SOC Design of a Neural Network for Real-Time Semantic Segmentation of 2Kx1K@60fps Video

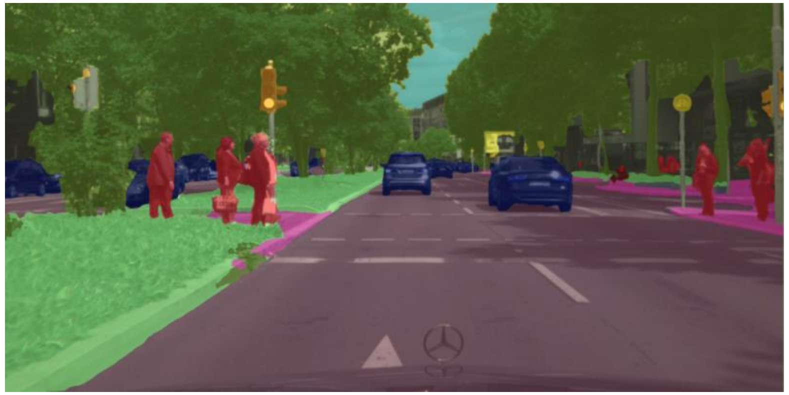
Youn-Long Lin

Department of Computer Science

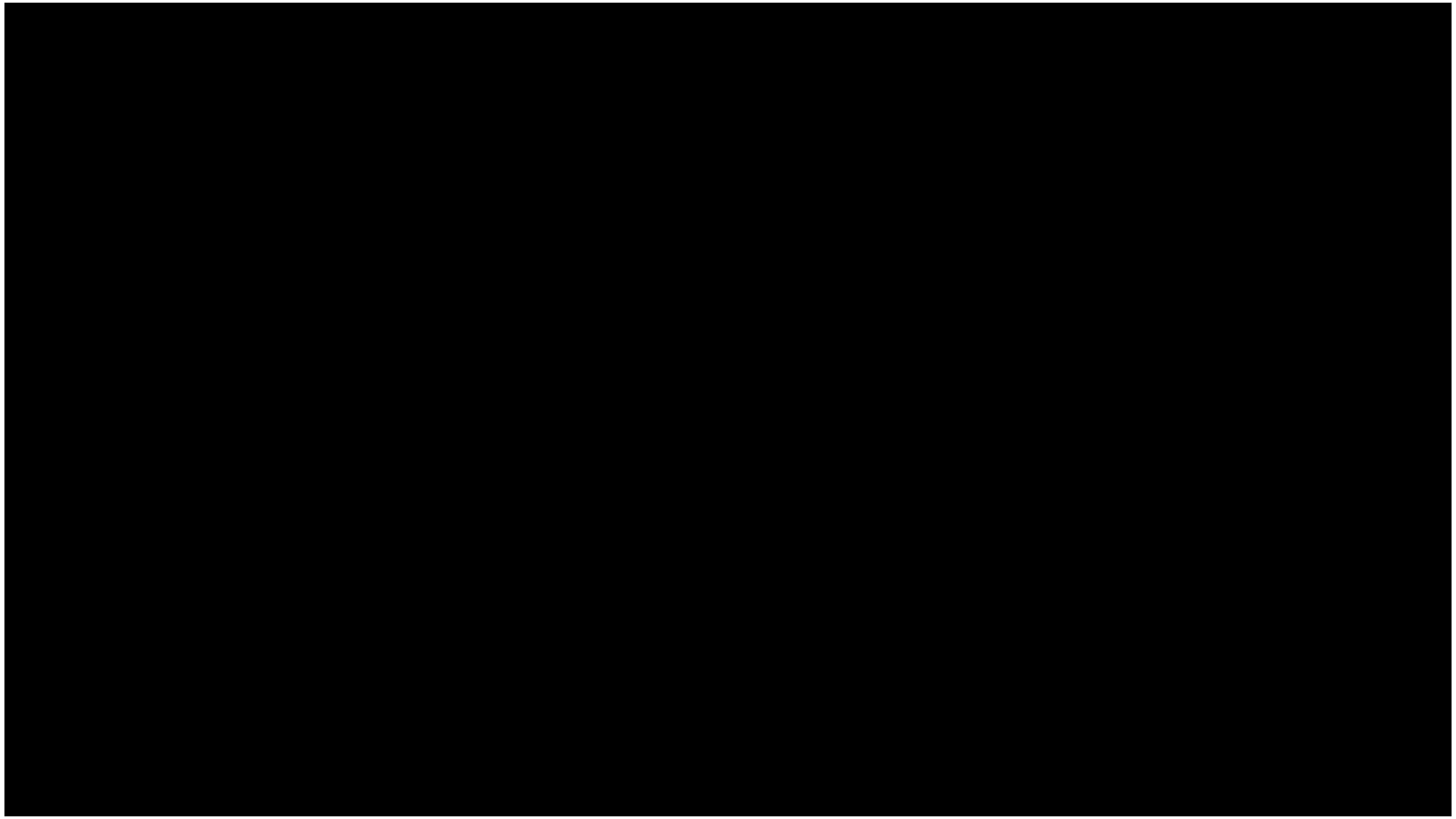
National Tsing Hua University

Project Goals

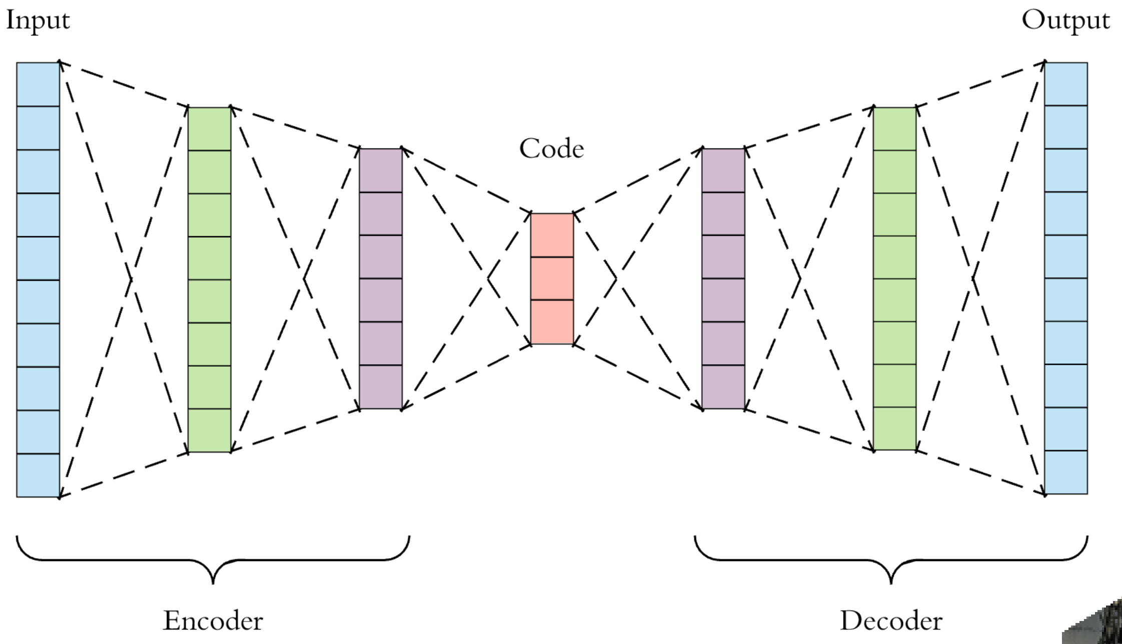
1. Real-time Semantic Segmentation of HD Video (1Kx2K@60fps)
2. Hardware-friendly neural network architecture
3. Proof-of-Concept using GPU
4. ASIC

