CAP 5516
Medical Image Computing
(Spring 2025)

Dr. Chen Chen
Associate Professor
Center for Research in Computer Vision (CRCV)
University of Central Florida
Office: HEC 221
Email: chen.chen@crcv.ucf.edu
Web: https://www.crcv.ucf.edu/chenchen/

Lecture 11
Efficient Deep Learning (1)

Real-world cases: how AI transformed biomedicine & healthcare

Case 1: CT-based pneumonia detection during the COVID-19 outbreak in China

- Each patient typically need to undergo CT imaging about 4 times from admission to discharge.
- For every CT scanning, staff need to manually contour three to four hundred CT images, and count the lung lobes or segments, and calculate the range of lesions in them to assess the severity.

This process can take **up to five to six hours**! But using AI, we just need **less than 20 seconds** for each scan, with a final accuracy rate of **over 90%**.

Information from China Science and Technology Museum

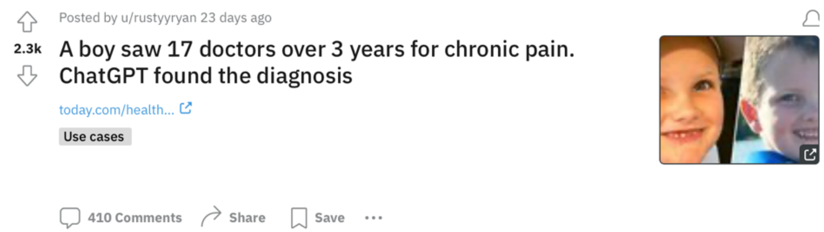
Slide Credit: Kai Zhang

Real-world cases: how AI transformed biomedicine & healthcare

Case 2: AI provides diagnosis for reference

- The mother plugged MRI notes into ChatGPT and got the suggestion that there may be Tethered Cord Syndrome (TCS).

- The boy visited neurosurgeon and finally being diagnosed and treated correctly.



No matter how many doctors the family saw, "the specialists would only address their individual areas of expertise", his mother says.

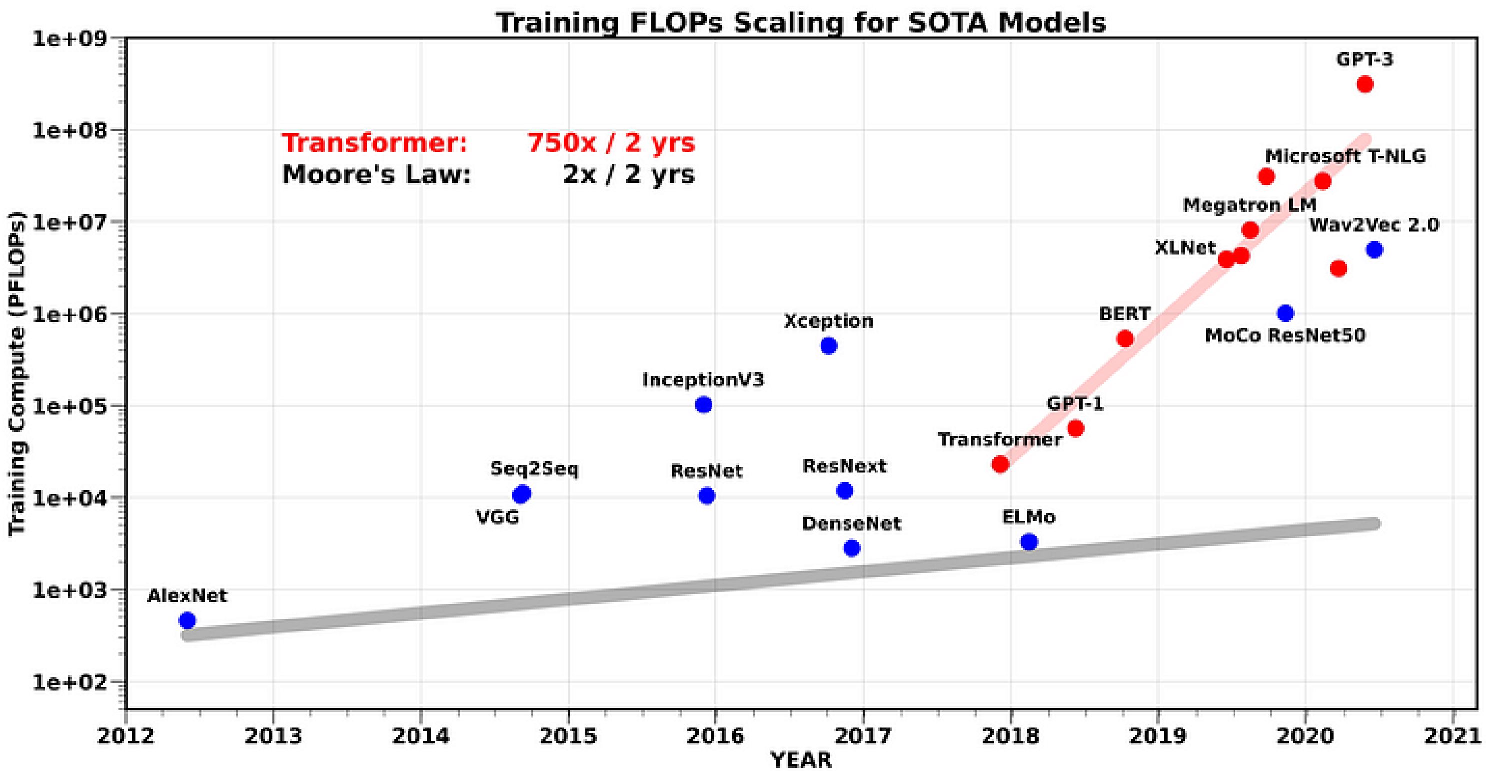
Information from Quora, today.com, RadiologyBusiness.com, and MDLinx.com

Slide Credit: Kai Zhang

UCF Center for Research in Computer Vision

AI Models, Compute, and Memory Wall

1 petaFLOPS (PFLOPS):
one quadrillion (10^{15}) floating-point operations



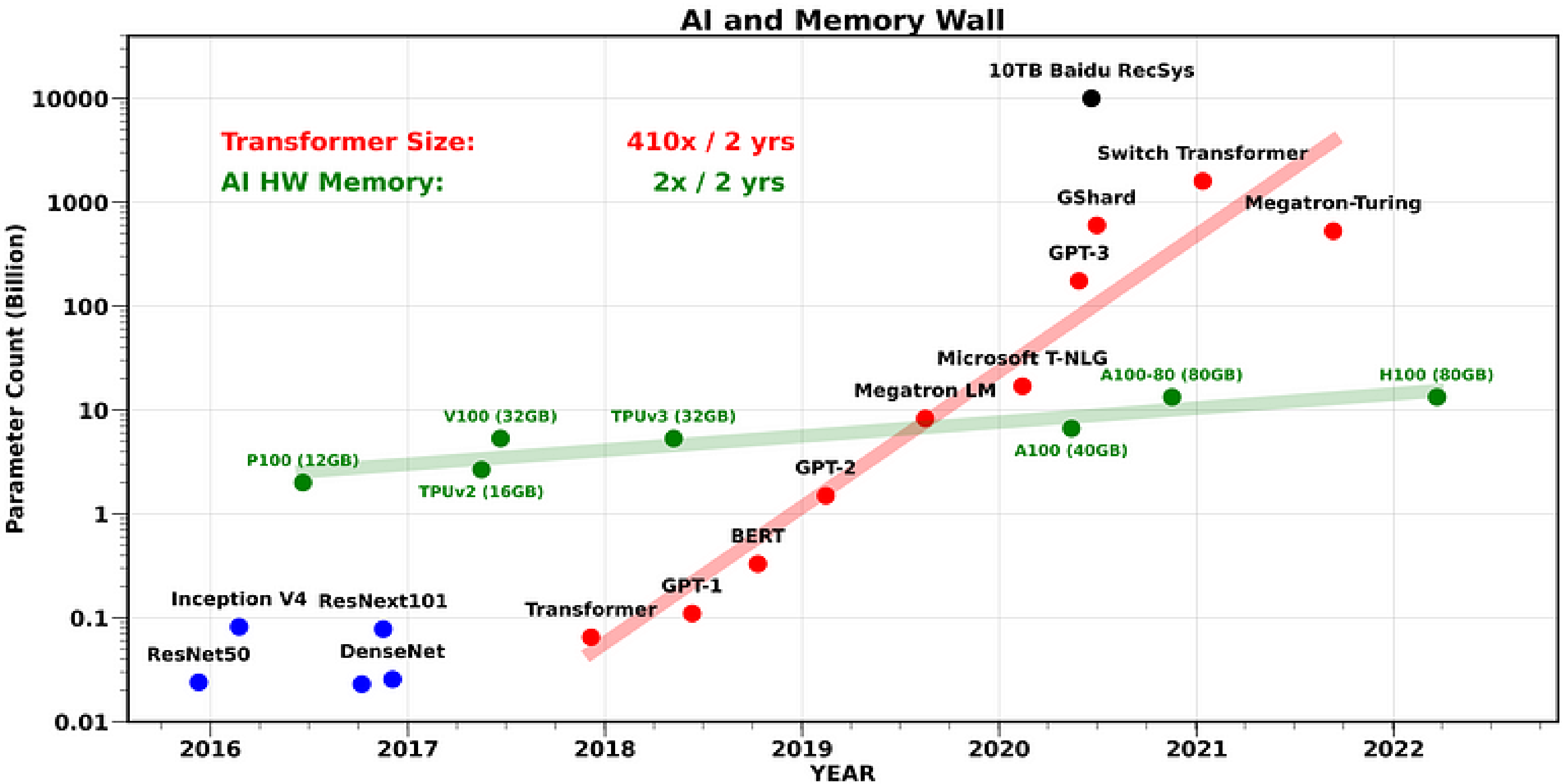
https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

