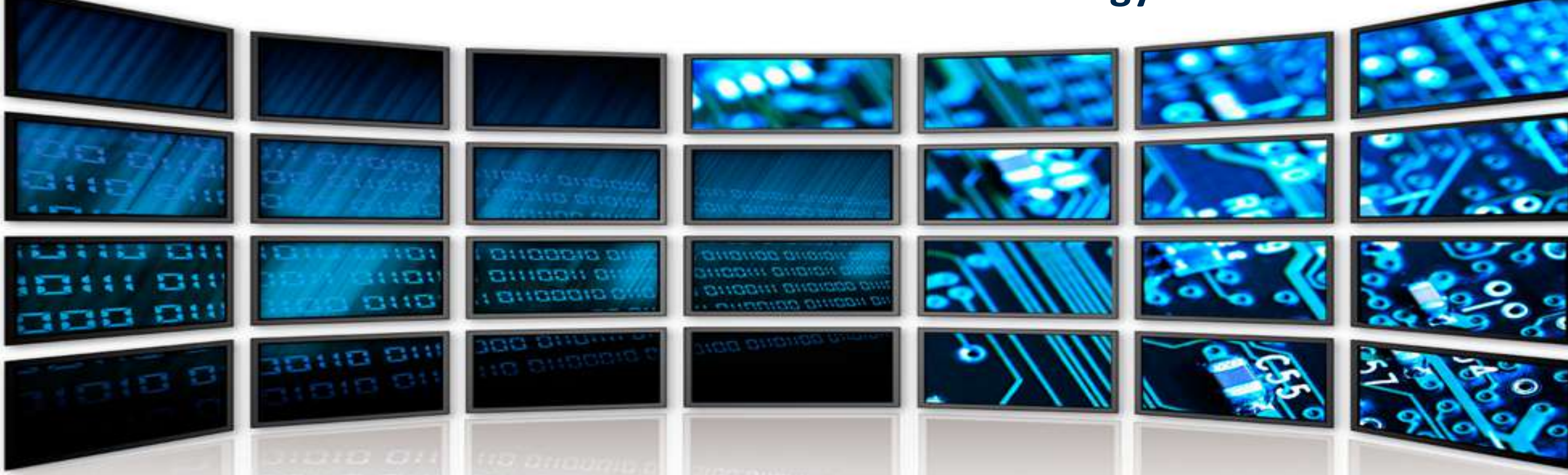
Understanding the Challenges of Algorithm and Hardware Co-design for Deep Neural Networks

Vivienne Sze

Massachusetts Institute of Technology



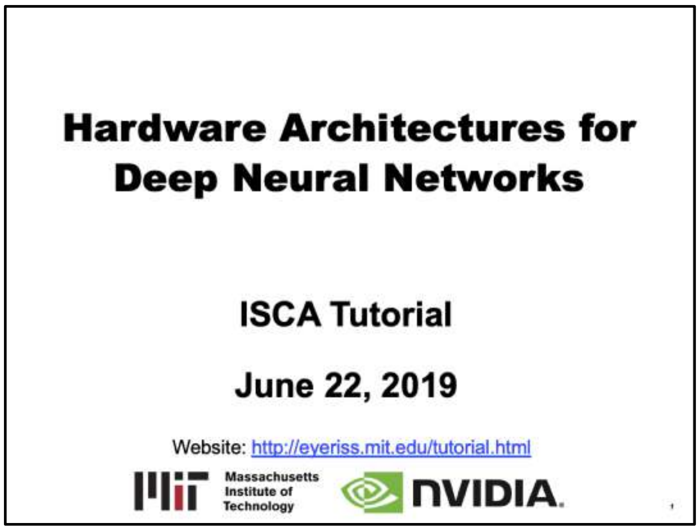
In collaboration with Yu-Hsin Chen, Joel Emer, Sertac Karaman, Fangchang Ma, Diana Wofk, Tien-Ju Yang, Zhengdong Zhang, Google Mobile Vision Team

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

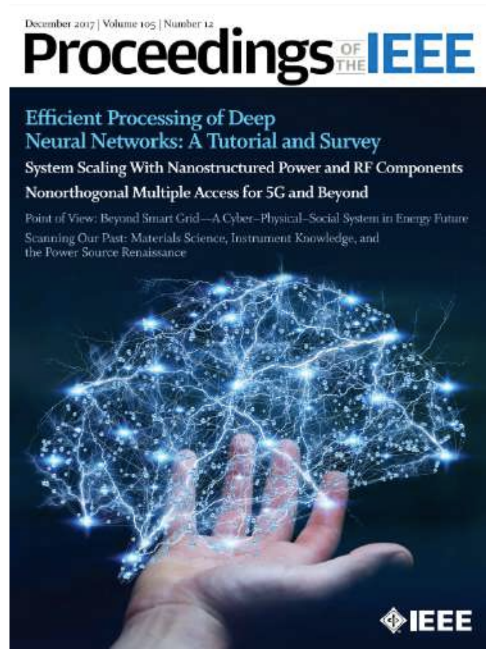
Follow @eems_mit

Energy-Efficient Processing of DNNs

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



http://eyeriss.mit.edu/tutorial.html



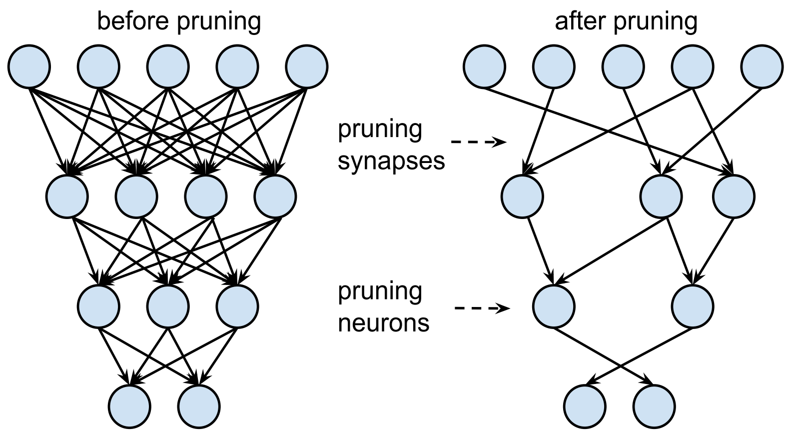
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various challenges to existing approaches

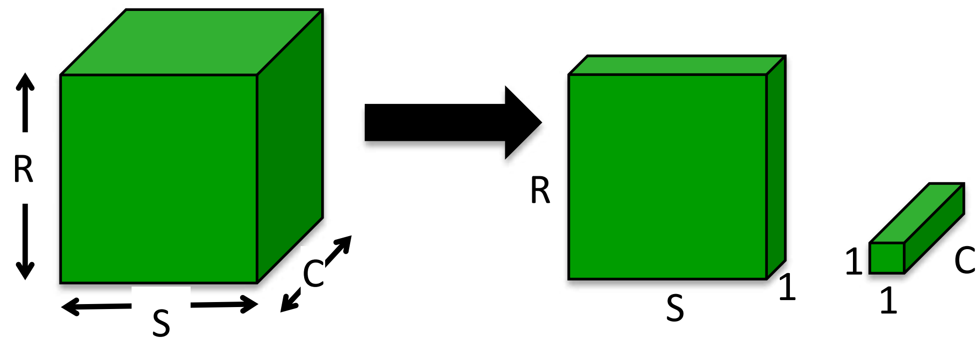
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

