Energy-Efficient Processing at the Edge: From Compressing to Understanding Pixels

Vivienne Sze

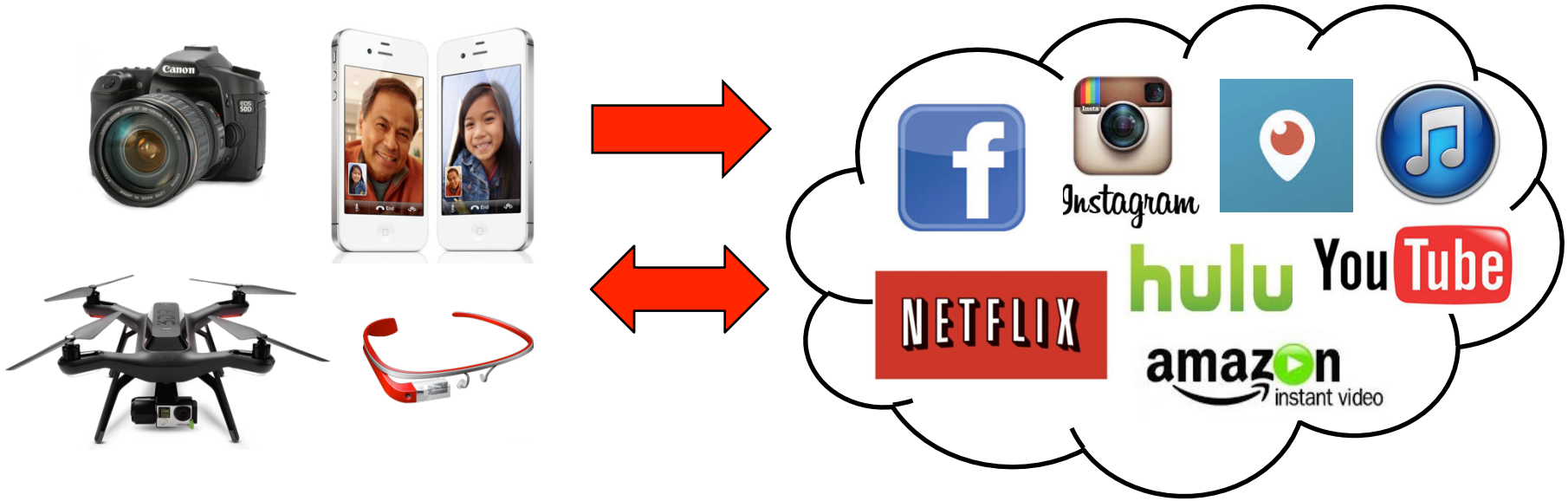
Massachusetts Institute of Technology

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

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

Video is the Biggest Big Data

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



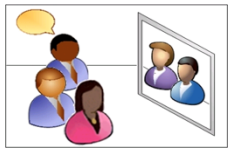
Energy limited due to battery capacity

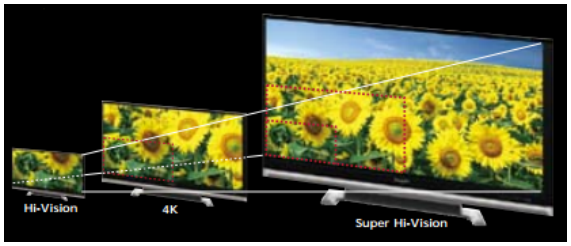
Power limited due to heat dissipation

Need energy-efficient pixel processing!

Energy-Efficient Pixel Processing

Next-Generation Video Coding (Compress Pixels)

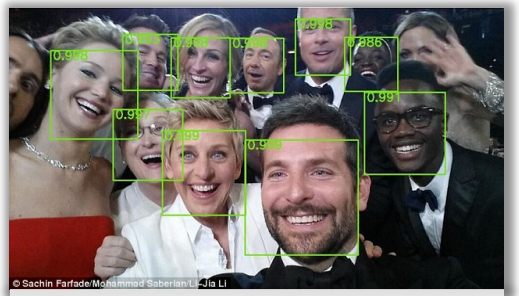




Ultra-HD

Goal: Increase coding efficiency, speed and energy-efficiency

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



Recognition

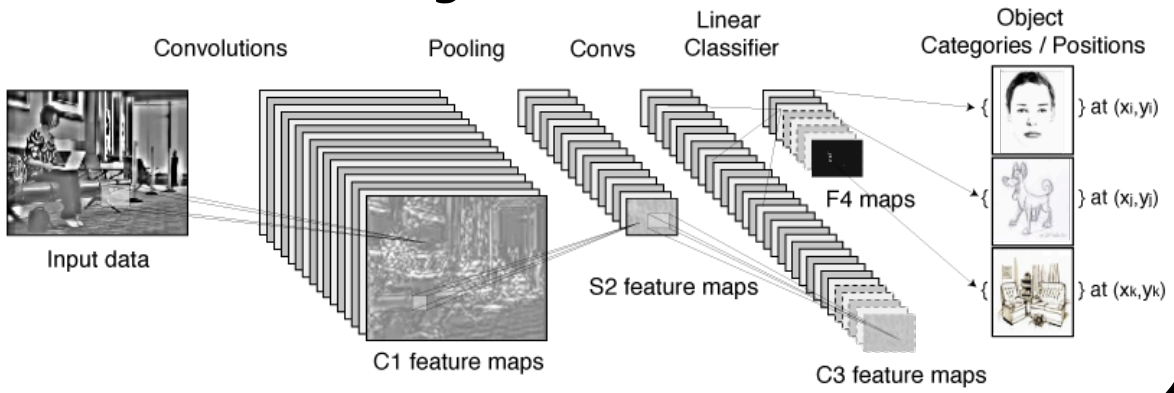
Self-Driving Cars

AI

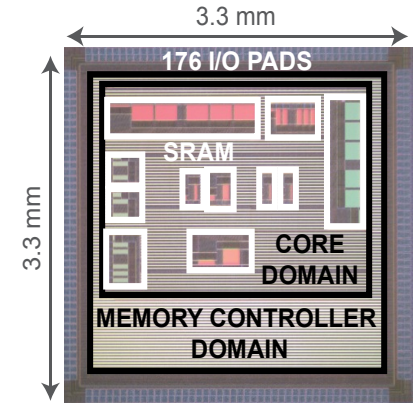
Goal: Make computer vision as ubiquitous as video coding

Energy-Efficient Cross-Layer Design

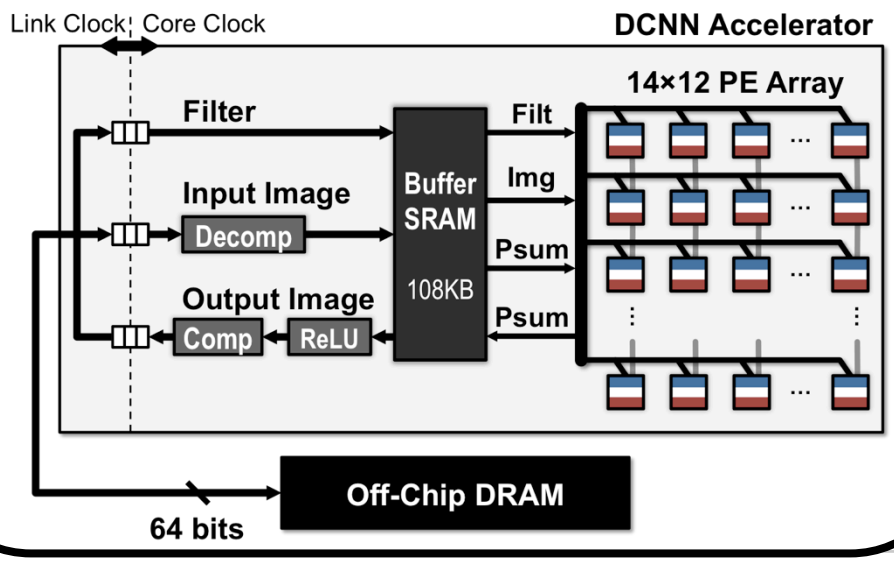
Algorithms



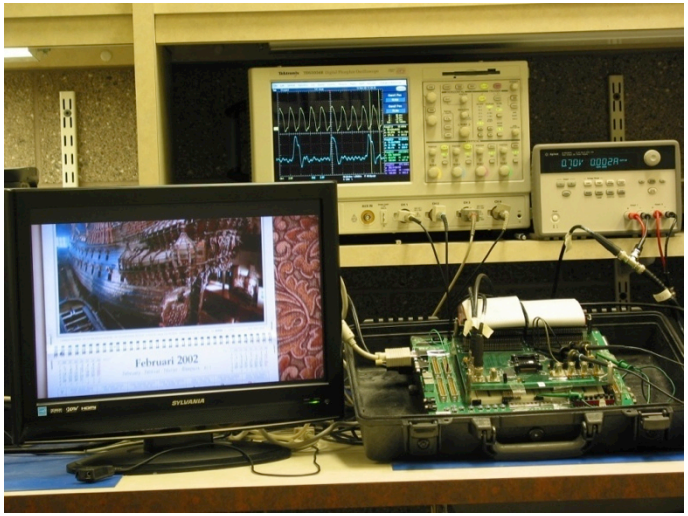
Circuits



Architectures



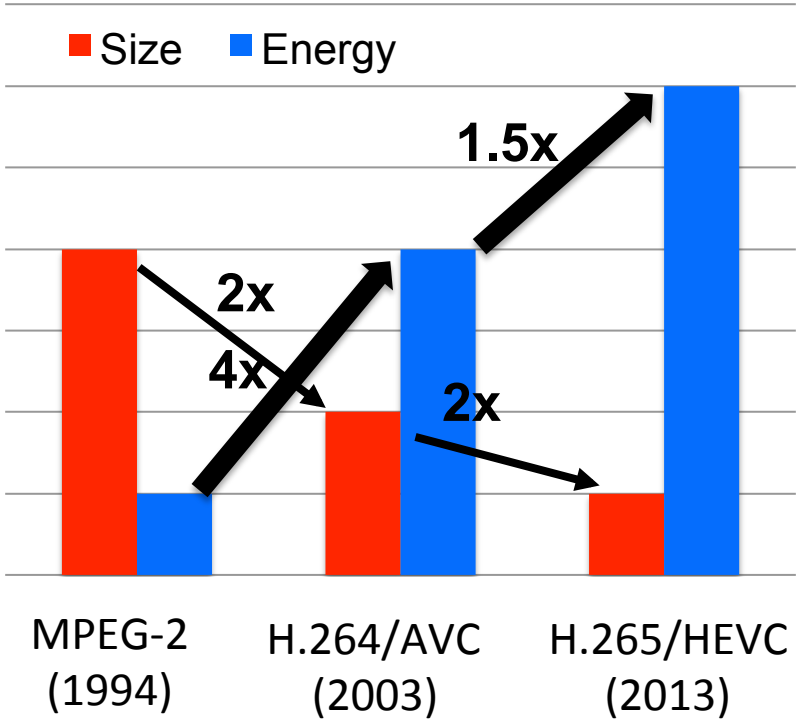
Systems



Energy-Efficient Video Compression

High Efficiency Video Coding (HEVC)

- HEVC achieves ~2x higher coding efficiency than H.264/AVC
- High throughput (Ultra-HD 8K @ 120fps) & low power
  - Implementation-friendly features (e.g. built-in parallelism)



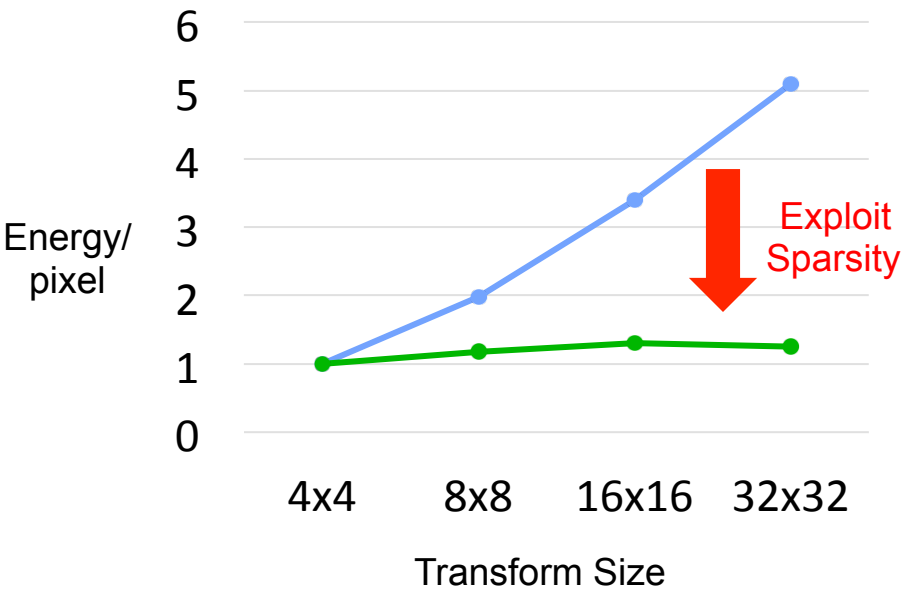
	Coding Efficiency	Efficient Implementation
Larger and Flexible Coding Block Size	X	
More Sophisticated Intra Prediction	X	
Larger Interpolation for Motion Comp.	X	
Larger Transform Size	X	
Parallel Deblocking Filter		X
Sample Adaptive Offset	X	
High-Throughput CABAC	X	X
High Level Parallel Tools		X

Joint algorithm and hardware design is required to address **coding efficiency, throughput and power challenges**

Efficient Hardware for HEVC

Energy-Efficient HEVC Transform

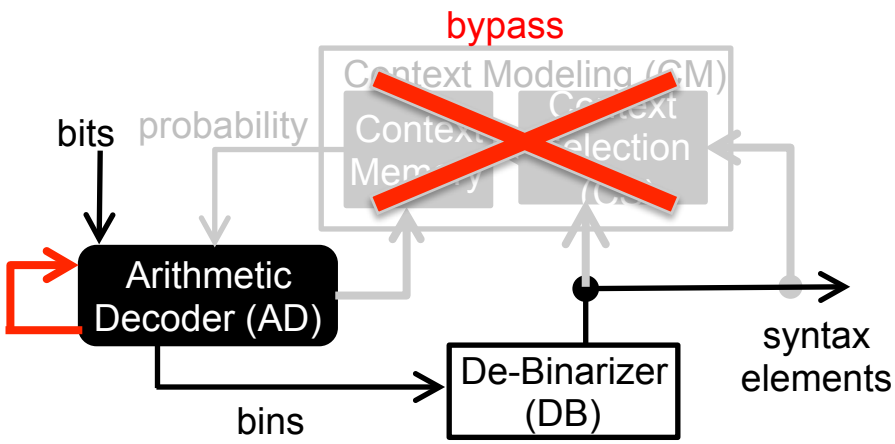
- Large transforms give coding gain: 7-9%
- Adapt to sparsity of coefficients
- **Enable constant energy/pixel for all transform sizes**



[M. Tikekar et al., ICIP 2014]

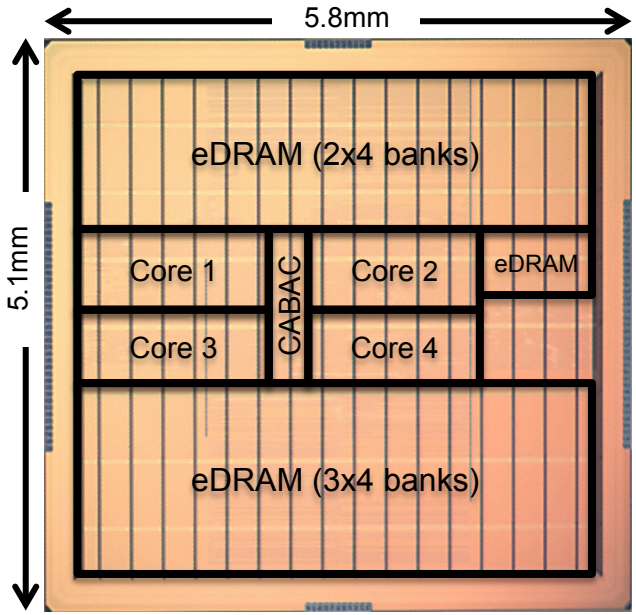
High-Throughput HEVC CABAC

- CABAC bottleneck in H.264/AVC
- **HEVC 4x faster than H.264/AVC**
- Ultra-HD 8K @ 120fps
- Also gives coding gain: 3-5%

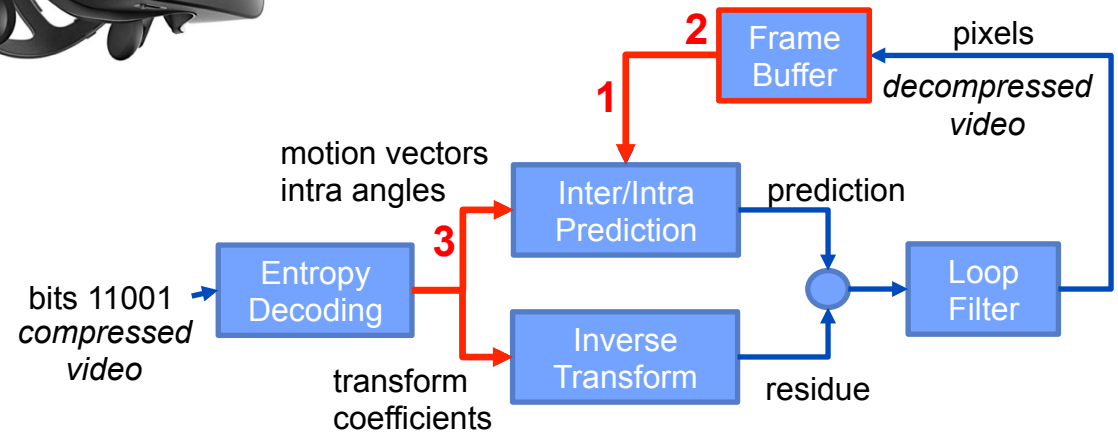


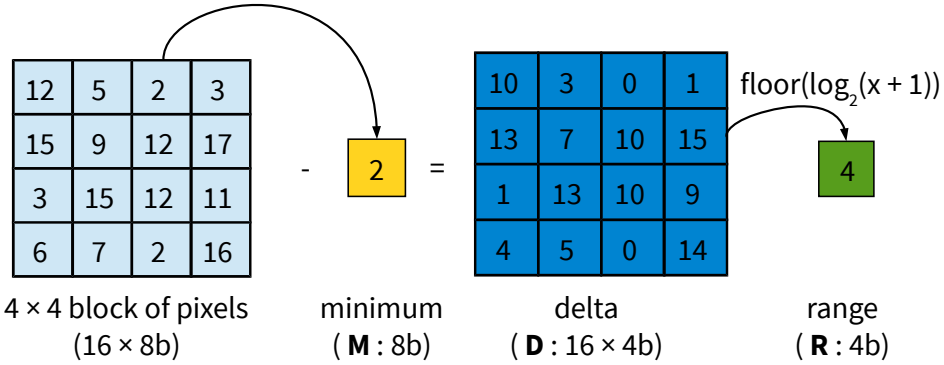
[Y.-H. Chen et al., TCSVT 2015]

Low Power HEVC Decoder for Wearable Devices



Reduce On-chip and Off-chip Data Movement



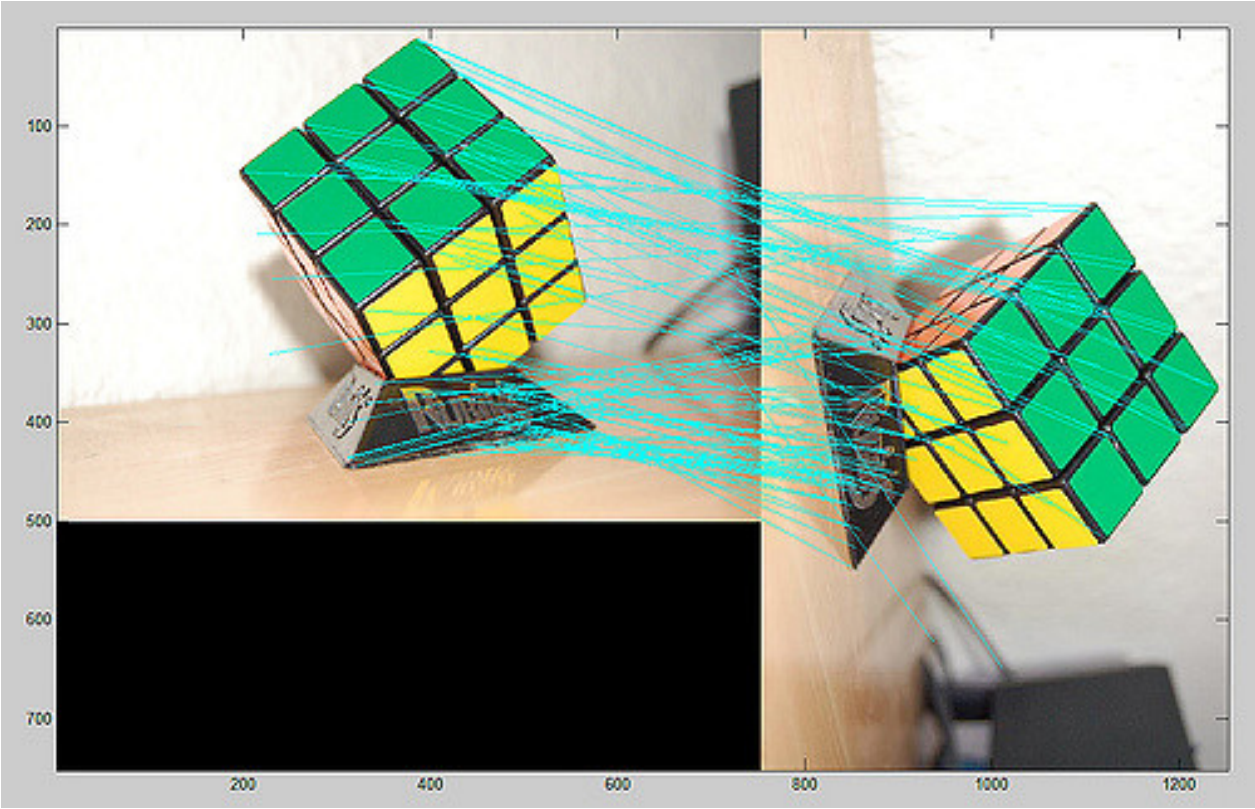


- **Memory consumes 2.8x to 6x** more energy than video decoder
- Use **low cost compression** to reduce storage cost of frame buffer
- **Total decoder system power of 25mW**

