Energy-Efficient Edge Computing for AI-driven Applications

Vivienne Sze

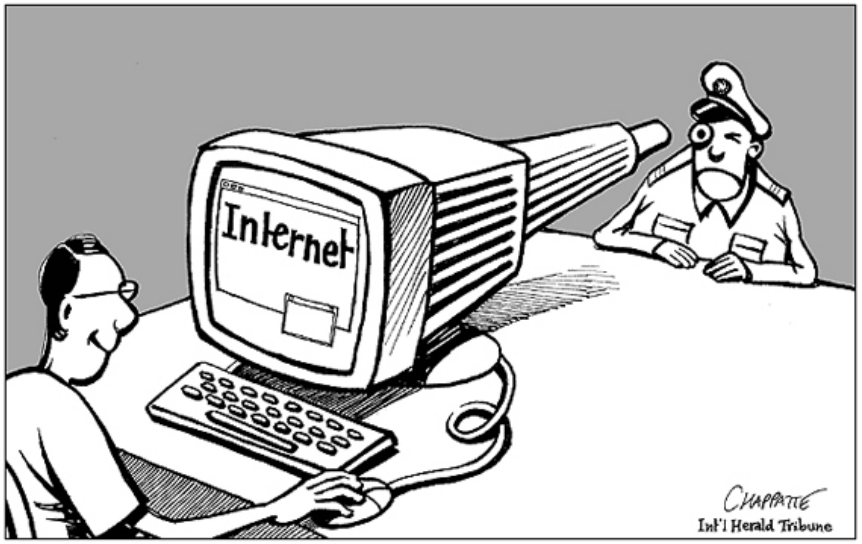
Massachusetts Institute of Technology

In collaboration with Luca Carlone, Yu-Hsin Chen, Joel Emer, Sertac Karaman, Tushar Krishna, Amr Suleiman, Tien-Ju Yang, Zhengdong Zhang

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

Processing at "Edge" instead of the "Cloud"

Privacy



Communication

Image source:
www.theregister.co.uk

Latency

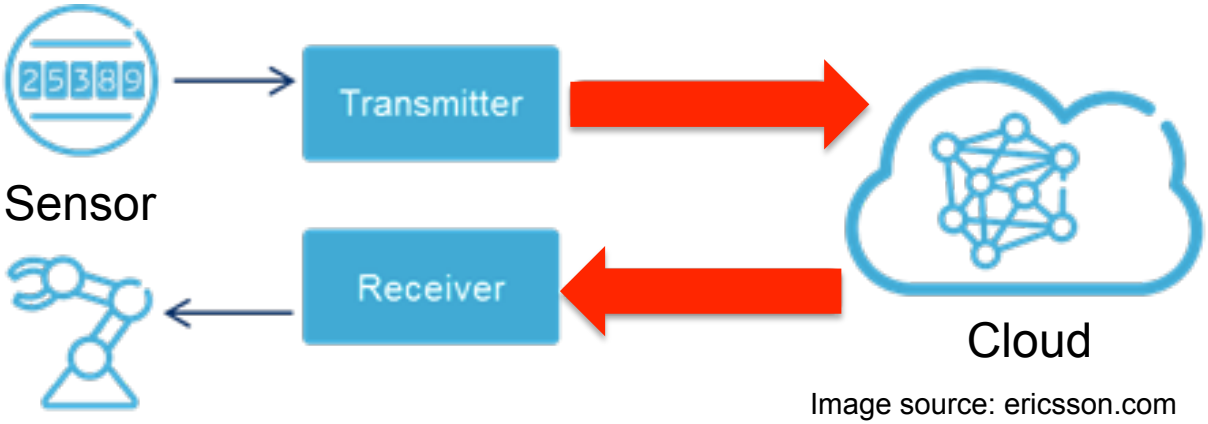
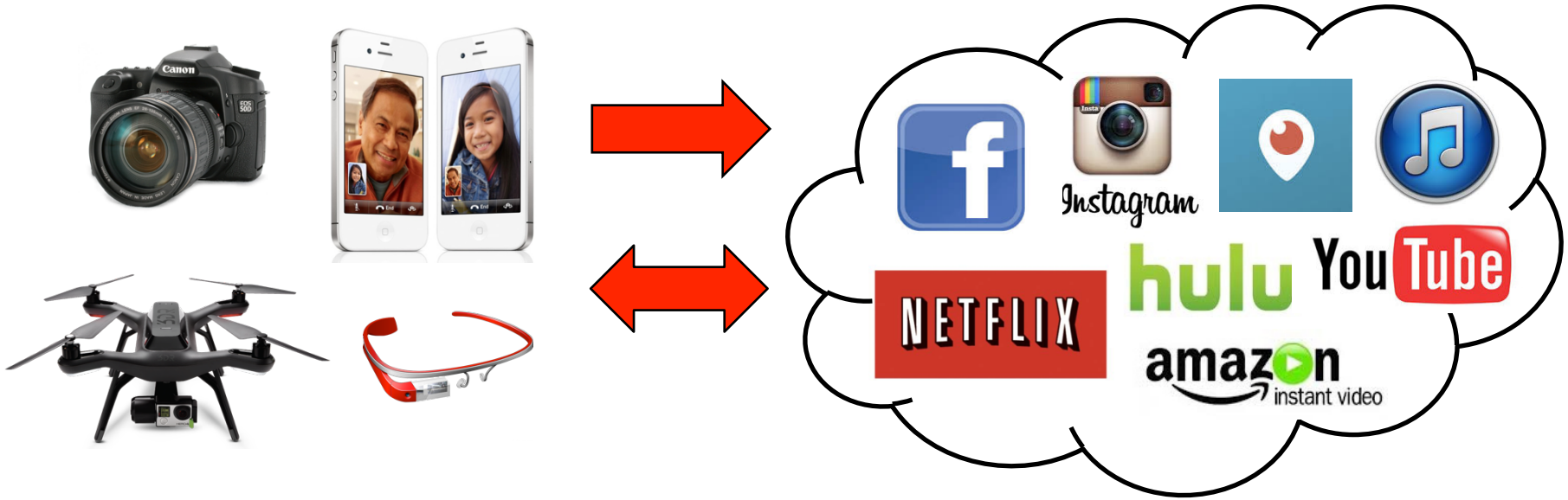


Image source: ericsson.com

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**



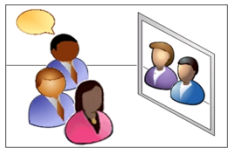
Energy limited due to battery capacity

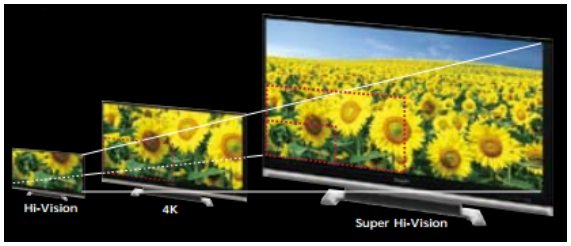
Power limited due to heat dissipation

Need energy-efficient pixel processing!

Energy-Efficient Pixel Processing

Next-Generation Video Coding (Compress Pixels)

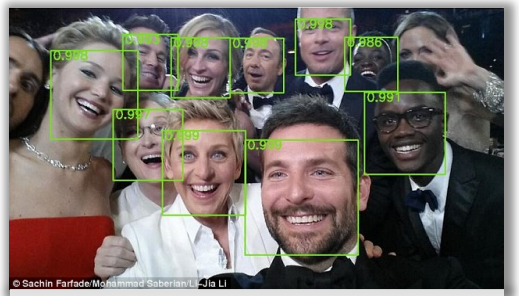




Ultra-HD

Goal: Increase coding efficiency, speed and energy-efficiency

Energy-Efficient Computer Vision & Deep Learning (Understand Pixels)



Recognition

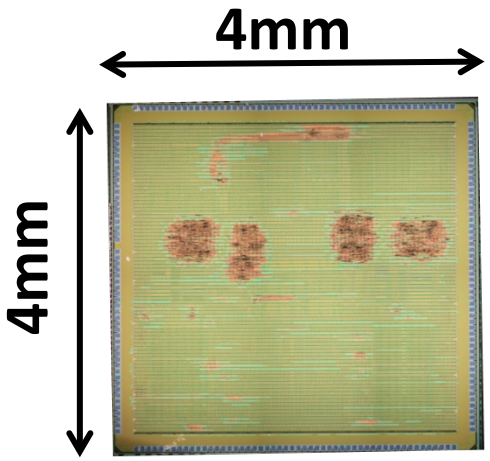
Self-Driving Cars

AI

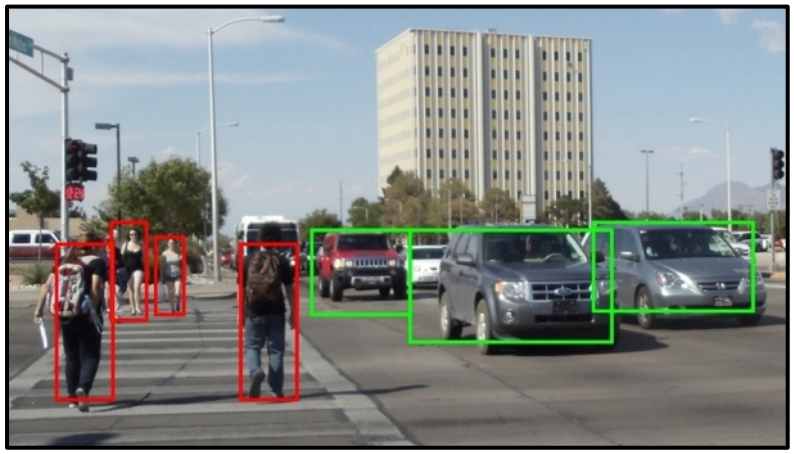
Goal: Make computer vision as ubiquitous as video coding

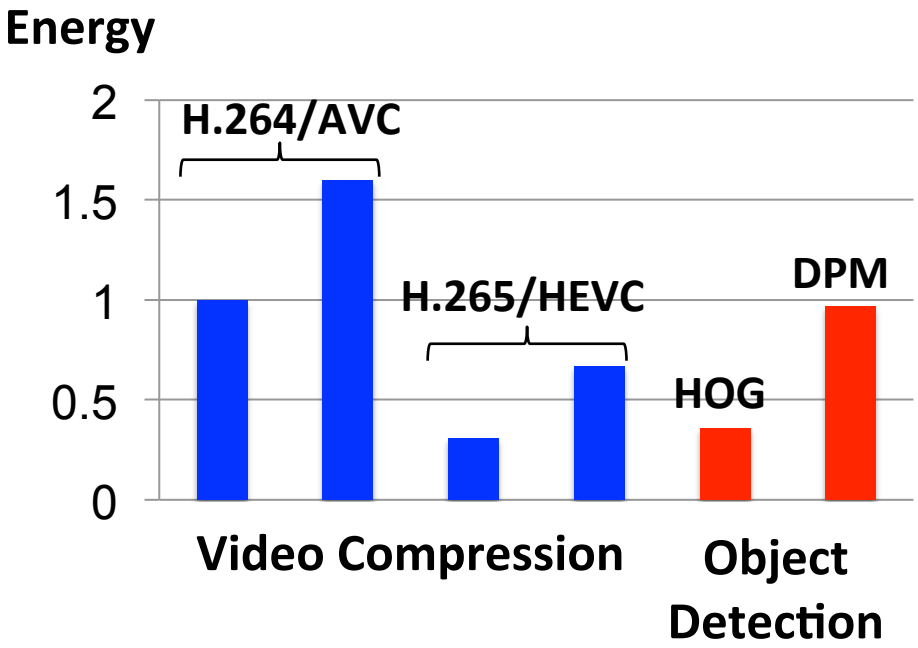
Typical Constraints of Video Coding

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



MIT Object Detection Chip [VLSI 2016]





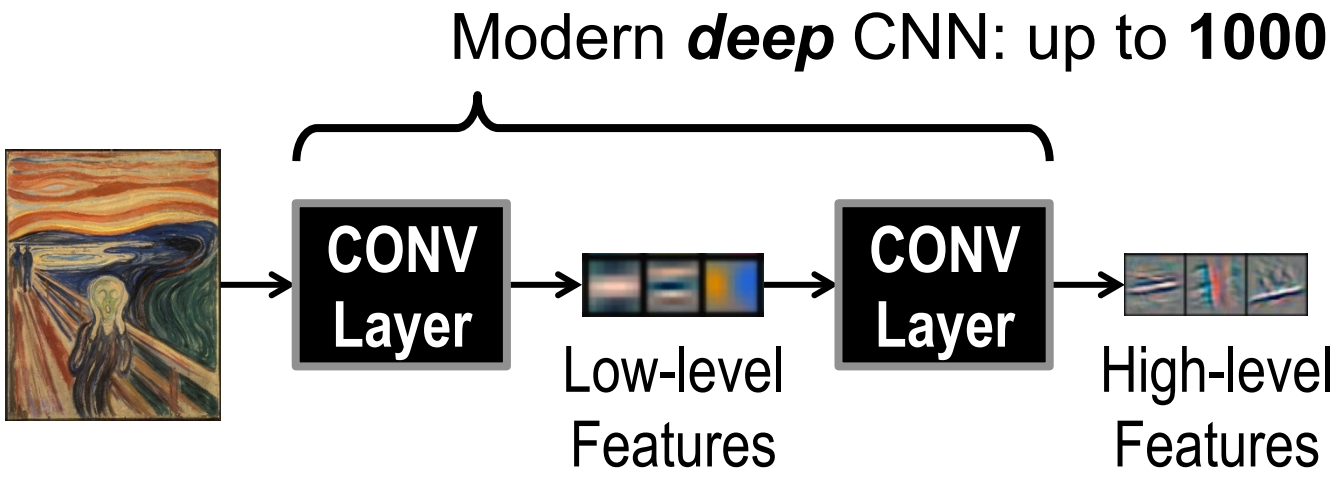
Outline

- **Energy-Efficient Hardware for Deep Neural Networks (DNNs)**
- **Limitations of Existing Efficient DNN Approaches**
- **Looking Beyond the DNN Accelerator for Acceleration**
- **Looking Beyond DNNs: Other forms of inference at the edge**

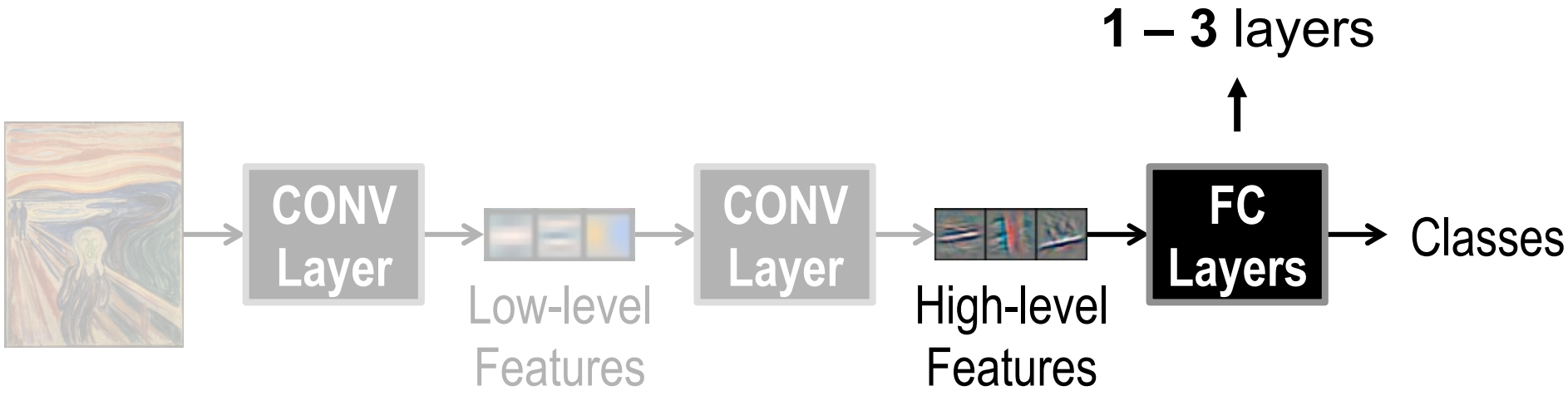
Energy-Efficient Hardware for Deep Neural Networks

Y.-H. Chen, T. Krishna, J. Emer, V. Sze,
"**Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks**," *JSSC* 2017.

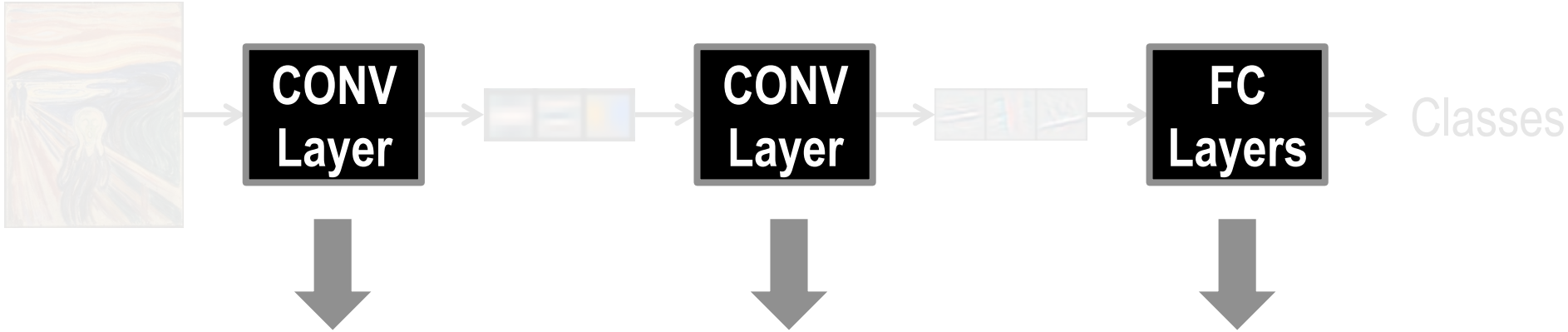
Deep Convolutional Neural Networks

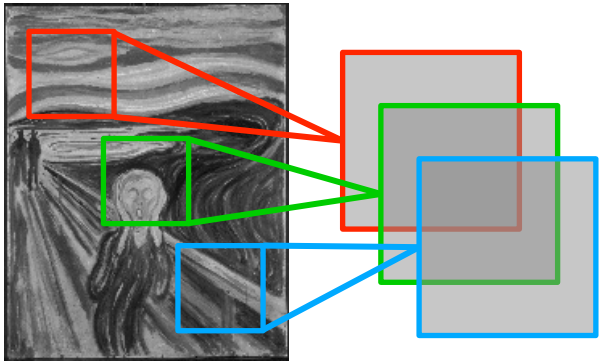


Deep Convolutional Neural Networks



Deep Convolutional Neural Networks



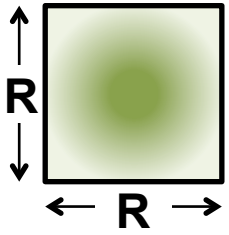


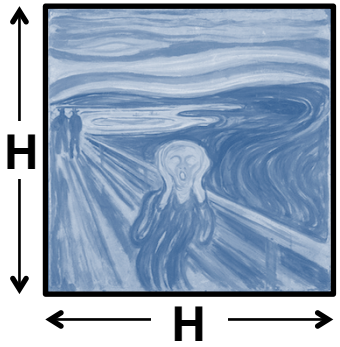
Convolutions account for more than 90% of overall computation, dominating **runtime** and **energy consumption**

High-Dimensional CNN Convolution

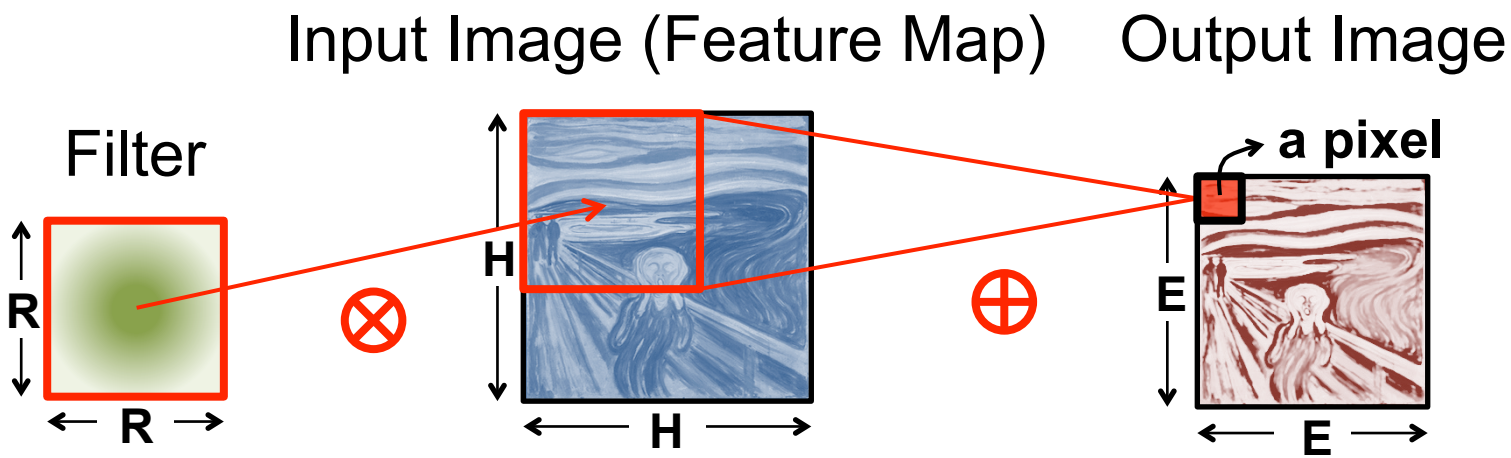
Input Image (Feature Map)

