Hardware for Machine Learning: Design Considerations

Vivienne Sze

In collaboration with Yu-Hsin Chen, Joel Emer, Tien-Ju Yang

Massachusetts Institute of Technology

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

Outline

- Selecting a Machine Learning Approach
- Limitations of Existing Efficient DNN Approaches
- Benchmarking Metrics for DNN Hardware

Selecting a Machine Learning Approach

A. Suleiman*, Y.-H. Chen*, J. Emer, V. Sze,
"Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," ISCAS 2017

Hand-Crafted vs. Learned Features

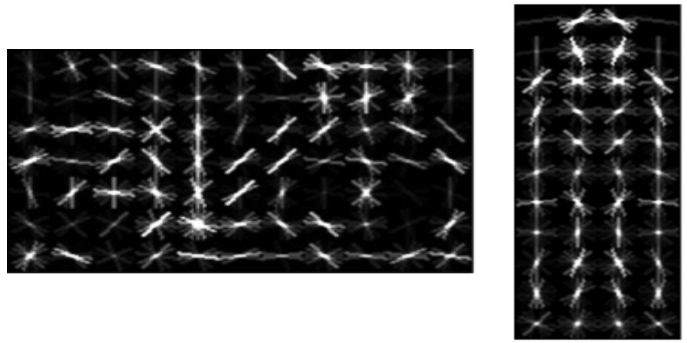
Image

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

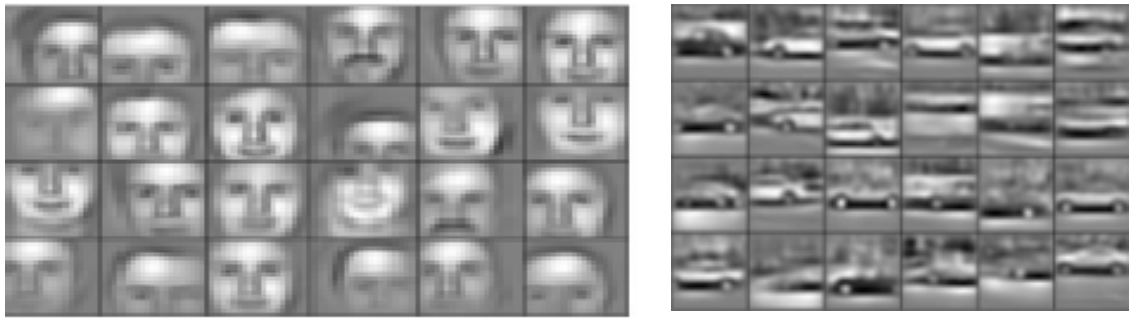
Trained weights (**w**)

(select class based on max or threshold)

Handcrafted Features (e.g. HOG)

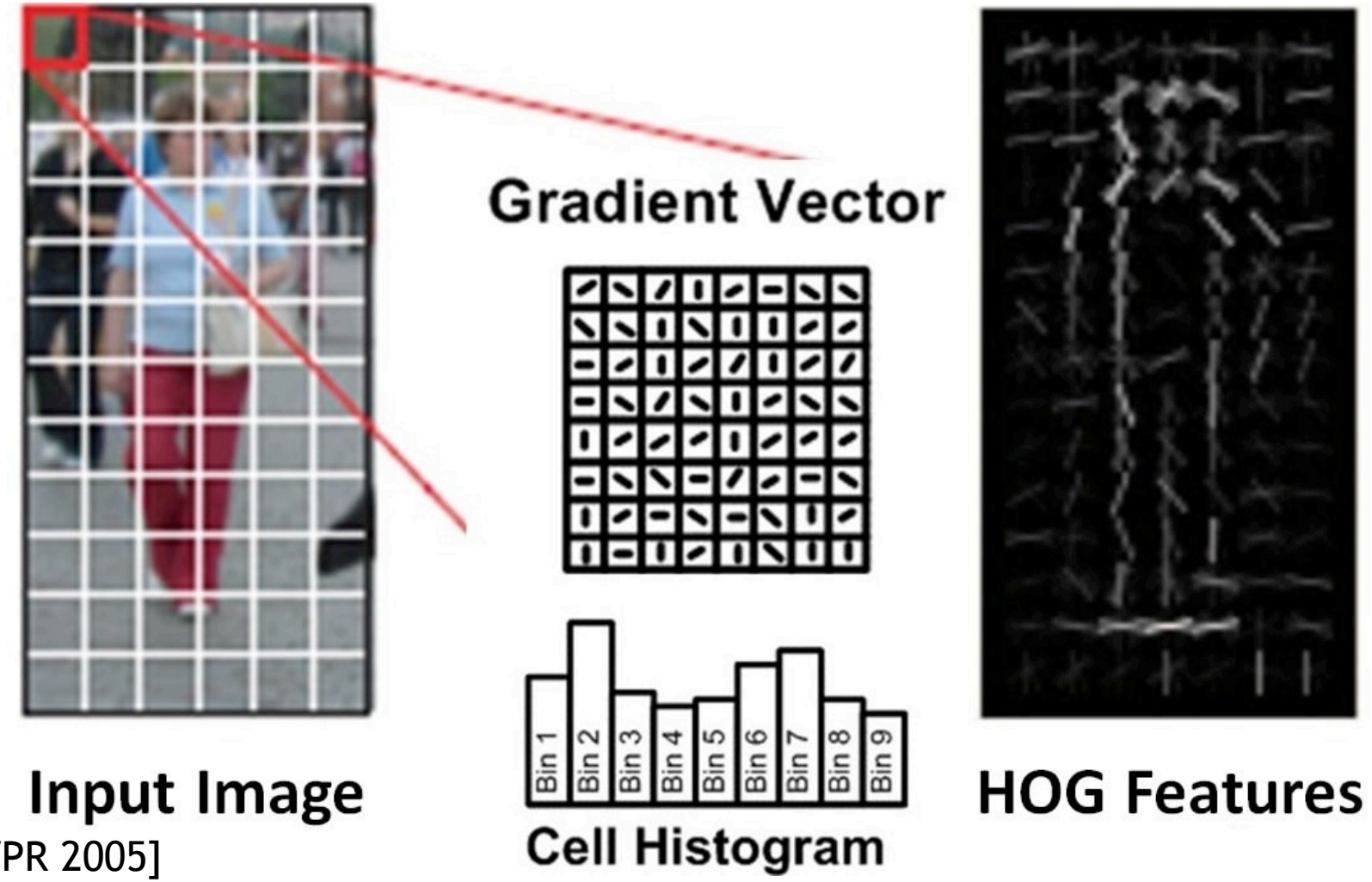


Learned Features (e.g. CNN)



Hand-Crafted Features (HOG)

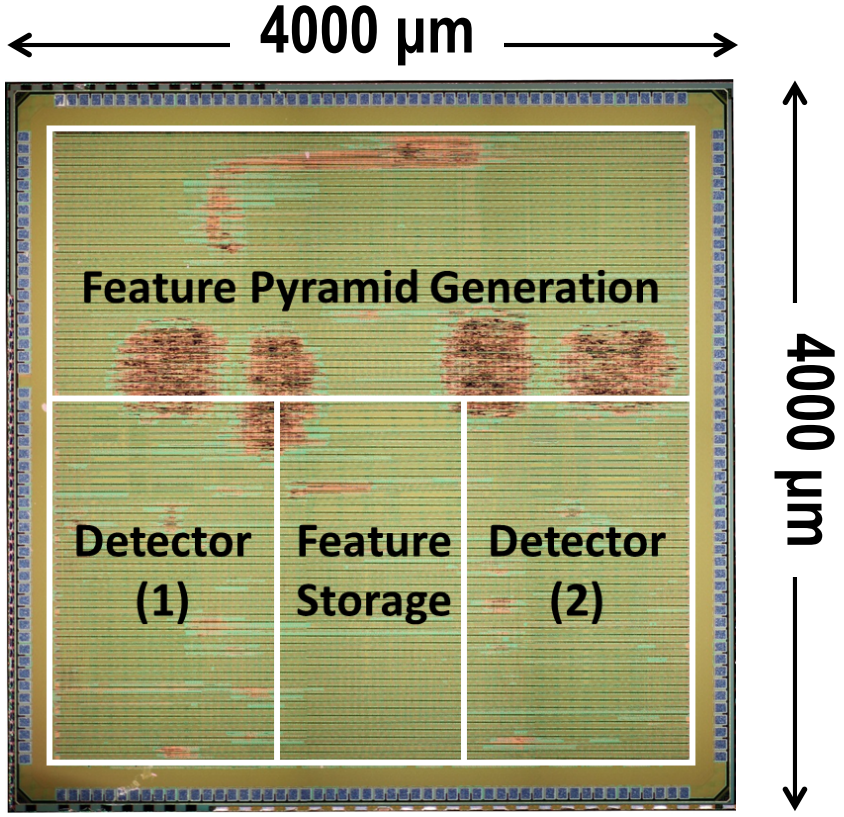
HOG = Histogram of Oriented Gradients



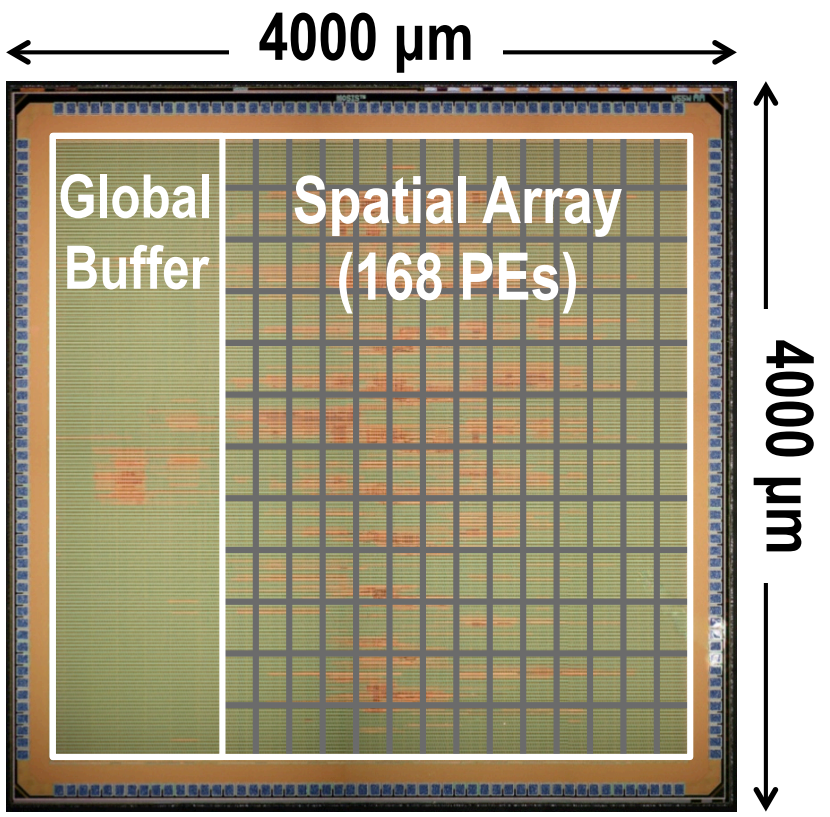
[Dalal and Triggs, CVPR 2005]

Compare HOG and CNN

Compare using measured results from test chips (65 nm)

