Design Space Exploration of FPGA-Based Deep Convolutional Neural Networks

Philipp Gysel
ECE Department
University of California, Davis

LEPS

Laboratory for Embedded and Programmable Systems

UC DAVIS
UNIVERSITY OF CALIFORNIA

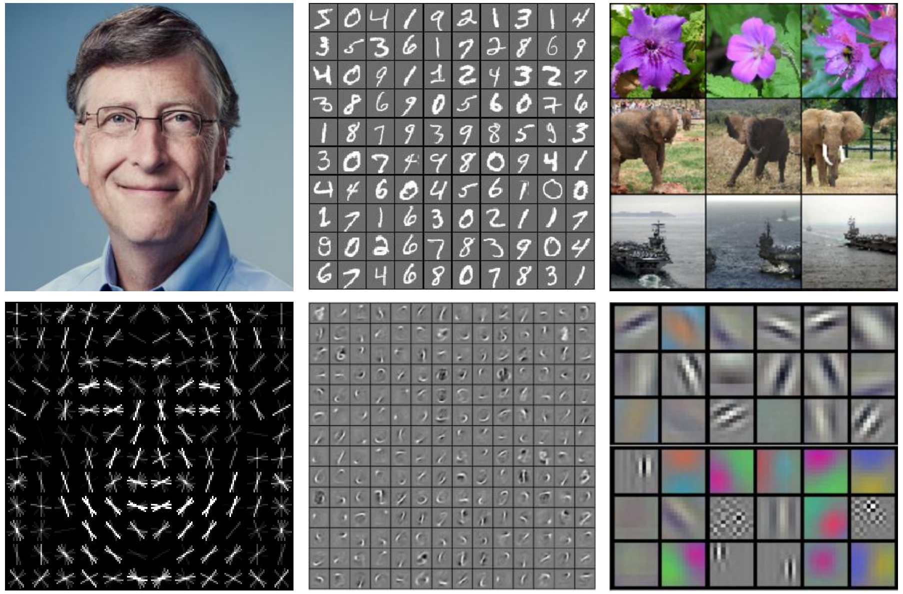
Machine Vision: Past, Present and Future!

Feature Extraction Approaches

- Hand crafted features such as HoG and SIFT
- Automated features extraction using Convolutional Neural Networks

CNN Based Feature Extraction

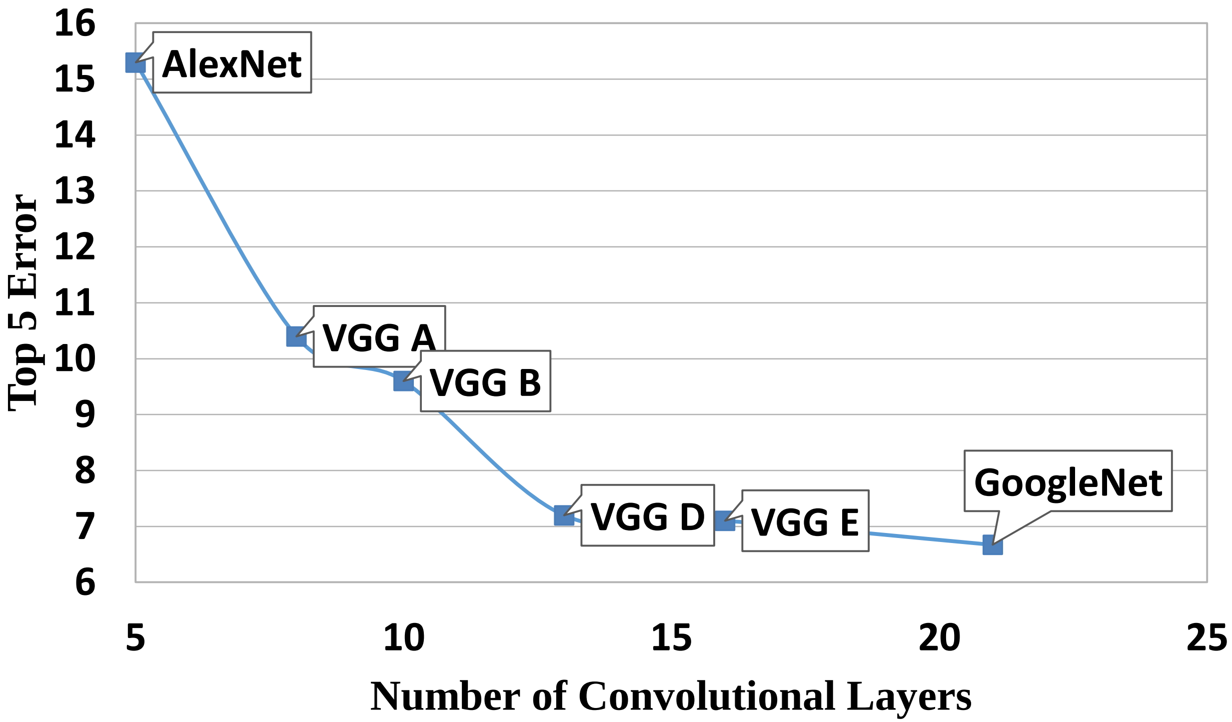
- Very effective in different vision tasks
- Very high computational complexity



dlib.net [Krizhevsky et al. 2012]

Precision – Depth Tradeoff

Dataset: ImageNet 2012 – Top 5 Error [Simonyan 2015]



Higher Depth

Higher Precision

Higher Execution Time

Higher Power Consumption

Mobile devices have to offload the computation to a cloud.