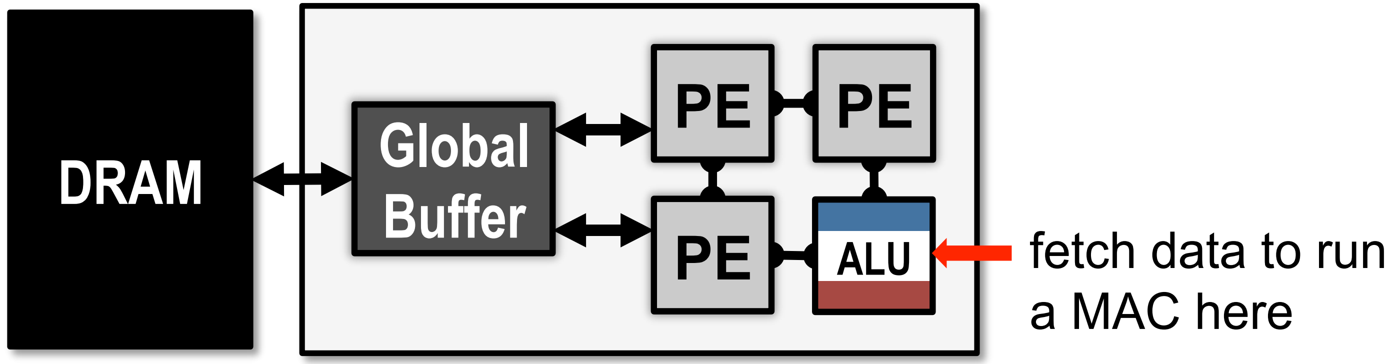


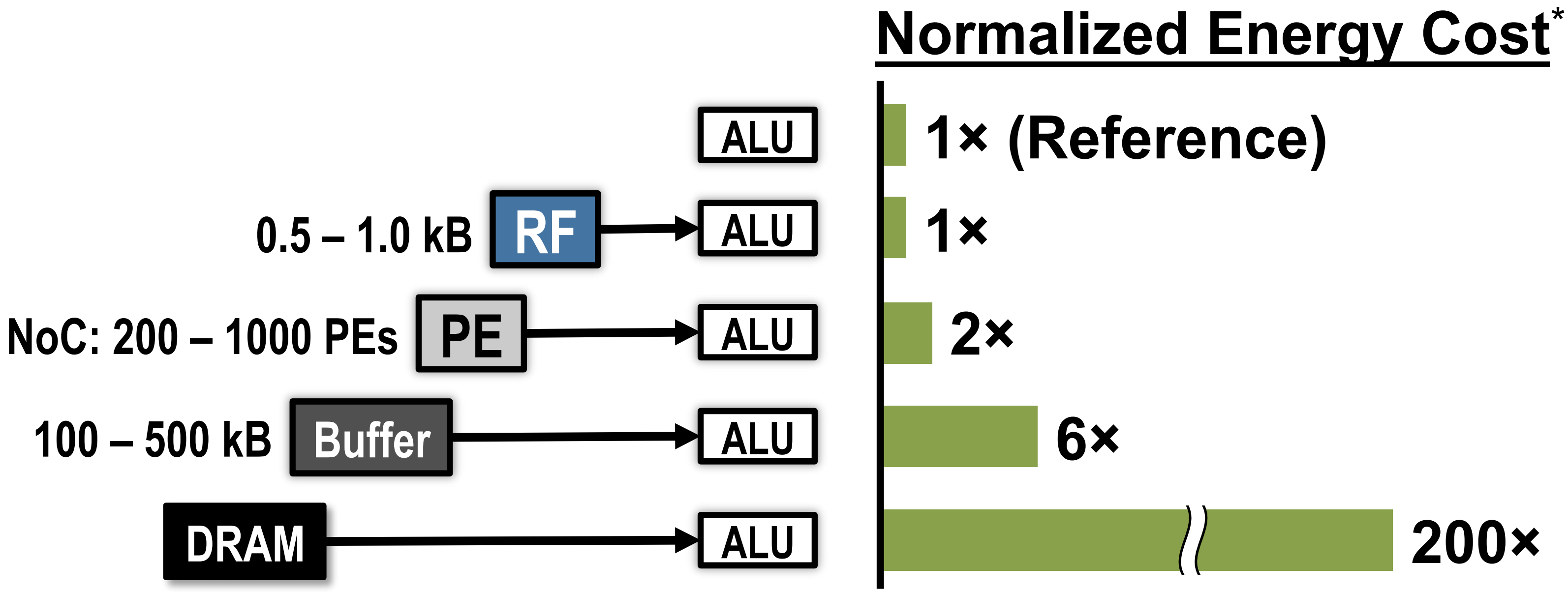
Examples: SqueezeNet, MobileNet

... also reduced precision

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

Data Movement is Expensive

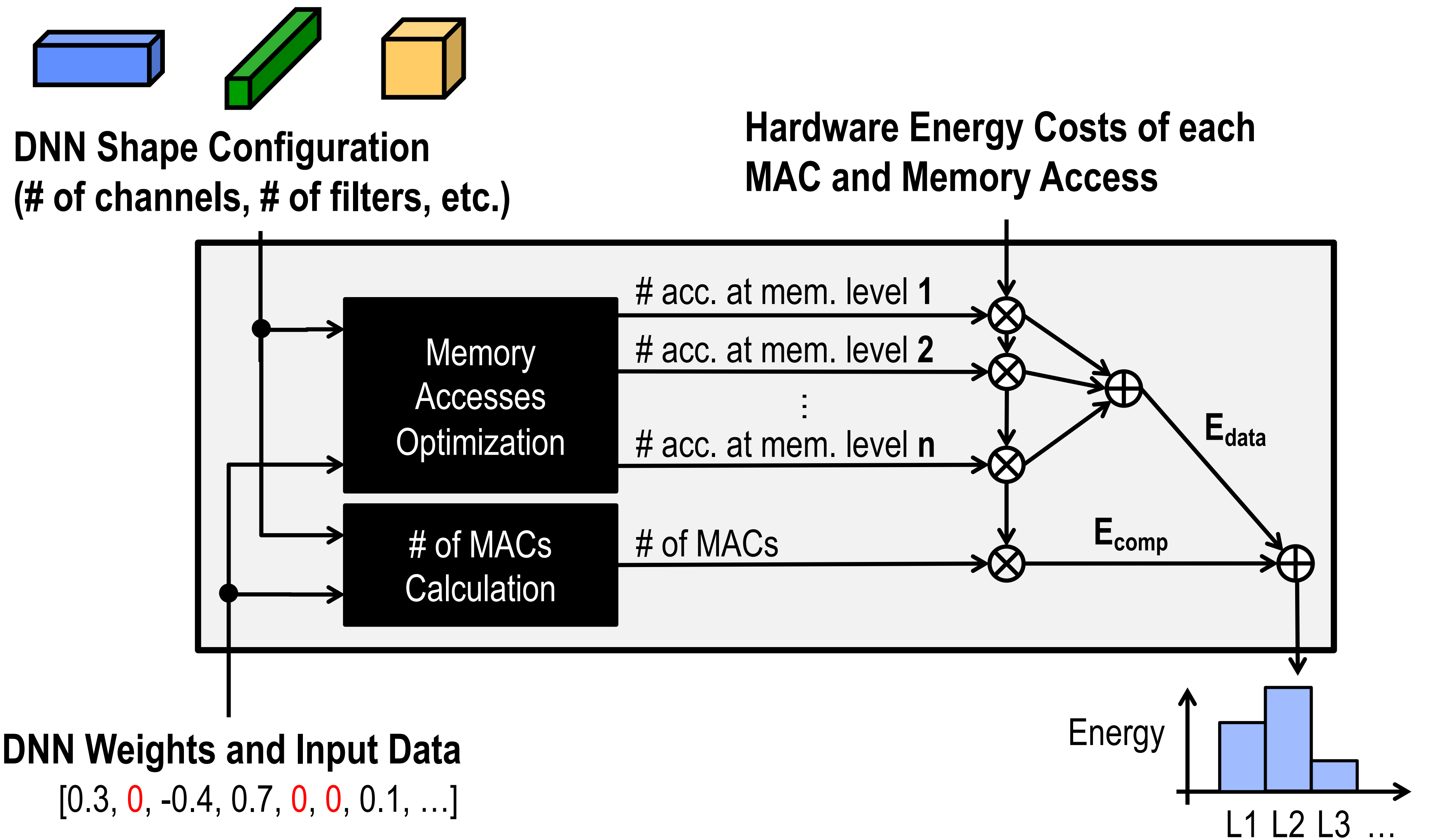




* measured from a commercial 65nm process

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

Energy-Evaluation Methodology



DNN Energy Consumption

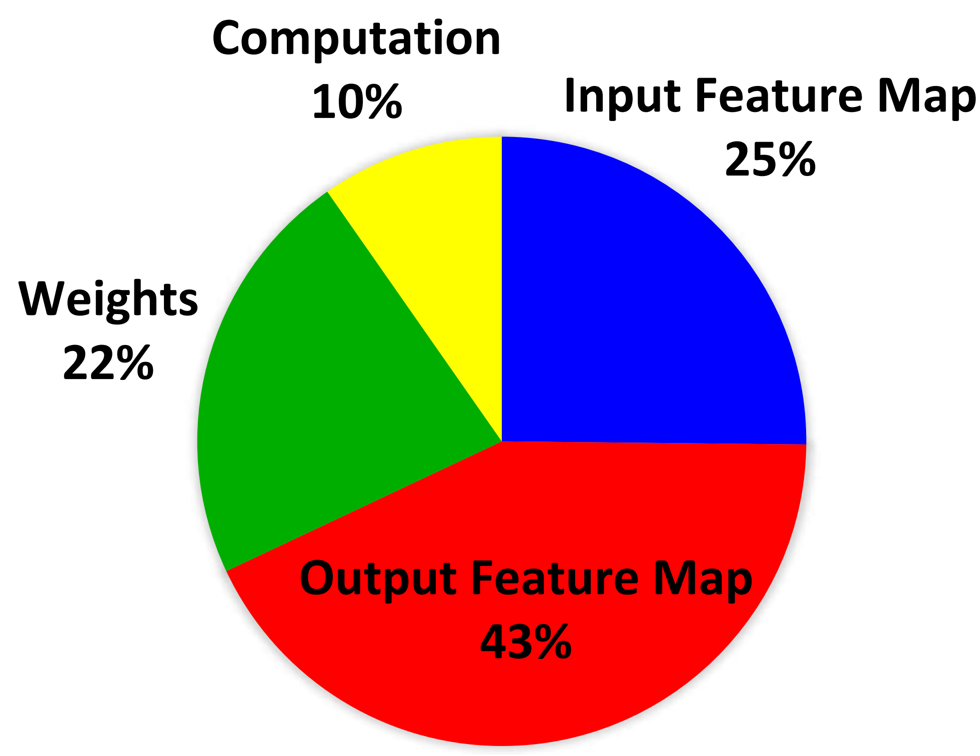
Tool available at: https://energyestimation.mit.edu/

[Yang et al., CVPR 2017]