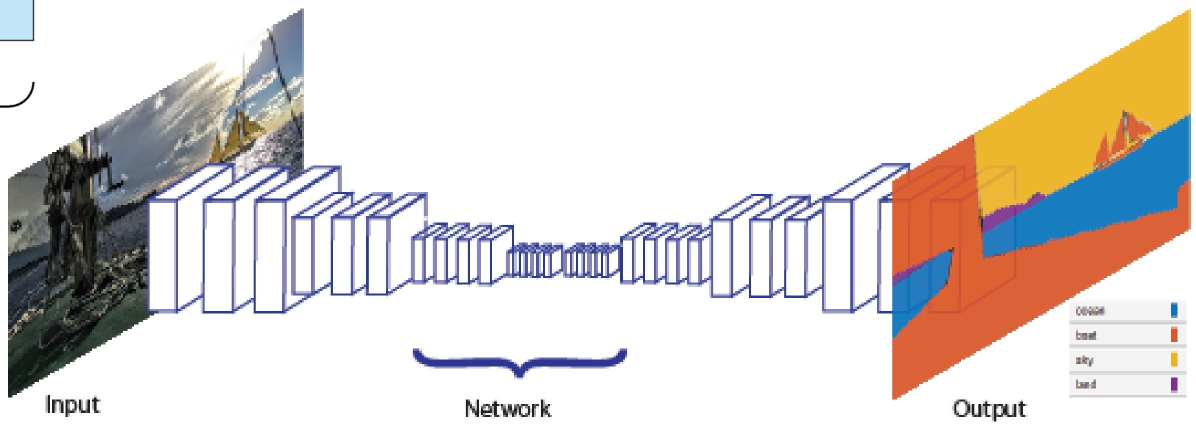


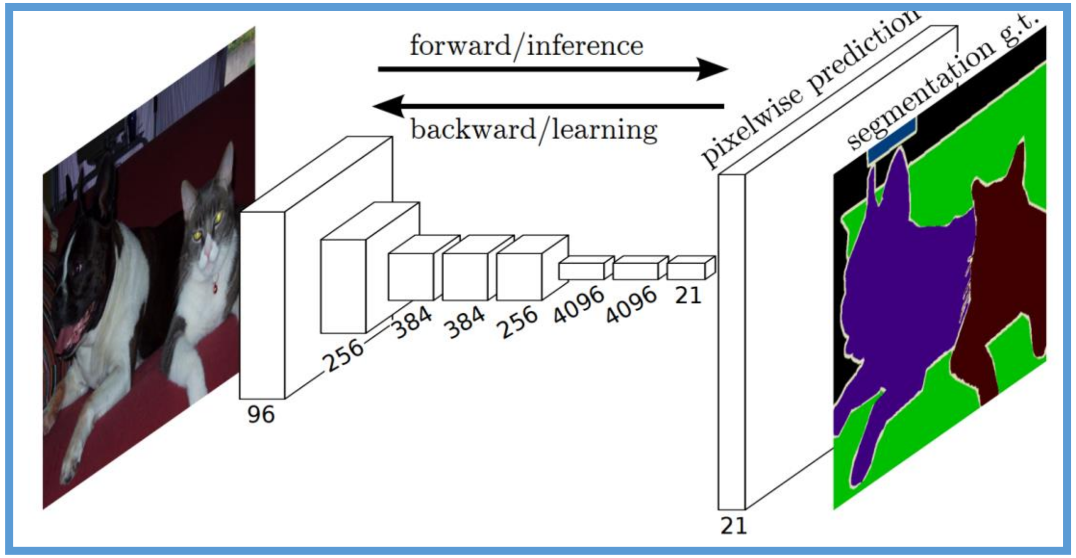
Semantic Segmentation – CityScapes Dataset