Implementation Choices

GPU

- High power consumption
- Inflexibility (Single and Floating point only)

ASIC

- Initial Cost
- Reusability
- Complexity of design

FPGA

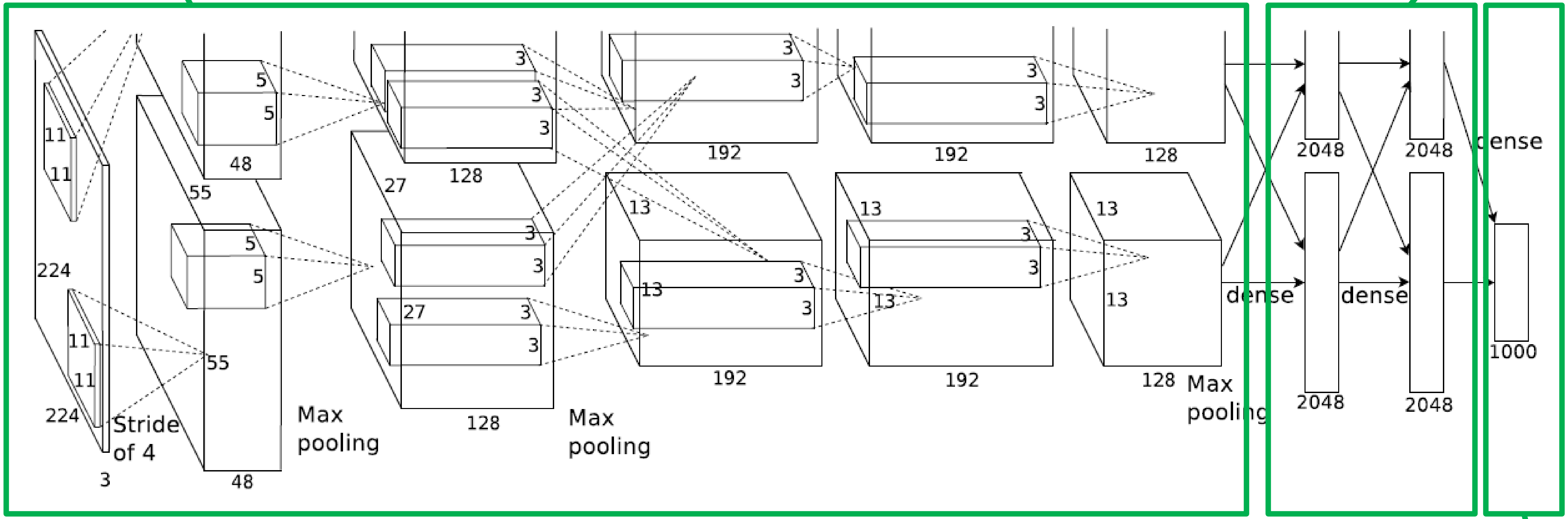
- Complexity of design

An energy efficient and fast implementation of DCNNs is very beneficial for mobile devices. This can be achieved by hardware based acceleration of DCNNs.

Convolutional Layers.
Over 90% of
computation time.

AlexNet

Fully Connected Layers.
They can extract local
and global features.



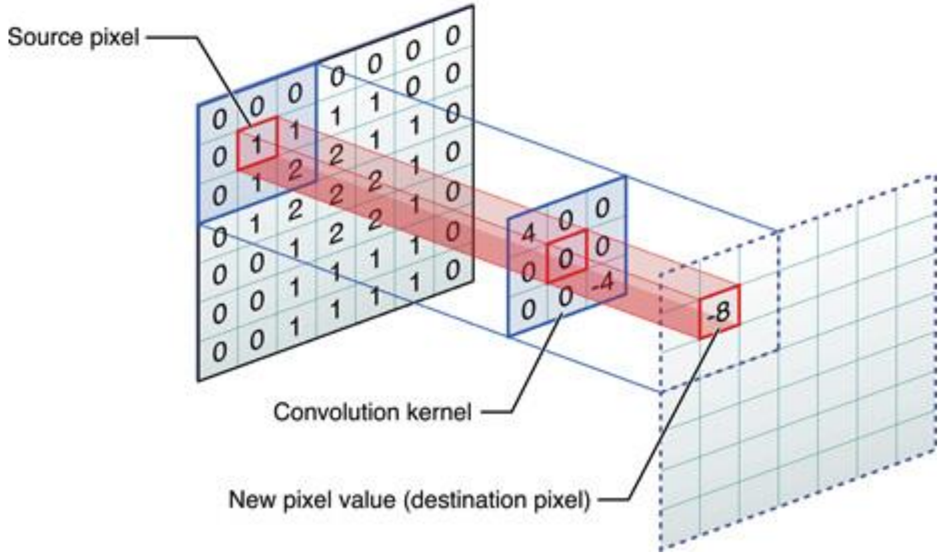
Classifier

[Krizhevsky 2012]

2D Convolution

- Center element of kernel is placed on each pixel of Input Feature Map (IFM)
- Convolution Result:

$$(4 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 1) + (-2 \times 4) = -8$$