Key Observations

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

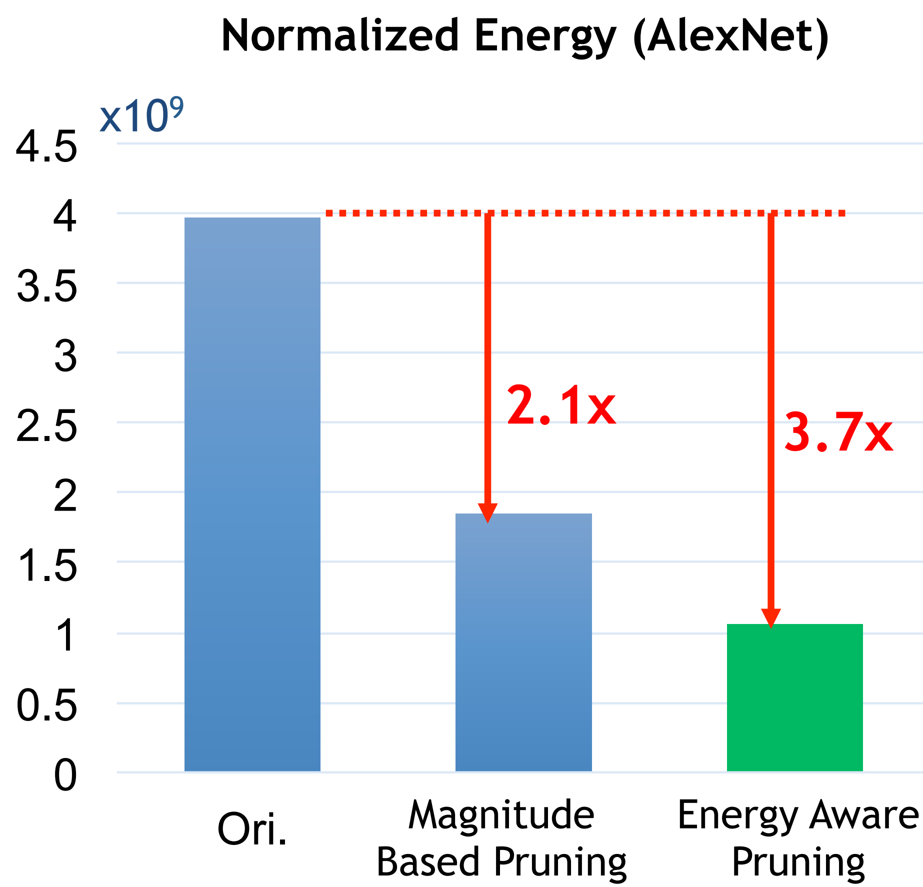
Energy Consumption of GoogLeNet



Energy-Aware Pruning

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

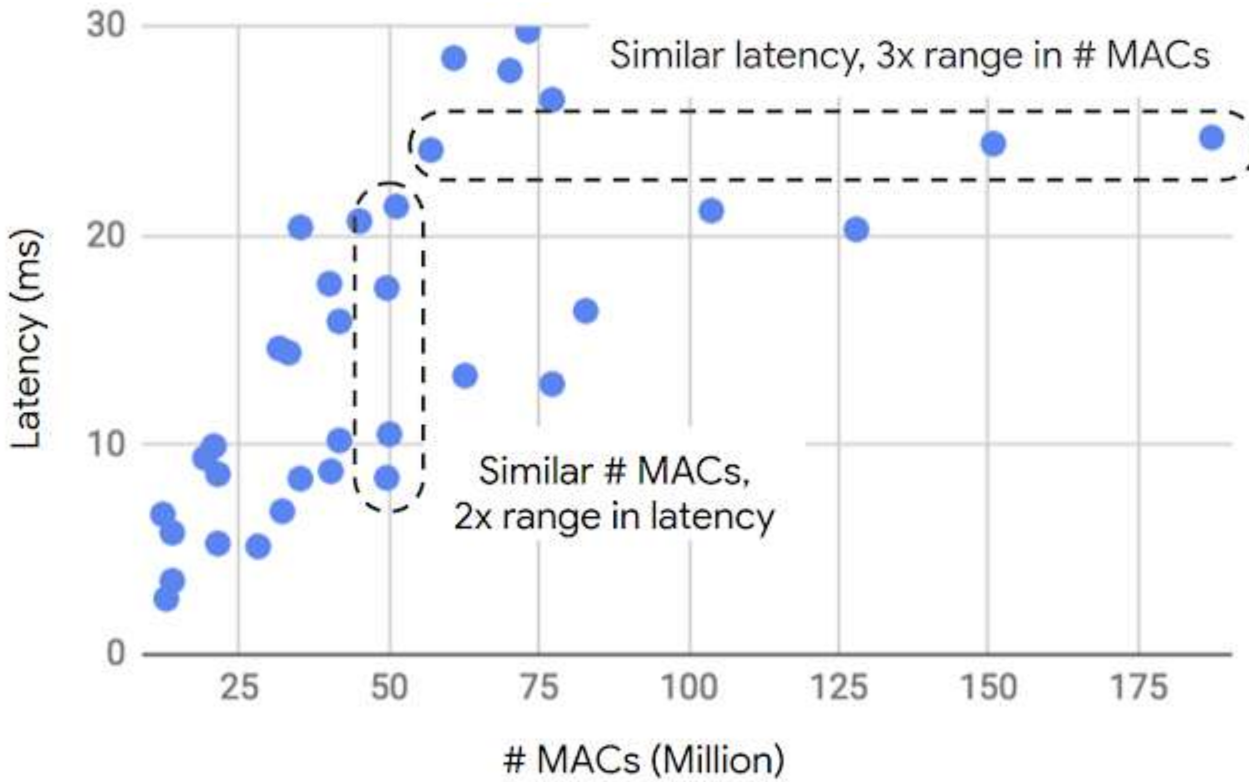
- Sort layers based on energy and prune layers that consume most energy first
- EAP reduces AlexNet energy by **3.7x** and outperforms the previous work that uses magnitude-based pruning by **1.7x**



Pruned models available at
http://eyeriss.mit.edu/energy.html

of Operations vs. Latency

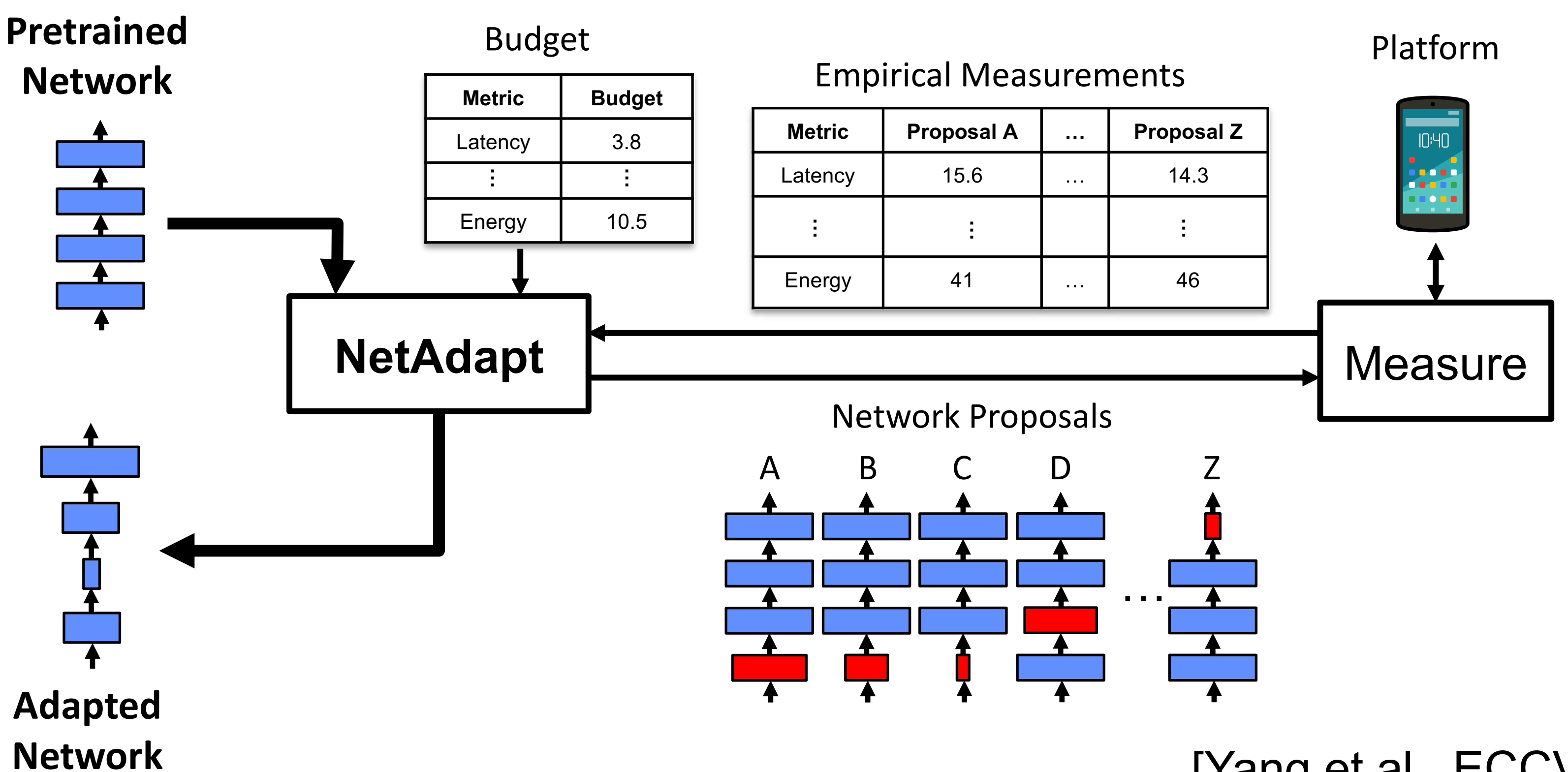
- # of operations (MACs) does not approximate latency well



Source: Google (https://ai.googleblog.com/2018/04/introducing-cvpr-2018-on-device-visual.html)

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)



Budget

Metric	Budget
Latency	3.8
⋮	⋮
Energy	10.5

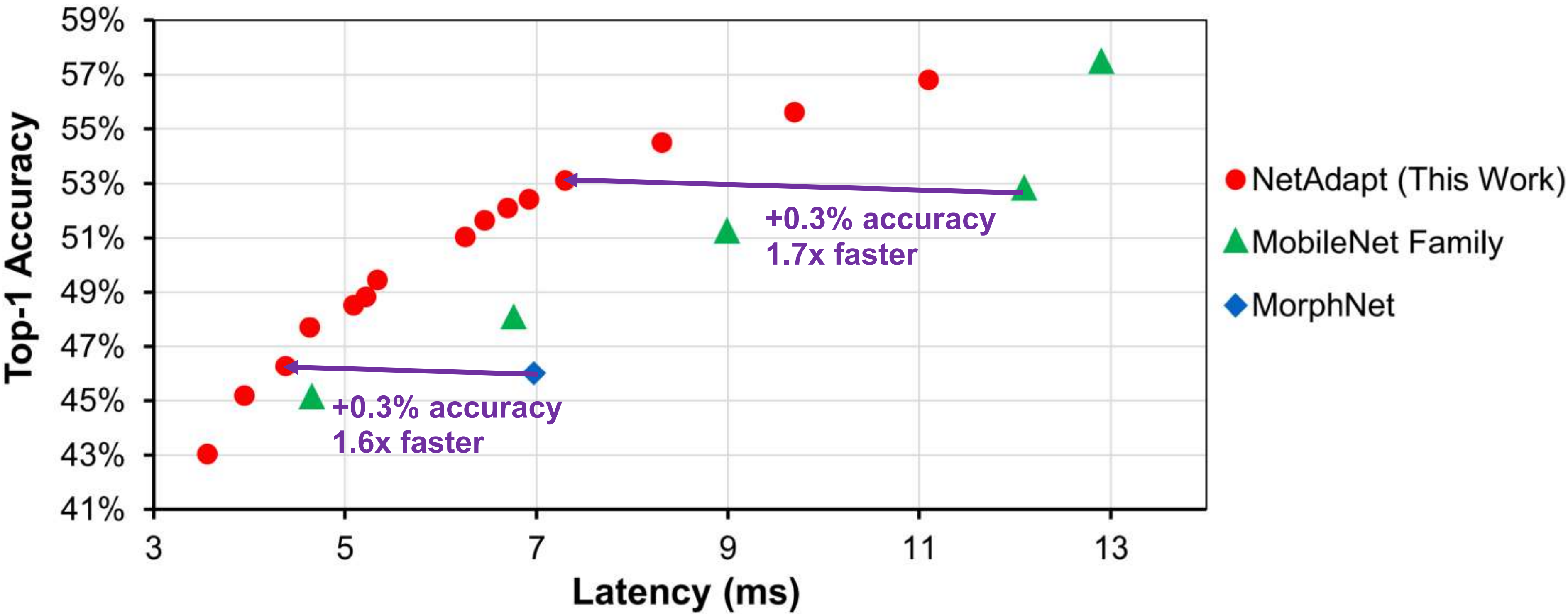
Empirical Measurements

Metric	Proposal A	…	Proposal Z
Latency	15.6	…	14.3
⋮	⋮		⋮
Energy	41	…	46

[Yang et al., ECCV 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

Reference:
MobileNet: Howard et al, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv 2017
MorphNet: Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks", CVPR 2018

Problem Formulation

$$\max_{Net} Accuracy(Net) \;\; subject\ to\ Resource_j(Net) \leq Budget_j, j = 1, \cdots, m$$