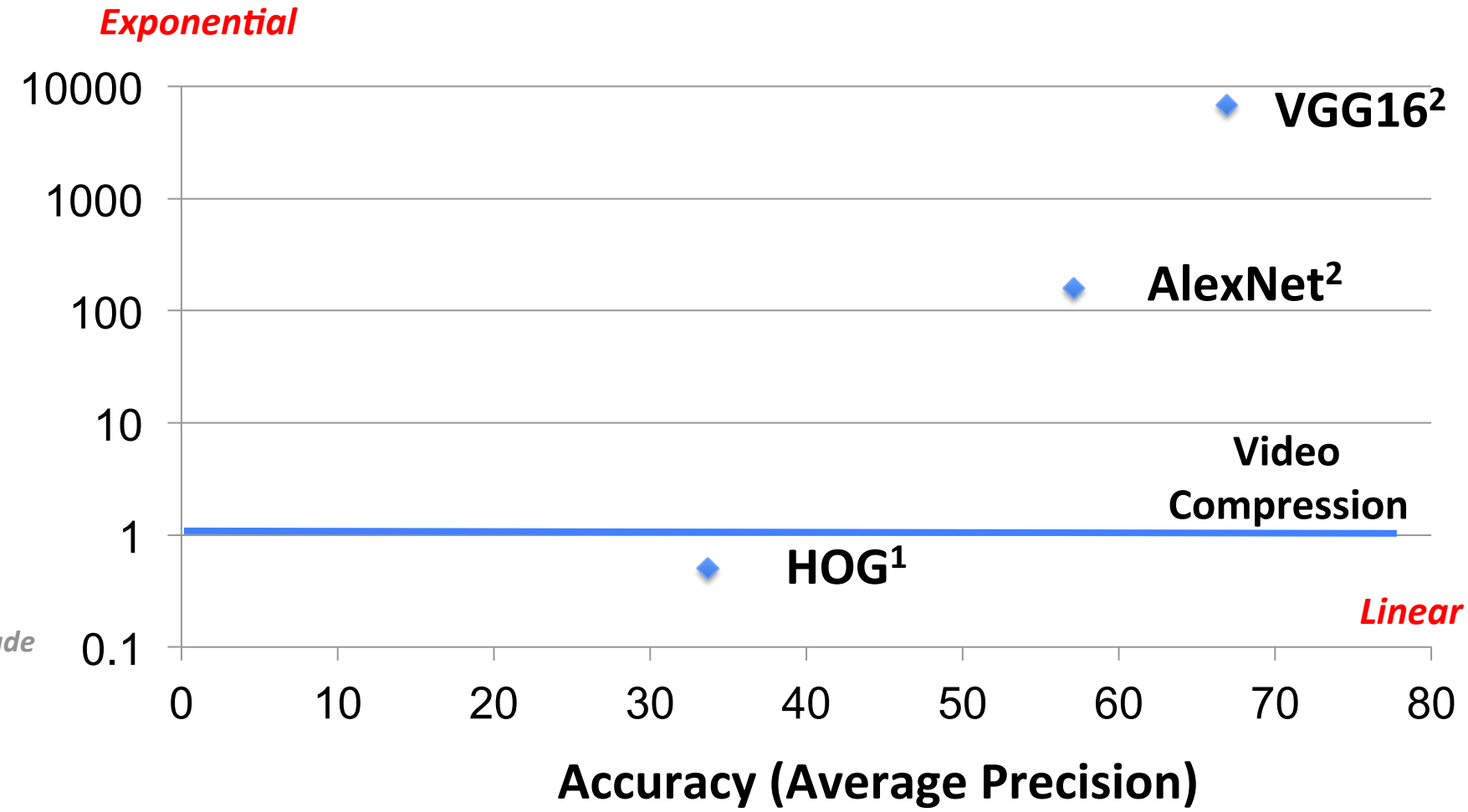


Object Detection using **HOG features** and Deformable Parts Models [Suleiman et al., VLSI 2016]



Eyeriss: **Convolutional Neural Networks** [Chen et al., ISSCC 2016, ISCA 2016]

Features: Energy vs. Accuracy Tradeoff



*Measured in 65nm**

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

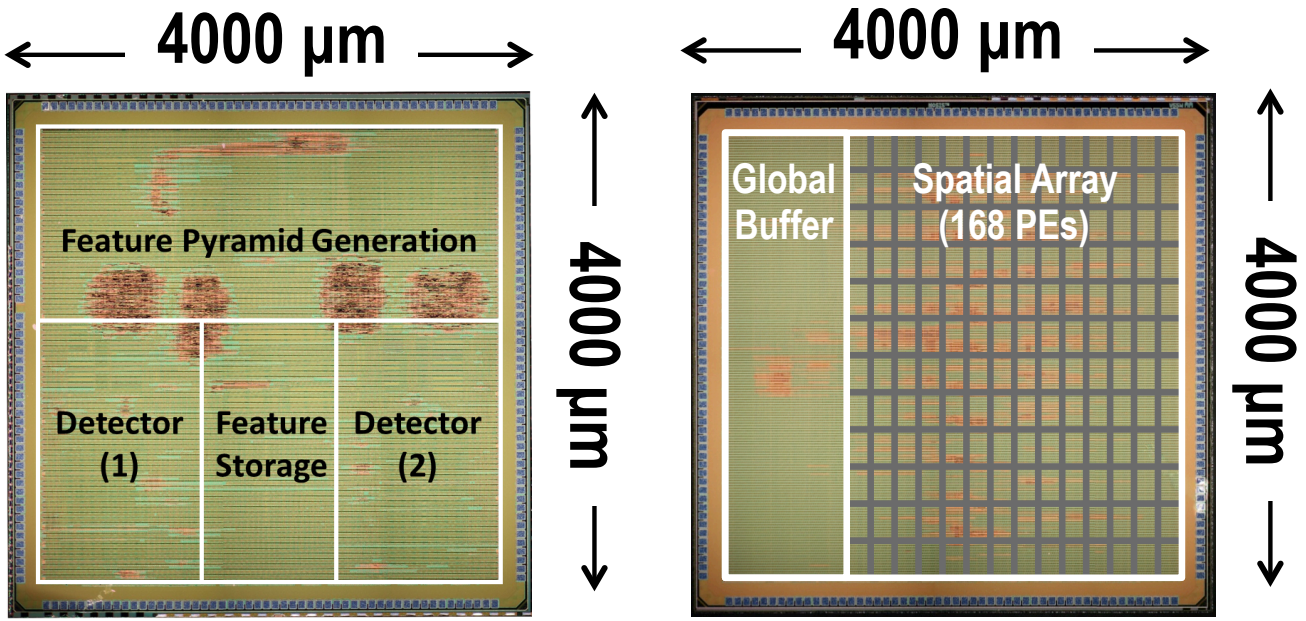
** Only feature extraction. Does not include data, augmentation, ensemble and classification energy, etc.*

Measured in on VOC 2007 Dataset

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

[Suleiman et al., ISCAS 2017]

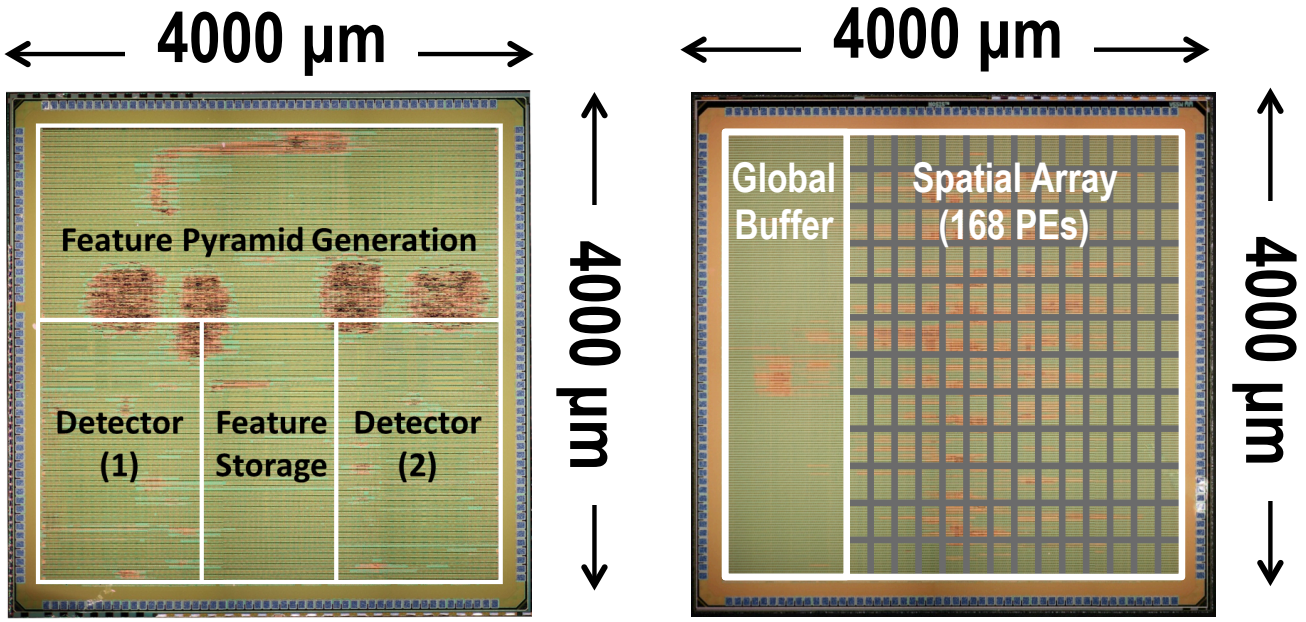
HOG vs. CNN: Hardware Cost



	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)

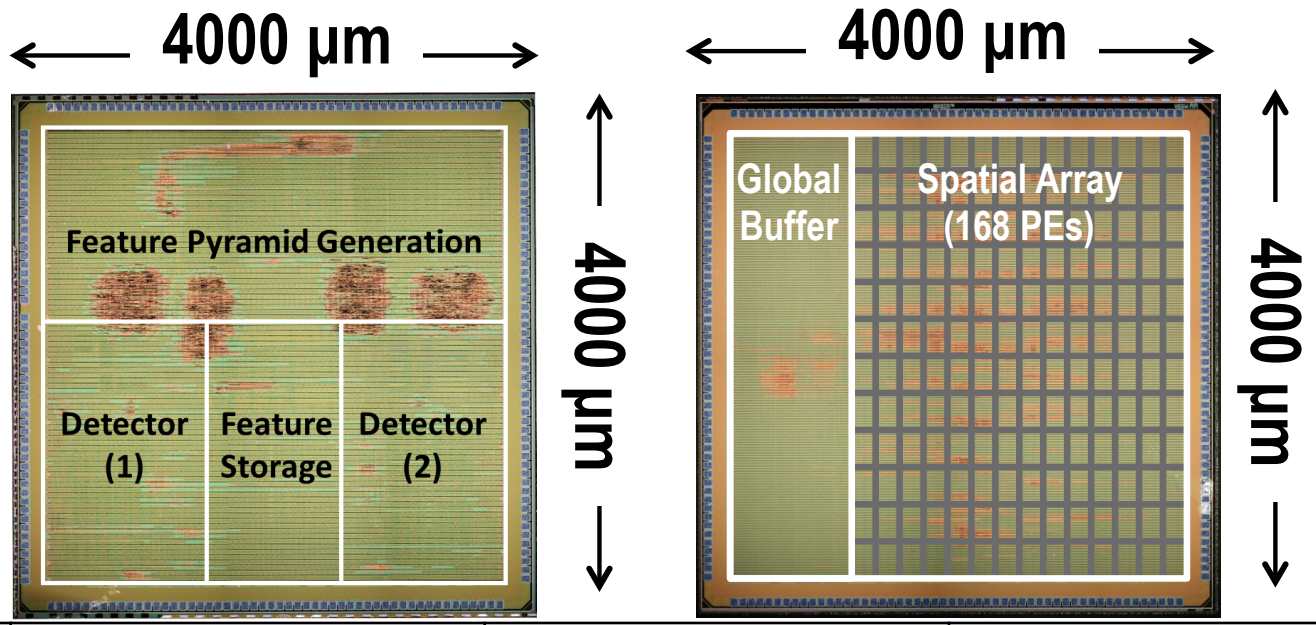
HOG vs. CNN: Throughput



	HOG	CNN (AlexNet)	CNN (VGG16)
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 (VGG16)
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/Mpixel 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