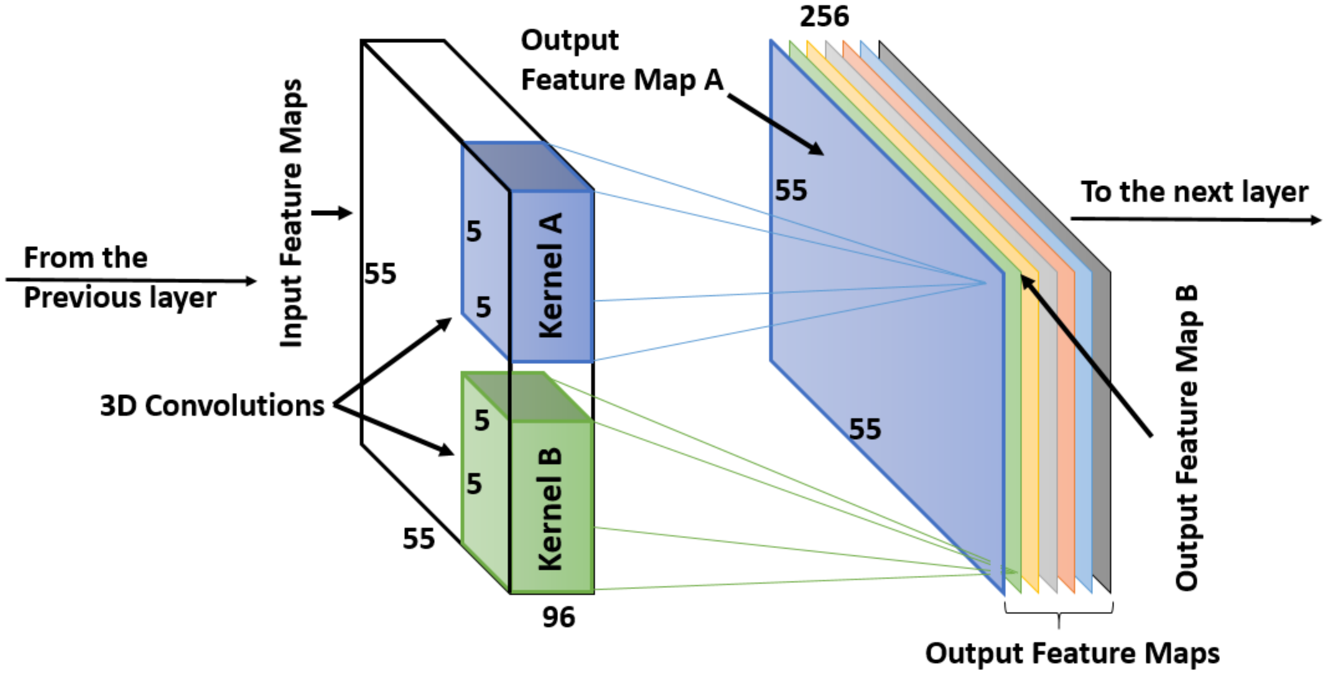


developer.apple.com

3D Convolution

- Each Output Feature Map (OFM) is the result of a 3D convolution of the Input Feature Map (IFM) with a Kernel stack.
- Example

