Efficient Computing for AI and Robotics

Vivienne Sze (@eems_mit)

Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Sertac Karaman, Tushar Krishna, Thomas Heldt, Trevor Henderson, Hsin-Yu Lai, Peter Li, Fangchang Ma, James Noraky, Gladynel Saavedra Peña, Charlie Sodini, Amr Suleiman, Nellie Wu, Diana Wofk, Tien-Ju Yang, Zhengdong Zhang

Slides available at
https://tinyurl.com/SzeMITDL2020

Processing at “Edge” instead of the “Cloud”

Communication

Privacy

Latency

Computing Challenge for Self-Driving Cars

JACK STEWART TRANSPORTATION 02.06.18 08:00 AM

SELF-DRIVING CARS USE CRAZY AMOUNTS OF POWER, AND IT'S BECOMING A PROBLEM

Shelley, a self-driving Audi TT developed by Stanford University, uses the brains in the trunk to speed around a racetrack autonomously.

NIKKI KAHN/THE WASHINGTON POST/GETTY IMAGES

WIRED

(Feb 2018)

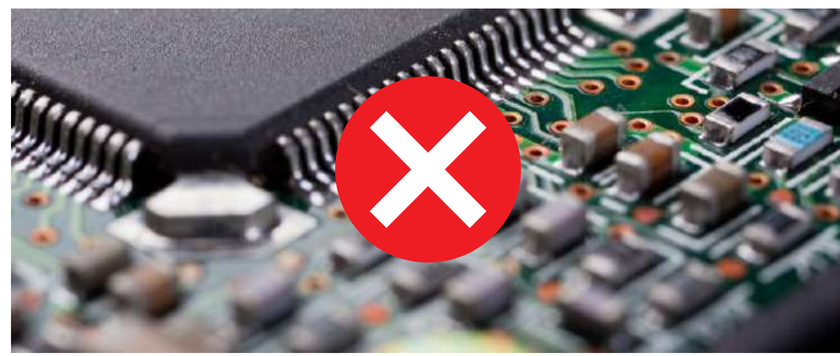
Cameras and radar generate ~6 gigabytes of data every 30 seconds.

Self-driving car prototypes use approximately 2,500 Watts of computing power.

Generates wasted heat and some prototypes need water-cooling!

Existing Processors Consume Too Much Power

< 1 Watt



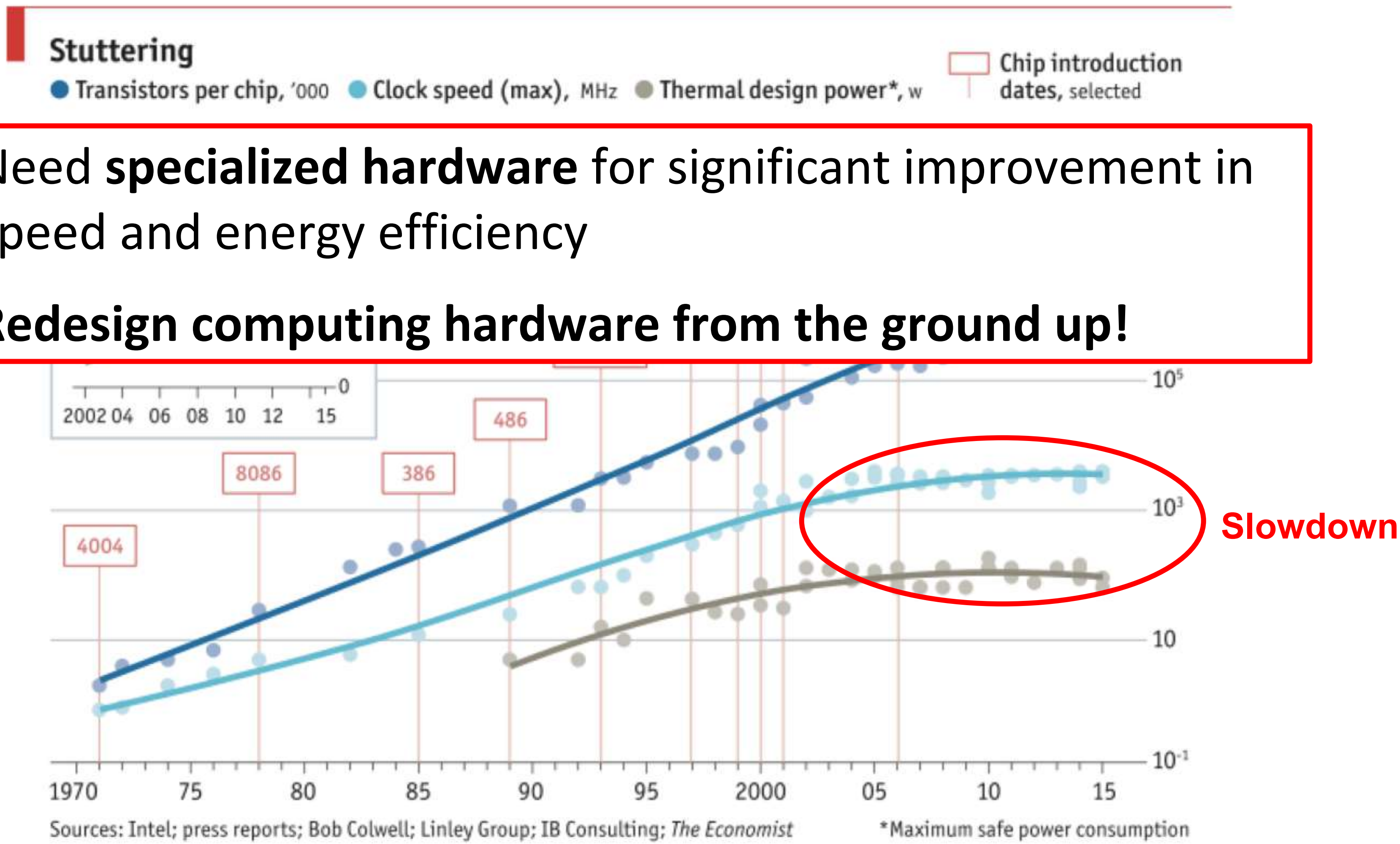
> 10 Watts

Transistors are NOT Getting More Efficient

Slow down of Moore's Law and Dennard Scaling

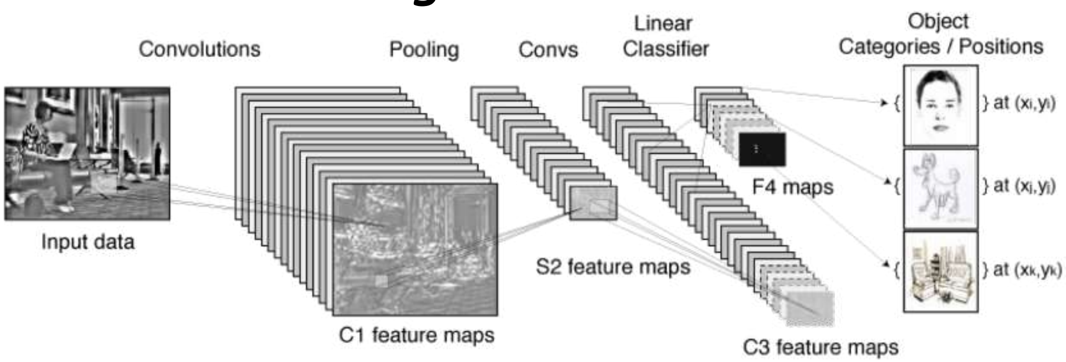
General purpose microprocessors not getting faster or more efficient

- Need **specialized hardware** for significant improvement in speed and energy efficiency
- **Redesign computing hardware from the ground up!**



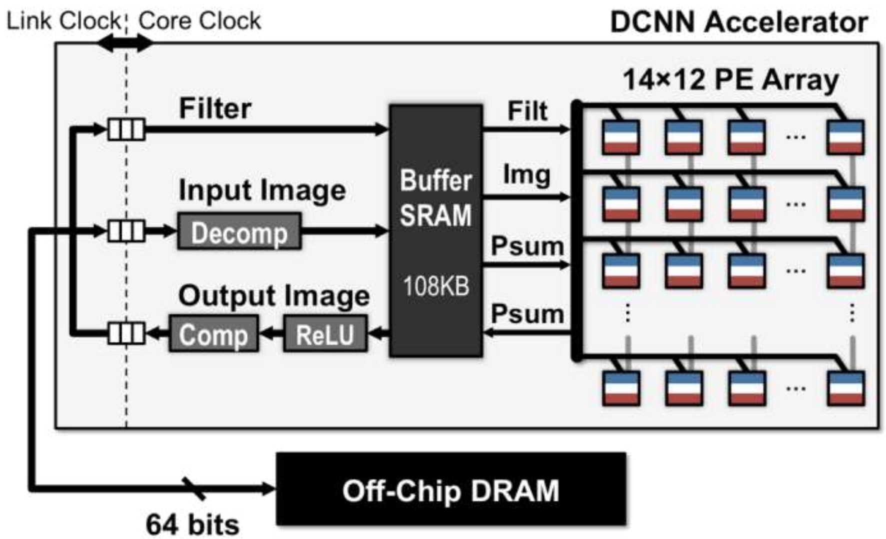
Energy-Efficient Computing with Cross-Layer Design

Algorithms

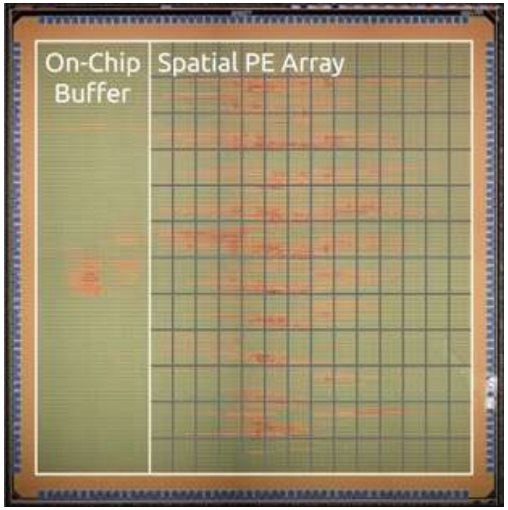


Systems

Architectures

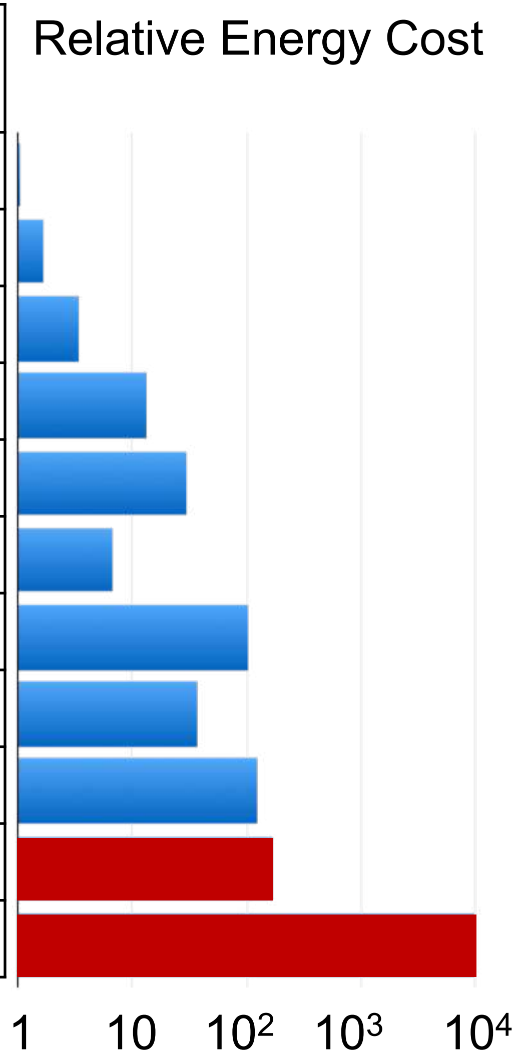


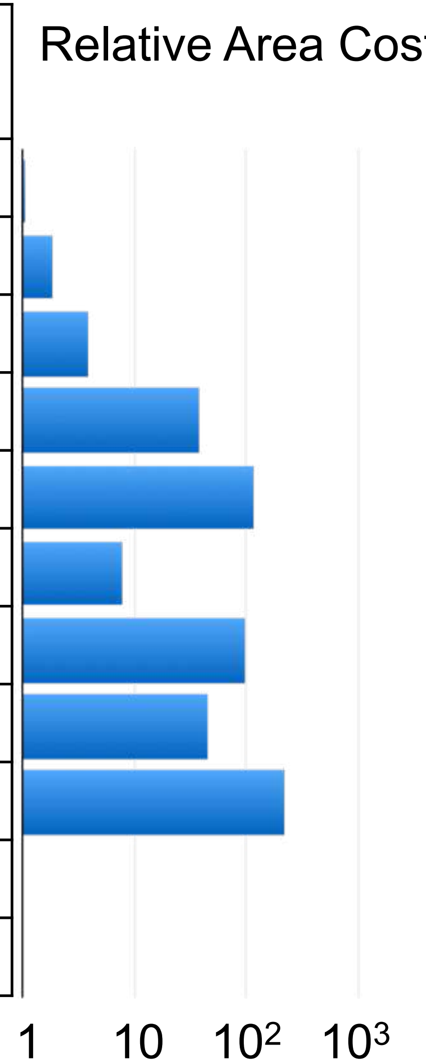
Circuits



Power Dominated by Data Movement

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A





Memory access is **orders of magnitude** higher energy than compute

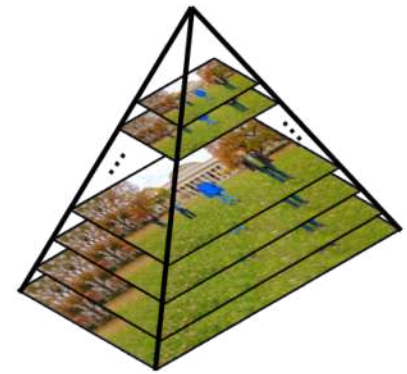
[**Horowitz**, *ISSCC* 2014]

Autonomous Navigation Uses a Lot of Data

- **Semantic Understanding**

- High frame rate
- Large resolutions
- Data expansion

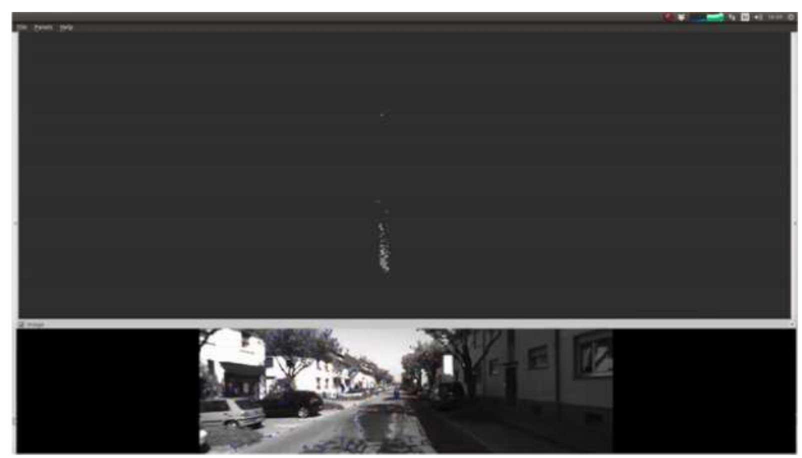
2 million pixels



10x-100x more pixels

- **Geometric Understanding**

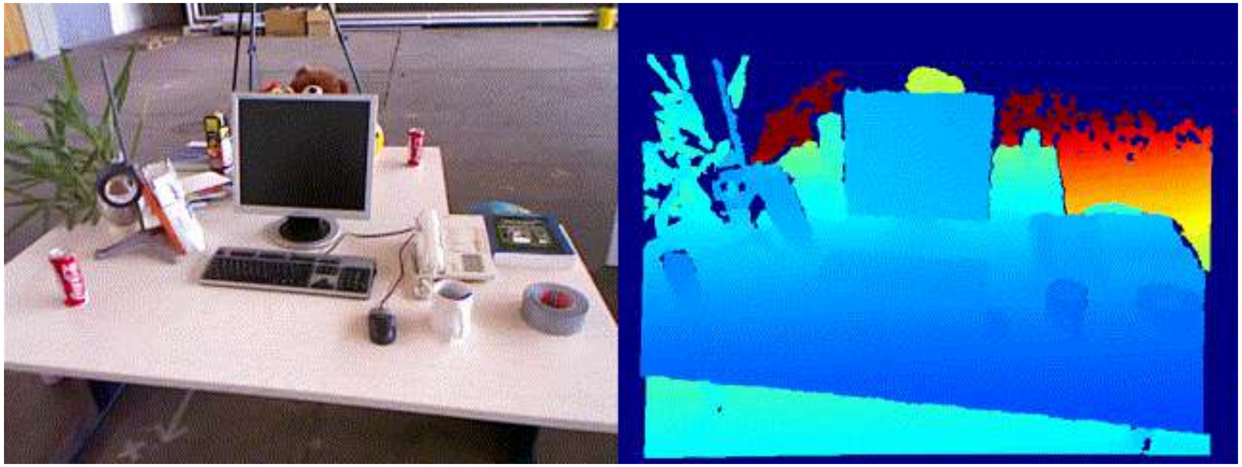
- Growing map size

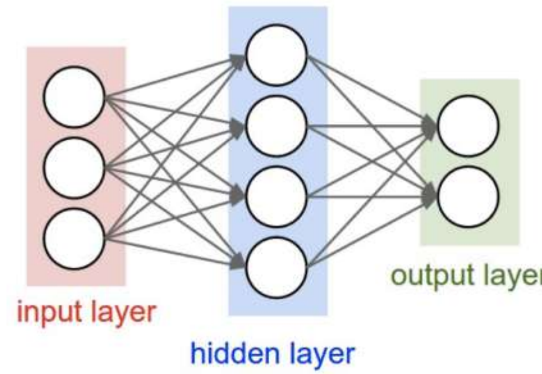


[**Pire**, *RAS* 2017]

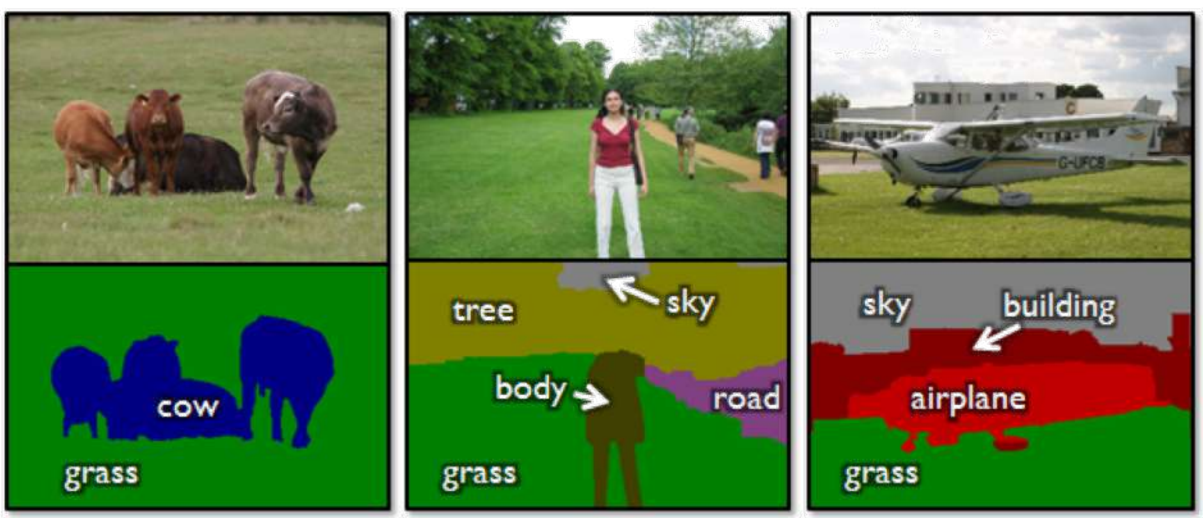
Understanding the Environment

Depth Estimation





Semantic Segmentation

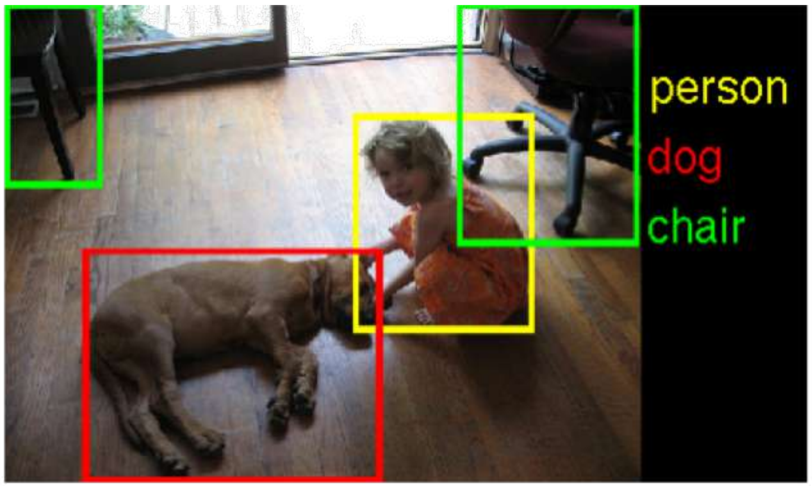


State-of-the-art approaches use **Deep Neural Networks,** which require **up to several hundred millions of operations and weights to compute!**
>100x more complex than video compression