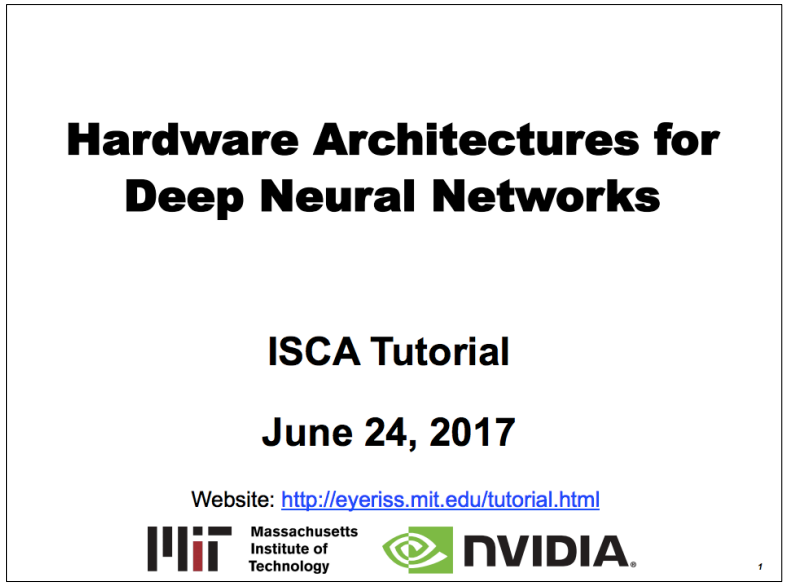
Limitations of Existing Efficient DNN Approaches

Y.-H. Chen*, T.-J. Yang*, J. Emer, V. Sze,

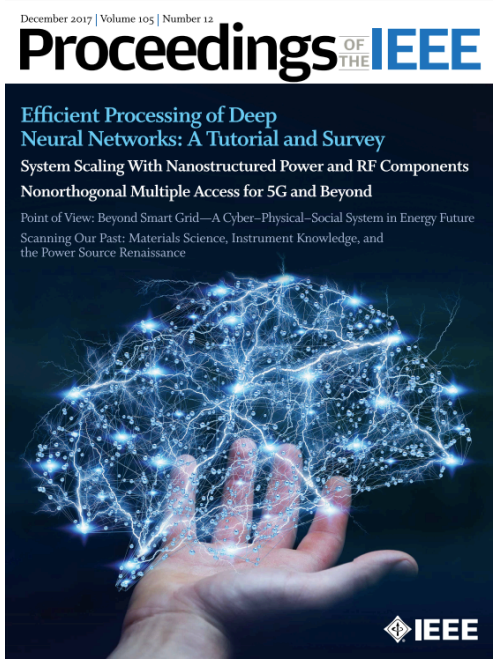
"Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," *SysML* 2018.

Energy-Efficient Processing of DNNs

A significant amount of algorithm and hardware research on energy-efficient processing of DNNs



eyeriss.mit.edu/tutorial.html



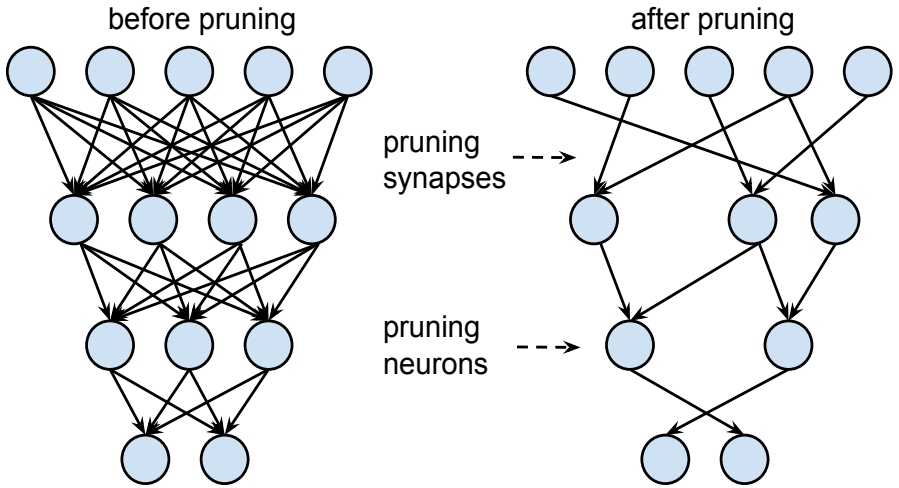
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

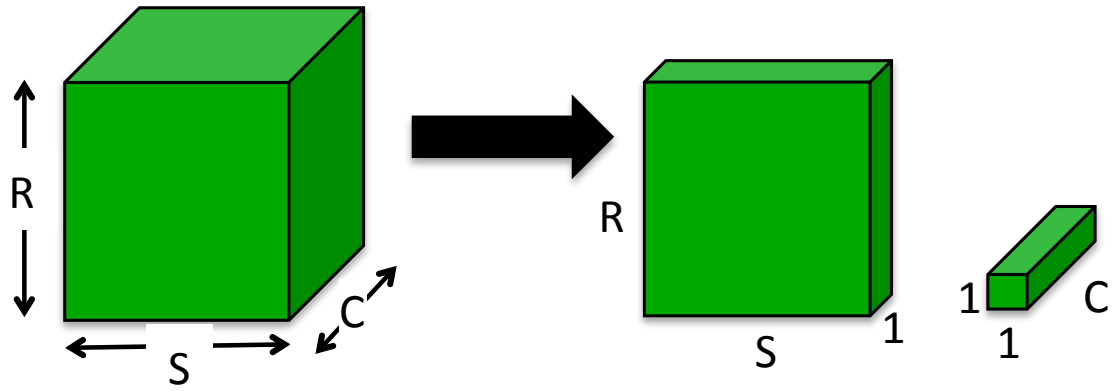
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

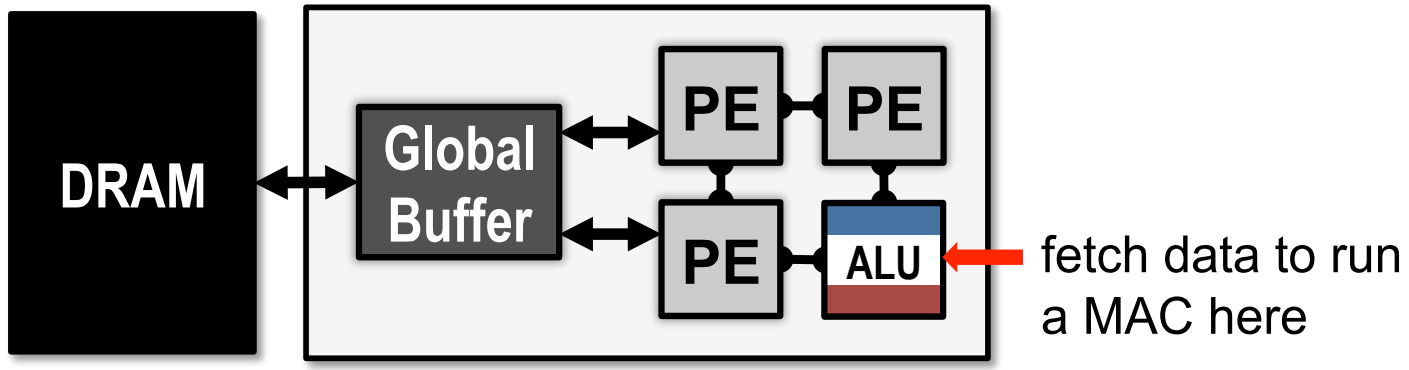


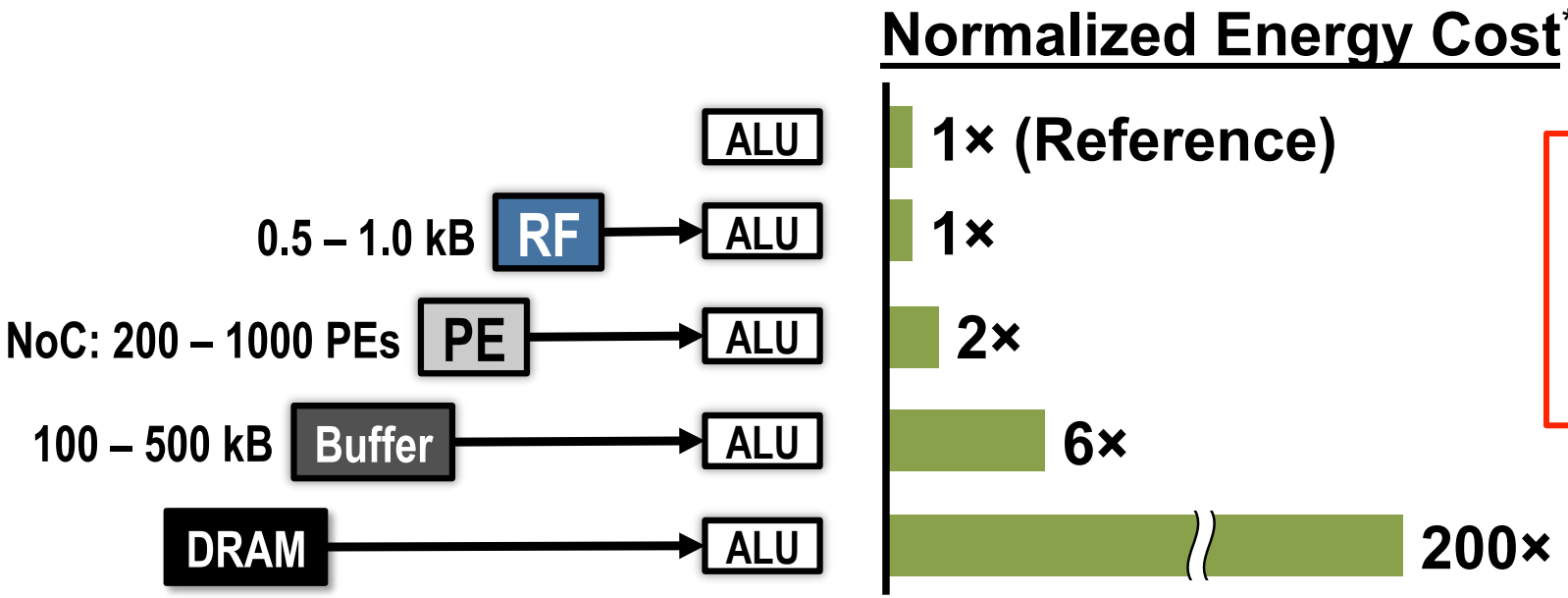
Examples: SqueezeNet, MobileNet

… also reduced precision

- Focus on reducing **number of MACs and weights**
- **Does it translate to energy savings?**