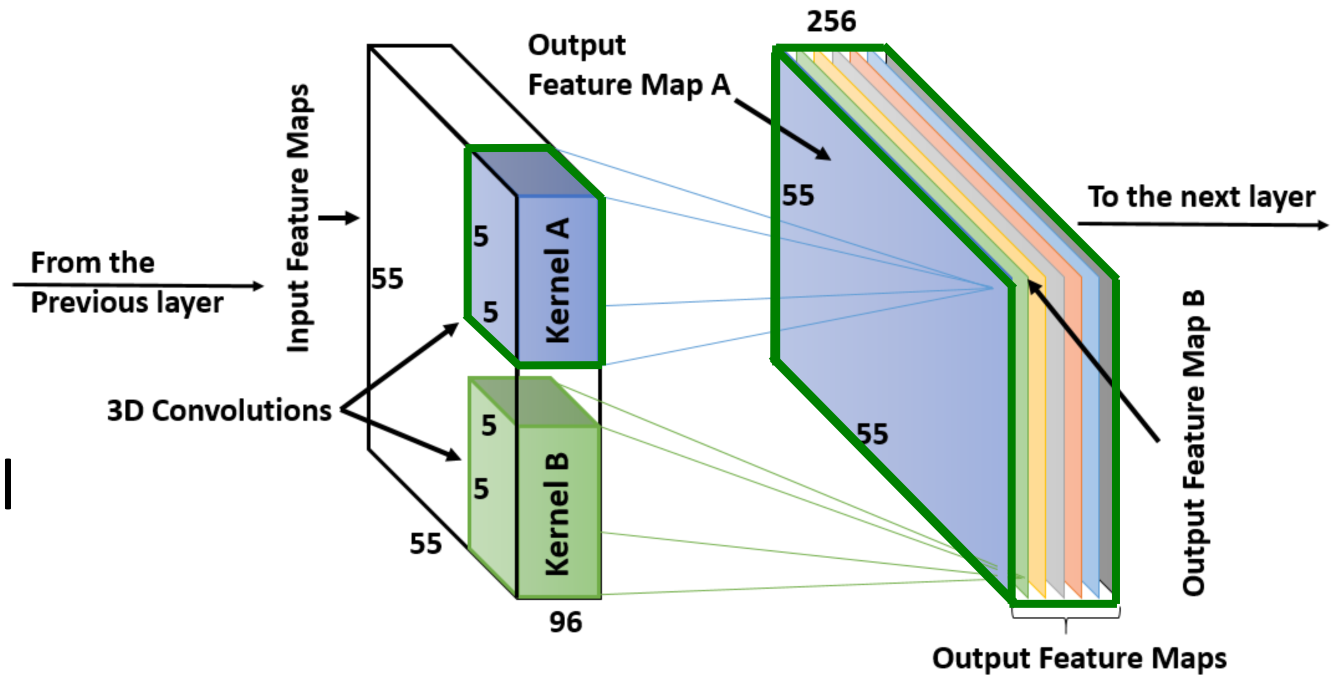
$$OFM_A = 3DConv(IFM, Kernel_A)$$



$$\forall\, i \in \{0, 1, \dots, 255\}: OFM_i = 3DConv\ (IFM, Kernel_i)$$

Parallelism Sources

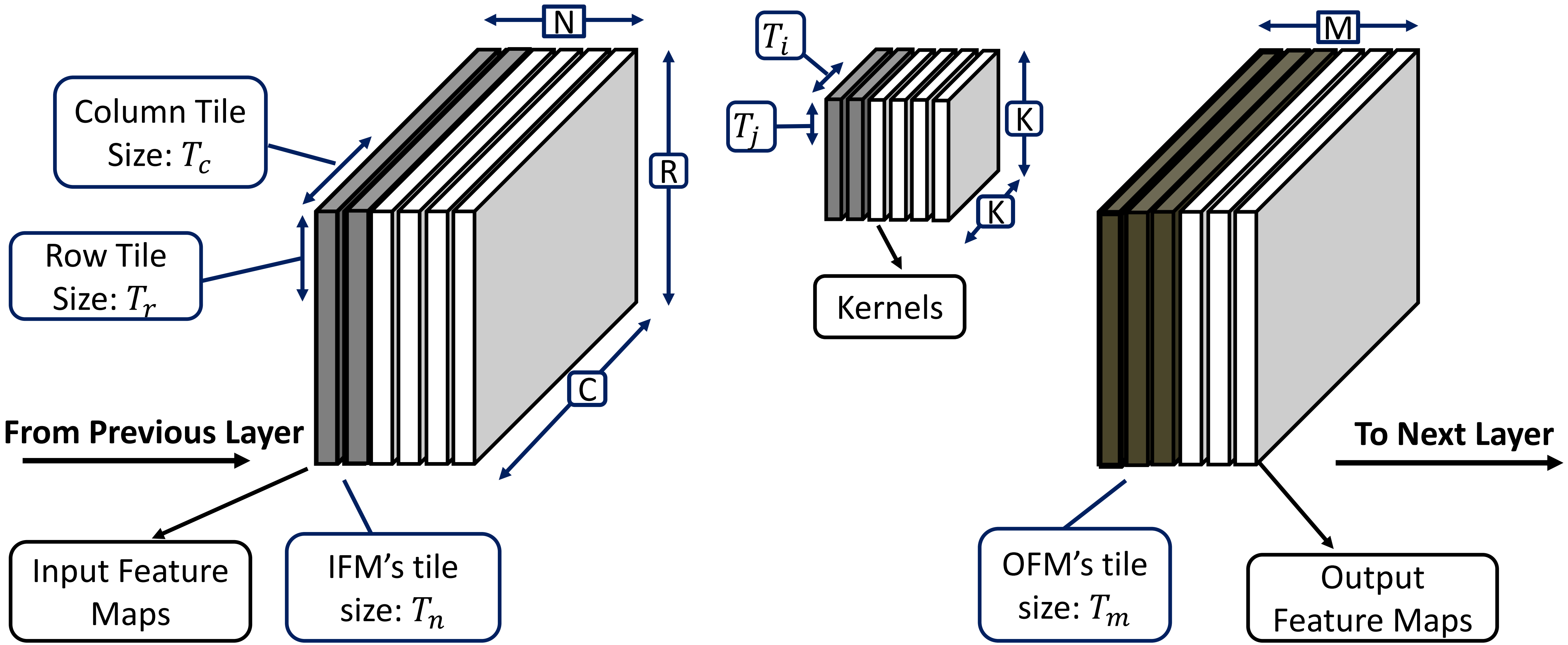
- Inter Layer Parallelism
- Inter Output Parallelism
  - Compute **different OFMs** in parallel
- Inter Kernel Parallelism
  - Compute **one OFM** in parallel
- Intra Kernel Parallelism
  - Compute **one convolution** in parallel



Design Philosophy

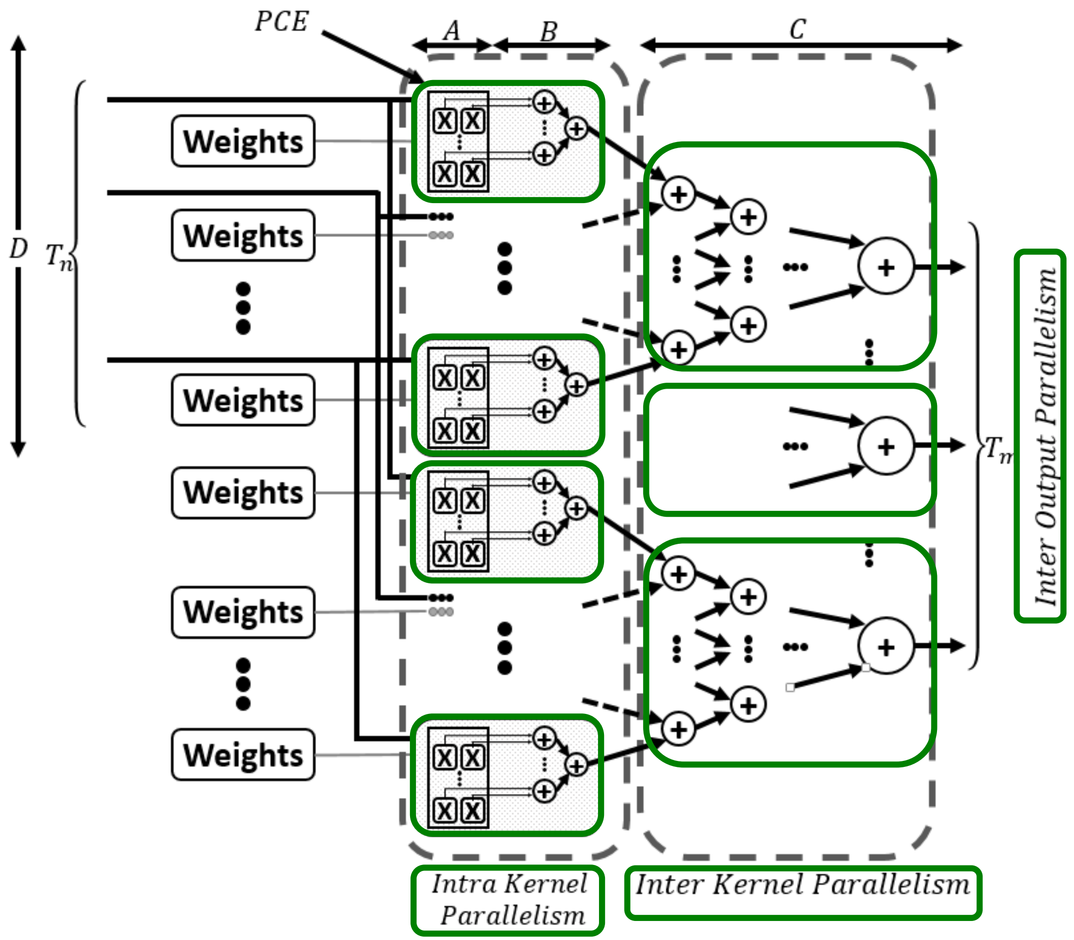
- DCNNs
  - Computation bound
  - Communication bound
- Computation – Communication balance
  - Memory model
  - Computation model