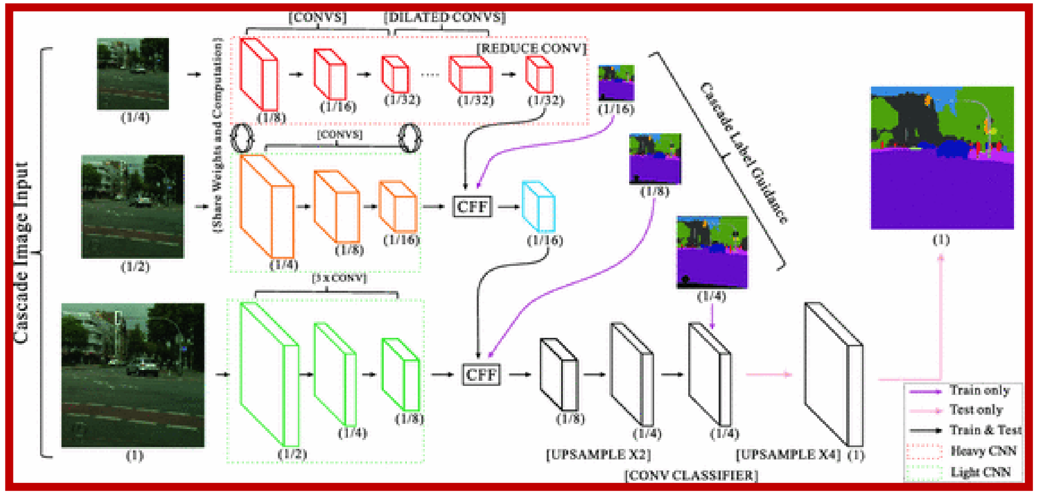


How to do it?
Autoencoder, Naturally





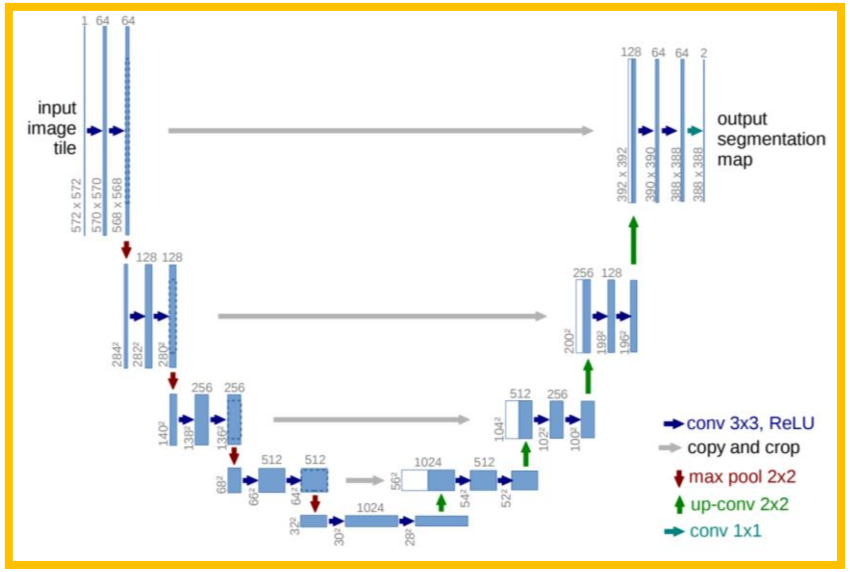




ICNet

FCN

Fully Convolutional Network

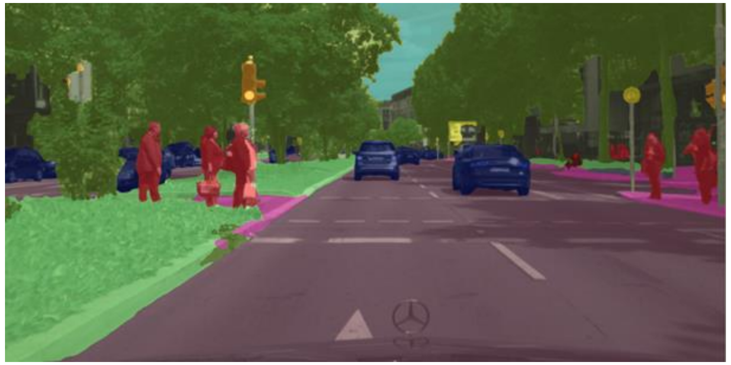


U-Net

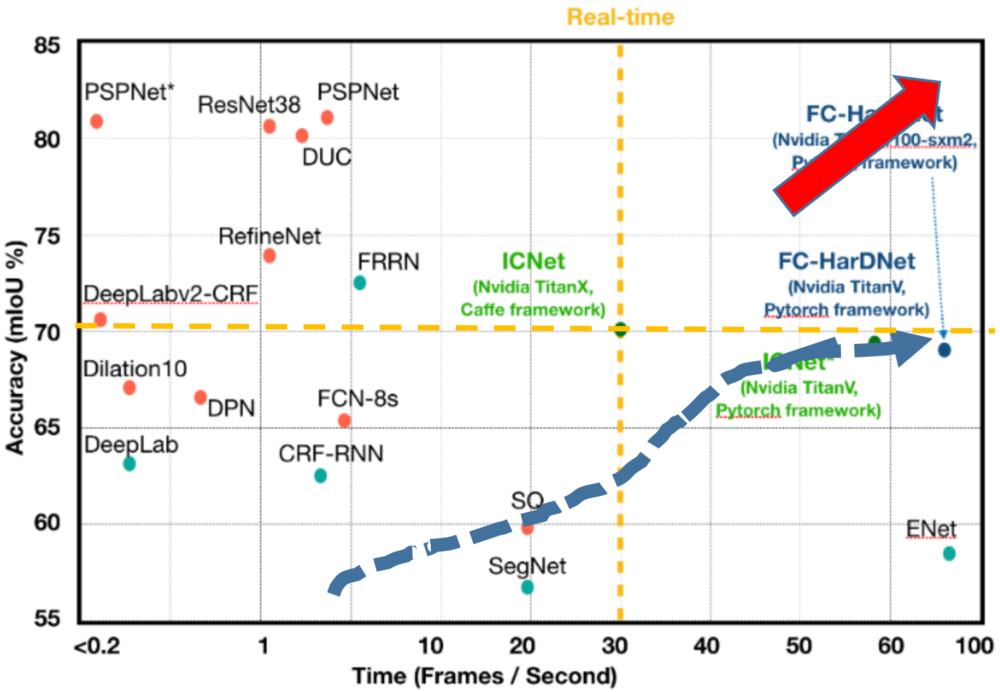
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).
- Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
- Zhao, H., Qi, X., Shen, X., Shi, J., & Jia, J. (2018). Icnet for real-time semantic segmentation on high-resolution images. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 405-420).

What we want for semantic segmentation?

mean IoU
(Intersect over Union)