The amount of compute needed to train SOTA Transformer models, has been growing at a rate of 750x/2yrs

Moore's Law: the principle that the speed and capability of computers can be expected to double every two years, as a result of increases in the number of transistors a microchip can contain.

UCF Center for Research in Computer Vision

AI Models, Compute, and Memory Wall



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

The number of parameters in large Transformer models has been exponentially increasing with a factor of 410x every two years, while the single GPU memory has only been scaled at a rate of 2x every 2 years

UCF CENTER FOR RESEARCH IN COMPUTER VISION

Challenges

- **Advanced Infrastructure Needs**: Training state-of-the-art deep learning models requires extensive computational power and substantial memory resources, often unavailable in the medical domain.

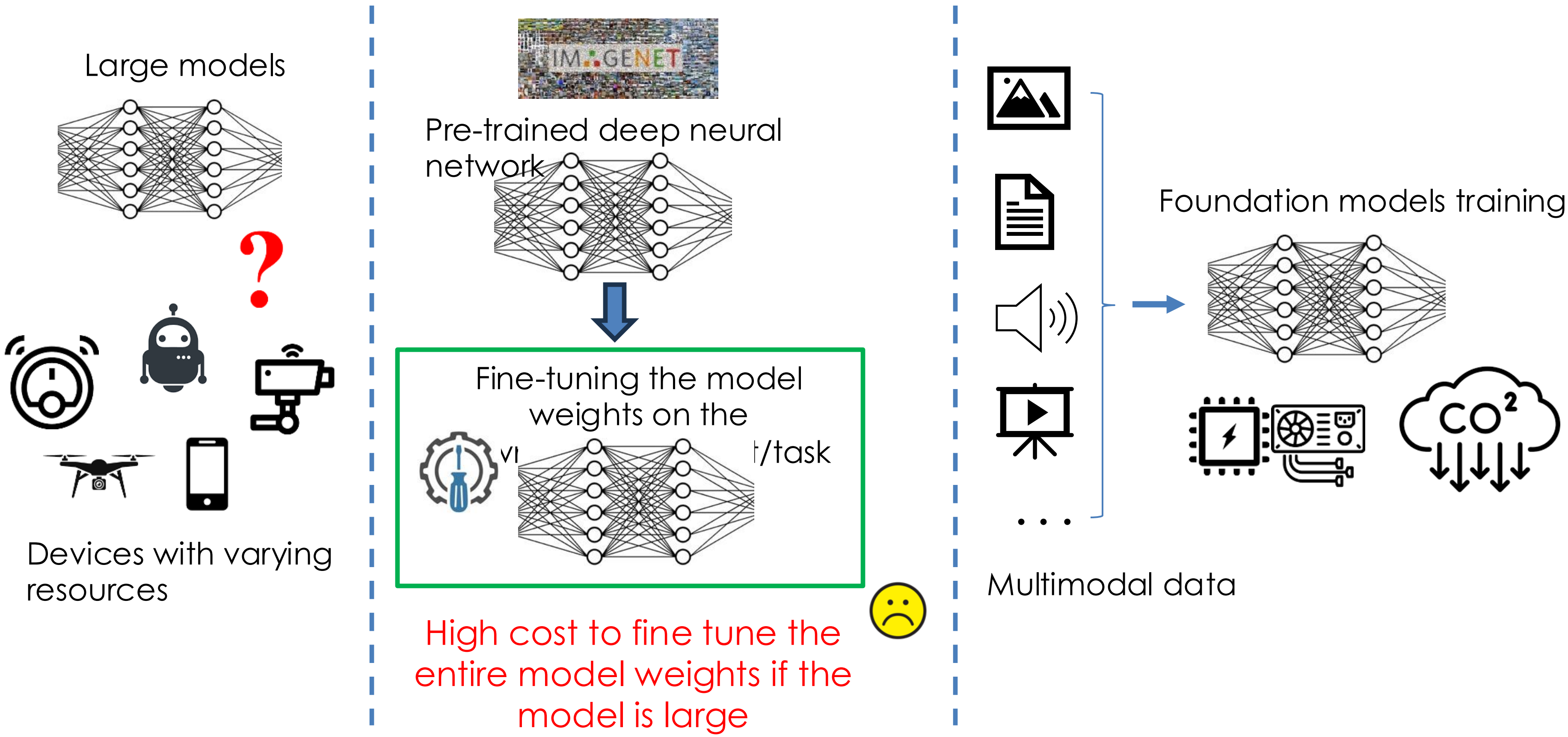
- **Resource Disparity**: Many hospitals, especially in less economically developed areas, lack the necessary GPU resources and rely solely on CPU machines, which are significantly less efficient for deep learning tasks.

- **Infrastructure Overhaul Challenges**: Upgrading existing hospital infrastructure to include advanced computational resources is a complex and costly endeavor.

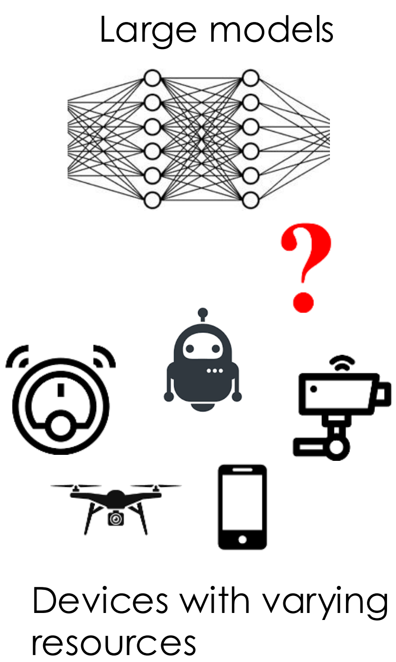
Efficient Deep Learning in Medical Imaging

- **Efficiency as a Necessity**: Given these constraints, there is a critical need to develop efficient machine learning models that can operate within the available infrastructure, particularly for crucial applications such as medical image analysis.