Data Movement is Expensive

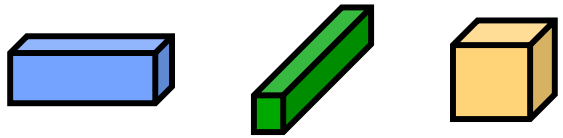




Energy of weight depends on **memory hierarchy** and **dataflow**

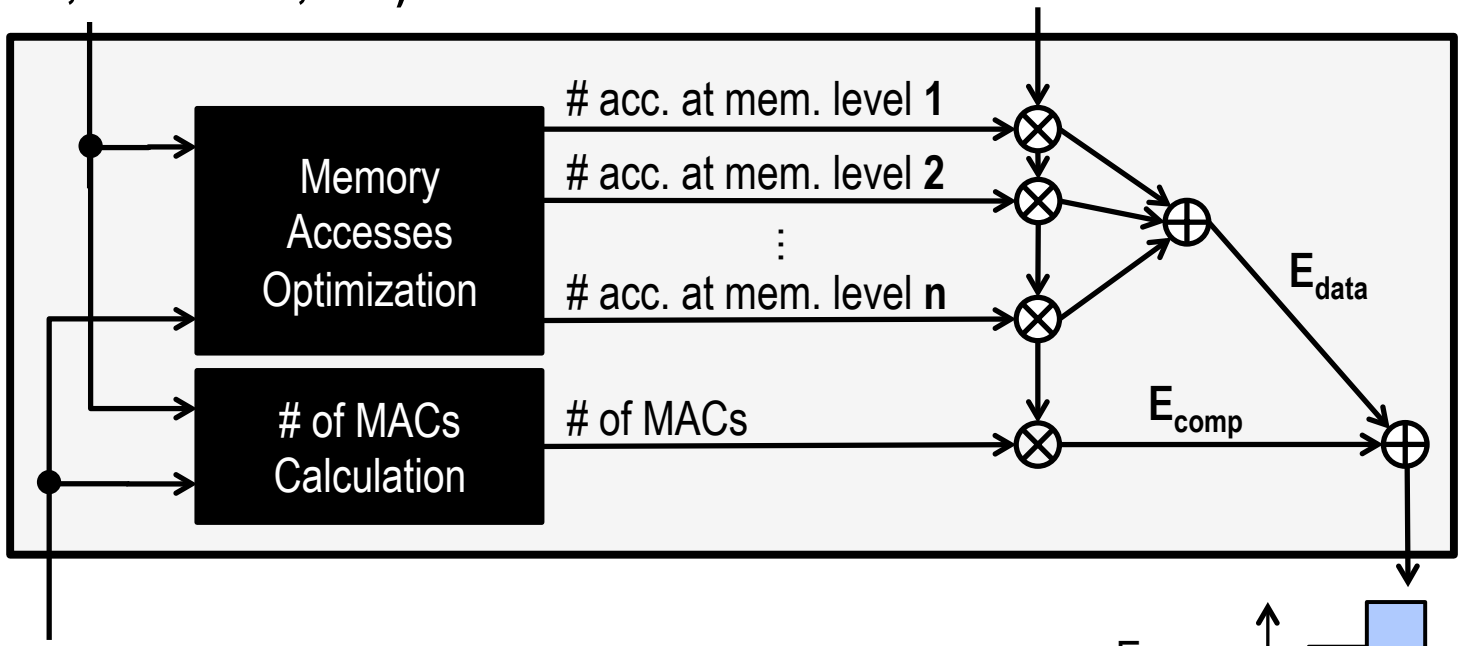
* measured from a commercial 65nm process

Energy-Evaluation Methodology

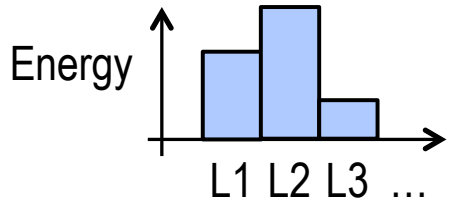


DNN Shape Configuration
(# of channels, # of filters, etc.)

Hardware Energy Costs of each
MAC and Memory Access



DNN Weights and Input Data
[0.3, 0, -0.4, 0.7, 0, 0, 0.1, …]



DNN Energy Consumption

Energy estimation tool available at eyeriss.mit.edu

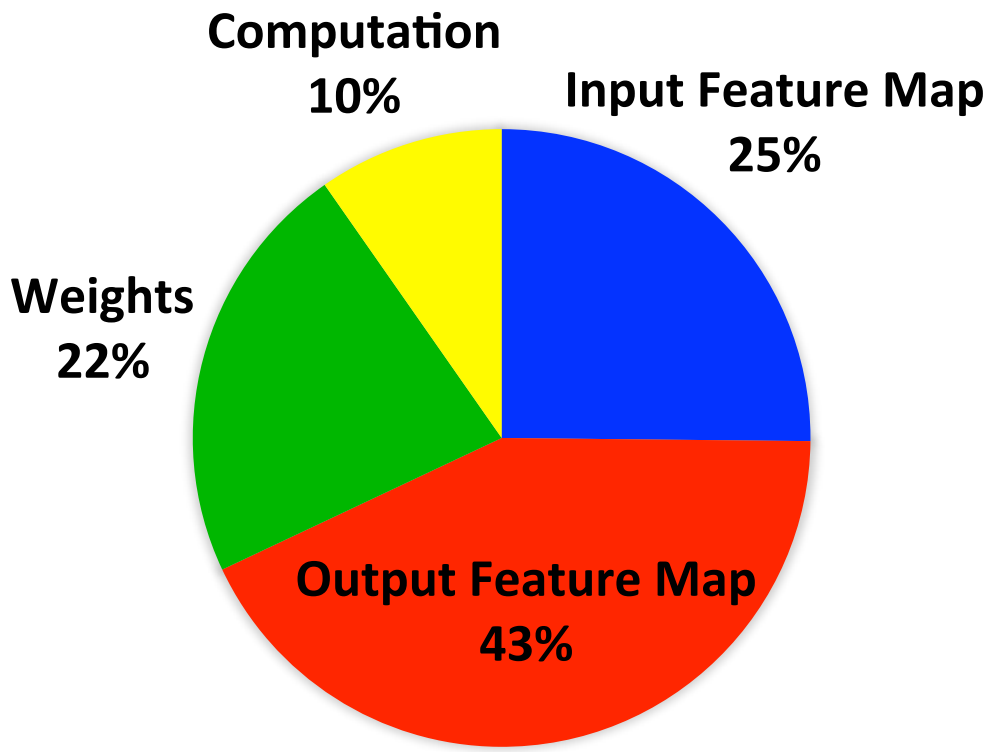
[Yang et al., CVPR 2017]

Key Observations

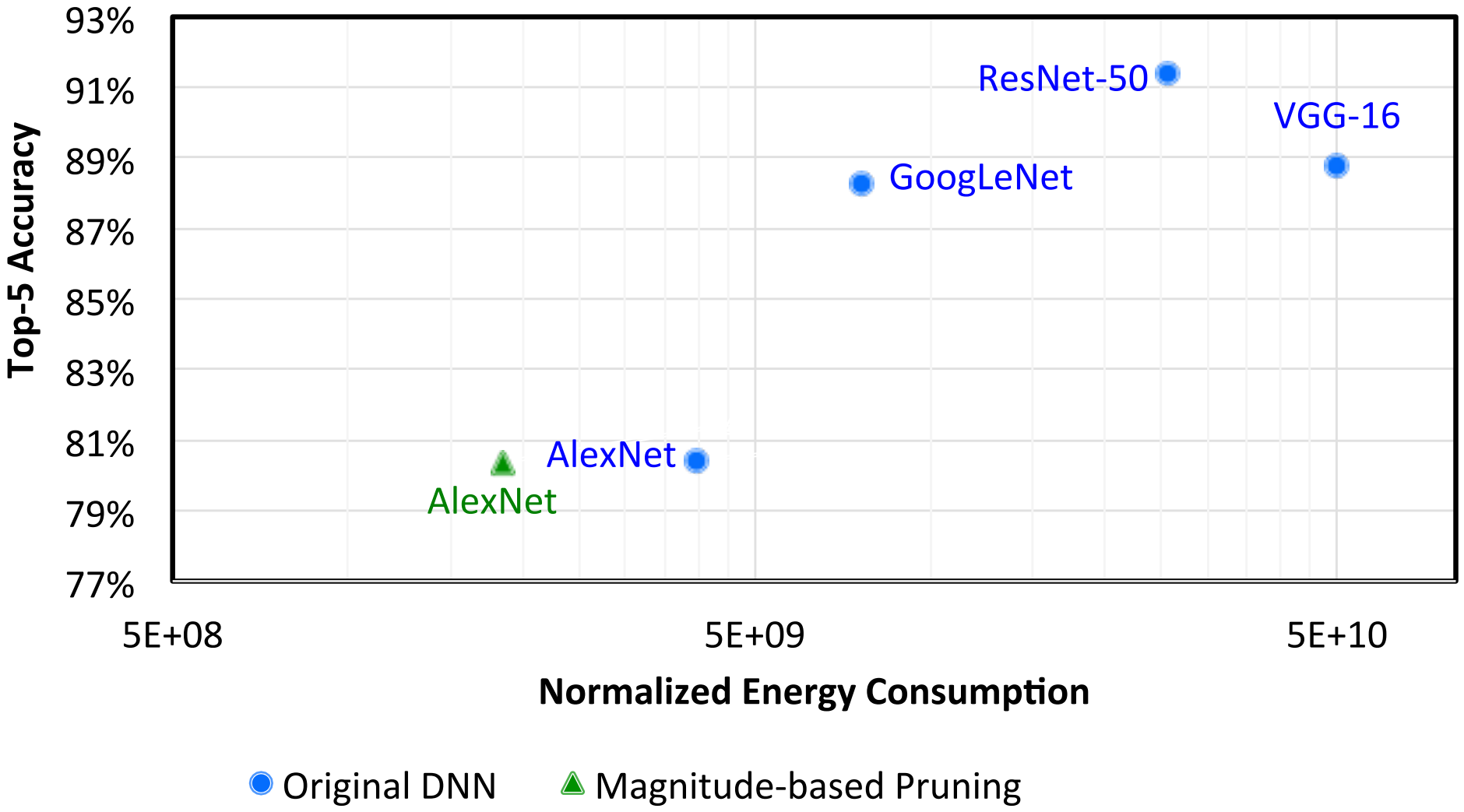
- Number of weights *alone* is not a good metric for energy
- All data types should be considered

Energy Consumption of GoogLeNet

[Yang et al., CVPR 2017]



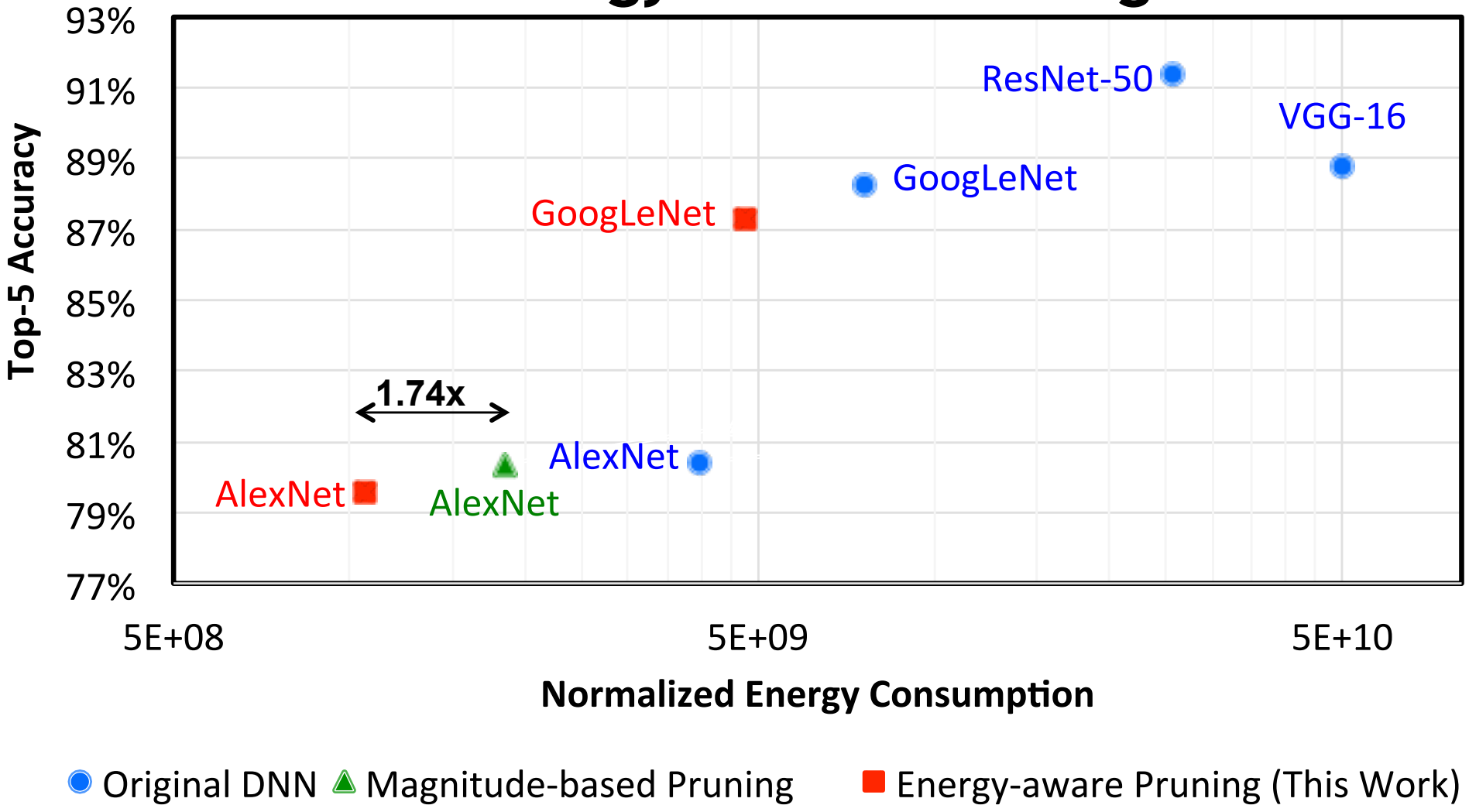
Magnitude-based Weight Pruning



[Yang et al., CVPR 2017]