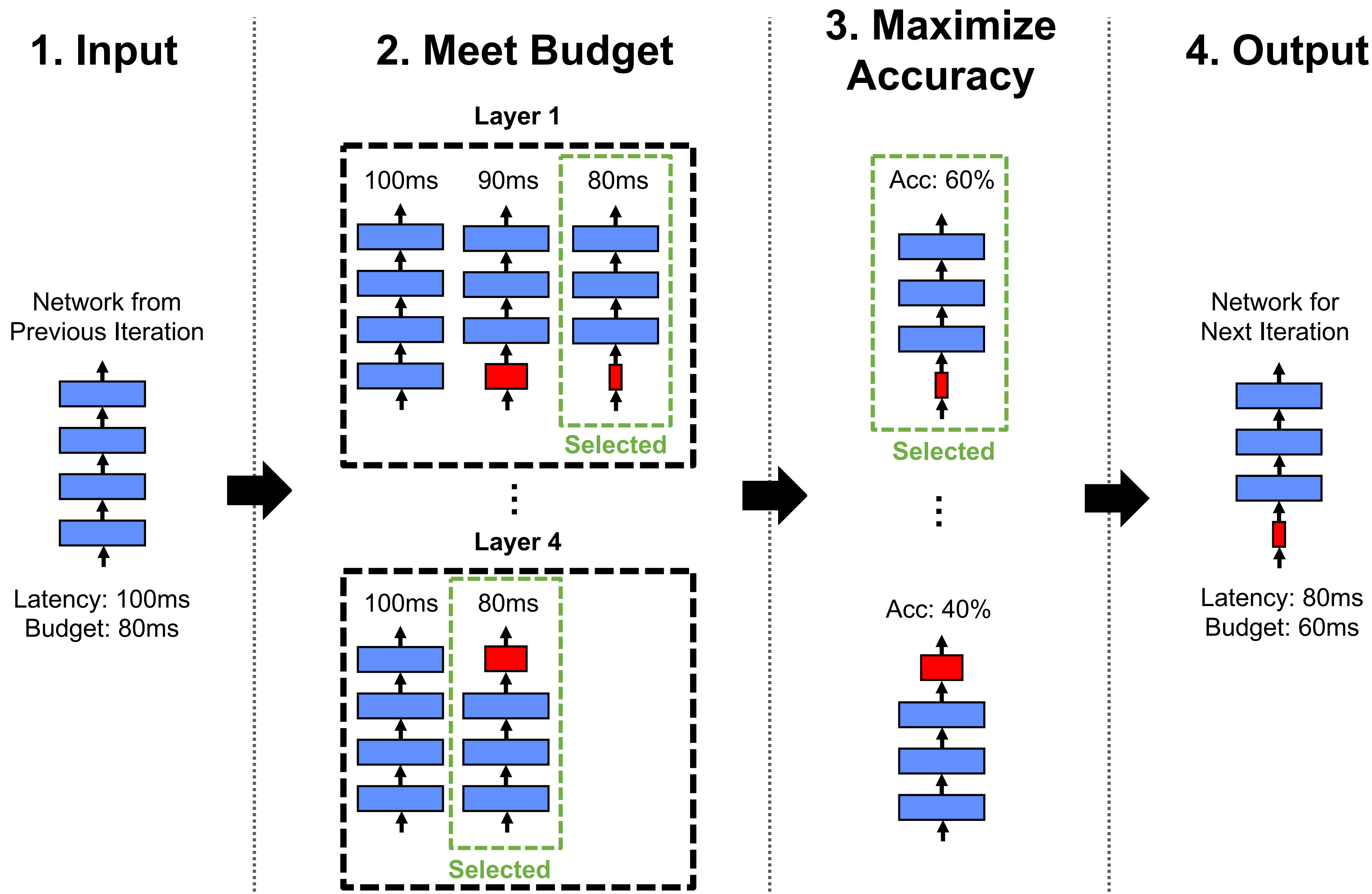
Break into a set of simpler problems and solve iteratively

$$\max_{Net_i} Acc(Net_i) \;\; subject\ to\ Res_j(Net_i) \leq Res_j(Net_{i-1}) - \Delta R_{i,j}, j = 1, \cdots, m$$

**Acc*: accuracy function, *Res*: resource evaluation function,
***ΔR*: resource reduction**, *Bud*: given budget
Budget incrementally tightens $\boldsymbol{Res_j(Net_{i-1}) - \Delta R_{i,j}}$

- **Advantages**
  - Supports multiple resource budgets at the same time
  - Guarantees that the budgets will be satisfied because the resource consumption decreases monotonically
  - Generates a family of networks (from each iteration) with different resource versus accuracy trade-offs
  - Intuitive and can easily set one additional hyperparameter ($\boldsymbol{\Delta R_{i,j}}$)

Simplified Example of One Iteration

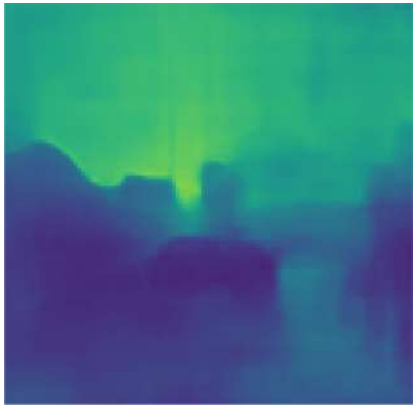


FastDepth: Fast Monocular Depth Estimation

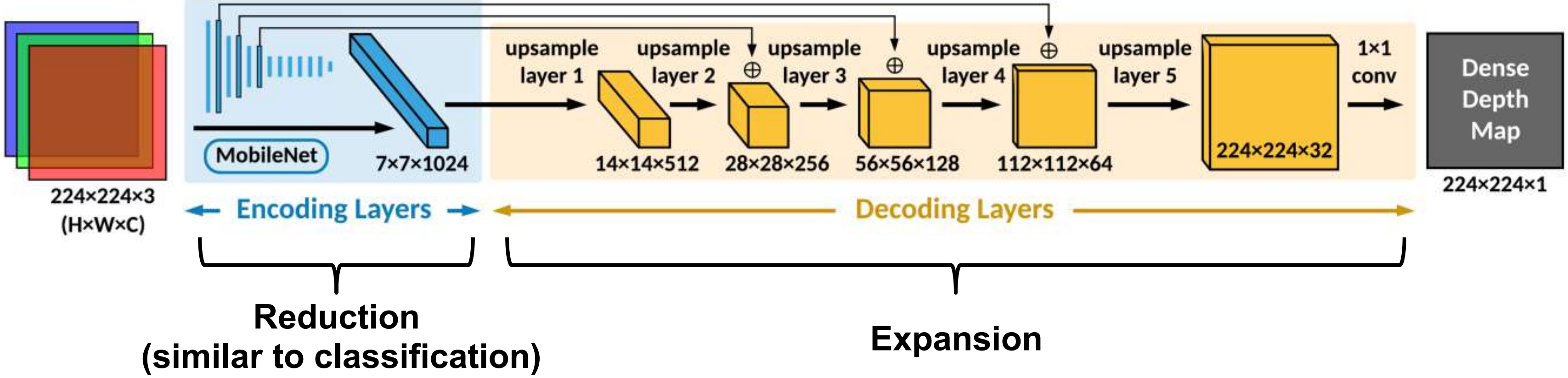
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction



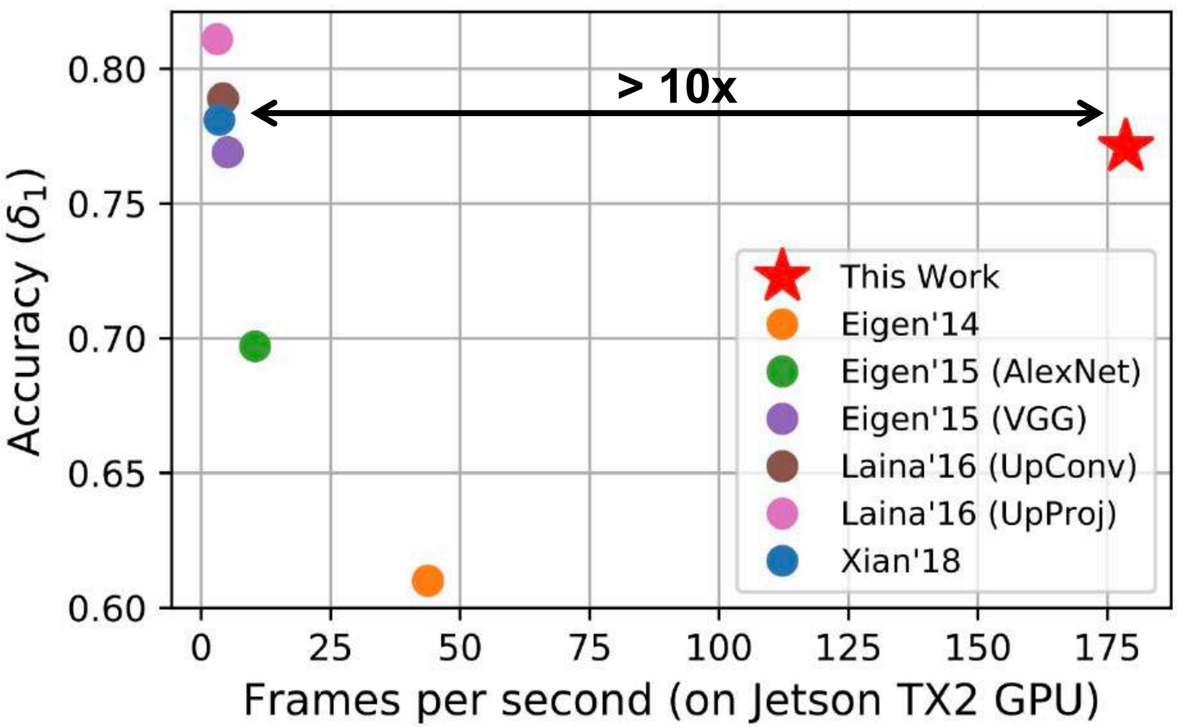
Auto Encoder DNN Architecture (Dense Output)



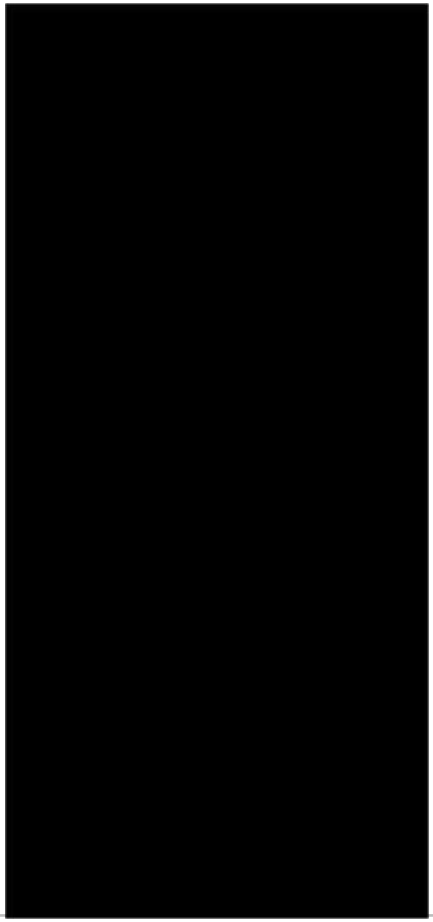
[Joint work with Sertac Karaman]

FastDepth: Fast Monocular Depth Estimation

Apply *NetAdapt*, *compact network design*, and *depth wise decomposition* to decoder layer to enable depth estimation at **high frame rates on an embedded platform** while still maintaining accuracy



Configuration: Batch size of one (32-bit float)