[Tikekar et al., VLSI 2017]

Compression Inspired by Computer Vision

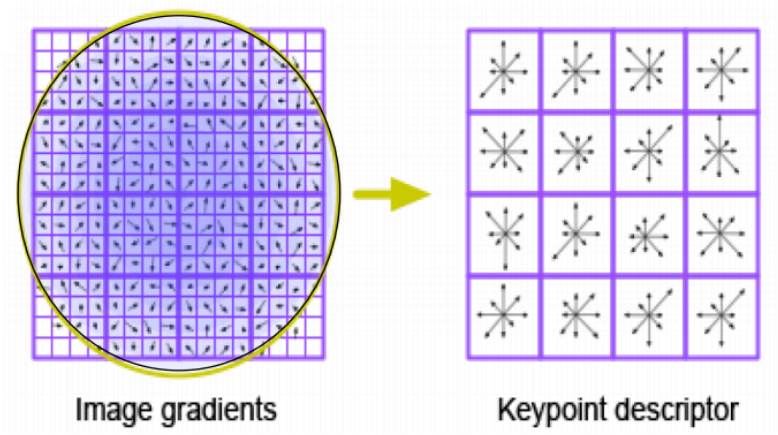
SIFT feature matching



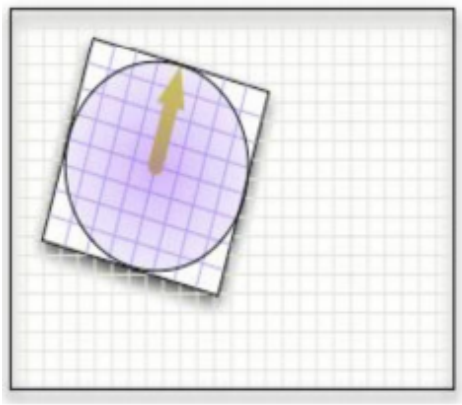
SIFT features are widely used to establish correspondence between two similar images

[DG Lowe, IJCV 2004]

SIFT is Rotate Invariant



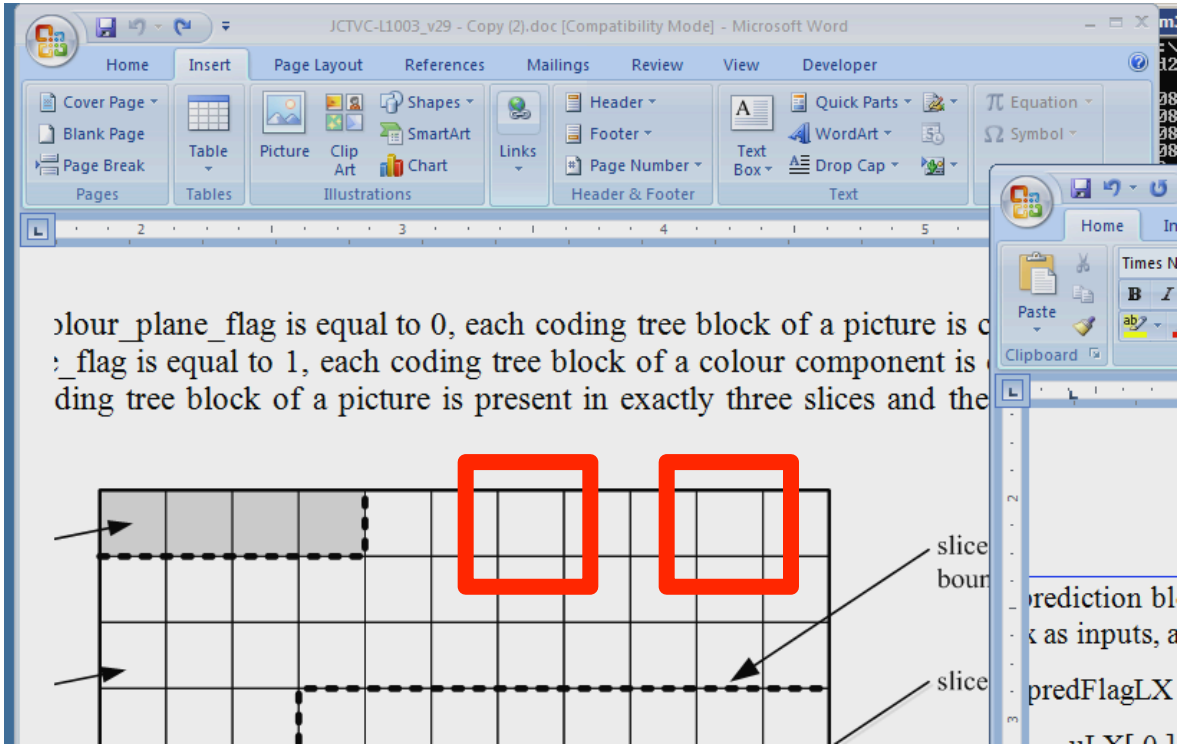
SIFT descriptor



Canonizing the rotation

Before the matching, SIFT descriptors are normalized to the canonical pose (dominant gradient) so that patches of different orientation can be matched.

Intra Block Copy for Still Image Coding

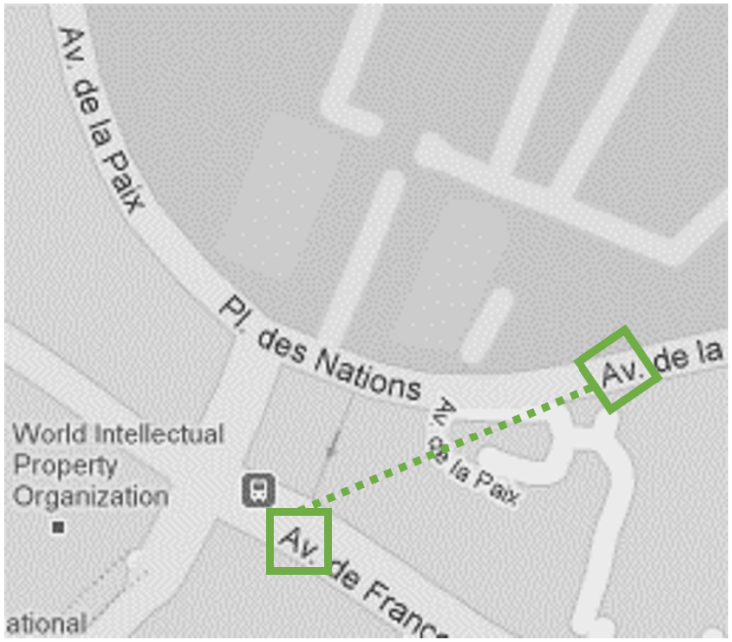


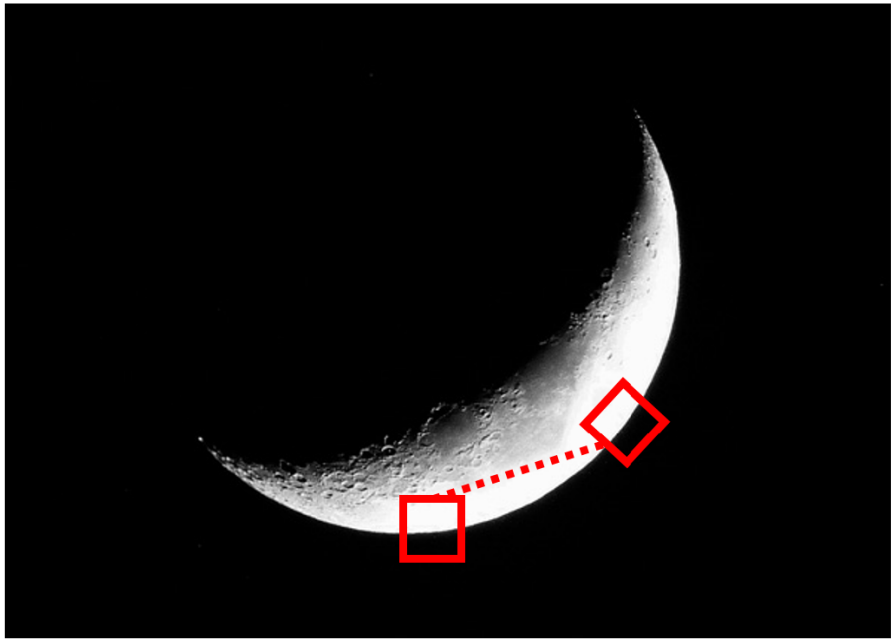
[Yu et al., JVT-C151],
[Budagavi et al., JCTVC-M0350],
[Peng et al, JCTVC-N0256]

Use one block to **predict** repetitive blocks. Only encode the **difference (residual).**

NOT rotate invariant. Limited to screen content.

Rotate Intra Block Copy





Repetitive structures with rotation

In both screen content and camera captured images

Reduction of Residual Energy

HEVC

HEVC + Block Copy

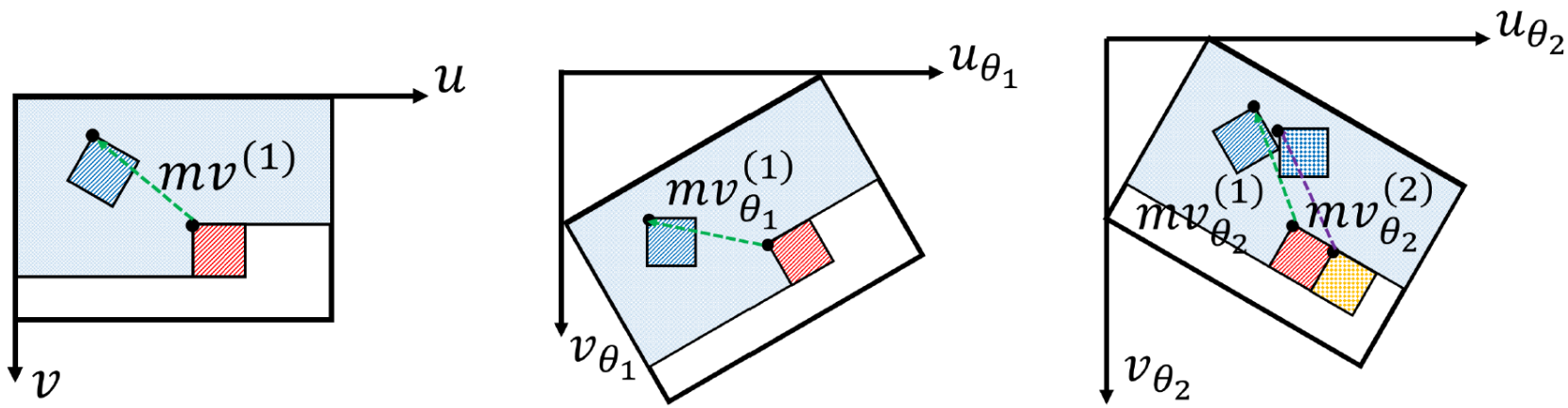
HEVC + Rotate Block Copy

40% reduction of residual energy over HEVC
27% reduction of residual energy over HEVC + Block Copy

However, there is overhead in signaling the rotate angle and motion vector

First frame of *ParkScene* Sequence

Motion Vector Prediction



Motion vectors need to be on the same rotated coordinate system

$mv_{\theta_2}^{(2)}$ **is encoded as** $mv_{\theta_2}^{(2)} - \mathrm{round}\left(R_{\theta_2-\theta_1} mv_{\theta_1}^{(1)}\right)$

Where $R_{\theta_2-\theta_1} = \begin{bmatrix} \cos(\theta_2-\theta_1) & -\sin(\theta_2-\theta_1) \\ \sin(\theta_2-\theta_1) & \cos(\theta_2-\theta_1) \end{bmatrix}$

Reduce average bit rate of motion vector difference by 25%

[Z. Zhang et al., ICIP 2015]

HEVC + Intra Block Copy vs. HEVC + Rotate Intra Block Copy

	Sequence	Residual reduction	BD-rate reduction
Class C	RaceHorse	23.66%	4.75
	PartyScene	27.64%	4.65
	BQMall	17.92%	2.70
	BasketballDrill	22.12%	3.52
Class D	BQSquare	30.82%	5.25
	BasketballPass	15.44%	1.87
	BlowingBubbles	7.59%	2.88
	RaceHorse	28.97%	4.62
Class E	FourPeople	18.09%	2.60
	Johnny	12.79%	2.41
	KristenAndSara	15.67%	2.48
Class F *screen content*	BasketballDrillText	21.15%	3.78
	SlideShow	29.01%	8.03
	SlideEditing	19.12%	0.74
Class C Average		22.83%	3.91
Class D Average		20.70%	3.66
Class E Average		15.52%	2.50
Class F Average		23.09%	4.18
Overall Average		20.71%	3.60

Evaluate on First Frame of JCT-VC test sequences

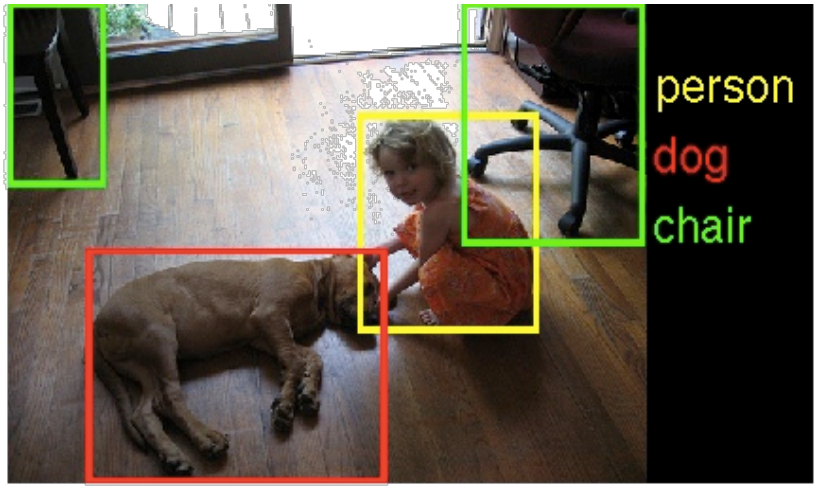
- Residual Energy reduction of 20.7%
- BD-rate savings of 3.6%

[Z. Zhang et al., ICIP 2015]

Energy-Efficient Deep Learning

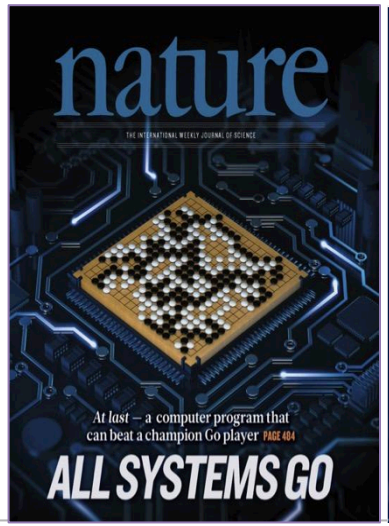
Example Applications of Deep Learning

Computer Vision

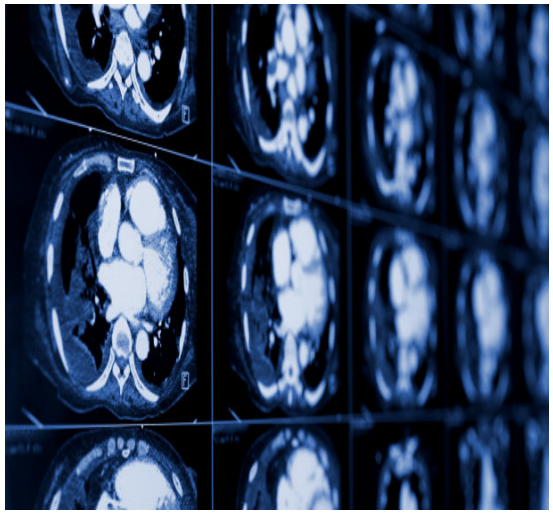


Speech Recognition

Game Play

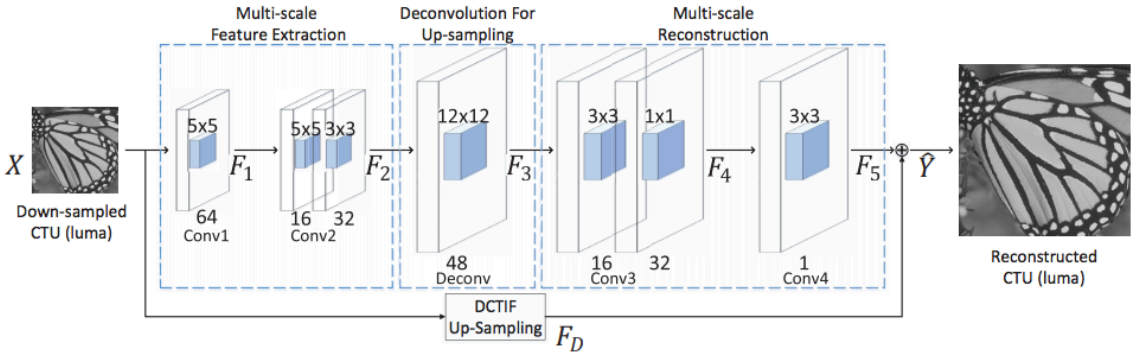


Medical



Using Deep Learning for Compression

Coding Tool in Video Codec

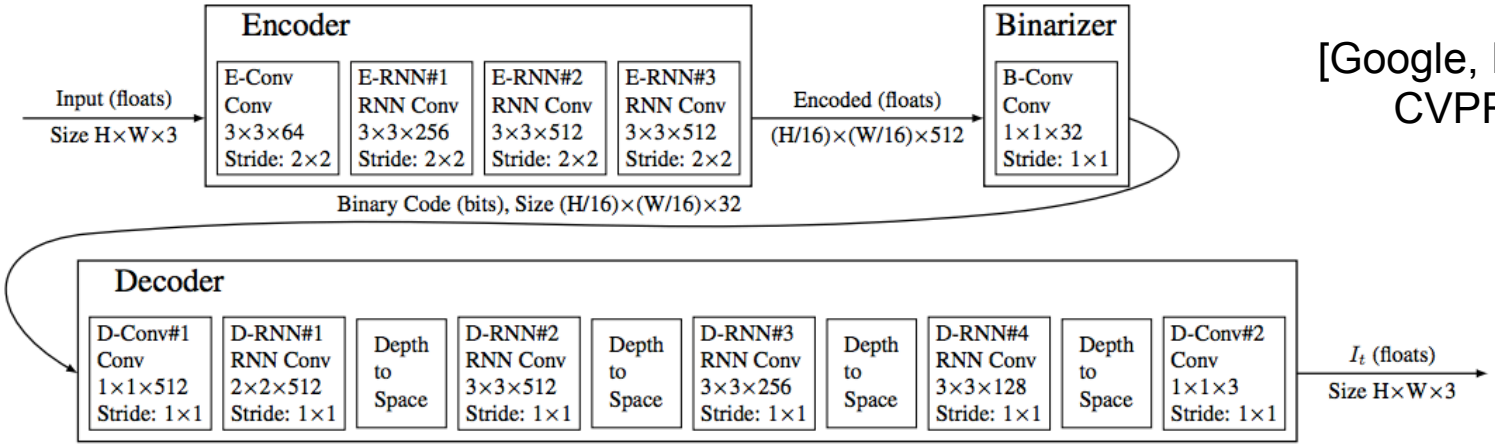


Intra Prediction (Upsampling) [Li et al., TCSVT 2017]

Proposed for VVC: prediction, loop filtering, upsampling, etc.