Research Challenges



Research Challenges

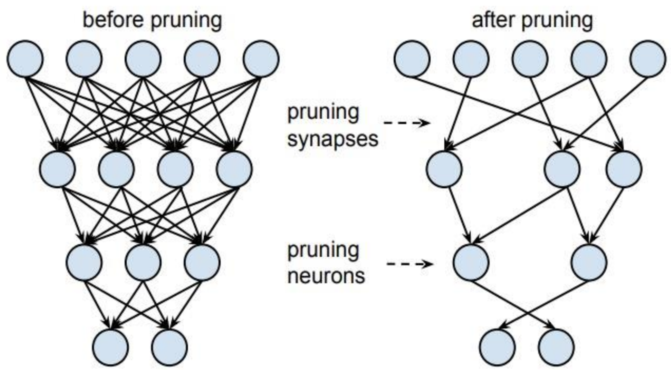


Efficient Neural Networks Design

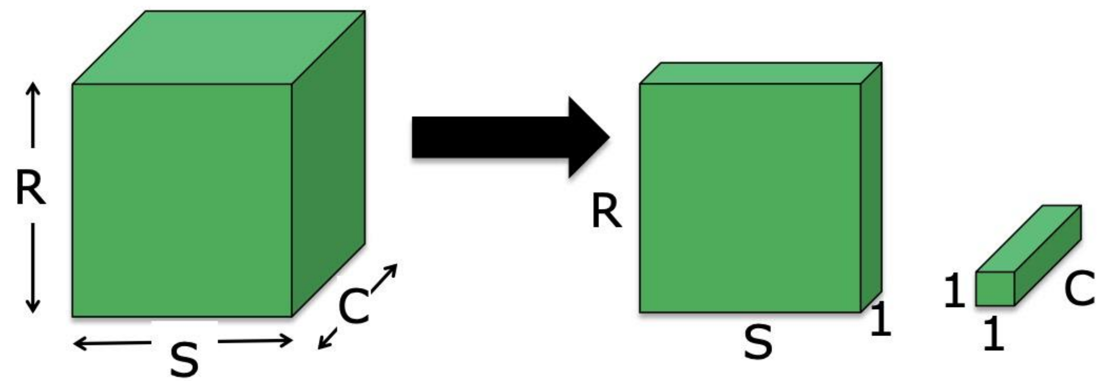
Efficient Neural Networks Design

Credit: Vivienne Sze

Network Pruning



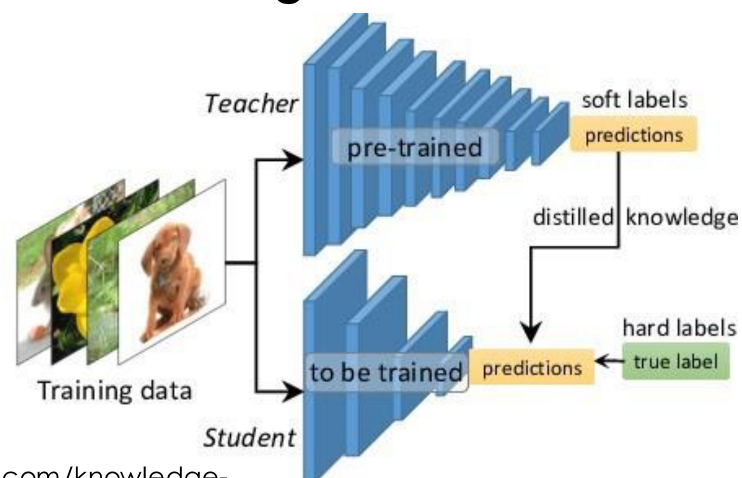
Efficient Network Architectures



[MobileNets, ShuffleNets, AdderNet]

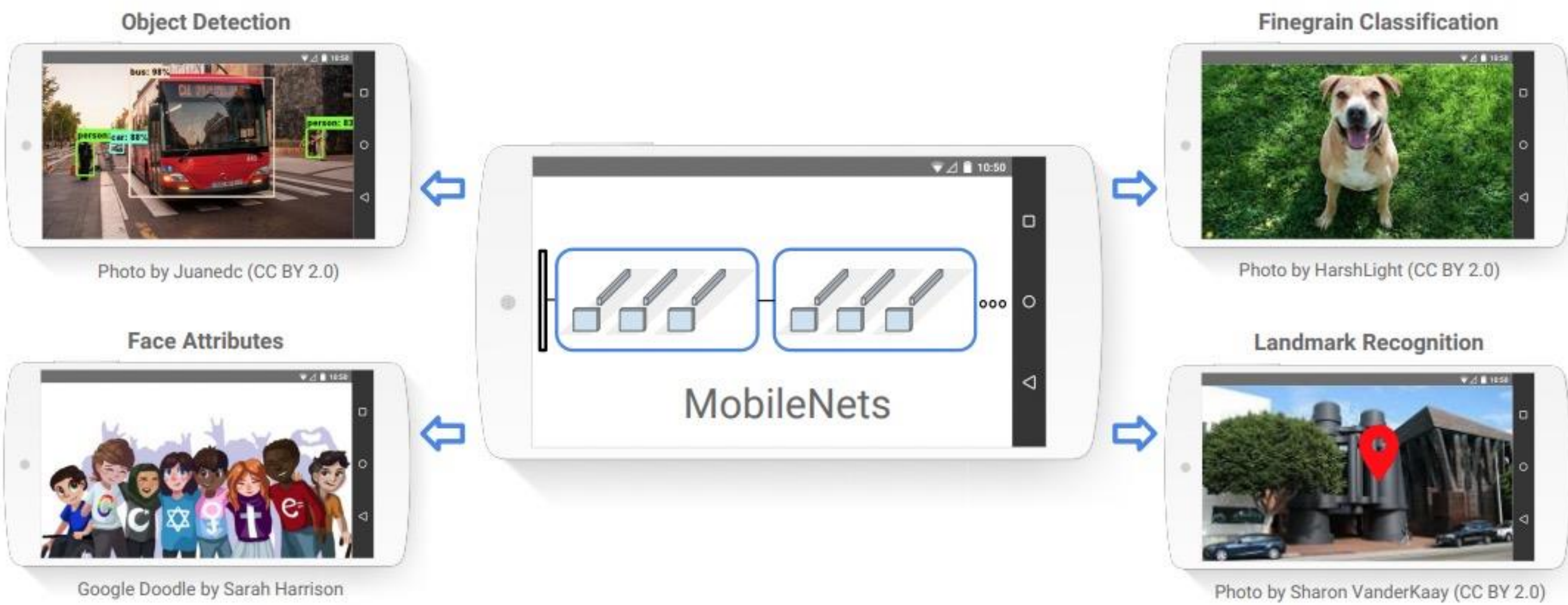
Reduce Precision

Knowledge Distillation



Source: https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764

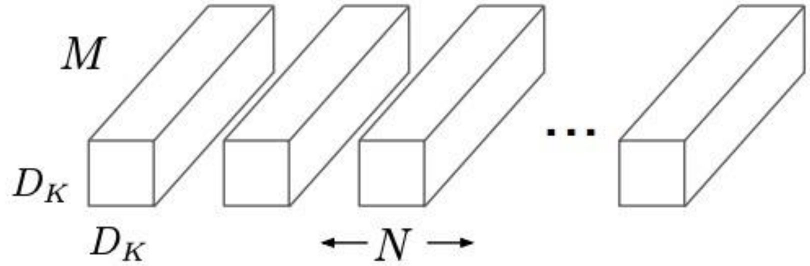
Efficient Network Architectures



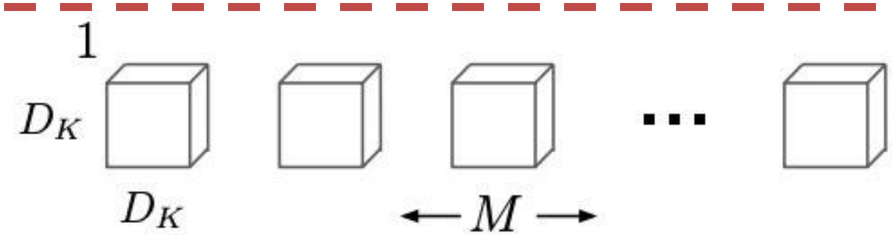
Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