Filter

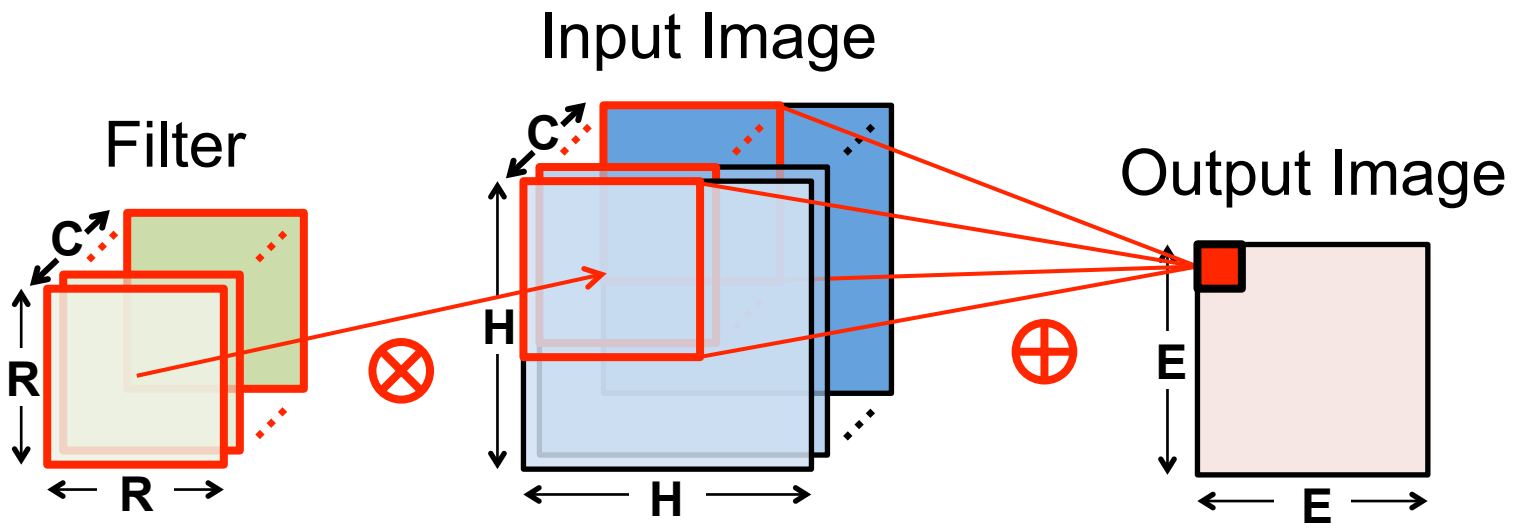




High-Dimensional CNN Convolution



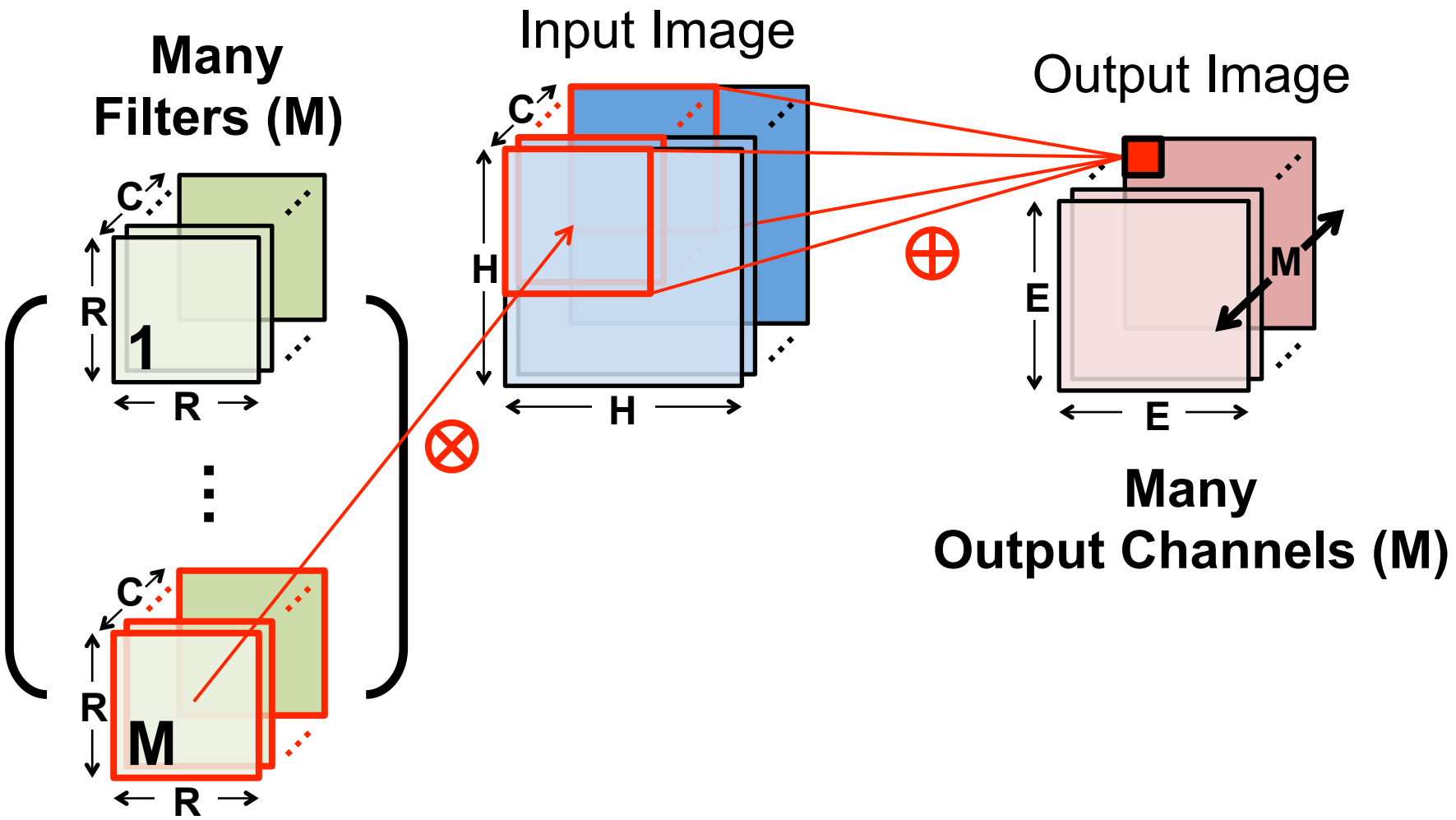
High-Dimensional CNN Convolution



Many Input Channels (C)

AlexNet: 3 – 192 Channels (C)

High-Dimensional CNN Convolution



AlexNet: 96 – 384 Filters (M)

High-Dimensional CNN Convolution

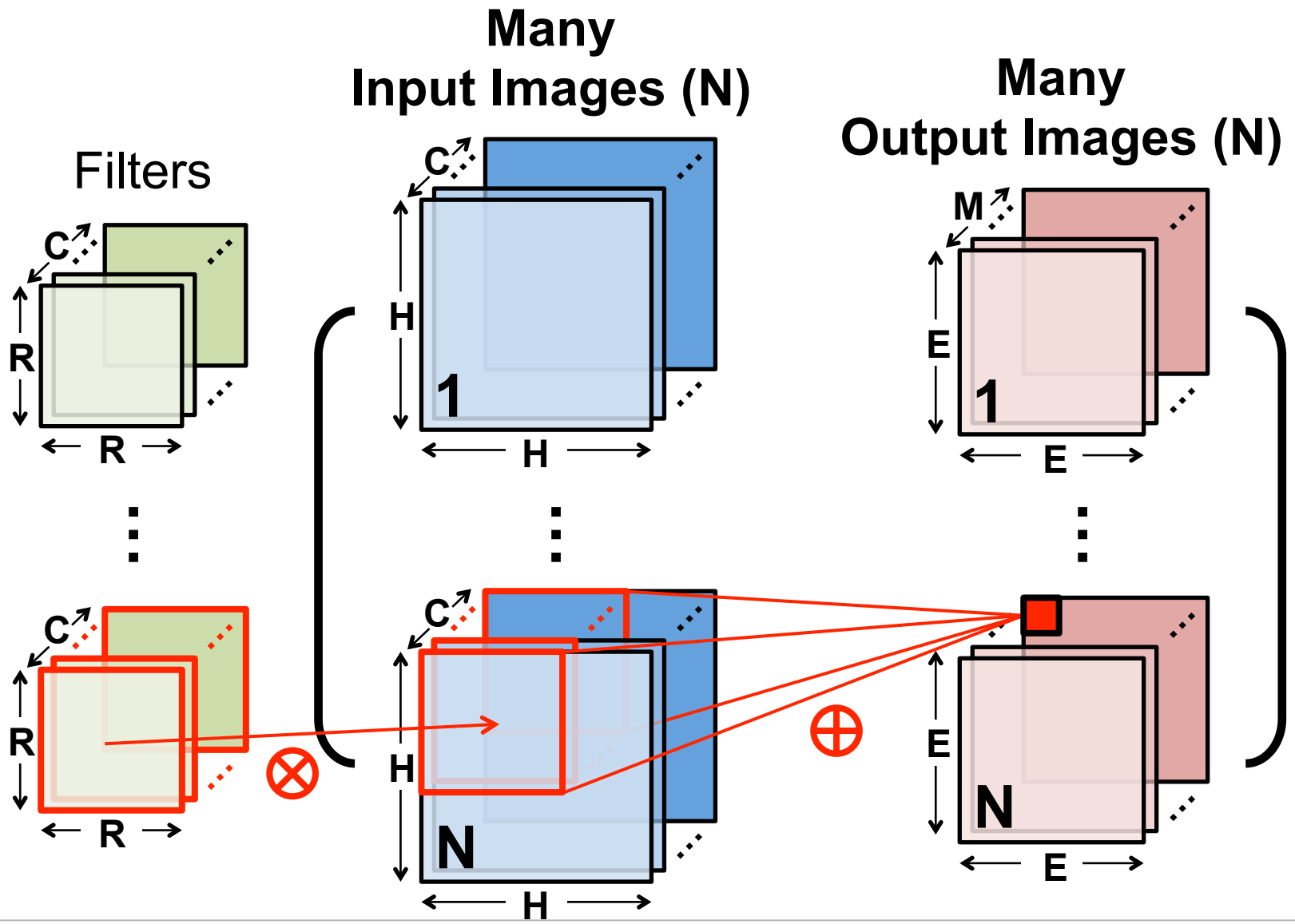
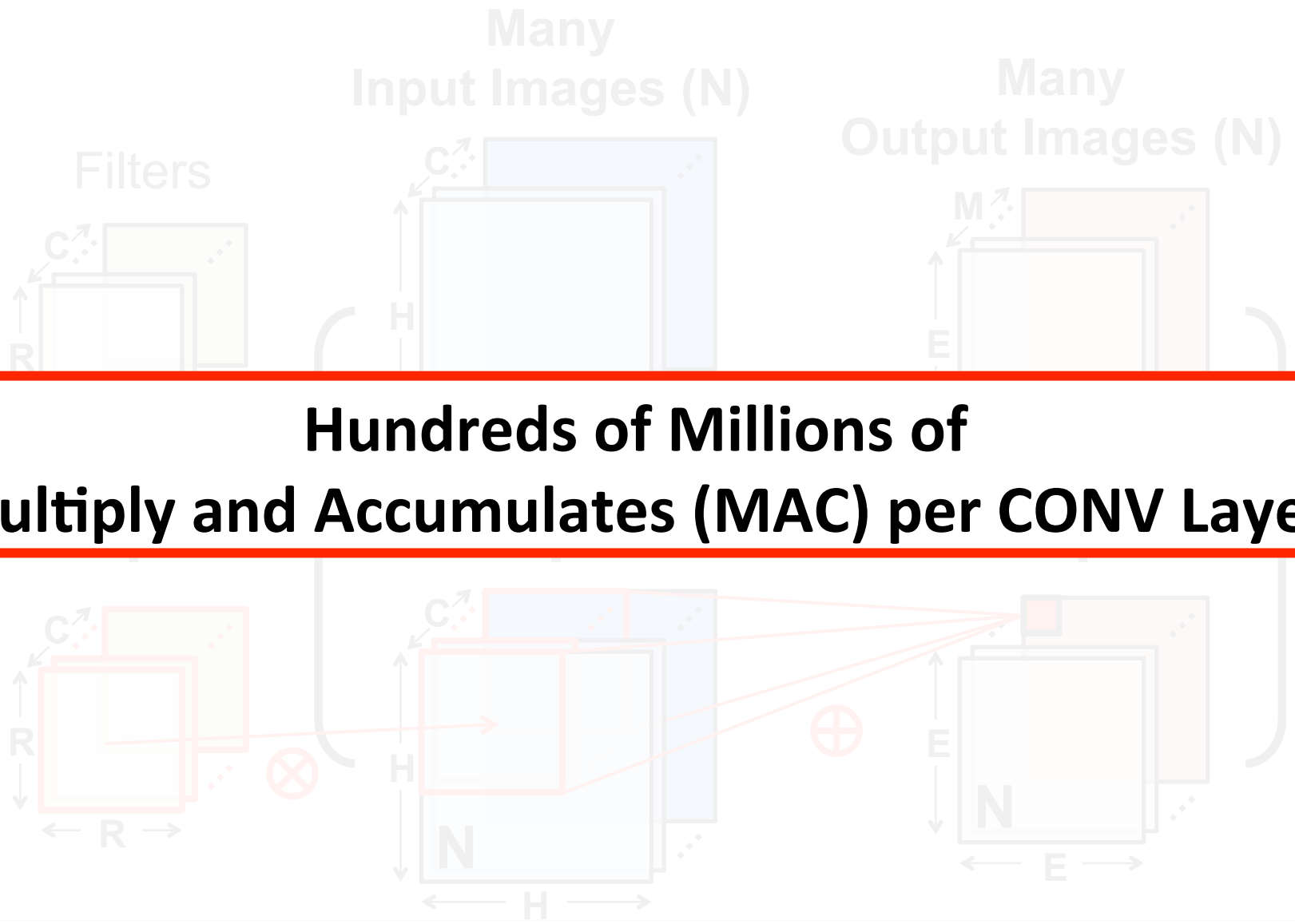


Image batch size: 1 – 256 (N)

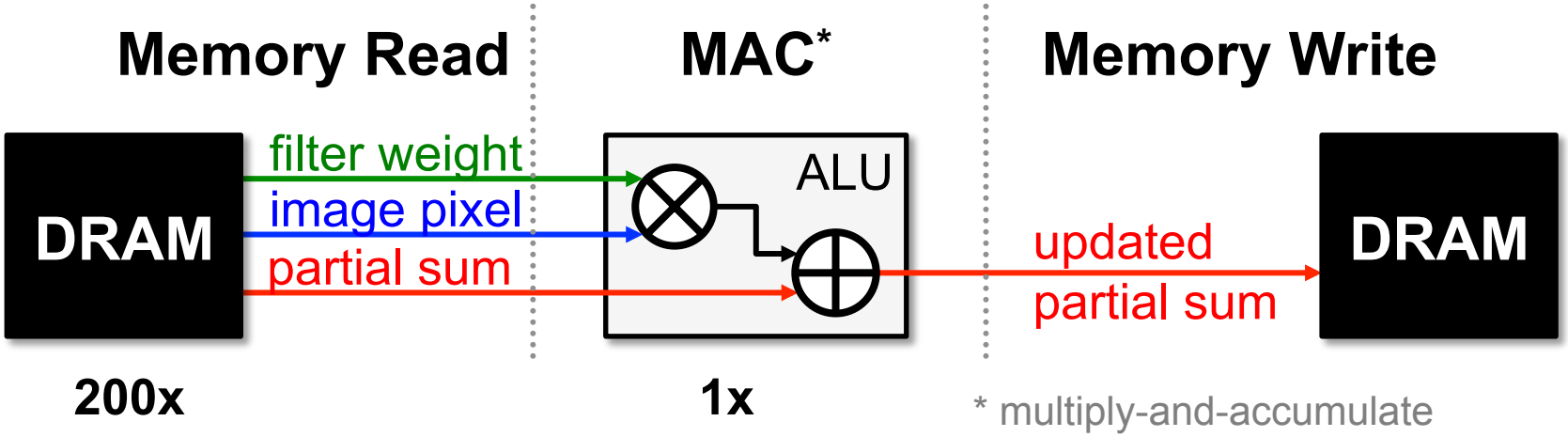
High-Dimensional CNN Convolution



Hundreds of Millions of
Multiply and Accumulates (MAC) per CONV Layer

Properties We Can Leverage

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

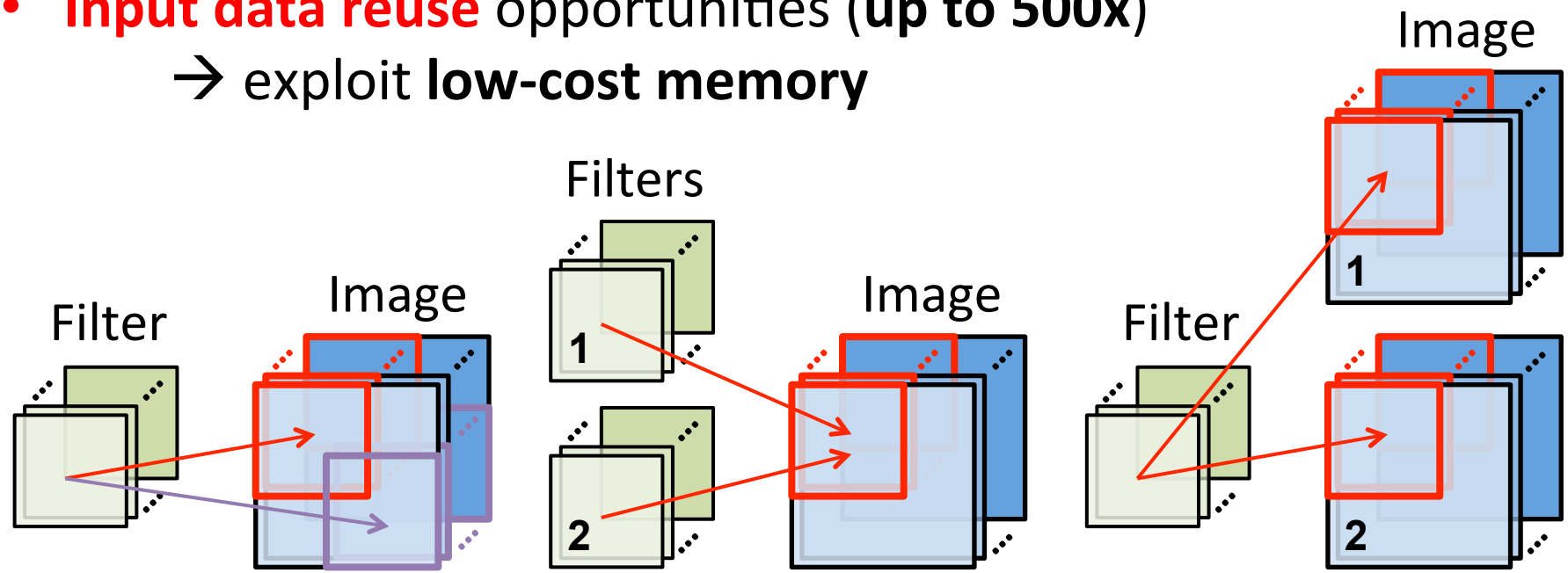


Worst Case: all memory R/W are **DRAM** accesses

- Example: AlexNet [NIPS 2012] 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**)
  → exploit **low-cost memory**



Convolutional Reuse (pixels, weights)

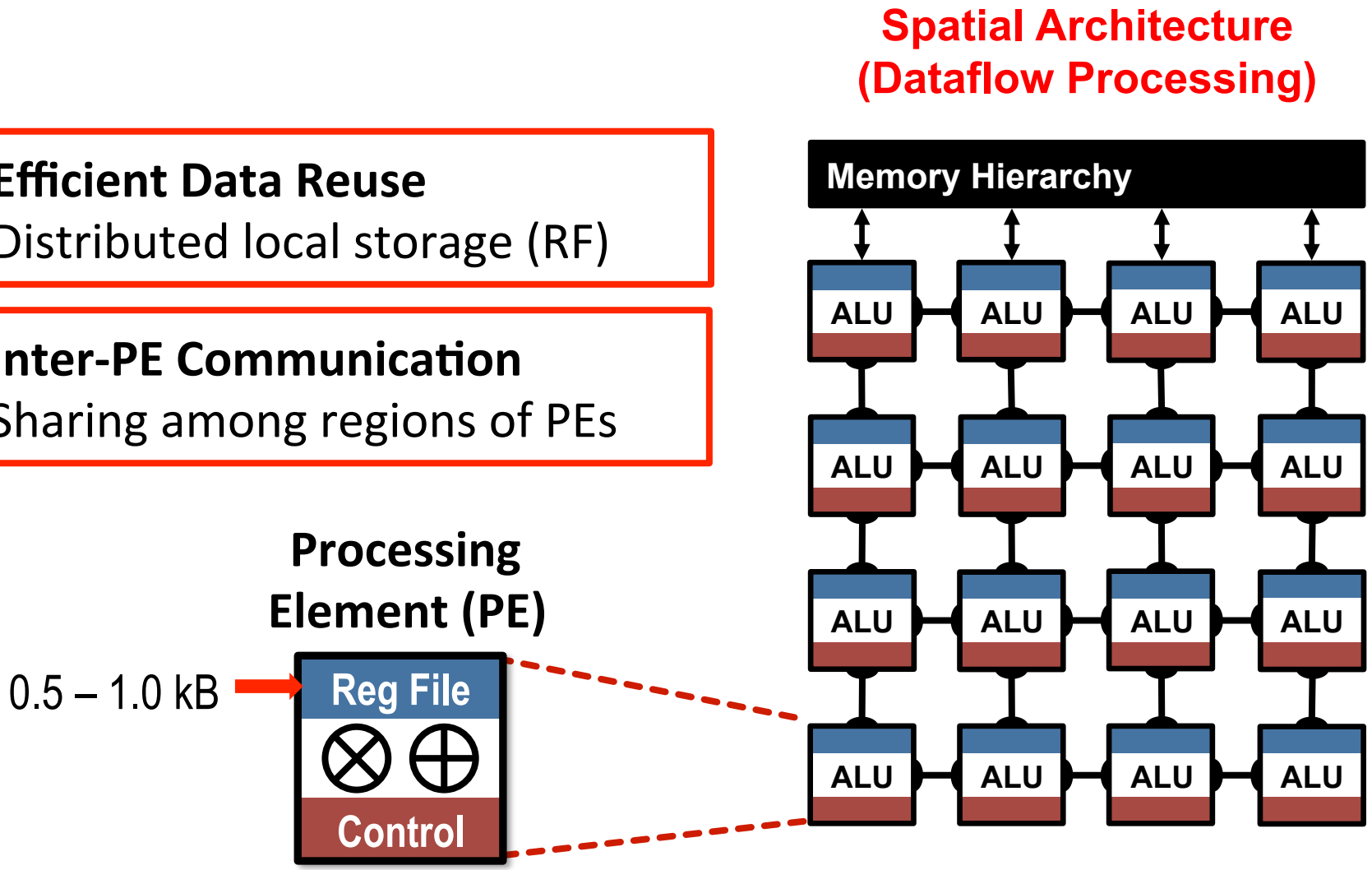
Image Reuse (pixels)

Filter Reuse (weights)