JVET AHG: Neural Networks in Video Coding

End to End Auto Encoder

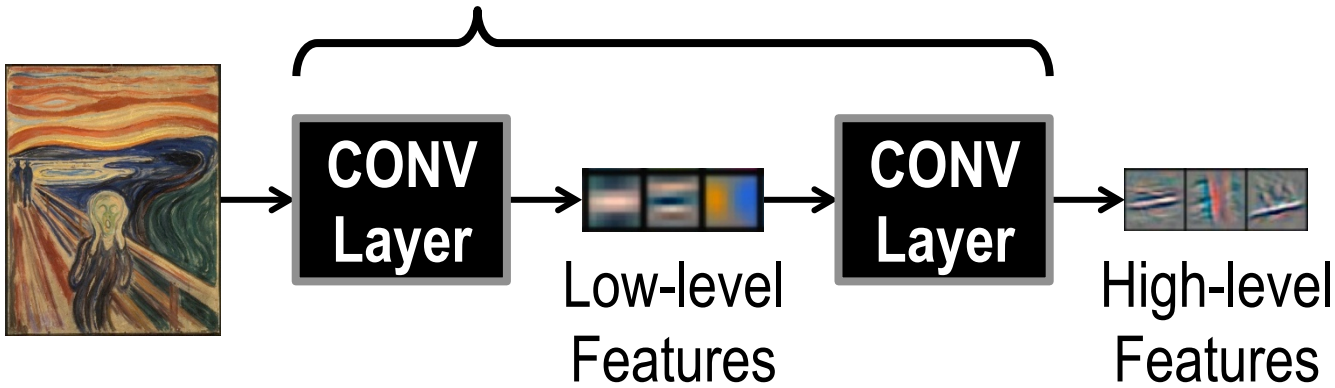


[Google, ICLR 2016, CVPR 2017]

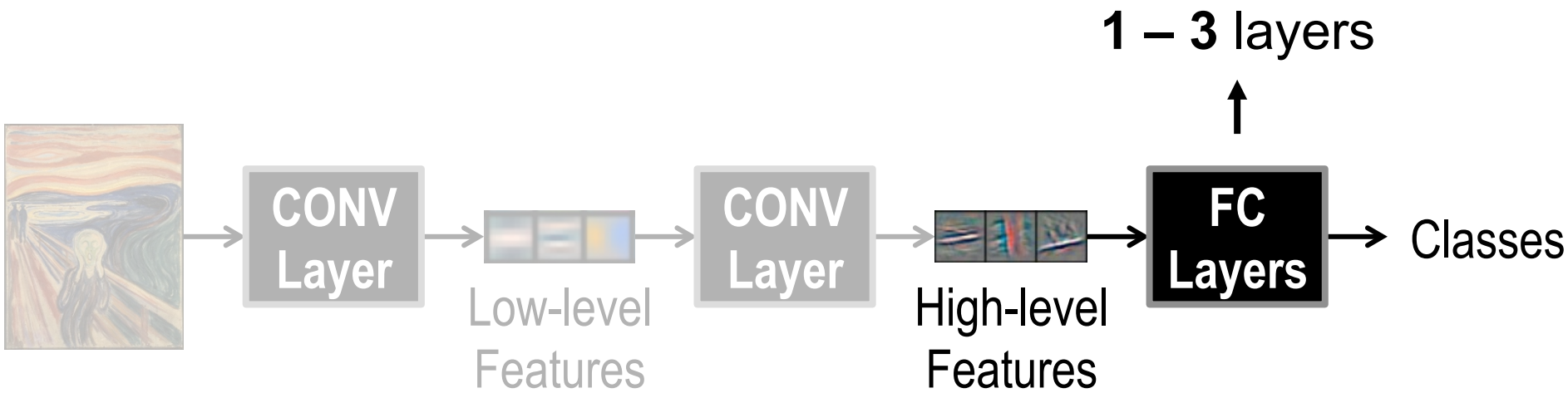
Challenge: Computation complexity higher than typical image processing

Deep Convolutional Neural Networks

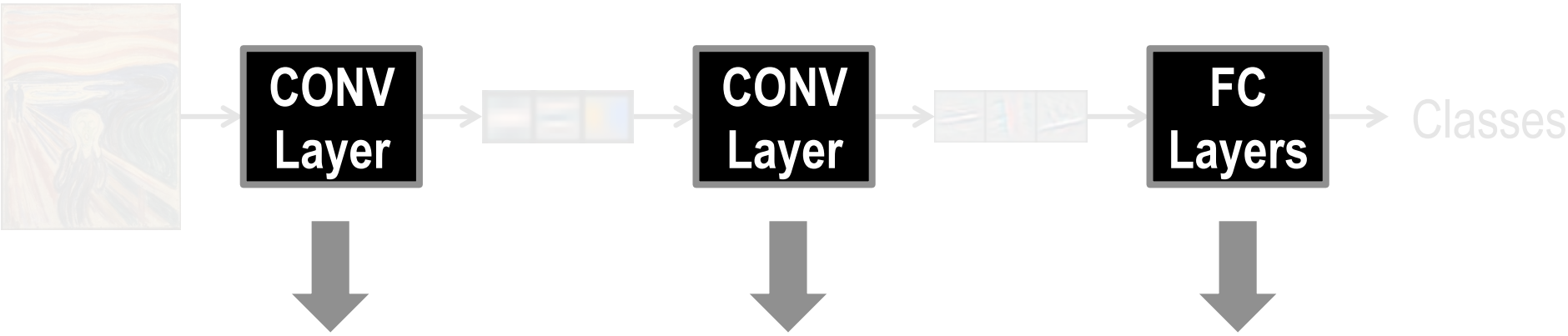
Modern ***deep*** CNN: up to **1000** CONV layers

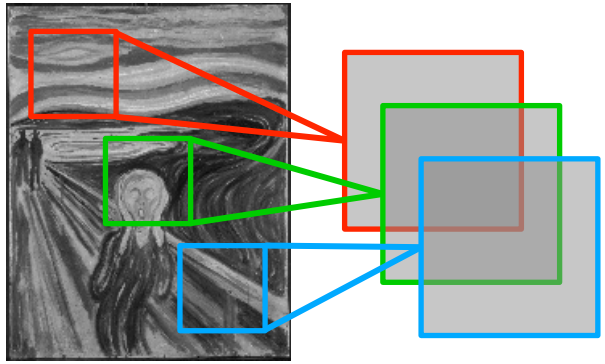


Deep Convolutional Neural Networks



Deep Convolutional Neural Networks



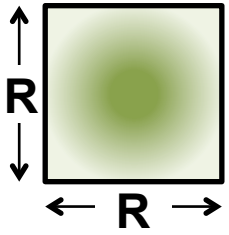


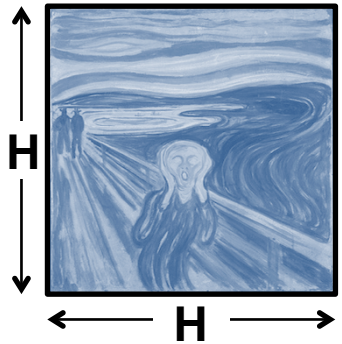
Convolutions account for more than 90% of overall computation, dominating **runtime** and **energy consumption**

High-Dimensional CNN Convolution

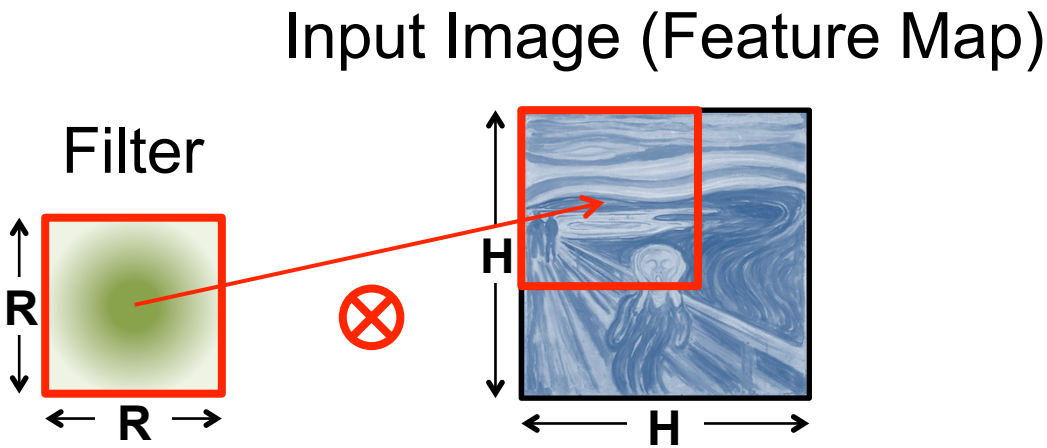
Input Image (Feature Map)

Filter



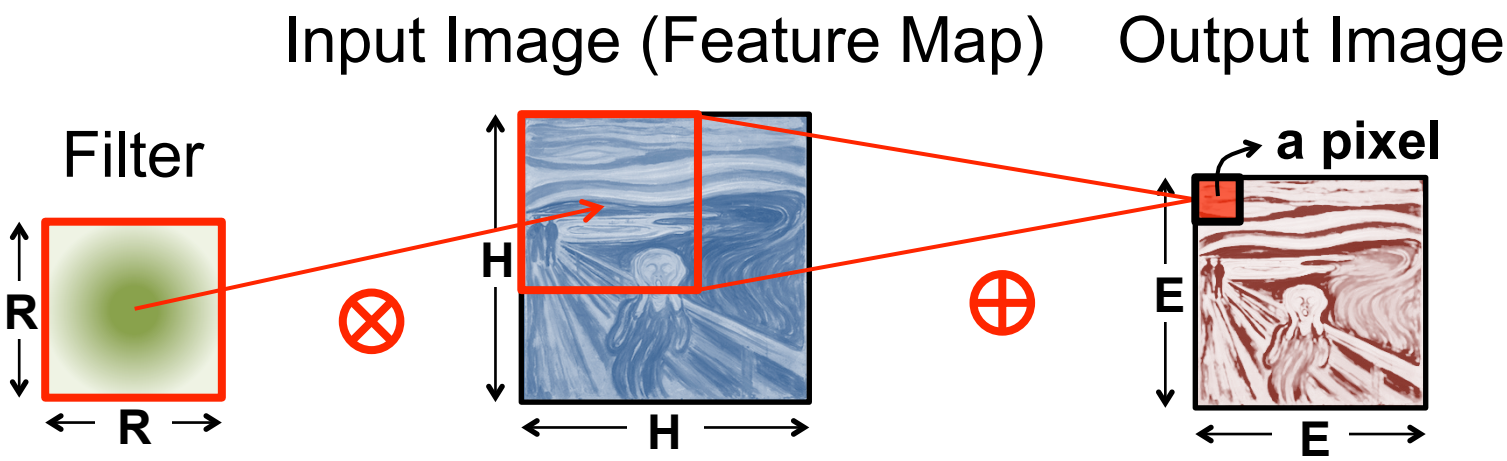


High-Dimensional CNN Convolution



Element-wise Multiplication

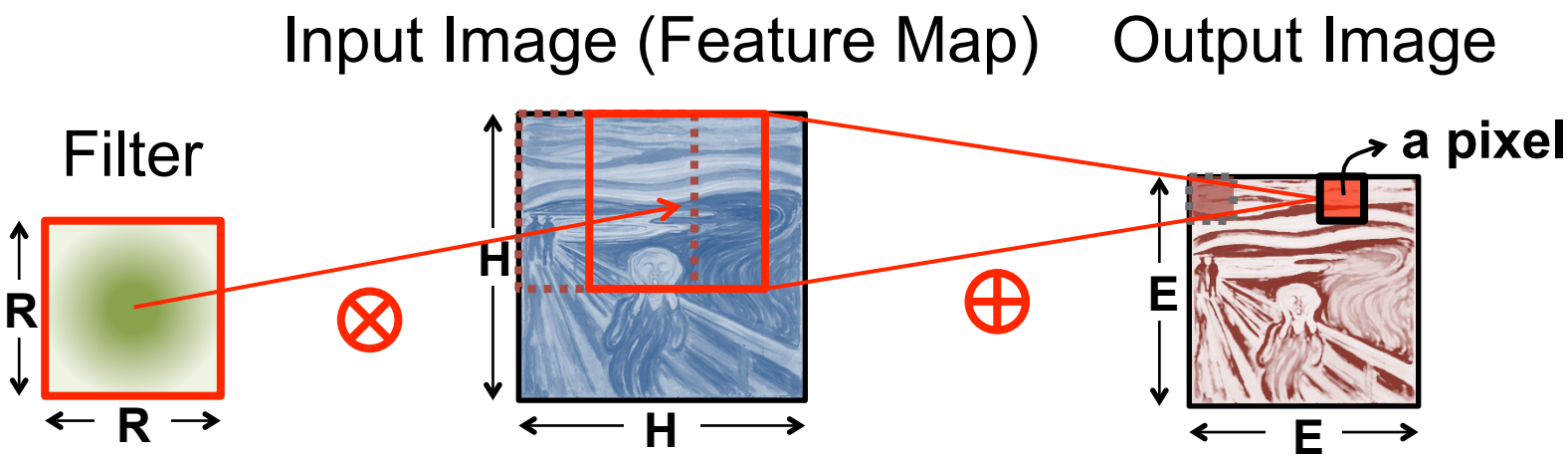
High-Dimensional CNN Convolution



Element-wise Multiplication

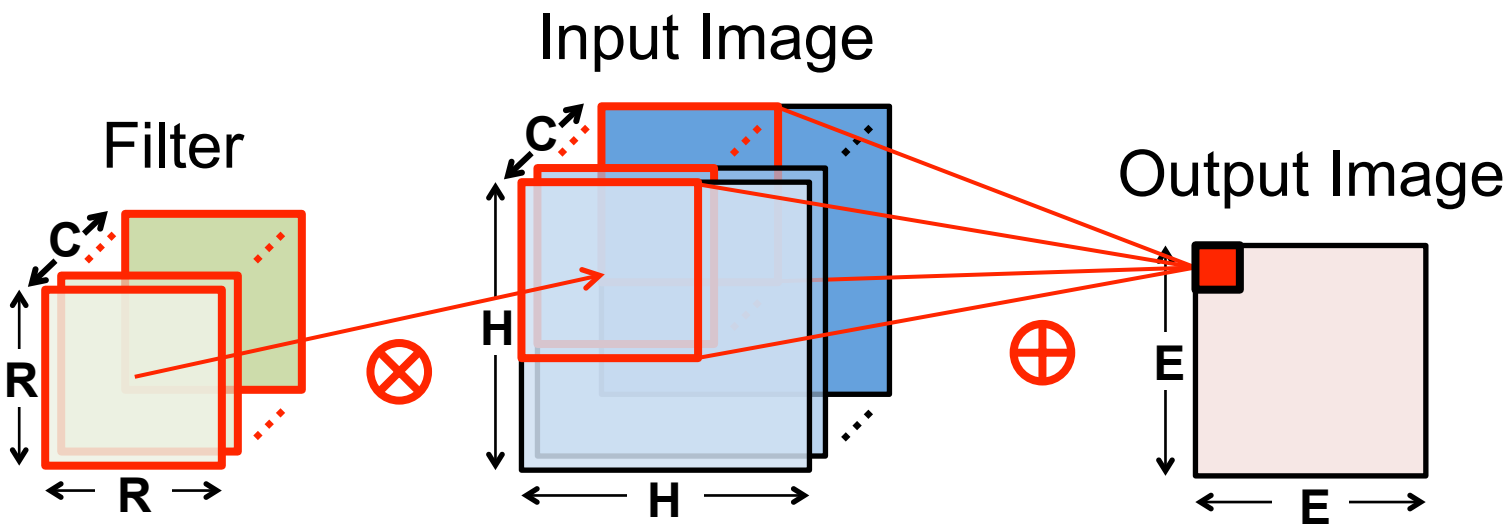
Partial Sum (psum) **Accumulation**

High-Dimensional CNN Convolution



Sliding Window Processing

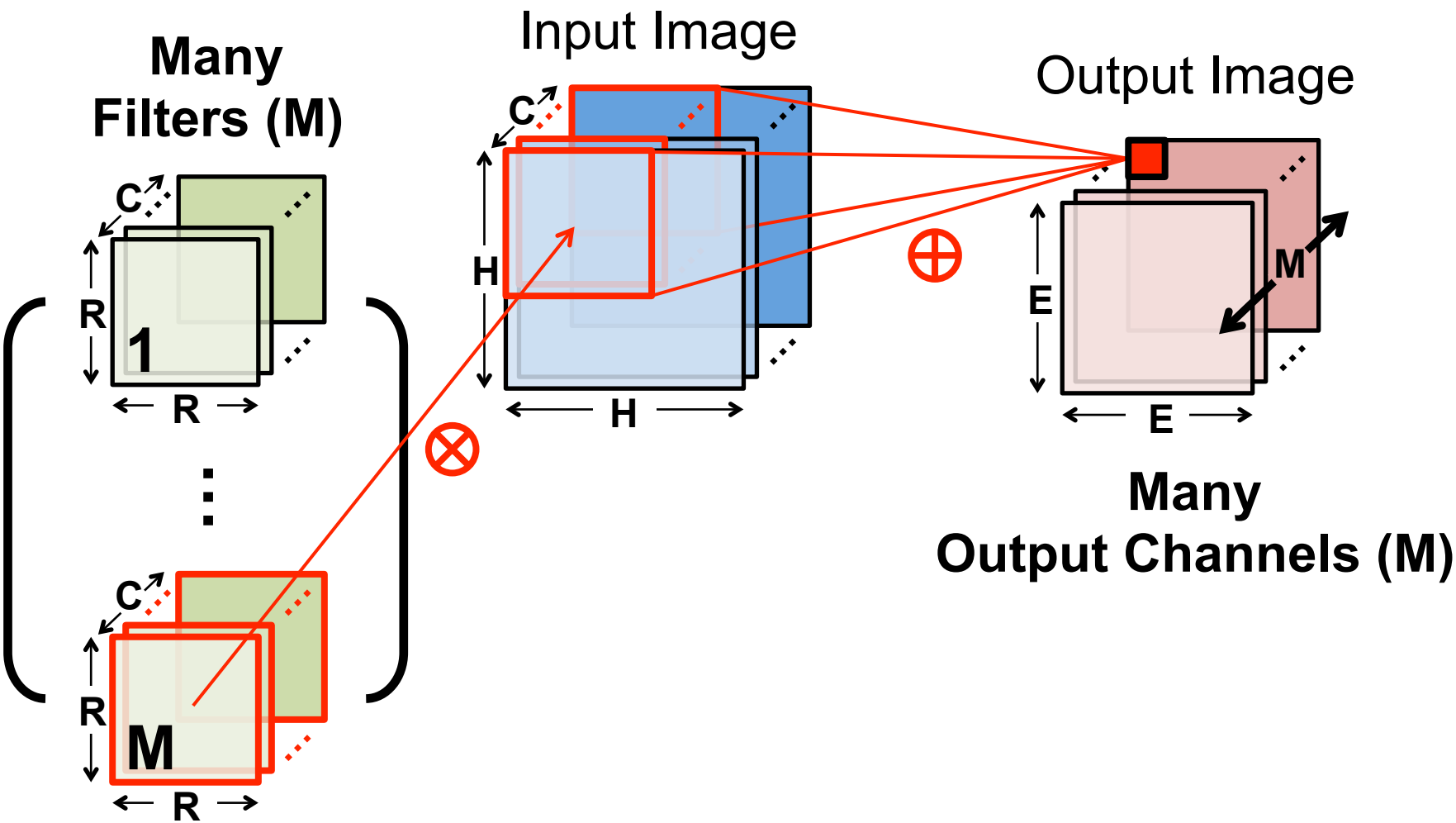
High-Dimensional CNN Convolution



Many Input Channels (C)

AlexNet: 3 – 192 Channels (C)

High-Dimensional CNN Convolution



AlexNet: 96 – 384 Filters (M)

High-Dimensional CNN Convolution

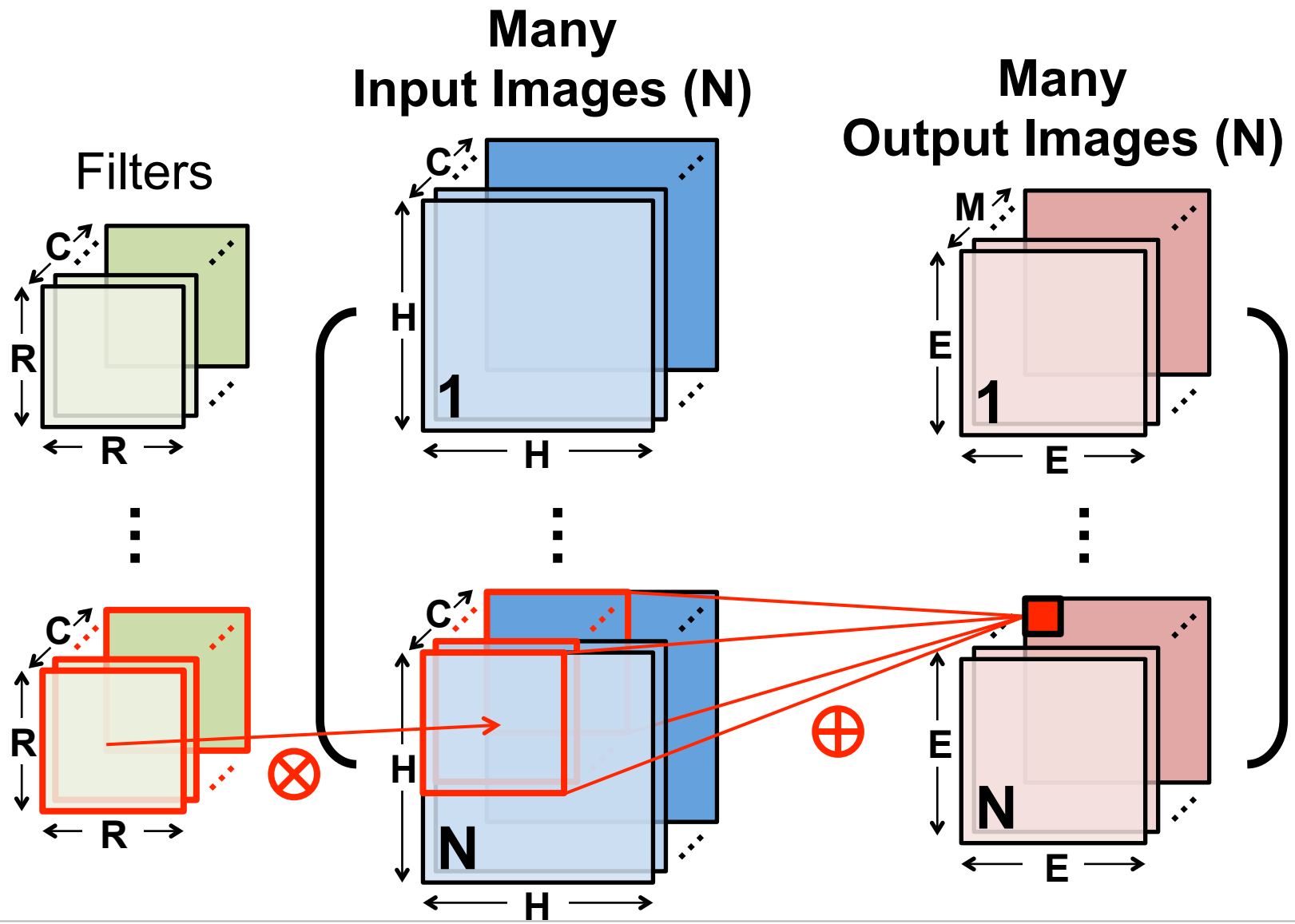


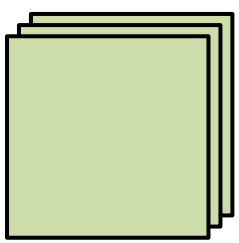
Image batch size: 1 – 256 (N)

Large Sizes with Varying Shapes

AlexNet[1] Convolutional Layer Configurations

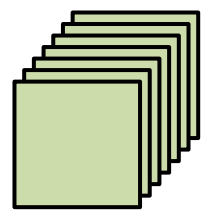
Layer	Filter Size (R)	# Filters (M)	# Channels (C)	Stride
1	11x11	96	3	4
2	5x5	256	48	1
3	3x3	384	256	1
4	3x3	384	192	1
5	3x3	256	192	1

