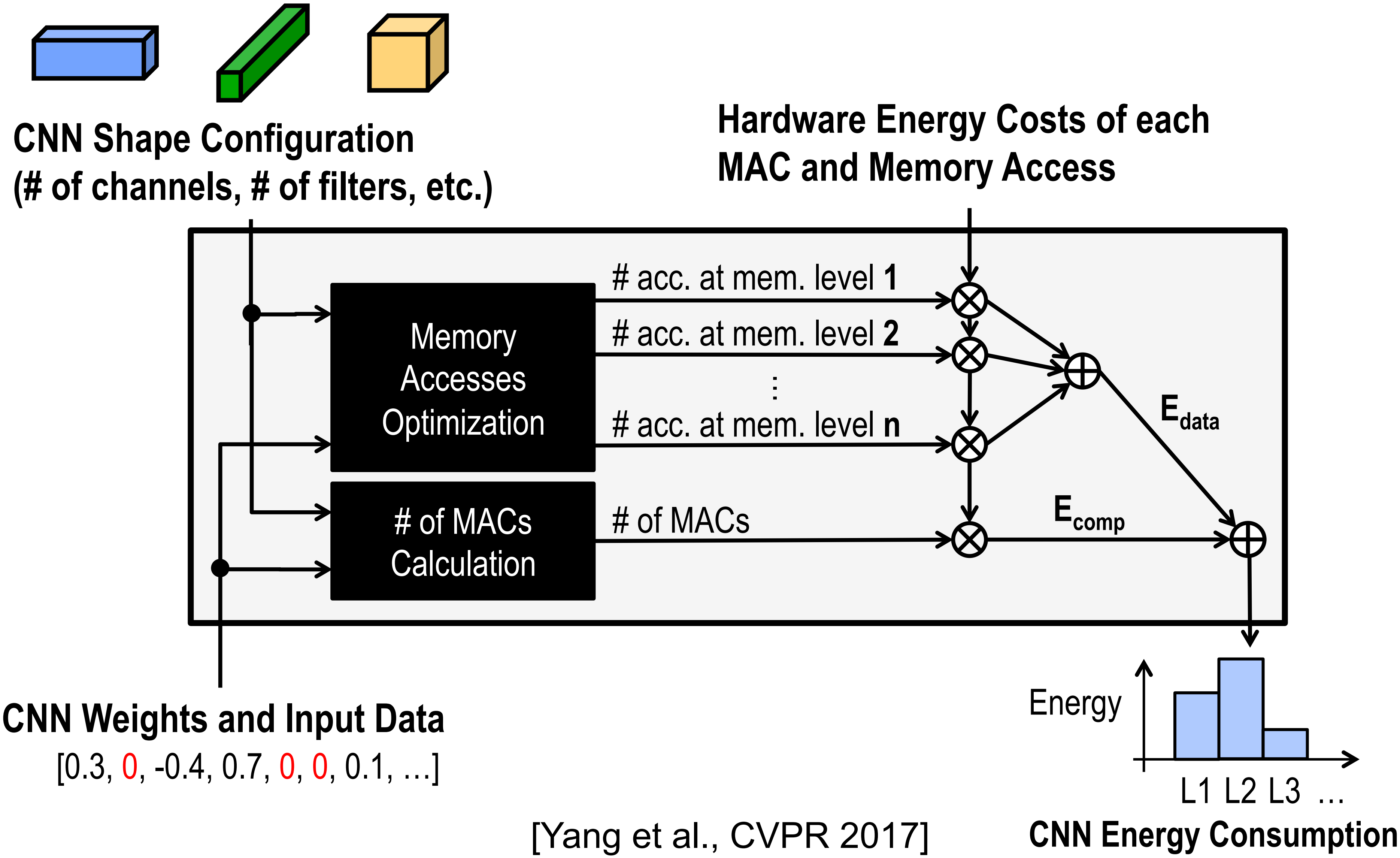
Example: *AlexNet*
Weight Reduction: *CONV layers 2.7x, FC layers 9.9x*
(Most reduction on fully connected layers)
Overall: *9x weight reduction, 3x MAC reduction*

[Han et al., NIPS 2015]

Key Metrics for Embedded DNN

- Accuracy → Measured on Dataset
- Speed → Number of MACs
- Storage Footprint → Number of Weights
- Energy → ?