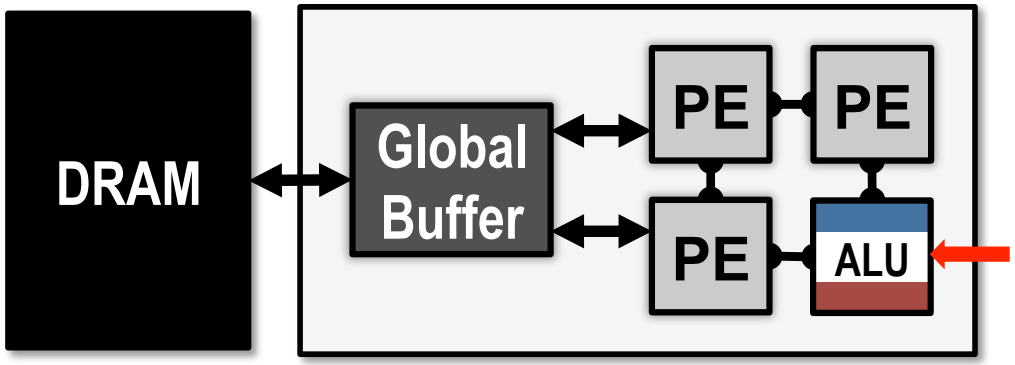
Advantages of Spatial Architecture

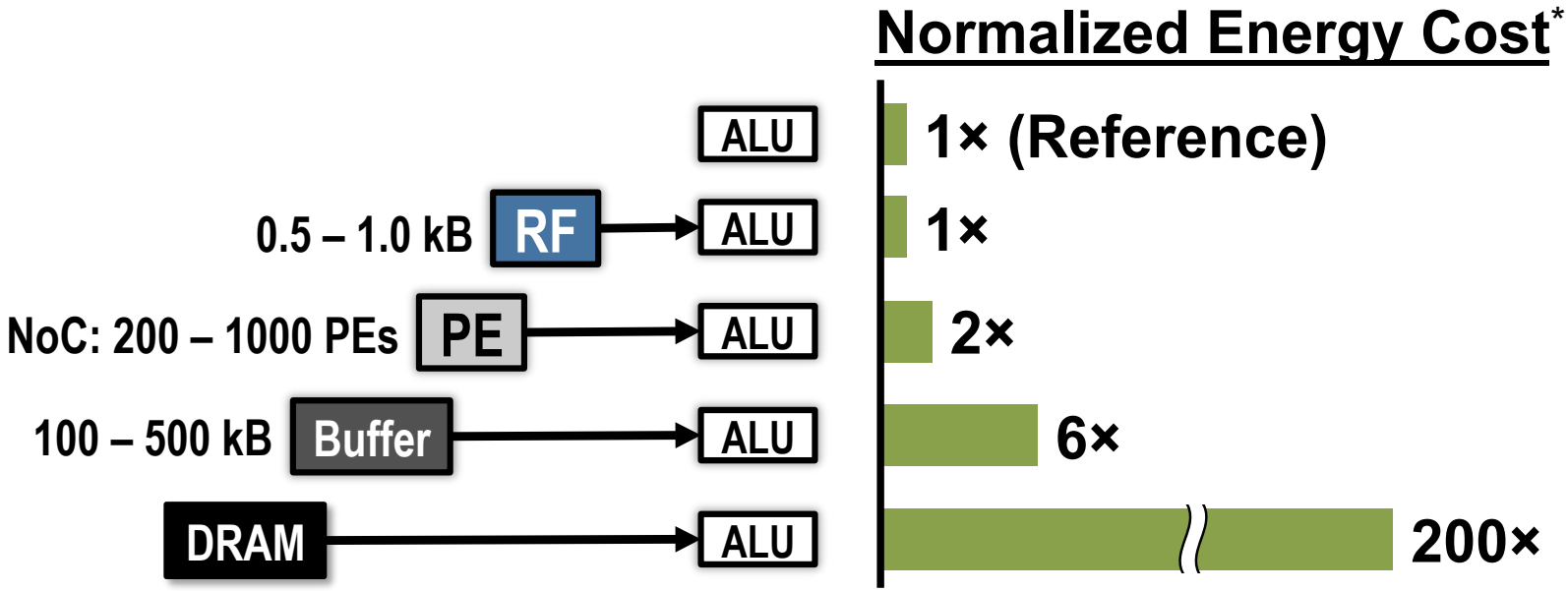
Efficient Data Reuse
Distributed local storage (RF)

Inter-PE Communication
Sharing among regions of PEs



How to Map the Dataflow?

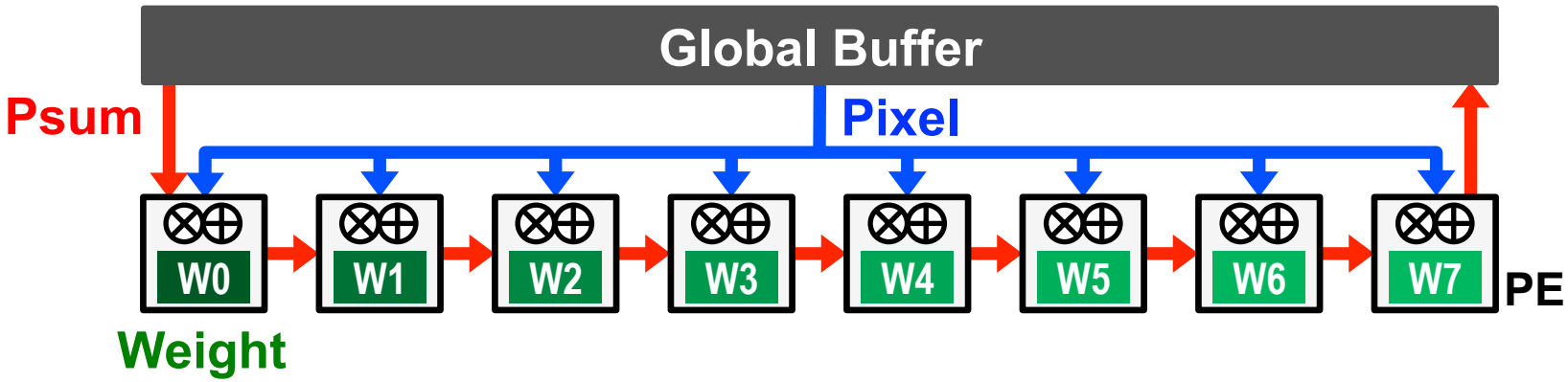




* measured from a commercial 65nm process

Maximize data reuse at lower levels of hierarchy

Weight Stationary (WS)

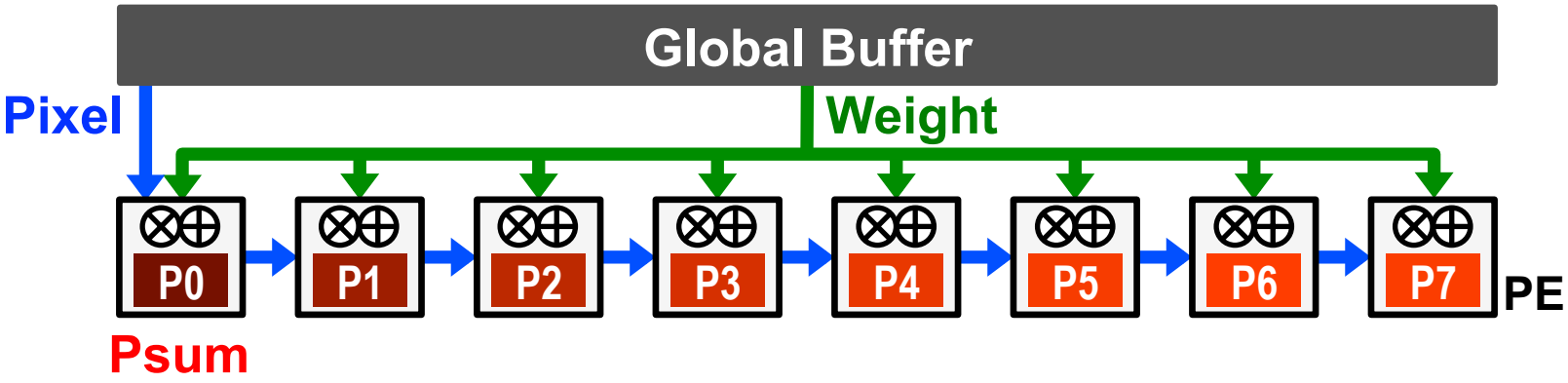


- **Minimize weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Examples:**

[**Chakradhar**, *ISCA* 2010] [**nn-X (NeuFlow)**, *CVPRW* 2014]

[**Park**, *ISSCC* 2015] [**Origami**, *GLSVLSI* 2015]

Output Stationary (OS)



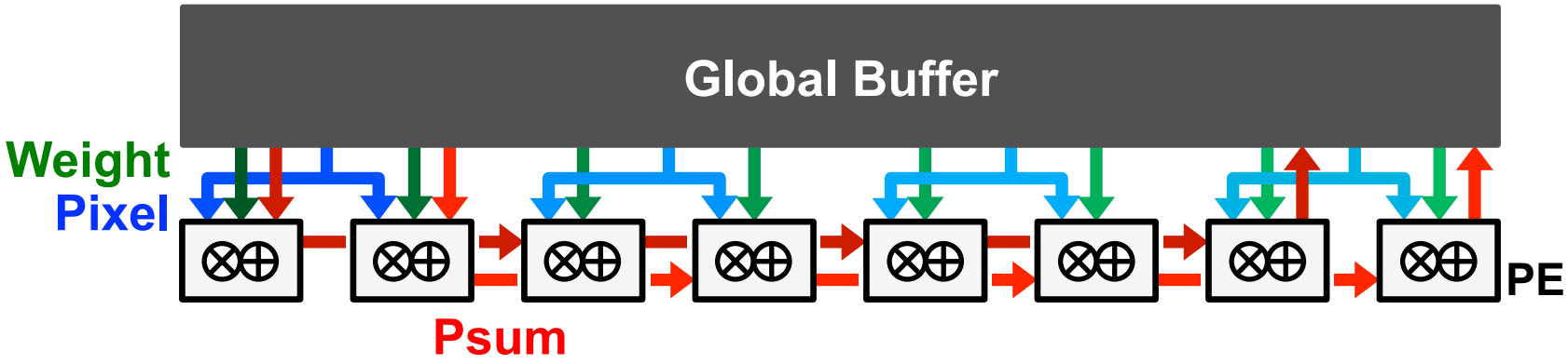
- **Minimize partial sum** R/W energy consumption
  - maximize local accumulation
- **Examples:**

  [**Gupta**, *ICML* 2015]

  [**Peemen**, *ICCD* 2013]

  [**ShiDianNao**, *ISCA* 2015]

No Local Reuse (NLR)

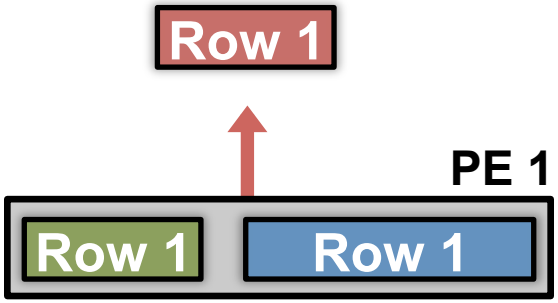


- Use a **large global buffer** as shared storage
  - Reduce **DRAM** access energy consumption
- **Examples:**

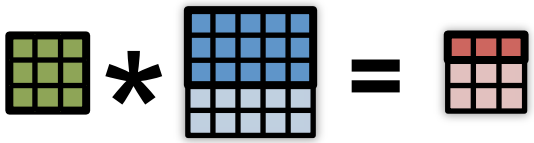
  [**DianNao**, *ASPLOS* 2014] [**DaDianNao**, *MICRO* 2014]

  [**Zhang**, *FPGA* 2015]

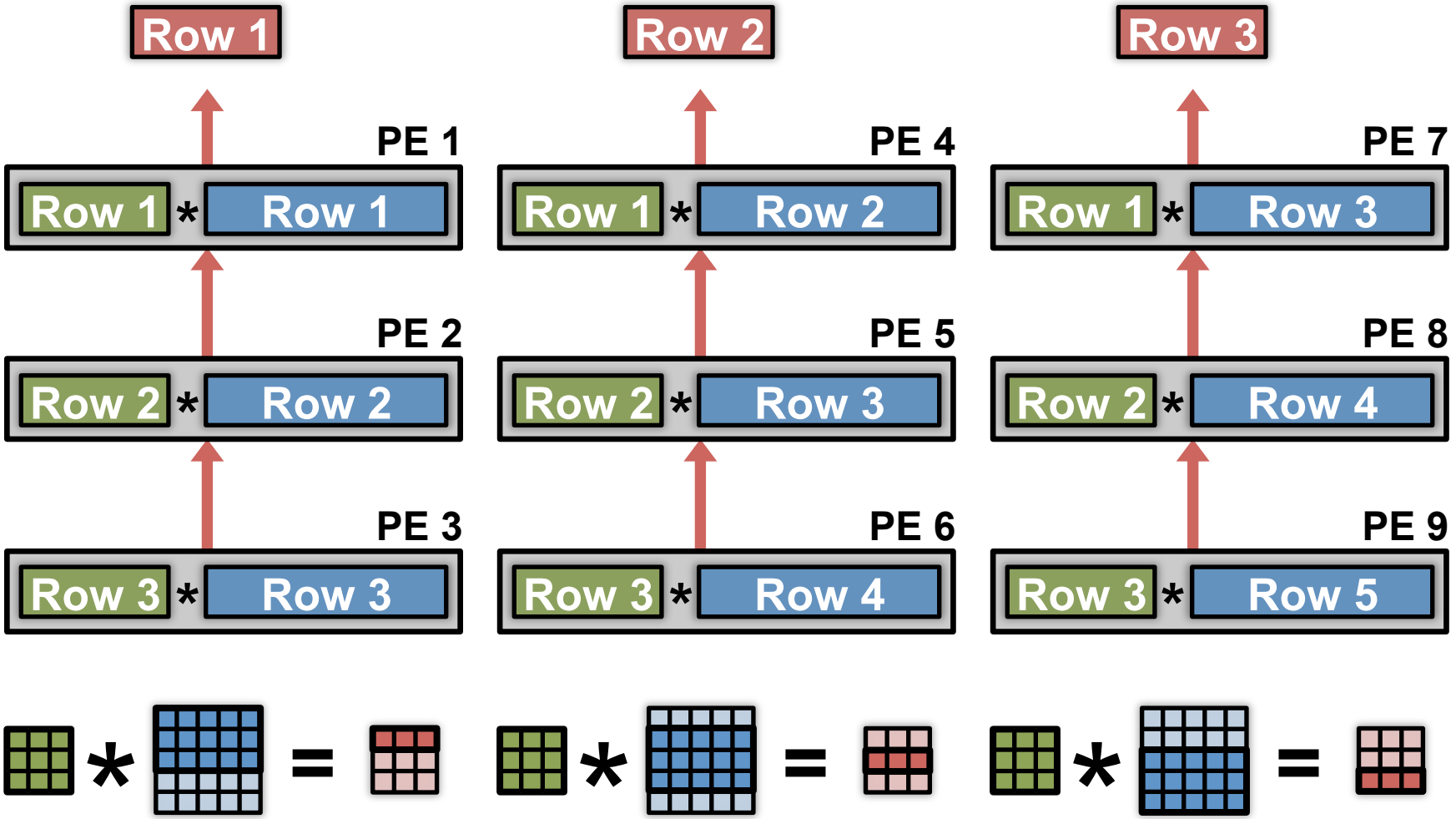
Row Stationary Dataflow



- 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



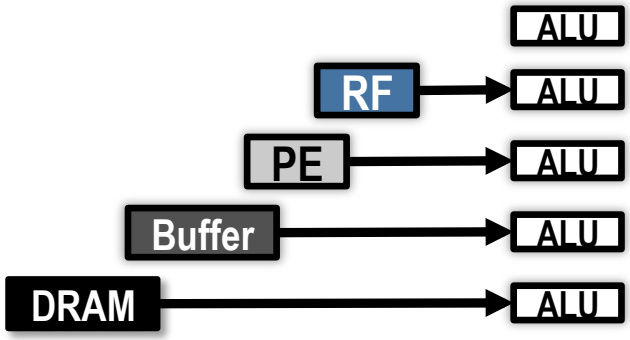
Optimize for **overall energy efficiency** instead for only a certain data type

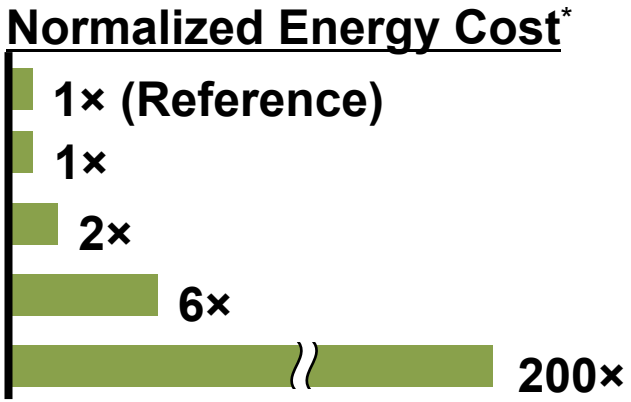
Evaluate Reuse in Different Dataflows

- **Weight Stationary**
  - Minimize movement of filter weights
- **Output Stationary**
  - Minimize movement of partial sums
- **No Local Reuse**
  - Don't use any local PE storage. Maximize global buffer size.
- **Row Stationary**