Reduce number of weights by removing small magnitude weights

Energy-Aware Pruning

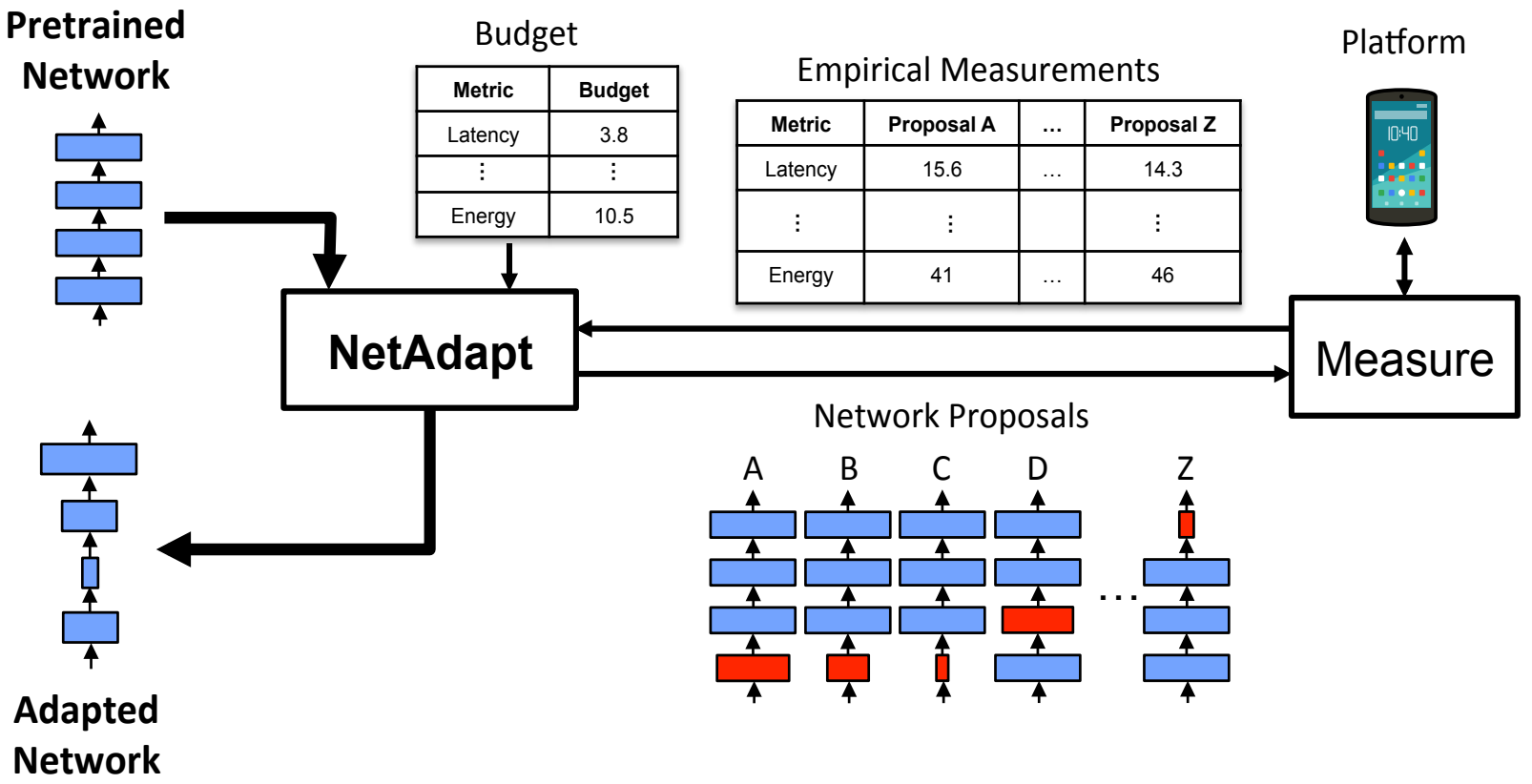


[Yang et al., CVPR 2017]

Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

NetAdapt: Platform-Aware DNN Adaptation

- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)

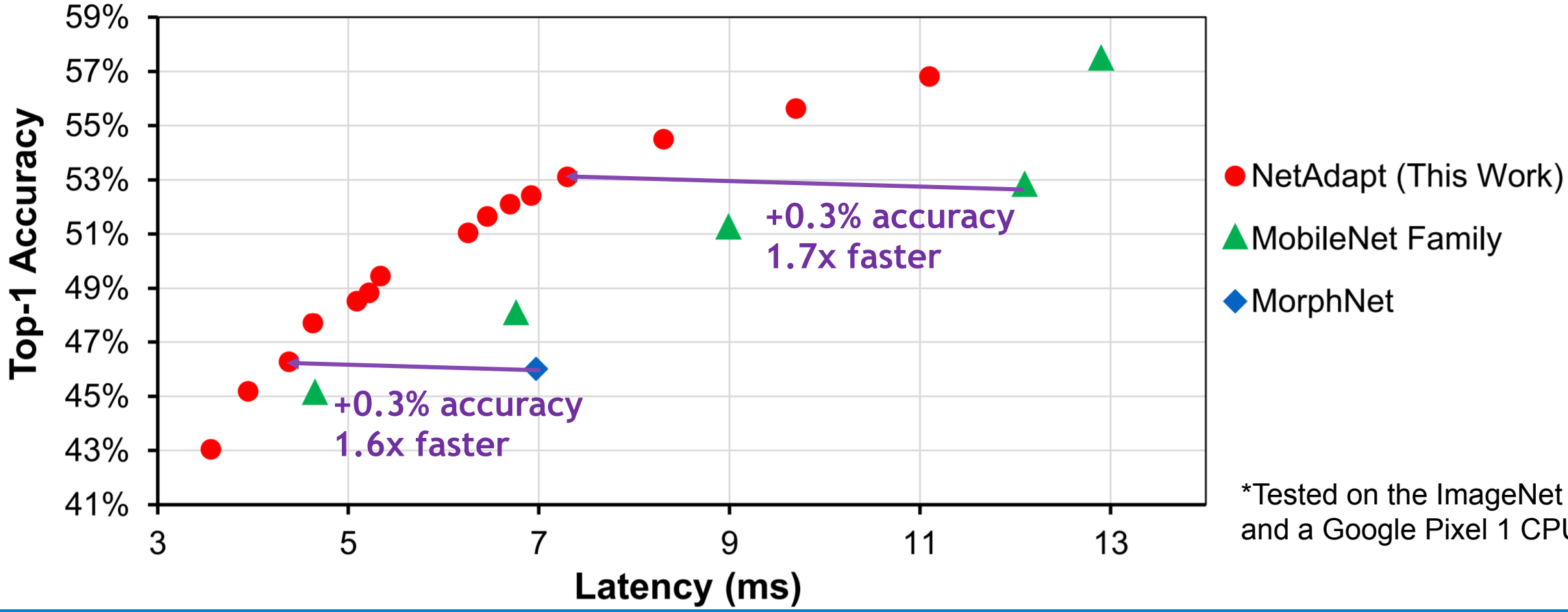


In collaboration with Google's Mobile Vision Team

[Yang et al., arXiv 2018]

Improved Latency vs. Accuracy Tradeoff

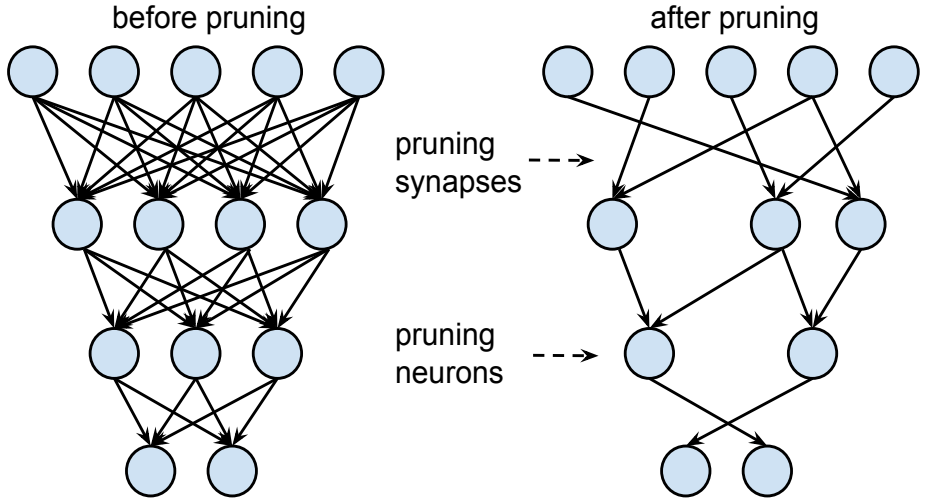
- NetAdapt boosts the real inference speed of MobileNet by up to 1.7x with higher accuracy



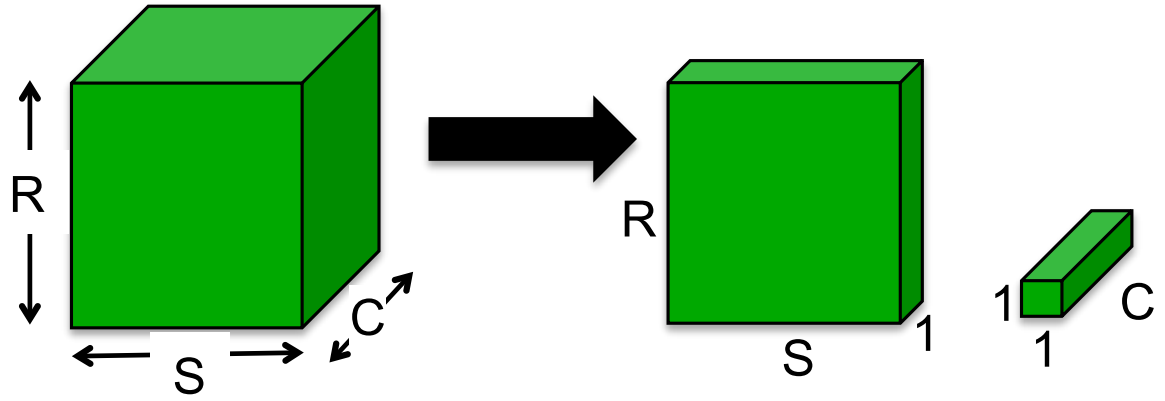
*Tested on the ImageNet dataset and a Google Pixel 1 CPU