Layer 1



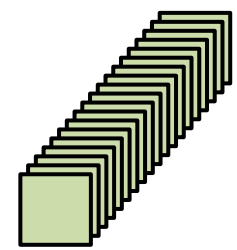
34k Params
105M MACs

Layer 2



307k Params
224M MACs

Layer 3



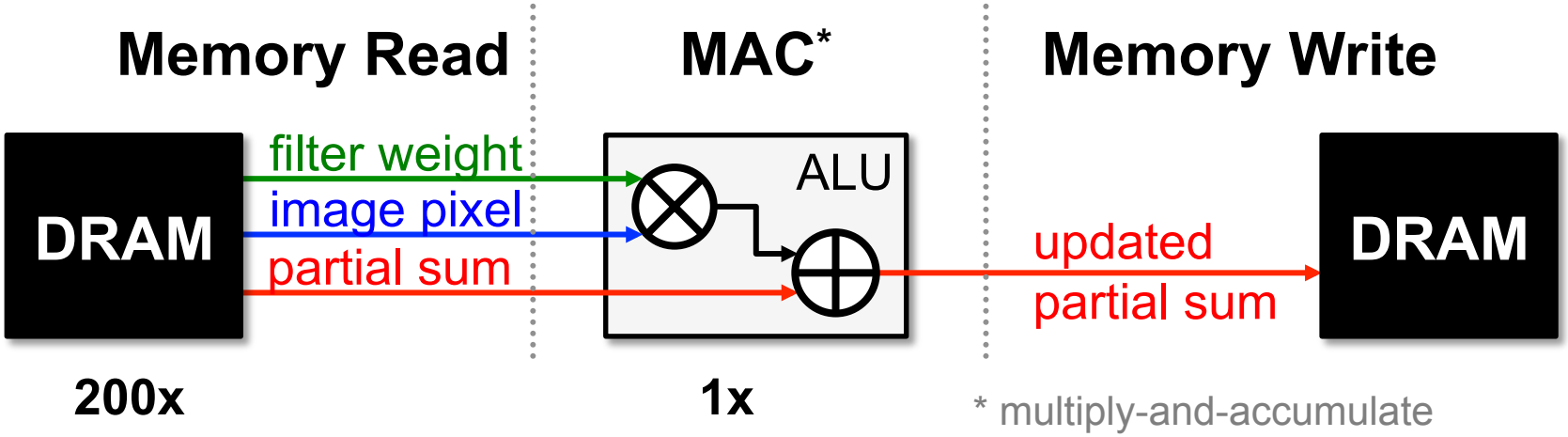
885k Params
150M MACs

1. [Krizhevsky, NIPS 2012]

Properties We Can Leverage

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

- Memory Access is the Bottleneck

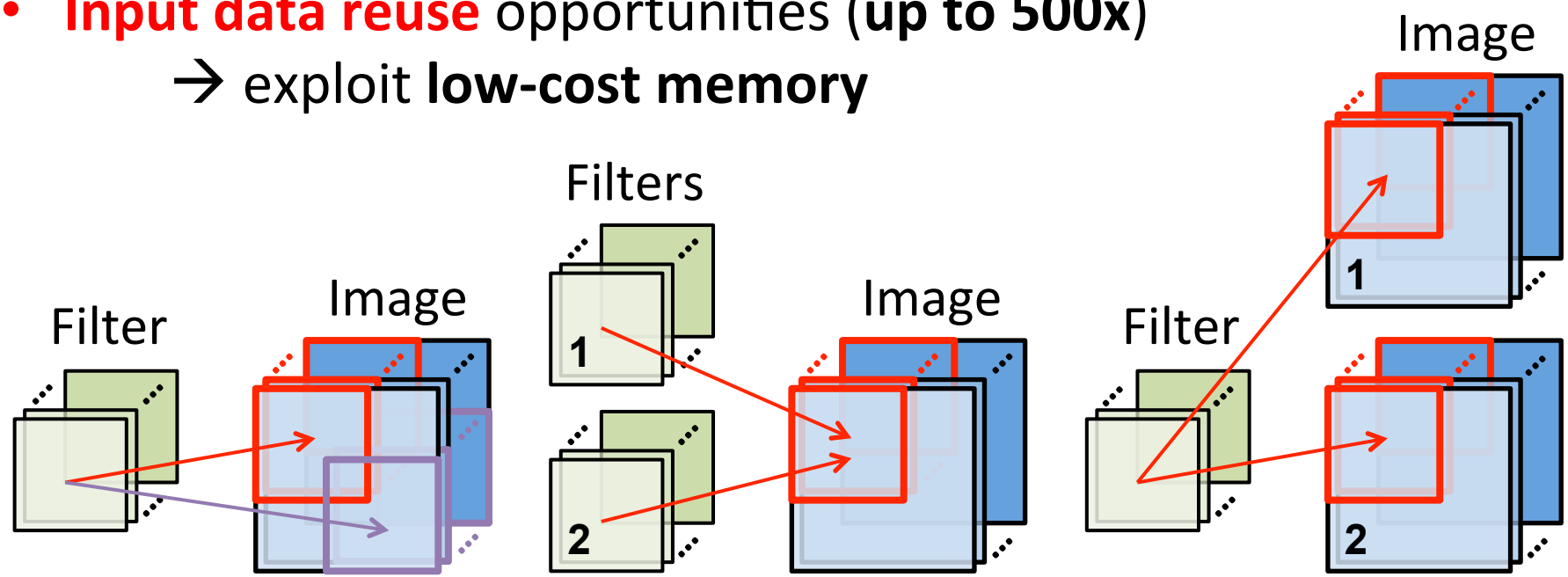


<u>Worst Case</u>: all memory R/W are **DRAM** accesses

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

Properties We Can Leverage

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



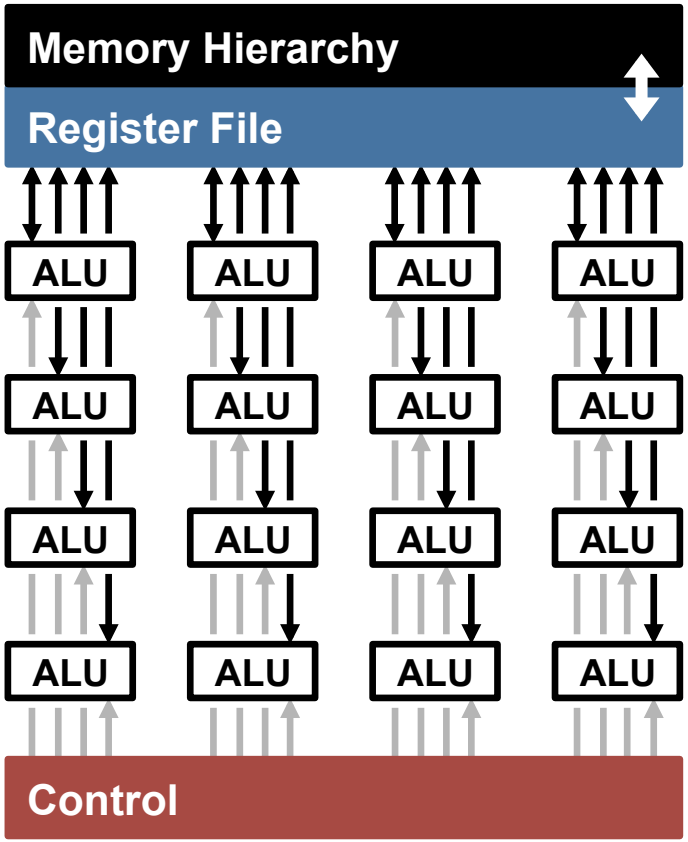
Convolutional Reuse
(pixels, weights)

Image Reuse
(pixels)

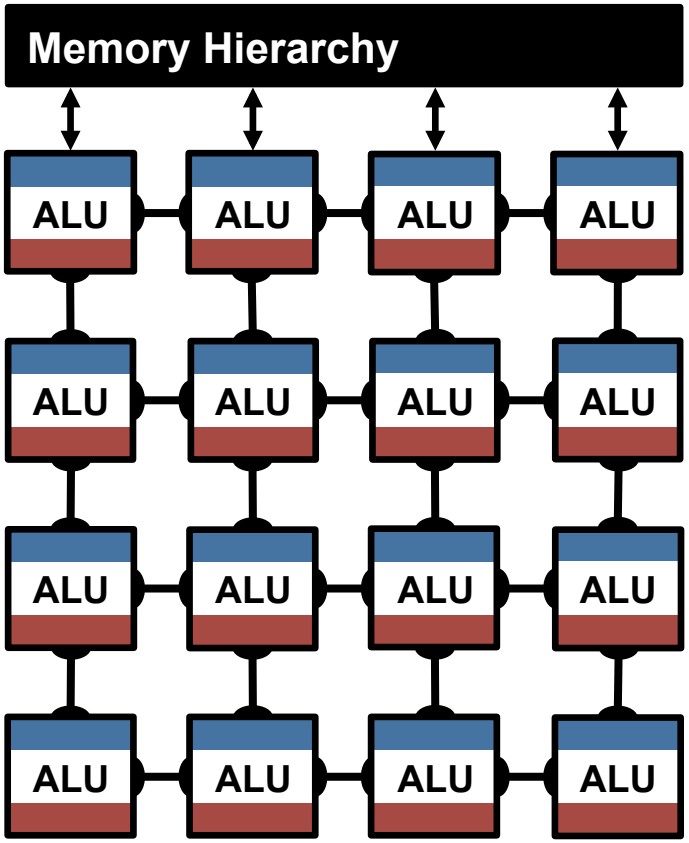
Filter Reuse
(weights)

Highly-Parallel Compute Paradigms

Temporal Architecture (SIMD/SIMT)



Spatial Architecture (Dataflow Processing)



Advantages of Spatial Architecture

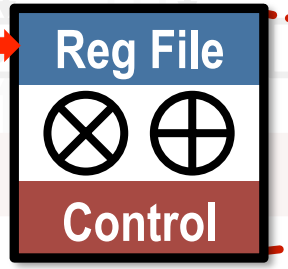
Temporal Architecture
(SIMD/SIMT)

Efficient Data Reuse
Distributed local storage (RF)

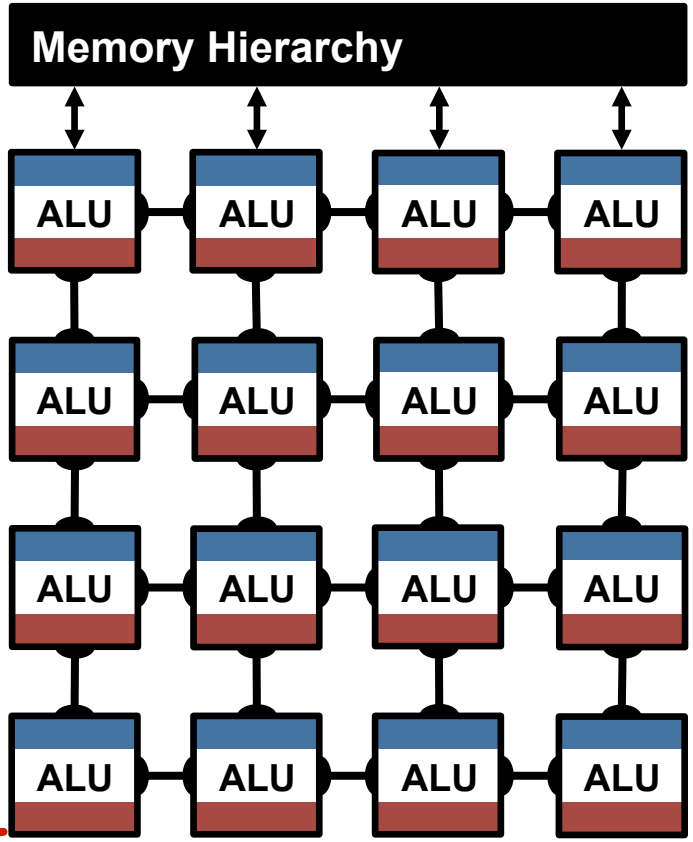
Inter-PE Communication
Sharing among regions of PEs

Processing Element (PE)

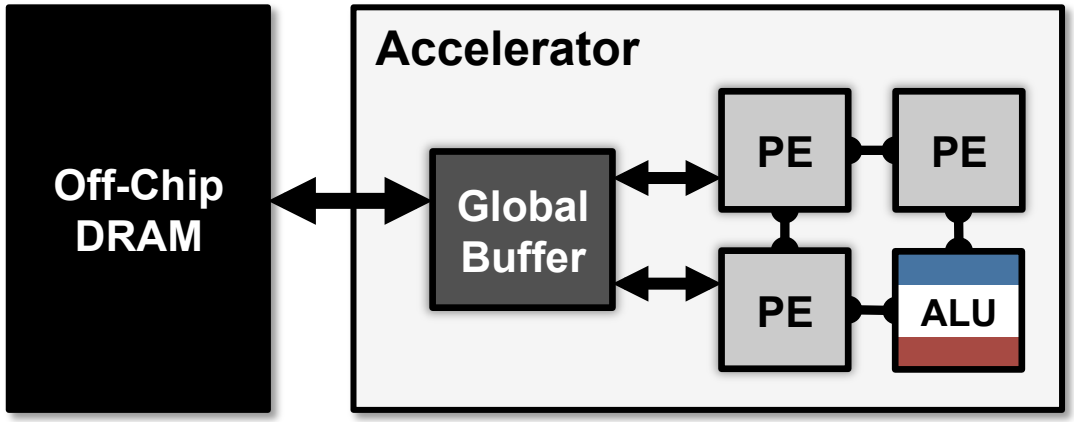
0.5 – 1.0 kB

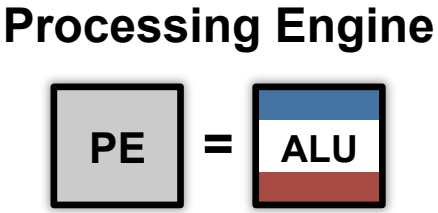


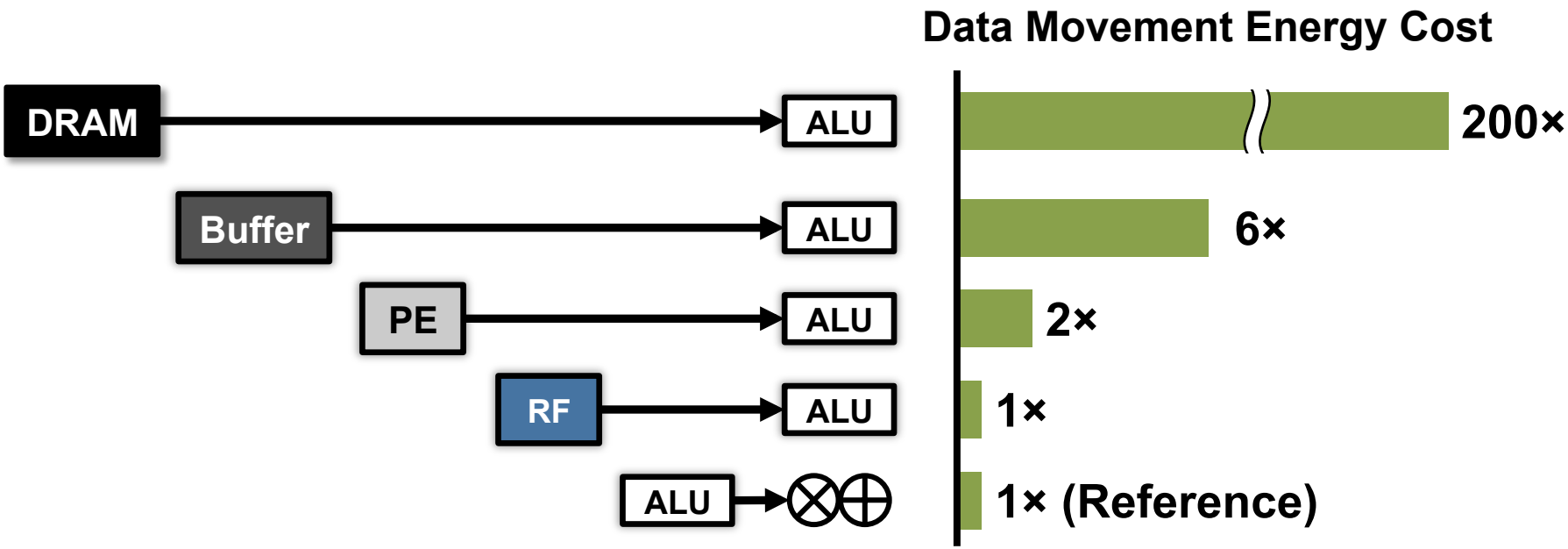
Spatial Architecture (Dataflow Processing)



Data Movement is Expensive



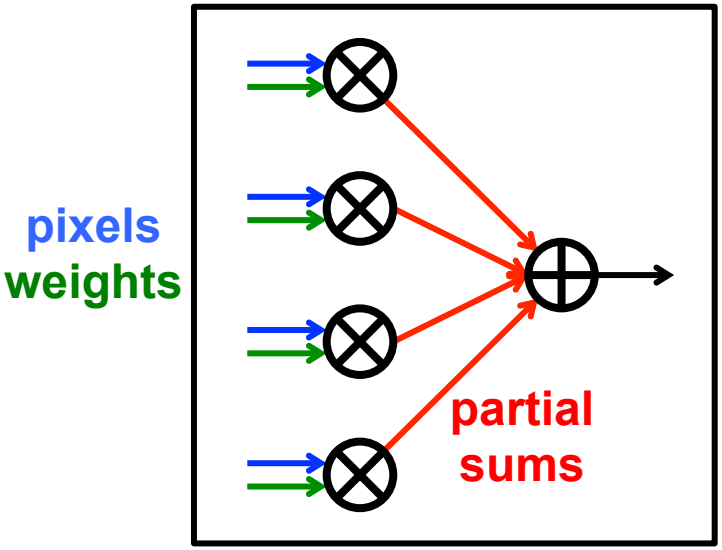




Maximize data reuse at lower levels of hierarchy

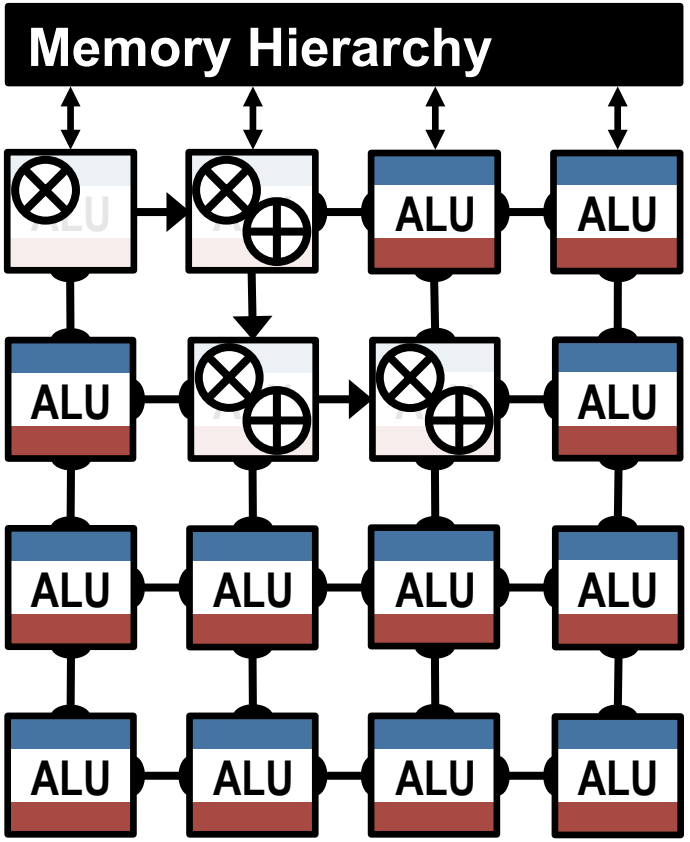
How to Map the Dataflow?

CNN Convolution



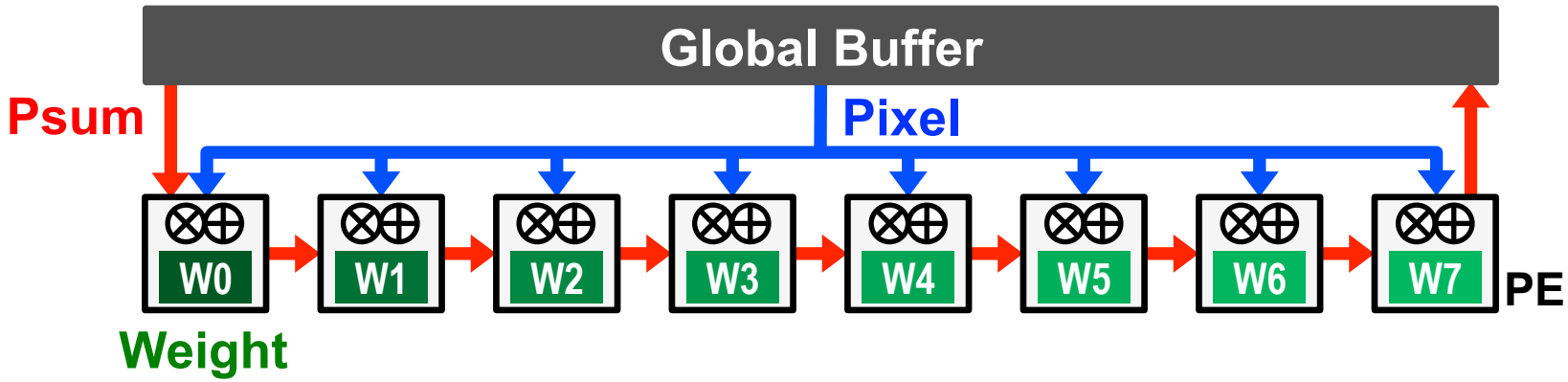
?