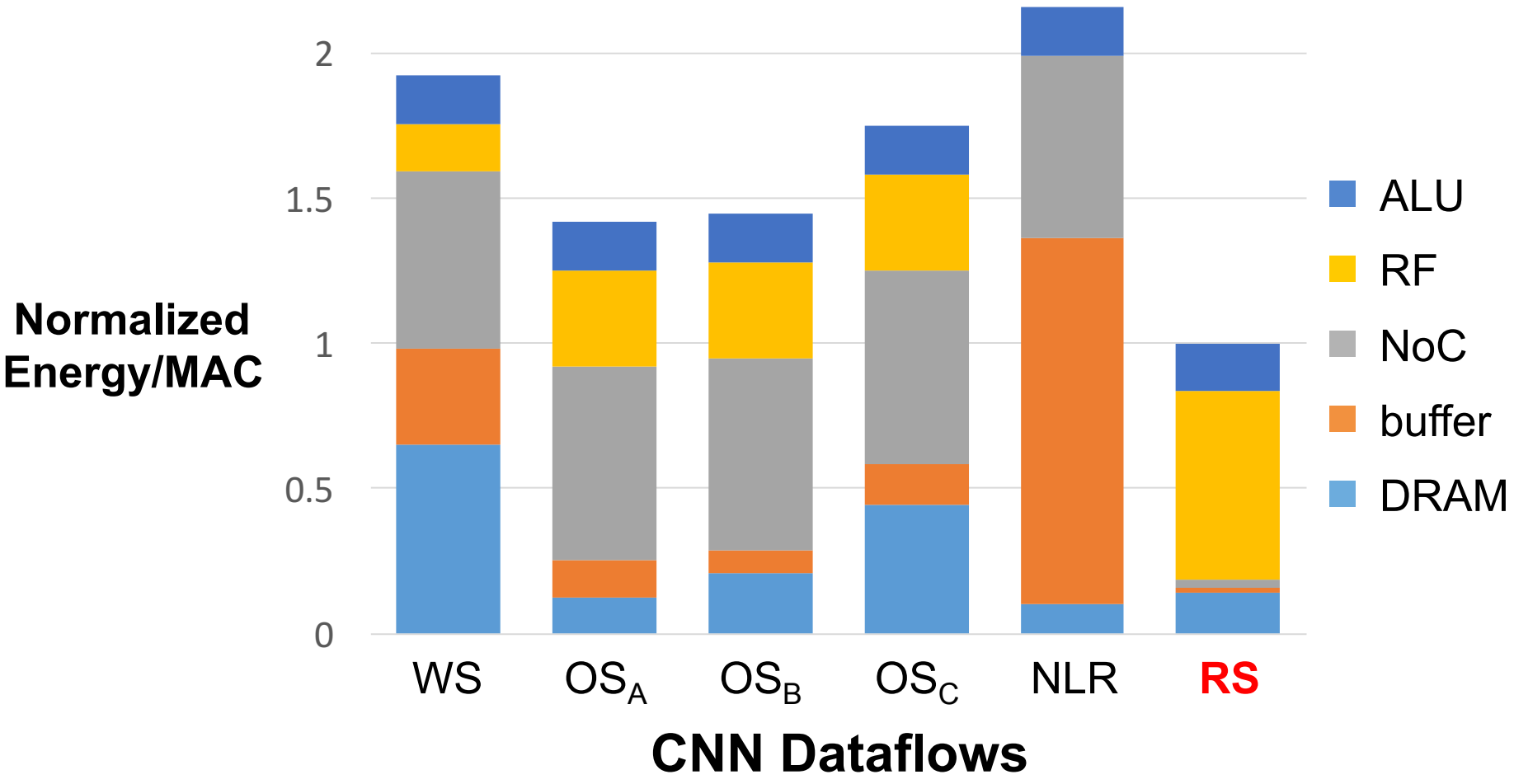
Evaluation Setup

- Same Total Area
- AlexNet
- 256 PEs
- Batch size = 16





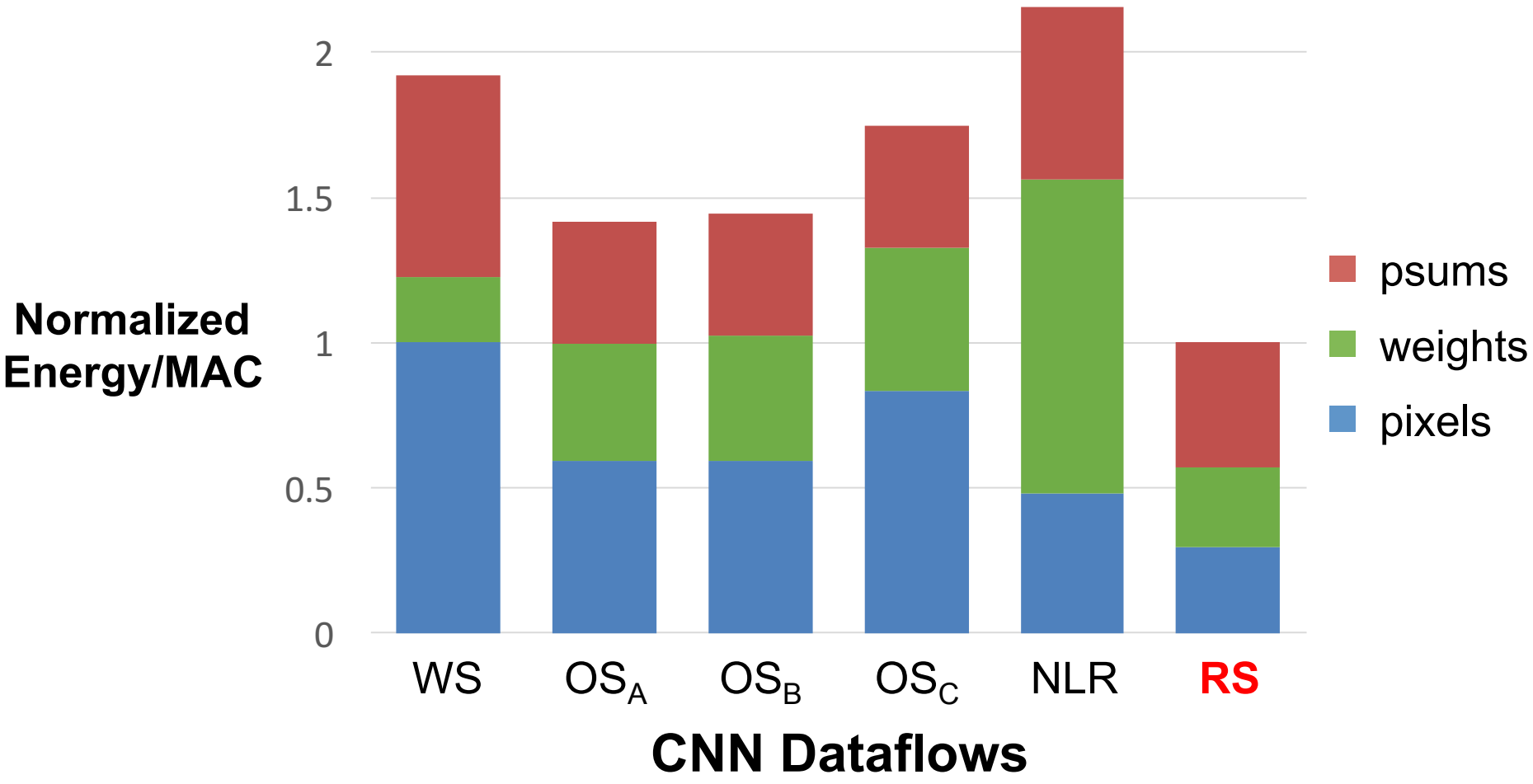
Dataflow Comparison: CONV Layers



RS uses **1.4× – 2.5× lower** energy than other dataflows

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

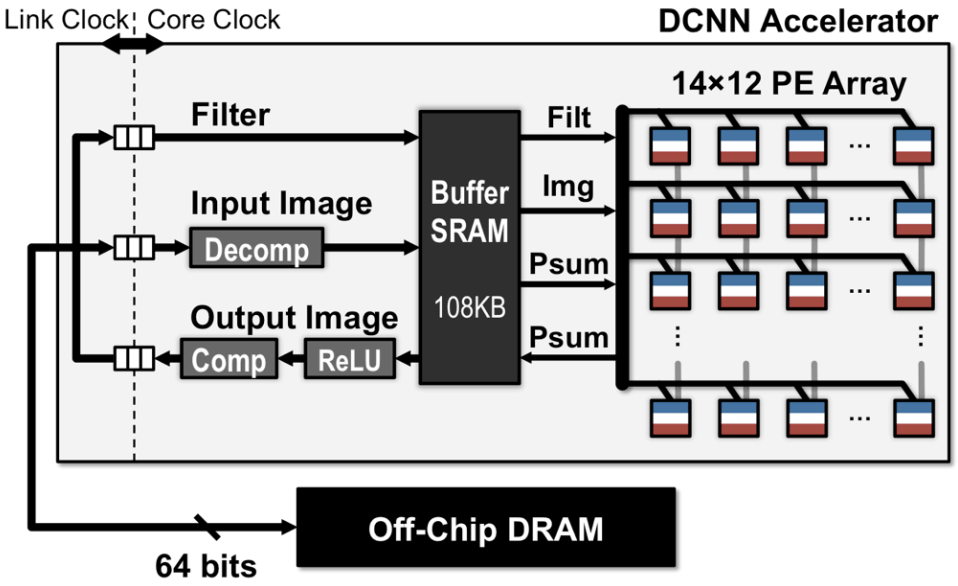
Dataflow Comparison: CONV Layers

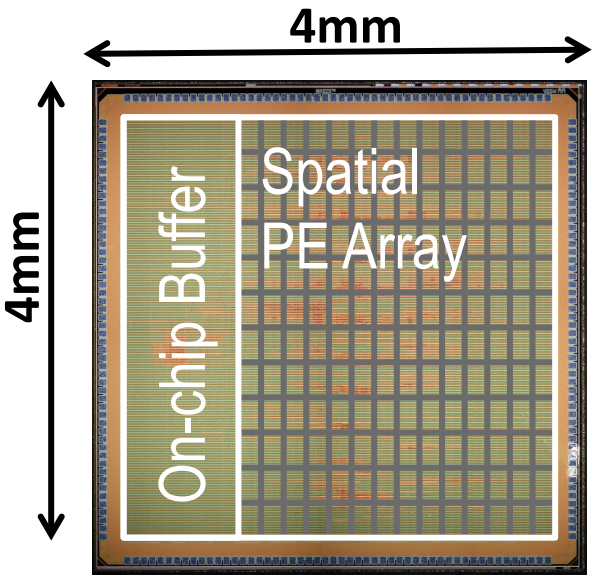


RS optimizes for the best **overall** energy efficiency

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

Eyeriss: Energy-Efficient Deep Learning

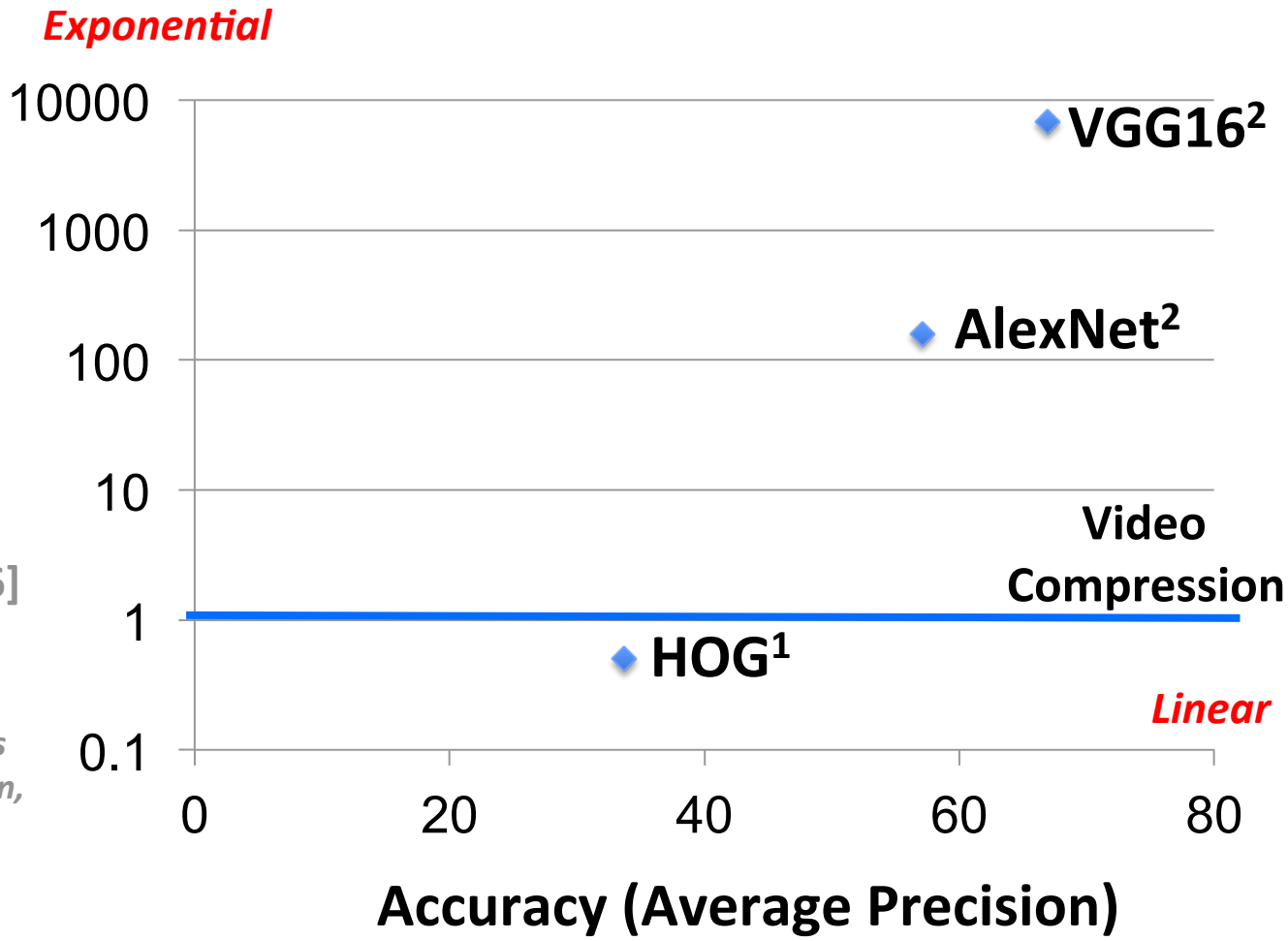




MIT Deep Learning Chip
[ISSCC 2016, ISCA 2016]

AlexNet: For 2.66 GMACs [8 billion 16-bit inputs (**16GB**) and 2.7 billion outputs (**5.4GB**)], only requires a total of **208.5MB** reads/writes from on-chip 108kB global buffer and **15.4MB** reads/writes from off-chip DRAM

Features: Energy vs. Accuracy



*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]

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

Hardware Architectures for Deep Neural Networks

ISCA Tutorial

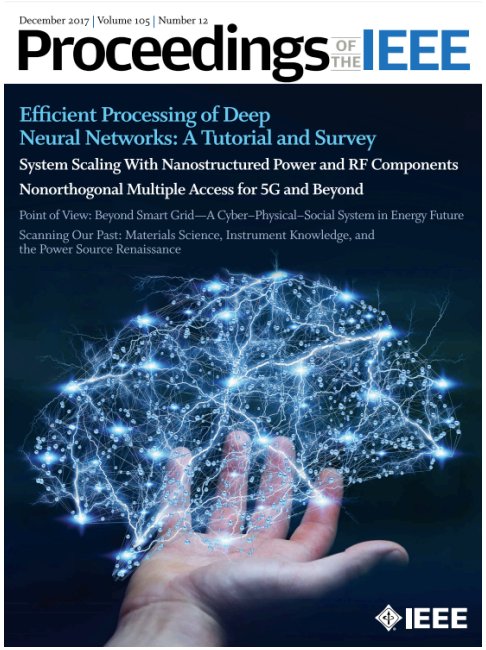
June 24, 2017

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

Massachusetts Institute of Technology

NVIDIA

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



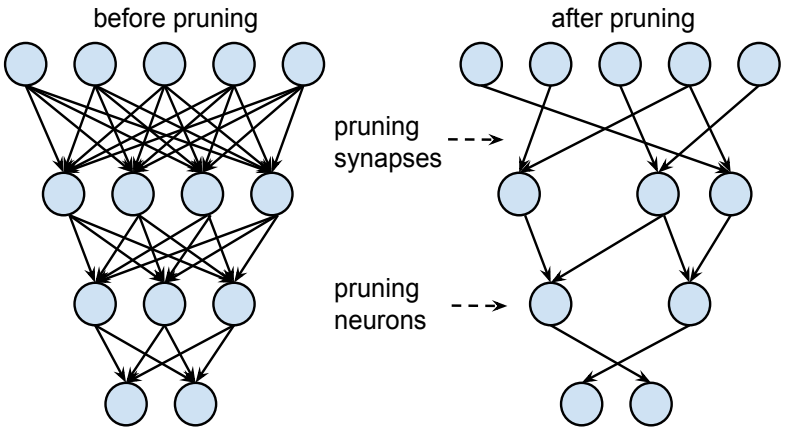
V. Sze, Y.-H. Chen, T-J. Yang, J. Emer, "***Efficient Processing of Deep Neural Networks: A Tutorial and Survey***," Proceedings of the IEEE, Dec. 2017

We identified various limitations to existing approaches

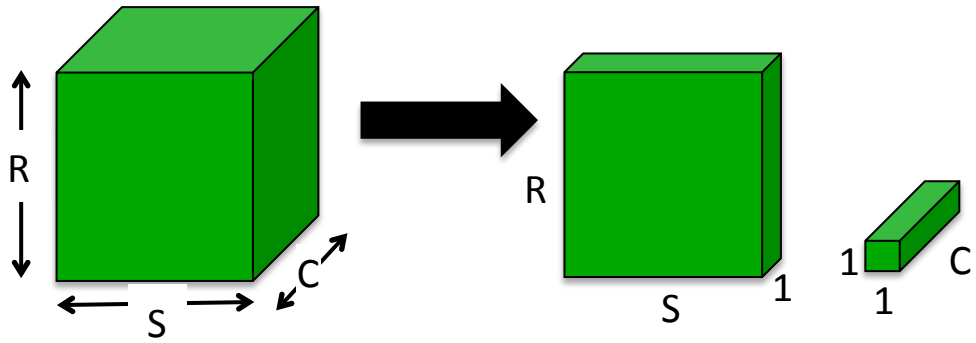
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures

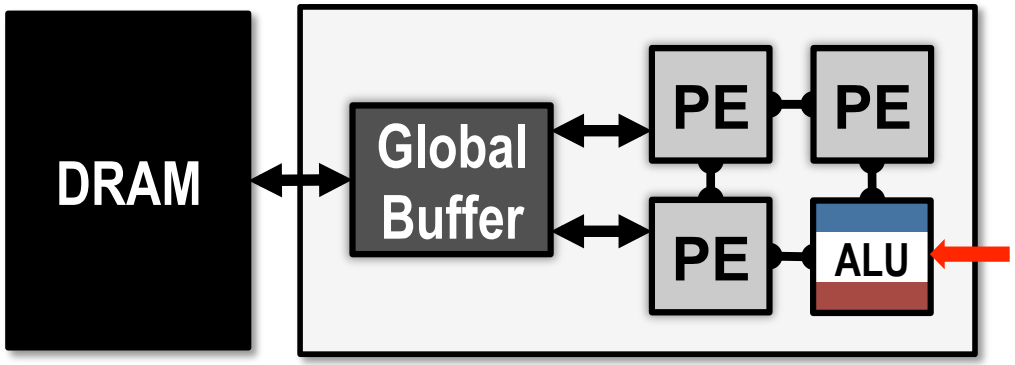


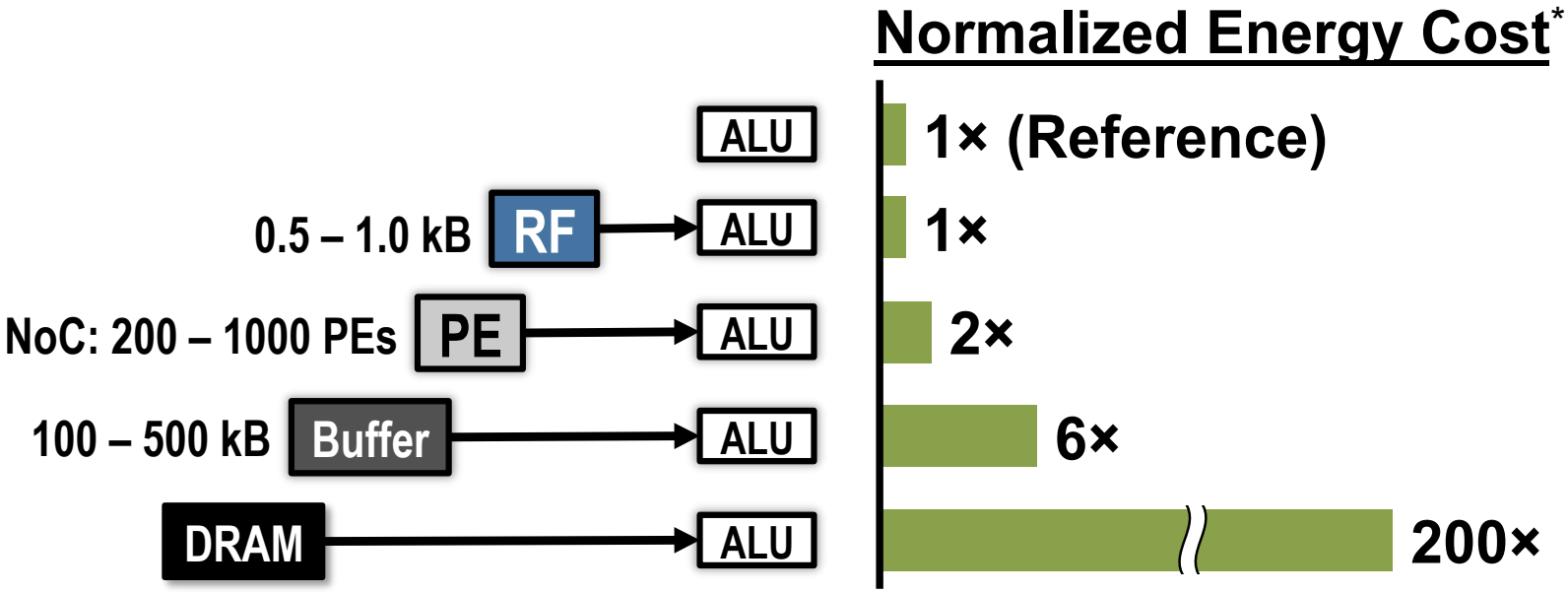
Examples: SqueezeNet, MobileNet

... also reduced precision

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

Data Movement is Expensive

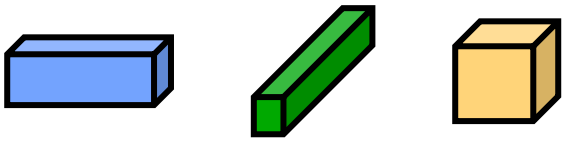




* measured from a commercial 65nm process

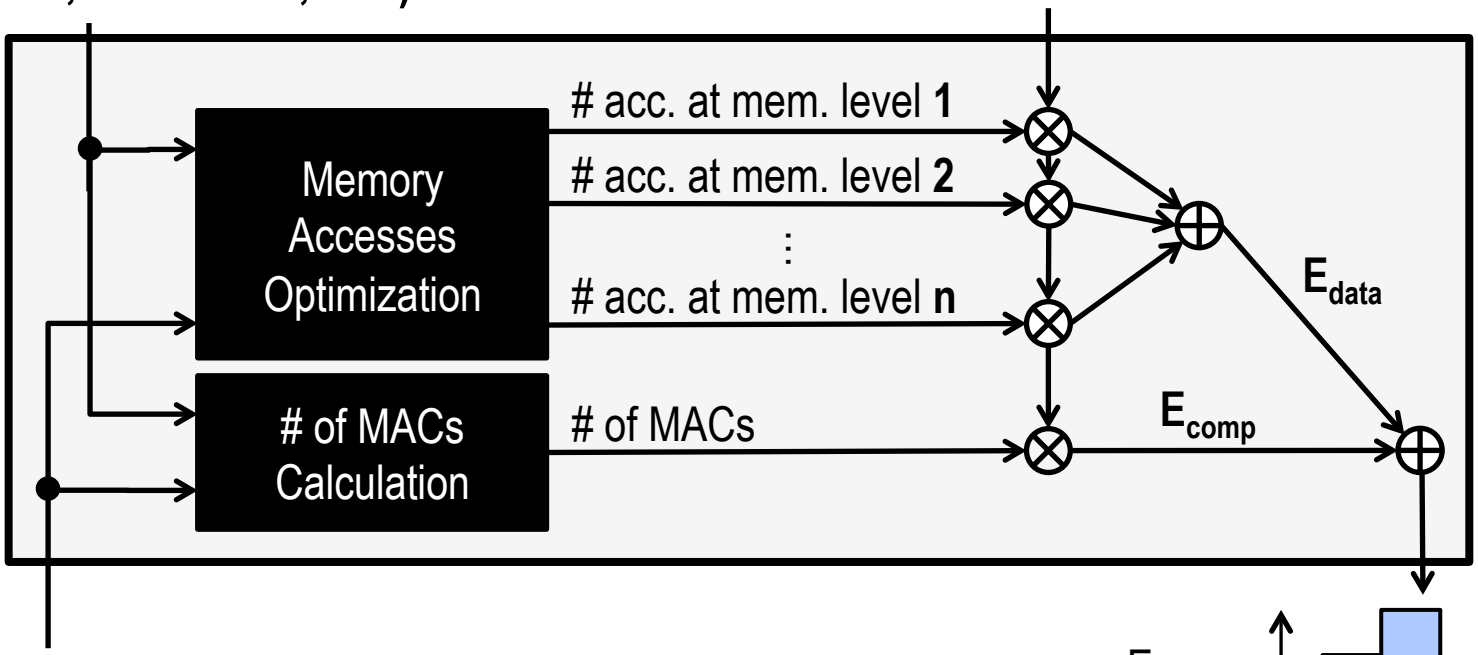
Energy of weight depends on **memory hierarchy** and **dataflow**

Energy-Evaluation Methodology



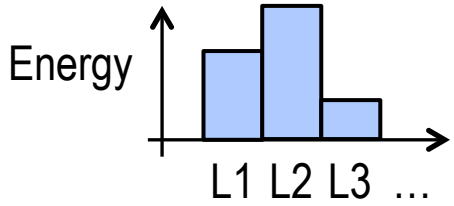
CNN Shape Configuration
(# of channels, # of filters, etc.)

Hardware Energy Costs of each MAC and Memory Access



CNN Weights and Input Data

[0.3, 0, -0.4, 0.7, 0, 0, 0.1, …]



CNN Energy Consumption

Energy Estimation Tool

Website: https://energyestimation.mit.edu/

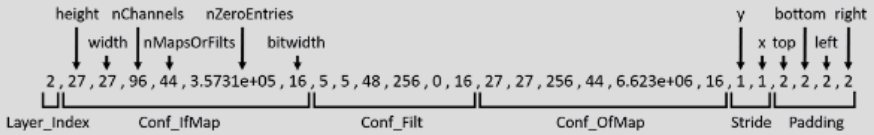
Deep Neural Network Energy Estimation Tool

Overview

This Deep Neural Network Energy Estimation Tool is used for evaluating and designing energy-efficient deep neural networks that are critical for embedded deep learning processing. Energy estimation was used in the development of the energy-aware pruning method (Yang et al., CVPR 2017), which reduced the energy consumption of AlexNet and GoogLeNet by 3.7x and 1.6x, respectively, with less than 1% top-5 accuracy loss. This website provides a simplified version of the energy estimation tool for shorter runtime (around 10 seconds).

Input

To support the variety of toolboxes, this tool takes a single network configuration file. The network configuration file is a txt file, where each line denotes the configuration of a CONV/FC layer. The format of each line is:



- Layer Index: the index of the layer, from 1 to the number of layers. It should be the same as the line number.
- Conf_IfMap, Conf_Filt, Conf_OfMap: the configuration of the input feature maps, the filters and the output feature maps. The configuration of each of the three data types is in the format of "height width number_of_channels number_of_maps_or_filts number_of_zero_entries bitwidth_in_bits".
- Stride: the stride of this layer. It is in the format of "stride_y stride_x".
- Pad: the amount of input padding. It is in the format of "pad_top pad_bottom pad_left pad_right".