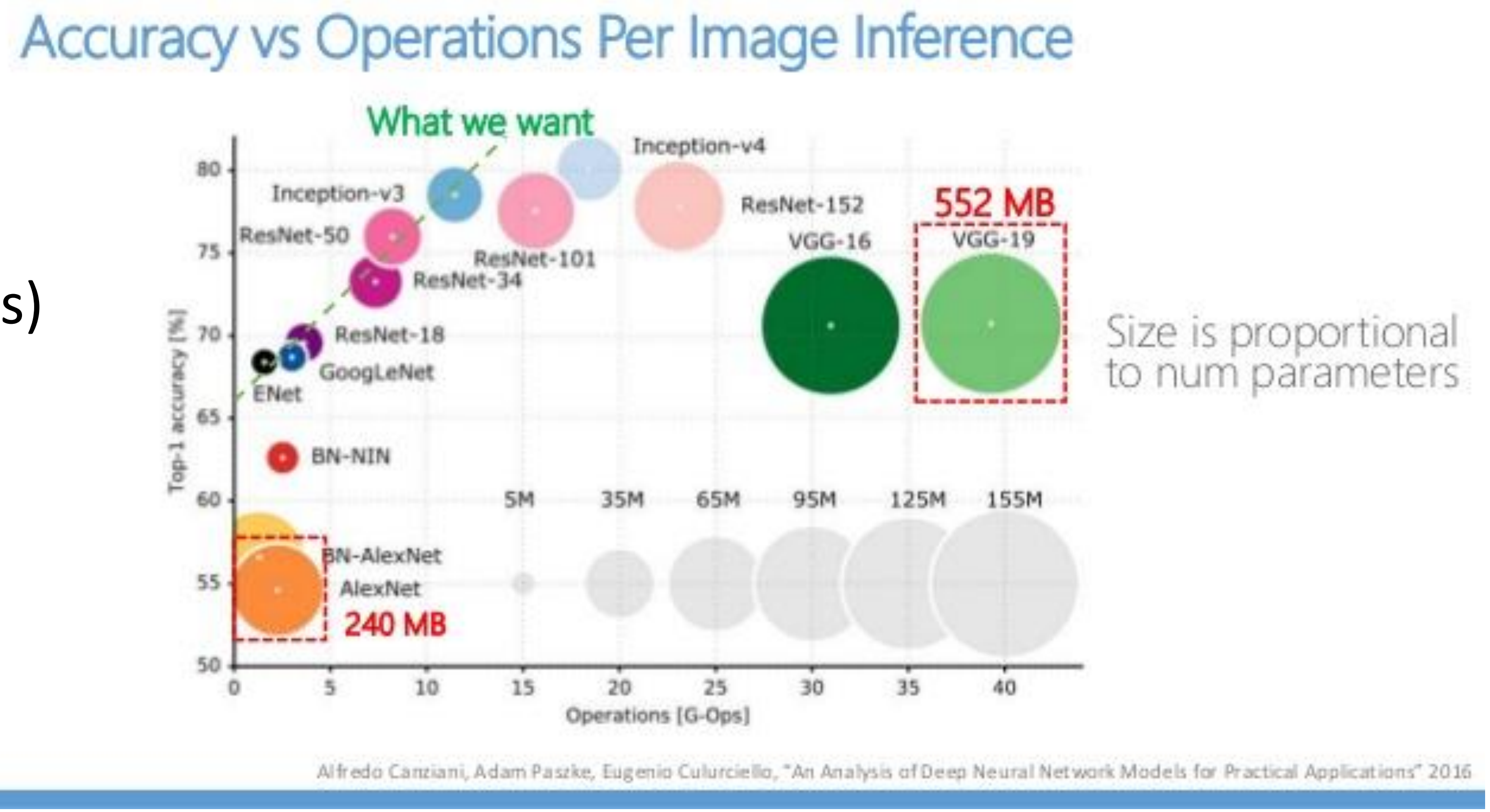


Accuracy vs frames per second

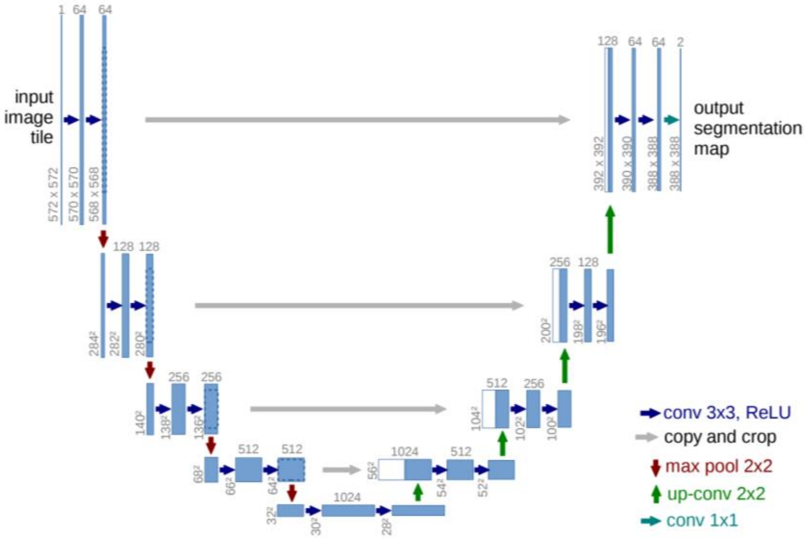


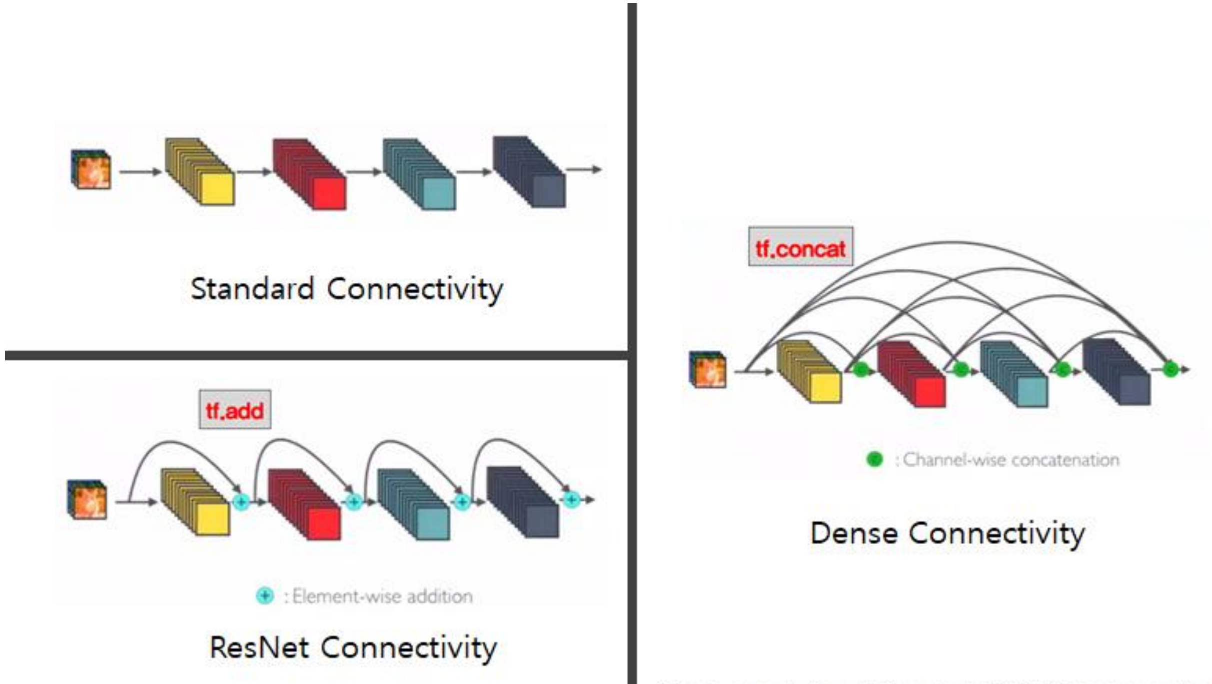
Network design tradeoffs

- Performance (Accuracy)
- Cost
  - Hardware gate count
  - Network size (Parameters)
- Inferencing
  - Time (# Operations?)
  - Energy consumption