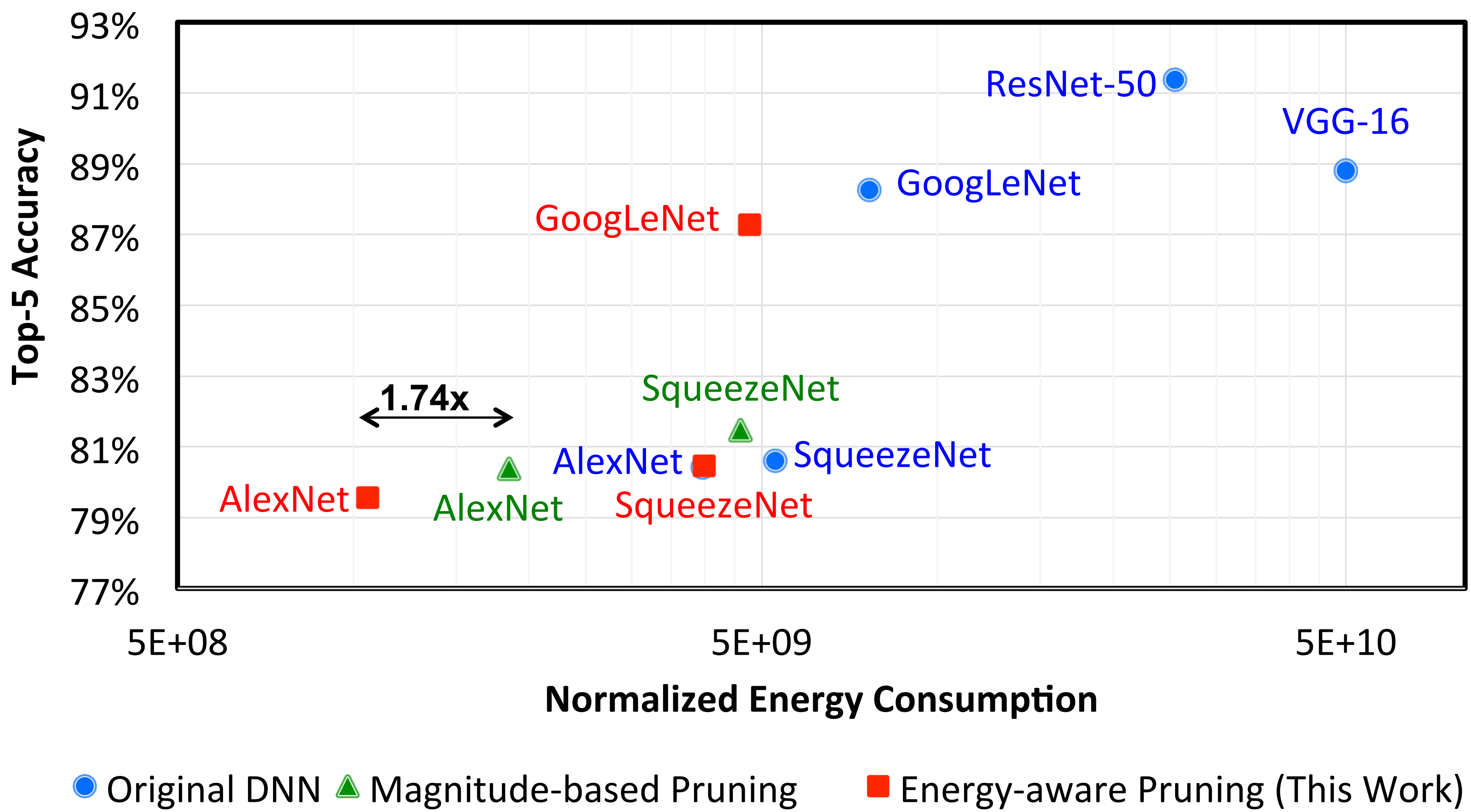
Energy-Evaluation Methodology



[Yang et al., CVPR 2017]

Energy estimation tool available at http://eyeriss.mit.edu

Energy-Aware Pruning



Remove weights from layers **in order of highest to lowest energy**
3.7x reduction in AlexNet / 1.6x reduction in GoogLeNet

[Yang et al., CVPR 2017]

Summary

- CNN gives higher accuracy than HOG features (2x) at the cost of increase energy (311x to 13486x)
- Energy gap due to (1) CNN requires more operations per pixel and (2) CNN requires a programmable architecture
- Joint algorithm and hardware design can deliver additional energy savings to help close this gap

More info about **Eyeriss** and **Tutorial on DNN Architectures** at http://eyeriss.mit.edu

V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "*Efficient Processing of Deep Neural Networks: A Tutorial and Survey*", arXiv, 2017

Follow @eems_mit