Tilling in Convolutional Layers



The Architecture Template

- Intra Kernel Parallelism
  - PCE: Parallel Convolution Engine
  - Here the number of parallel multiplications (T_k) is 4.
- Inter Kernel Parallelism
  - Convolve different IFMs
- Inter Output Parallelism
  - PCEs with different weights



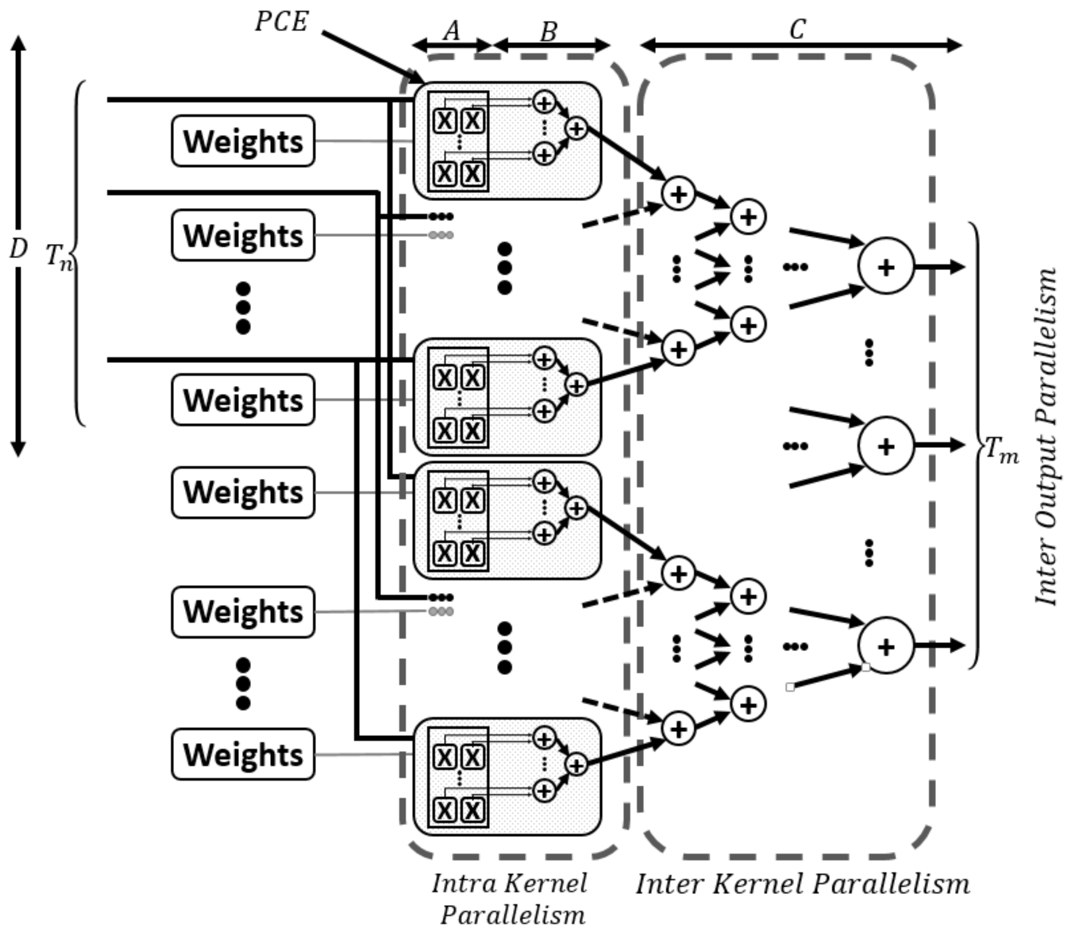
Computation Model

- Number of cycles in tiled model:

$$Cycles = \#Rounds \times \#Operations\ per\ round$$

- $\#Rounds = \left\lceil \frac{M}{T_m} \right\rceil \times \left\lceil \frac{\mathrm{N}}{T_n} \right\rceil \times \frac{RC}{T_r T_c} \times \left\lceil \frac{K}{T_i} \right\rceil \times \left\lceil \frac{K}{T_i} \right\rceil$

- $\#Ops\ per\ round = (T_r T_c \times \left\lceil \frac{T_i T_j}{T_k} \right\rceil + P)$