Many Efficient DNN Design Approaches

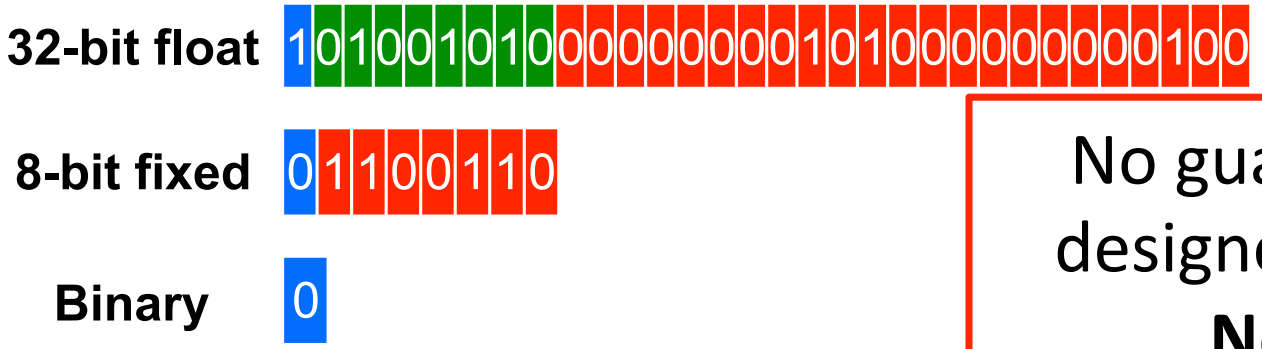
Network Pruning



Compact Network Architectures



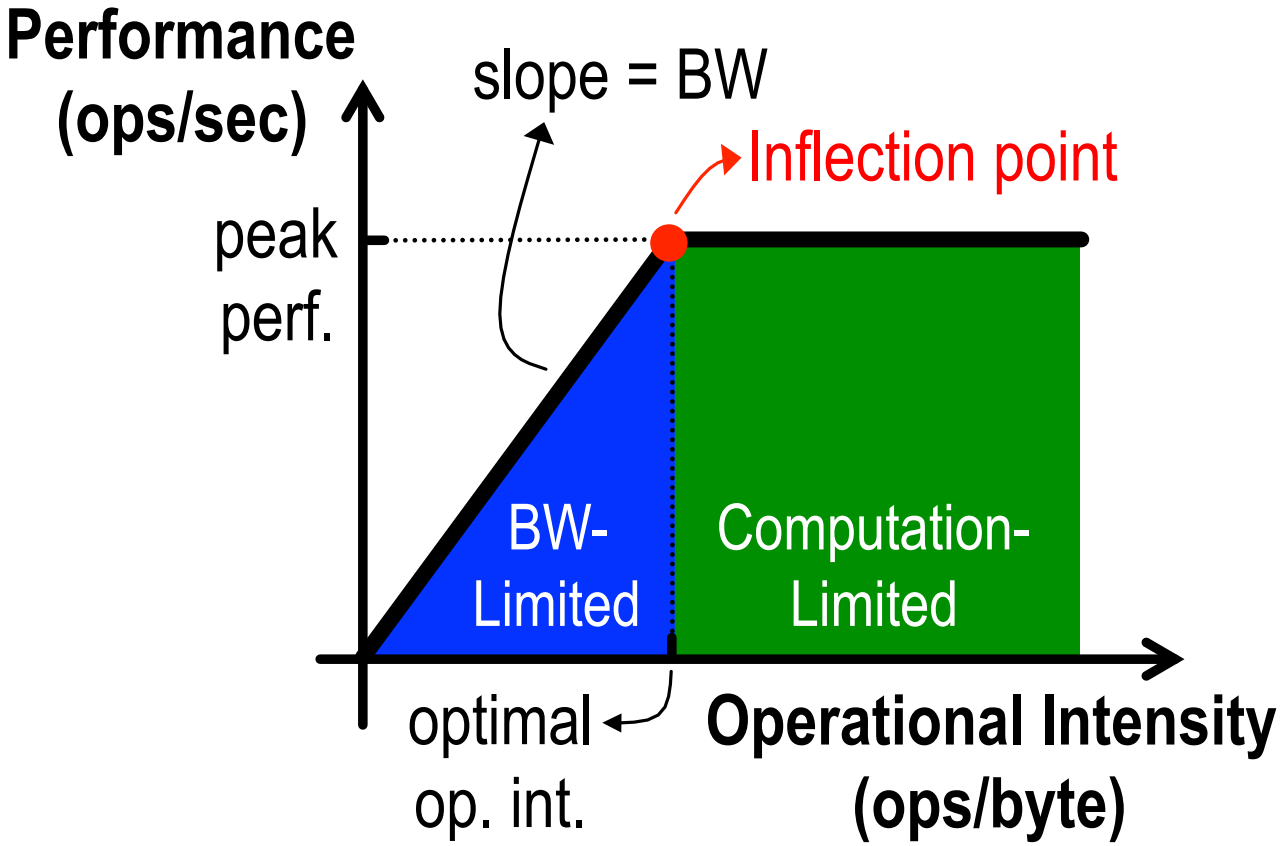
Reduce Precision

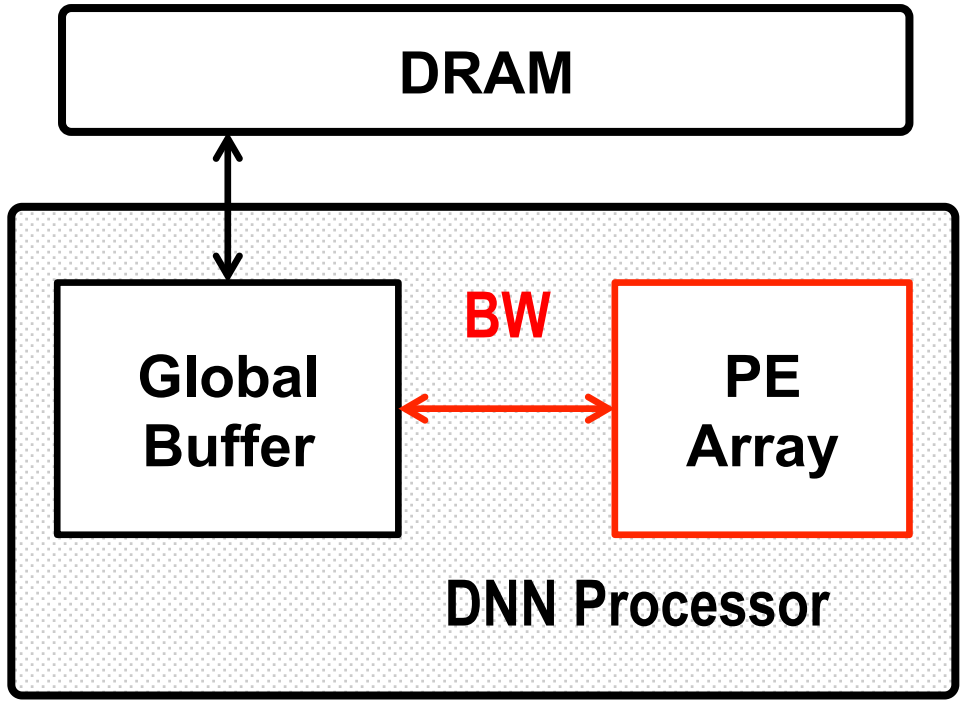


No guarantee that DNN algorithm designer will use a given approach.
Need flexible hardware!

Roofline Model

A tool that visualizes the performance of an architecture under various degrees of operational intensity

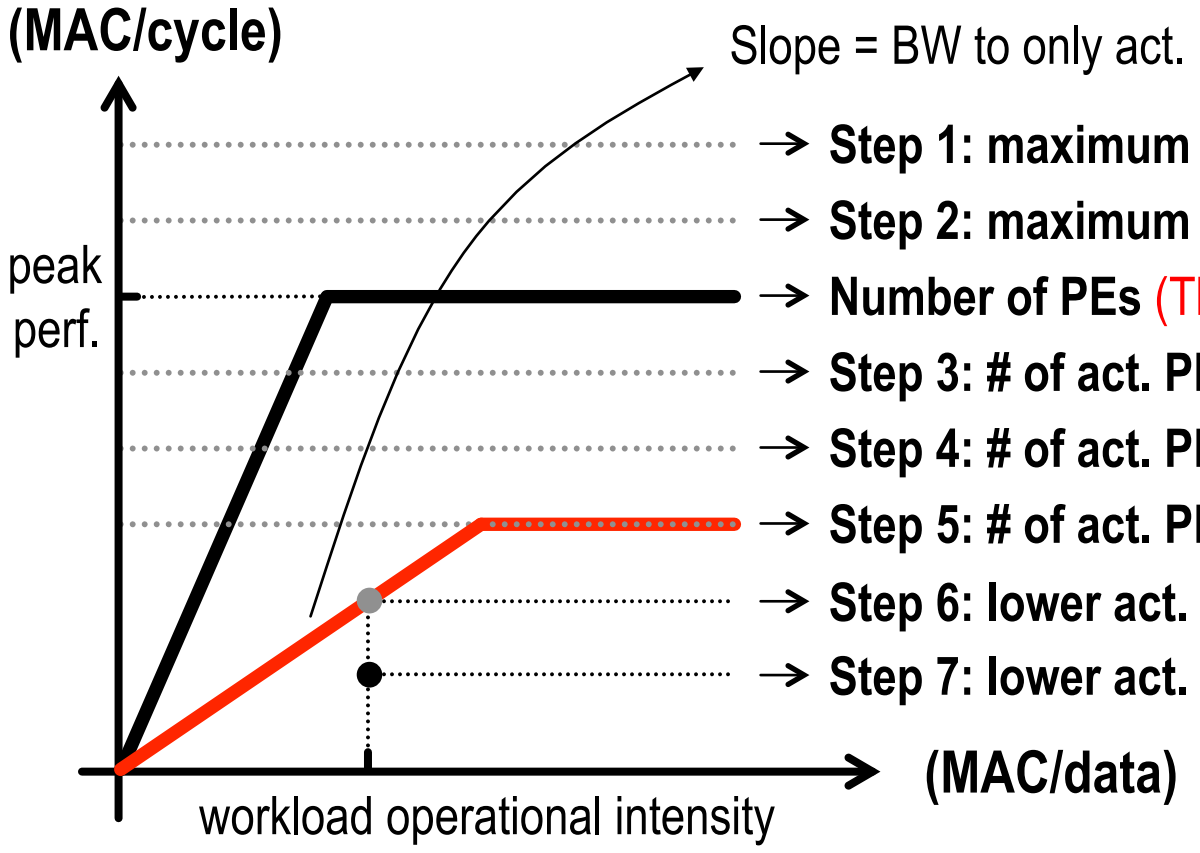




[Williams et al., Comm ACM 2009]

Eyexam: Understanding Sources of Inefficiencies in DNN Accelerators

A systematic way to evaluate how each architectural decision affects performance (throughput) for a given DNN workload