Efficient Network Architectures

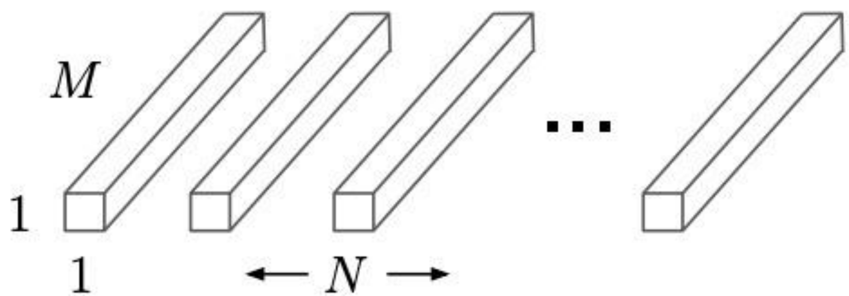
- MobileNet



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters

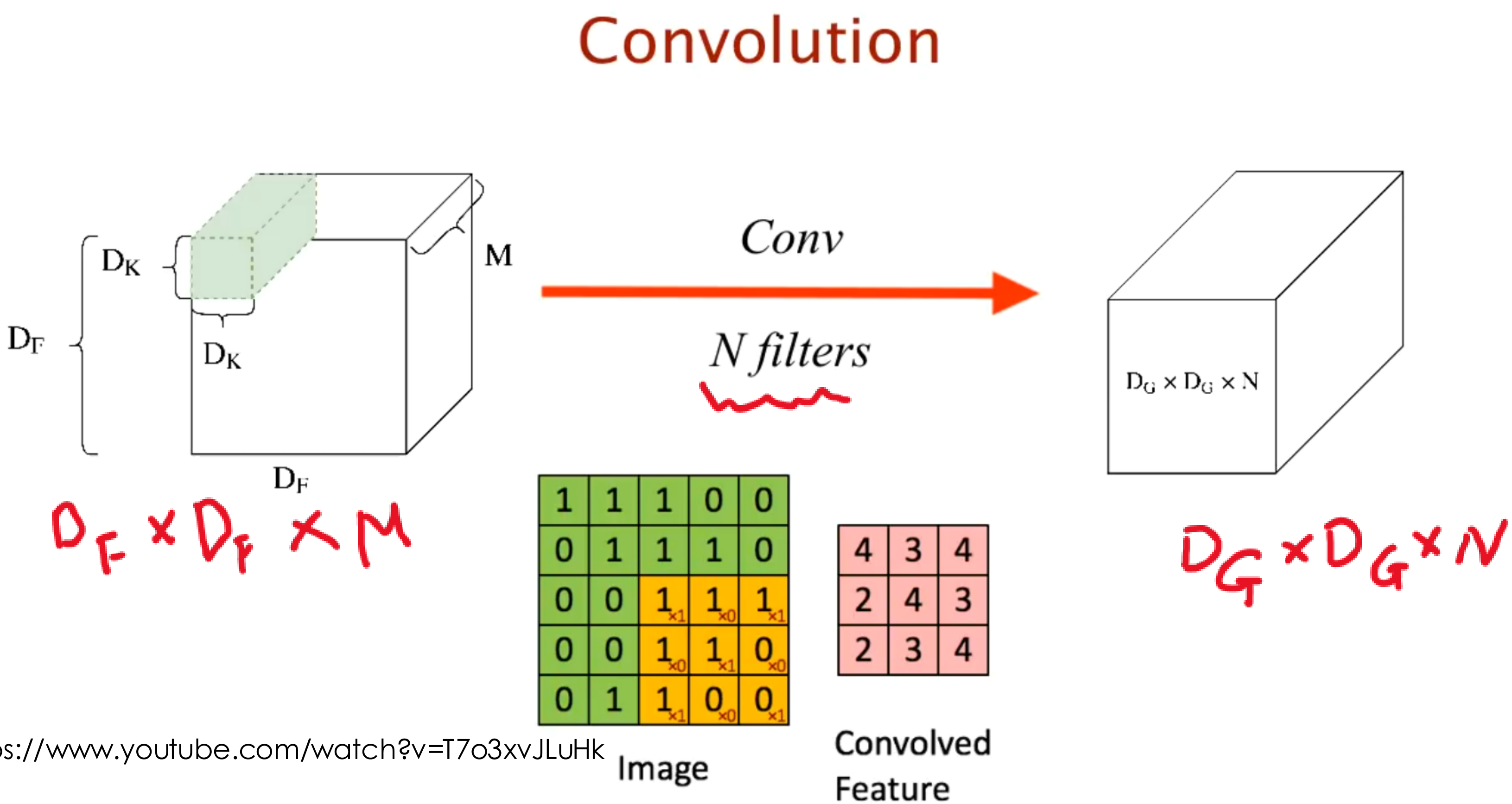


(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

MobileNet

- Determine the number of multiplications



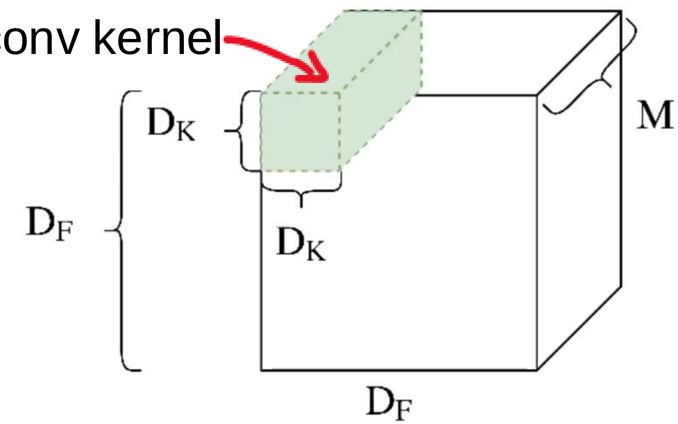
Source: https://www.youtube.com/watch?v=T7o3xvJLuHk

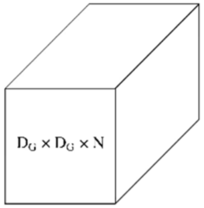
MobileNet

- Determine the number of multiplications

Convolution

One standard conv kernel





Output feature map

$D_G \times D_G \times N$

Multiplications one position ?

$$D_K^2 \times M$$

Multiplications one kernel over the entire input?

$$D_G^2 \times D_K^2 \times M$$

Multiplications N kernel ?

$$N \times D_G^2 \times D_K^2 \times M$$

MobileNet

Depthwise Separable Convolution

1. Depthwise Convolution: Filtering Stage

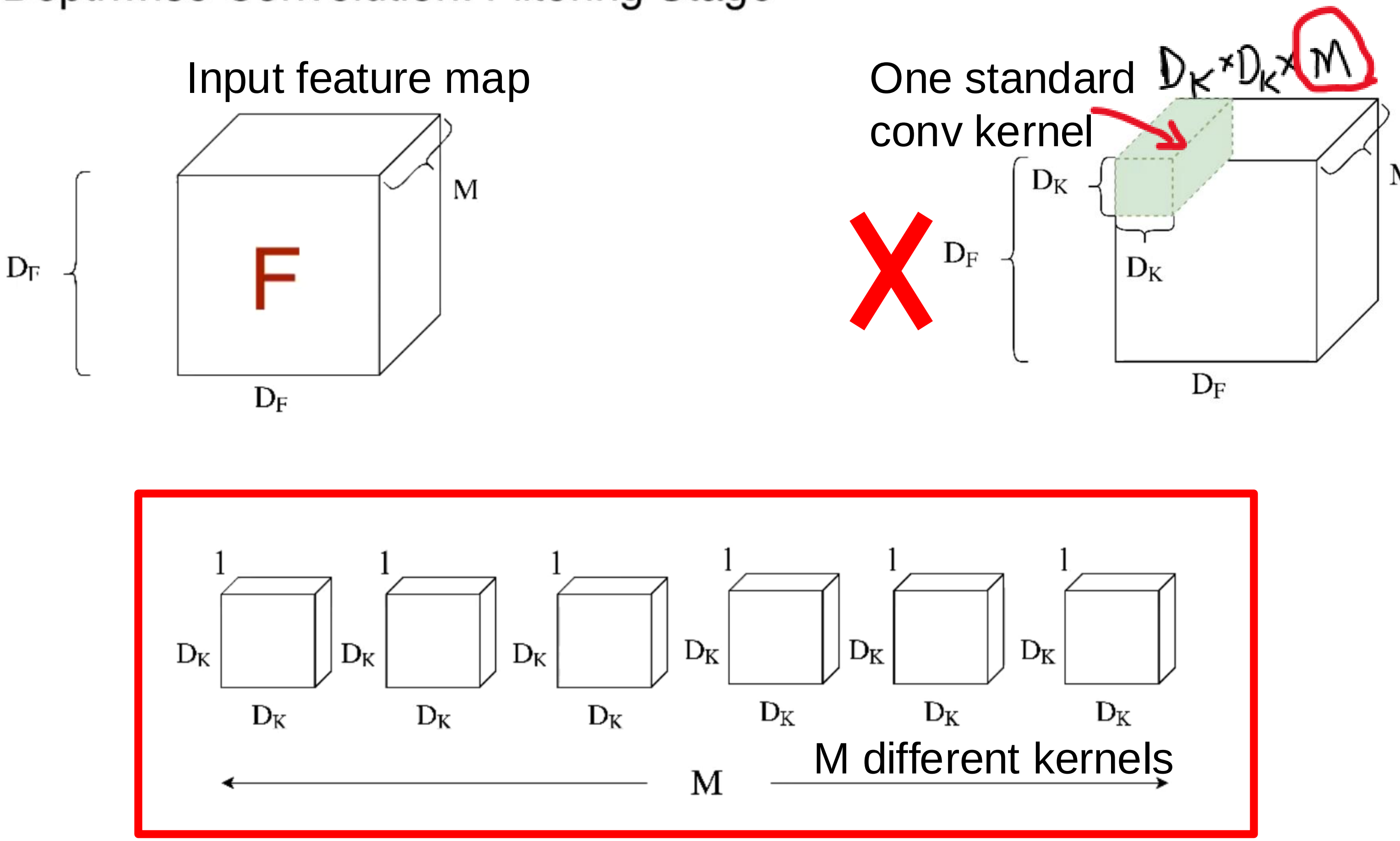
2. Pointwise Convolution: Combination Stage