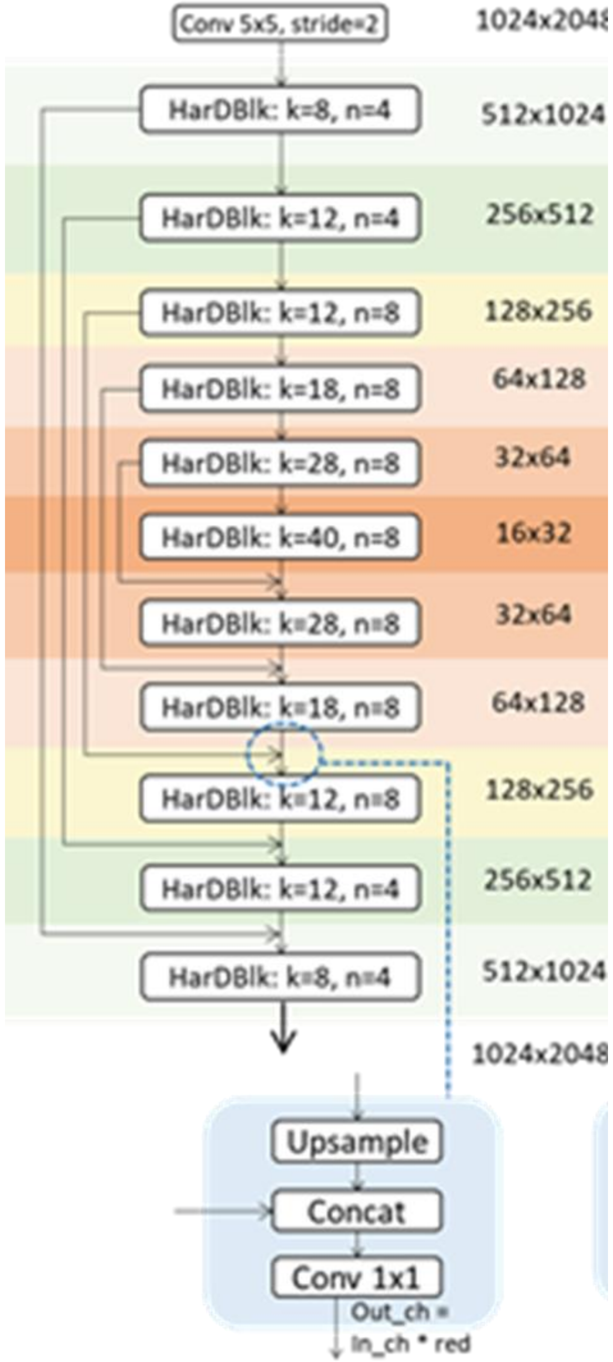
U-Net & DenseNet-Inspired Network Design