Deep Neural Networks

*Deep Neural Networks (DNNs) have become a **cornerstone of AI***

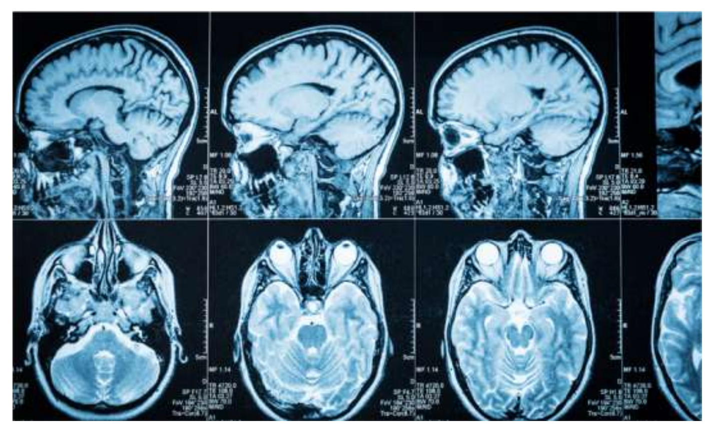
Computer Vision



Speech Recognition

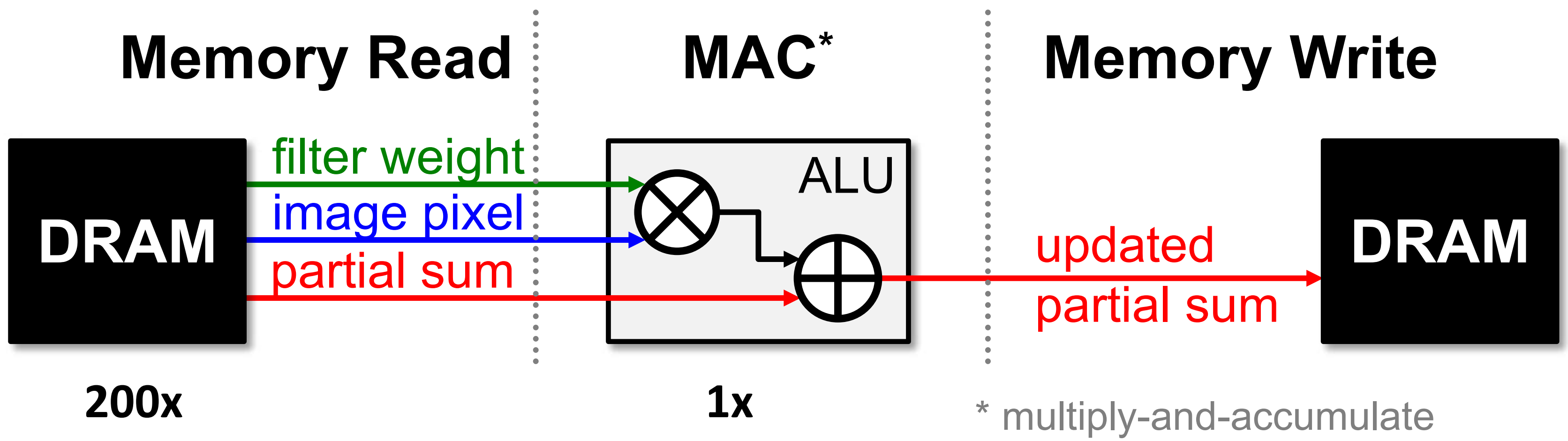
Game Play

Medical



Properties We Can Leverage

- Operations exhibit **high parallelism**
  → **high throughput** possible
- Memory Access is the Bottleneck

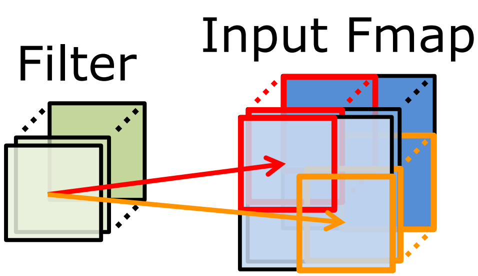


<u>Worst Case</u>: all memory R/W are **DRAM** accesses

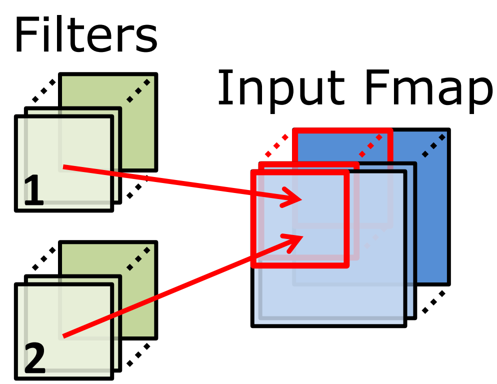
- Example: AlexNet has **724M** MACs
  → **2896M** DRAM accesses required

Properties We Can Leverage

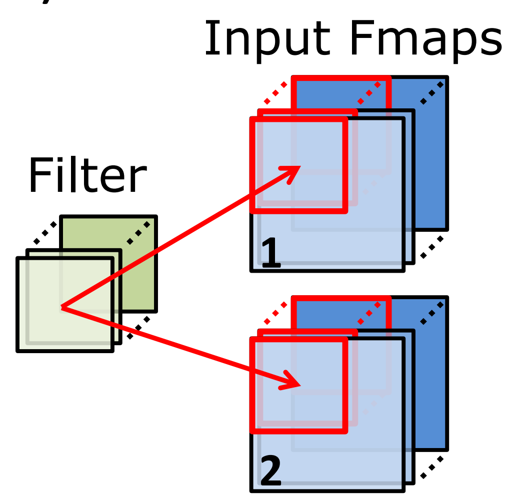
- Operations exhibit **high parallelism**
  → **high throughput** possible
- **Input data reuse** opportunities (**up to 500x**)



Convolutional Reuse
(Activations, Weights)
CONV layers only
(sliding window)

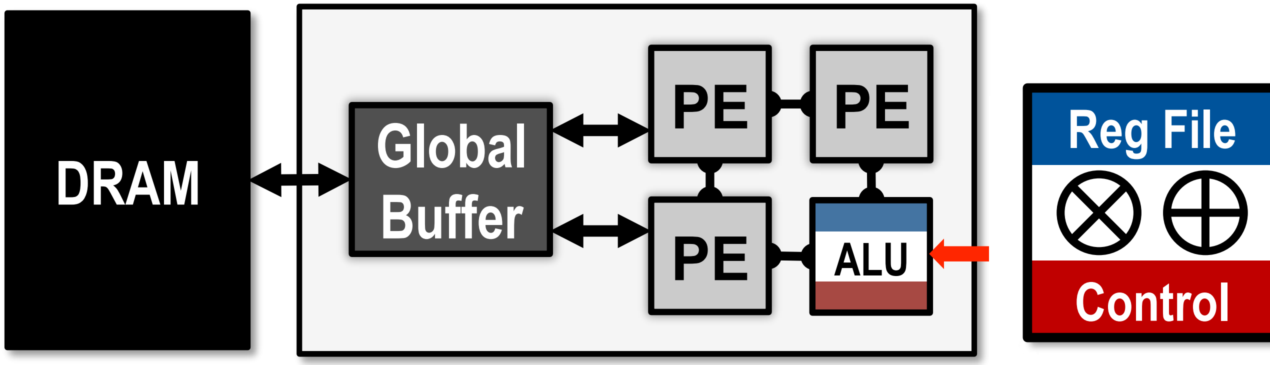


Fmap Reuse
(Activations)
CONV and FC layers

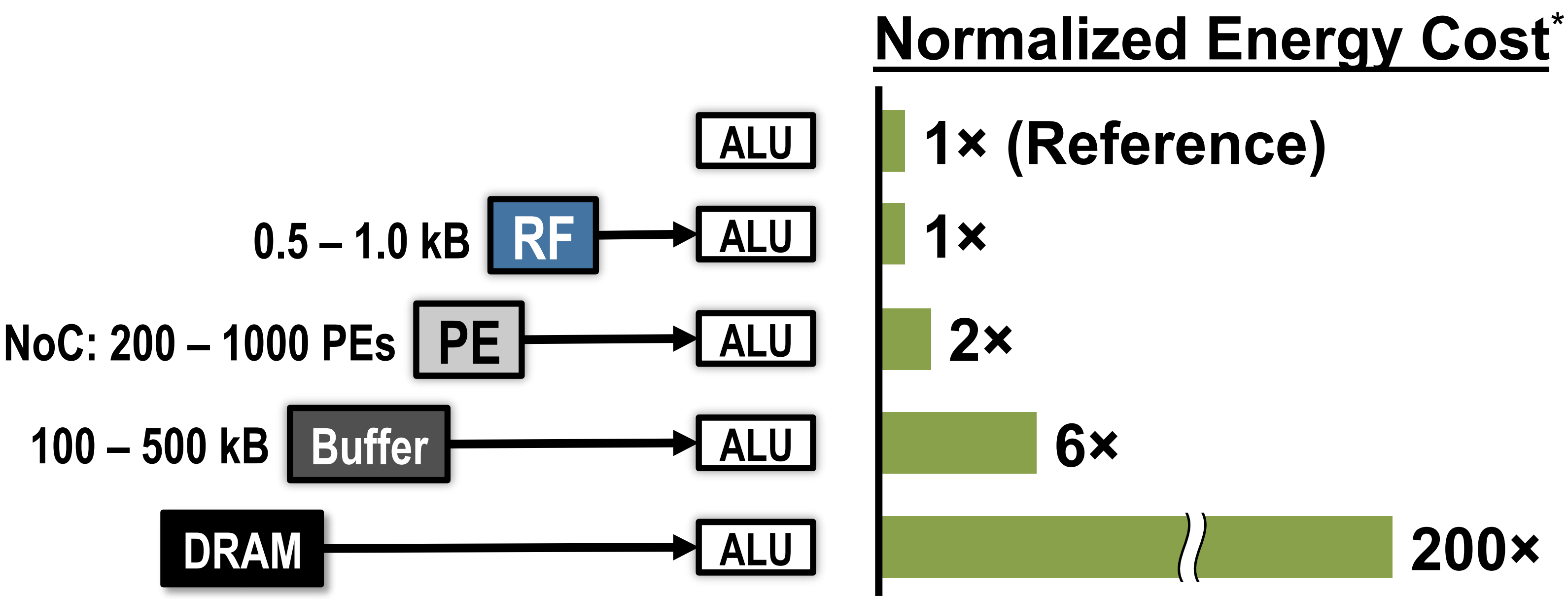


Filter Reuse
(Weights)
CONV and FC layers
(batch size > 1)

Exploit Data Reuse at Low-Cost Memories



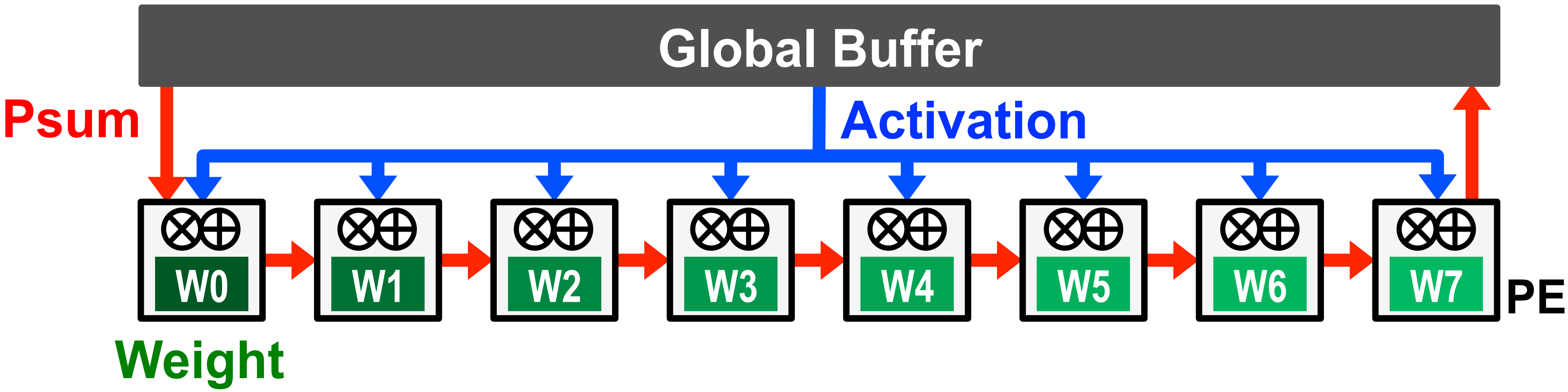
Specialized hardware with small (< 1kB) low cost memory near compute



* measured from a commercial 65nm process

Farther and **larger** memories consume more power

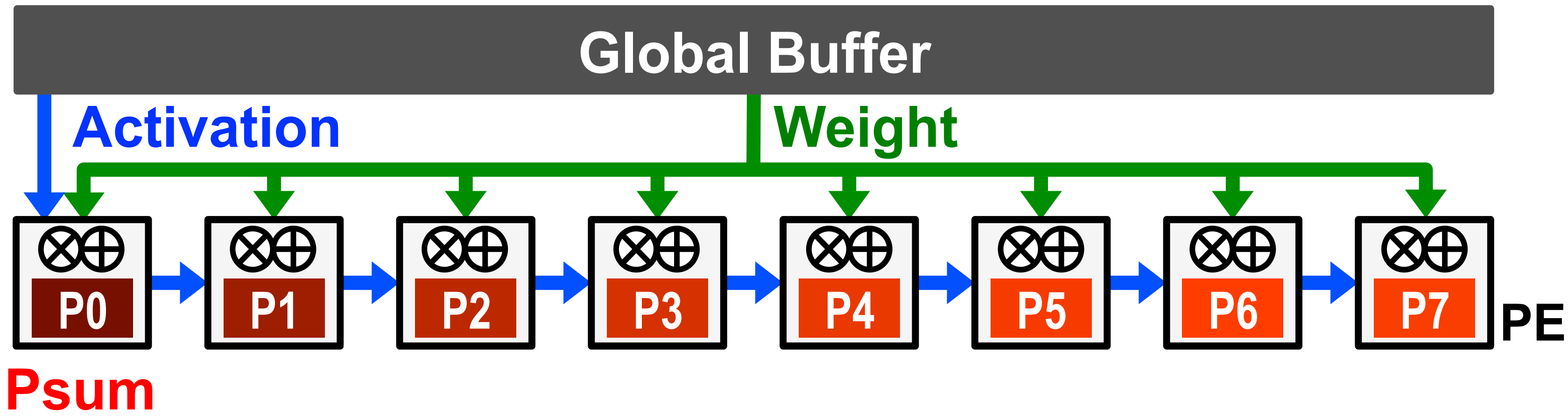
Weight Stationary (WS)



- **Minimize** **weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Broadcast** **activations** and **accumulate** **partial sums** **spatially** across the PE array
- Examples: **TPU [Jouppi**, *ISCA* 2017], **NVDLA**

 [**Chen**, *ISCA* 2016]

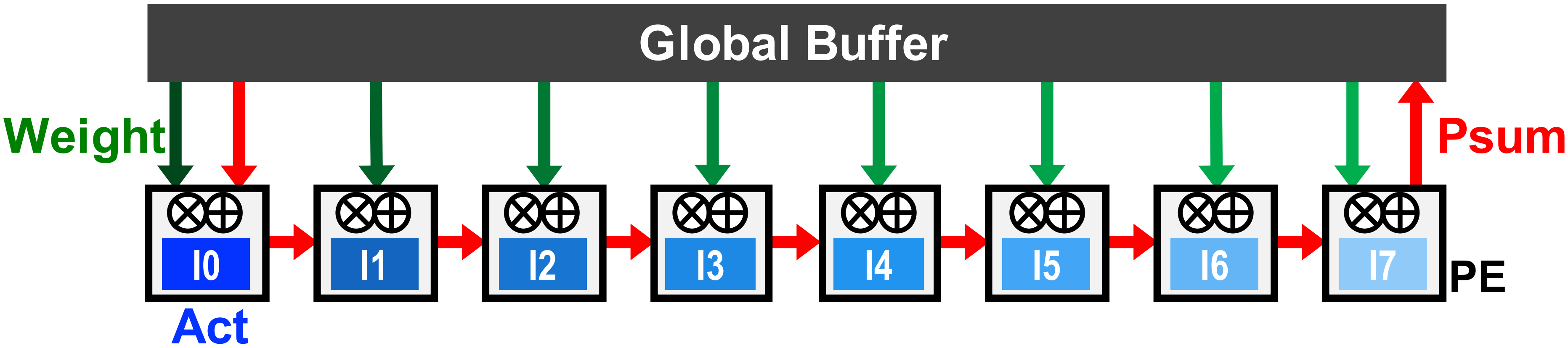
Output Stationary (OS)



- **Minimize partial sum** R/W energy consumption
  - maximize local accumulation
- **Broadcast/Multicast filter weights** and **reuse activations spatially** across the PE array
- Examples: [**Moons**, *VLSI* 2016], [**Thinker**, *VLSI* 2017]

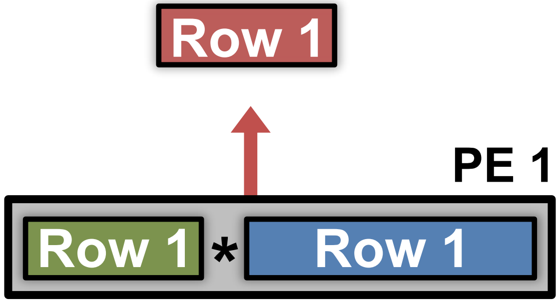
[**Chen**, *ISCA* 2016]

Input Stationary (IS)

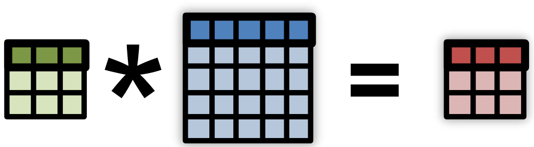


- **Minimize** **activation** read energy consumption
  - maximize convolutional and fmap reuse of activations
- **Unicast** **weights** and **accumulate** **partial sums** **spatially** across the PE array
- Example: [**SCNN**, *ISCA* 2017]

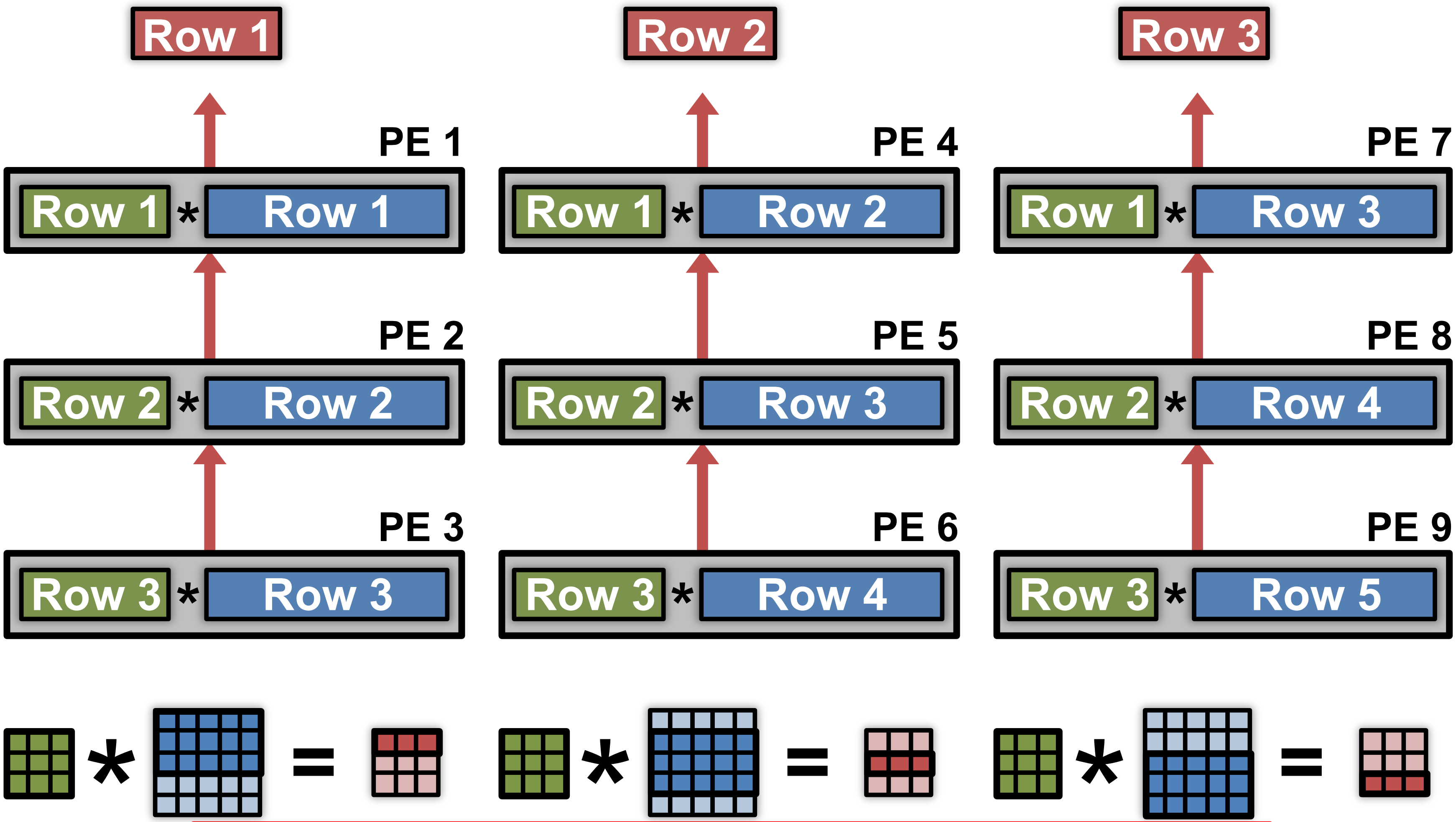
Row Stationary Dataflow



- Maximize row **convolutional reuse** in RF
  - Keep a **filter** row and **fmap** sliding window in RF
- Maximize row **psum accumulation** in RF