~40fps on an iPhone

Models available at *http://fastdepth.mit.edu*

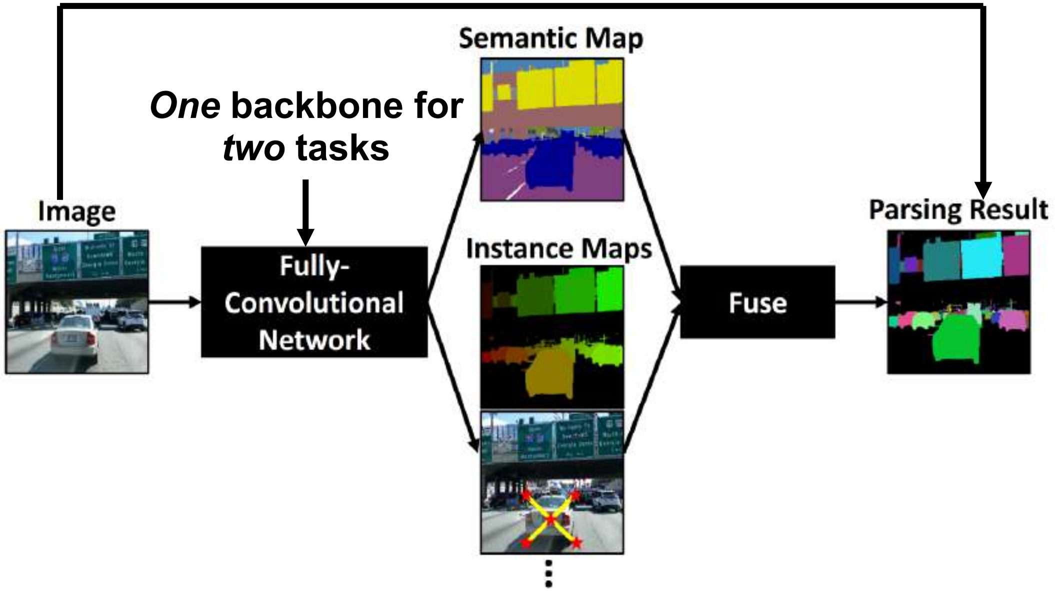
[Wofk*, Ma* et al., ICRA 2019]

DeeperLab: Single-Shot Image Parser

Results from Xception

Joint Semantic and Instance Segmentation (high resolution input image)

One-shot parsing for efficient processing



Fully convolutional, one-shot parsing
(bottom-up approach)

http://deeperlab.mit.edu/

[Yang et al., arXiv 2019]

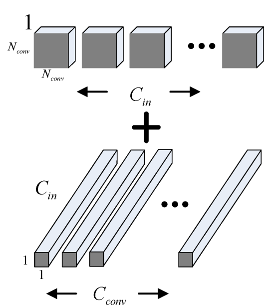
In collaboration with Google's Mobile Vision Team

DeeperLab: Efficient Image Parsing

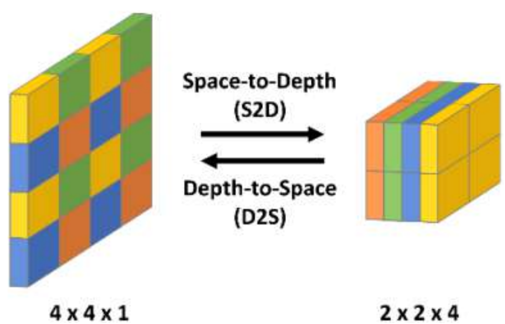
Address memory requirement for large feature map

1 Wide MobileNet: Increase kernel size rather than depth

3x3 → 5x5



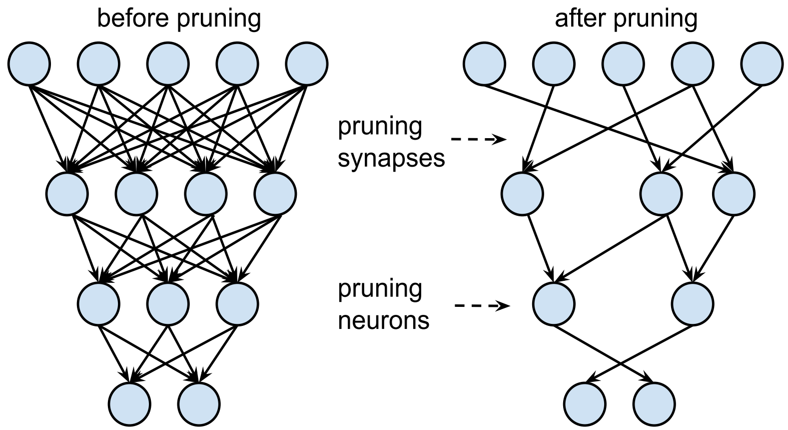
2 Space-to-depth/depth-to-space: Avoid upsampling



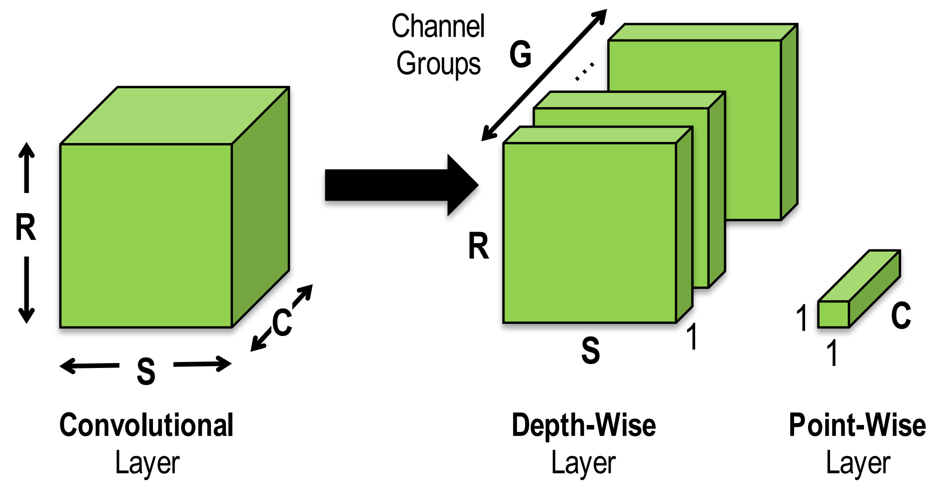
Achieves near real-time 6.19 fps on GPU (V100) with 25.2% PQ and 49.8% PC on Mapillary Vistas dataset

Many Efficient DNN Design Approaches

Network Pruning



Compact Network Architectures



Reduce Precision

32-bit float 10100101000000000101000000000100

8-bit fixed 01100110

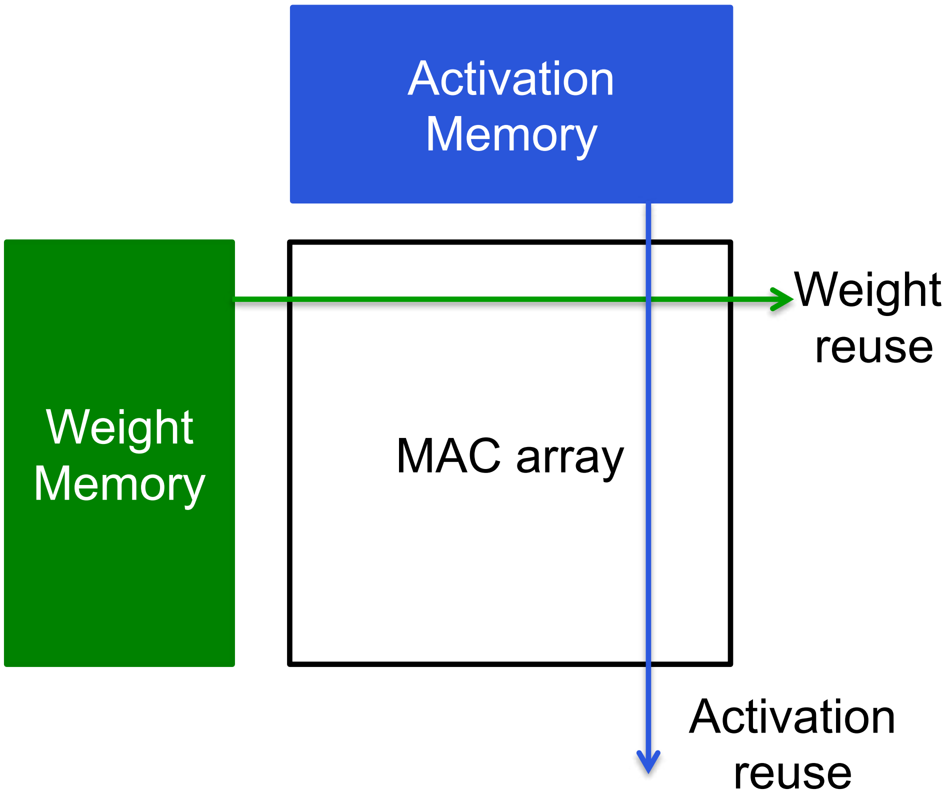
Binary 0

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

[Chen et al., SysML 2018]