Spatial Architecture (Dataflow Processing)



Goal: Increase reuse of input data (**weights** and **pixels**) and local **partial sums** accumulation

Weight Stationary (WS)

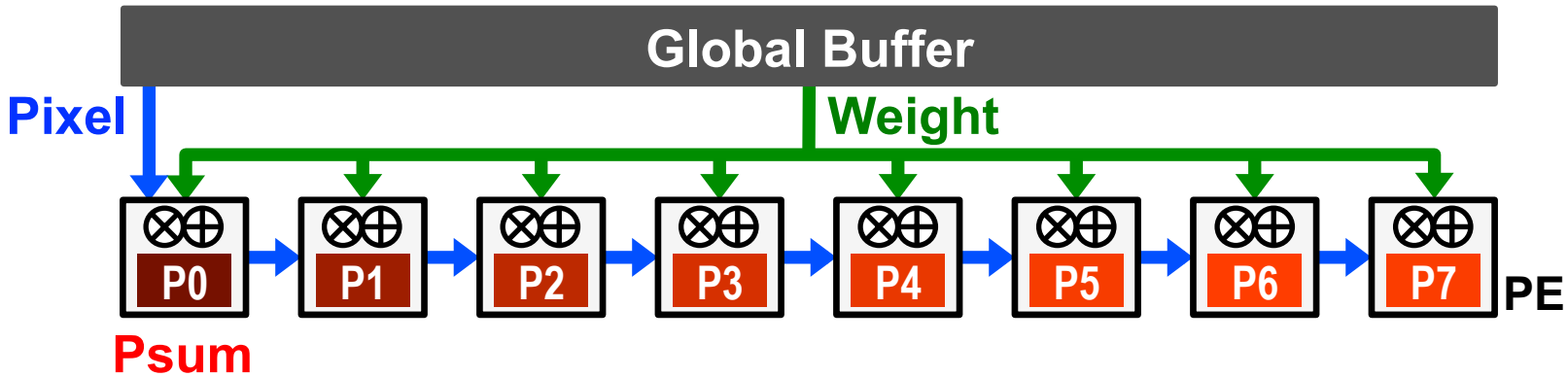


- **Minimize weight** read energy consumption
  - maximize convolutional and filter reuse of weights
- **Examples:**

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

[**Park**, *ISSCC* 2015] [**Google's TPU**, *ISCA* 2017]

Output Stationary (OS)



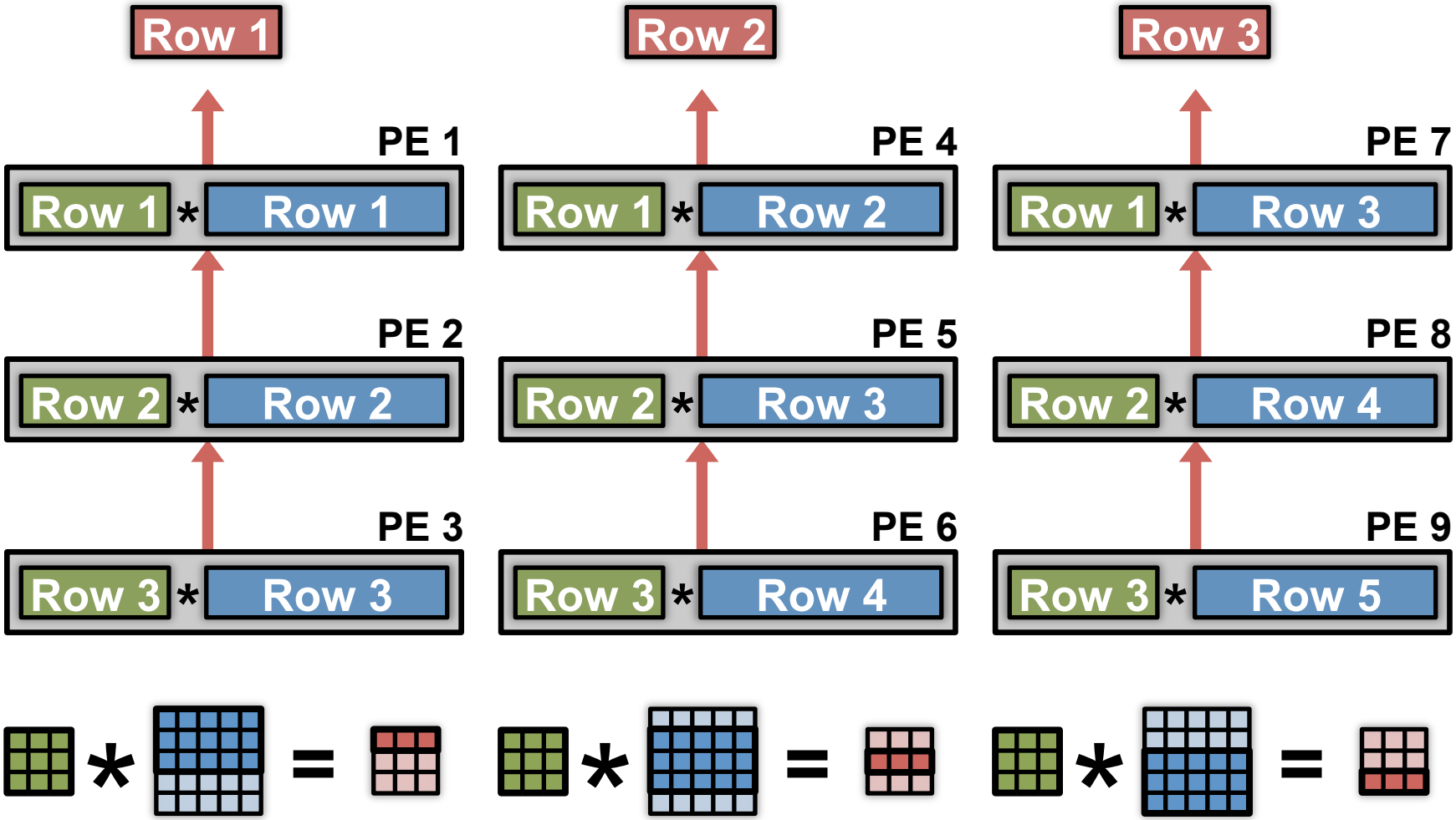
- **Minimize partial sum** R/W energy consumption
  - maximize local accumulation
- **Examples:**

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

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

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

Row Stationary Dataflow

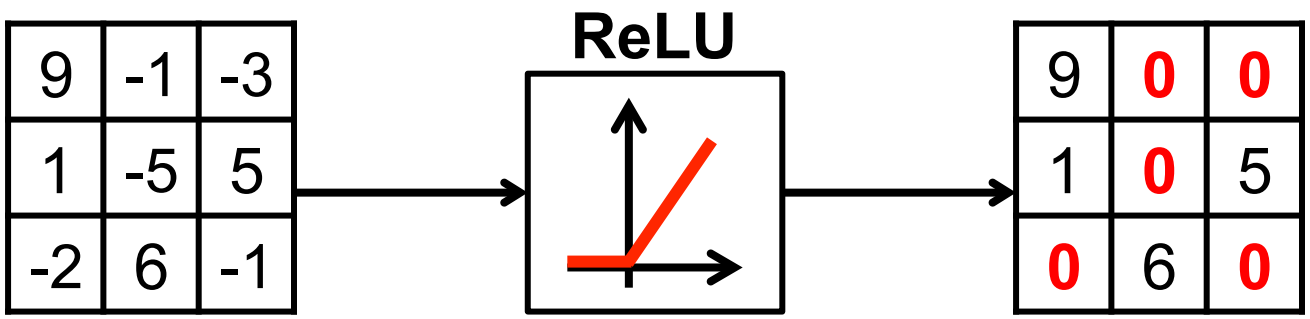


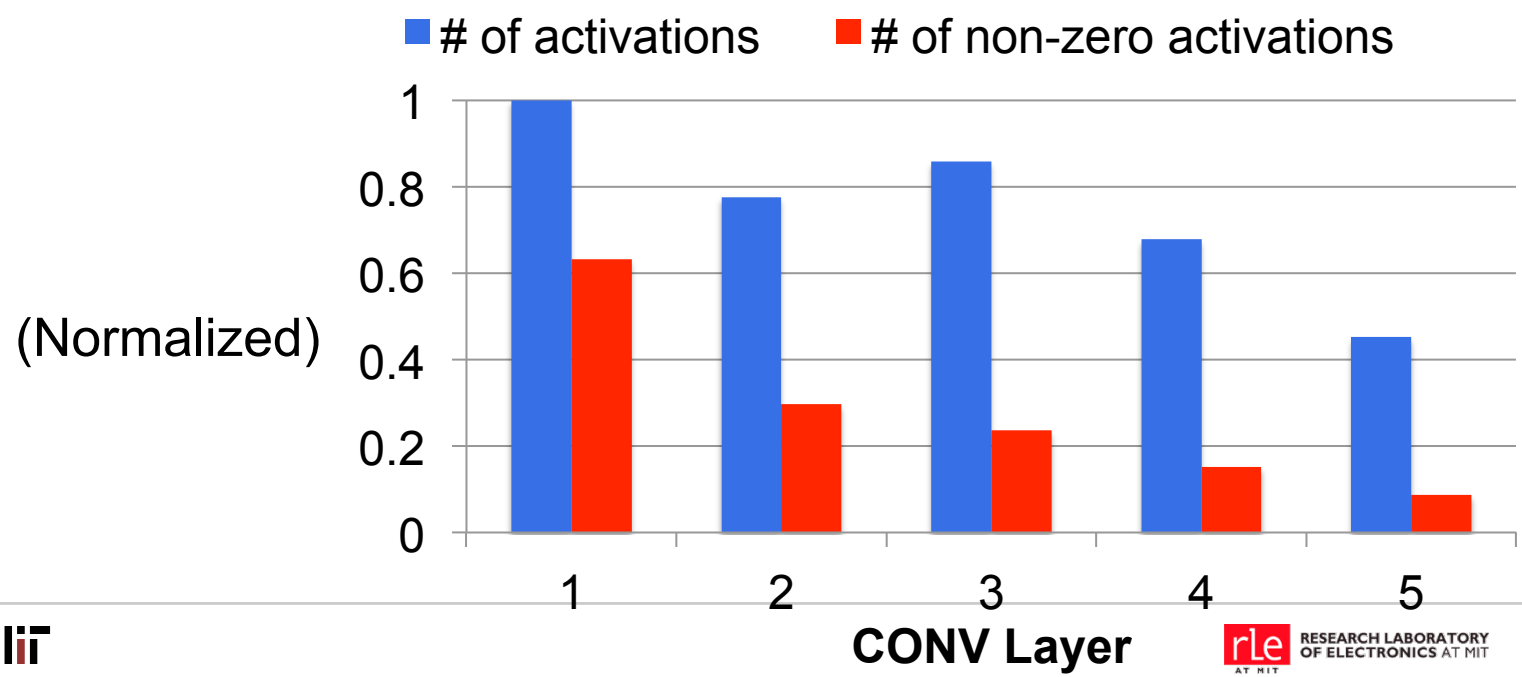
Optimize for **overall energy efficiency** instead for only a certain data type for up to 2.5x energy savings

[Chen et al., Eyeriss, ISCA 2016]

Sparsity in Data

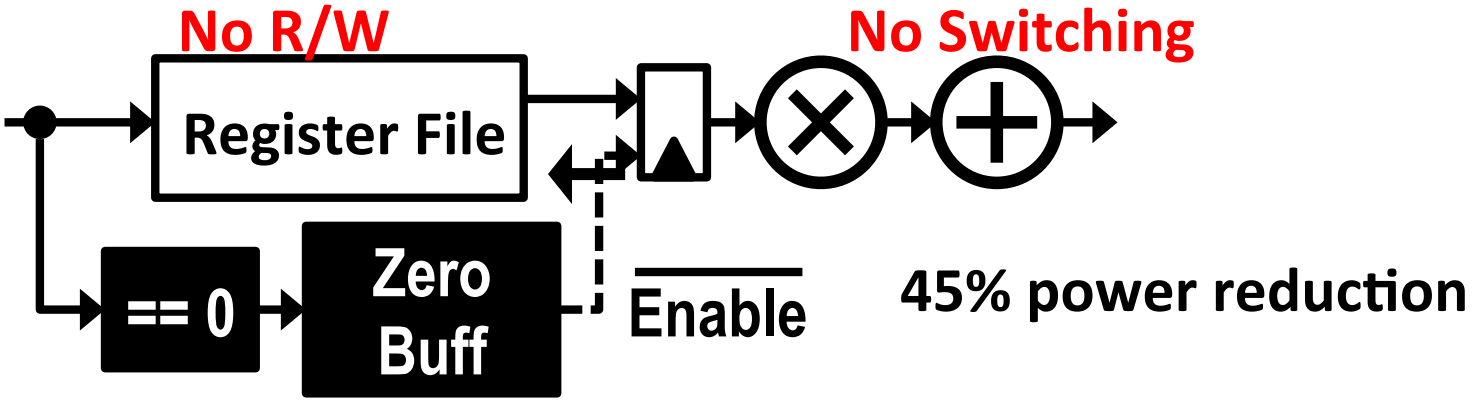
Many **zeros** in **output fmaps** after **ReLU**



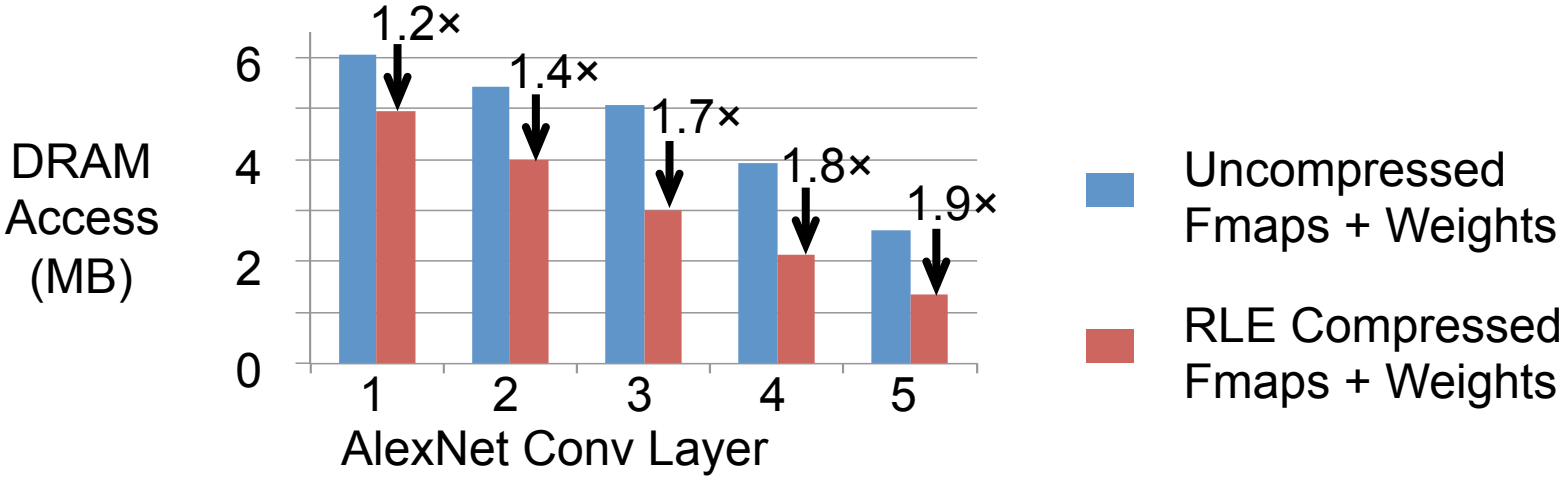


Exploit Sparsity

Method 1. Skip memory access and computation

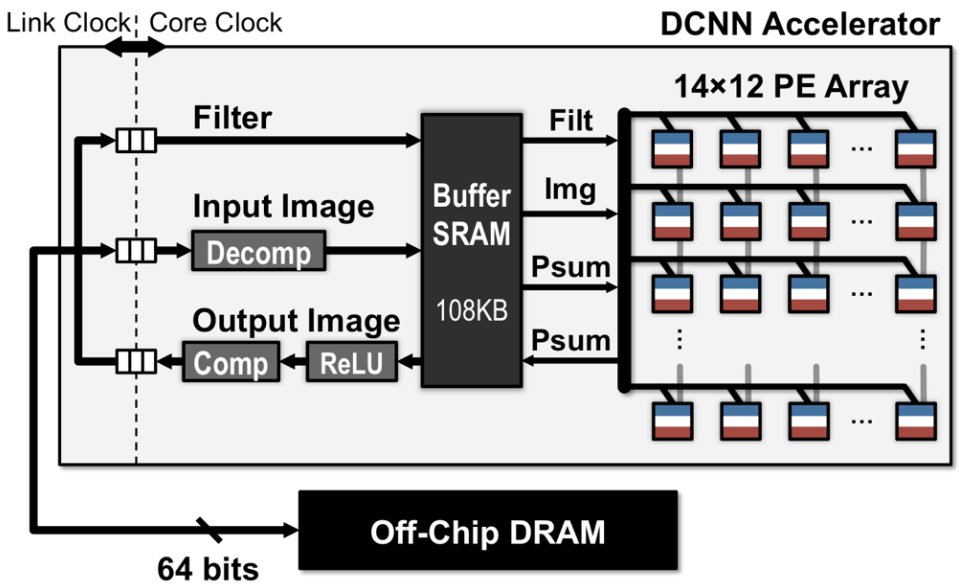


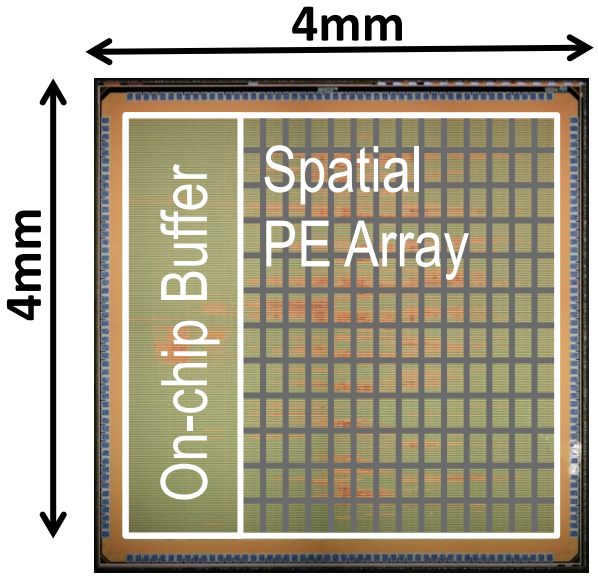
Method 2. Compress data to reduce storage and data movement



[Chen et al., ISSCC 2016]

Eyeriss: Energy-Efficient Deep Learning



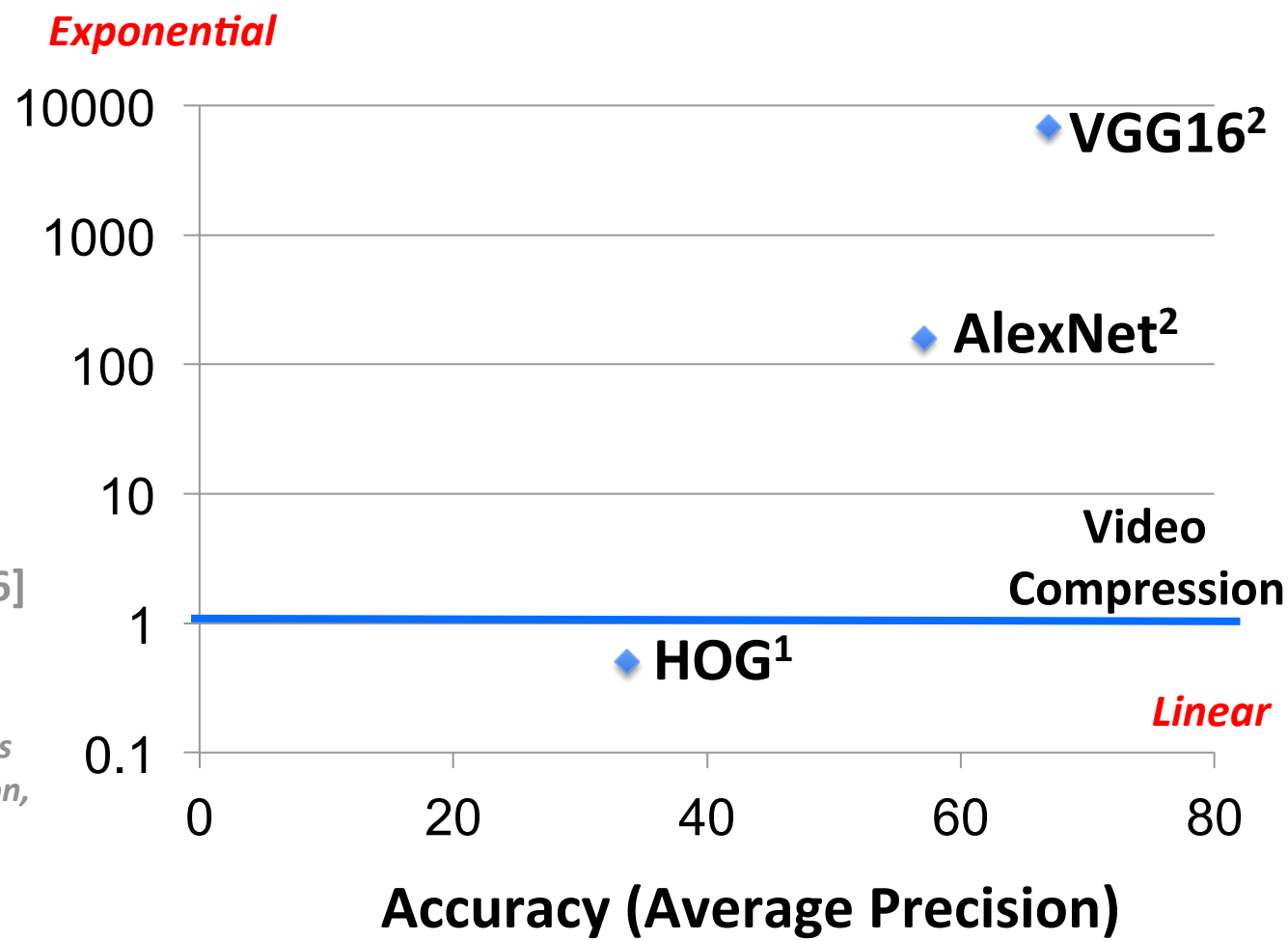


MIT Deep Learning Chip
[ISSCC 2016, ISCA 2016]

278mW for AlexNet @ 30fps (batch size 4) in 65nm LP CMOS

> 10x more energy-efficient than mobile GPUs

Features: Energy vs. Accuracy



Energy/ Pixel (nJ)

*Measured in 65nm**

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

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

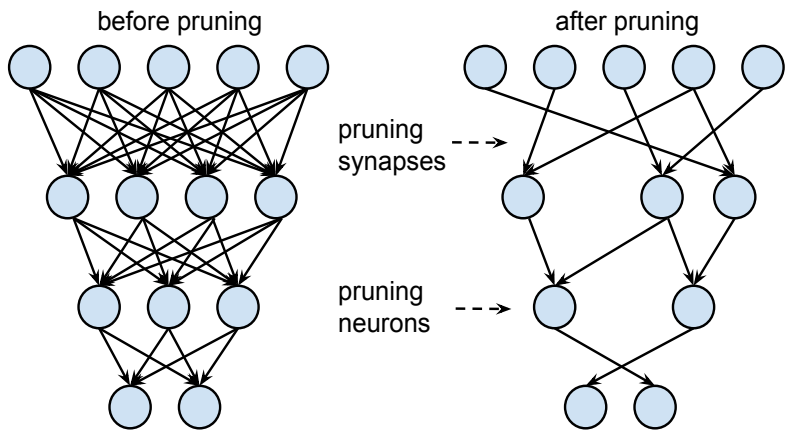
Measured in on VOC 2007 Dataset

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

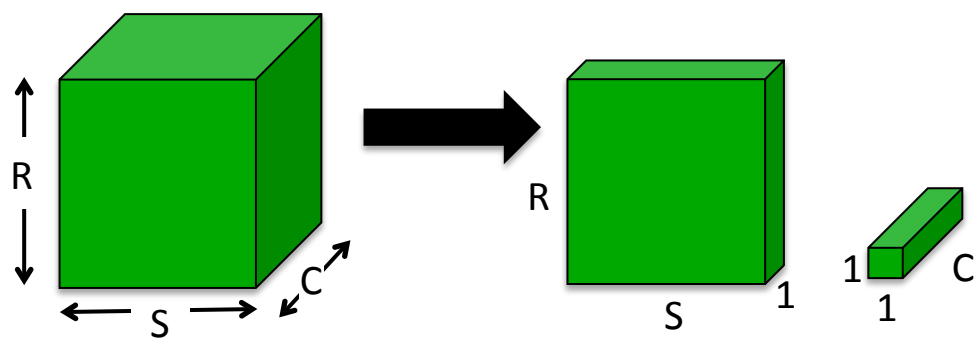
Design of Efficient DNN Algorithms

- Popular efficient DNN algorithm approaches

Network Pruning



Compact Network Architectures



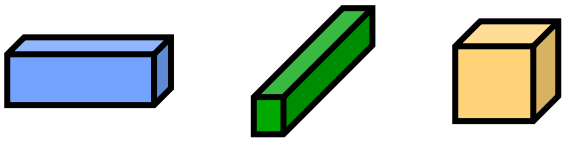
Examples: SqueezeNet, MobileNet

... also reduced precision

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

[Chen et al., SysML 2018]