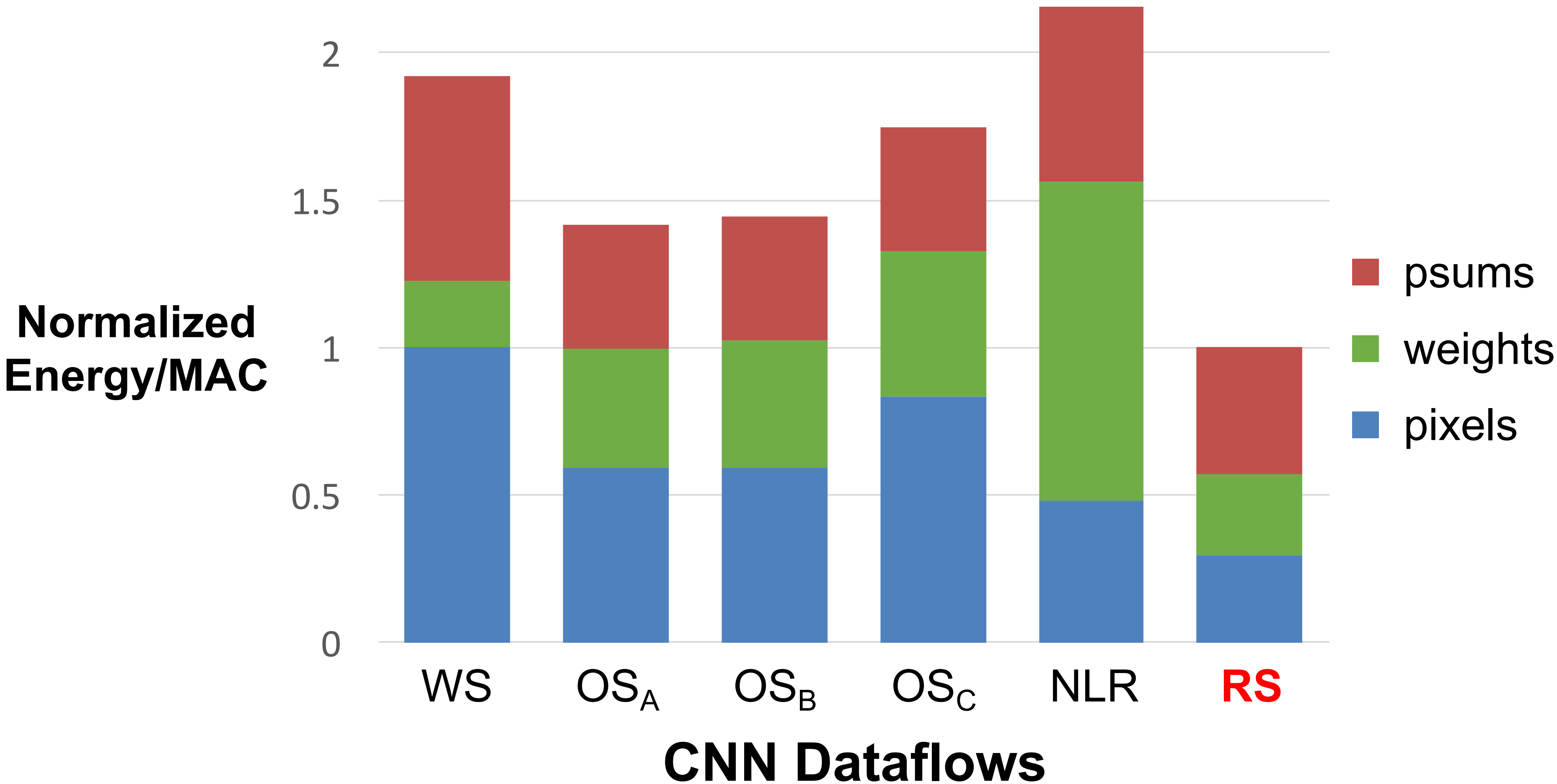


Row Stationary Dataflow



Optimize for **overall energy efficiency** instead for only a certain data type

Dataflow Comparison: CONV Layers

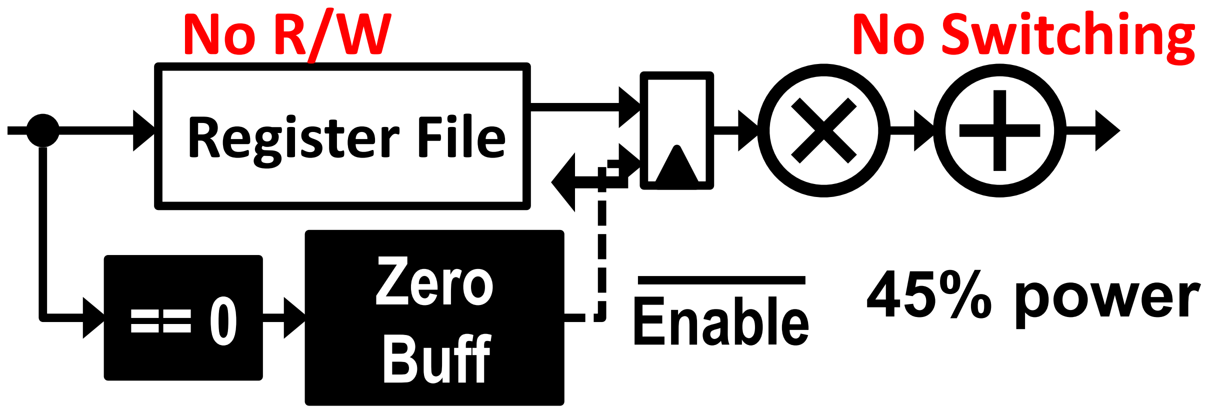


RS optimizes for the best **overall** energy efficiency

[**Chen**, *ISCA* 2016]

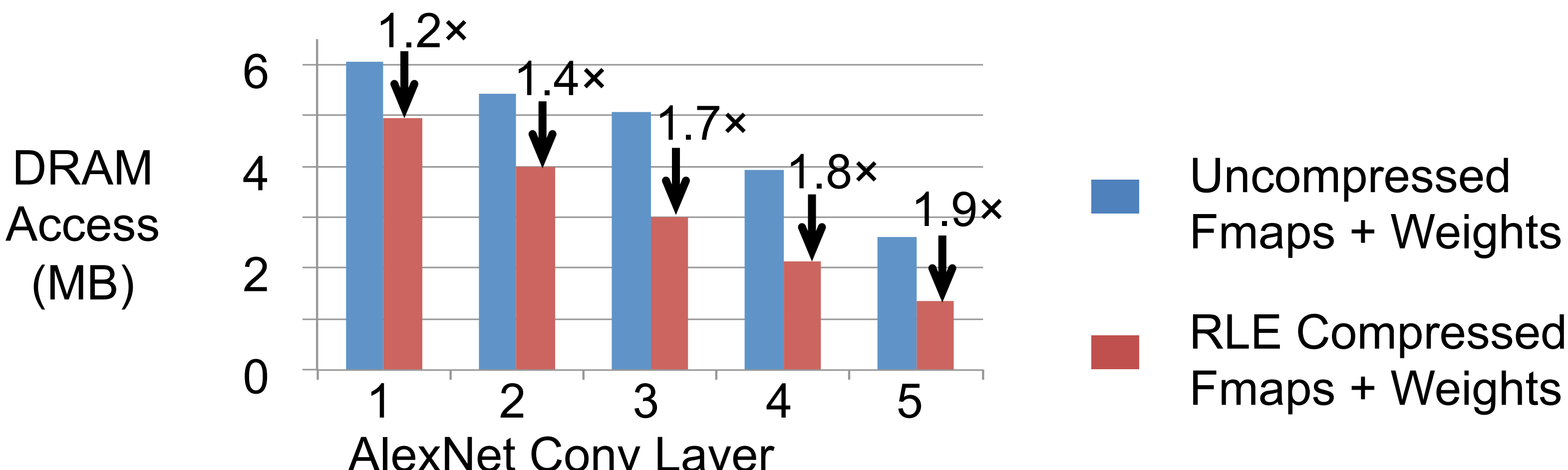
Exploit Sparsity

Method 1. Skip memory access and computation



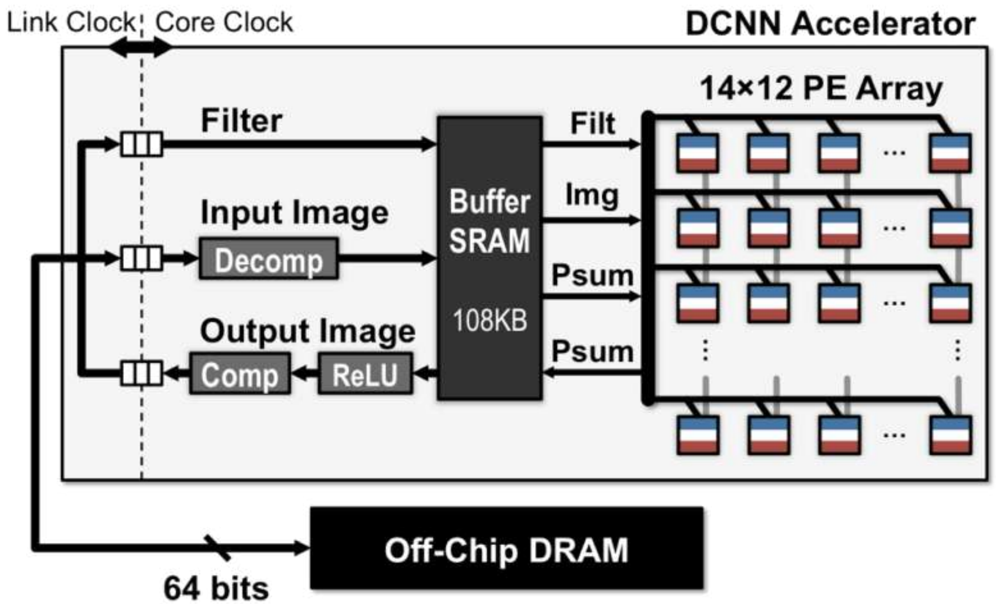
45% power reduction

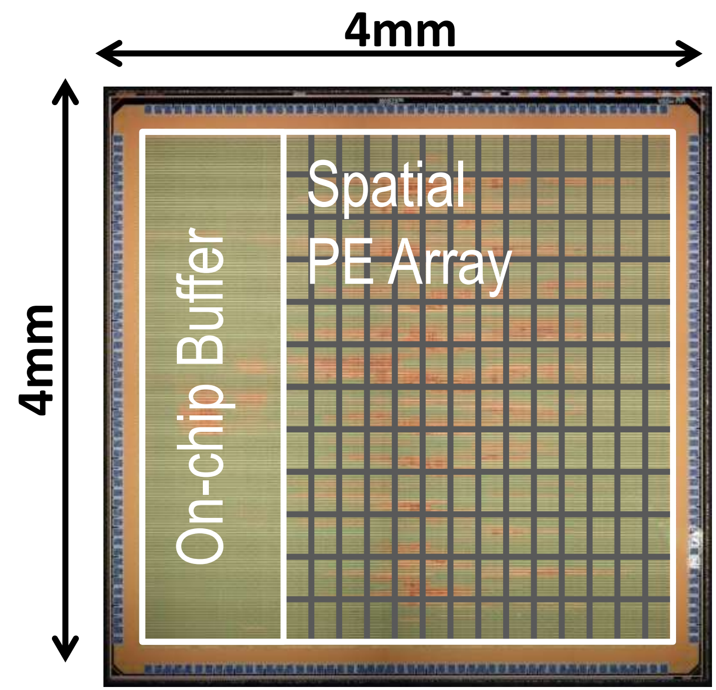
Method 2. Compress data to reduce storage and data movement



[**Chen**, *ISSCC* 2016]

Eyeriss: Deep Neural Network Accelerator





[**Chen**, *ISSCC* 2016]

Exploits data reuse for **100x** reduction in memory accesses from global buffer and **1400x** reduction in memory accesses from off-chip DRAM

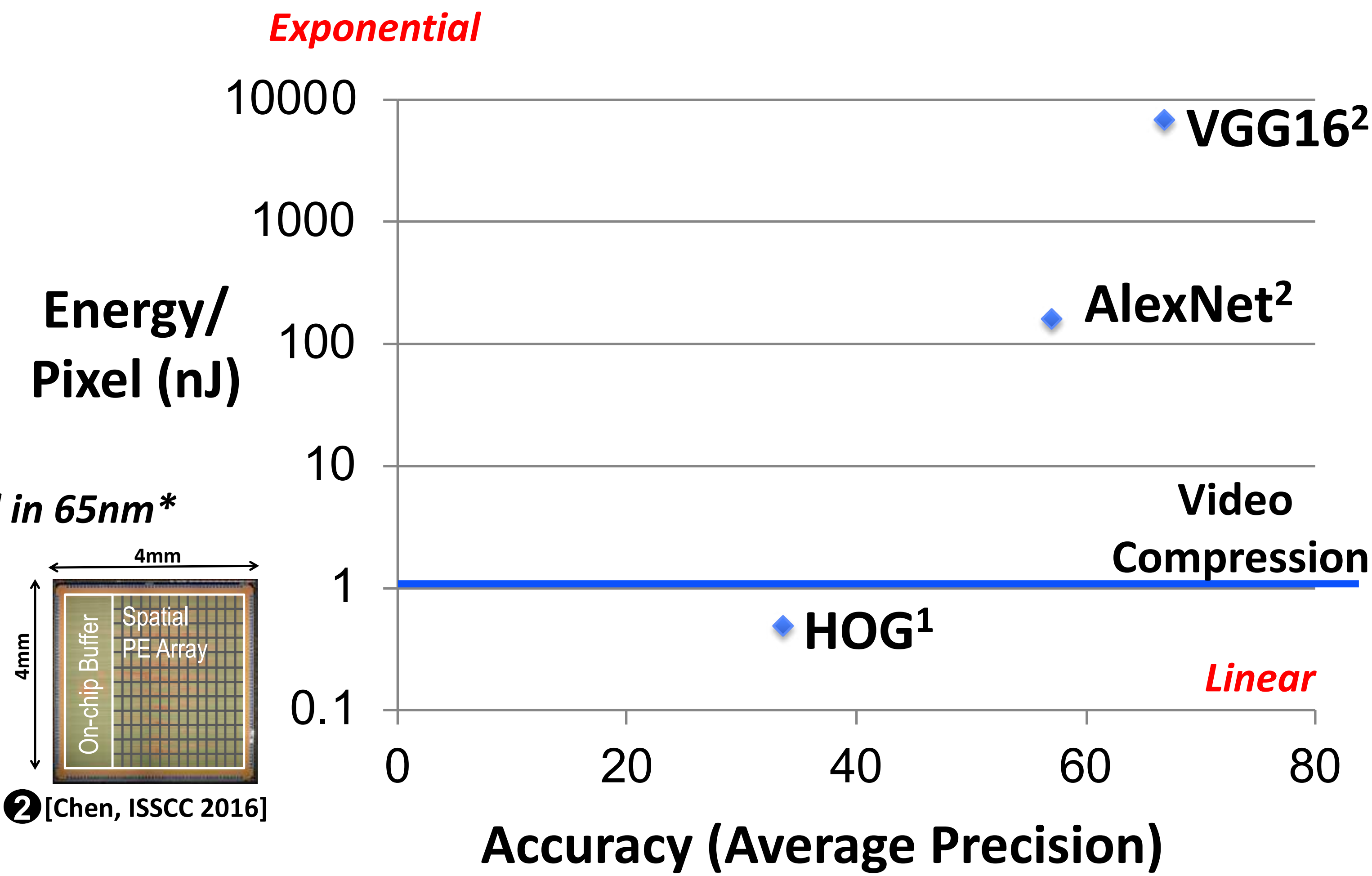
Overall **>10x energy reduction** compared to a mobile GPU (Nvidia TK1)

Results for AlexNet

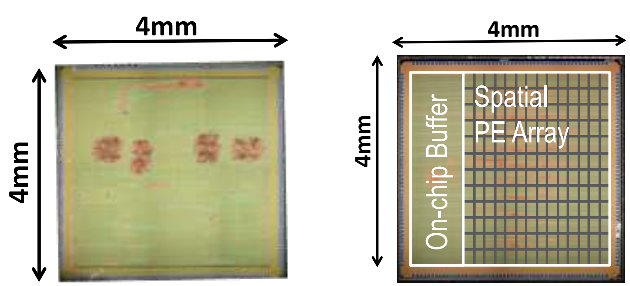
Eyeriss Project Website: http://eyeriss.mit.edu

[Joint work with Joel Emer]

Features: Energy vs. Accuracy



*Measured in 65nm**



❶ [Suleiman, VLSI 2016] ❷ [Chen, ISSCC 2016]

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

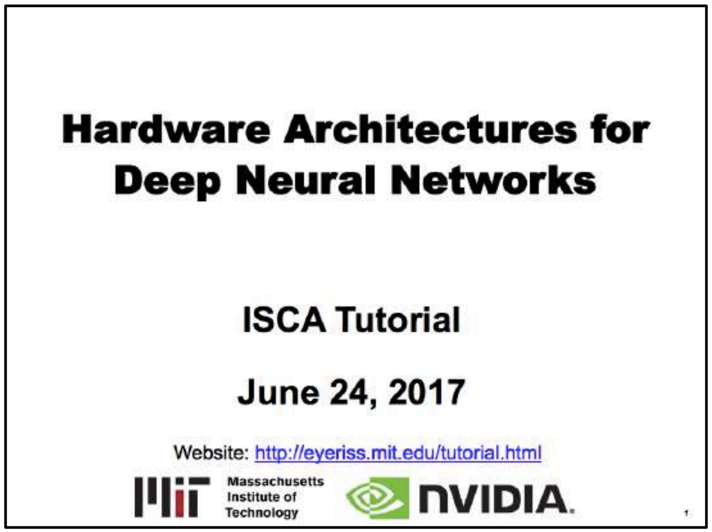
Measured in on VOC 2007 Dataset

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

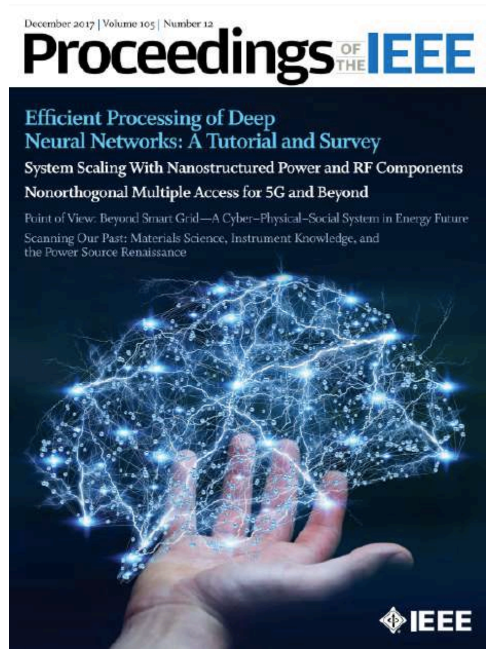
[Suleiman*, Chen*, *ISCAS* 2017]

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



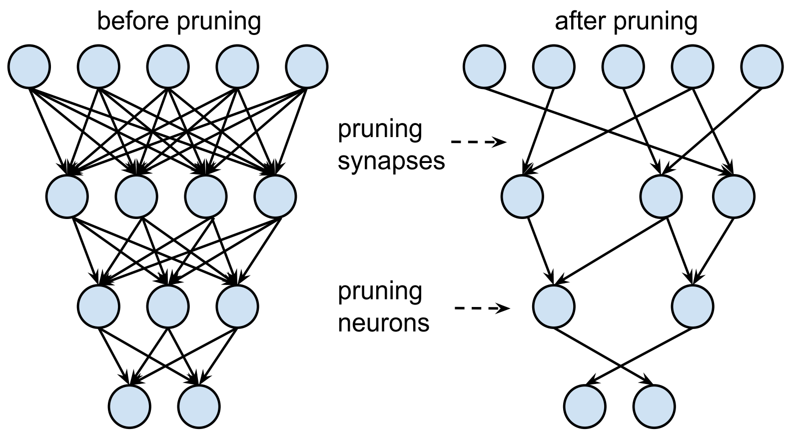
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