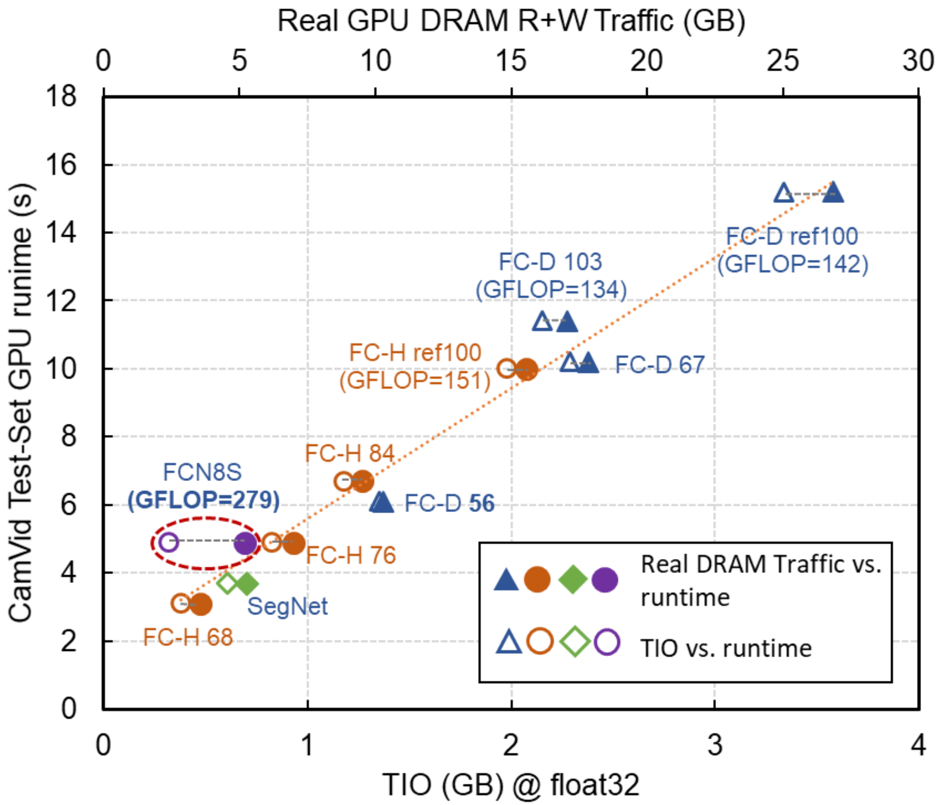


Densely connected convolution networks CVPR 2017 oral presentation slide

Proposed Network



DRAM Traffic vs Run-Time



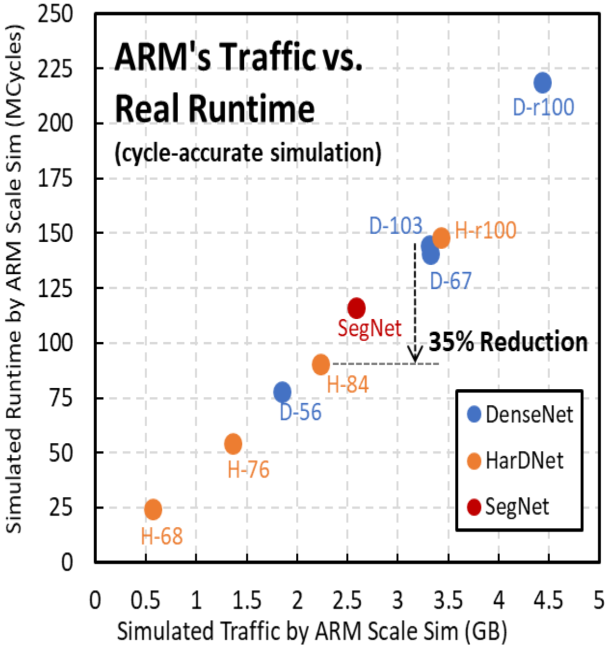
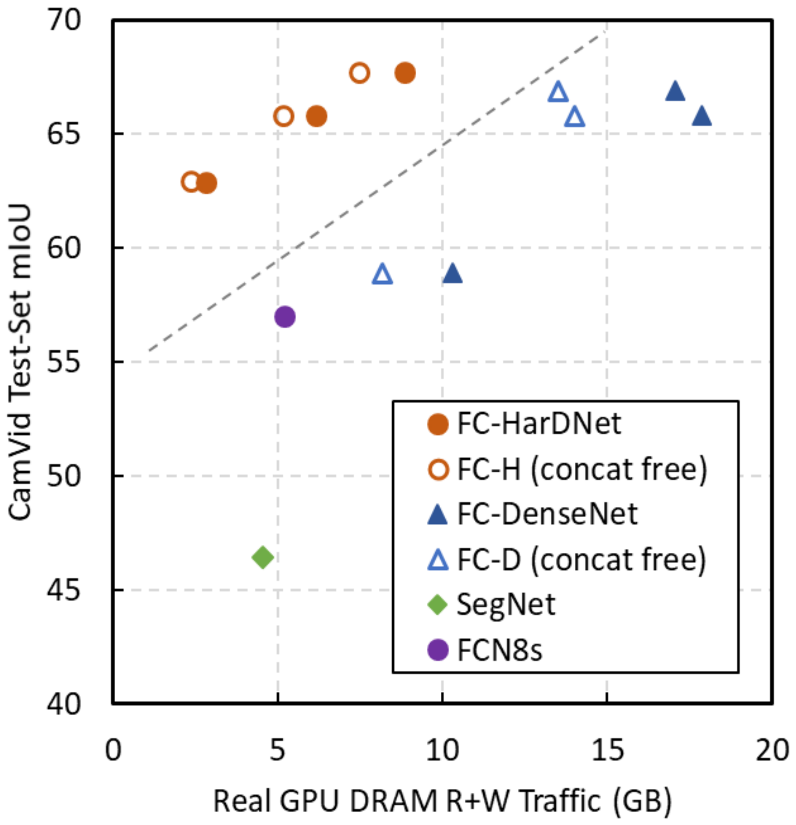


Figure 3: Runtime vs. DRAM traffic measured by the simulation of ARM Scale.

Low DRAM Traffic (Run-Time) and High Accuracy



To make an ASIC

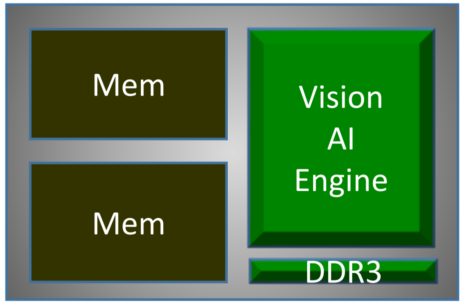
USD8000/300W

SPECIFICATIONS

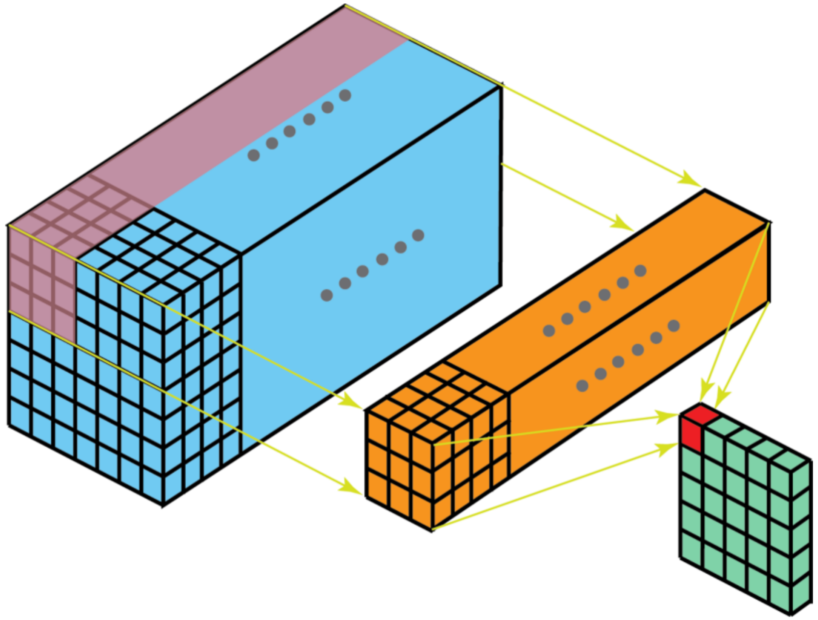
	Tesla V100 PCIe	Tesla V100 SXM2
GPU Architecture	NVIDIA Volta	
NVIDIA Tensor Cores	640	
NVIDIA CUDA® Cores	5,120	
Double-Precision Performance	7 TFLOPS	7.5 TFLOPS
Single-Precision Performance	14 TFLOPS	15 TFLOPS
Tensor Performance	112 TFLOPS	120 TFLOPS
GPU Memory	16 GB HBM2	
Memory Bandwidth	900 GB/sec	
ECC	Yes	
Interconnect Bandwidth*	32 GB/sec	300 GB/sec
System Interface	PCIe Gen3	NVIDIA NVLink
Form Factor	PCIe Full Height/Length	SXM2
Max Power Comsumption	250 W	300 W
Thermal Solution	Passive	
Compute APIs	CUDA, DirectCompute, OpenCL™, OpenACC	