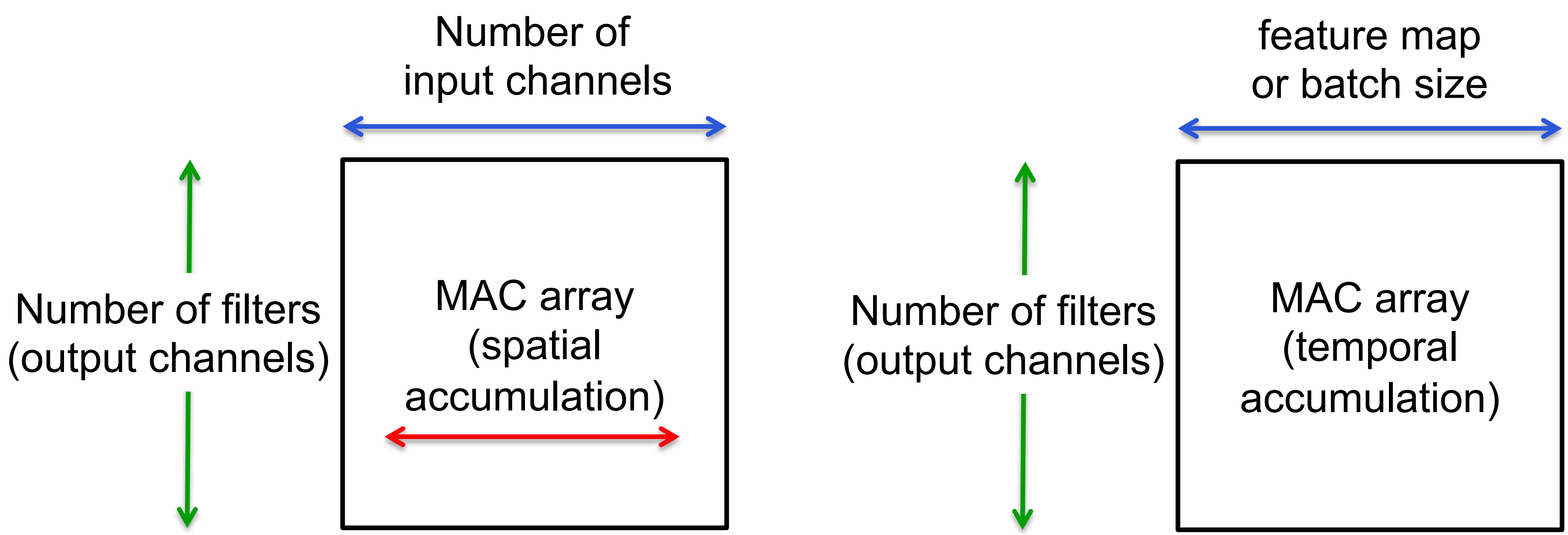
Existing DNN Architectures

- 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

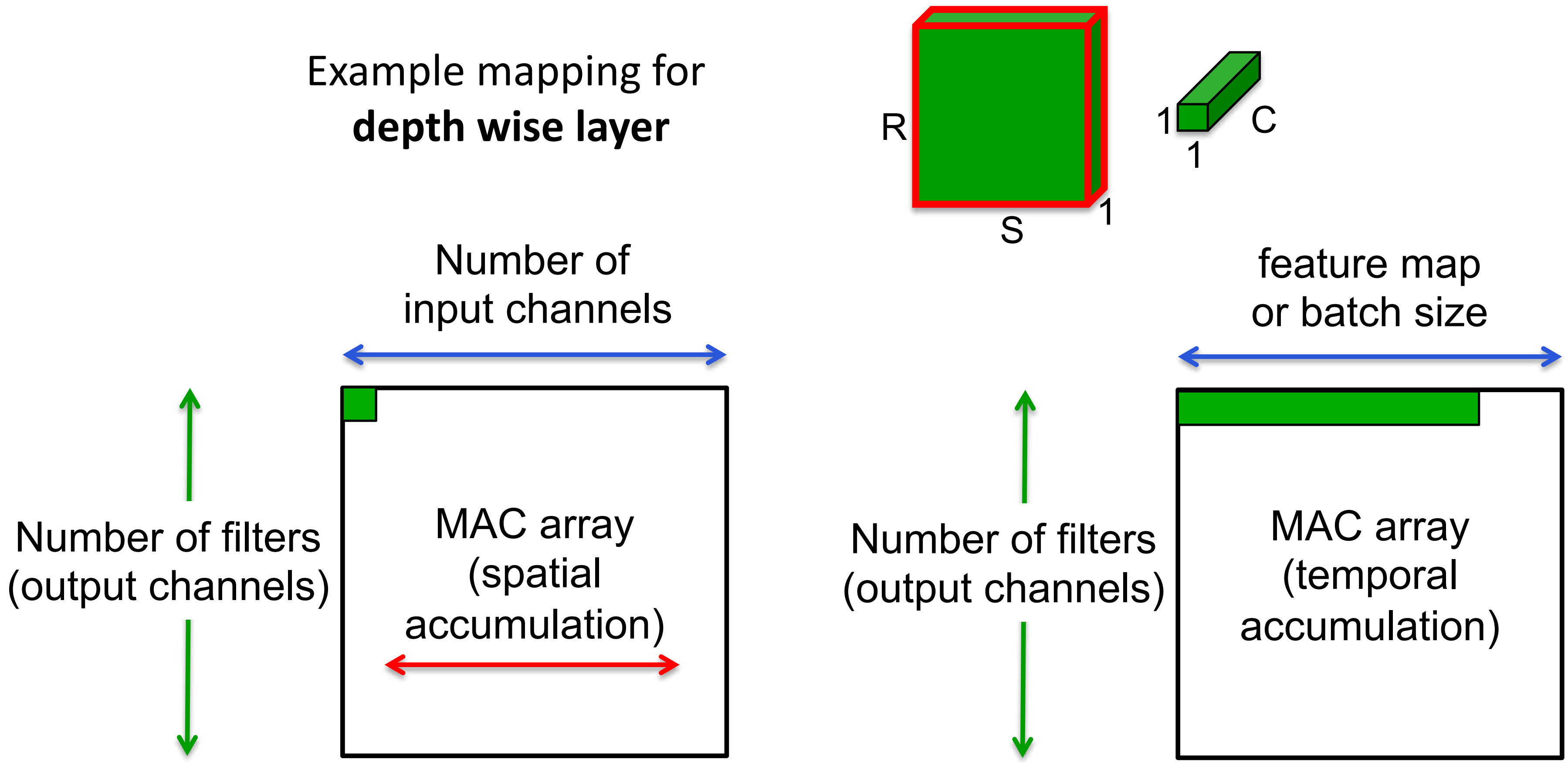
- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)



Limitation of Existing DNN Architectures

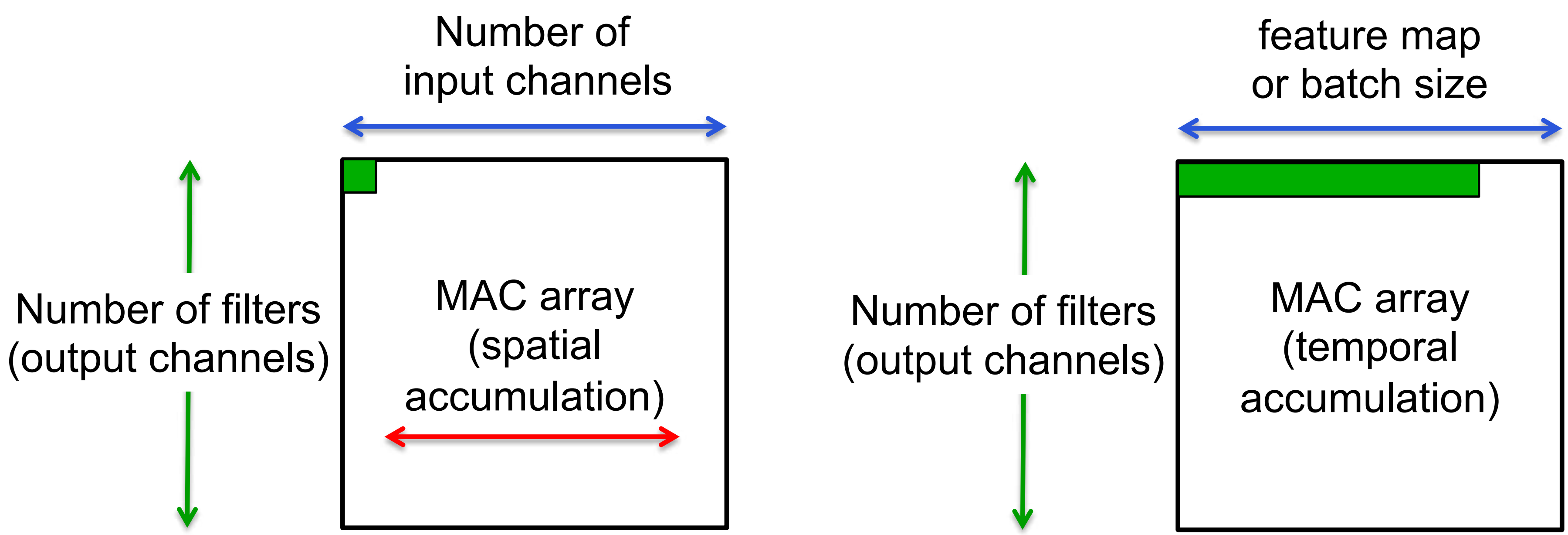
- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)

Example mapping for
depth wise layer



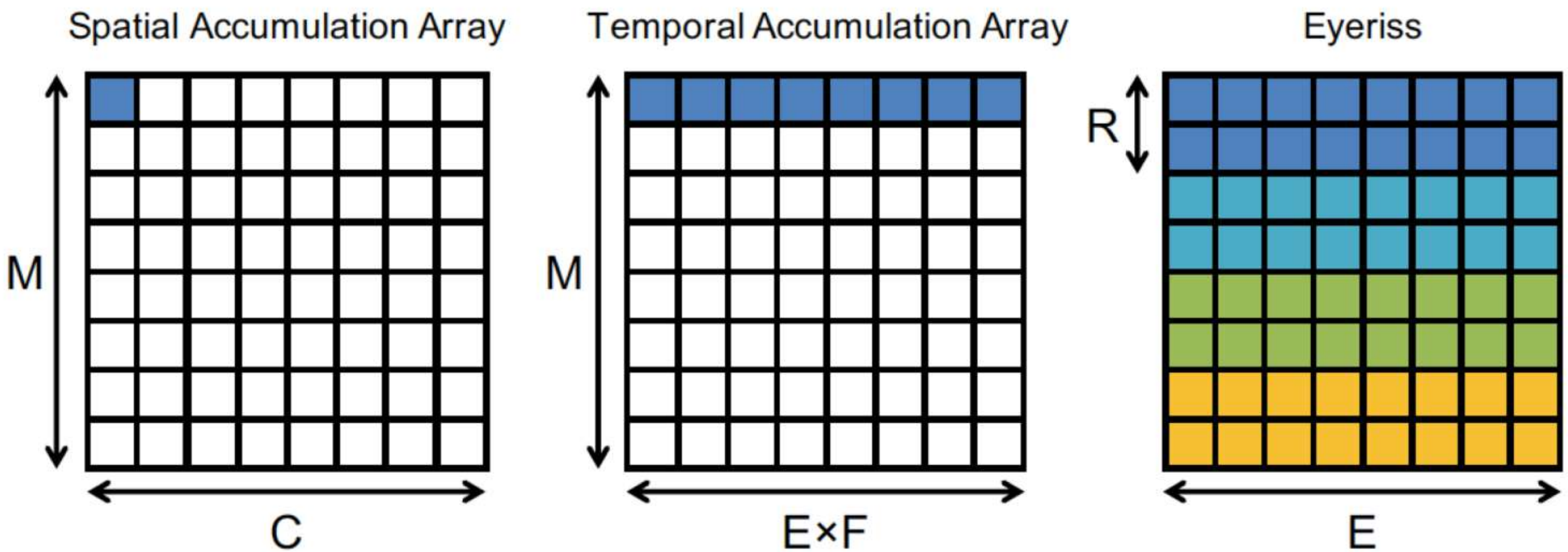
Limitation of Existing DNN Architectures

- **Example:** Reuse and array utilization depends on # of channels, feature map/batch size
  - Not efficient across all network architectures (e.g., compact DNNs)
  - Less efficient as array scales up in size
  - Can be challenging to exploit sparsity