Slope = BW to only act. PE

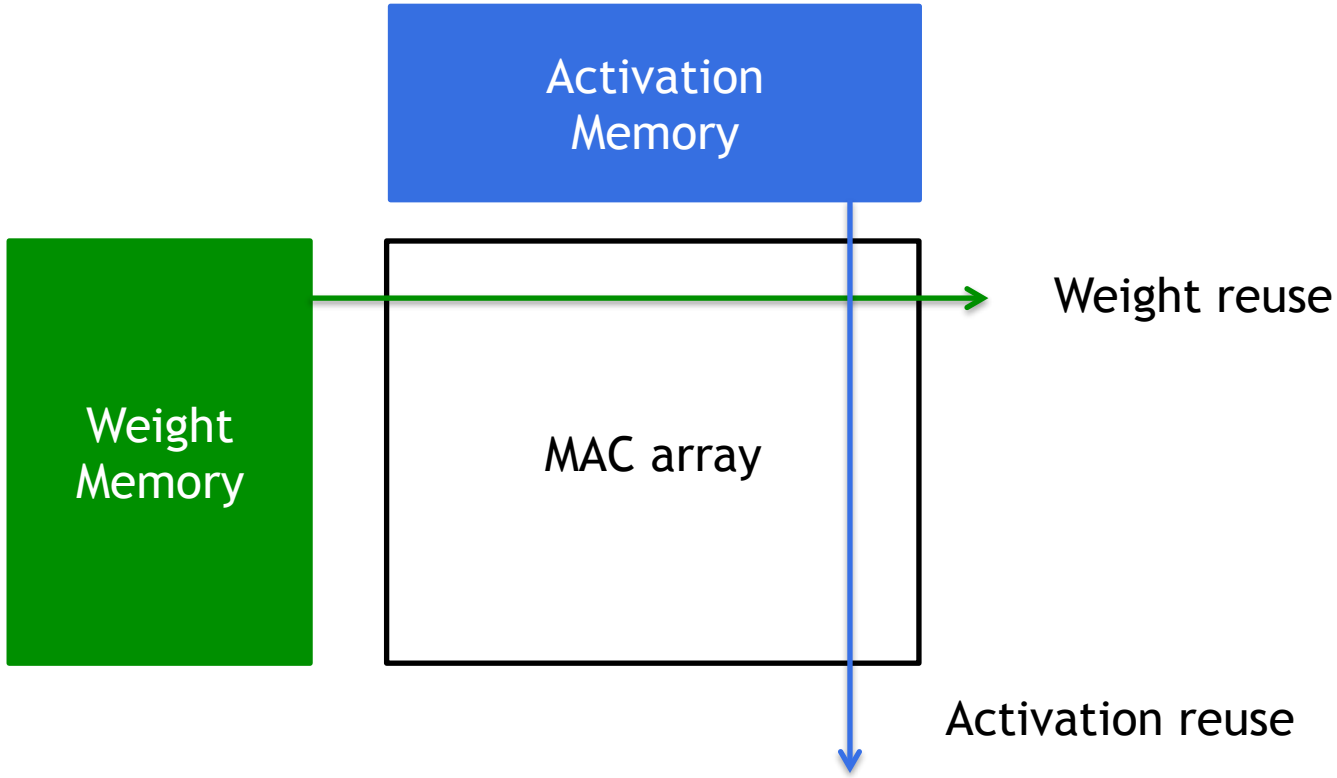
Tightens the roofline model

- → **Step 1: maximum workload parallelism**
- → **Step 2: maximum dataflow parallelism**
- → **Number of PEs** (Theoretical Peak Performance)
- → **Step 3: # of act. PEs under a finite PE array size**
- → **Step 4: # of act. PEs under fixed PE array dimensions**
- → **Step 5: # of act. PEs under fixed storage capacity**
- → **Step 6: lower act. PE utilization due to insufficient avg. BW**
- → **Step 7: lower act. PE utilization due to insufficient inst. BW**

[Chen et al., Eyeriss v2, arXiv 2018]

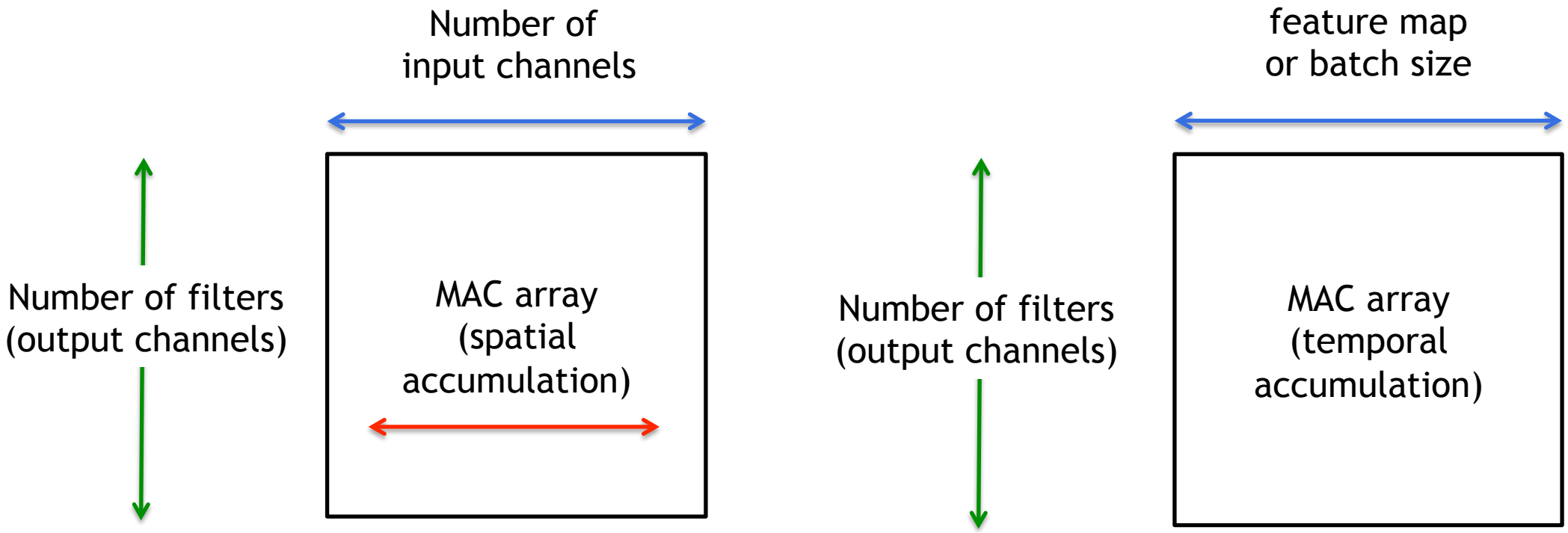
Existing DNN Architectures

- Specialized DNN hardware often rely on certain properties of DNN in order to achieve high energy-efficiency
- Example: Reduce memory access by amortizing across MAC array



Limitation of Existing DNN Architectures

- Example: reuse depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)
  - Can be challenging to exploit sparsity



Existing Sparse DNN Architectures

- Sparse DNN architectures translate sparsity from pruning into improved energy-efficiency and throughput
  - Perform only non-zero MACs and move data in compressed format
- Existing sparse DNN architectures optimized for either CONV or FC layer due to different BW and data reuse requirements
- Efficient for sparse DNNs, but overhead for dense DNNs
  - Compressed format results in memory overhead for dense DNNs
  - Additional control to identify location of non-zero values results in energy overhead for dense DNNs

Since there is **no guarantee in degree of sparsity**,
it is important to **evaluate the overhead on dense DNNs**

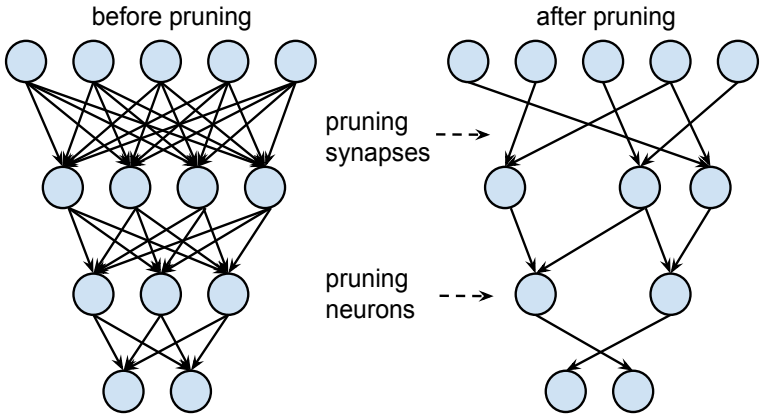
Need More Comprehensive Benchmarks

Processors should support a **diverse set of DNNs** that utilize different techniques

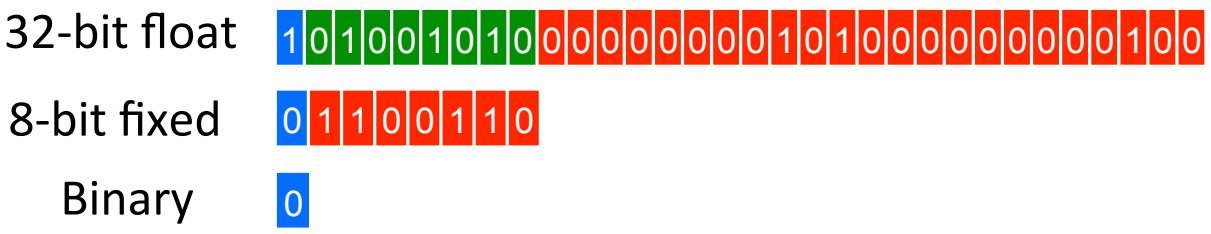
Example:

- Sparse **and** Dense
- Large **and** Compact network architectures
- Different Layers (e.g., CONV **and** FC)
- Variable Bit-width

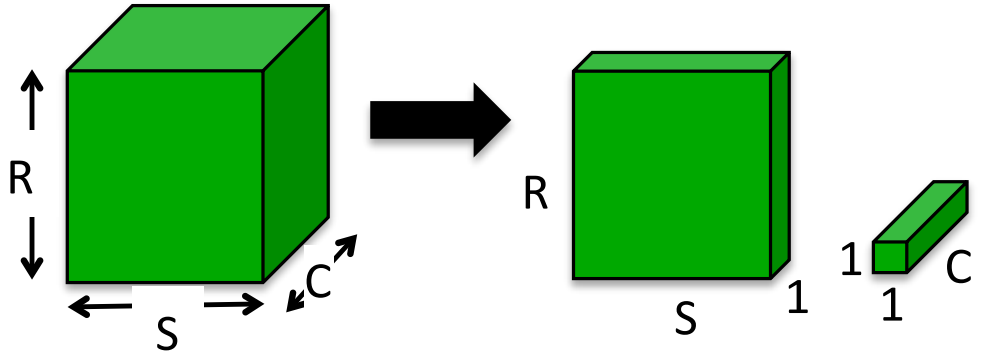
Network Pruning



Reduce Precision



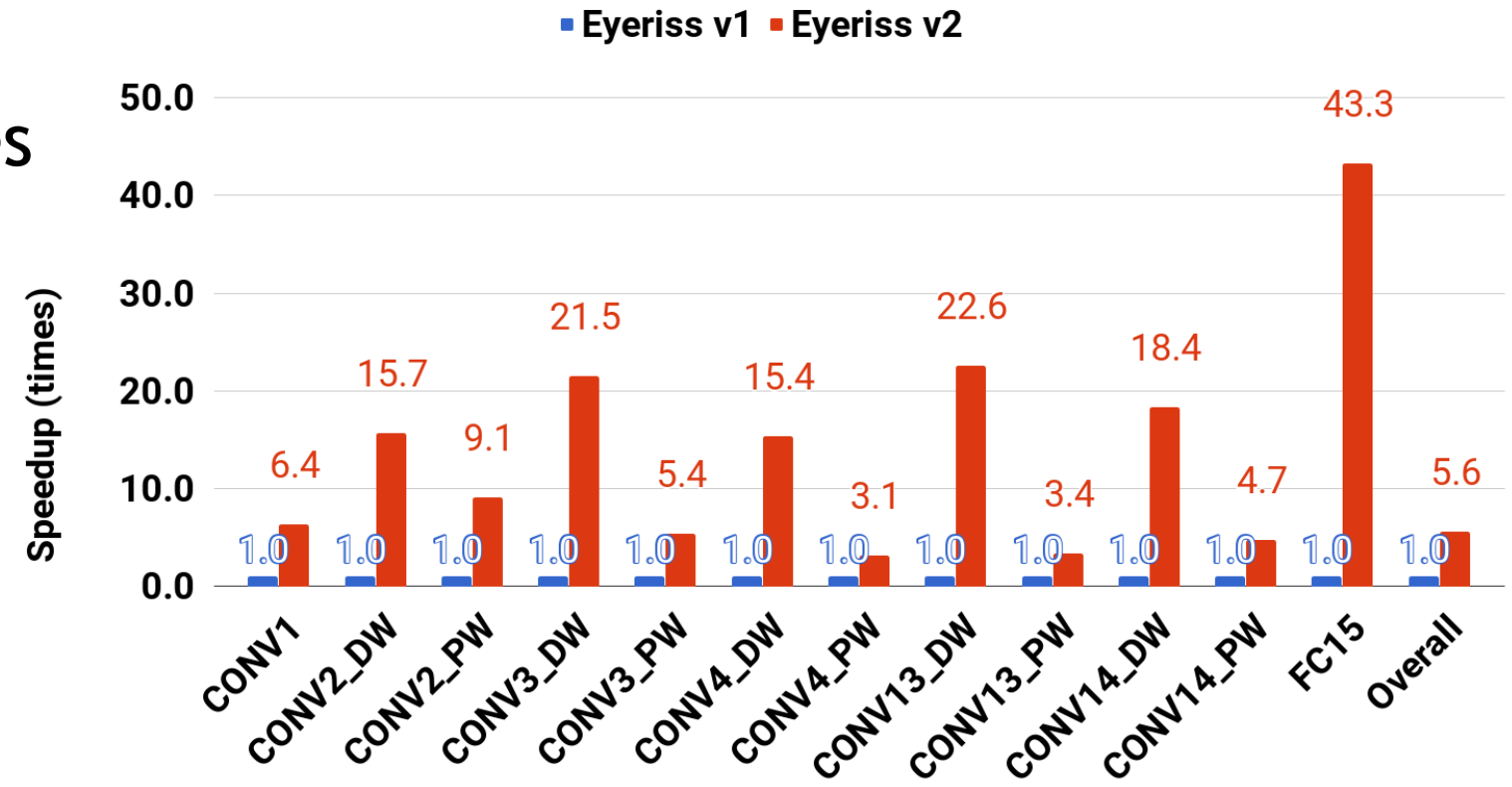
Compact Network Architecture



Eyeriss v2: Balancing Flexibility and Efficiency

Efficiently supports

- Wide range of filter shapes
  - Large and Compact
- Different Layers
  - e.g., CONV and FC
- Wide range of sparsity
  - Dense and Sparse