Memory Model

- Computation to communication ratio:

$$CTC = \frac{Total\ Computation}{Total\ Communication}$$

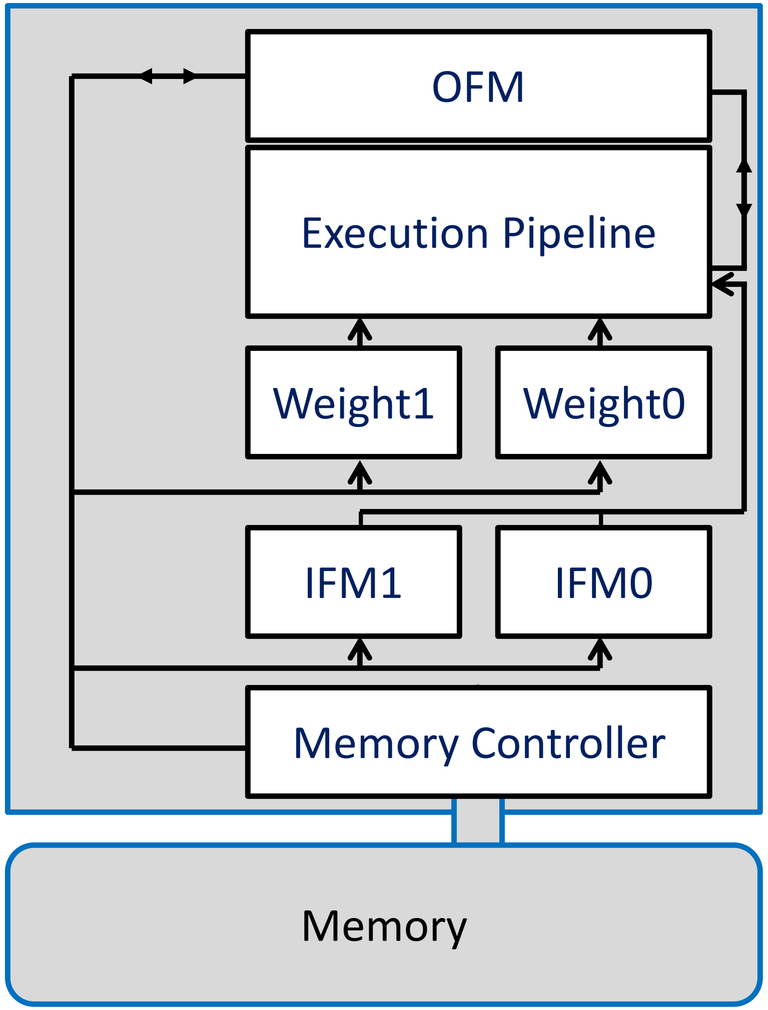
$$= \frac{2 \times M \times N \times R \times C \times K \times K}{\alpha_{in} \times \beta_{in} + \alpha_{out} \times \beta_{out} + \alpha_{wght} \times \beta_{wght}}$$

- Weight's buffer size

$$\beta_{wght} = T_m \times T_n \times T_i \times T_j$$

- Number of loads and stores of weights

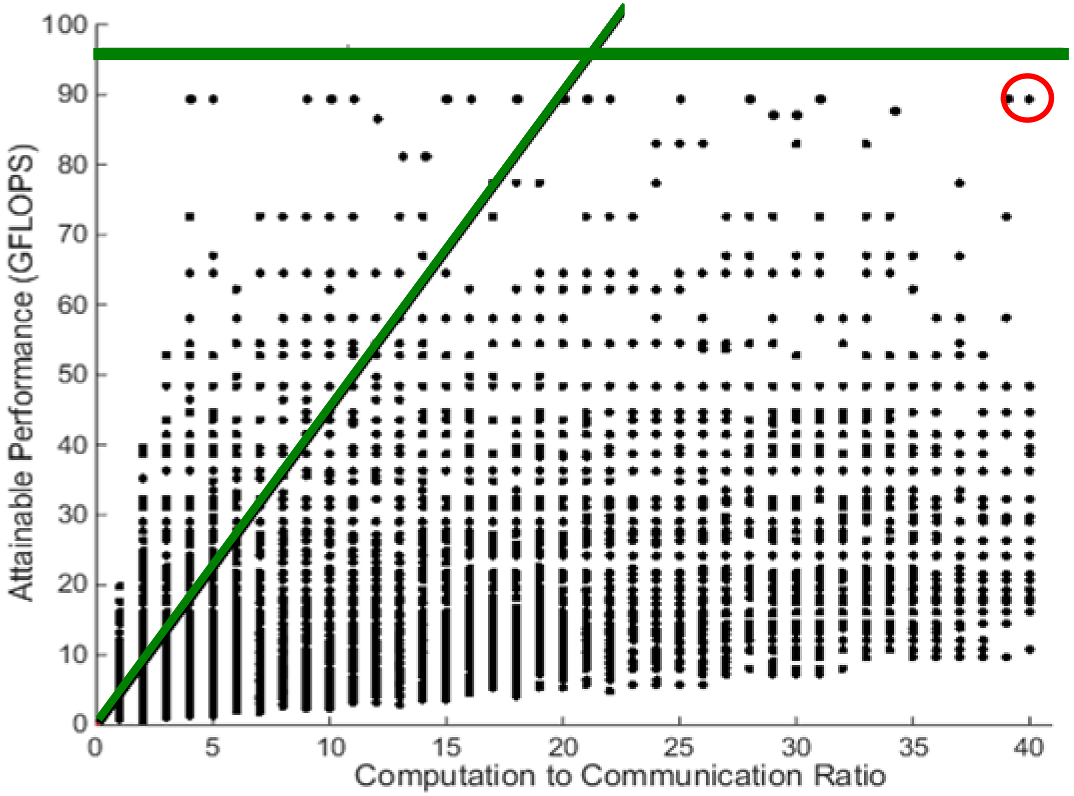
$$\alpha_{wght} = \frac{M}{T_m} \times \frac{N}{T_n} \times \frac{R}{T_r} \times \frac{C}{T_c} \times \frac{K}{T_i} \times \frac{K}{T_j}$$



Design Space Exploration

- Goal
  - Maximize throughput and CTC
- Constraints
  - Memory bandwidth
  - On-chip memory
  - Area limit (computation)
- Approach
  - Explore the design space for different values of T_m, T_n, T_r, T_c, T_i and T_j.