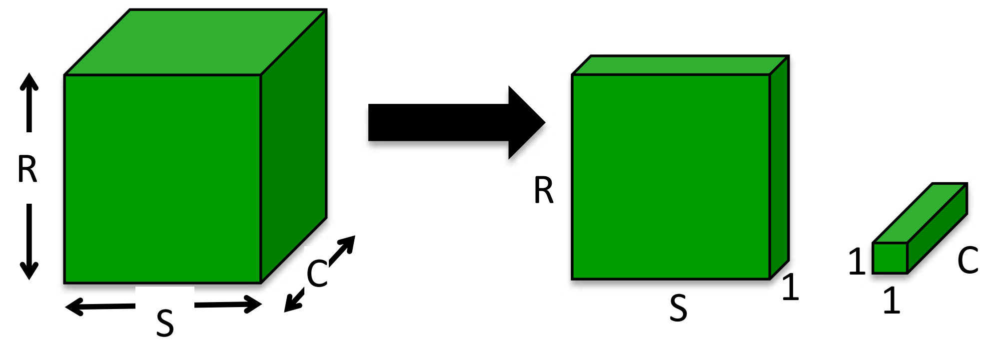
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

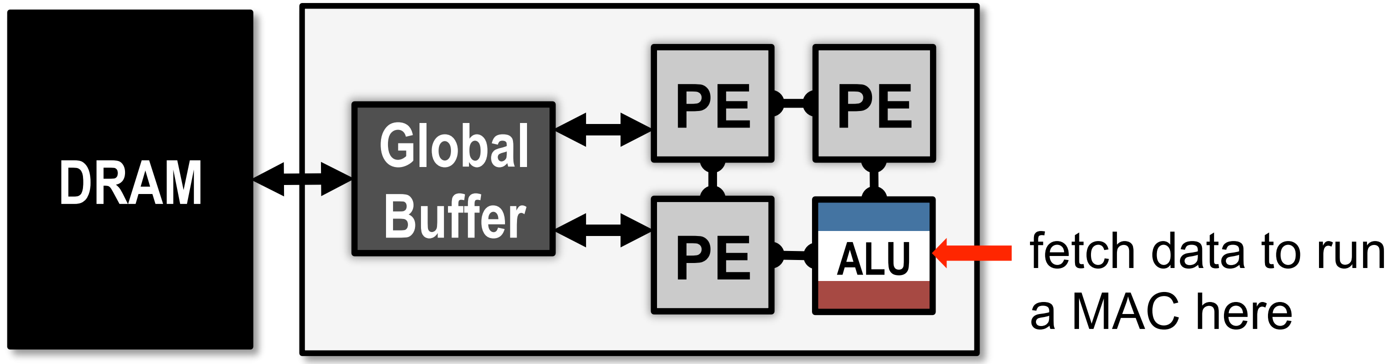


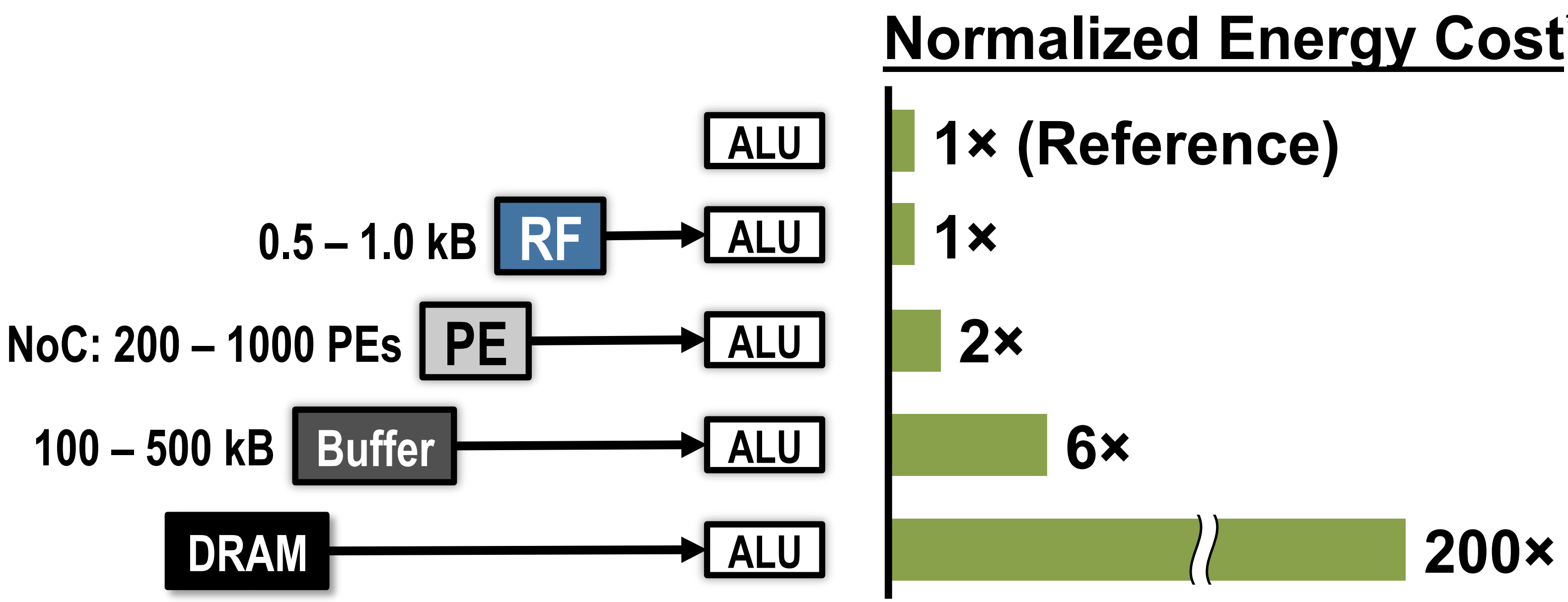
Examples: SqueezeNet, MobileNet

... also reduced precision

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

Data Movement is Expensive

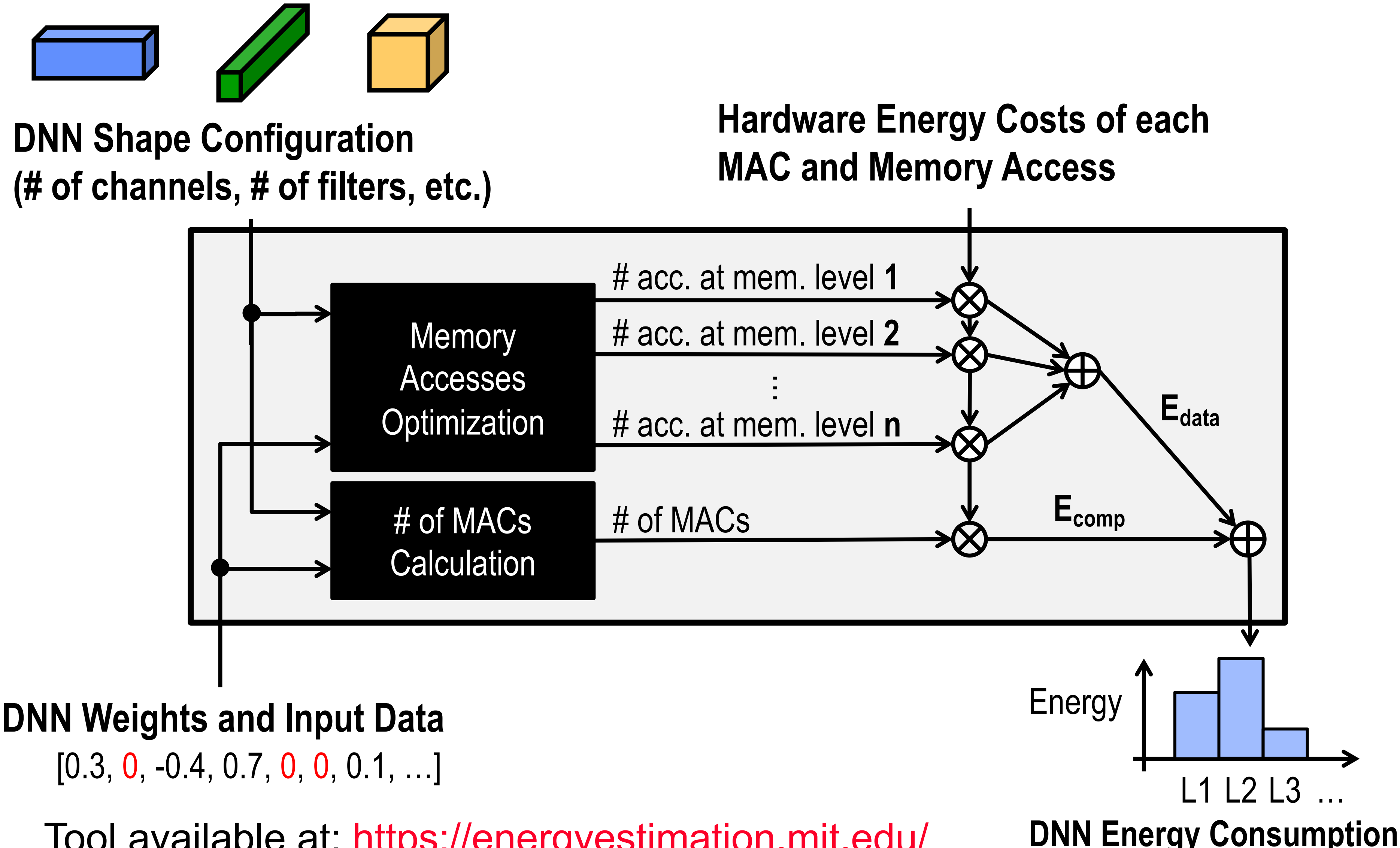




* measured from a commercial 65nm process

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

Energy-Evaluation Methodology



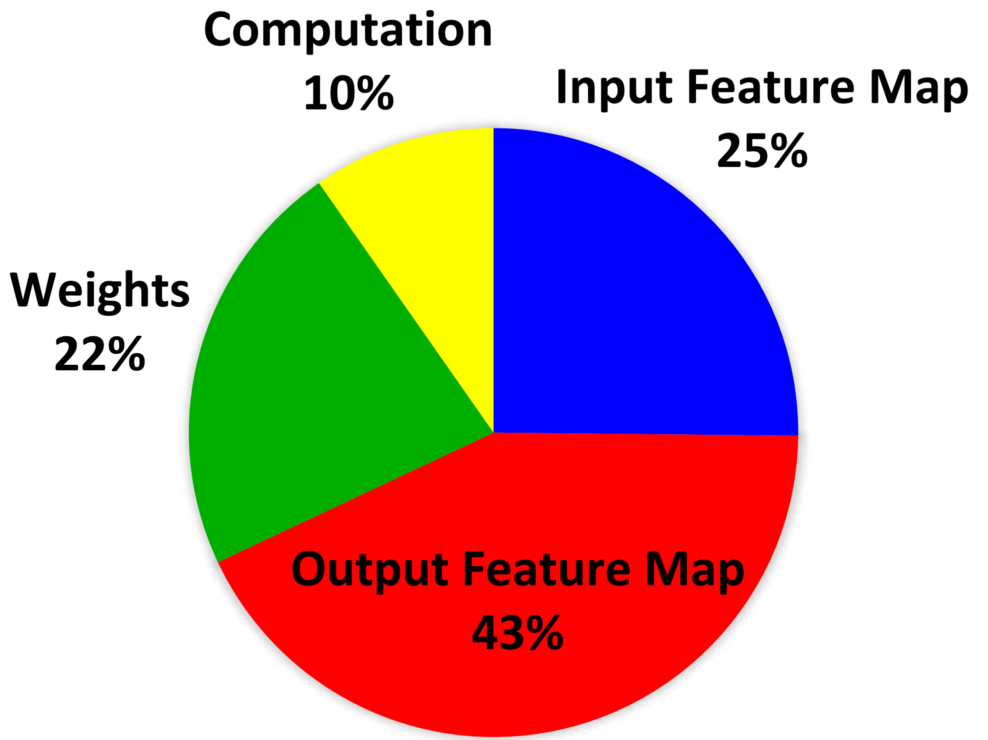
Tool available at: https://energyestimation.mit.edu/

[**Yang**, *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

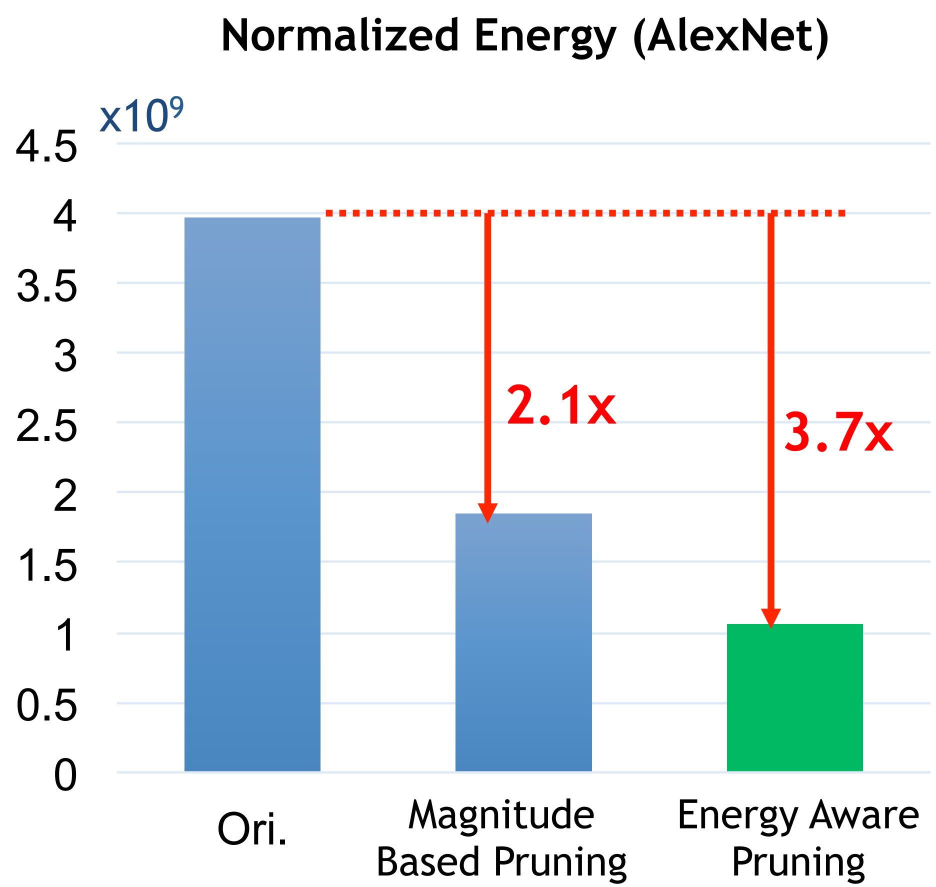


[**Yang**, *CVPR* 2017]

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**

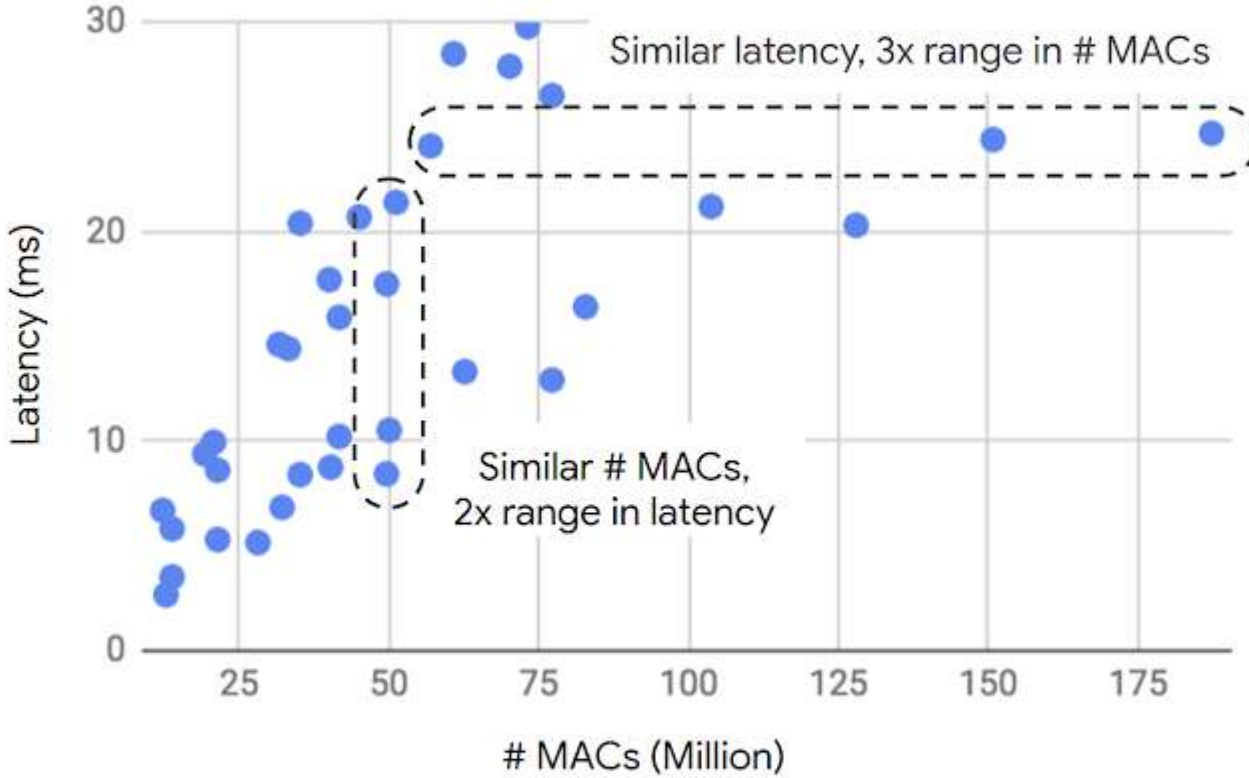


Pruned models available at
http://eyeriss.mit.edu/energy.html

[**Yang**, *CVPR* 2017]

of Operations vs. Latency

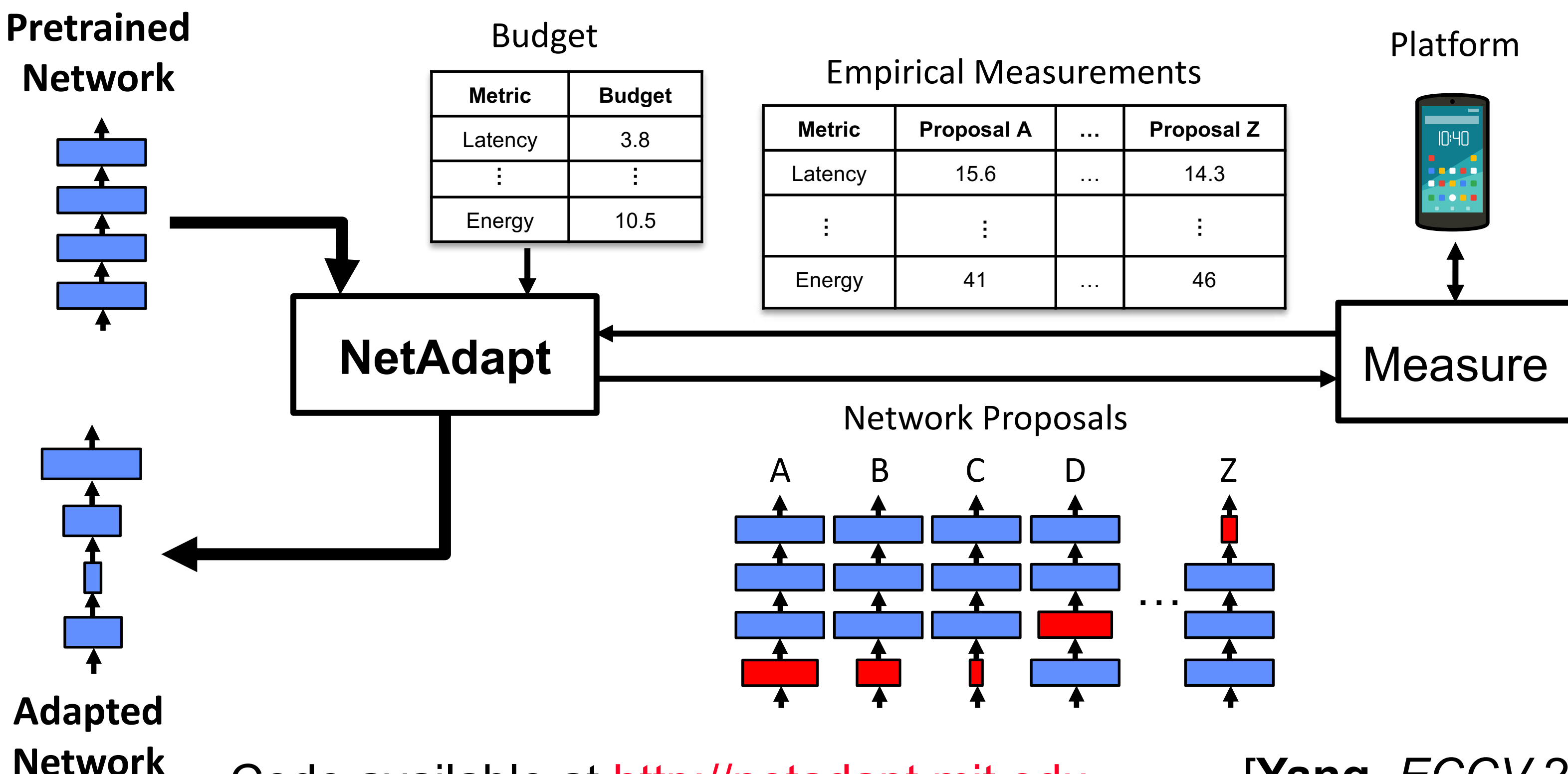
- # of operations (MACs) does not approximate latency well