Therefore, there will be 25 entries separated by commas in each line.

Running the Estimation Model

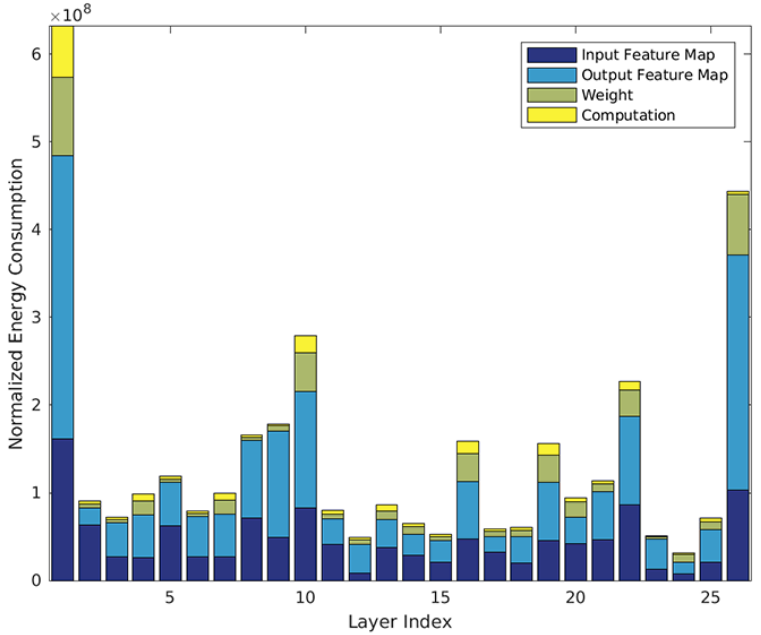
After creating your text file, follow these steps to upload your text file and run the estimation model:

1. Check the "I am not a robot" checkbox and complete the Google reCAPTCHA challenge. Help us prevent spam.
2. Click the "Choose File" button below to choose your text file from your computer.
3. Click the "Run Estimation Model" button below to upload your text file and run the estimation model.

Input DNN Configuration File

```
Layer_Index,Input_Feature_Map,Output_Feature_Map,Weight,Computation
1,161226686.785535,323273662,88858340.625,58290651
2,63540403.7543396,19104256.6840292,4770357.50868125,3263307.50868125
3,26787638.0555562,39583335.5555542,3272222.77777708,2285942.77777708
4,26018817.2746958,48841502.8019458,15927826.1926396,7847418.06763958
5,62285050.8236438,49433953.294575,4188476.6472875,3227376.6472875
6,27267689.7685187,45381705.7407417,3740581.20370417,2666586.20370417
7,26787131.0480146,48586492.3413917,16216779.2956958,8136371.17069583
```

Output DNN energy breakdown across layers

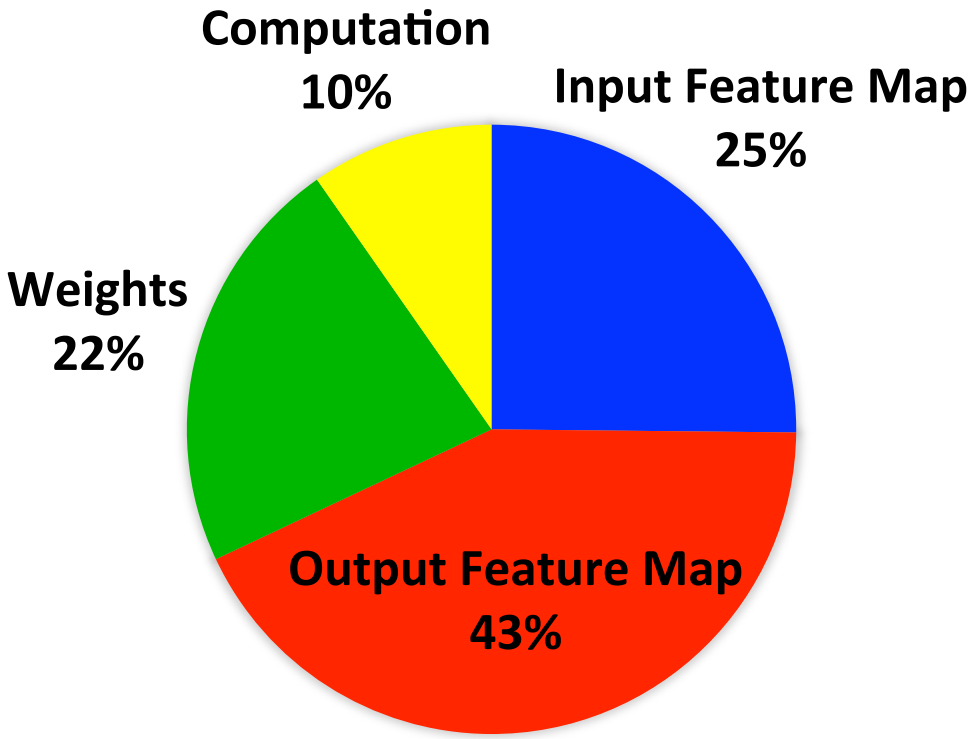


[Yang et al., CVPR 2017]

Key Observations

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

Energy Consumption of GoogLeNet

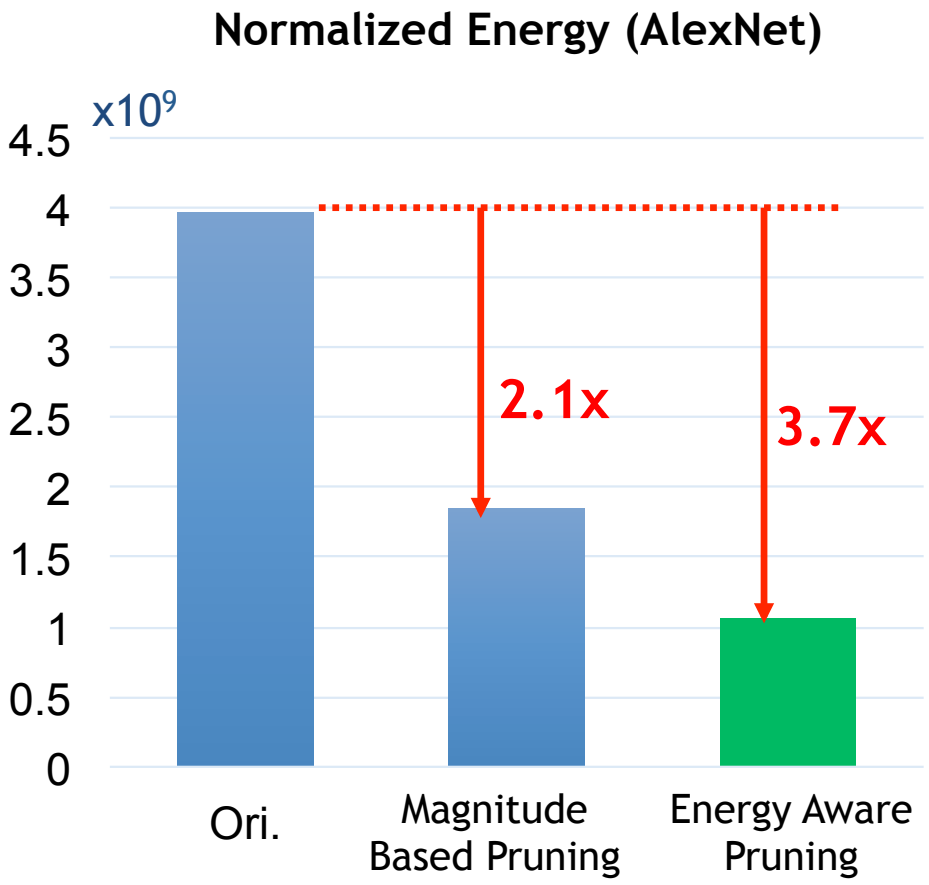


[Yang et al., CVPR 2017]

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**

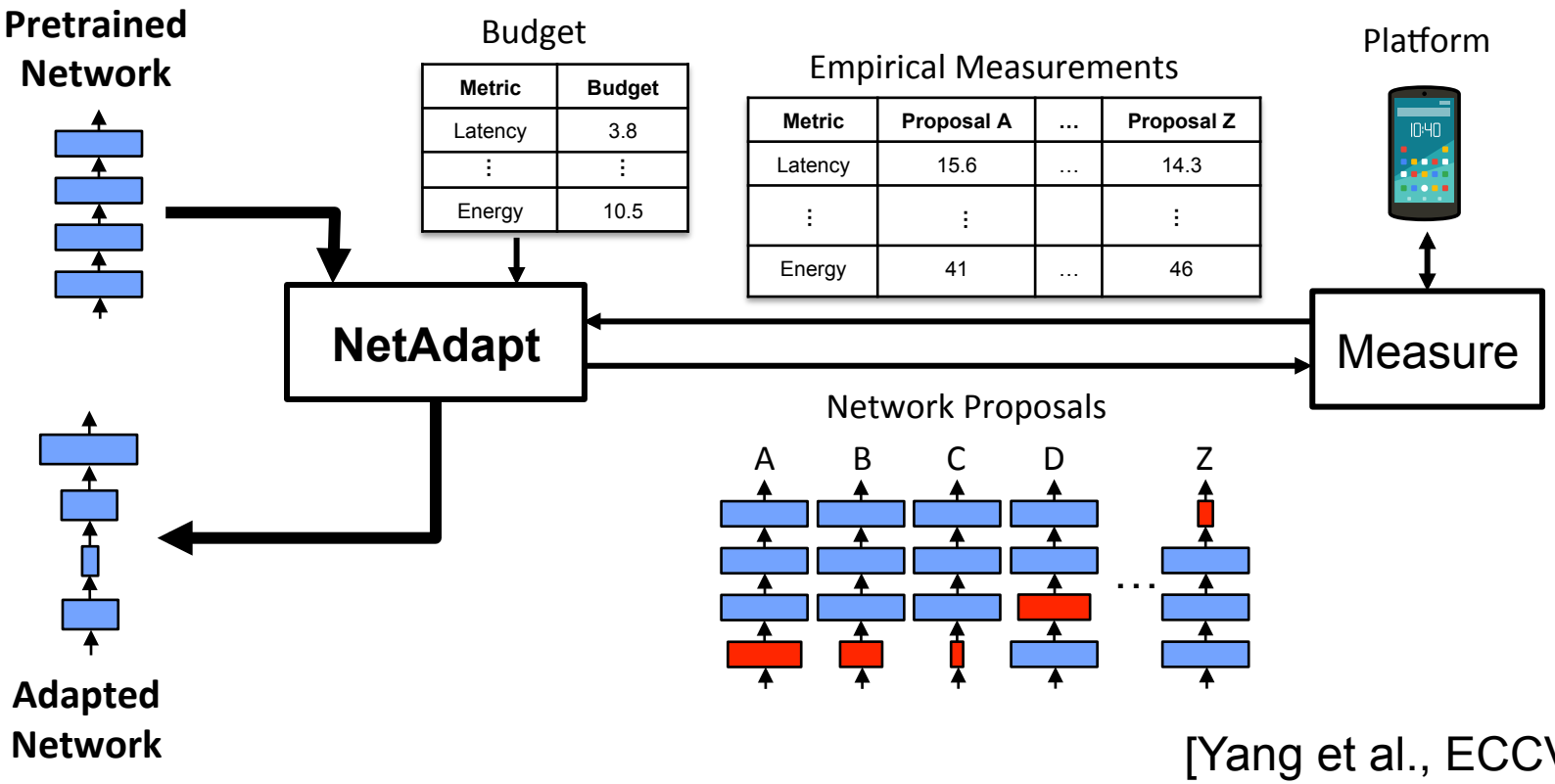


Pruned models available at
http://eyeriss.mit.edu/energy.html

[Yang et al., CVPR 2017]

NetAdapt: Platform-Aware DNN Adaptation

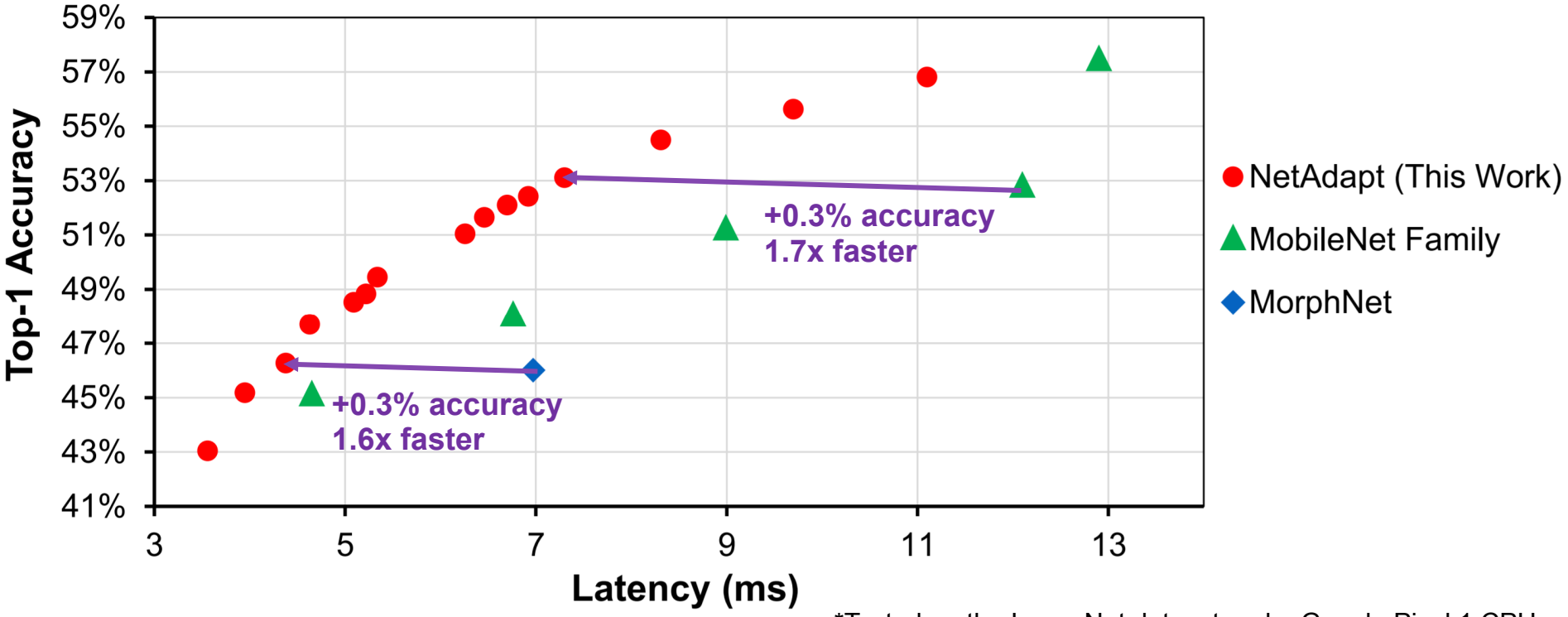
- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)



[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

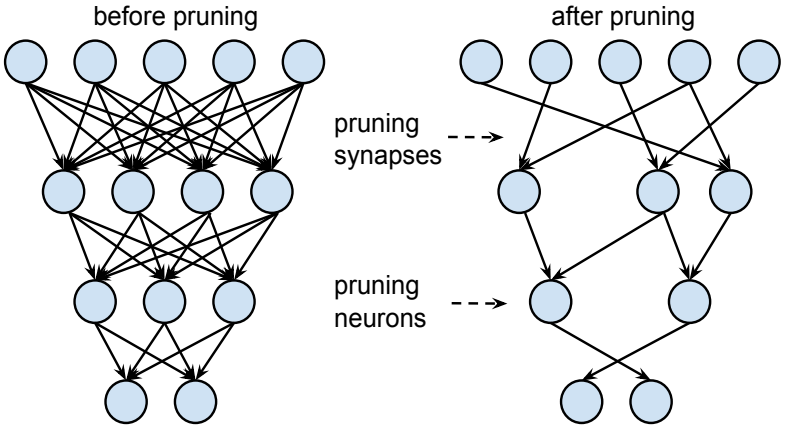


*Tested on the ImageNet dataset and a Google Pixel 1 CPU

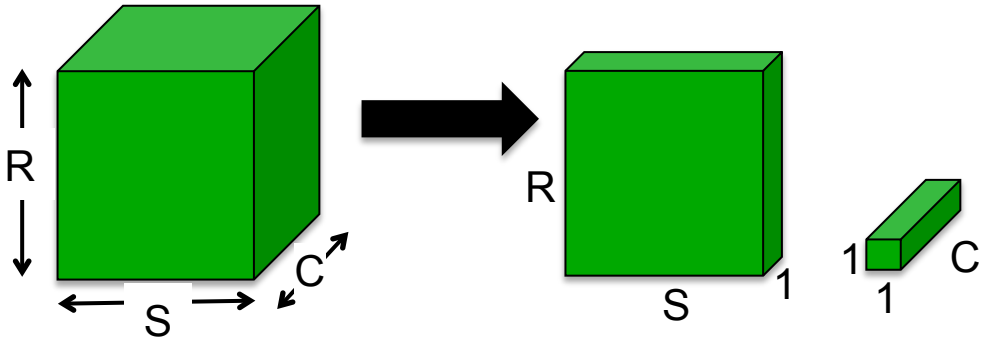
Reference:
MobileNet: Howard et al, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv 2017
MorphNet: Gordon et al., "Morphnet: Fast & simple resource-constrained structure learning of deep networks", CVPR 2018

Many Efficient DNN Design Approaches

Network Pruning



Compact Network Architectures



Reduce Precision

32-bit float 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0

8-bit fixed 0 1 1 0 0 1 1 0

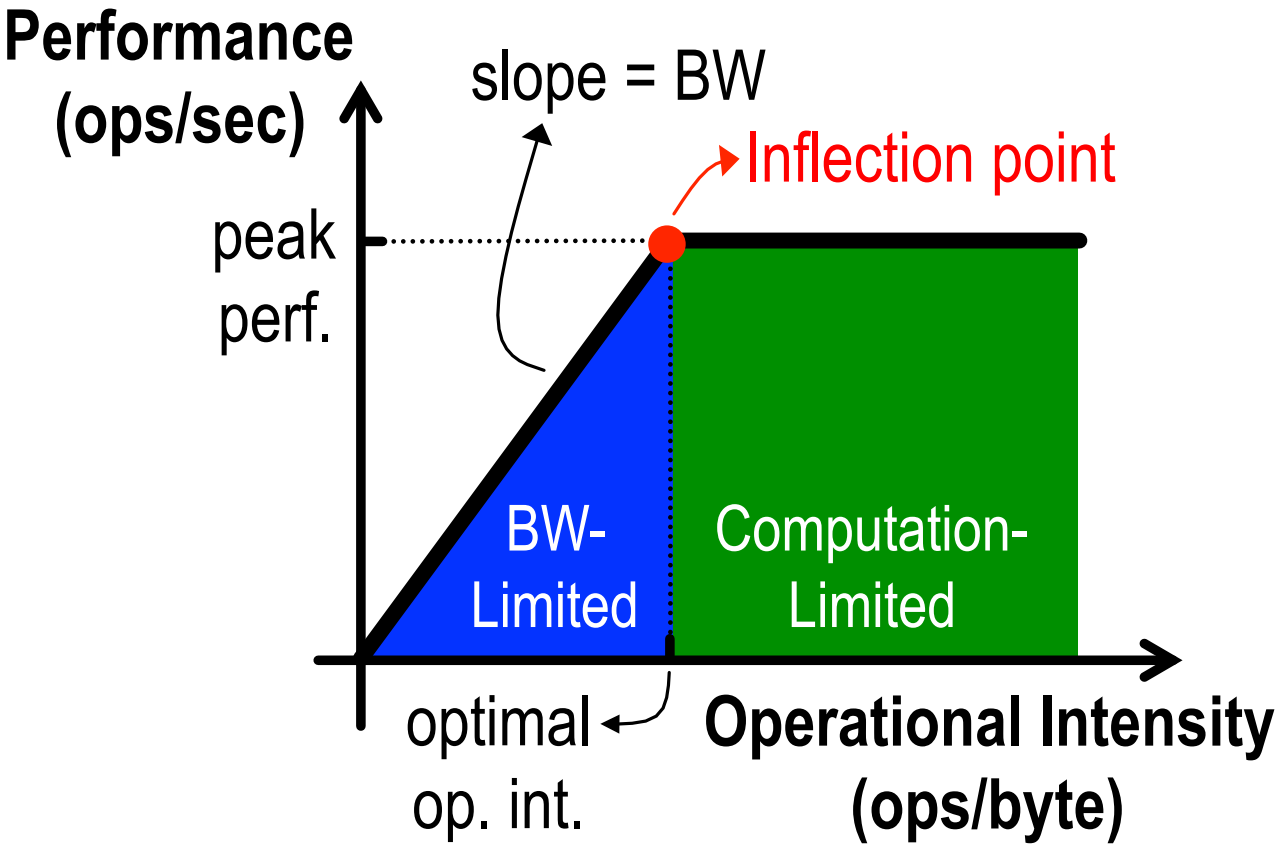
Binary 0

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

[Chen et al., SysML 2018]