- AlexNet CONV1:

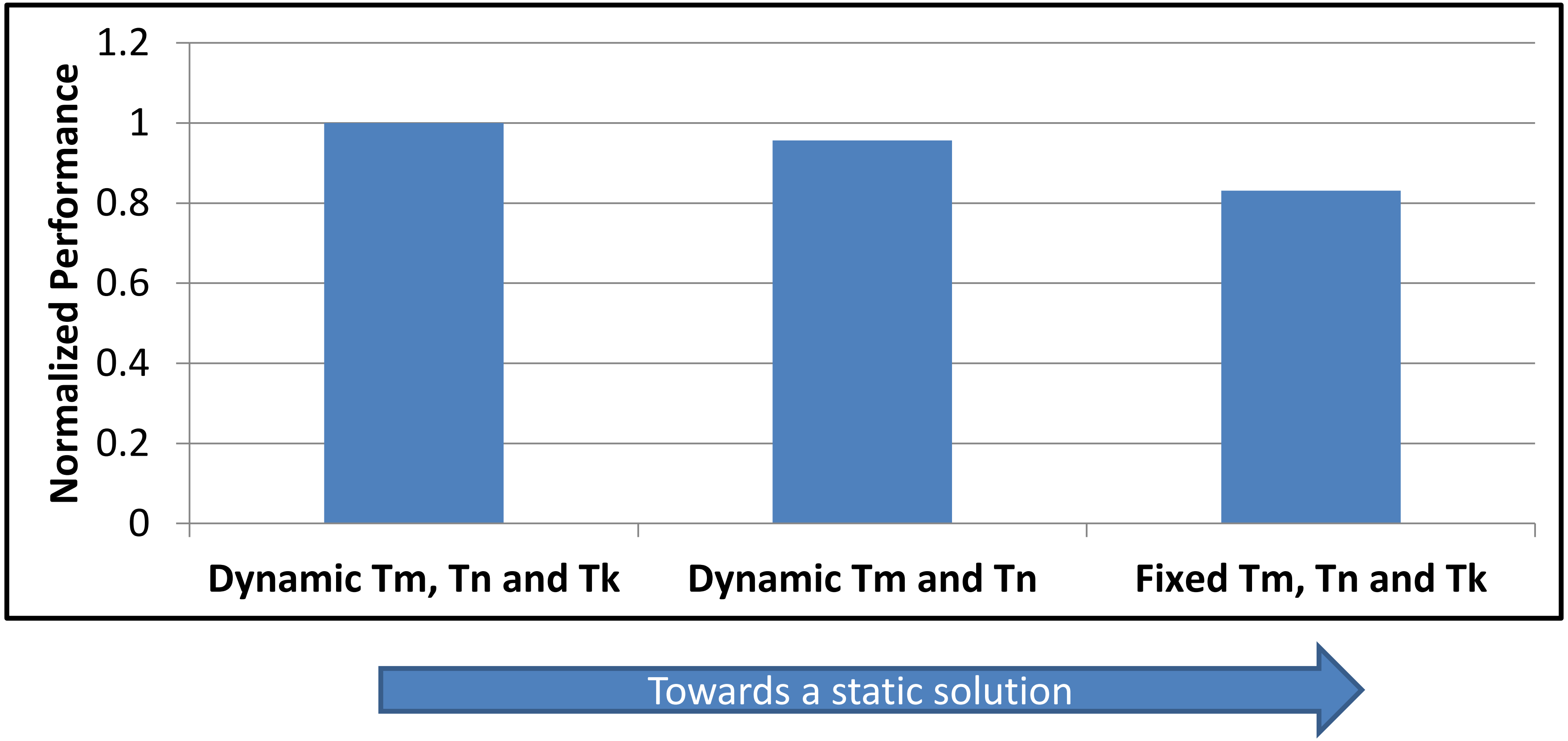


Re-configurability Effects (1)

Layer	Dynamic T_m, T_n and T_k					Dynamic T_m and T_n					Fixed T_m, T_n and T_k				
	T_m	T_n	T_k	Cycles	GFLOPS	T_m	T_n	T_k	Cycles	GFLOPS	T_m	T_n	T_k	Cycles	GFLOPS
1	16	3	10	117975	86	48	3	3	124025	85	16	3	9	127050	83
2	4	24	5	233280	96	10	16	3	255879	87	16	3	9	279936	80
3	15	32	1	79092	95	16	10	3	79092	95	16	3	9	87204	86
4	15	32	1	118638	95	32	5	3	118638	95	16	3	9	129792	86
5	10	48	1	79092	95	10	16	3	79092	95	16	3	9	86528	86
Sum				628077					656726					755642	

Towards a static solution

Re-configurability Effects (2)