Need Flexible Dataflow

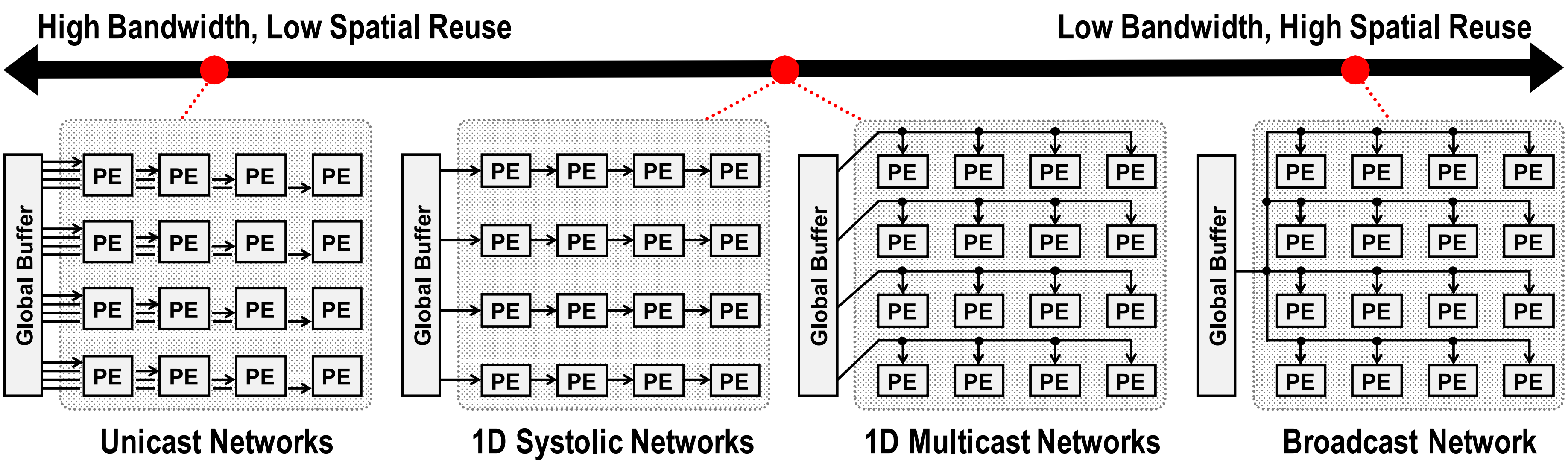
- Use flexible dataflow (**Row Stationary**) to exploit reuse in any dimension of DNN to increase energy efficiency and array utilization



Example: Depth-wise layer

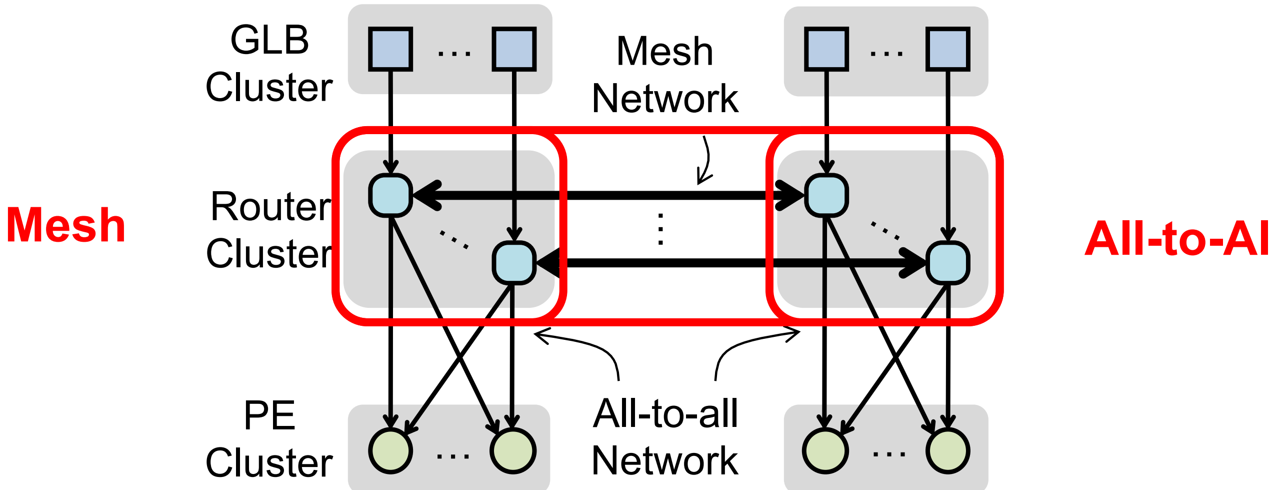
Need Flexible NoC for Varying Reuse

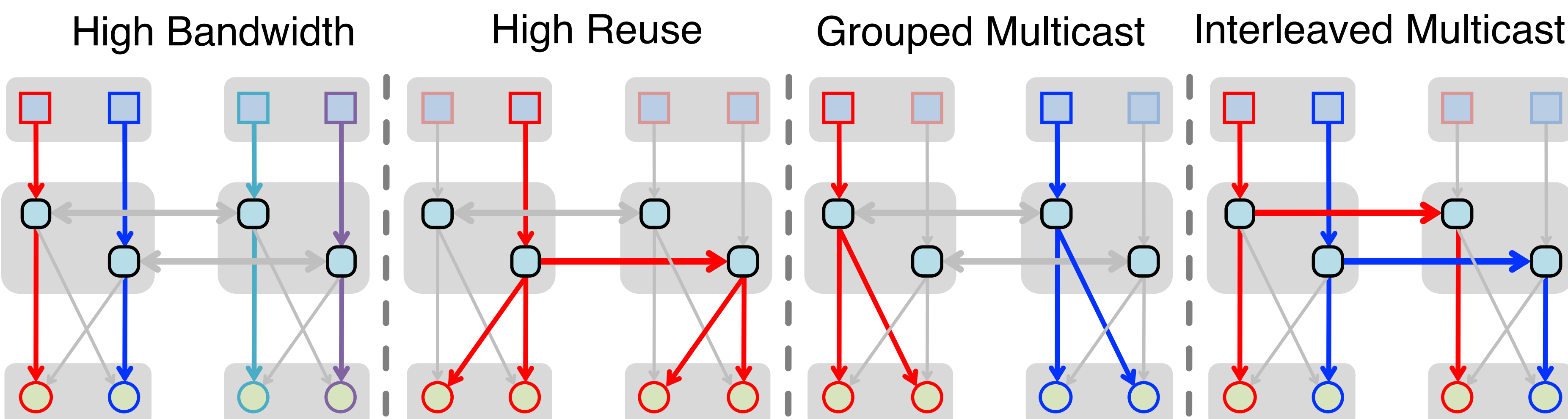
- When reuse available, need **multicast** to exploit spatial data reuse for energy efficiency and high array utilization
- When reuse not available, need **unicast** for high BW for weights for FC and weights & activations for high PE utilization
- An **all-to-all** satisfies above but too expensive and not scalable



[Chen et al., JETCAS 2019]

Hierarchical Mesh



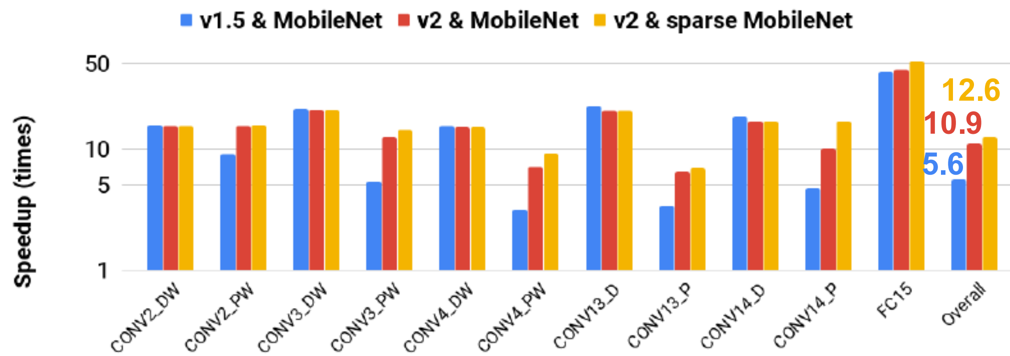


[Chen et al., JETCAS 2019]

Eyeriss v2: Balancing Flexibility and Efficiency

Efficiently supports

- Wide range of filter shapes
  - Large **and** Compact
- Different Layers
  - **CONV, FC, depth wise, etc.**
- Wide range of sparsity
  - Dense **and** Sparse
- **Scalable architecture**



Speed up over Eyeriss v1 scales with number of PEs

# of PEs	256	1024	16384
AlexNet	17.9x	71.5x	1086.7x
GoogLeNet	10.4x	37.8x	448.8x
MobileNet	15.7x	57.9x	873.0x

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

[Chen et al., JETCAS 2019]

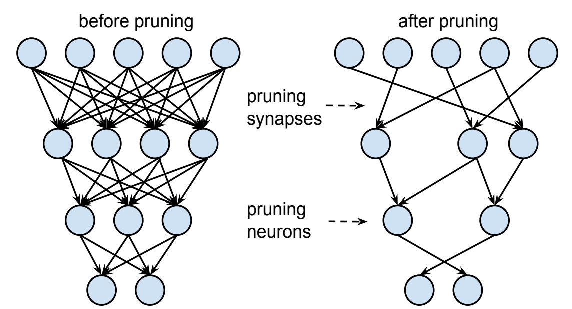
Need More Comprehensive Benchmarks

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

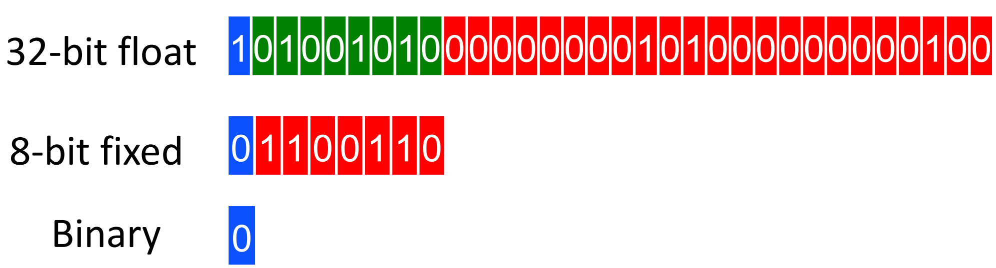
Example:

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

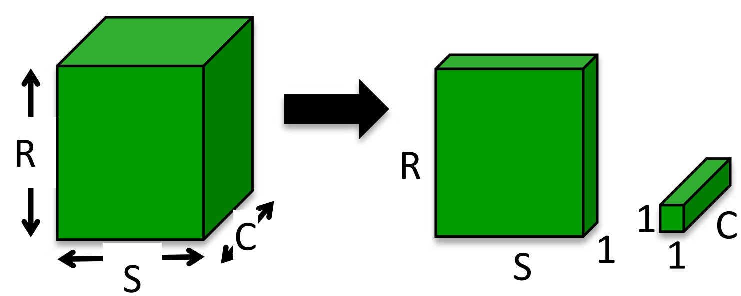
Network Pruning



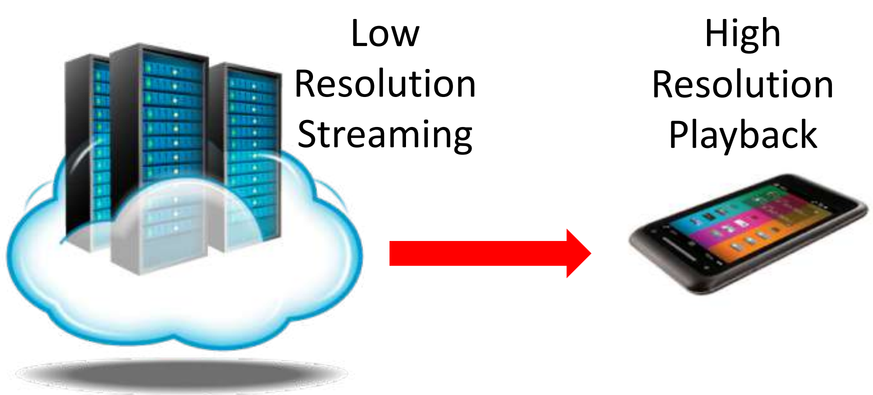
Reduce Precision



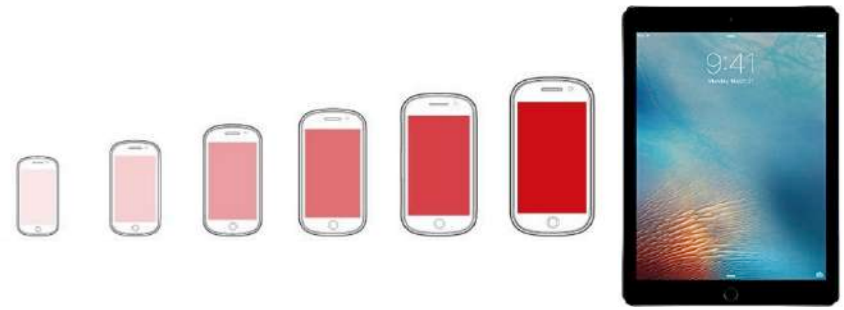
Compact Network Architecture



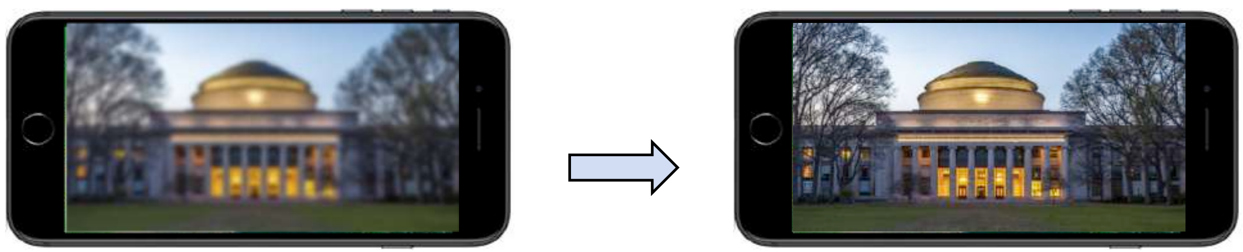
Super-Resolution on Mobile Devices



Transmit low resolution for lower bandwidth

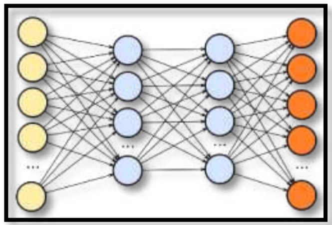


Screens are getting larger



Use **super-resolution** to improve the viewing experience of lower-resolution content (*reduce communication bandwidth*)

FAST: A Framework to Accelerate SuperRes



SR algorithm

Compressed video

FAST

SR
15x faster

Real-time

A framework that accelerates **any SR** algorithm by up to **15x** when running on compressed videos

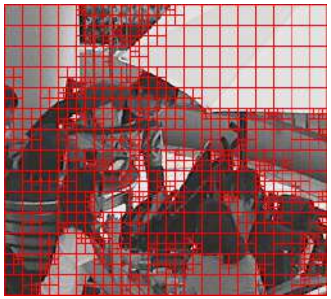
[Zhang et al., CVPRW 2017]

Free Information in Compressed Videos

Compressed video

Decode

Pixels



Block-structure

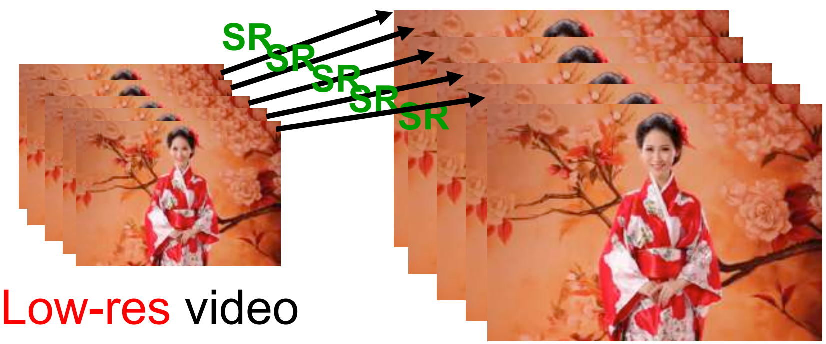
Motion-compensation

Video as a stack of pixels

Representation in compressed video

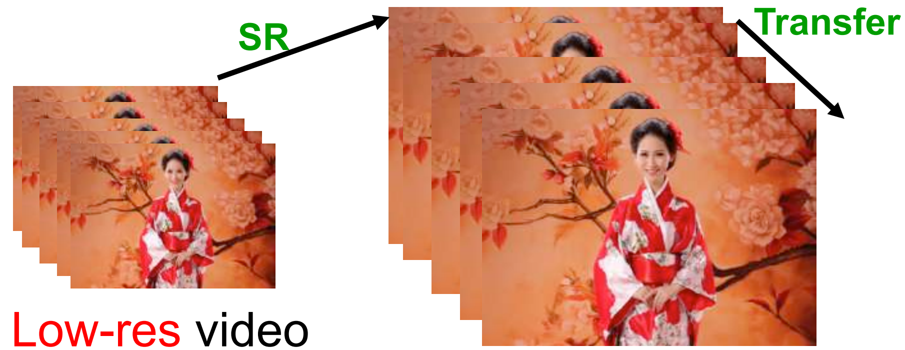
This representation can help **accelerate** super-resolution

Transfer is Lightweight



Low-res video

High-res video



Low-res video

High-res video

Transfer allows SR to run on only a subset of frames

Fractional Interpolation **Bicubic Interpolation**

Skip Flag

The complexity of the transfer is comparable to bicubic interpolation.
Transfer **N** frames, accelerate by **N**

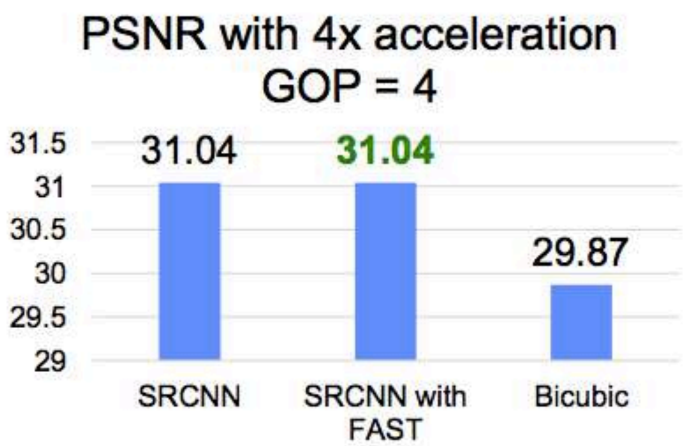
Evaluation: Accelerating SRCNN

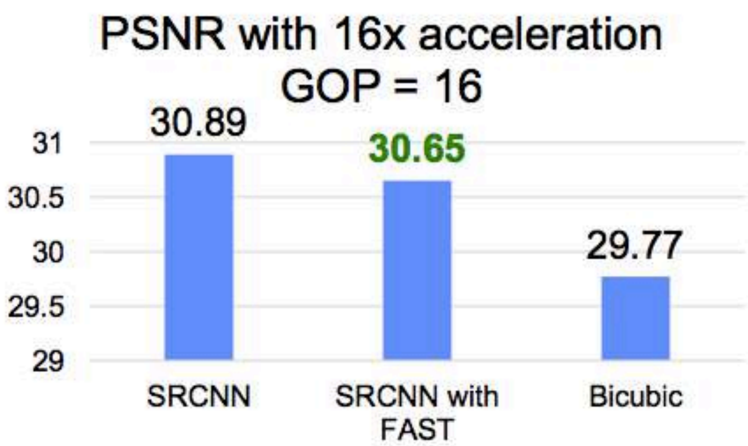
PartyScene

RaceHorse

BasketballPass

Examples of videos in the test set (20 videos for HEVC development)





$4\times$ acceleration with NO PSNR LOSS. $16\times$ acceleration with 0.2 dB loss of PSNR

Visual Evaluation

SRCNN **FAST + SRCNN** **Bicubic**

Look ***beyond*** the DNN accelerator for opportunities to accelerate DNN processing (e.g., structure of data and temporal correlation)

Code released at www.rle.mit.edu/eems/fast

[Zhang et al., CVPRW 2017]

Summary of Key Insights

- **Design considerations for co-design of algorithm and hardware**
  – Incorporate ***direct metrics*** into algorithm design for improved efficiency
  – Diverse workloads requires a ***flexible dataflow and NoC*** to exploit data ***reuse in any dimension*** and increase core utilization for speed and scalability
- **Accelerate deep learning by looking beyond the accelerator**
  – Exploit data representation for FAST Super-Resolution

Acknowledgements

Joel Emer

Sertac Karaman

Thomas Heldt

Research conducted in the **MIT Energy-Efficient Multimedia Systems Group** would not be possible without the support of the following organizations:

References

- **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 Conference, February 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.*
  - *Hardware Architecture for Deep Neural Networks: http://eyeriss.mit.edu/tutorial.html*

- **Co-Design of Algorithms and Hardware for Deep Neural Networks**
  - *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), 2017.*
  - *Energy estimation tool:* http://eyeriss.mit.edu/energy.html
  - *T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, V. Sze, H. Adam, "NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications," European Conference on Computer Vision (ECCV), 2018.*
  - *D. Wofk*, F. Ma*, T.-J. Yang, S. Karaman, V. Sze, "FastDepth: Fast Monocular Depth Estimation on Embedded Systems," IEEE International Conference on Robotics and Automation (ICRA), May 2019. http://fastdepth.mit.edu/*
  - *T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, L.-C. Chen, "DeeperLab: Single-Shot Image Parser," arXiv, February 2019.*

References

- **Energy-Efficient Hardware for Deep Neural Networks**
  - **Project website:** http://eyeriss.mit.edu
  - *Y.-H. Chen, T. Krishna, J. Emer, V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," IEEE Journal of Solid State Circuits (JSSC), ISSCC Special Issue, Vol. 52, No. 1, pp. 127-138, January 2017.*
  - *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.*
  - *Y.-H. Chen, T.-J. Yang, J. Emer, V. Sze, "Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices," IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), June 2019.*
  - *Eyexam: https://arxiv.org/abs/1807.07928*

- ***Looking beyond DNN Accelerator for Acceleration***
  - *Z. Zhang, V. Sze, "FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos," CVPR Workshop on New Trends in Image Restoration and Enhancement, July 2017. www.rle.mit.edu/eems/fast*