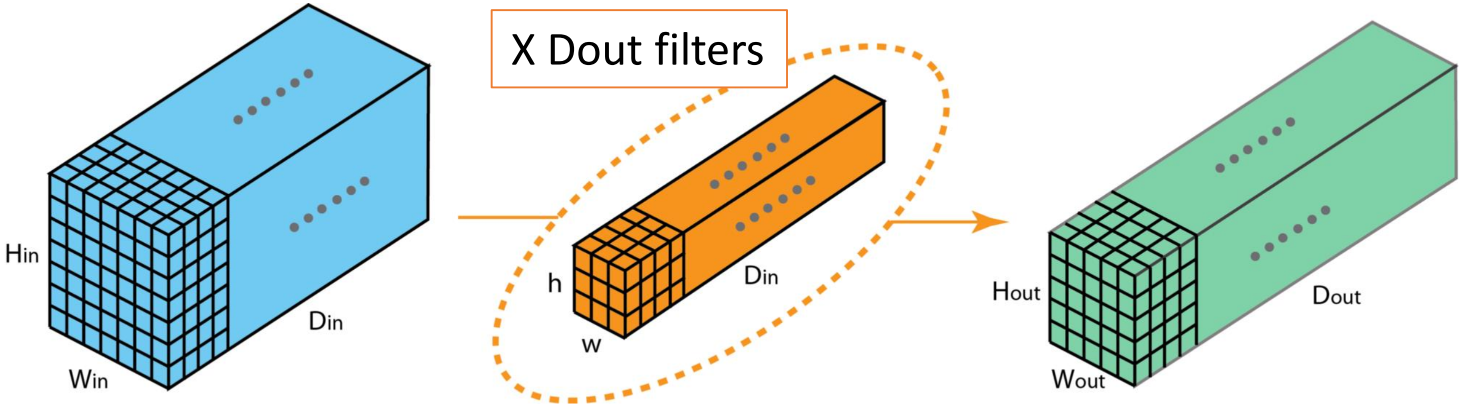
USD100/10W

ASIC



> 99% Computes are 2D Convolution (Multi-Channel, Multi-Filter)

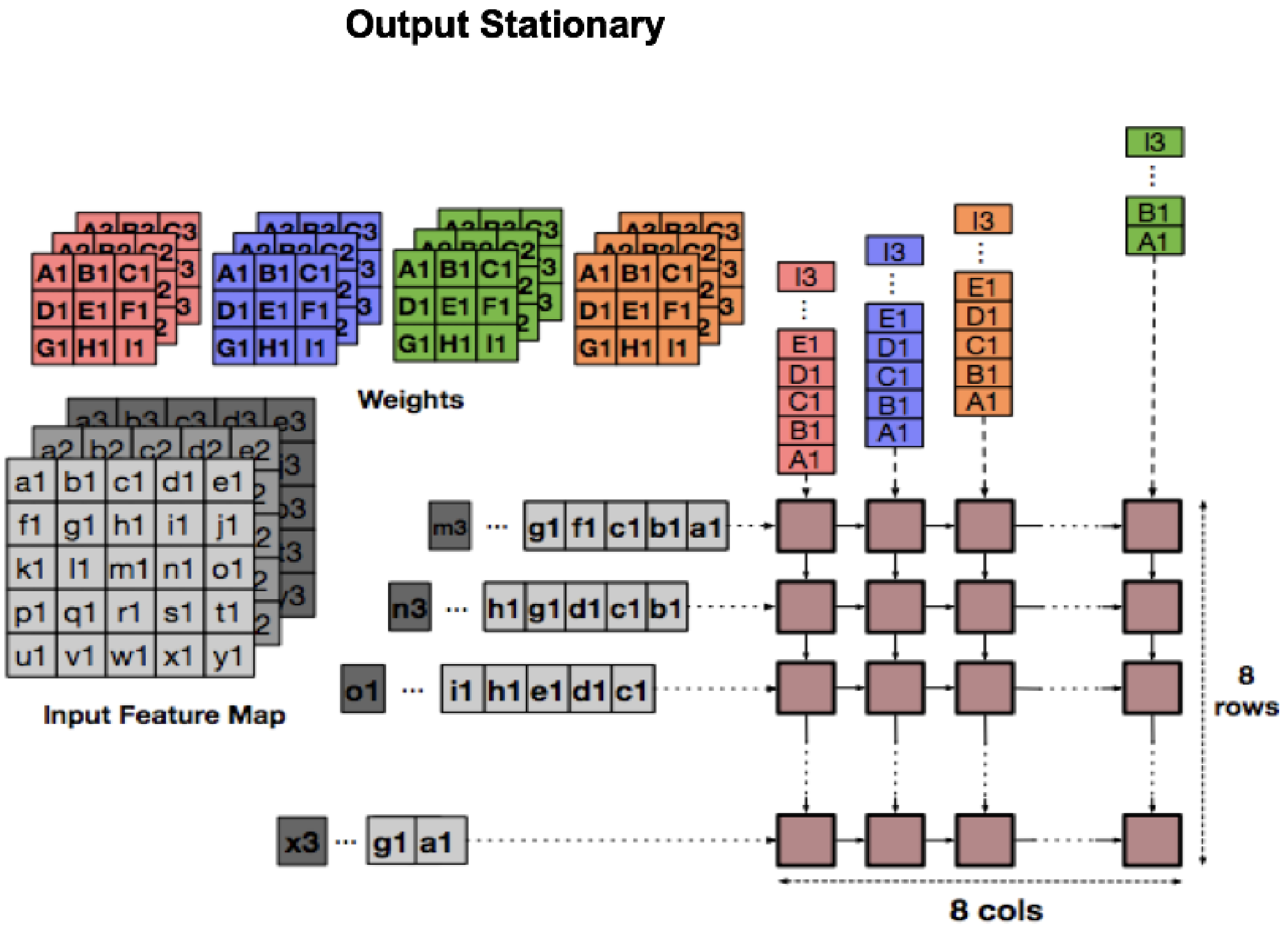





```
for (m = 0; m < numOutputLayers; m++)          //Loop 1
  for (n = 0; n < numInputLayers; n++)         //Loop 2
    for (h = 0; h < outputHeight; h++)         //Loop 3
      for (w = 0; w < outputWidth; w++)        //Loop 4
        for (i = 0; i < kernelHeight; i++)     //Loop 5
          for (j = 0; j < kernelWidth; j++)    //Loop 6
            out[m][h][w] +=
              in[n][h * S + i][w * S + j] *
              kernel[m][n][i][j];
```

A Comprehensive Introduction to Different Types of Convolutions in Deep Learning -- Towards intuitive understanding of convolutions through visualizations
Kunlun Bai