Roofline Model

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

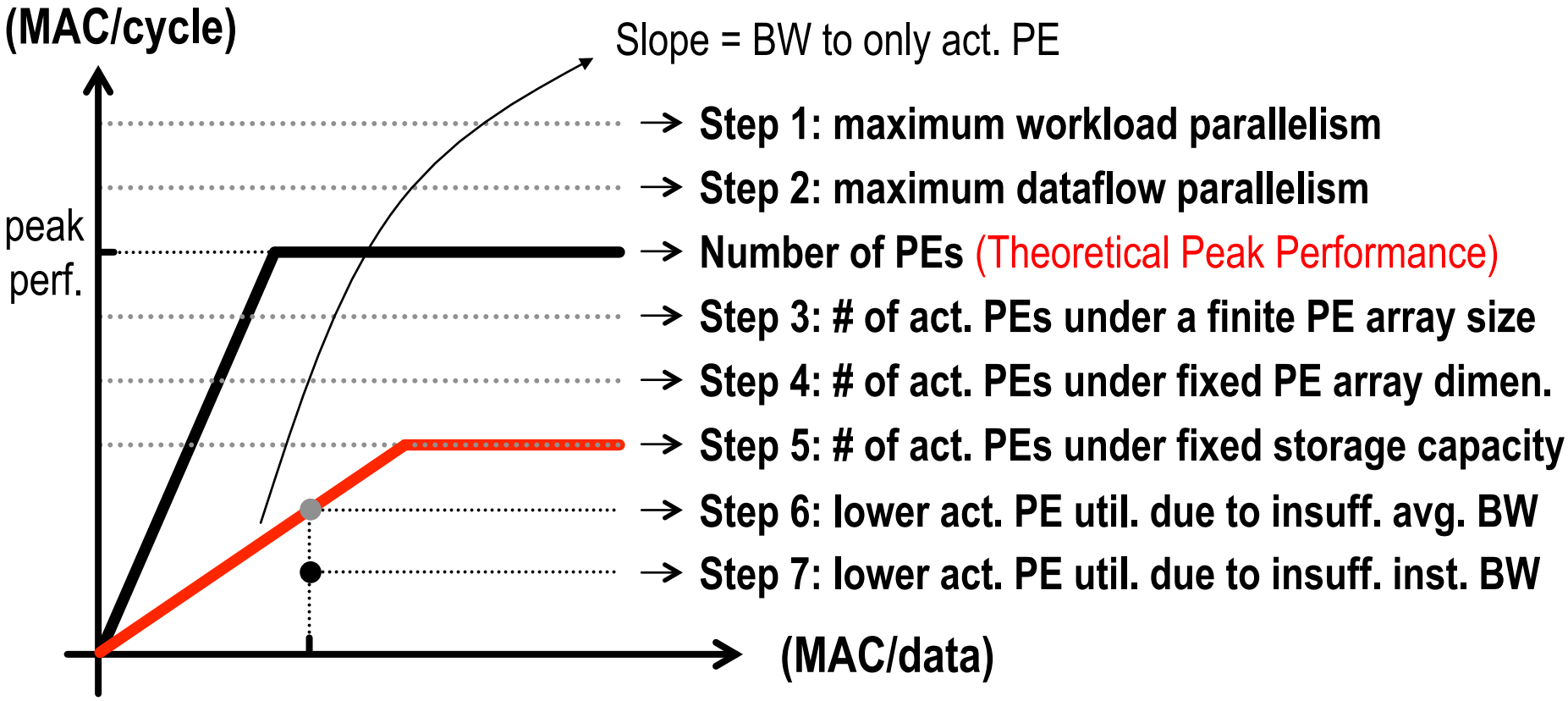


[Williams et al., Comm ACM 2009]

Eyexam: Inefficiencies in DNN Accelerators

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

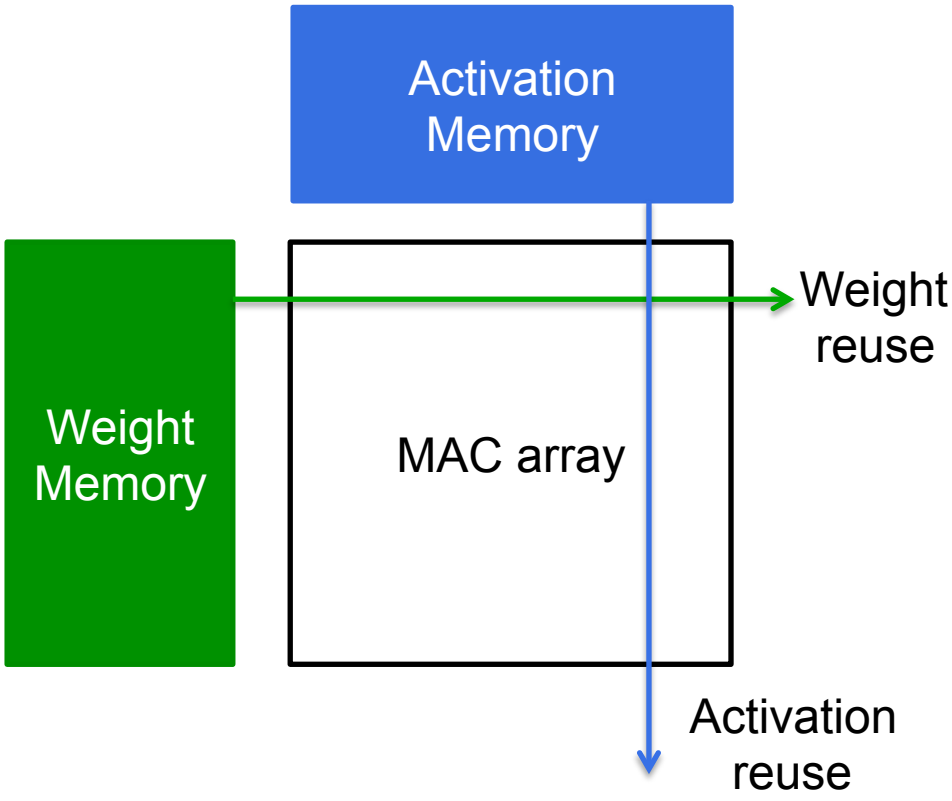
Tightens the roofline model



[Chen et al., arXiv, 2018]

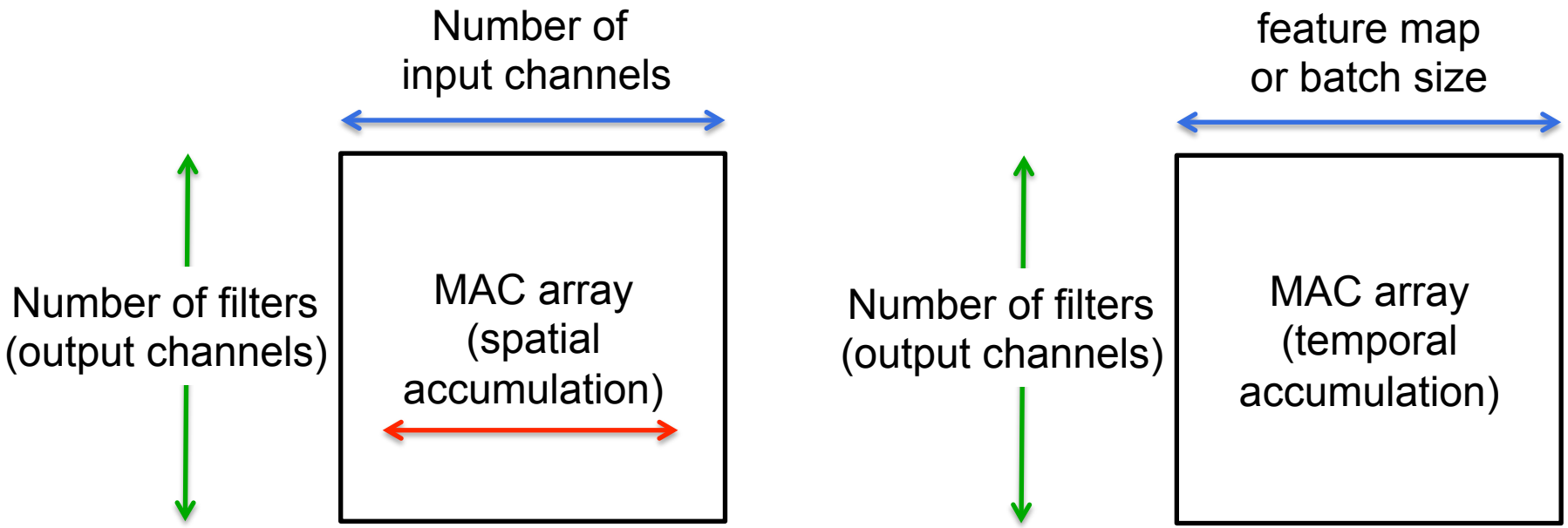
Existing DNN Architectures

- 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



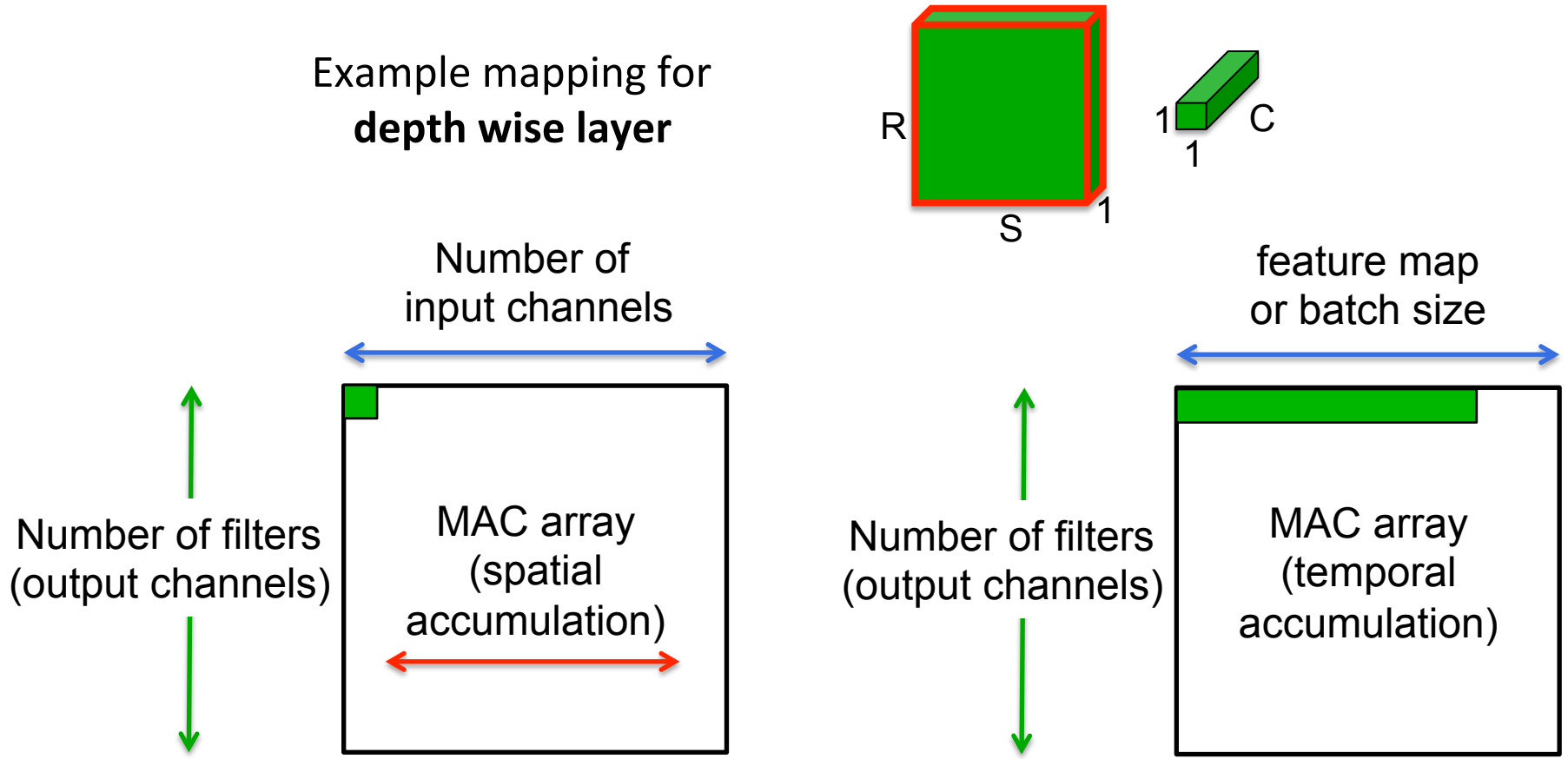
Limitation of Existing DNN Architectures

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



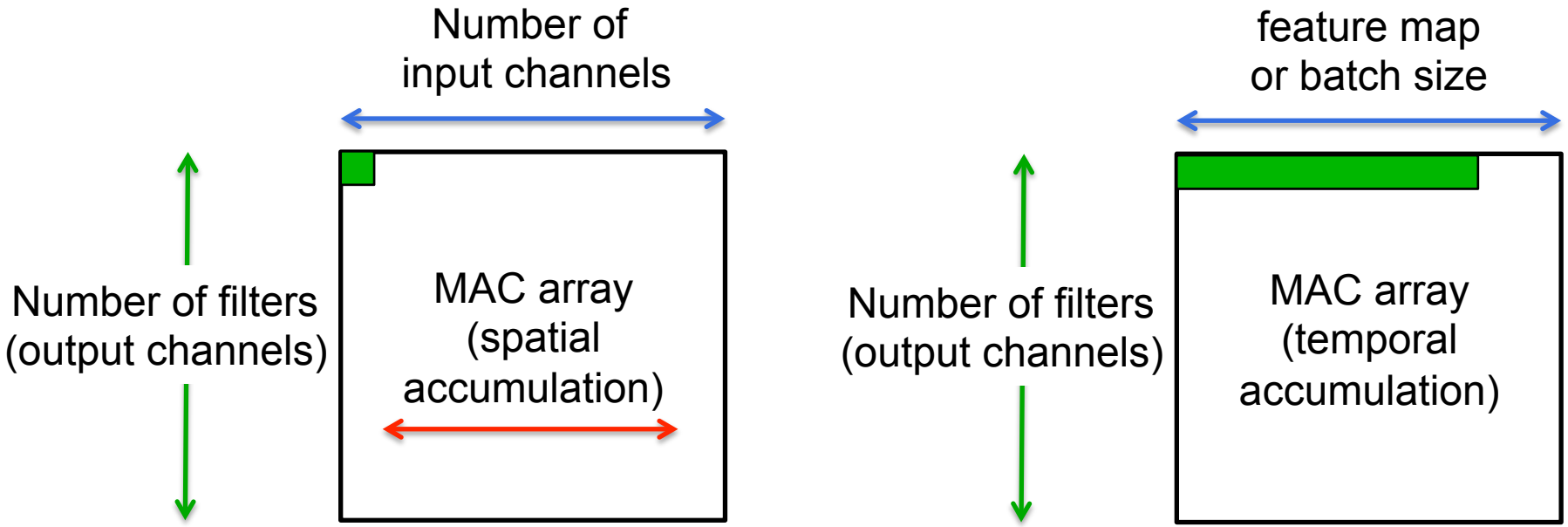
Limitation of Existing DNN Architectures

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



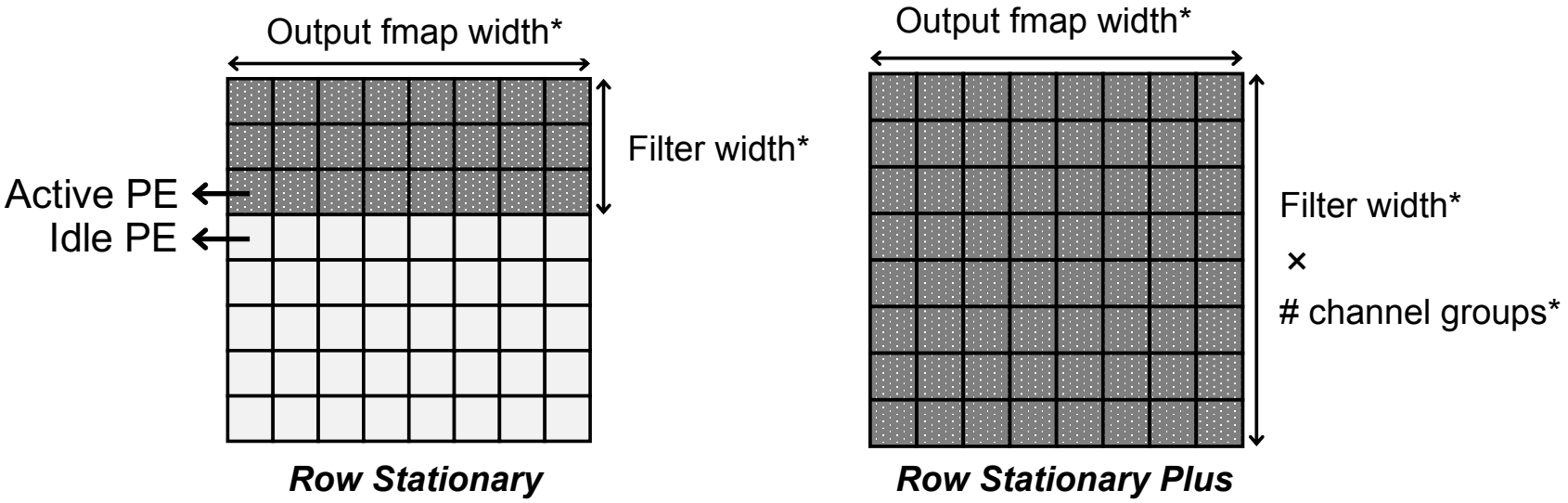
Limitation of Existing DNN Architectures

- **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



Eyeriss v2: Balancing Flexibility and Efficiency

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



*tiling parameters

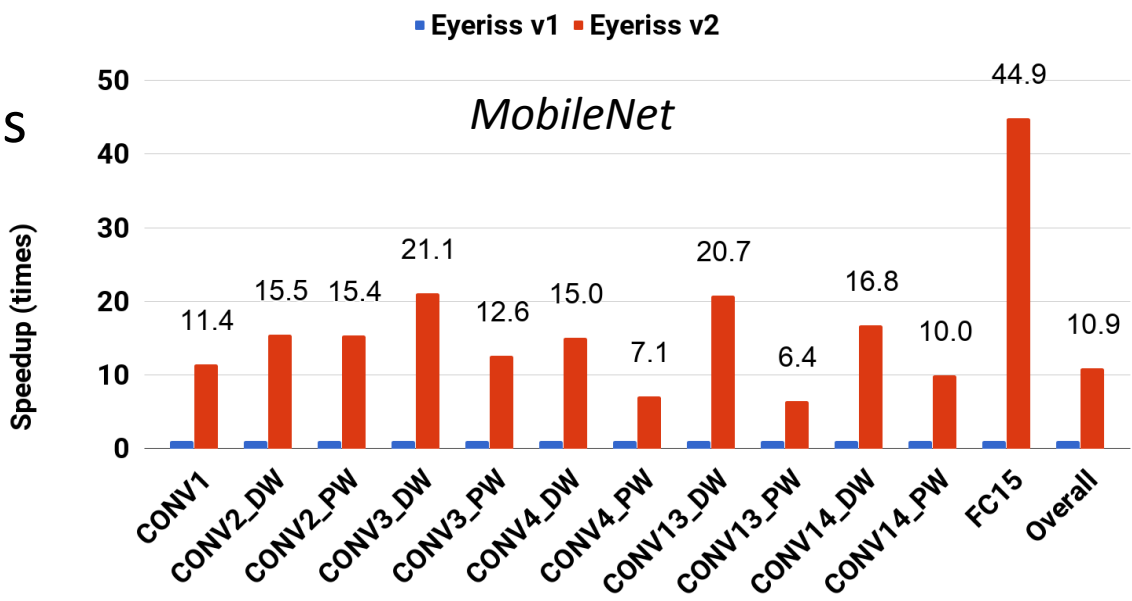
Eyeriss v2: Balancing Flexibility and Efficiency

- Flexible dataflow, called Row-Stationary Plus (RS+), that enables the spatial mapping 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

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**



[Chen et al., arXiv 2018]

https://arxiv.org/abs/1807.07928

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

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, Dec. 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

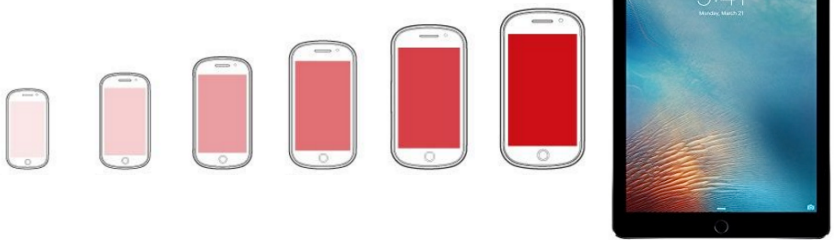
Looking Beyond the DNN Accelerator for Acceleration

Z. Zhang, V. Sze, "**FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos**," *CVPRW* 2017

Super-Resolution on Mobile Devices

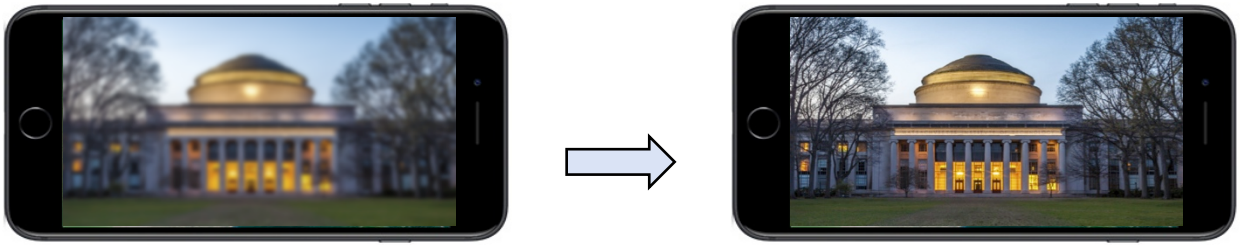
Low Resolution Streaming

High Resolution Playback



Transmit low resolution for lower bandwidth

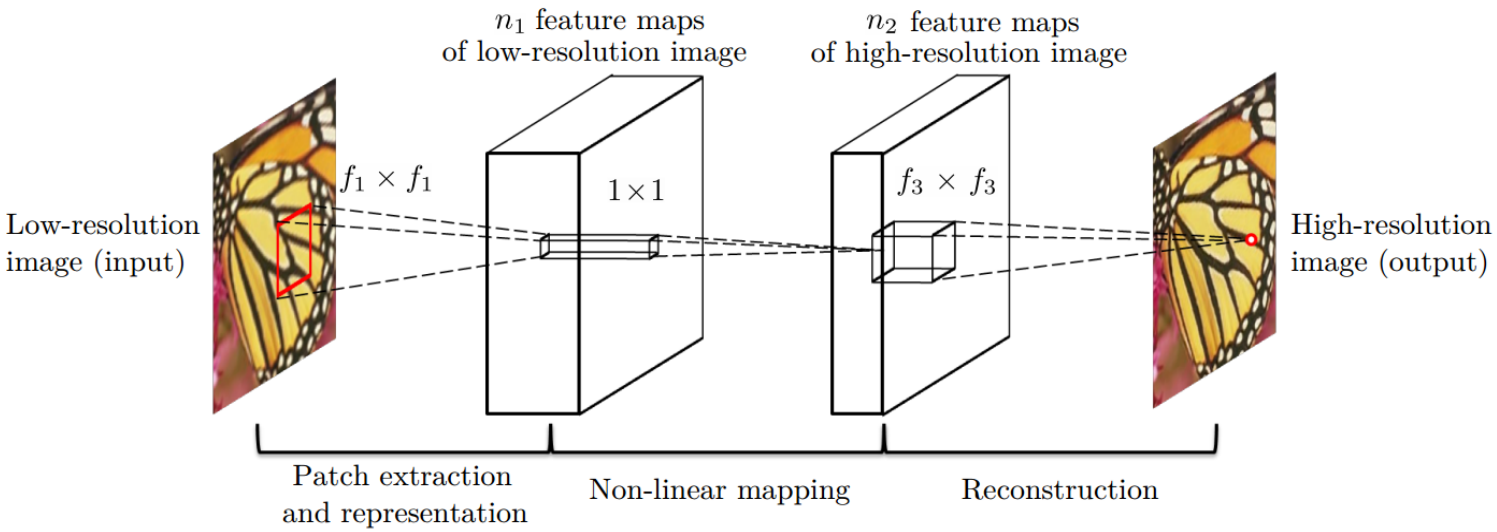
Screens are getting larger



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

Complexity of Super Resolution Algorithms

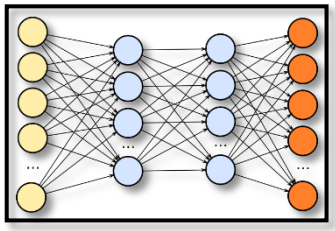
SRCNN (Dong et, al. ECCV 14)