Energy-Evaluation Methodology



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

Hardware Energy Costs of each MAC and Memory Access

Memory Accesses Optimization

# of MACs Calculation

acc. at mem. level **1**

acc. at mem. level **2**

⋮

acc. at mem. level **n**

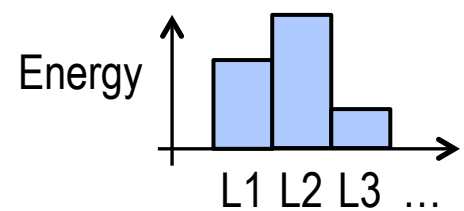
of MACs

E_{data}

E_{comp}

CNN Weights and Input Data

[0.3, 0, -0.4, 0.7, 0, 0, 0.1, …]

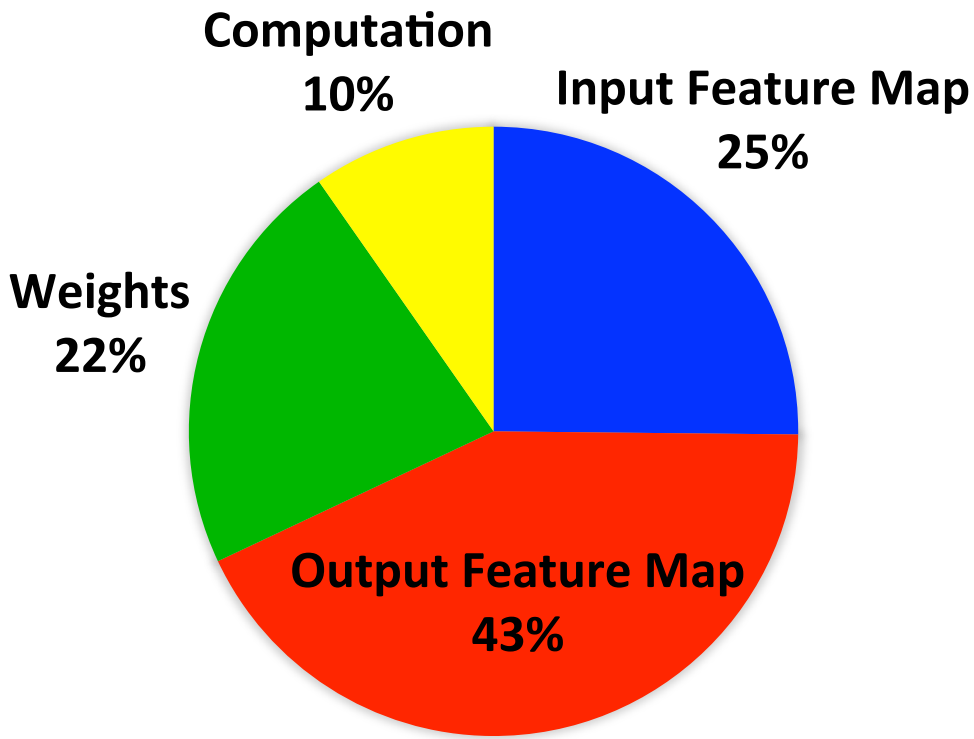


CNN Energy Consumption

Key Observations

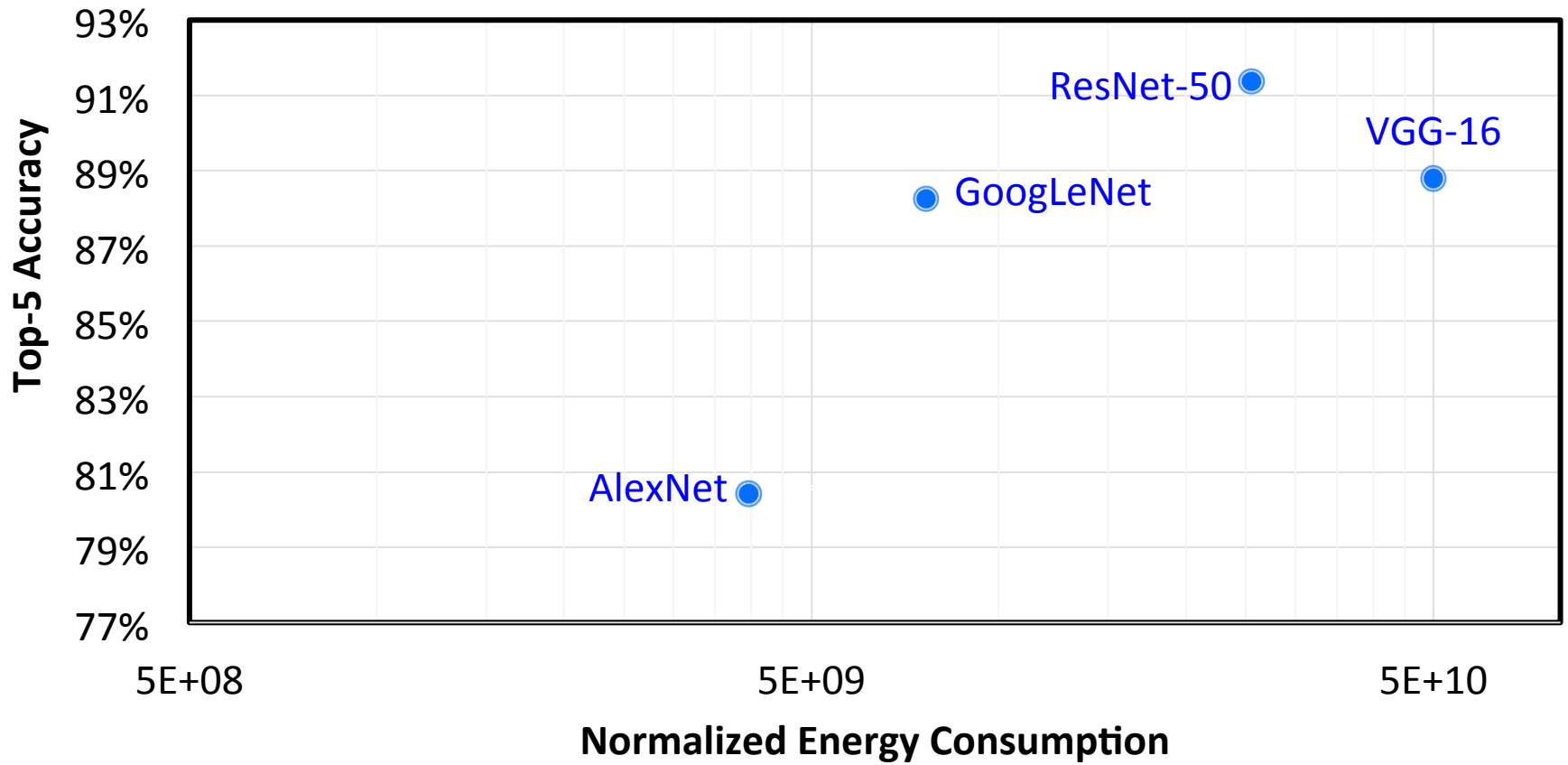
- Number of weights ***alone*** is not a good metric for energy
- **All data types** should be considered

Energy Consumption of GoogLeNet



[Yang et al., CVPR 2017]

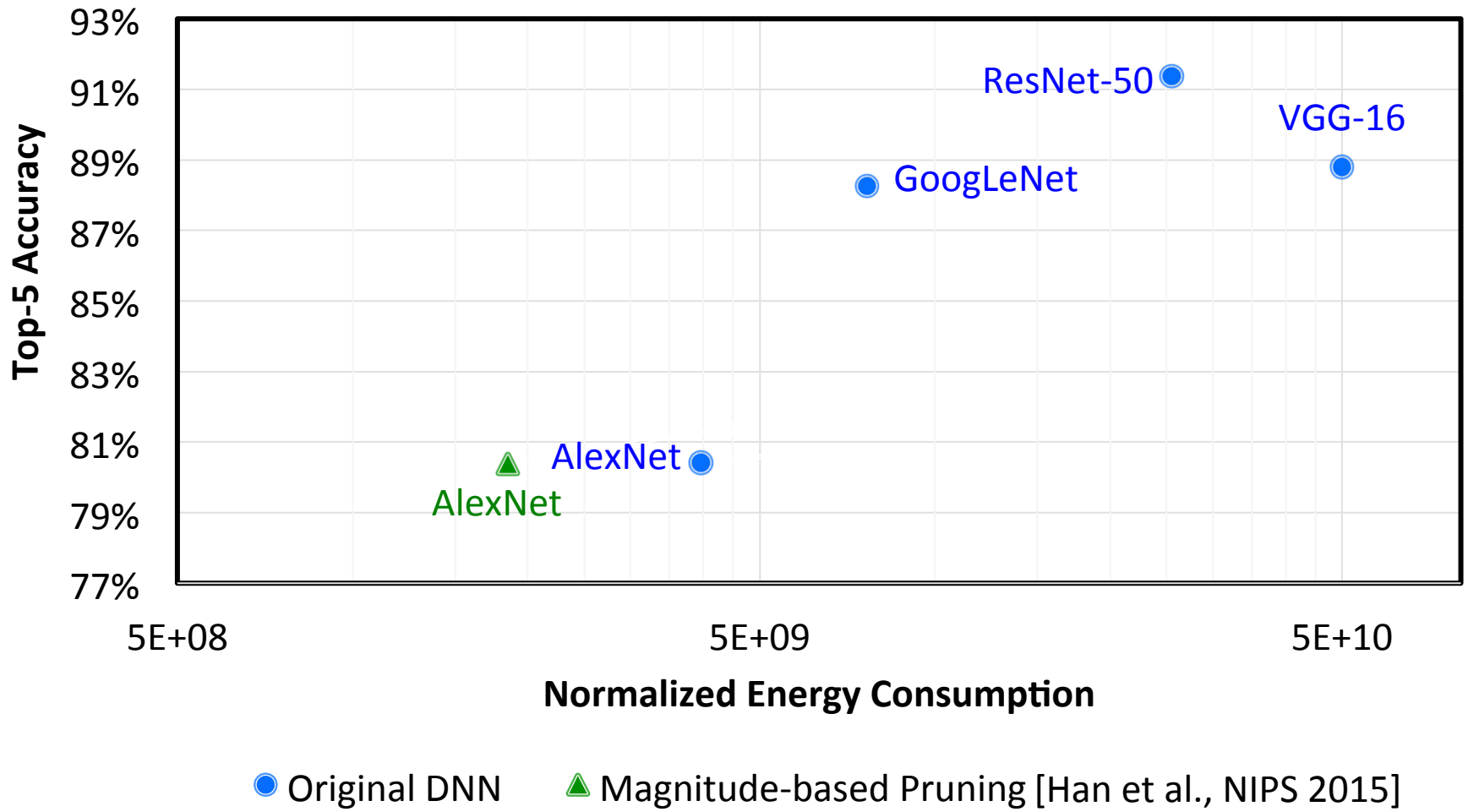
Energy Consumption of Existing DNNs



Deeper CNNs with fewer weights do not necessarily consume less energy than shallower CNNs with more weights

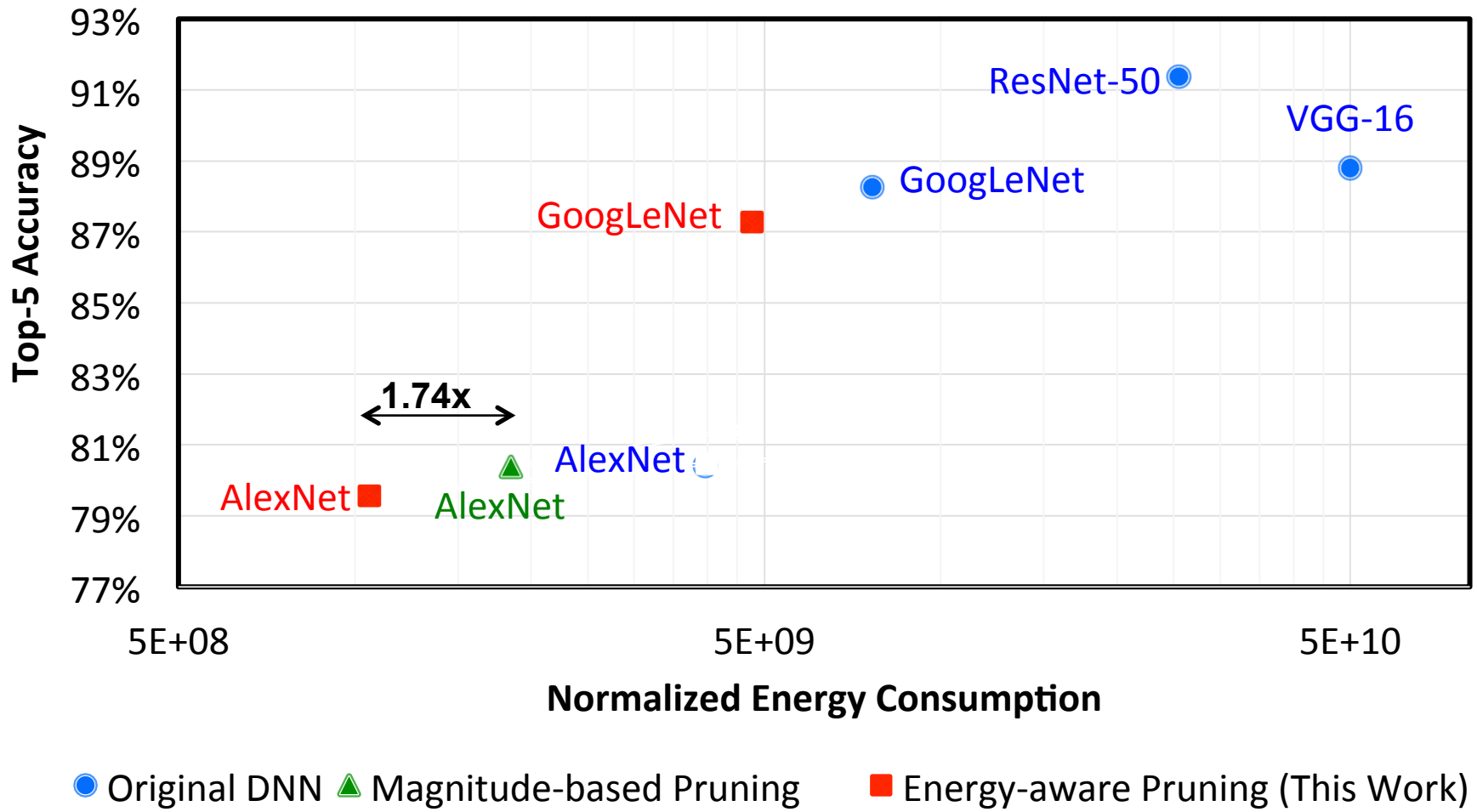
[Yang et al., CVPR 2017]

Magnitude-based Weight Pruning



Reduce number of weights by **removing small magnitude weights**

Energy-Aware Pruning

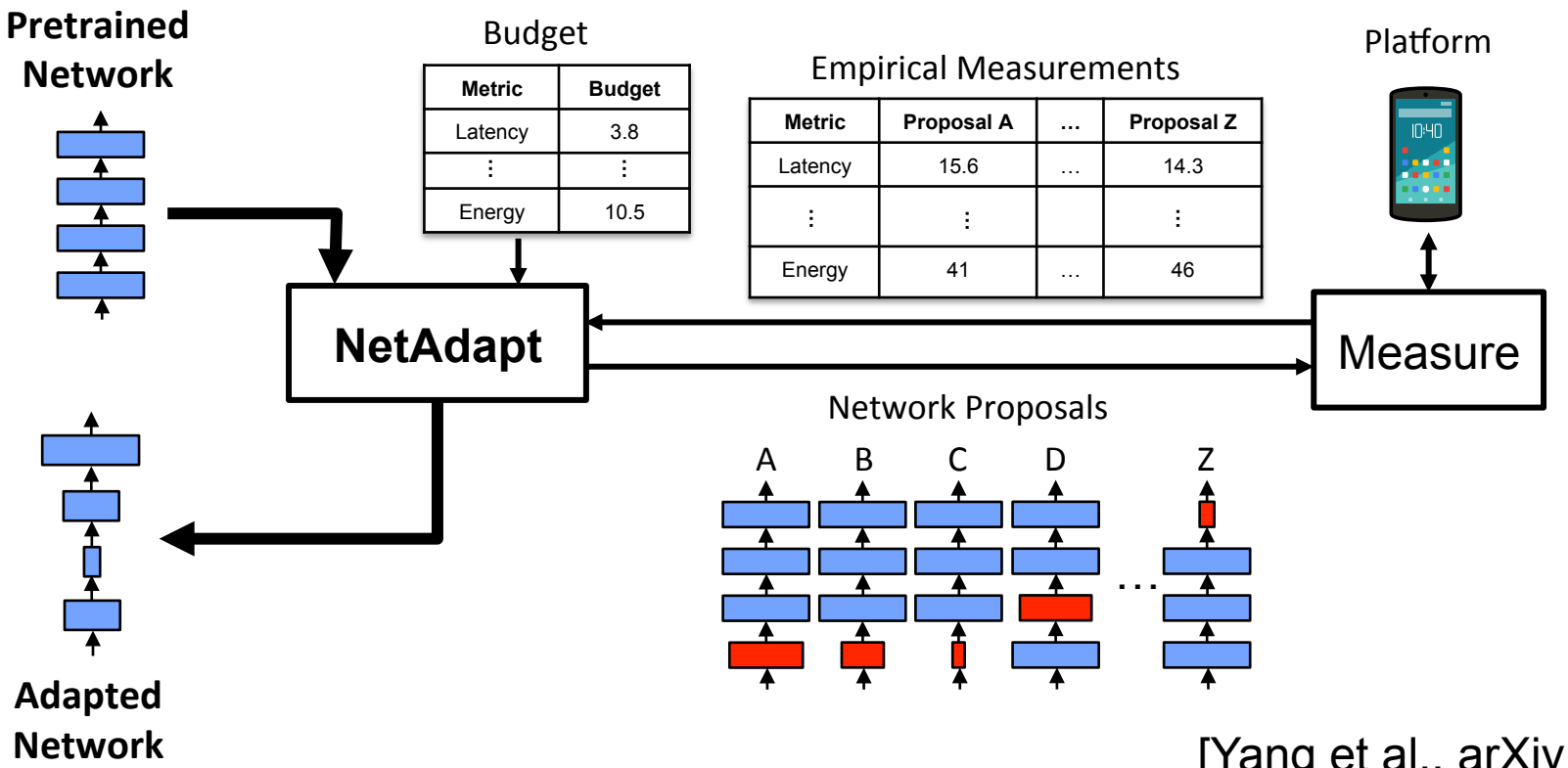


Directly target energy and incorporate it into the optimization of DNNs to provide greater energy savings

[Yang et al., CVPR 2017]

NetAdapt: Platform-Aware DNN Adaptation

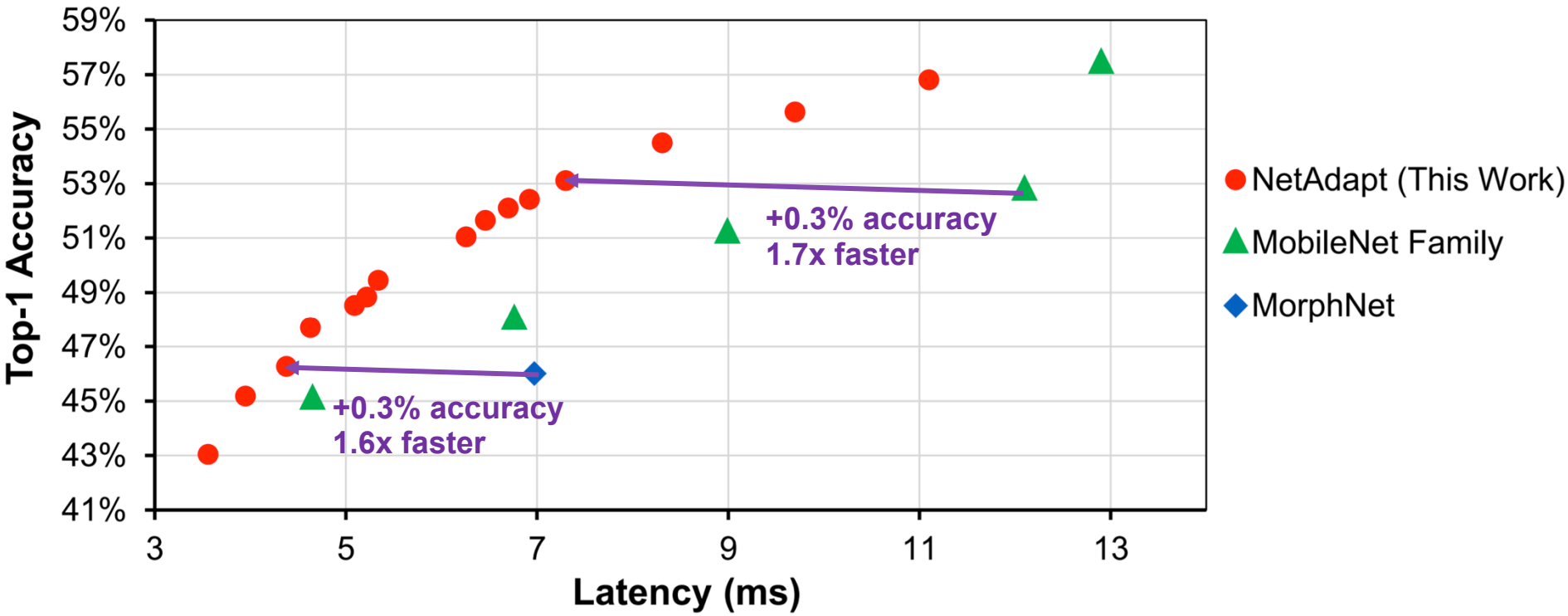
- **Automatically adapt DNN** to a mobile platform to reach a target latency or energy budget
- Use **empirical measurements** to guide optimization (avoid modeling of tool chain or platform architecture)