Reduce computation (# of multiplications)

MobileNet

Depthwise Separable Convolution

1. Depthwise Convolution: Filtering Stage

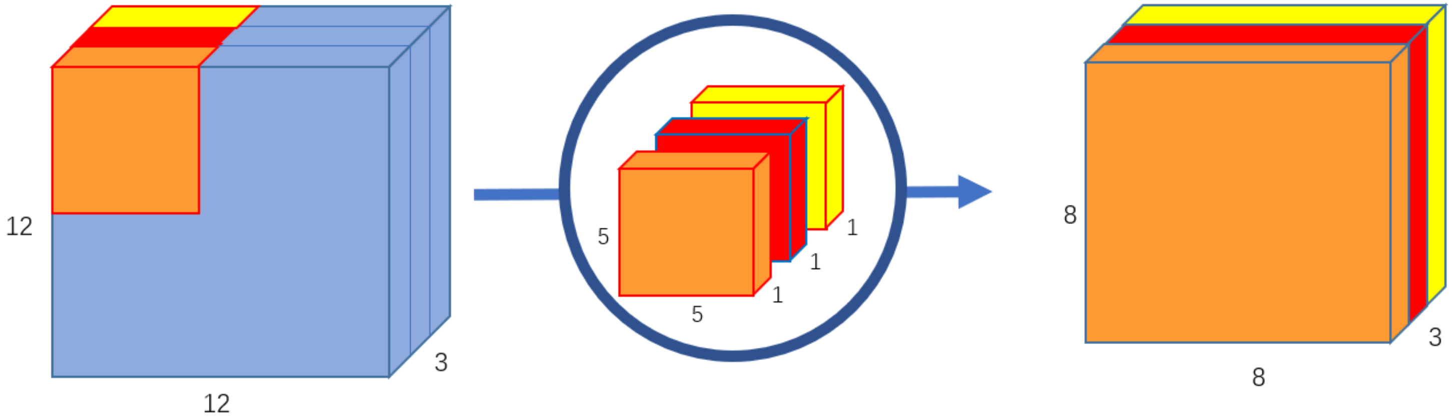


MobileNet

Depthwise Separable Convolution

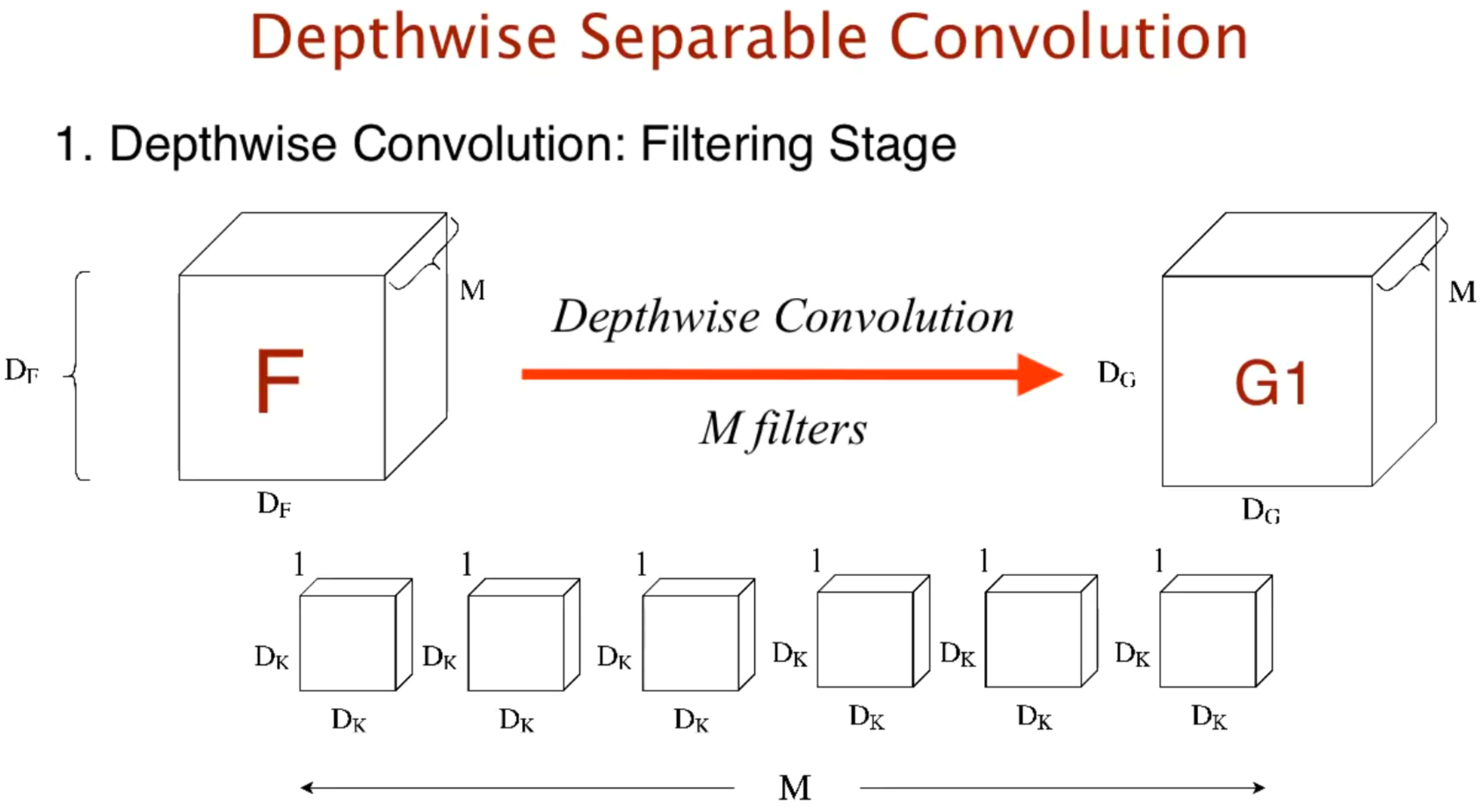
1. Depthwise Convolution: Filtering Stage

Example:



https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

MobileNet

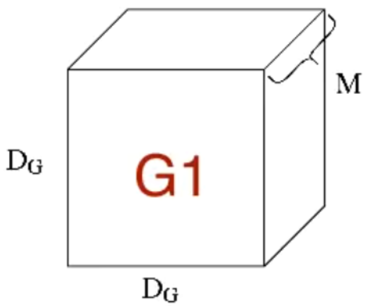


Source: https://www.youtube.com/watch?v=T7o3xvJLuHk

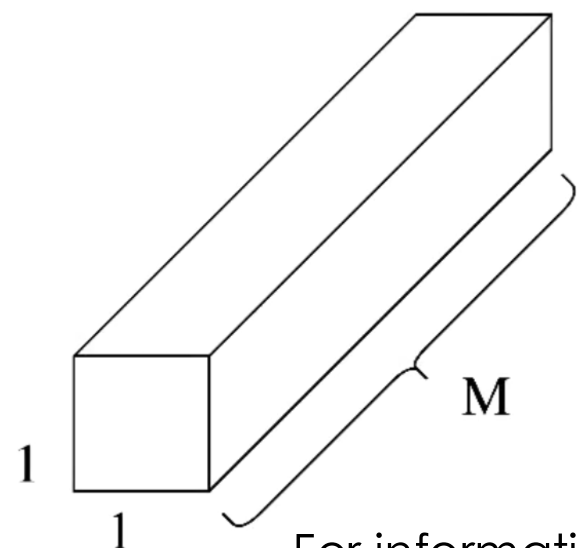
MobileNet

Depthwise Separable Convolution

2. Pointwise Convolution: Combination stage



Filter shape



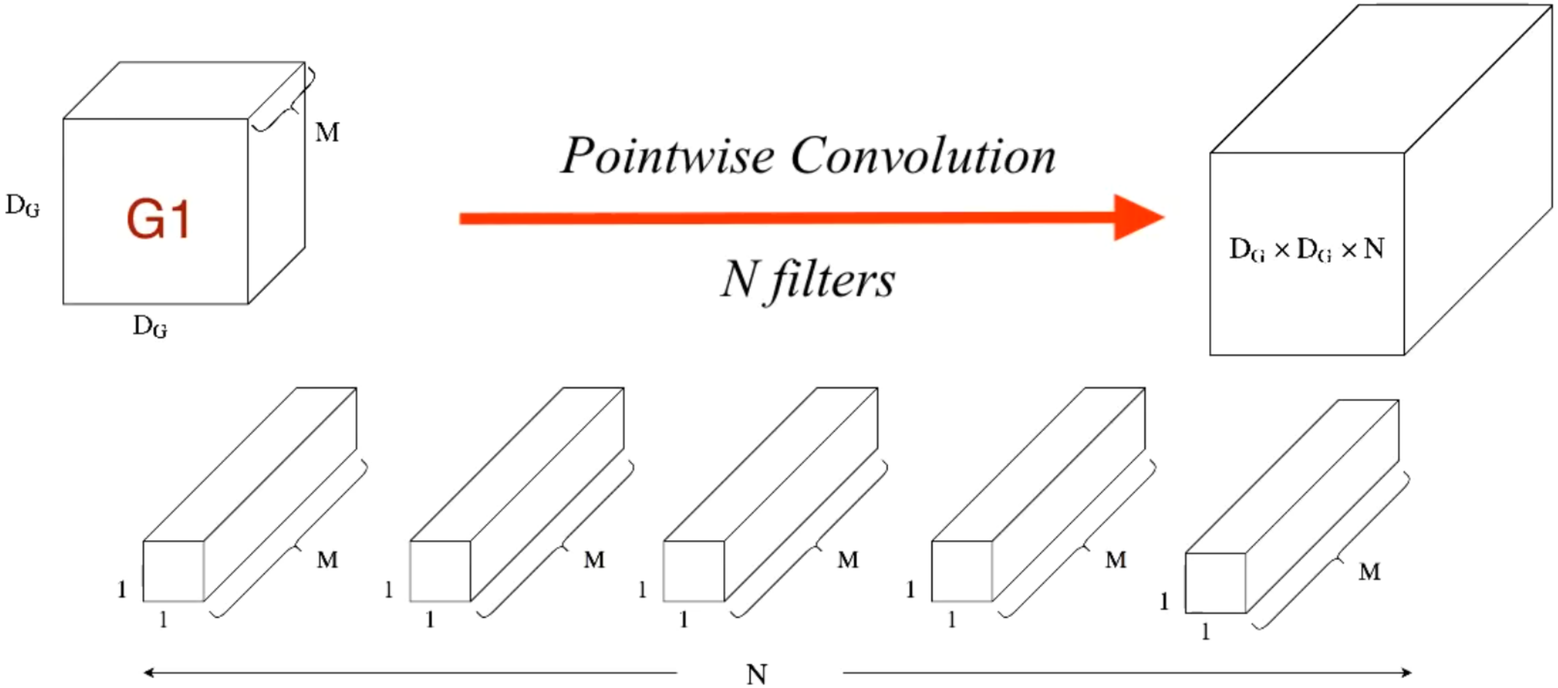
For information aggregation across channels

Source: https://www.youtube.com/watch?v=T7o3xvJLuHk

MobileNet

Depthwise Separable Convolution

2. Pointwise Convolution: Combination stage



Source: https://www.youtube.com/watch?v=T7o3xvJLuHk

MobileNet

Mults once = D_K^2

Mults 1 Channel $= D_G^2 \times D_K^2$

DC Mults = $M \times D_G^2 \times D_K^2$

Mults once = M

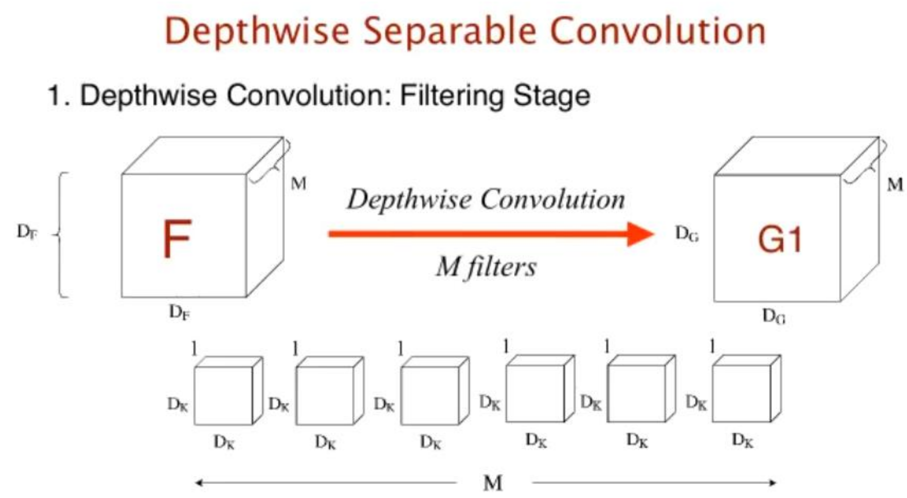
Mults 1 Kernel = $D_G \times D_G \times M$

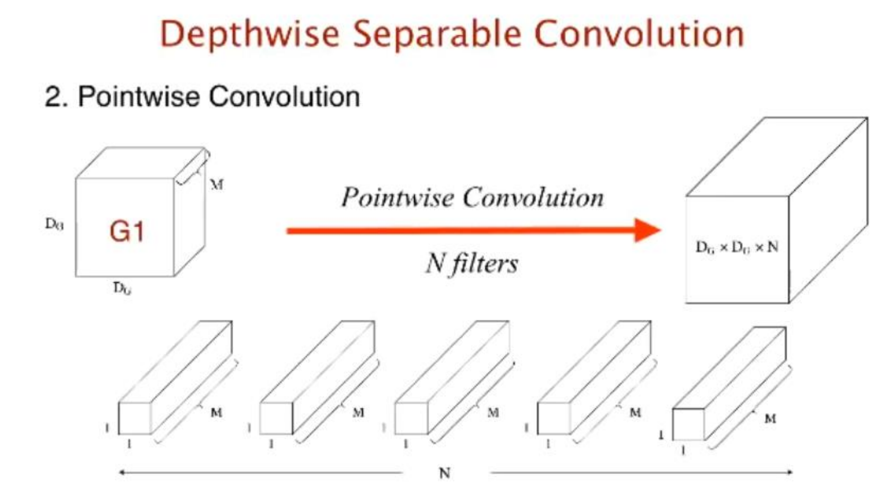
PC Mults = $N \times D_G \times D_G \times M$

Total = DC Mults + PC Mults

$$M \times D_G^2 \times D_K^2 + N \times D_G^2 \times M$$

$$M \times D_G^2 \,(D_K^2 + N)$$





MobileNet

Comparison Standard Vs. Depthwise

$$\frac{No.Mults\ in\ Depthwise\ Separable\ Conv}{No.Mults\ in\ Standard\ Conv} = \frac{M \times D_G^2\ (\ D_K^2 + N)}{N\ \times D_G \times D_G \times D_K \times D_K \times M}$$

$$\frac{No.\,Mults\ in\ Depthwise\ Separable\ Conv}{No.\,Mults\ in\ Standard\ Conv} = \frac{D_K^2 + N}{(\ D_K^2 \times N)} = \frac{1}{N} + \frac{1}{D_K^2}$$

E.g. $N = 1{,}024 \qquad D_K = 3$

$$\frac{No.\,Mults\ in\ Depthwise\ Separable\ Conv}{No.\,Mults\ in\ Standard\ Conv} = \frac{1}{1024} + \frac{1}{3^2} = 0.112$$