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)



Metric	Budget
Latency	3.8
⋮	⋮
Energy	10.5

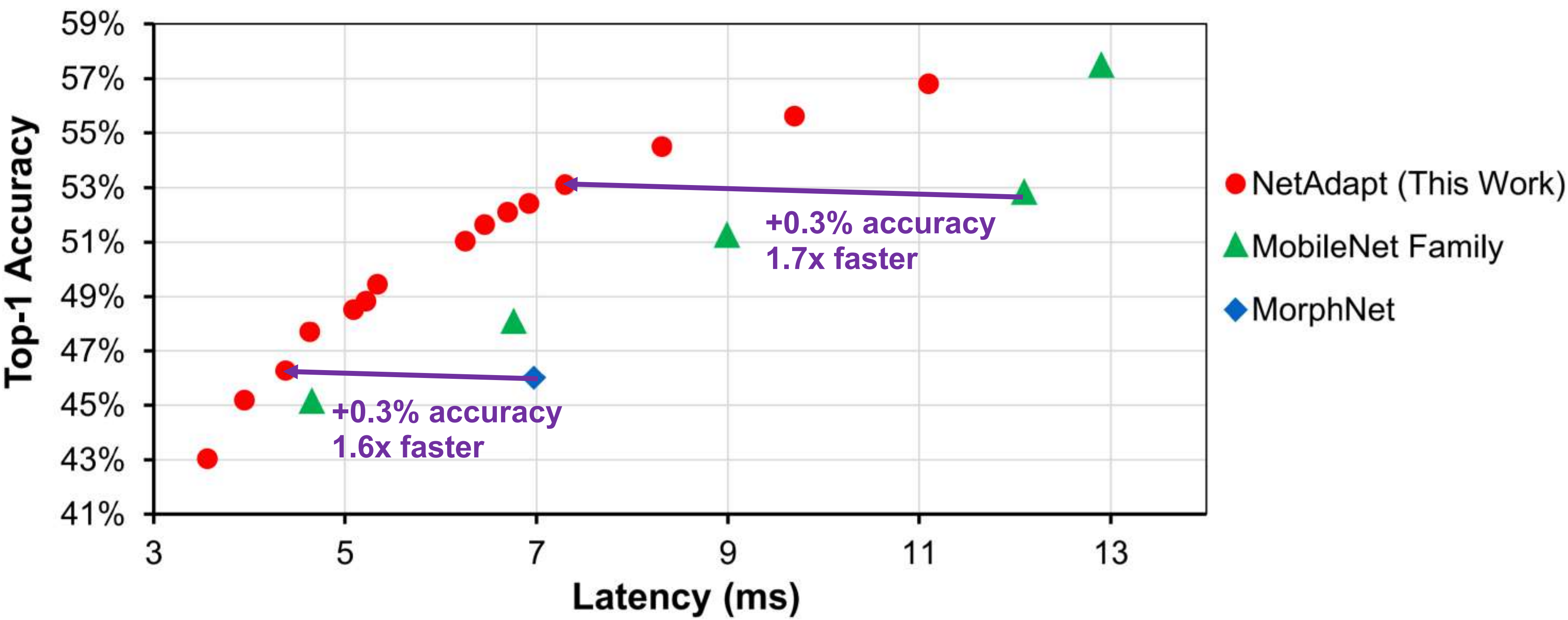
Metric	Proposal A	…	Proposal Z
Latency	15.6	…	14.3
⋮	⋮		⋮
Energy	41	…	46

Code available at http://netadapt.mit.edu

[**Yang**, *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

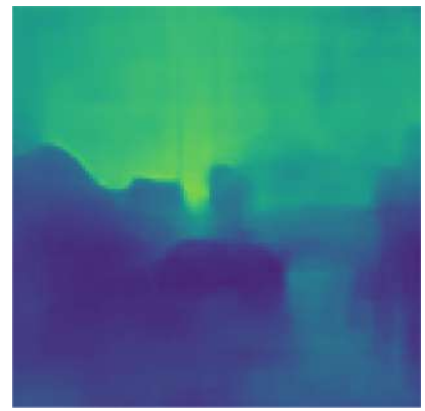
[**Yang**, *ECCV* 2018]

FastDepth: Fast Monocular Depth Estimation

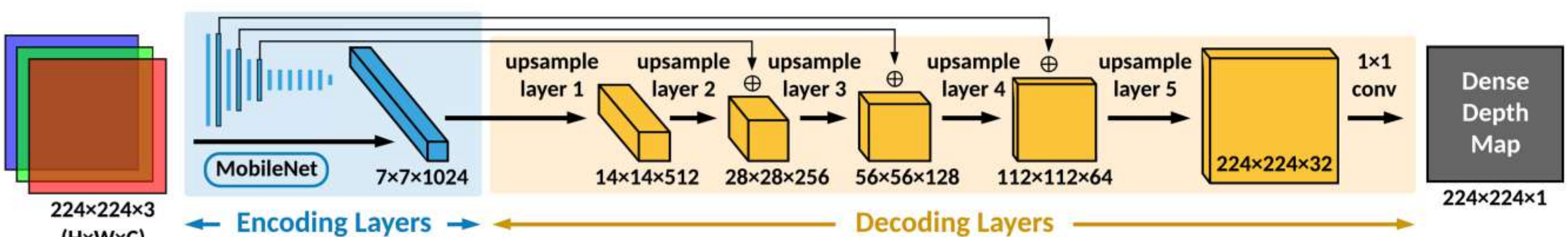
Depth estimation from a single RGB image desirable, due to the relatively low cost and size of monocular cameras.

RGB

Prediction



Auto Encoder DNN Architecture (Dense Output)



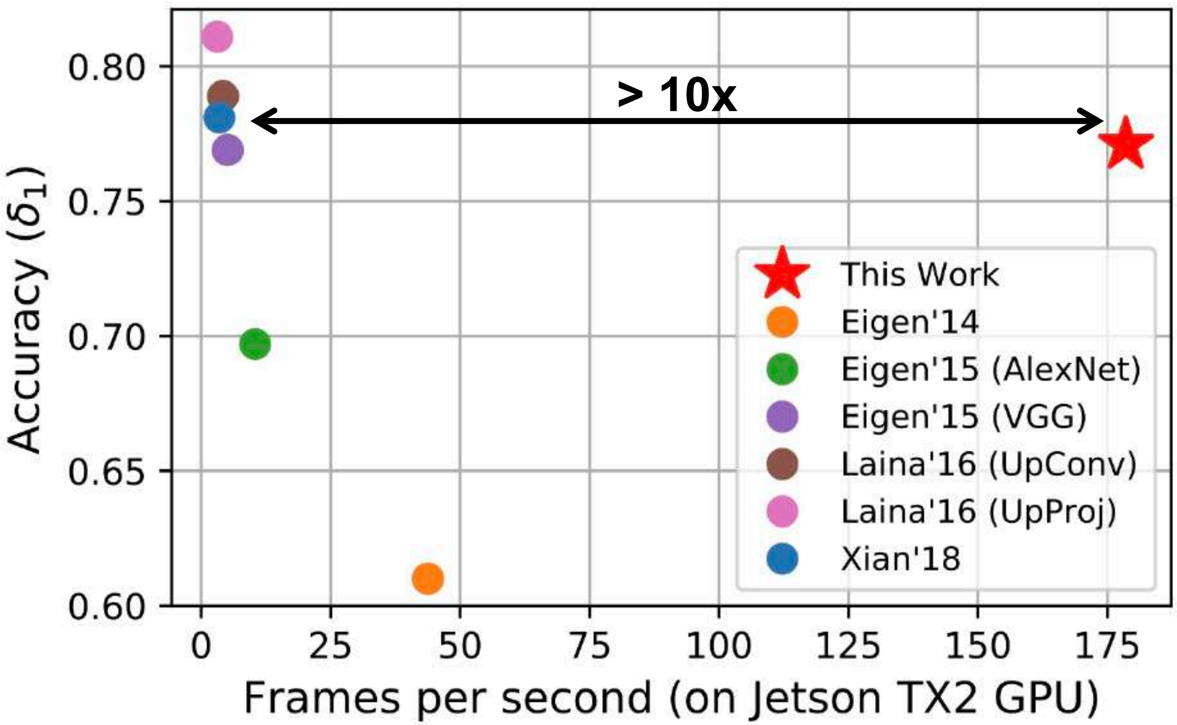
Reduction (similar to classification)

Expansion

[Joint work with Sertac Karaman]

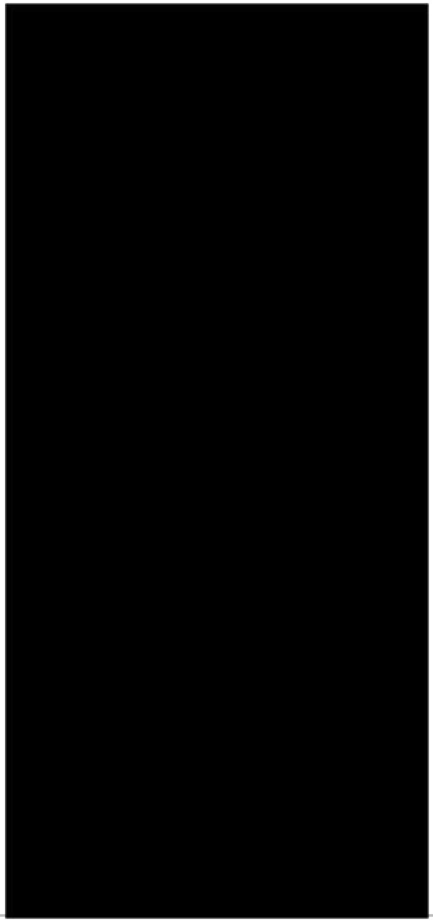
FastDepth: Fast Monocular Depth Estimation

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)

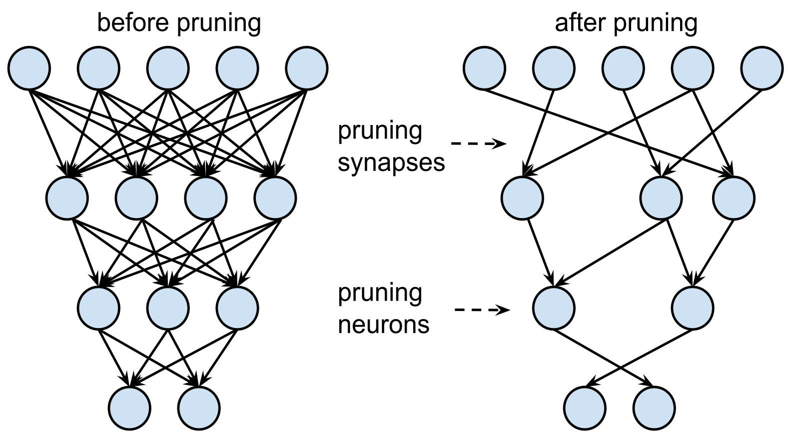
Models available at [*http://fastdepth.mit.edu*](http://fastdepth.mit.edu)



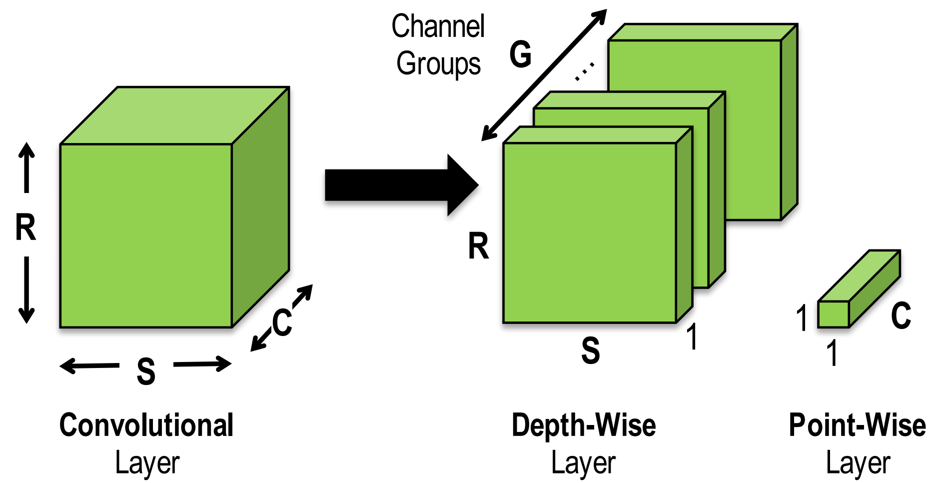
~40fps on an iPhone

Many Efficient DNN Design Approaches

Network Pruning



Compact Network Architectures



Reduce Precision

32-bit float 10100101000000000101000000000100

8-bit fixed 01100110

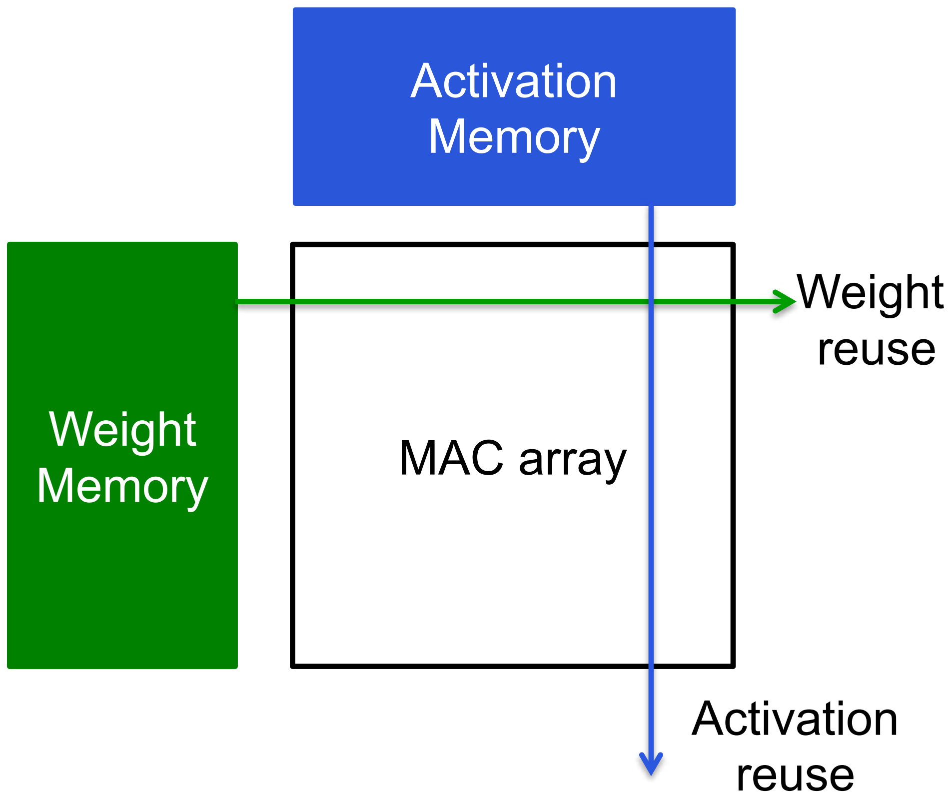
Binary 0

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

[**Chen***, **Yang***, *SysML* 2018]

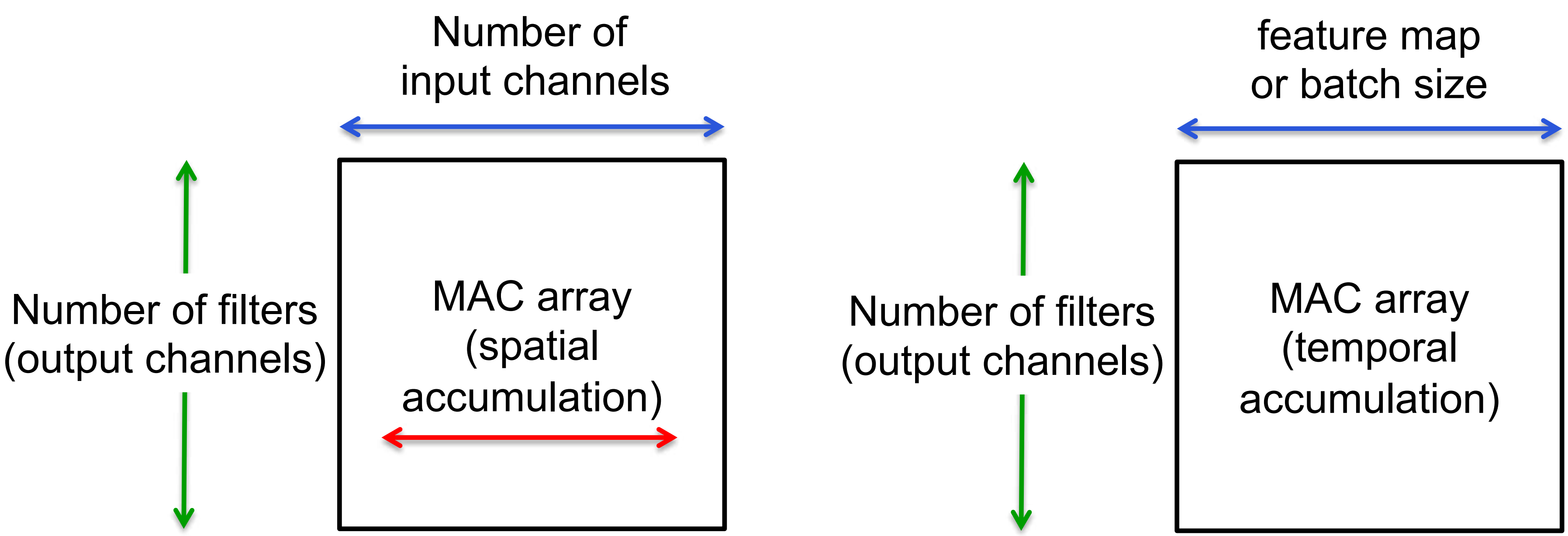
Existing DNN Architectures

- 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



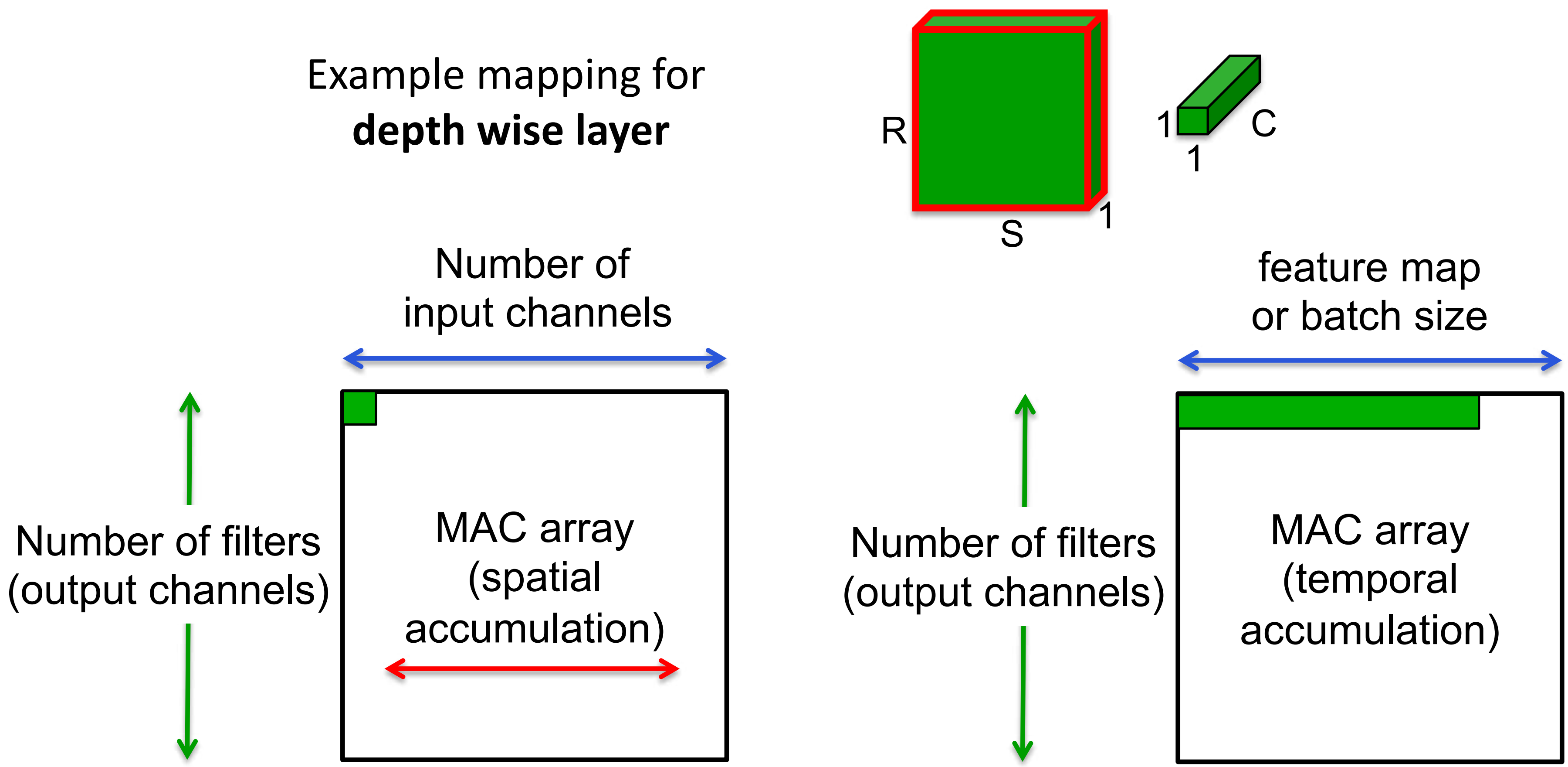
Limitation of Existing DNN Architectures