[Yang et al., arXiv 2018]

Latency vs. Accuracy Tradeoff with NetAdapt

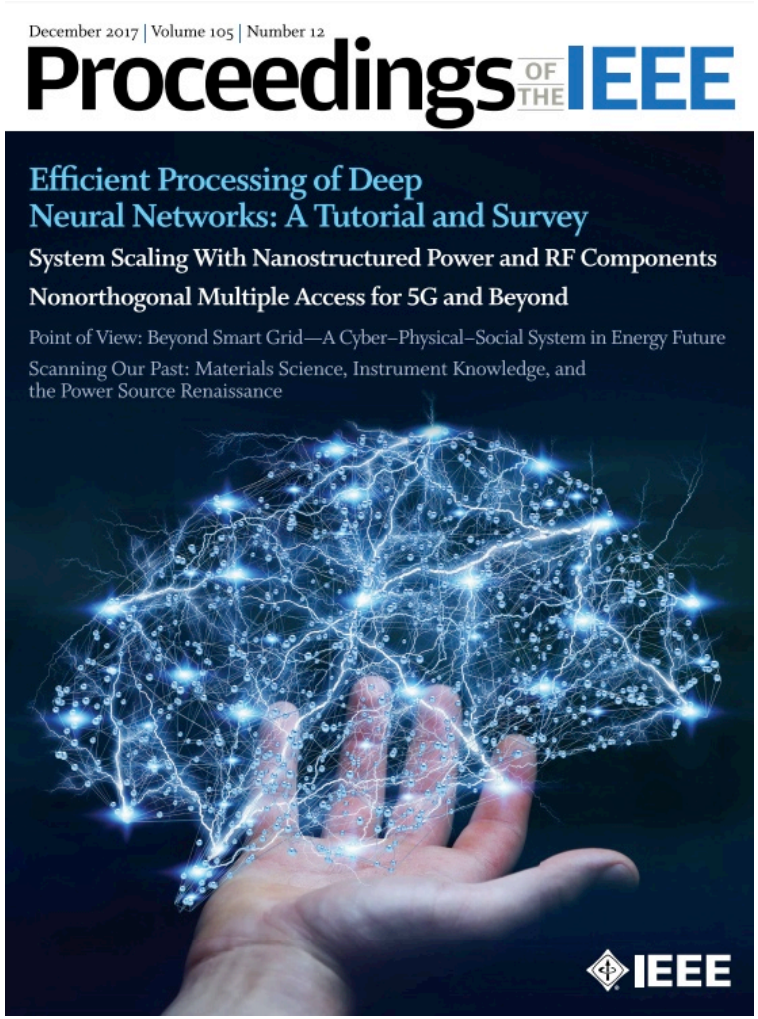
- **NetAdapt boosts <u>the real inference speed</u> of MobileNet by 1.7x with higher accuracy**

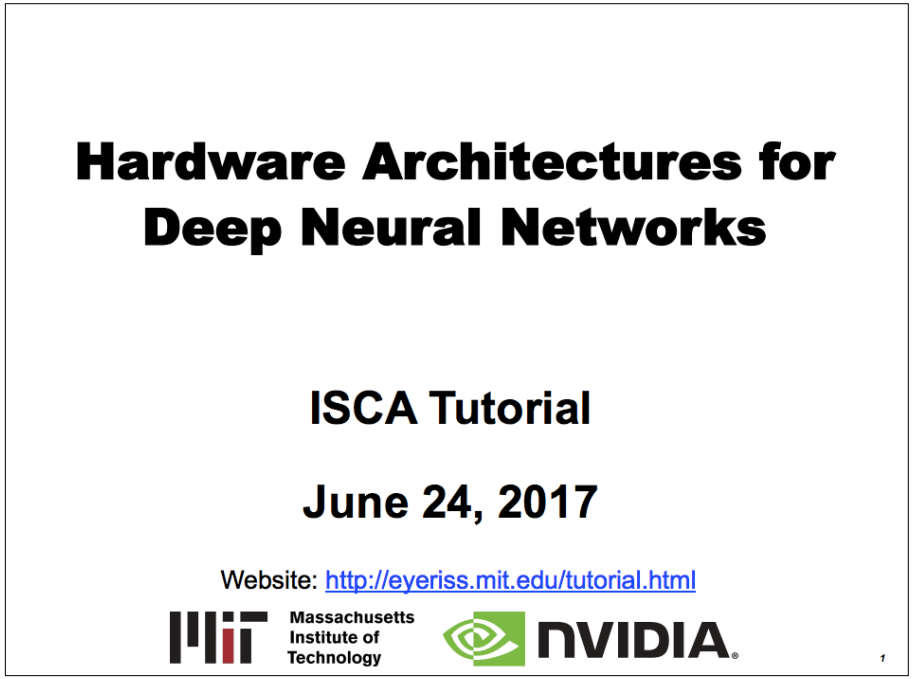


*Tested on the ImageNet dataset and a Google Pixel 1 CPU

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

Tutorial Material on Efficient DNNs



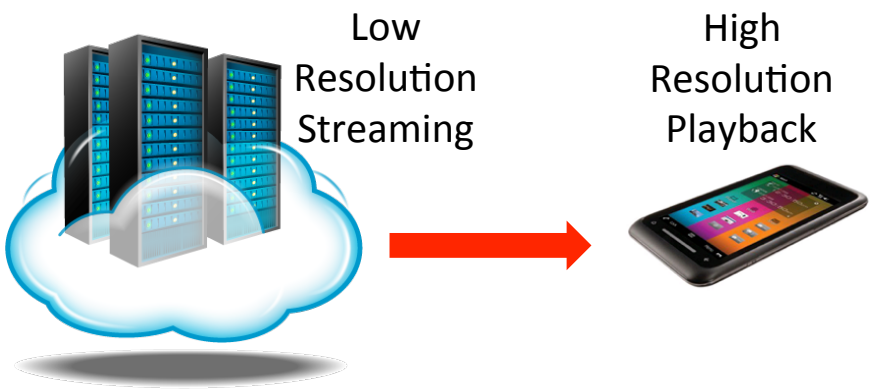


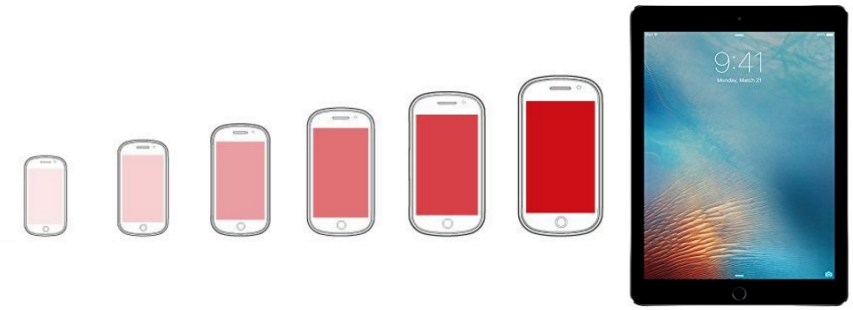
http://eyeriss.mit.edu/tutorial.html

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

Efficient Computer Vision using Compression

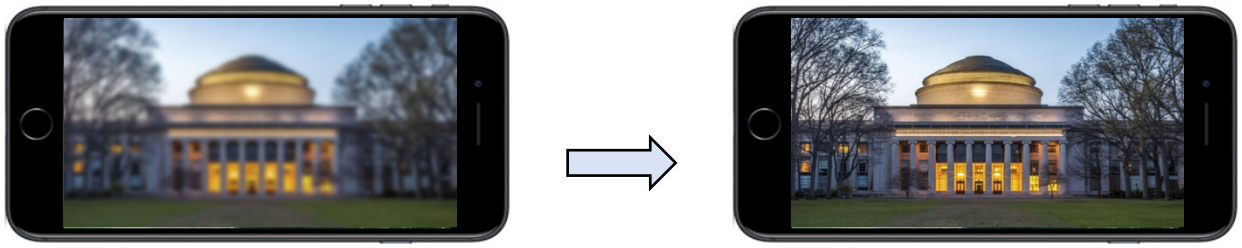
Super-Resolution on Mobile Devices





Transmit low resolution for lower bandwidth

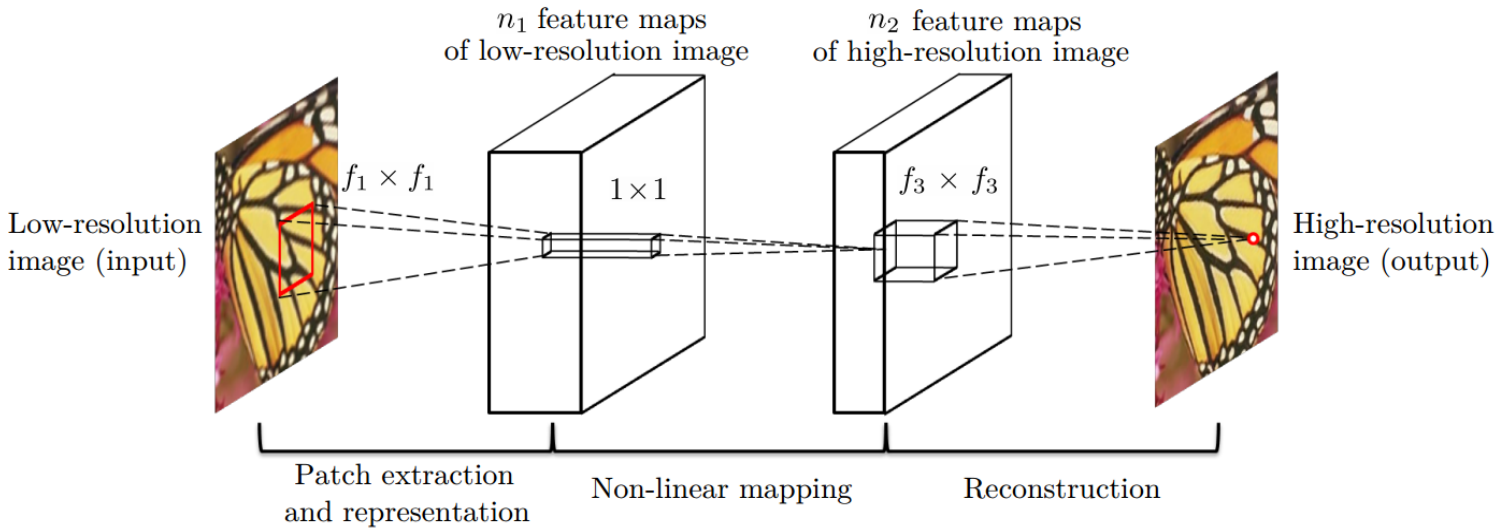
Screens are getting larger



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

Complexity of Super Resolution Algorithms

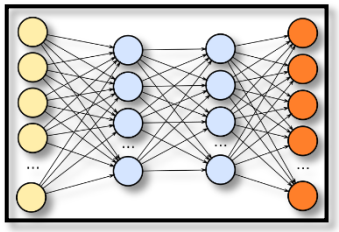
SRCNN (Dong et, al. ECCV 14)



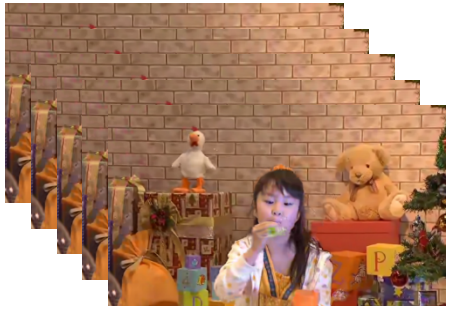
8032 MACs/pixel → ~500 GMAC/s for HD @ 30 fps

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

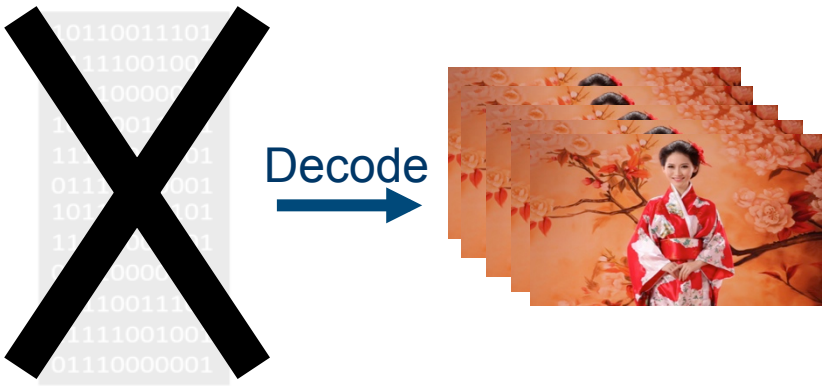
SR
15x faster

Real-time

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

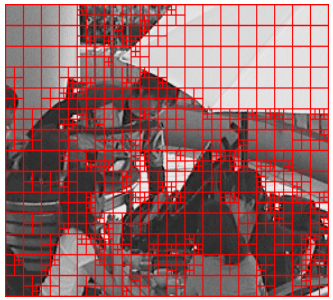
[Zhang et al., CVPRW 2017]

Free Information in Compressed Videos



Compressed video

Pixels



Block-structure

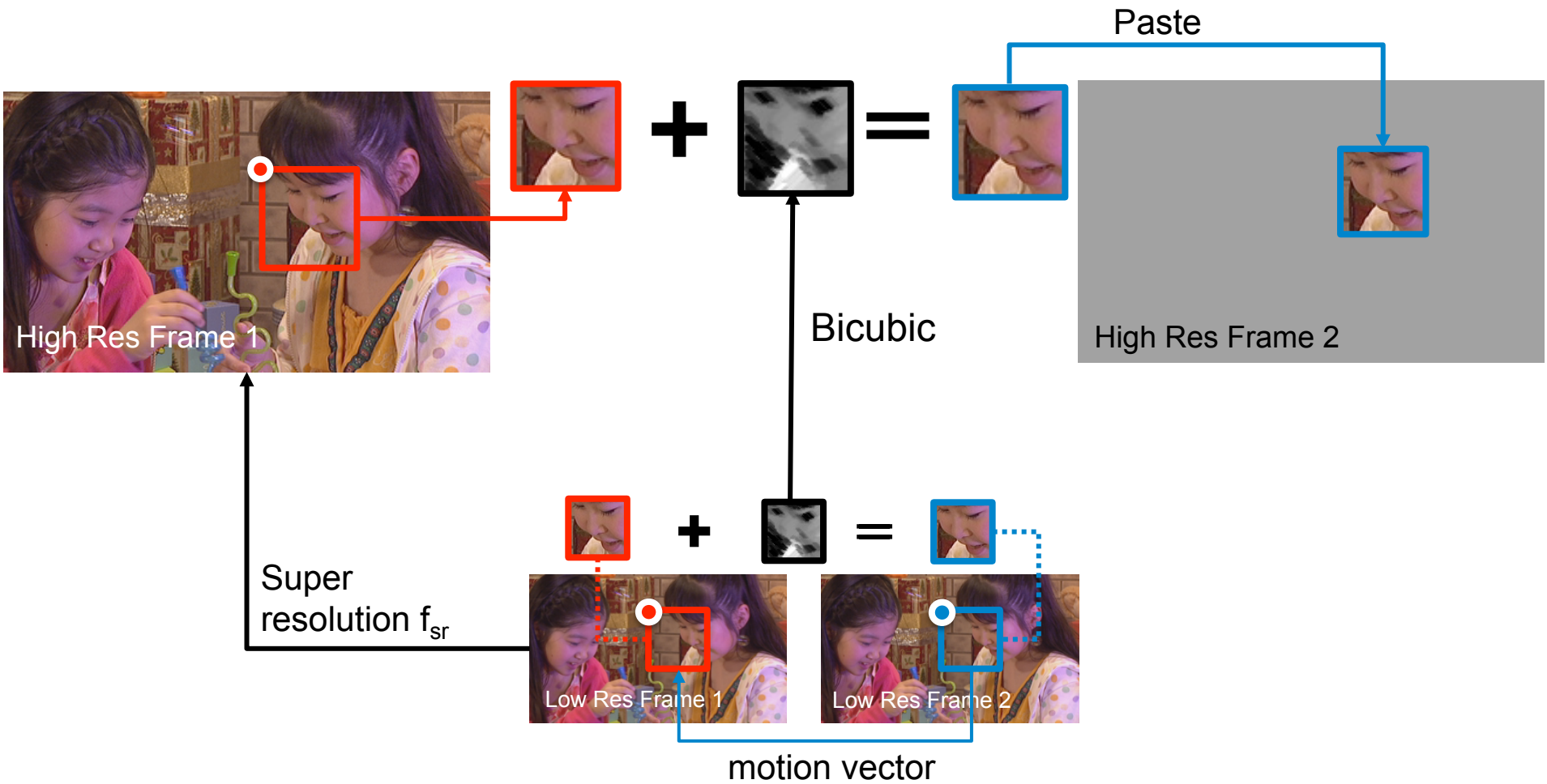
Motion-compensation

Video as a stack of pixels

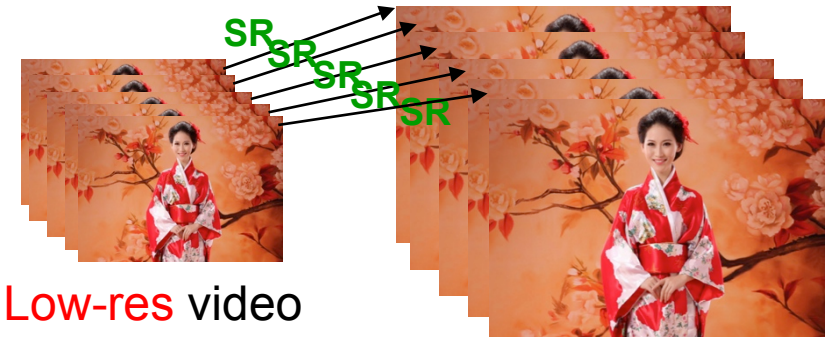
Representation in compressed video

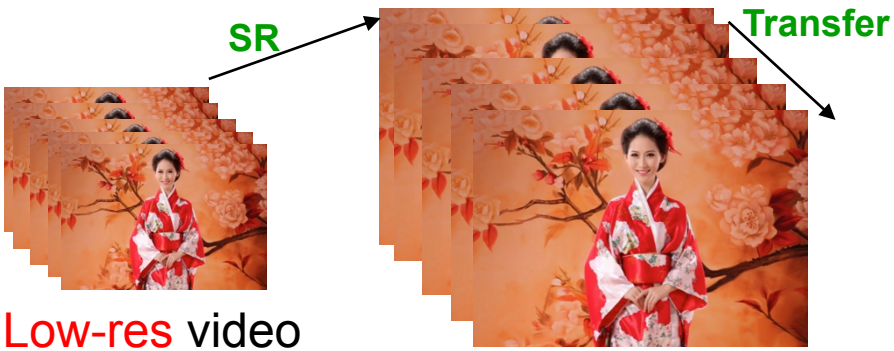
This representation can help **accelerate** super-resolution

Transfer the Super-Resolution Results



Transfer is Lightweight





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**

Challenge 1: Scene Transition

Transfer **will NOT work** if there is a **transition of scenes**

Group-of-Picture (GoP) Structure

GoP structure in the compressed video **provides video segmentation for free**

Challenge 2: Prediction Error

Ground-truth

Non-Adaptive

Artifacts when missing high frequency components of residual

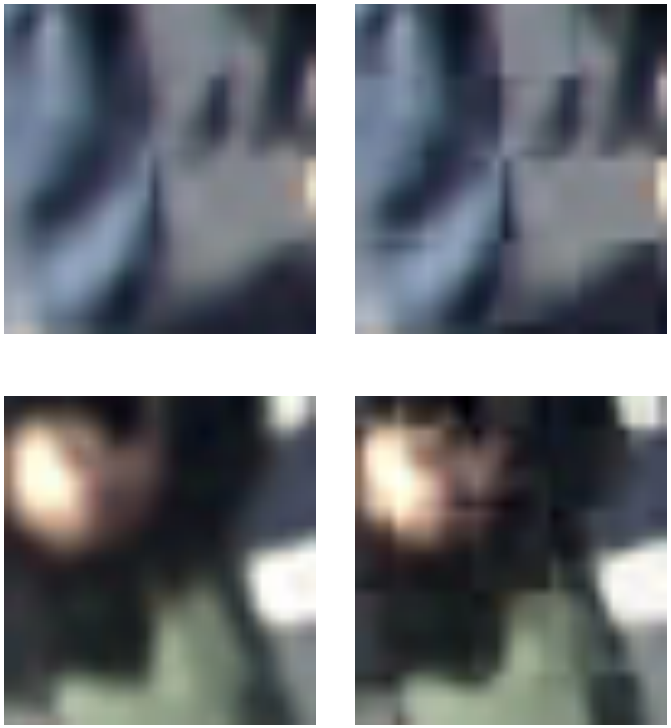
Adaptive

Use lightweight metric to identify occurrence and skip transfer

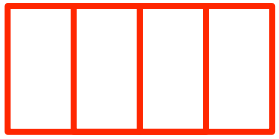
FAST **skips transfer** on blocks with **large residual**