MobileNet

Depthwise Separable Convolution

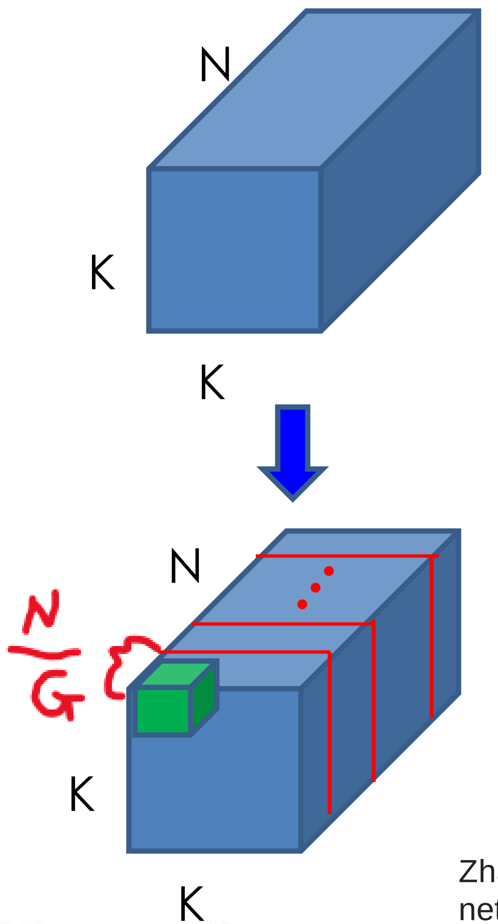
Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

Efficient Network Architectures

- ShuffleNet
  - Group convolution



M filters/kernels are also divided into G groups

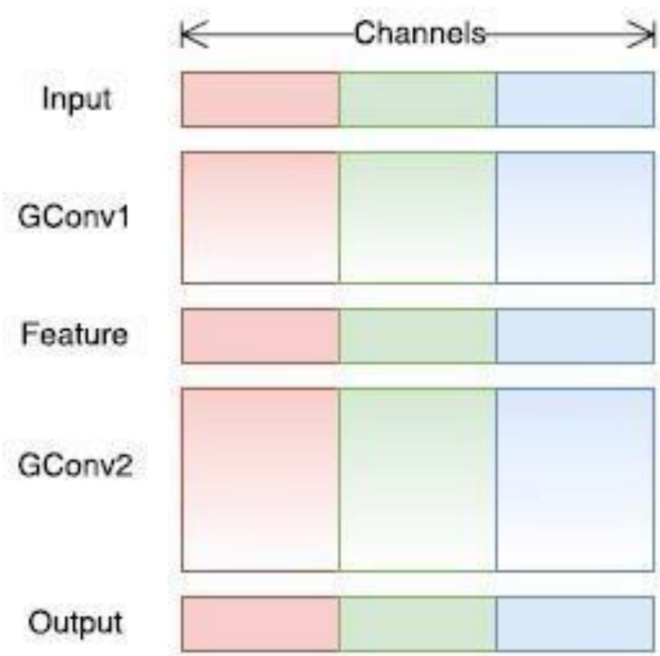
Each group has M/G filters

In each group, the filter has size: m x m x (N/G)

Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Efficient Network Architectures

- Group convolution



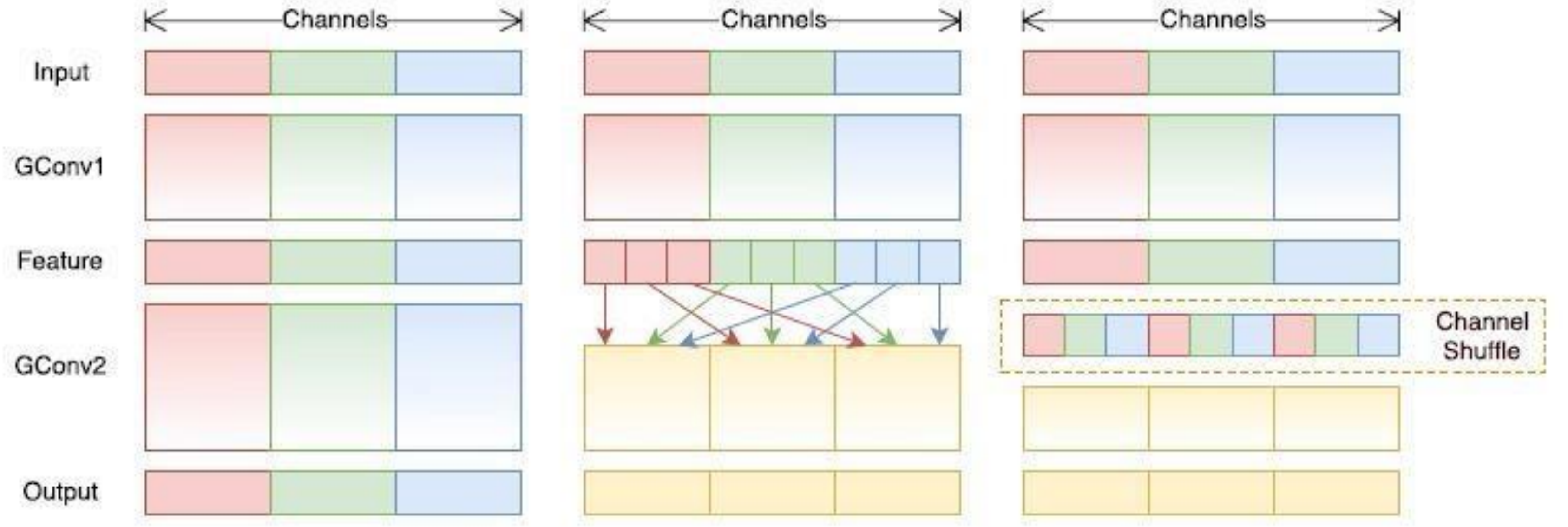
If multiple group convolutions stack together, there is one side effect!

Outputs from a certain group only relate to the inputs within the group.

No information exchange across groups.

Efficient Network Architectures

- Shuffled Group convolution



Zhang, Xiangyu, et al. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

Efficient Network Architectures

- GhostNet

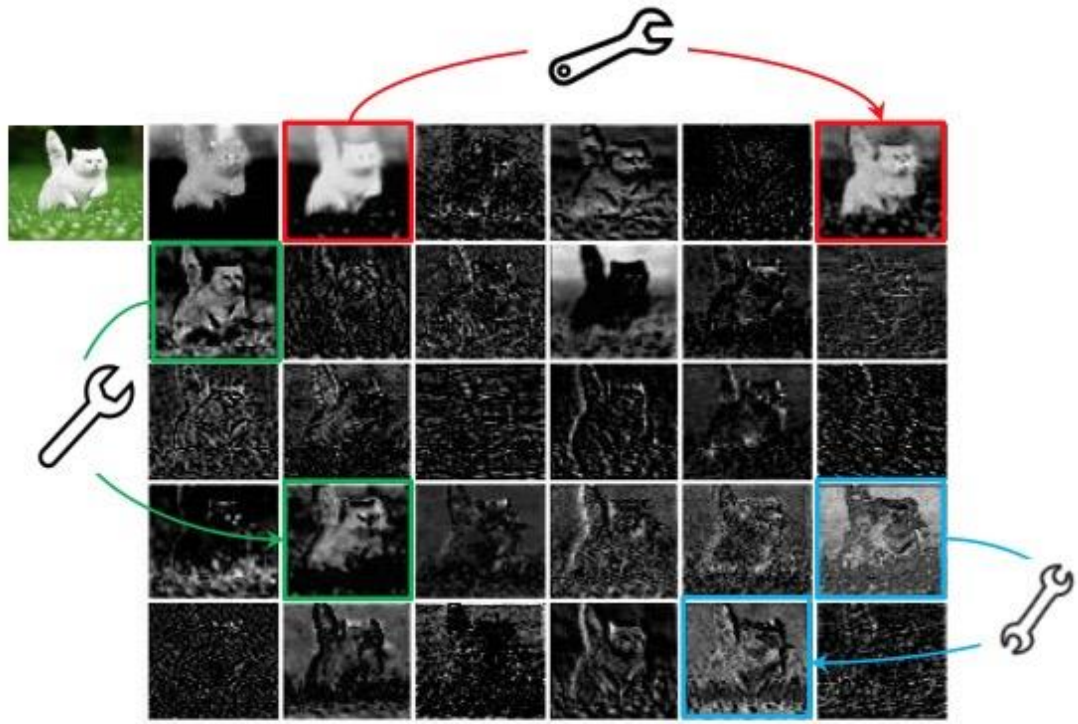


Figure 1. Visualization of some feature maps generated by the first residual group in ResNet-50, where three similar feature map pair examples are annotated with boxes of the same color. One feature map in the pair can be approximately obtained by transforming the other one through cheap operations (denoted by spanners).