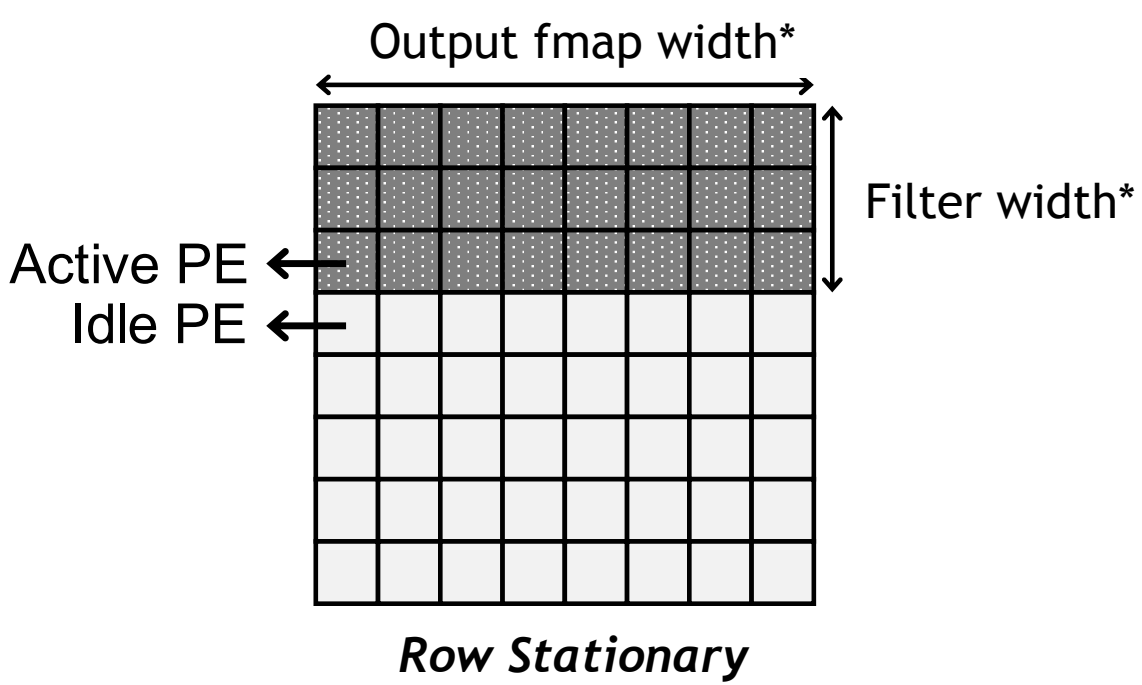
Over an order of magnitude faster and more energy efficient than Eyeriss v1

[Chen et al., arXiv 2018]

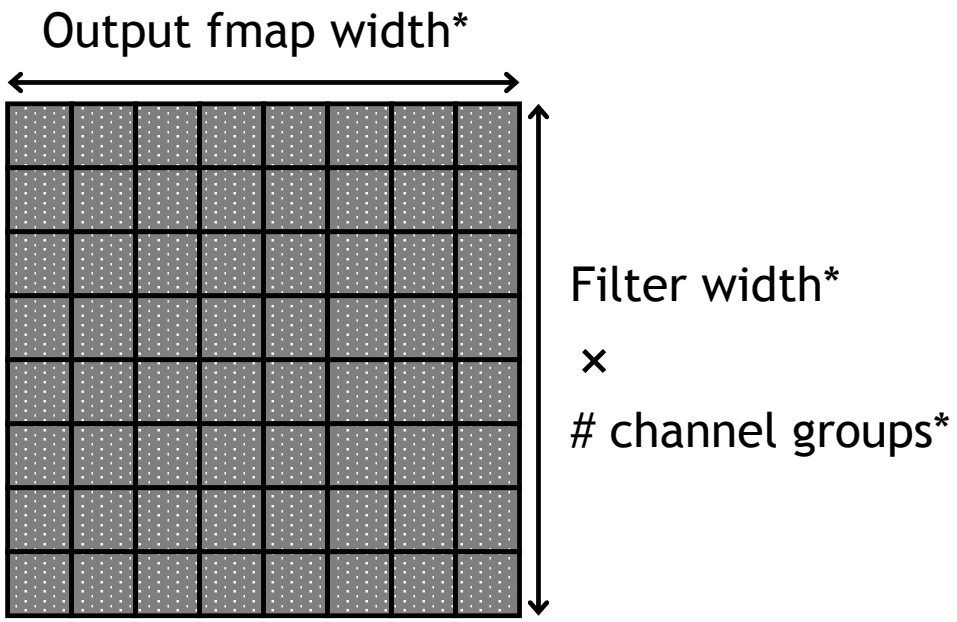
eyeriss.mit.edu

Eyeriss v2: Balancing Flexibility and Efficiency

- **Flexible dataflow**, called **Row-Stationary Plus (RS+)**, that enables the spatial tiling of data from all dimensions for high PE array utilization and data reuse for various layer shapes and sizes



Row Stationary



Row Stationary Plus

*tiling parameters

[Chen et al., arXiv 2018]

eyeriss.mit.edu

Eyeriss v2: Balancing Flexibility and Efficiency

- **Flexible dataflow**, called **Row-Stationary Plus (RS+)**, that enables the spatial tiling of data from all dimensions for high PE array utilization and data reuse for various layer shapes and sizes
- **Flexible NoC** to support RS+ that can operate in different modes for different requirements
  - Utilizes multicast to exploit spatial data reuse
  - Utilizes unicast for high BW for weights for FC and weights & activations for compact network architectures
- Processes data in both compressed and raw format to minimize data movement for both CONV and FC layers
  - Exploit sparsity in **both** weights and activations

[Chen et al., arXiv 2018]

eyeriss.mit.edu

Benchmarking Metrics for DNN Hardware

How can we compare designs?

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

Metrics for DNN Hardware

- **Accuracy**
  - Quality of result for a given task
- **Throughput**
  - Analytics on high volume data
  - Real-time performance (e.g., video at 30 fps)
- **Latency**
  - For interactive applications (e.g., autonomous navigation)
- **Energy and Power**
  - Edge and embedded devices have limited battery capacity
  - Data centers have stringent power ceilings due to cooling costs
- **Hardware Cost**
  - $$$

Specifications to Evaluate Metrics

- **Accuracy**
  - Difficulty of dataset and/or task should be considered
- **Throughput**
  - Number of cores (include utilization along with peak performance)
  - Runtime for running specific DNN models
- **Latency**
  - Include batch size used in evaluation
- **Energy and Power**
  - Power consumption for running specific DNN models
  - Include external memory access
- **Hardware Cost**
  - On-chip storage, number of cores, chip area + process technology

Example: Metrics of Eyeriss Chip

ASIC Specs	Input
Process Technology	65nm LP TSMC (1.0V)
Total Core Area (mm^2)	12.25
Total On-Chip Memory (kB)	192
Number of Multipliers	168
Clock Frequency (MHz)	200
Core area (mm^2) / multiplier	0.073
On-Chip memory (kB) / multiplier	1.14
Measured or Simulated	Measured

Metric	Units	Input
Name of CNN Model	Text	AlexNet
Top-5 error classification on ImageNet	#	19.8
Supported Layers		All CONV
Bits per weight	#	16
Bits per input activation	#	16
Batch Size	#	4
Runtime	ms	115.3
Power	mW	278
Off-chip Access per Image Inference	MBytes	3.85
Number of Images Tested	#	100

Comprehensive Coverage

- All metrics should be reported for fair evaluation of design tradeoffs
- Examples of what can happen if certain metric is omitted:
  - **Without the accuracy given for a specific dataset and task**, one could run a simple DNN and claim low power, high throughput, and low cost – however, the processor might not be usable for a meaningful task
  - **Without reporting the off-chip bandwidth**, one could build a processor with only multipliers and claim low cost, high throughput, high accuracy, and low chip power – however, when evaluating system power, the off-chip memory access would be substantial
- Are results measured or simulated? On what test data?

Evaluation Process

The evaluation process for whether a DNN system is a viable solution for a given application might go as follows:

1. **Accuracy** determines if it can perform the given task
2. **Latency** and throughput determine if it can run fast enough and in real-time
3. **Energy** and power consumption will primarily dictate the form factor of the device where the processing can operate
4. **Cost**, which is primarily dictated by the chip area, determines how much one would pay for this solution

Summary

- The number of weights and MACs are not sufficient for evaluating the energy consumption and latency of DNNs
  - Designers of efficient DNN algorithms should directly target direct metrics such as energy and latency and incorporate into the design
- Many of the existing DNN processors rely on certain properties of the DNN which cannot be guaranteed as the wide range techniques used for efficient DNN algorithm design has resulted in a more diverse set of DNNs
  - DNN hardware used to process these DNNs should be sufficiently flexible to support a wide range of techniques efficiently
- Evaluate DNN hardware on a comprehensive set of benchmarks and metrics

References

- Overview Paper
  - V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "*Efficient Processing of Deep Neural Networks: A Tutorial and Survey*", Proceedings of the IEEE, 2017
- More info about Eyeriss and Tutorial on DNN Architectures
  http://eyeriss.mit.edu
- MIT Professional Education Course on "Designing Efficient Deep Learning Systems" ***Next offering: July 23-24, 2018 on MIT campus***
  http://professional-education.mit.edu/deeplearning

For updates on Eyerissv2, Eyexam, NetAdapt, etc. Follow @eems_mit

or join EEMS news mailing list: http://mailman.mit.edu/mailman/listinfo/eems-news

References

- *Y.-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE International Conference on Solid-State Circuits (ISSCC), pp. 262-264, February 2016.*
- *Y.-H. Chen, J. Emer, V. Sze, "Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks," International Symposium on Computer Architecture (ISCA), pp. 367-379, June 2016.*
- *A. Suleiman, Z. Zhang, V. Sze, "A 58.6mW Real-time Programmable Object Detection with Multi-Scale Multi-Object Support Using Deformable Parts Models on 1920×1080 Video at 30fps," IEEE Symposium on VLSI Circuits (VLSI-Circuits), pp. 184-185, June 2016.*
- *A. Suleiman*, Y.-H. Chen*, J. Emer, V. Sze, "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," IEEE International Symposium of Circuits and Systems (ISCAS), Invited Paper, May 2017.*
- *T.-J. Yang, Y.-H. Chen, V. Sze, "Designing Energy-Efficient Convolutional Neural Networks using Energy-Aware Pruning," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.*
- *T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, V. Sze, H. Adam, "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," arXiv, April 2018.*
- *V. Sze, Y.-H. Chen, T.-J. Yang, J. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, December 2017.*
- *Y.-H. Chen*, T.-J. Yang*, J. Emer, V. Sze, "Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," SysML Conference, February 2018.*