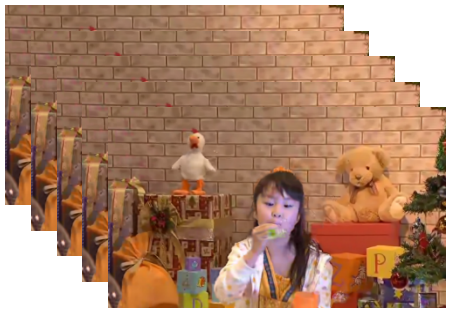
8032 MACs/pixel → ~500 GMAC/s for HD @ 30 fps

State-of-the-art super resolution algorithms use CNNs
→ **computationally expensive**, especially at high resolutions (HD or 4K)

FAST: A Framework to Accelerate SuperRes



SR algorithm



Compressed video

FAST

SR
15x faster

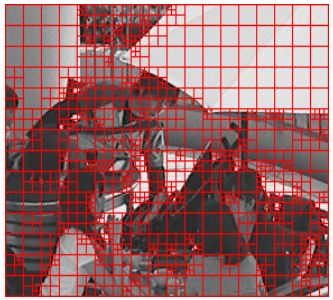
Real-time

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

Decode



Compressed video

Pixels

Block-structure

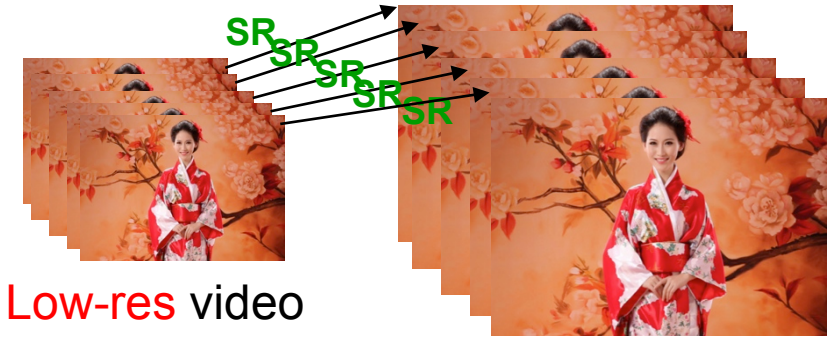
Motion-compensation

Video as a stack of pixels

Representation in compressed video

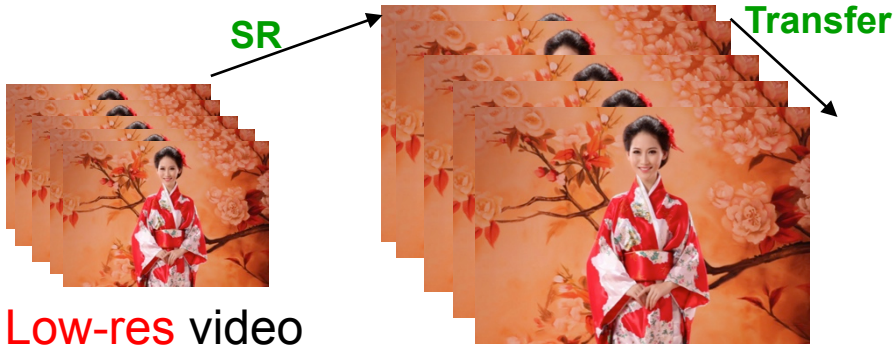
This representation can help **accelerate** super-resolution

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**

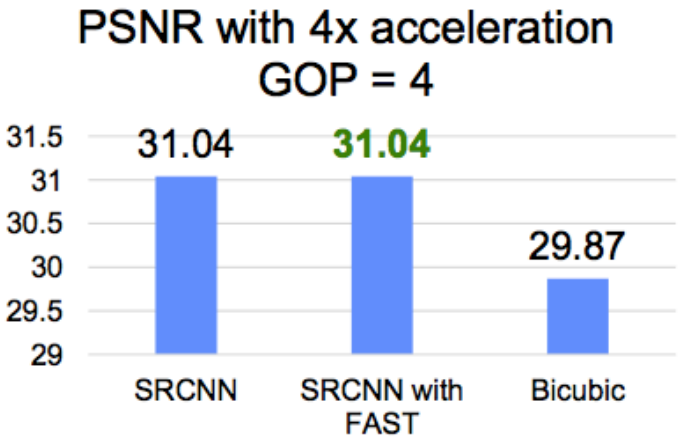
Evaluation: Accelerating SRCNN

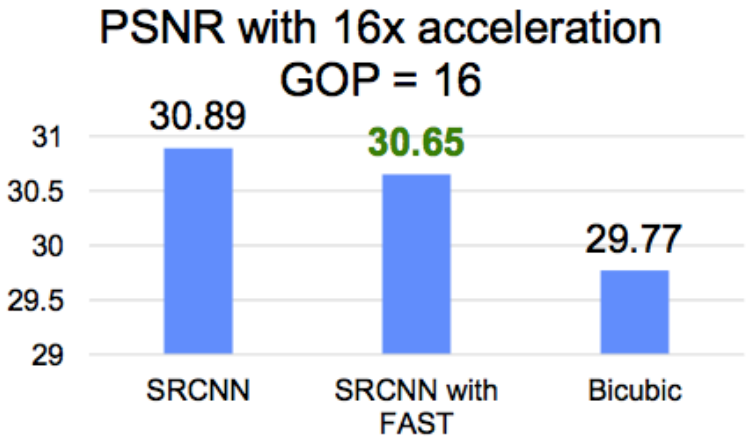
PartyScene

RaceHorse

BasketballPass

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





4 × acceleration with NO PSNR LOSS. 16 × 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)

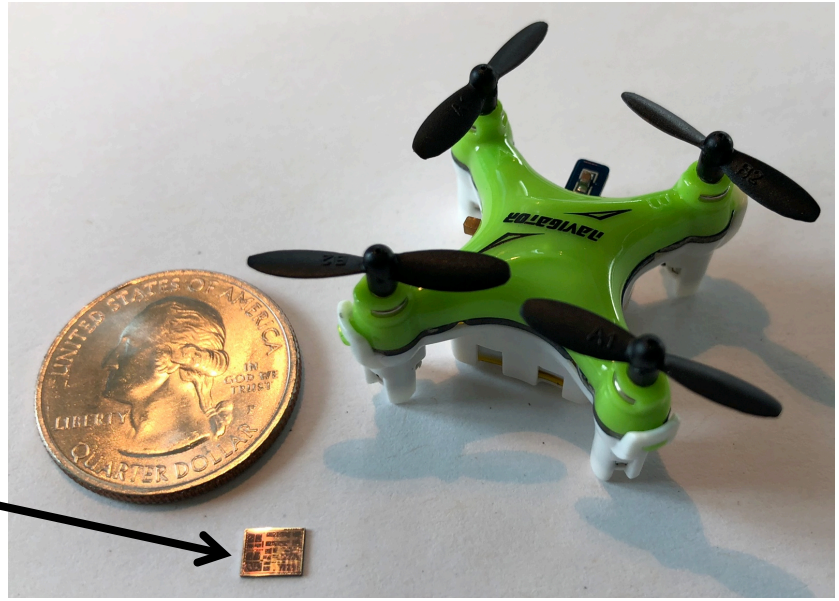
Code released at www.rle.mit.edu/eems/fast

Beyond Deep Neural Networks

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**," *Symposium on VLSI* 2018

Energy-Efficient Autonomous Navigation

Navion Chip
Localization and Mapping at 2mW
(full integration on-chip)



Enable energy-efficient navigation for **Search and Rescue**

http://navion.mit.edu

[Zhang et al., RSS 2017],
[Suleiman et al., VLSI 2018]

Localization and Mapping Using VIO*

VIO determines location/orientation of drone from images and IMU (also used by headset in Augmented Reality and Virtual Reality)

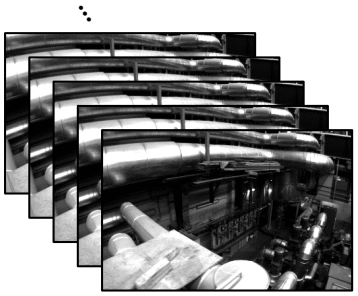
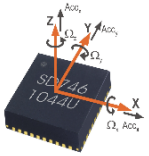


Image sequence

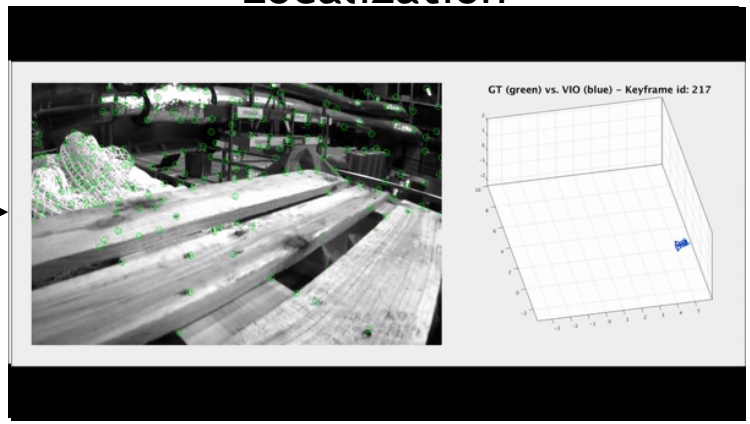
IMU

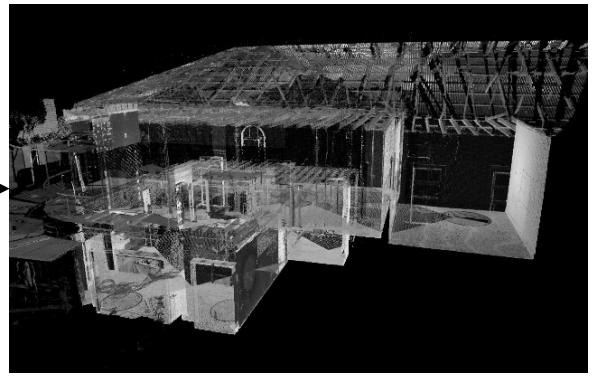
Inertial Measurement Unit



Visual-Inertial Odometry (VIO)

Localization

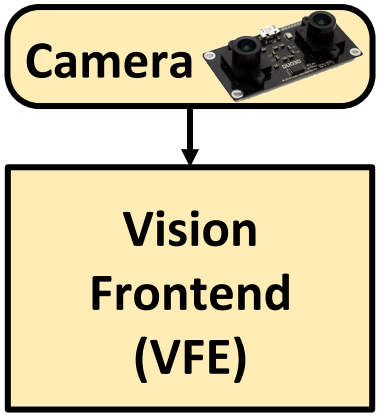




Mapping

***Subset of SLAM algorithm**
(Simultaneous Localization And Mapping)

Frontend: Processing Sensors Data

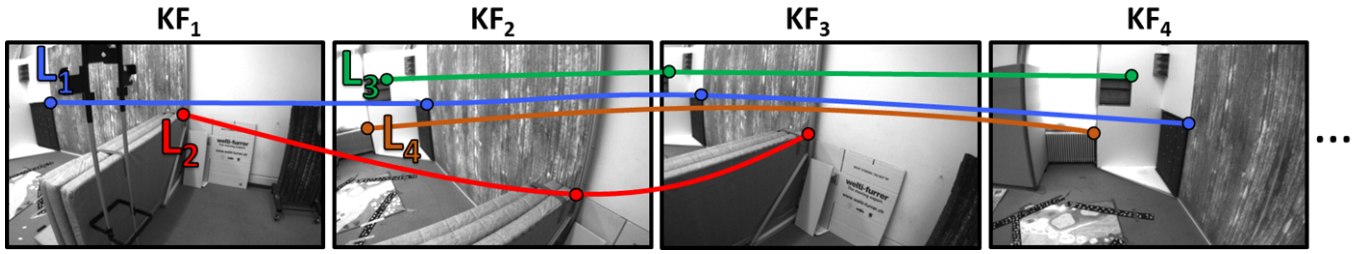


[Zhang et al., RSS 2017]

Frontend: Processing Sensors Data

Vision Frontend (VFE)

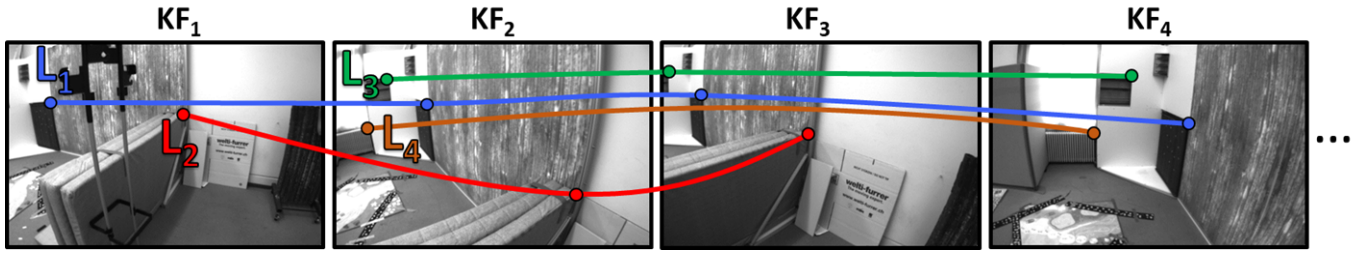
Feature Tracks

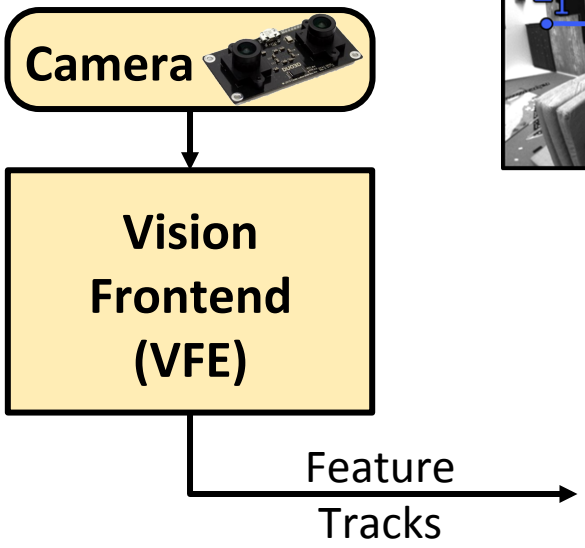


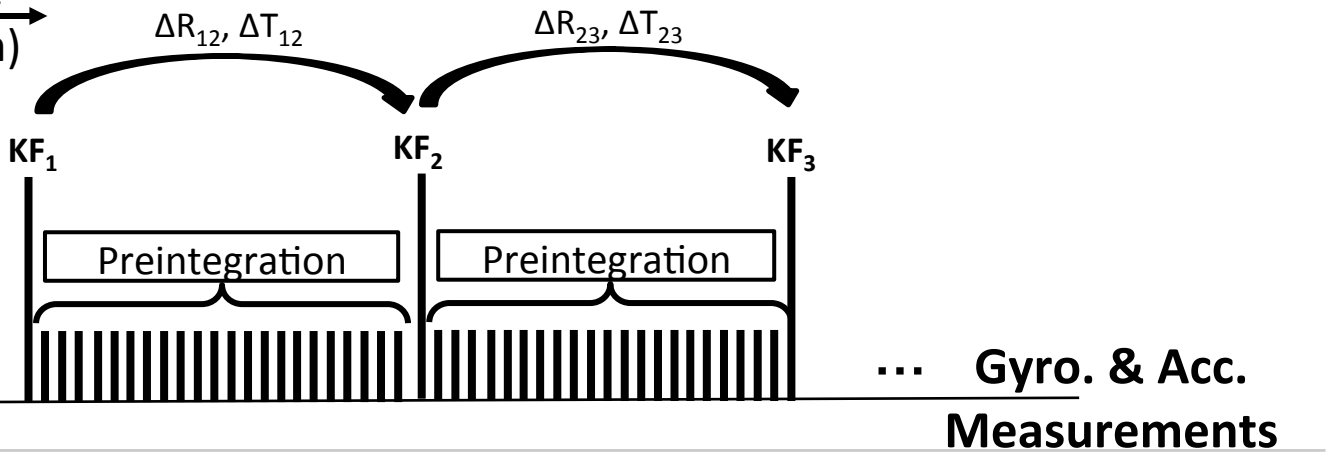
IMU Frontend (IFE)

[Zhang et al., RSS 2017]

Frontend: Processing Sensors Data

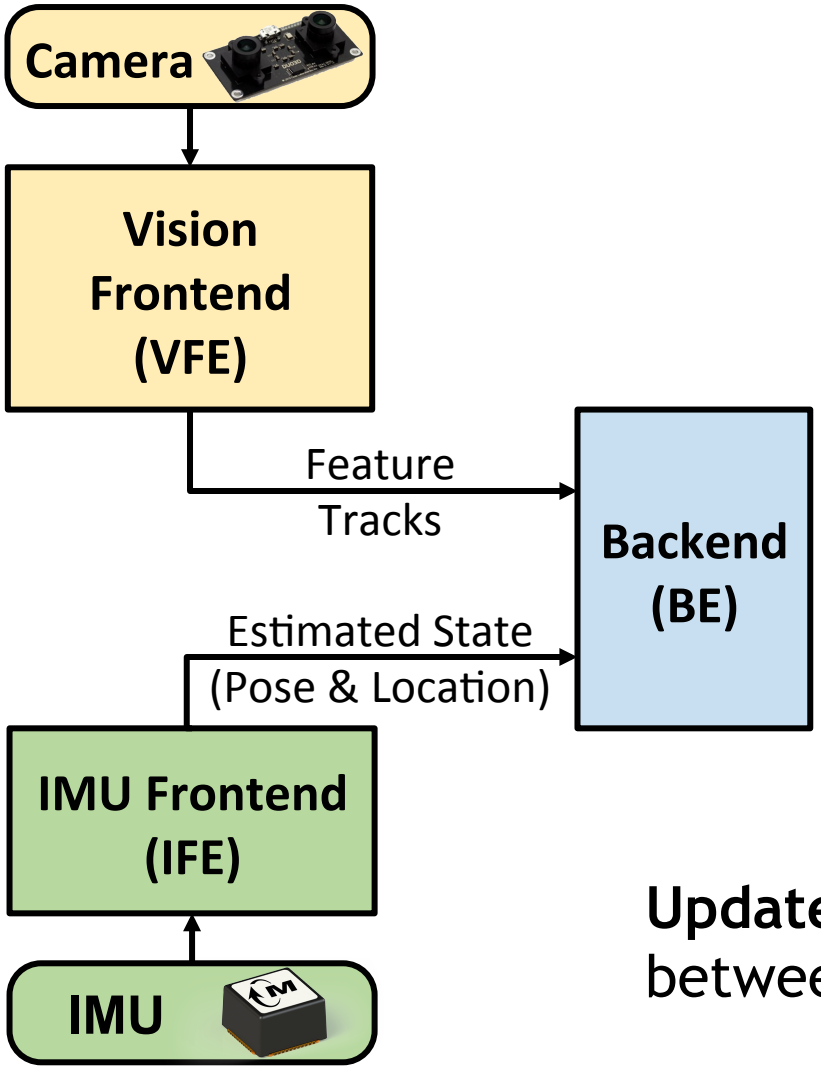


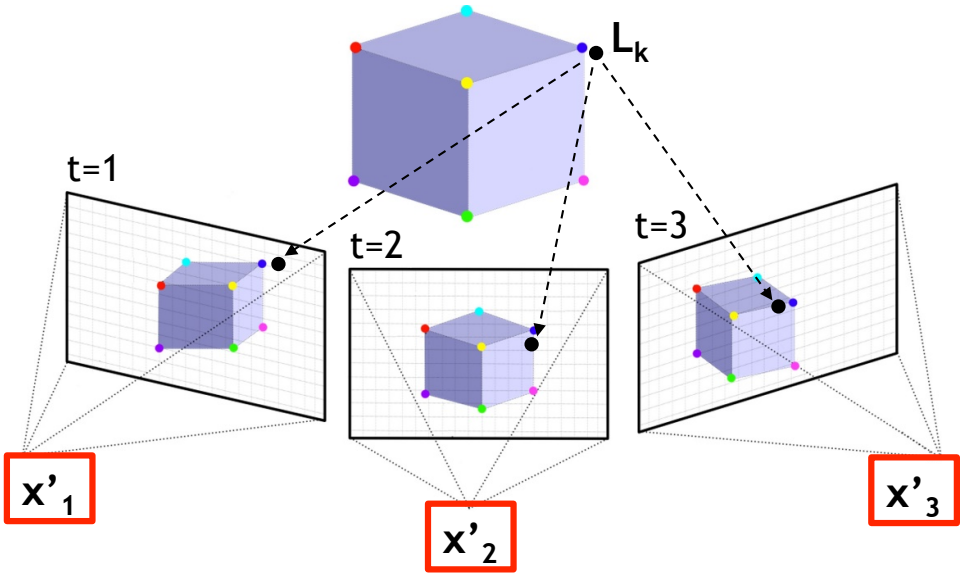




[Zhang et al., RSS 2017]