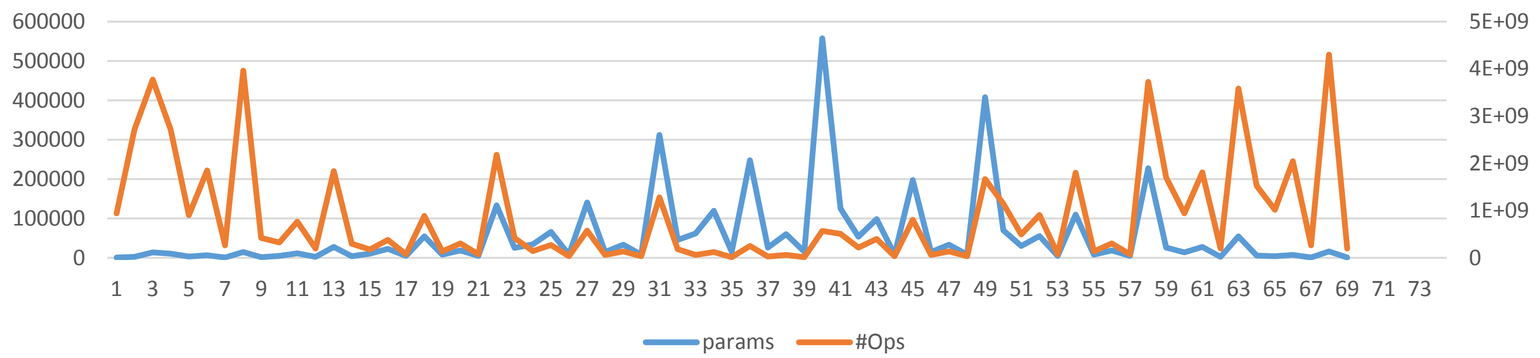
2D Convolution with Systolic Array



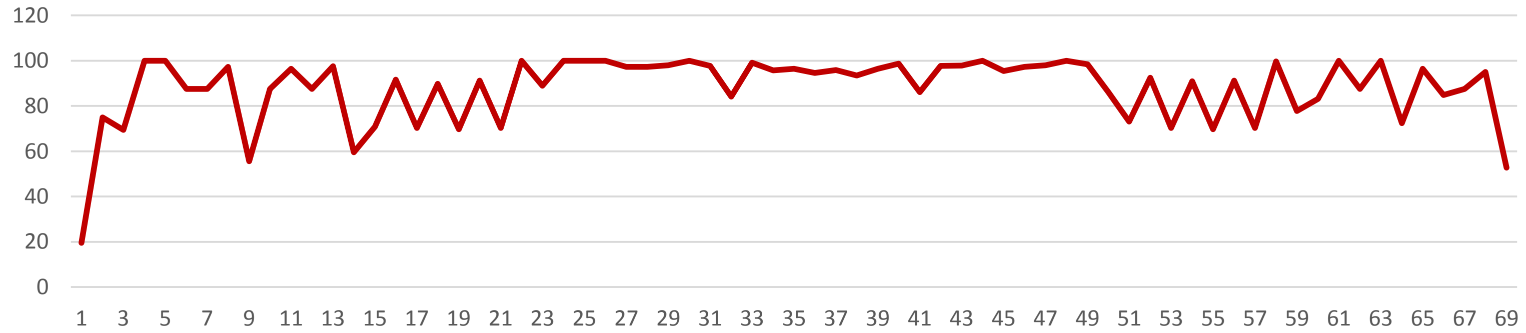
Citation: Ananda Samajdar, Yuhao Zhu, Paul Whatmough, Matthew Mattina, and Tushar Krishna. Scale-sim: Systolic cnn accelerator. arXiv preprint arXiv:1811.02883, 2018.

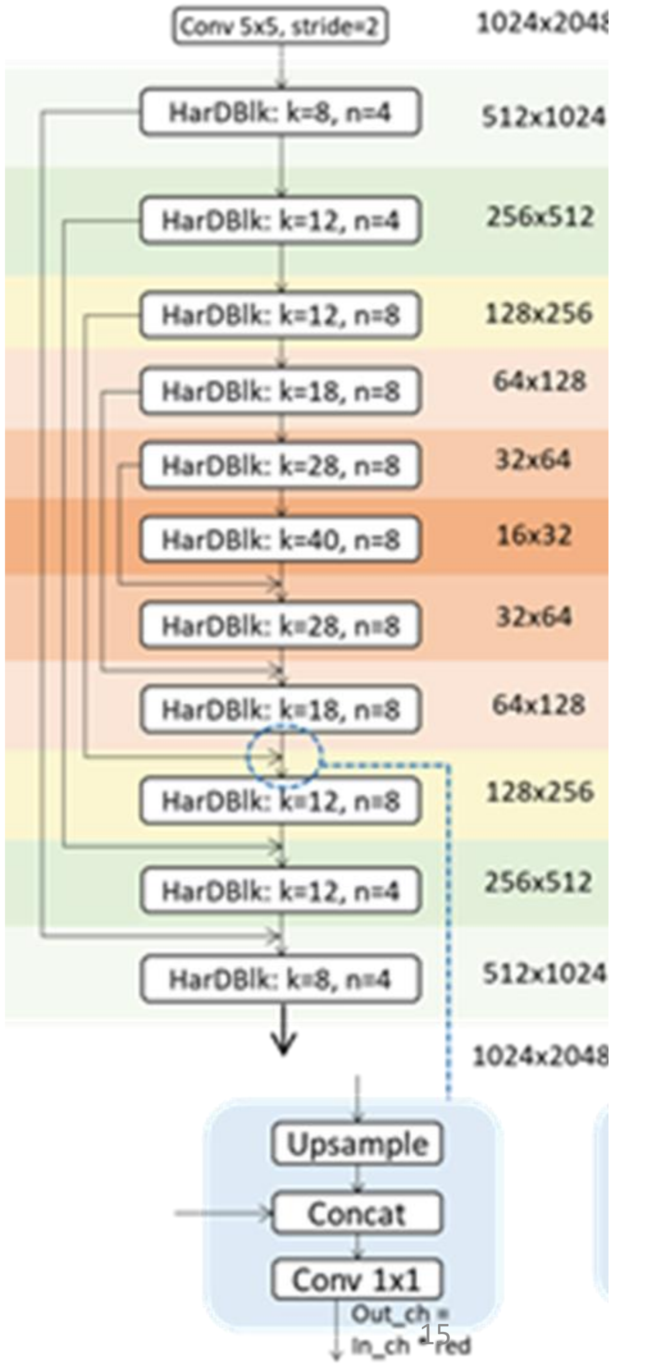
Memory bound vs Compute Bound

Per Layer # Paras vs #Ops



Per Layer Utilization of 9216 MACs





Results

- Network
  - 69-Layer Convolutional Neural Network (Other versions: 84, …)
  - 3.8M parameters
  - 59.685G Operations per 1Kx2K frame inference
- PyTorch on GPU Implementation
  - 80fps on a TWCC nVidia Tesla V100 32GB GPU (300Watt, USD8,000)
  - 59.658Gops * 80 / 120Tops ~ 4% utilization
- Preliminary ASIC Design
  - 9216 MACs (Mutli-Add) = 18,432 PEs
  - Peak performance 9.216 Tera ops @ 500 MHz
  - 3.846M Clock Cycles to inference a 1Kx2K frame
  - 59.685G / (3.846M * 18432) ~ 85% utilization

Thank you!!