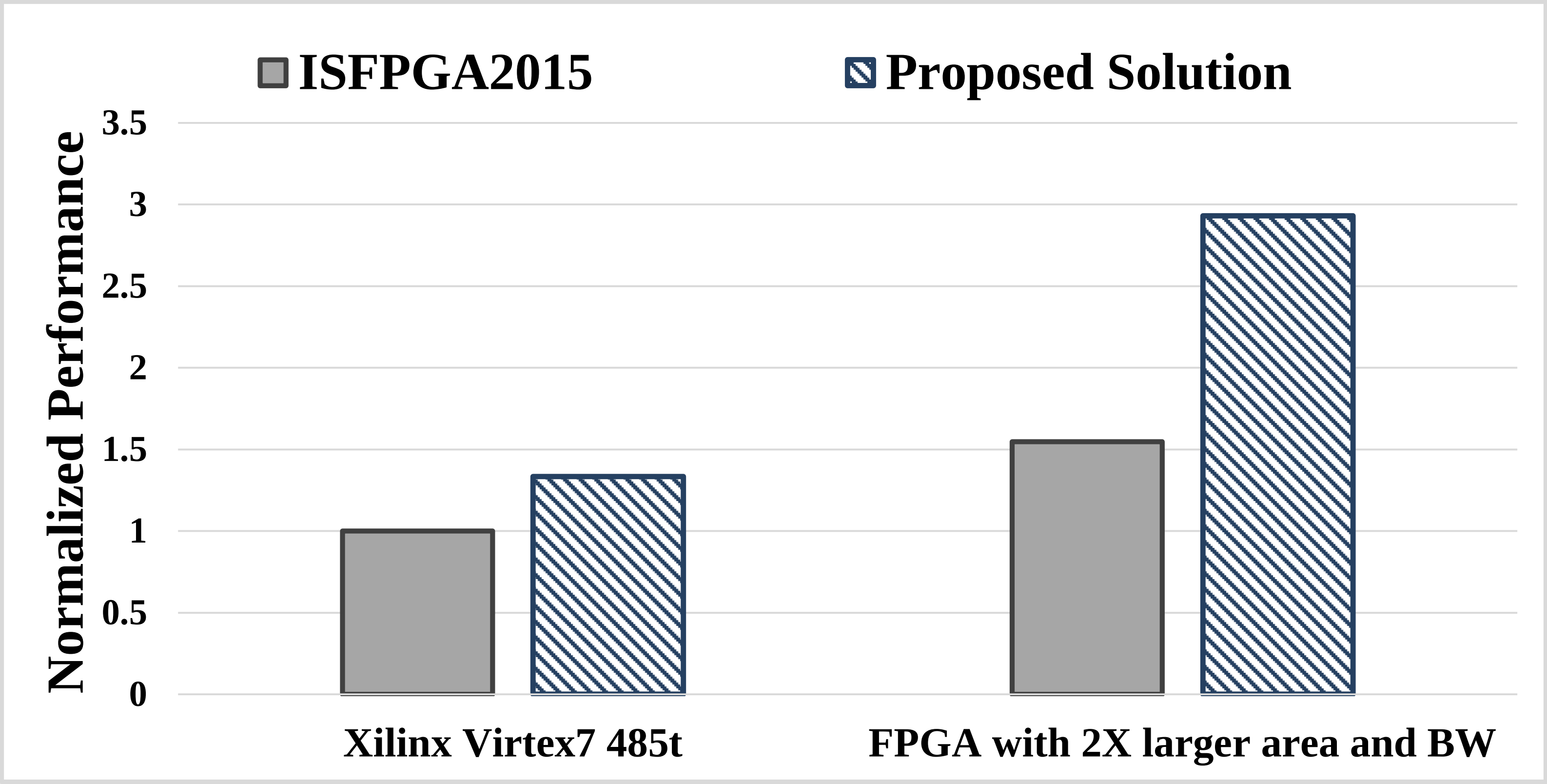


– Dynamic re-configurability has a minimal effect on the performance.

Performance Comparison (1)

	ICCD2013 [Peeman 2013]	**PACT2010 [Cadambi 2010]**	**ISCA2010 [Chakradhar 2010]**	**ISFPGA2015 [Zhang 2015]**	**Proposed Sol.**
Precision	Fixed	Fixed	Fixed	32bits float	32bits float
Frequency	150 MHz	125 MHz	200 MHz	100 MHz	100 MHz
FPGA Chip	VLX240T	SX240T	SX240T	VX485T	VX485T
Performance	17 GOPs	7.0 GOPs	16 GOPs	61.62 GFLOPs	84.2 GFLOPs
GOPs/Slice	4.5E-04	1.9E-04	4.3E-04	8.12E-04	11.09E-04

Performance Comparison (2)



Conclusion

- Template architecture for convolution acceleration
- Analytically characterize performance and memory requirements
- Expand the design space to find best architecture
  - Parallel Convolution Engines
  - Tiling scheme expanded to kernel level
- Simulation shows speedup of 1.4X … 1.9X over existing accelerators

Thank you!

References

- [Krizhevsky 2012] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
- [Simonyan 2015] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [Peeman 2013] M. Peemen, A. A. Setio, B. Mesman, and H. Corporaal. Memory-centric accelerator design for convolutional neural networks. In Computer Design (ICCD), 2013 IEEE 31st International Conference on, pages 13{19. IEEE, 2013.
- [Farabet 2009] Farabet, Clément, et al. "Neuflow: A runtime reconfigurable dataflow processor for vision." *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE, 2011.
- [Cadambi 2010] Cadambi, Srihari, et al. "A programmable parallel accelerator for learning and classification." *Proceedings of the 19th international conference on Parallel architectures and compilation techniques*. ACM, 2010.
- [Chakradhar 2010] Chakradhar, Srimat, et al. "A dynamically configurable coprocessor for convolutional neural networks." *ACM SIGARCH Computer Architecture News*. Vol. 38. No. 3. ACM, 2010.
- [Zhang 2015] Zhang, Chen, et al. "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks." *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2015.