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



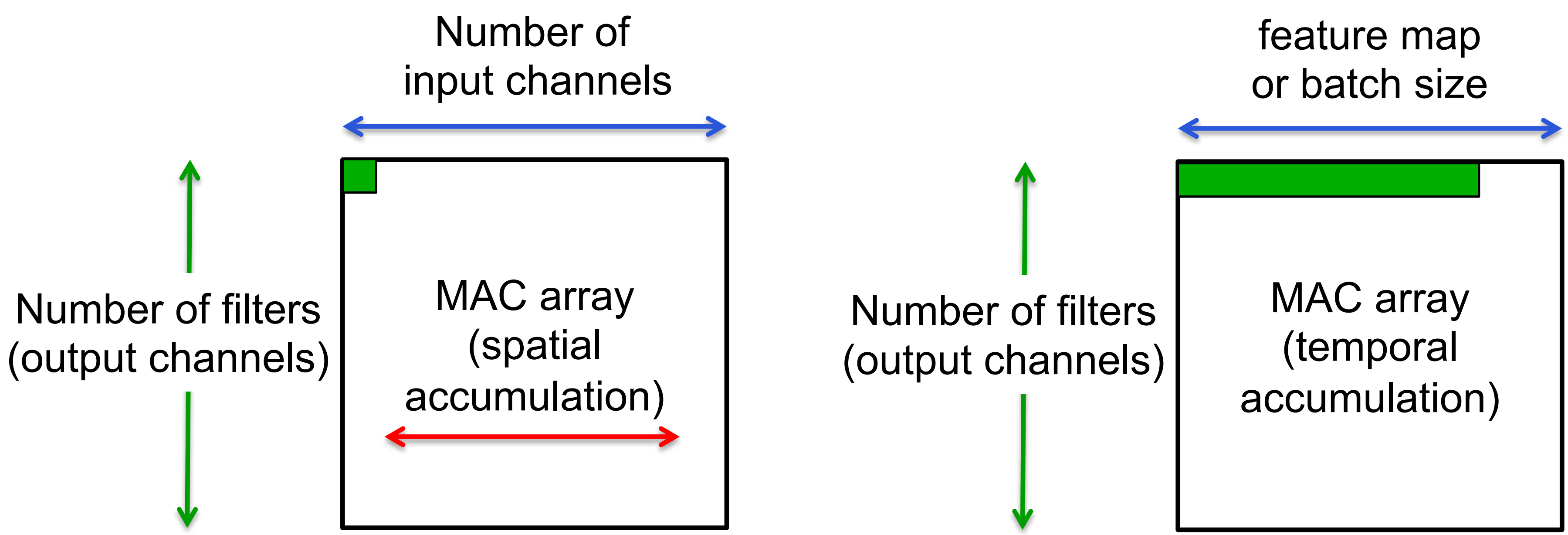
Limitation of Existing DNN Architectures

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



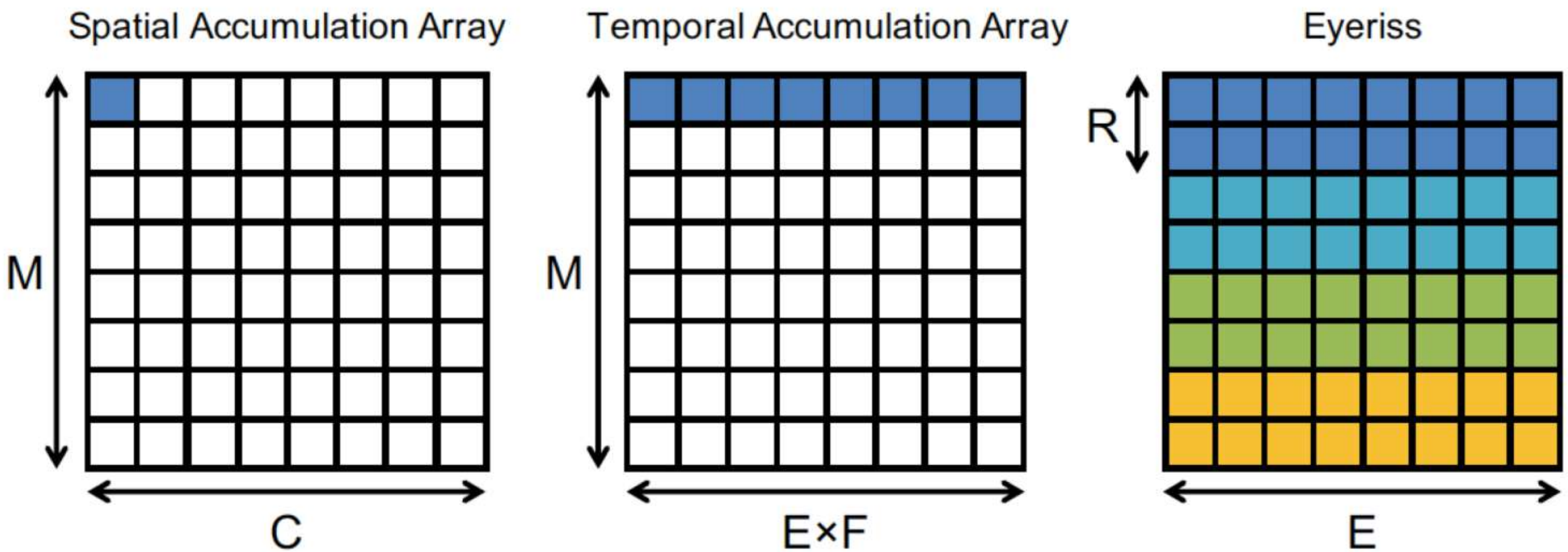
Limitation of Existing DNN Architectures

- **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

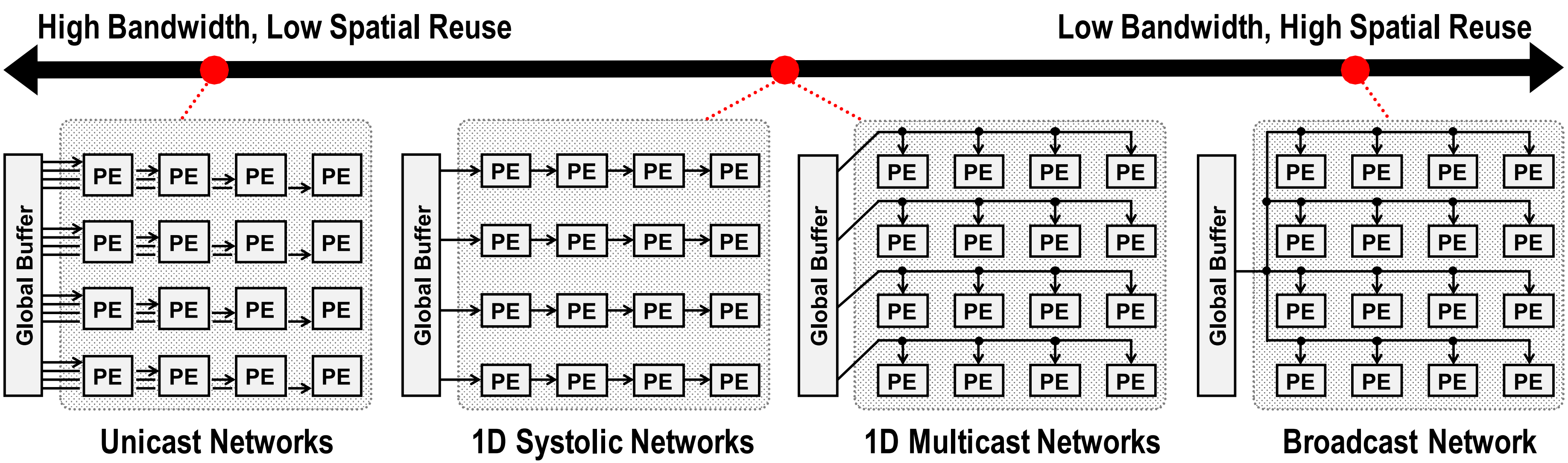
- Use flexible dataflow (**Row Stationary**) to exploit reuse in any dimension of DNN to increase energy efficiency and array utilization



Example: Depth-wise layer

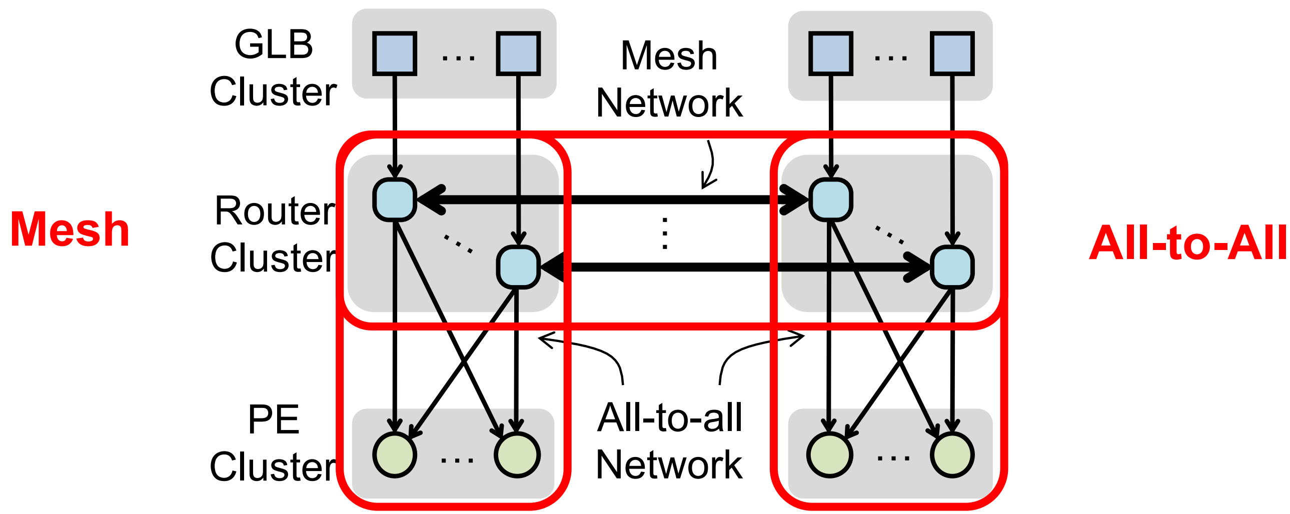
Need Flexible NoC for Varying Reuse

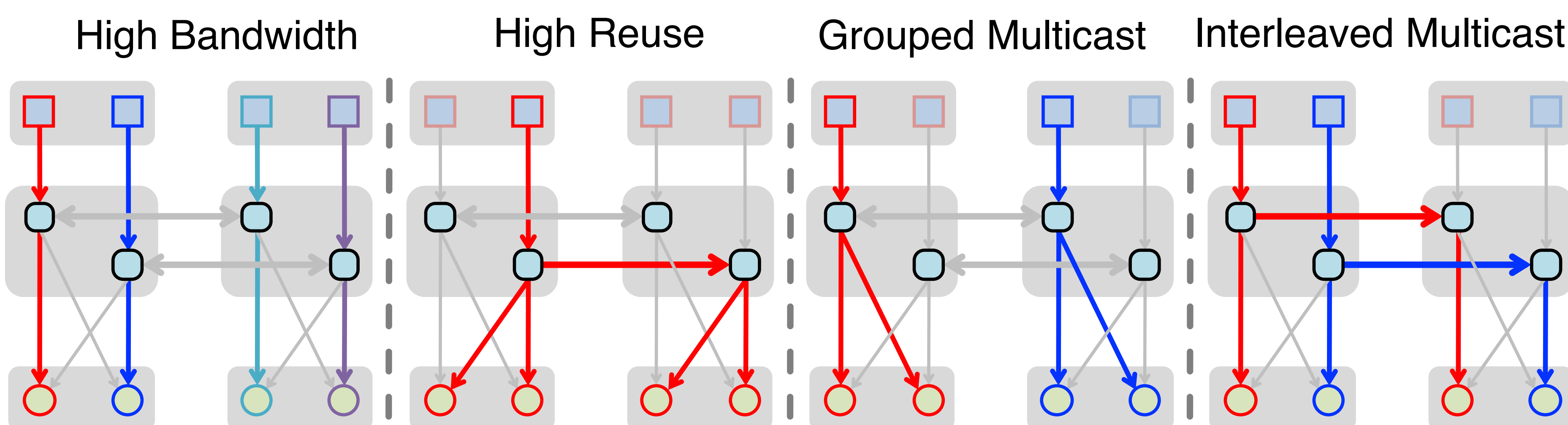
- When reuse available, need **multicast** to exploit spatial data reuse for energy efficiency and high array utilization
- When reuse not available, need **unicast** for high BW for weights for FC and weights & activations for high PE utilization
- An **all-to-all** satisfies above but too expensive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



[**Chen**, *JETCAS* 2019]

Hierarchical Mesh



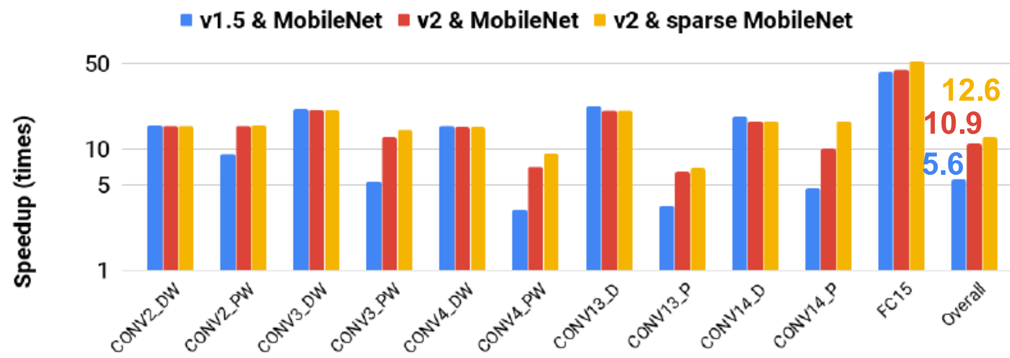


[**Chen**, *JETCAS* 2019]

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

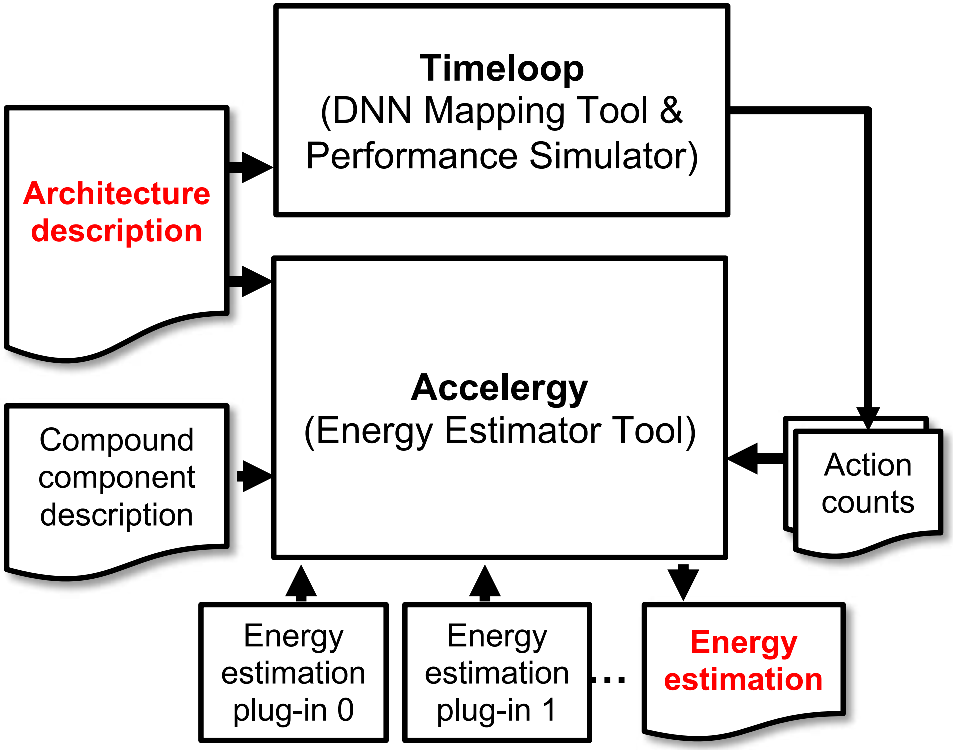
# of PEs	256	1024	16384
AlexNet	17.9x	71.5x	1086.7x
GoogLeNet	10.4x	37.8x	448.8x
MobileNet	15.7x	57.9x	873.0x

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

[**Chen**, *JETCAS* 2019]

DL Processor Evaluation Tools

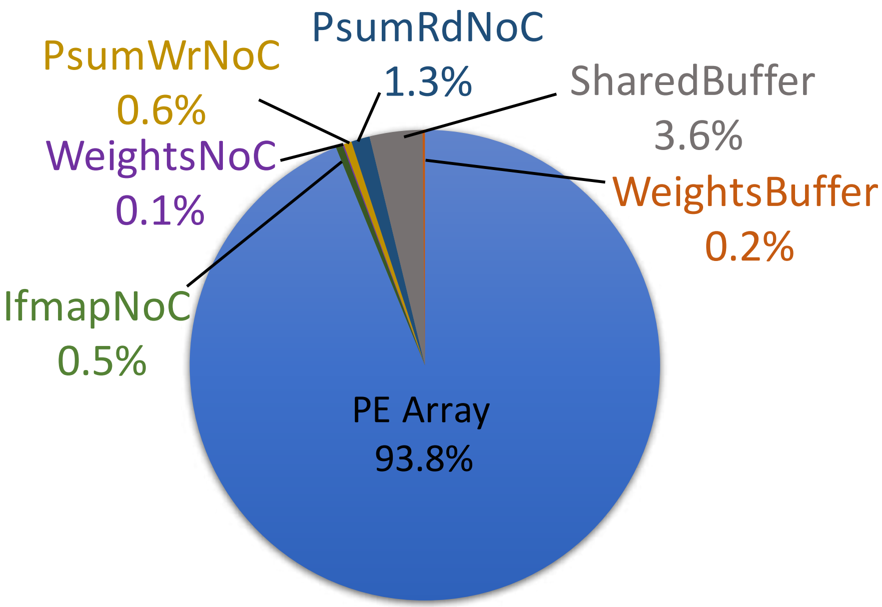
- Require systematic way to
  - Evaluate and compare wide range of DL processor designs
  - Rapidly explore design space
- Accelergy [**Wu**, *ICCAD* 2019]
  - Early stage energy estimation tool at the architecture level
    - Estimate energy consumption based on architecture level components (e.g., # of PEs, GLB size, NoC)
  - Evaluate architecture level energy impact of emerging devices
    - Plug-ins for different technologies
- Timeloop [**Parashar**, *ISPASS* 2019]
  - DNN mapping tool
  - Performance Simulator → Action counts



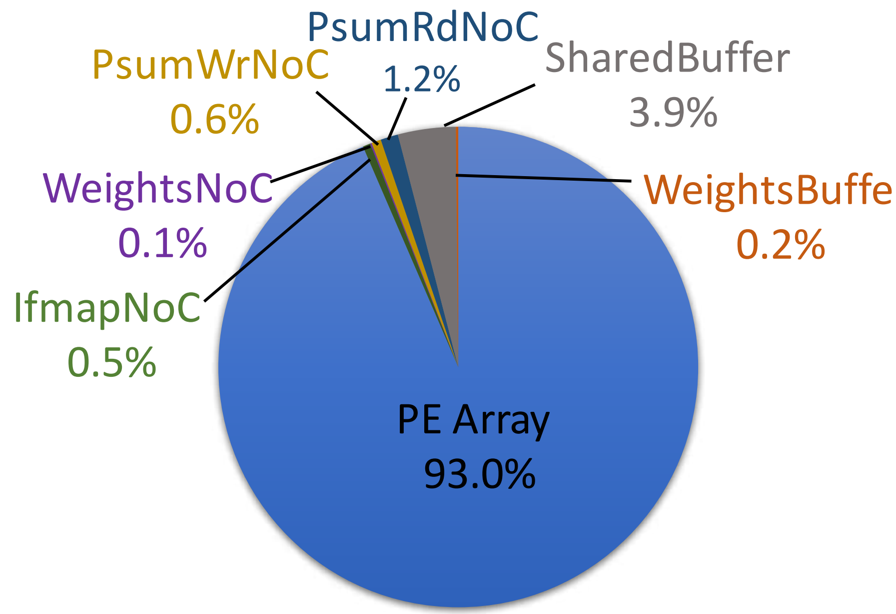
Open-source code available at:
http://accelergy.mit.edu