Challenge 3: Blocking Artifacts

SRCNN | **No deblocking**

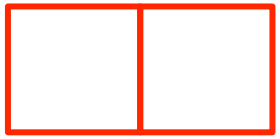


Overlapped Block Processing



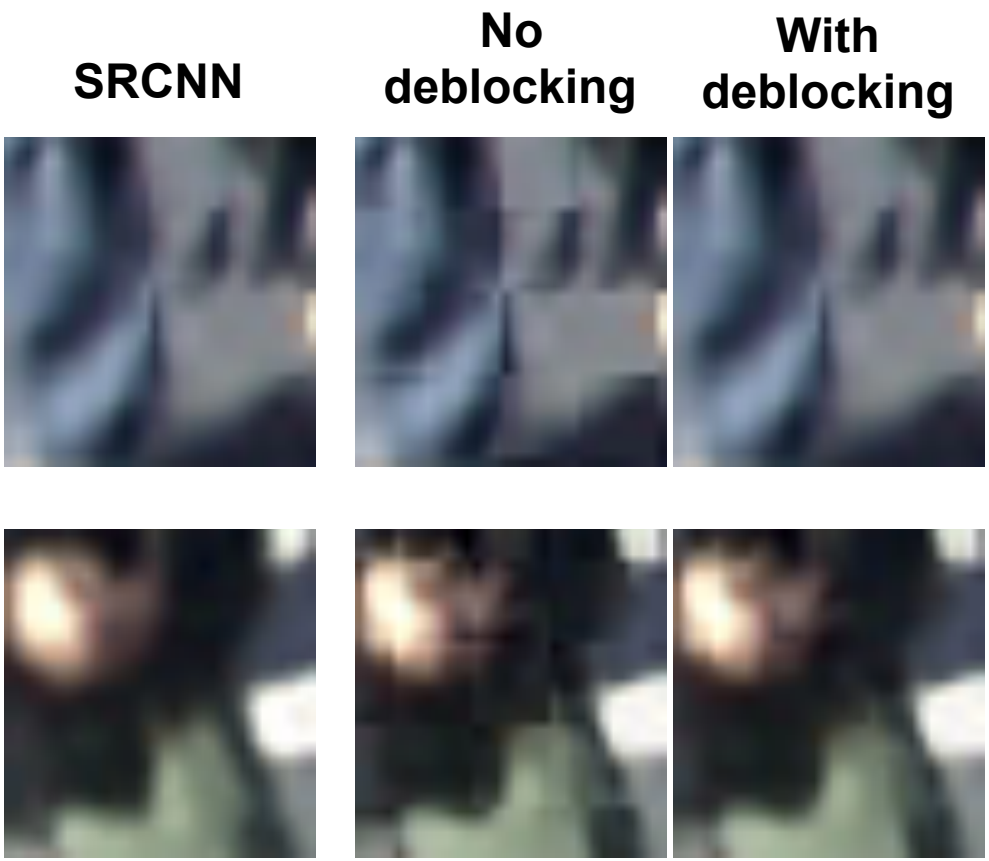
No blocking artifacts
Process each pixel multiple times

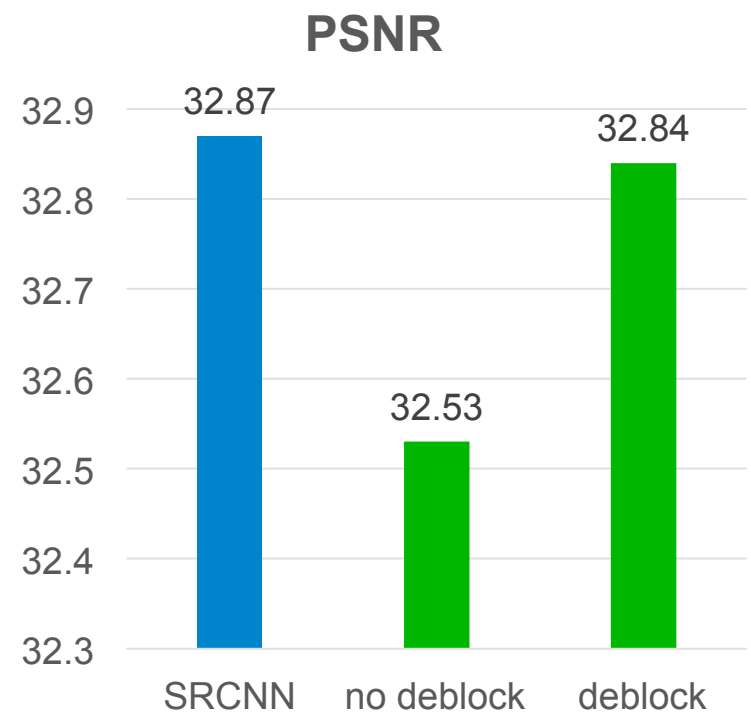
Non-Overlapped Block Processing



Process each pixel once
Blocking artifacts

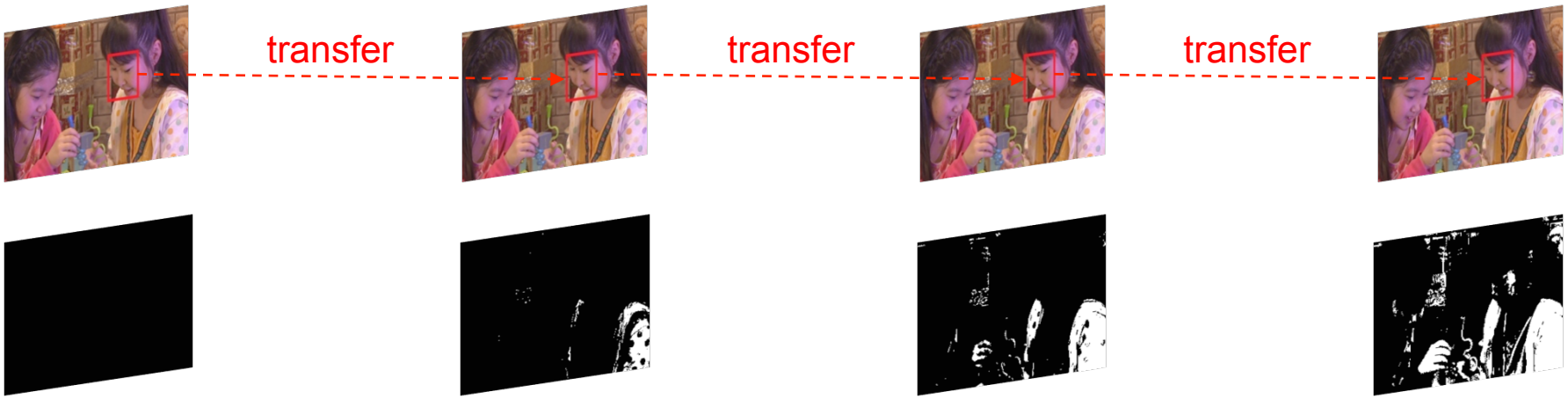
Challenge 3: Blocking Artifacts



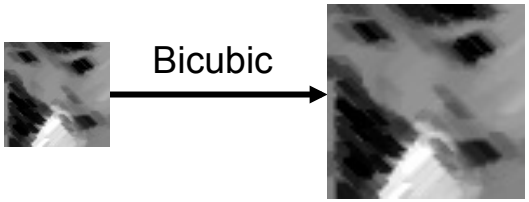


FAST applies the **deblocking filter** to alleviate the blocking effect caused by **non-overlapping block division**

Challenge 4: Accumulated Error



When a SR result gets transferred multiple times, the error **accumulates**



FAST estimates the accumulated error as the **accumulated Laplacian of the residual**, and stops the transfer when it exceeds a threshold

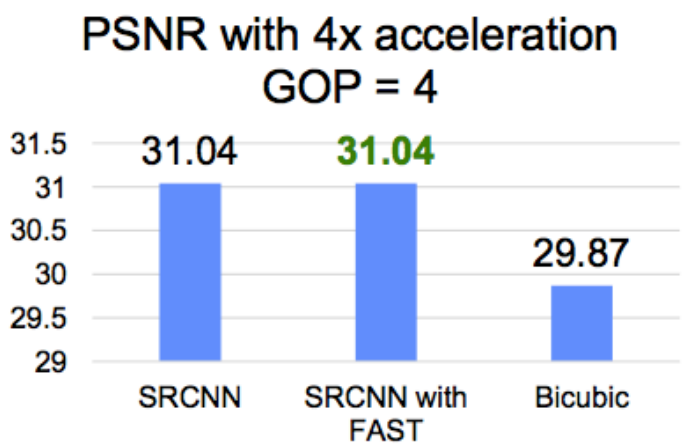
Evaluation: Accelerating SRCNN

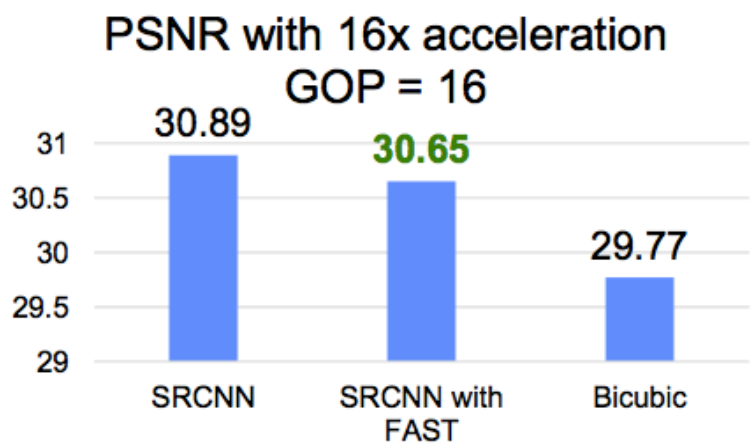
PartyScene

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

Bicubic

SRCNN

SRCNN with FAST

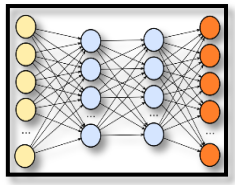
Ground-truth

Visual Evaluation

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

Summary of FAST

- **Transfer** the SR results guided by **motion vectors**
- **Adaptively** perform the transfer by thresholding on the **residue,** and **accumulated Laplacian**
- **Accelerates** SR algorithms by up to **15x** with **minimal PSNR loss**



SR algorithm

Compressed video

FAST

SR

15x faster

Real-time

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

[Zhang et al., CVPRW 2017]

Enable Real-time Navigation on nano drone

Image source: Cheerson

Big battery

Mobile GPU

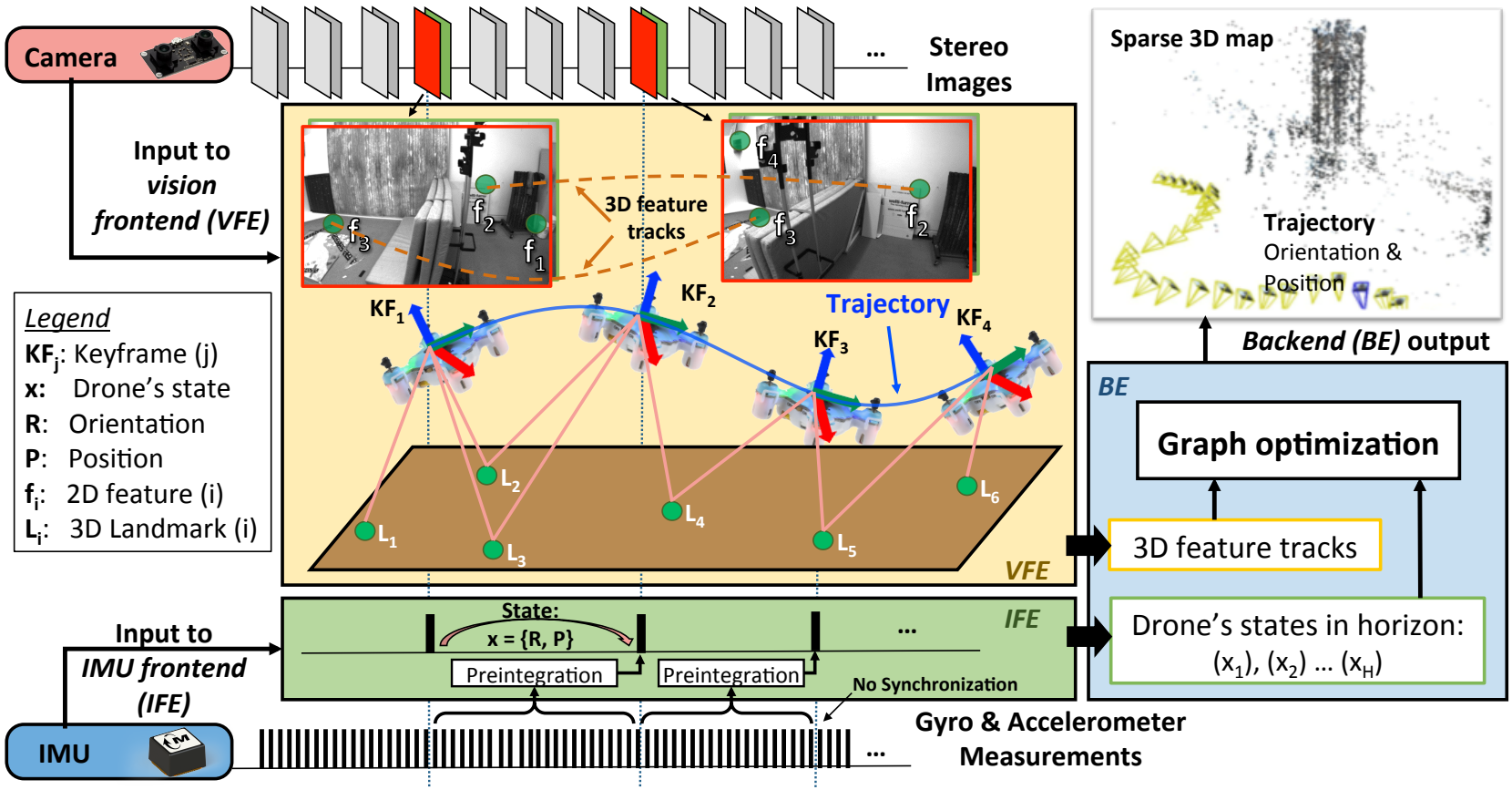
Enable energy-efficient navigation for **Search and Rescue**

[Zhang et al., RSS 2017]

http://navion.mit.edu

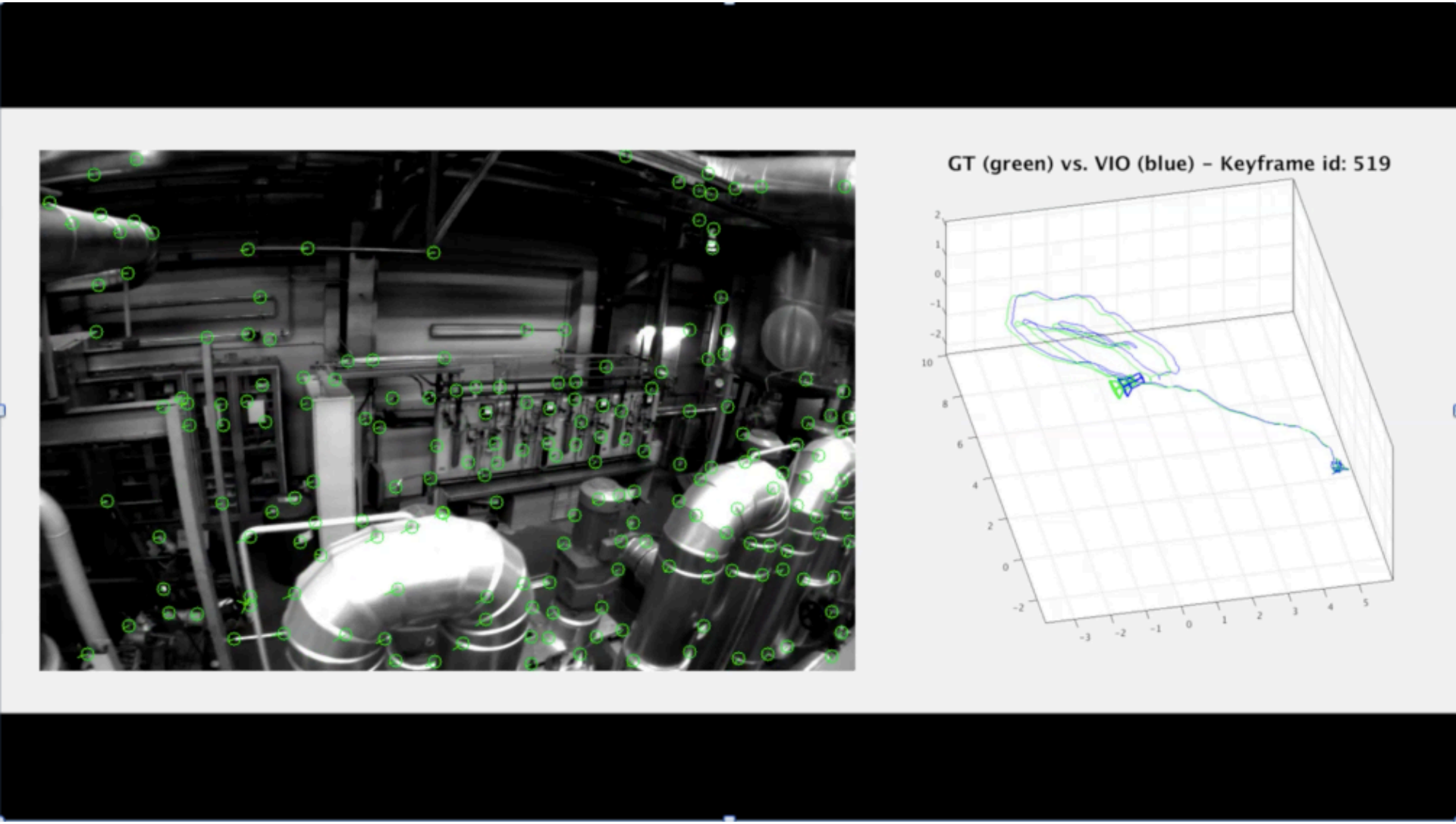
Localization with Visual Inertial Odometry

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



Navion: Fully integrated VIO system on-chip consuming **< 30mW**

Visual Inertial Odometry Demo

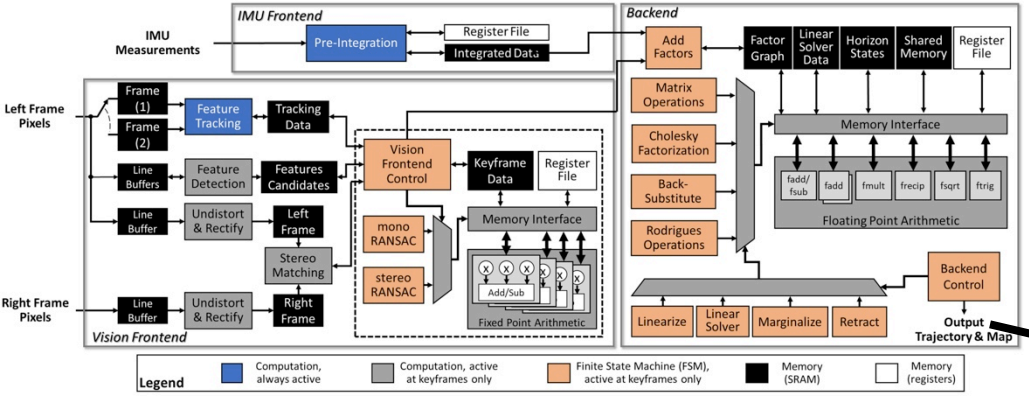


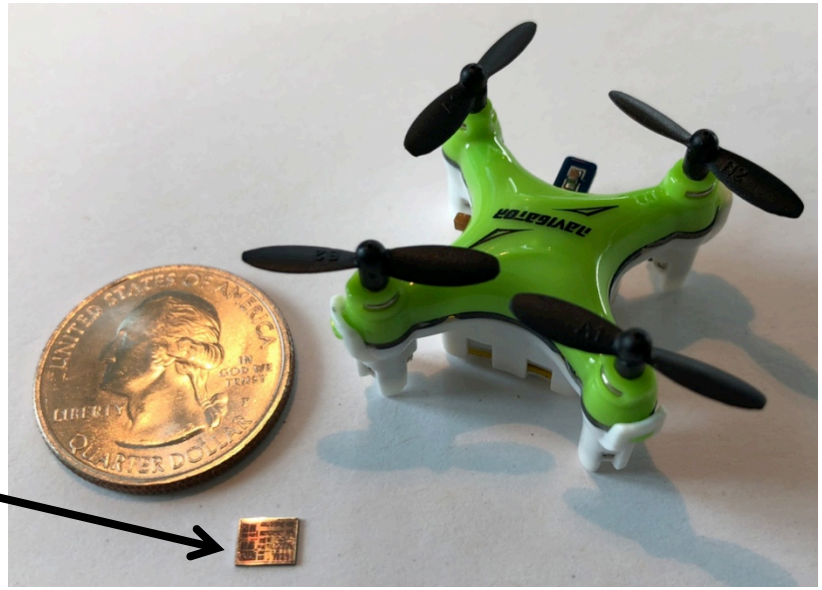
Compression to Reduce Energy and Cost

Memory dominates energy and chip area.

Navion Architecture

Processes high dimensional data





[Suleiman et al., VLSI 2018]

Apply various **compression techniques** to reduce on-chip storage cost by 4.1x. Entire VIO system is fully integrated on chip (20mm^2).

Summary

- Video is perhaps the biggest of the 'big data' being collected and transmitted.
- Moving from compressing to understanding pixels at the edge increasingly desirable due to privacy/security and latency constraints. However, energy significantly limited at edge.
- Co-design of algorithms and hardware can enable energy-efficient video coding, computer vision and deep learning such that they can efficiently operate on edge devices such as smartphones and autonomous vehicles/drones.
- Bridging the gap between video coding, computer vision and deep learning plays an important role in overcoming many of the challenges faced by next generation of edge devices.

Acknowledgements

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

- **Energy-Efficient Video Coding**
  - *M. Tikekar, C.-T. Huang, V. Sze, A. Chandrakasan, "Energy and Area-Efficient Hardware Implementation of HEVC Inverse Transform and Dequantization," IEEE International Conference on Image Processing (ICIP), pp. 2100-2104, October 2014.*
  - *Y.-H. Chen, V. Sze, "A Deeply Pipelined CABAC Decoder for HEVC Supporting Level 6.2 High-Tier Applications," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), Vol. 25, No. 5, pp. 856-868, May 2015.*
  - *M. Tikekar, V. Sze, A. Chandrakasan, "A Fully-Integrated Energy-Efficient H.265/HEVC Decoder with eDRAM for Wearable Devices," IEEE Symposium on VLSI Circuits (VLSI-Circuits), June 2017.*
- **Compression Inspired by Computer Vision**
  - *Z. Zhang, V. Sze, "Rotate Intra Block Copy for Still Image Coding," IEEE International Conference on Image Processing (ICIP), September 2015.*

References

- **Energy-Efficient Deep Learning**

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

References

- **Efficient Computer Vision using Compression**
  - *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.*
  - *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.*