Backend: Reduce Inconsistency

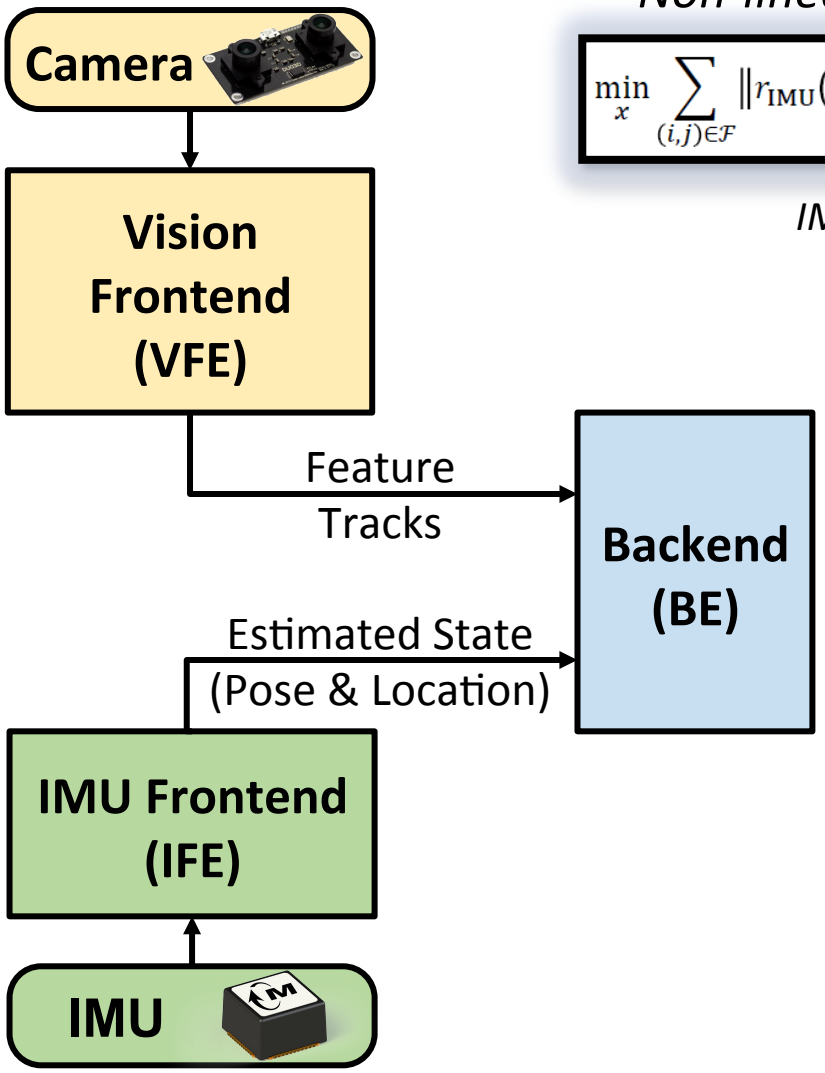




Update states (x_i) to minimize inconsistencies between measurements across time

[Zhang et al., RSS 2017]

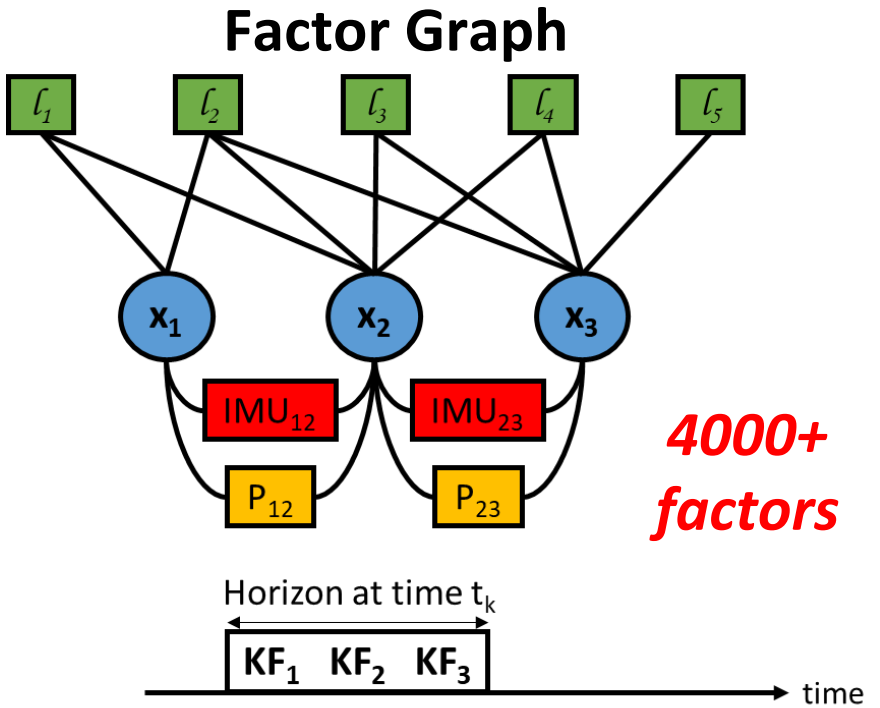
Backend: Factor Graph to Infer State of Drone



Non-linear least squares factor graph optimization

$$\min_{x} \sum_{(i,j)\in\mathcal{F}} \left\| r_{\text{IMU}}(x, \Delta\tilde{R}_{ij}, \Delta\tilde{p}_{ij}, \Delta\tilde{v}_{ij}) \right\|^2 + \sum_{k\in\mathcal{L}} \sum_{i\in\mathcal{F}_k} \left\| r_{\text{CAM}}(x, l_k, u_{ik}^l, u_{ik}^r) \right\|^2 + \left\| r_{\text{PRIOR}}(x) \right\|^2$$

IMU Factors *Vision Factors* *Other Factors*



[Zhang et al., RSS 2017]

Backend: Factor Graph to Infer State of Drone

Non-linear least squares factor graph optimization

$$\min_{x} \sum_{(i,j)\in\mathcal{F}} \left\| r_{\text{IMU}}(x, \Delta\tilde{R}_{ij}, \Delta\tilde{p}_{ij}, \Delta\tilde{v}_{ij}) \right\|^2 + \sum_{k\in\mathcal{L}} \sum_{i\in\mathcal{F}_k} \left\| r_{\text{CAM}}(x, l_k, u_{ik}^l, u_{ik}^r) \right\|^2 + \left\| r_{\text{PRIOR}}(x) \right\|^2$$

IMU Factors *Vision Factors* *Other Factors*

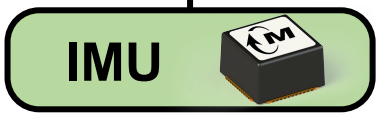
Vision Frontend (VFE)

Feature Tracks

Backend (BE)

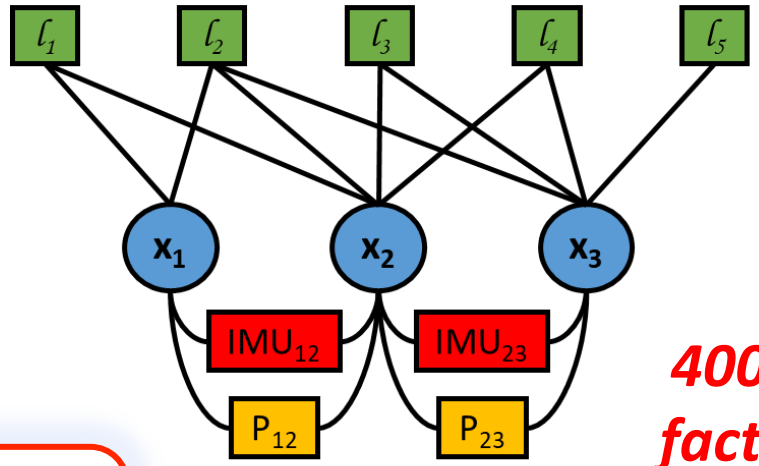
Estimated State (Pose & Location)

IMU Frontend (IFE)

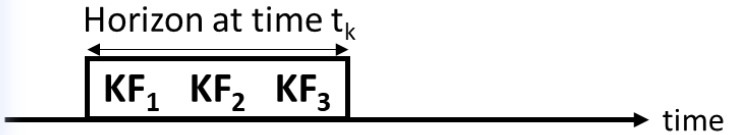


Updated States (x_i) & Sparse 3D Map

Factor Graph

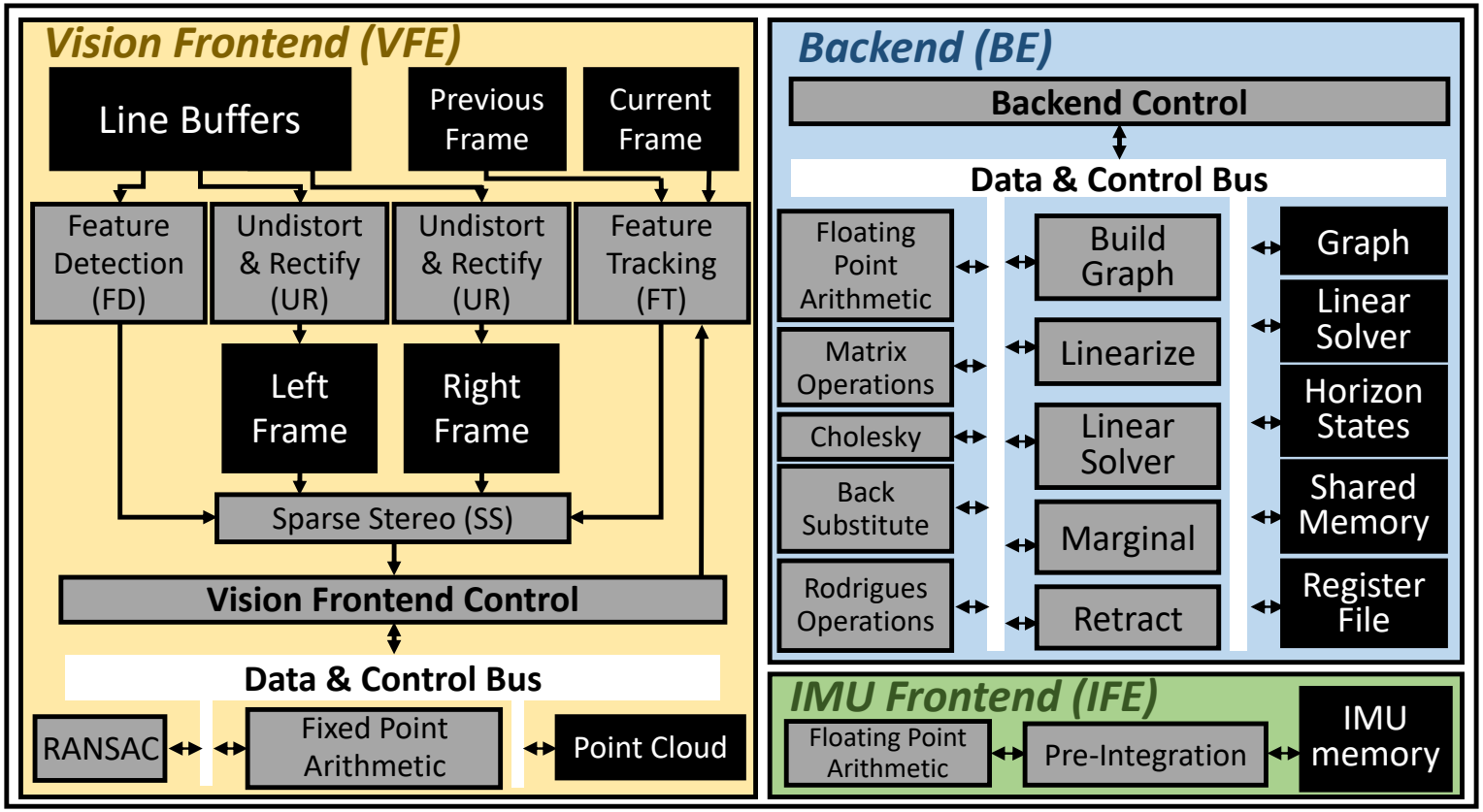


4000+ factors



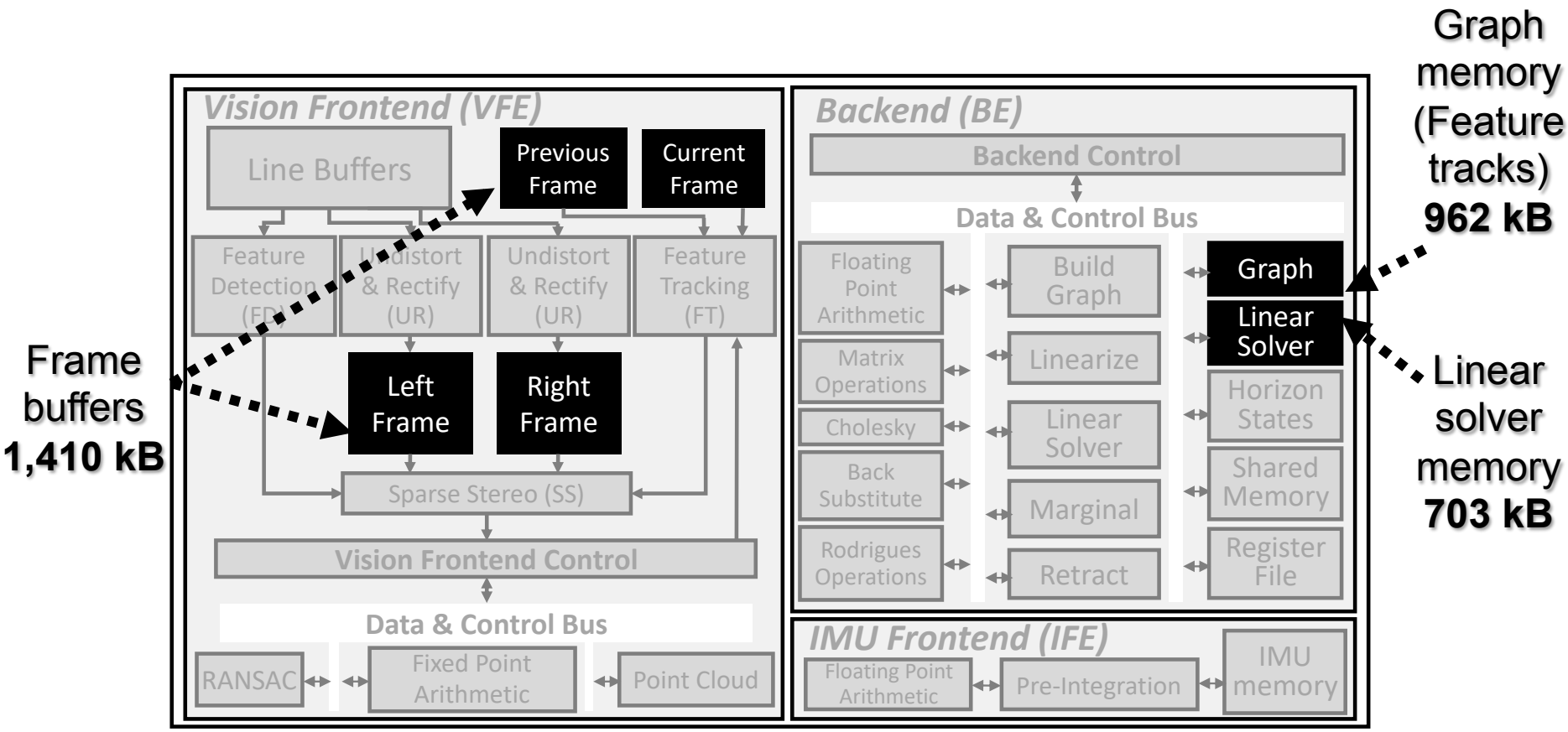
[Zhang et al., RSS 2017]

Navion Chip Architecture



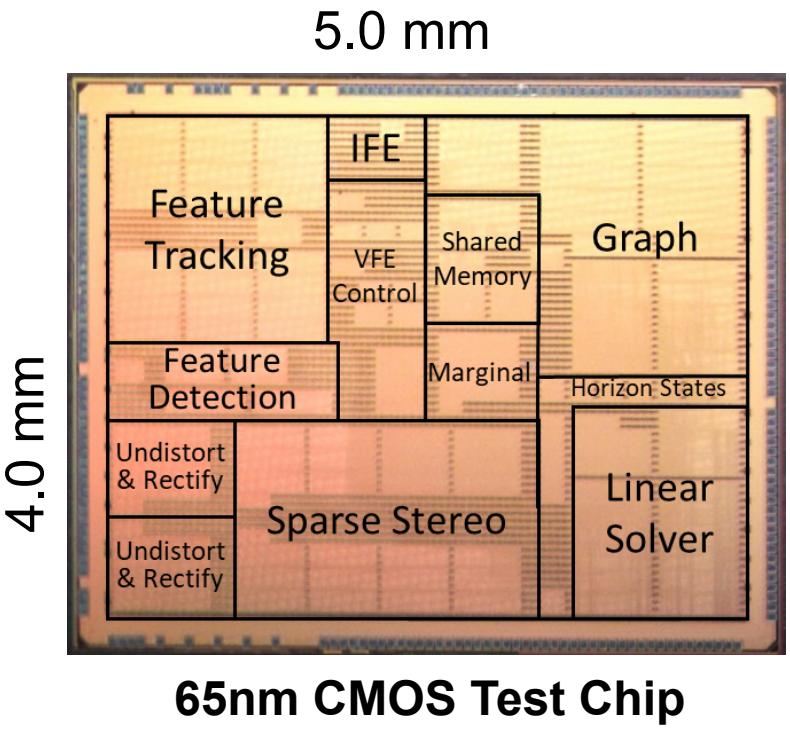
Navion is a <u>fully integrated</u> system:
No off-chip storage or processing

Reduce Memory



Use Compression and Exploit Sparsity to reduce memory by 4.1x

Navion Evaluation

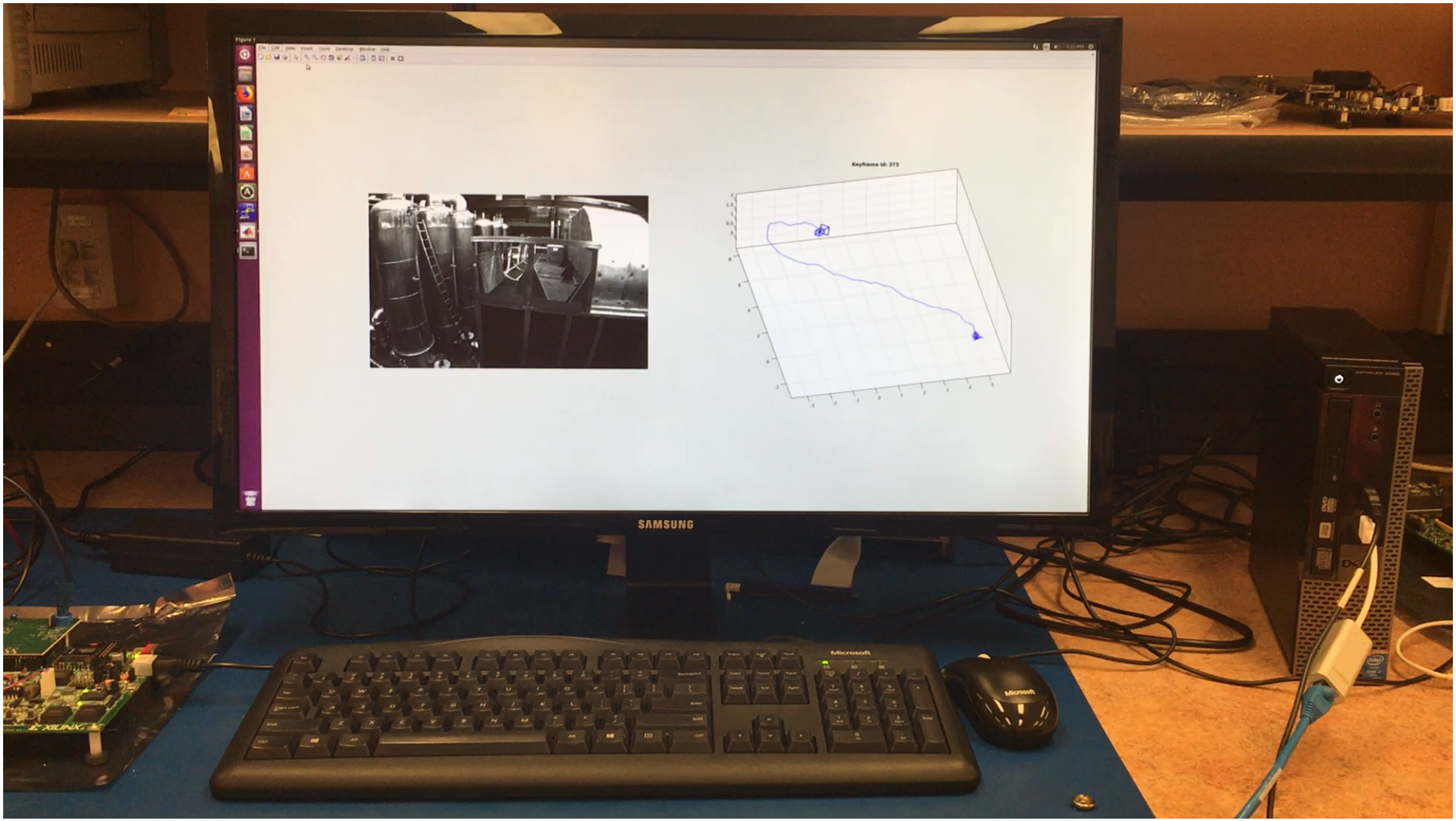


65nm CMOS Test Chip

Over 250 configurable parameters *to adapt to different sensors and environments*

- **Peak Performance @ Maximum Configuration**
  - VFE: 28 – 171 fps (71 fps average)
  - BE: 16 – 90 fps (19 fps average)
  - Average Power Consumption: 24mW
  - Trajectory Error: 0.28%
- **Real-Time Performance @ Optimized Configuration**
  - VF: 20 fps
  - BE: 5 fps
  - Average Power Consumption: 2mW
  - Trajectory Error: 0.27%

Navion System Demo



https://youtu.be/X5VZkPo_704

Summary

- **Data movement dominates energy consumption in deep learning hardware**
  - Use dataflow that maximizes data reuse for ***all data types***
- **Design considerations for deep learning at the edge**
  - Incorporate ***direct metrics*** into algorithm design for improved efficiency
  - Use a ***flexible dataflow*** and NoC to exploit data reuse for energy efficiency and increase PE utilization for speed
- **Accelerate deep learning by looking beyond the accelerator**
  - Exploit data representation for FAST Super-Resolution
- **Other forms of inference at the edge beyond deep learning**
  - Graphical models for localization and mapping in nanodrones

Acknowledgements

Joel Emer

Tushar Krishna

Sertac Karaman

Luca Carlone

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

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH · NSF · ANALOG DEVICES · BROADCOM · Google · intel · NVIDIA · DARPA · SRC · 3M · QUALCOMM · SAMSUNG · Texas Instruments · tsmc

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**, December 2017

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

MIT Professional Education Course on
"Designing Efficient Deep Learning Systems"
http://professional-education.mit.edu/deeplearning

For updates Follow @eems_mit

http://mailman.mit.edu/mailman/listinfo/eems-news

References

- **Energy-Efficient Hardware for Deep Neural Networks**
  - *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. http://eyeriss.mit.edu*
  - *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.*

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.*
  - *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," ECCV 2018.*
  - *Y.-H. Chen, J. Emer, V. Sze, "Eyeriss v2: A Flexible and High-Performance Accelerator for Emerging Deep Neural Networks," arXiv 2018. https://arxiv.org/abs/1807.07928*

References

- **Looking Beyond the 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*
- **Looking Beyond DNNs: Other forms of inference at the edge**
  - *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. http://navion.mit.edu*
  - *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: An Energy-Efficient Visual-Inertial Odometry Accelerator for Micro Robotics and Beyond," IEEE Hot Chips: A Symposium for High-Performance Chips, August 2018.*