Accelergy Estimation Validation

- Validation on Eyeriss [Chen, *ISSCC* 2016]
  - Achieves 95% accuracy compared to post-layout simulations
  - Can accurately captures energy breakdown at different granularities



Ground Truth Energy Breakdown



Accelergy Energy Breakdown

Open-source code available at: http://accelergy.mit.edu

[**Wu**, *ICCAD* 2019]

Visual-Inertial Localization

Determines location/orientation of robot from images and IMU

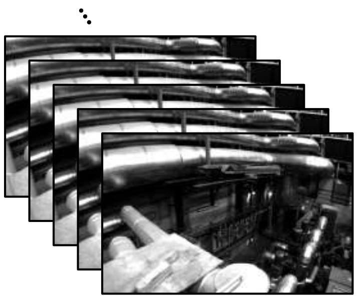
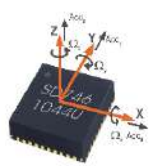


Image sequence

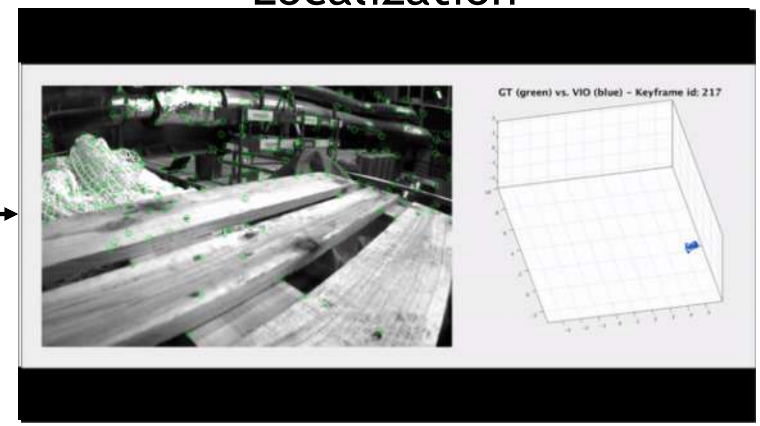
IMU

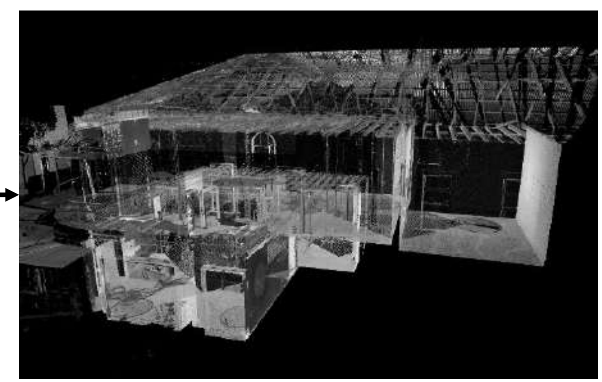
Inertial Measurement Unit



Visual-Inertial Odometry (VIO)*

Localization





Mapping

***Subset of SLAM algorithm**

(Simultaneous Localization And Mapping)

Localization at Under 25 mW

First chip that performs ***complete*** Visual-Inertial Odometry

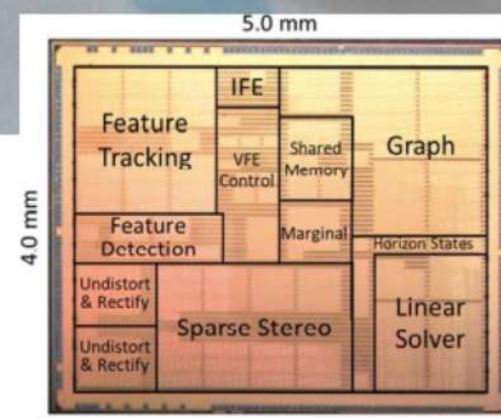
Front-End for camera
(Feature detection, tracking, and outlier elimination)

Front-End for IMU
(pre-integration of accelerometer and gyroscope data)

Back-End Optimization of Pose Graph

Consumes **684× and 1582×** less energy than mobile and desktop CPUs, respectively

Technology	65nm CMOS	Supply	1 V
Chip area (mm²)	4.0 x 5.0	Resolution	752x480
Core area (mm²)	3.54 x 4.54	Camera rate	28 - 171 fps
Logic gates	2,043 kgates	Keyframe rate	16 - 90 fps
SRAM	854KB	Average Power	24 mW
VFE Frequency	62.5 MHz	GOPS	10.5 – 59.1
BE Frequency	83.3 MHz	GFLOPS	1 – 5.7



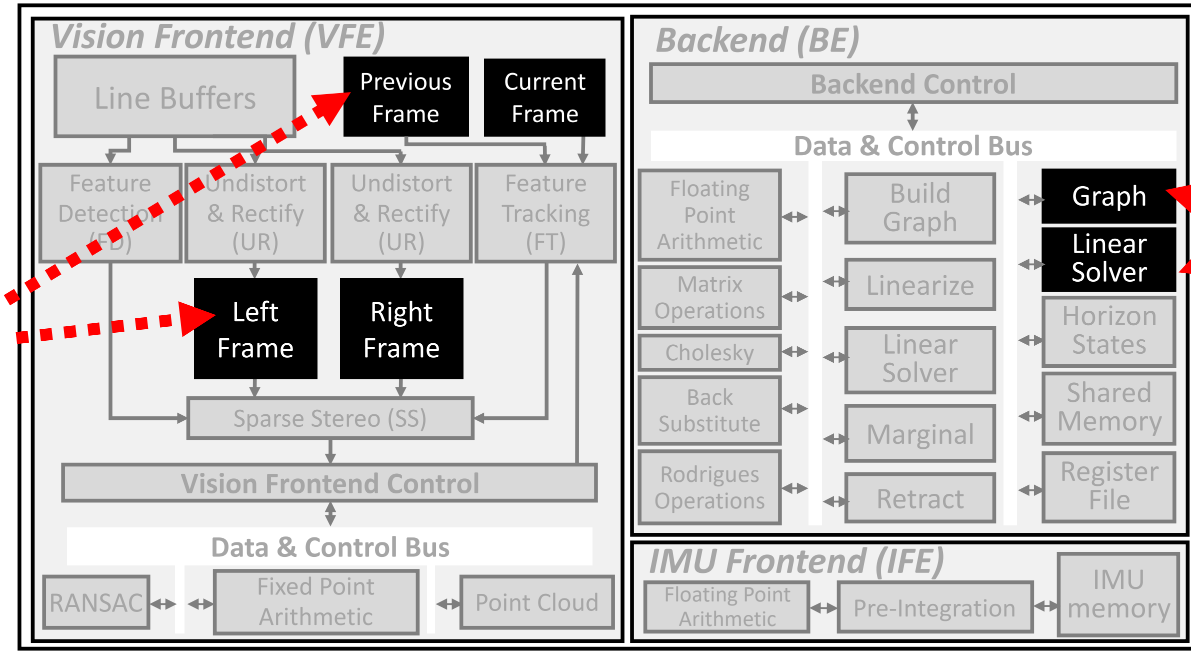
[**Zhang***, **Suleiman***, *RSS* 2017], [**Suleiman**, *VLSI* 2018]

[Joint work with Sertac Karaman]

Key Methods to Reduce Data Size

Navion: *Fully integrated system – no off-chip processing or storage*

Apply Low Cost Frame Compression



Exploit Sparsity in Graph and Linear Solver

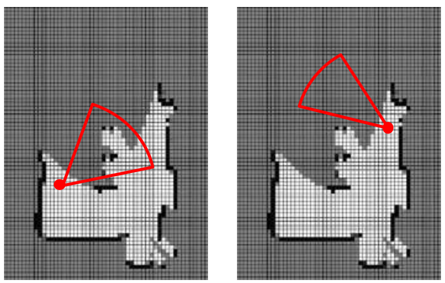
Use **compression** and **exploit sparsity** to reduce memory down to 854kB

Where to Go Next: Planning and Mapping

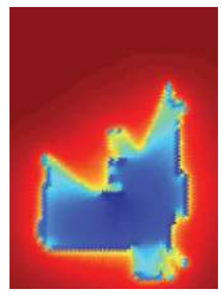
Robot Exploration: *Decide where to go by computing Shannon Mutual Information*

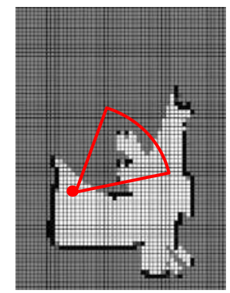
Select candidate scan locations → Compute Shannon MI and choose best location → Move to location and scan → Update Occupancy Map →

Where to scan?

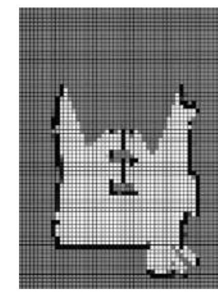


Mutual Information



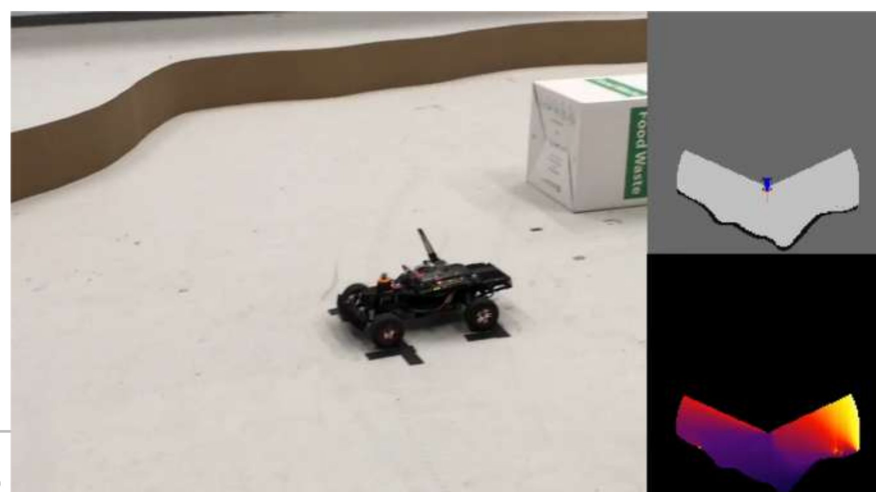


Updated Map



Exploration with a mini race car using motion capture for localization

[**Zhang**, *ICRA* 2019]

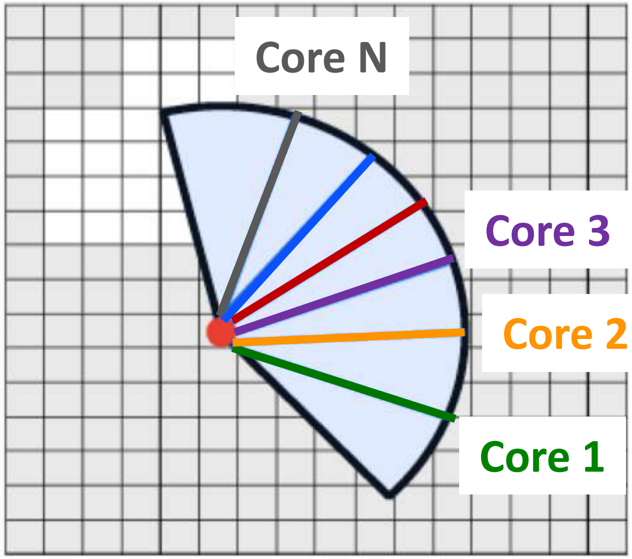


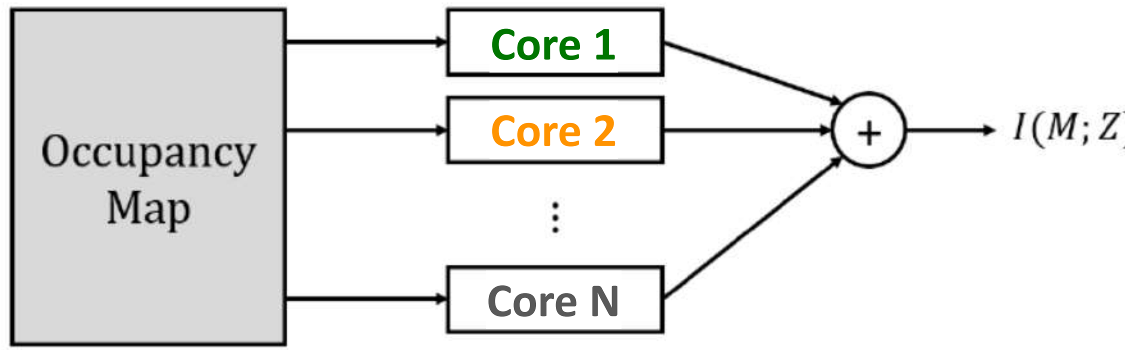
Occupancy map with planned path

MI surface

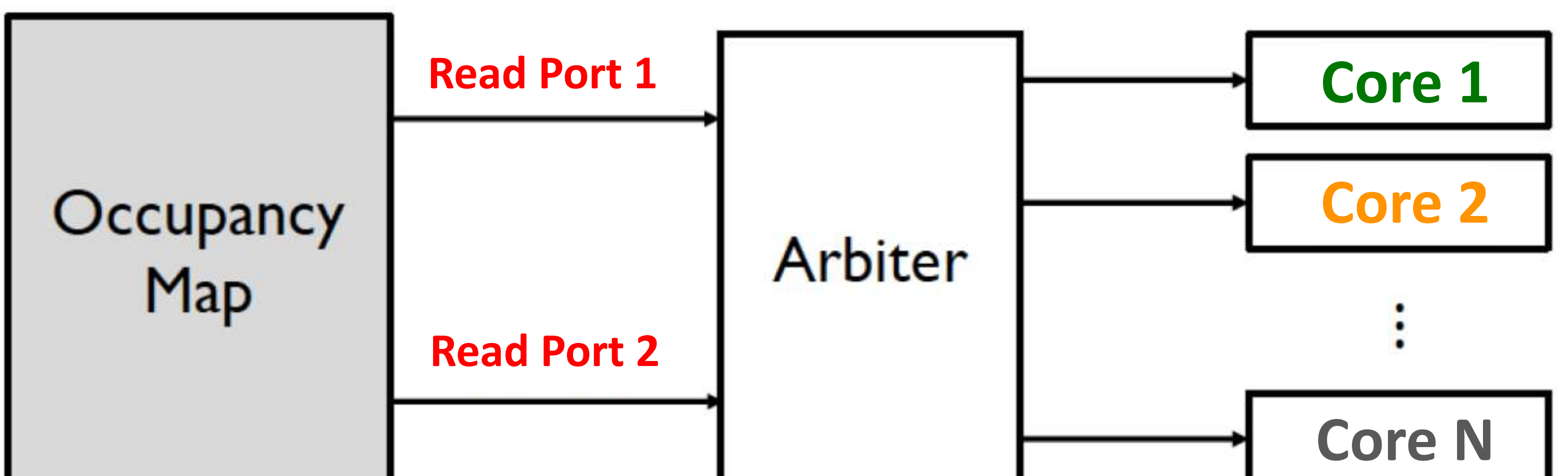
Challenge is Data Delivery to All Cores

Process multiple beams in parallel





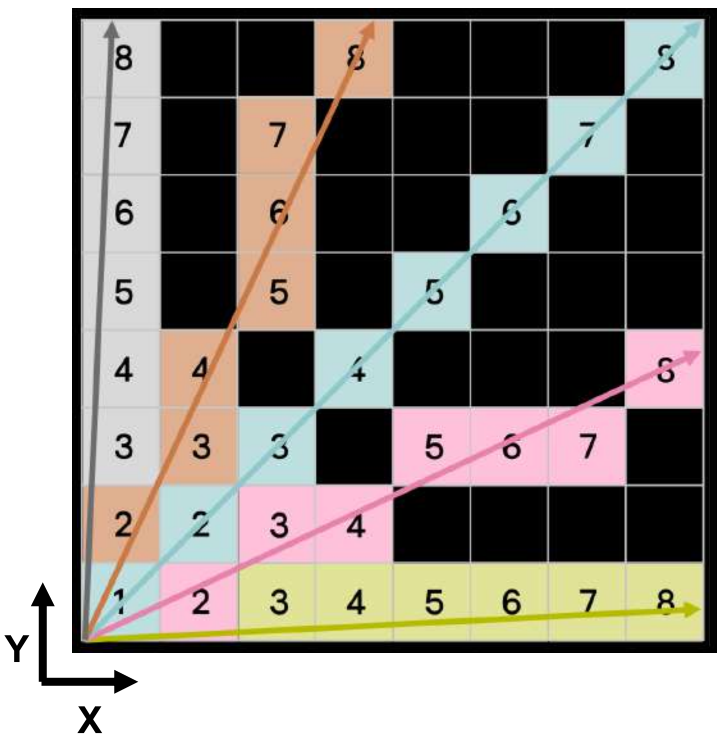
Data delivery from memory is limited



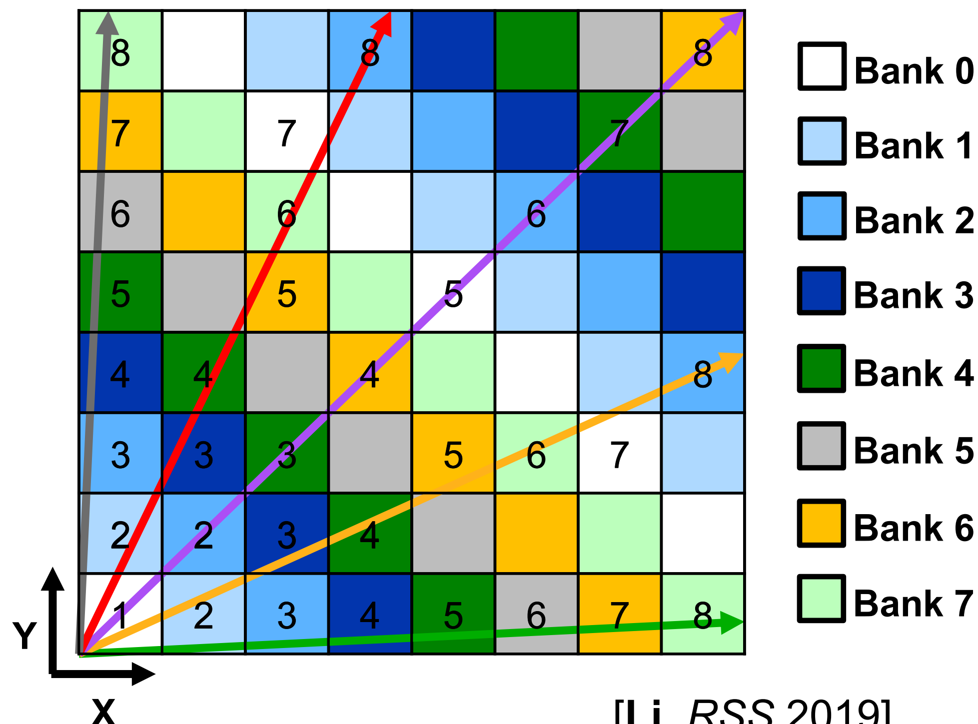
Specialized Memory Architecture

Break up map into **separate memory banks** and novel storage pattern to minimize read conflicts when processing different beams in parallel.

Memory Access Pattern



Diagonal Banking Pattern



[**Li**, *RSS* 2019]

Compute the mutual information for an **entire map** of 20m x 20m at 0.1m resolution **in under a second** → a 100x speed up versus CPU for 1/10th of the power.

Monitoring Neurodegenerative Disorders

Dementia affects 50 million people worldwide today (75 million in 10 years) [World Alzheimer's Report]

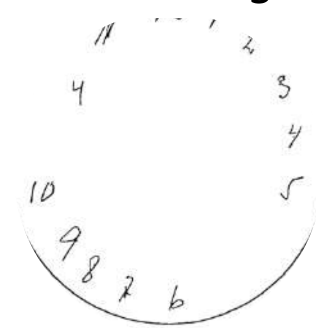
Mini-Mental State Examination (MMSE)

Q1. What is the year? Season? Date?

Q2. Where are you now? State? Floor?

Q3. Could you count backward from 100 by sevens? (93, 86, …)

Clock-drawing test



Agrell et al. *Age and Ageing,* 1998.

- Neuropsychological assessments are **time consuming** and **require a trained specialist**
- Repeat **medical assessments** are **sparse**, mostly **qualitative**, and suffer from **high retest variability**

[Joint work with Thomas Heldt and Charlie Sodini]

Use Eye Movements for *Quantitative* Evaluation

Eye movements can be used to quantitatively evaluate severity, progression or regression of neurodegenerative diseases

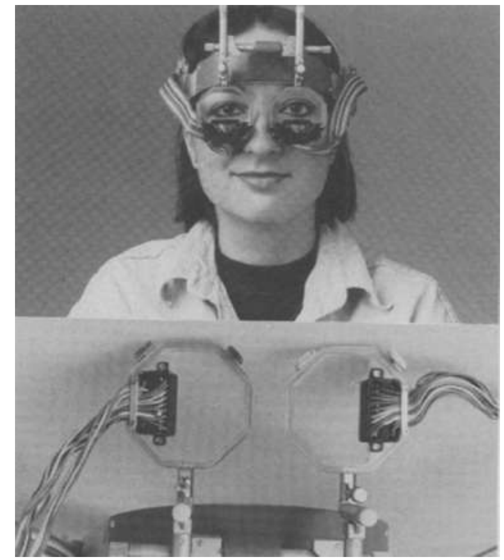
High-speed camera

Phantom v25-11

Substantial head support

SR EYELINK 1000 PLUS

IR illumination

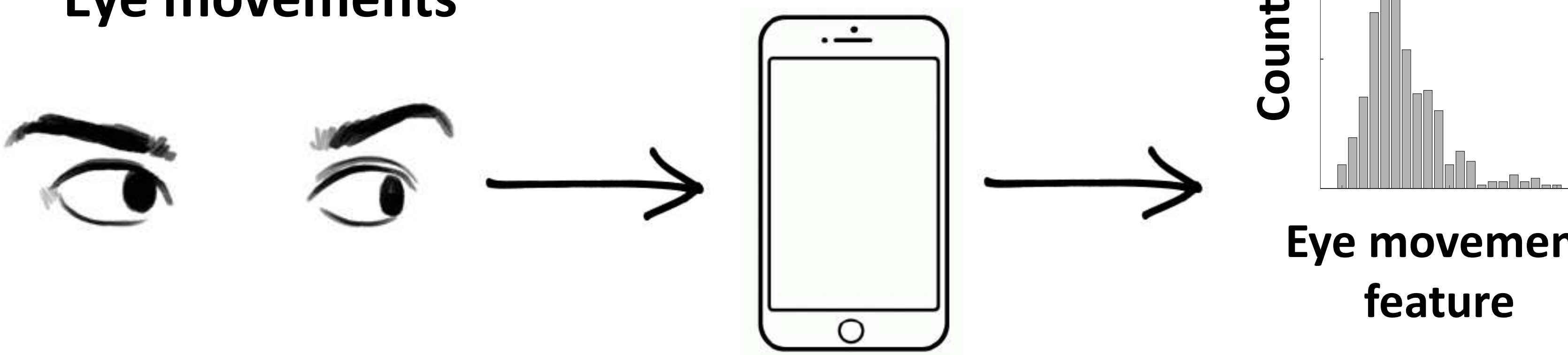


Reulen et al., *Med. & Biol. Eng. & Comp,* 1988.

Clinical measurements of saccade latency are done in constrained environments that rely on specialized, costly equipment.

Measure Eye Movements Using Phone

Eye movements **Smartphone**

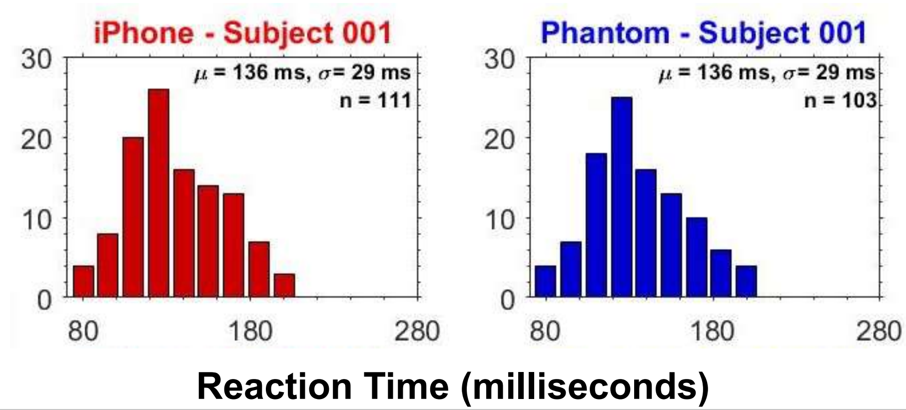


Develop algorithm to measure eye movement using a **consumer-grade camera** rather than high-cost research-grade camera.

Enable low-cost in-home longitudinal measurements.

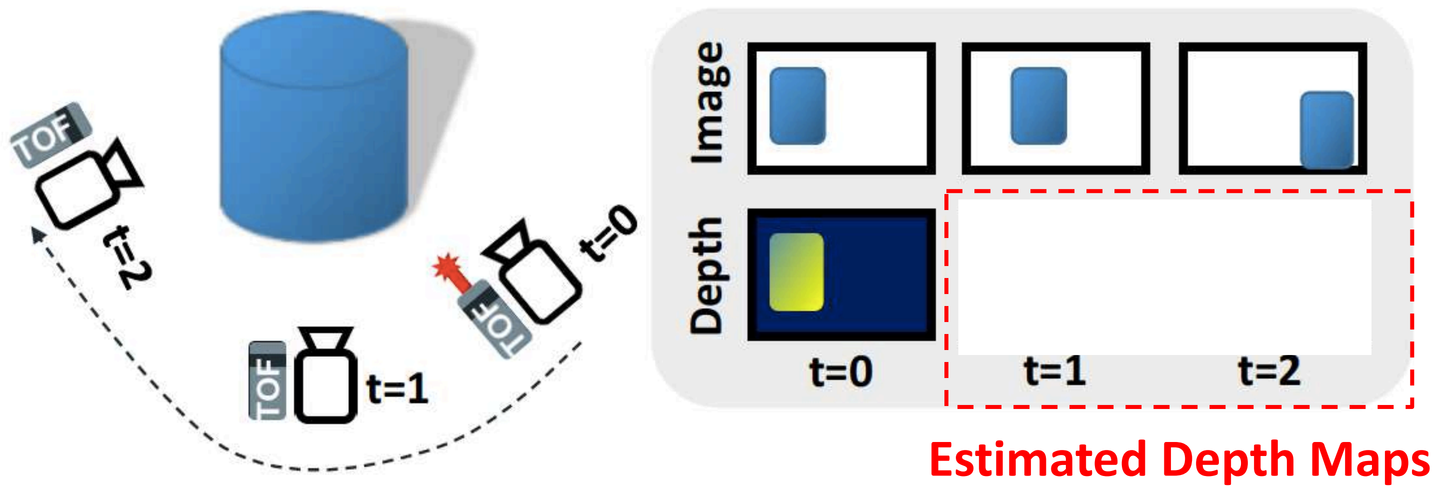
iPhone 6 (< $1k)

Phantom ($100k)



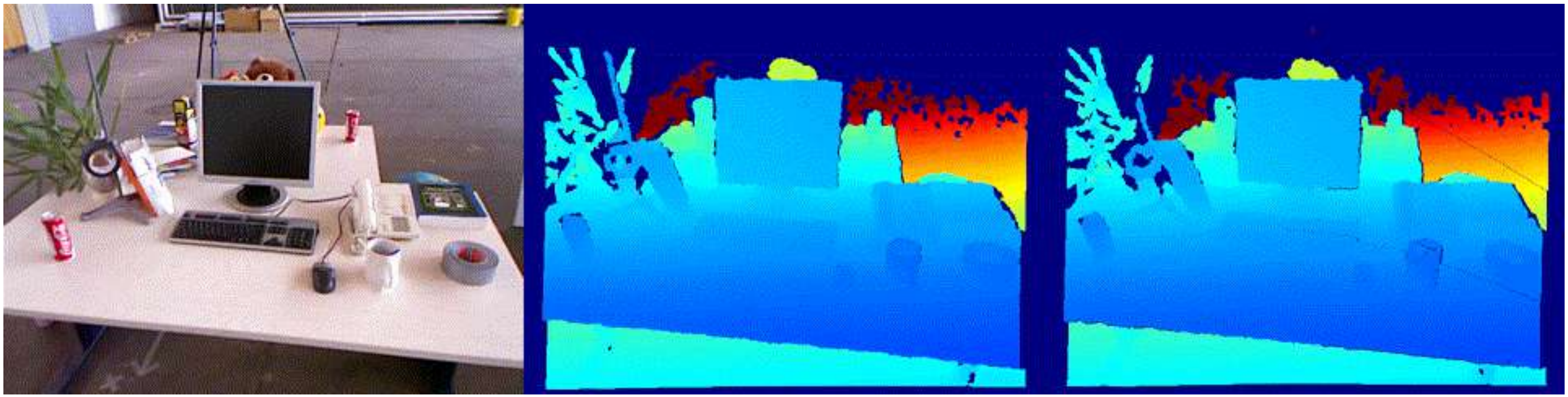
Low Power 3D Time of Flight Imaging

- Pulsed Time of Flight: Measure distance using round trip time of laser light for each image pixel
  - **Illumination + Imager Power: 2.5 – 20 W for range from 1 - 8 m**
- Use computer vision techniques and passive images to estimate changes in depth without turning on laser
  - **CMOS Imaging Sensor Power: < 350 mW**



Real-time Performance on Embedded Processor
VGA @ 30 fps on Cortex-A7 (< 0.5W active power)

Results of Low Power Depth ToF Imaging



RGB Image

Depth Map **Ground Truth**

Depth Map **Estimated**

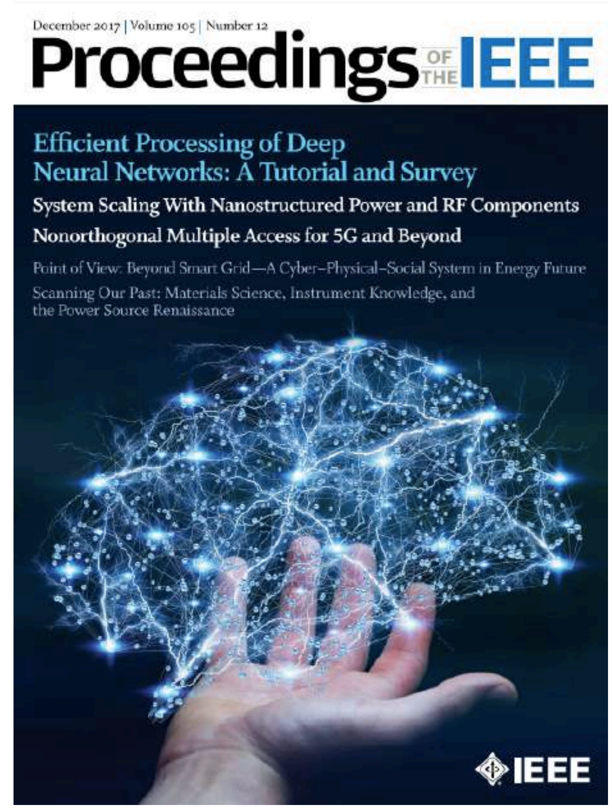
Mean Relative Error: 0.7%
Duty Cycle (on-time of laser): 11%

[**Noraky**, *ICIP* 2017]

Summary

- Energy-Efficient AI extends the reach of AI beyond the cloud by **reducing communication requirements, enabling privacy, and providing low latency** so that AI can be used in wide range of applications ranging from robotics to health care.
- **Cross-layer design with specialized hardware** enables energy-efficient AI, and will be critical to the progress of AI over the next decade.

Additional Resources



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**, Dec. 2017

Book Coming Spring 2020!

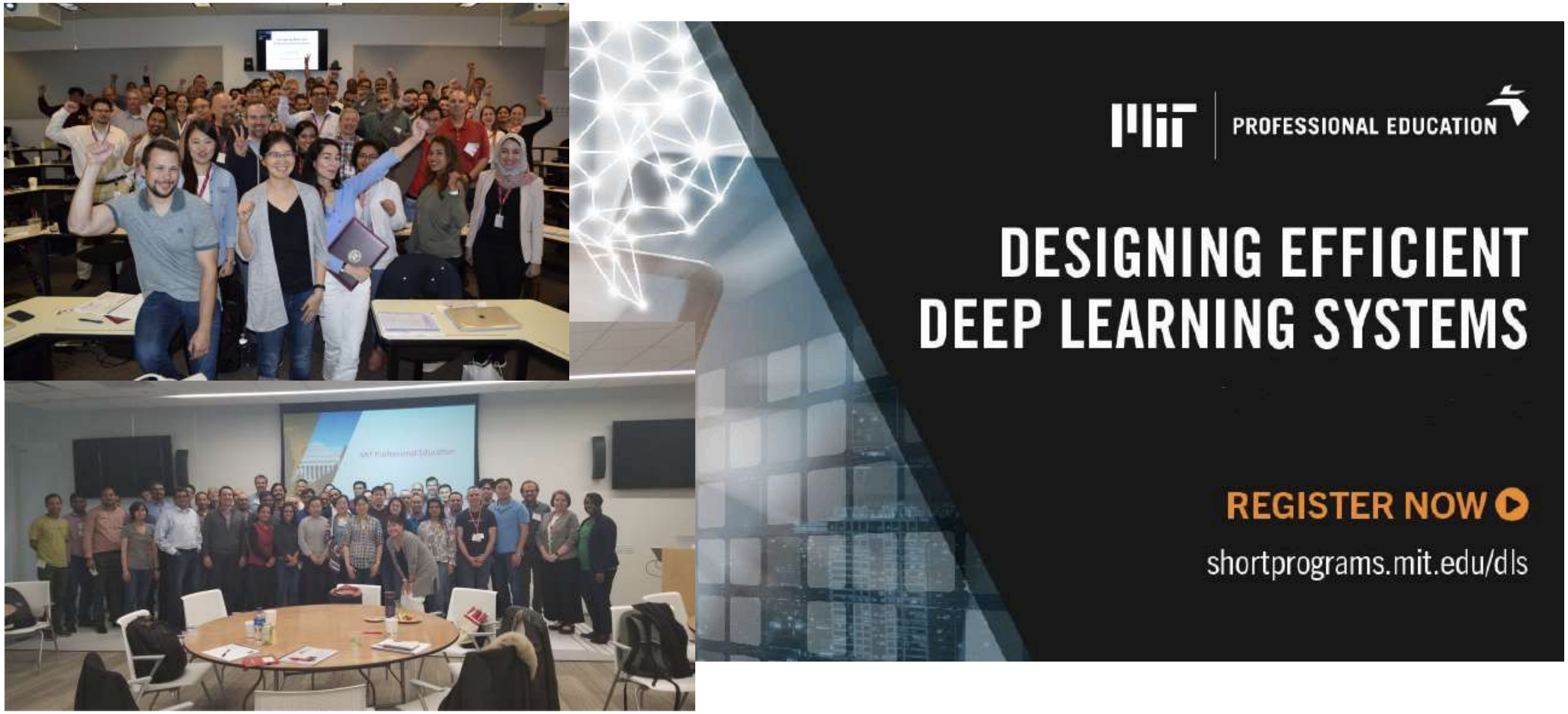
More info about
Tutorial on DNN Architectures
http://eyeriss.mit.edu/tutorial.html

For updates

EEMS Mailing List

Follow @eems_mit

Additional Resources



MIT Professional Education Course on

"Designing Efficient Deep Learning Systems"

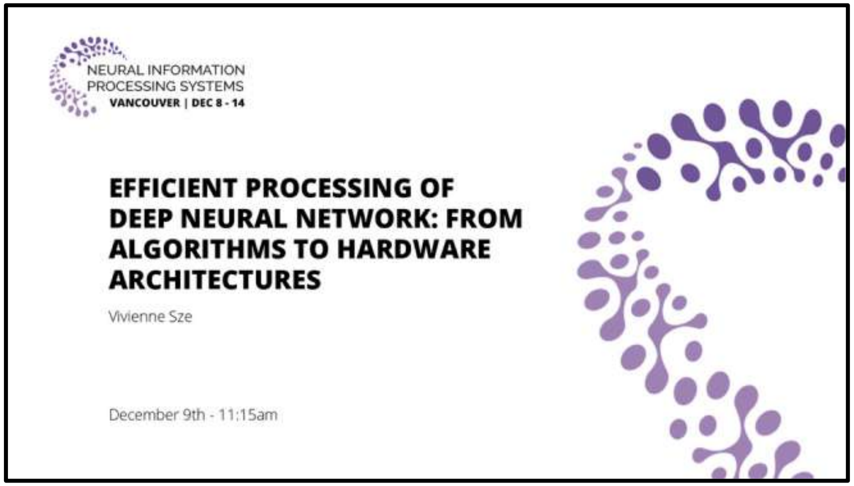
http://shortprograms.mit.edu/dls

Next Offering: July 20-21, 2020 on MIT Campus

Additional Resources

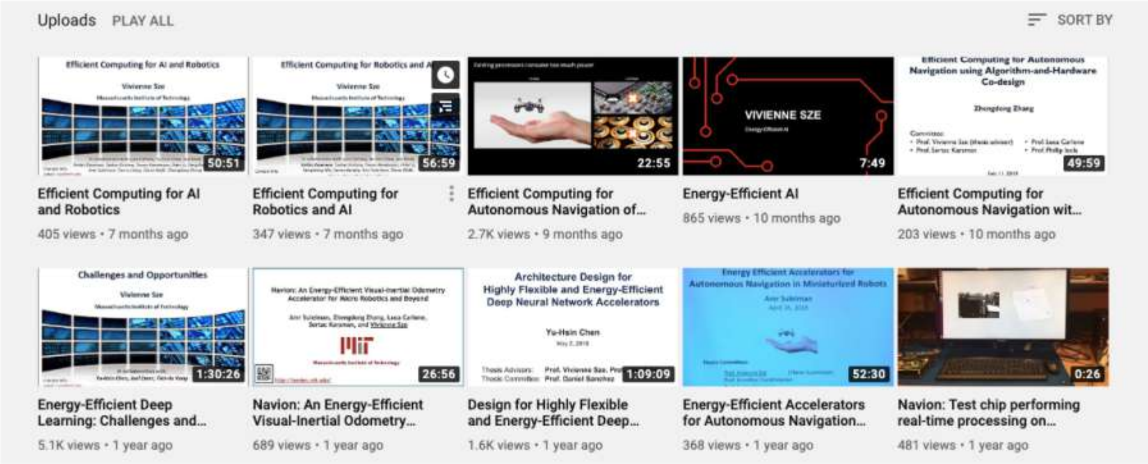
Talks and Tutorial Available Online

https://www.rle.mit.edu/eems/publications/tutorials/



YouTube Channel

EEMS Group – PI: Vivienne Sze



Acknowledgements

Joel Emer

Thomas Heldt

Sertac Karaman

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

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
- **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

References

- **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/*
- **Energy-Efficient Visual Inertial Localization**
  - **Project website:** *http://navion.mit.edu*
  - *A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Symposium on VLSI Circuits (VLSI-Circuits), June 2018.*
  - *Z. Zhang*, A. Suleiman*, L. Carlone, V. Sze, S. Karaman, "Visual-Inertial Odometry on Chip: An Algorithm-and-Hardware Co-design Approach," Robotics: Science and Systems (RSS), July 2017.*
  - *A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, V. Sze, "Navion: A 2mW Fully Integrated Real-Time Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," IEEE Journal of Solid State Circuits (JSSC), VLSI Symposia Special Issue, Vol. 54, No. 4, pp. 1106-1119, April 2019.*

References

- **Fast Shannon Mutual Information for Robot Exploration**
  - *Z. Zhang, T. Henderson, V. Sze, S. Karaman, "FSMI: Fast computation of Shannon Mutual Information for information-theoretic mapping," IEEE International Conference on Robotics and Automation (ICRA), May 2019.*
  - *P. Li*, Z. Zhang*, S. Karaman, V. Sze, "High-throughput Computation of Shannon Mutual Information on Chip," Robotics: Science and Systems (RSS), June 2019*
- **Low Power Time of Flight Imaging**
  - *J. Noraky, V. Sze, "Low Power Depth Estimation of Rigid Objects for Time-of-Flight Imaging," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2019.*
  - *J. Noraky, V. Sze, "Depth Estimation of Non-Rigid Objects For Time-Of-Flight Imaging," IEEE International Conference on Image Processing (ICIP), October 2018.*
  - *J. Noraky, V. Sze, "Low Power Depth Estimation for Time-of-Flight Imaging," IEEE International Conference on Image Processing (ICIP), September 2017.*
- **Monitoring Neurodegenerative Disorders Using a Phone**
  - *H.-Y. Lai, G. Saavedra Peña, C. Sodini, T. Heldt, V. Sze, "Enabling Saccade Latency Measurements with Consumer-Grade Cameras," IEEE International Conference on Image Processing (ICIP), October 2018.*
  - *G. Saavedra Peña, H.-Y. Lai, V. Sze, T. Heldt, "Determination of saccade latency distributions using video recordings from consumer-grade devices," IEEE International Engineering in Medicine and Biology Conference (EMBC), 2018.*
  - *H.-Y. Lai, G. Saavedra Peña, C. Sodini, V. Sze, T. Heldt, "Measuring Saccade Latency Using Smartphone Cameras," to appear in IEEE Journal of Biomedical and Health Informatics (JBHI)*