Han, Kai, et al. "Ghostnet: More features from cheap operations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Efficient Network Architectures

- GhostNet

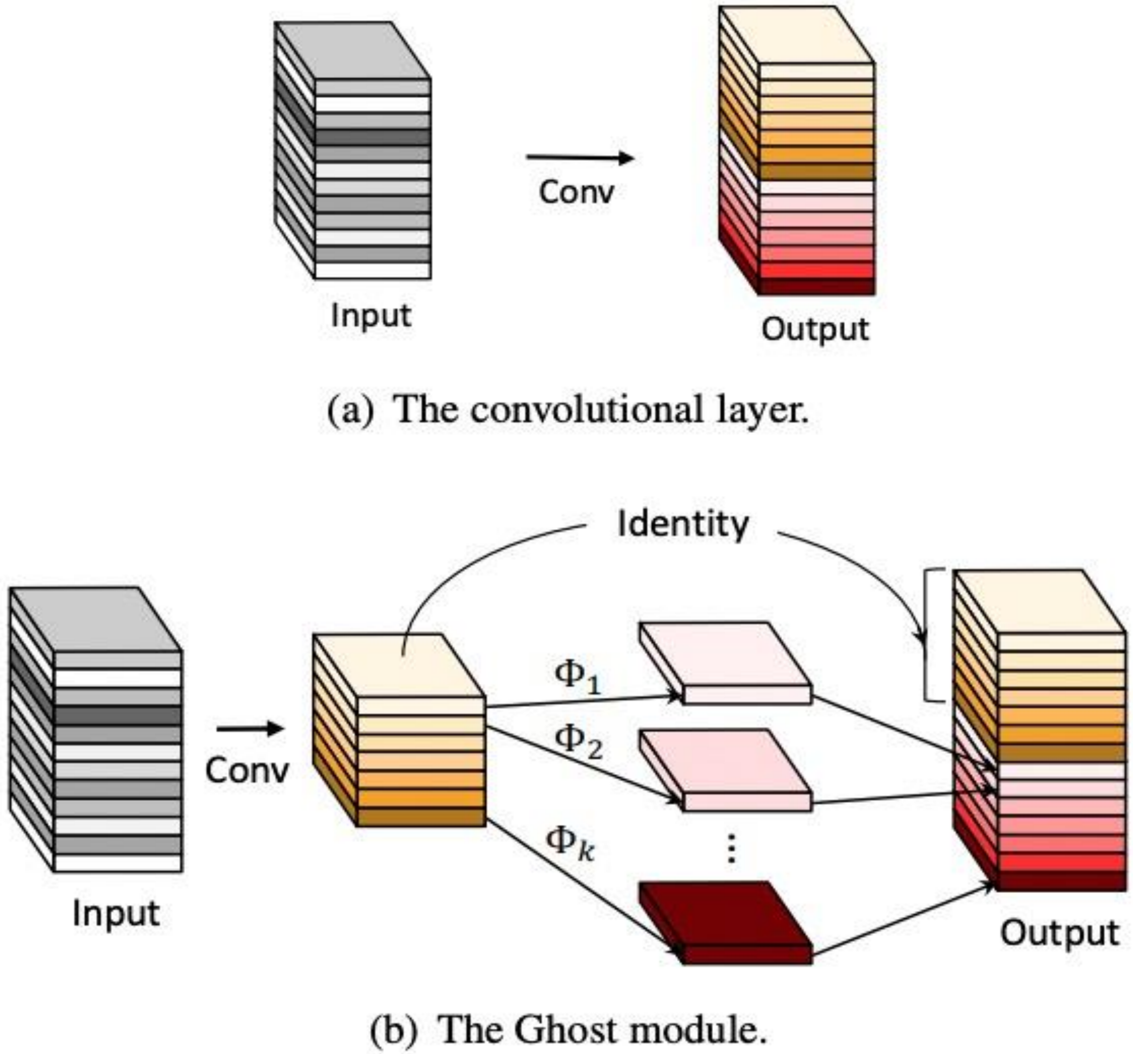


Figure 2. An illustration of the convolutional layer and the proposed Ghost module for outputting the same number of feature maps. Φ represents the cheap operation.

Han, Kai, et al. "Ghostnet: More features from cheap operations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Efficient Network Architectures

- GhostNet

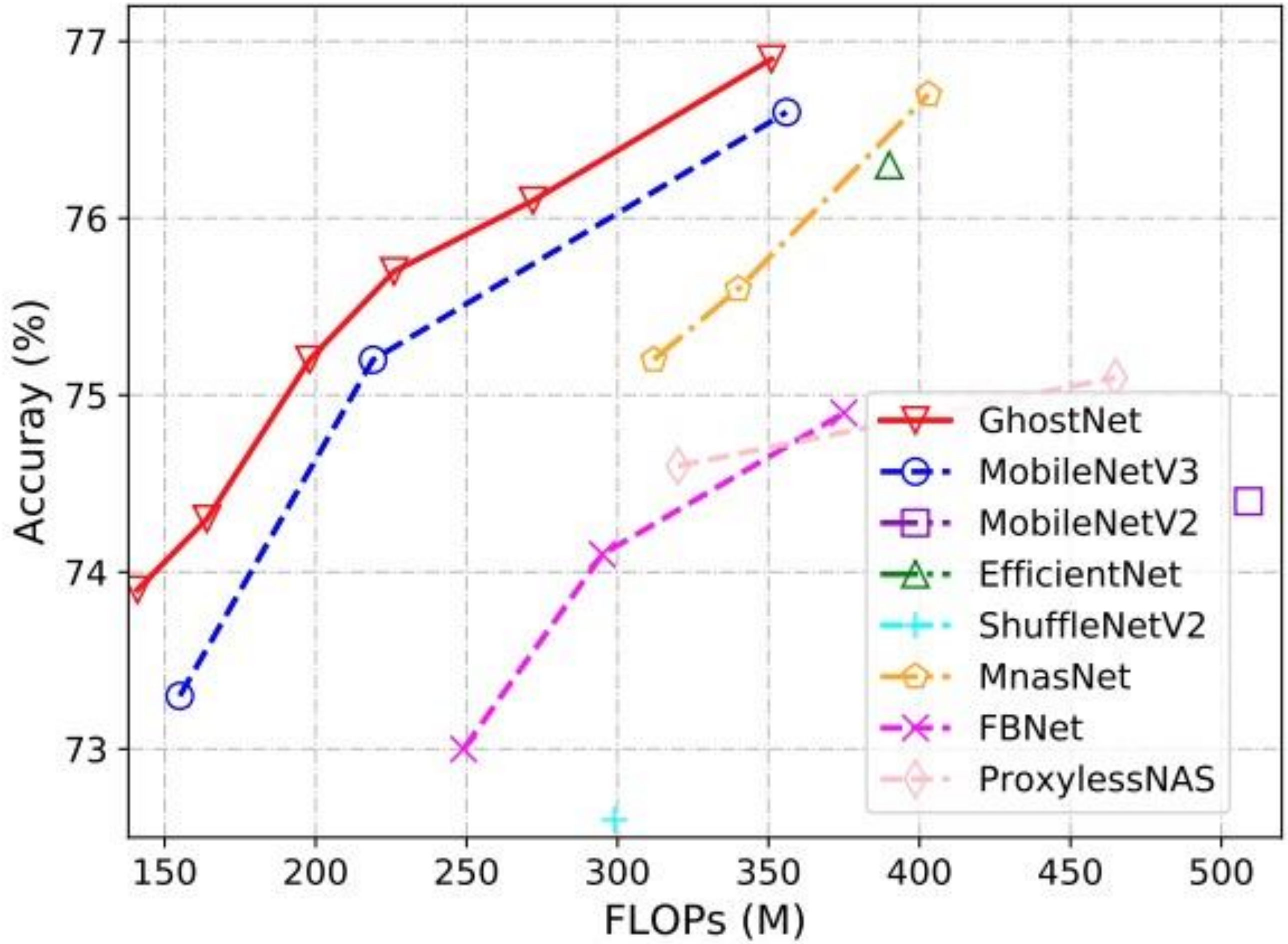


Figure 6. Top-1 accuracy *v.s.* FLOPs on ImageNet dataset.

Han, Kai, et al. "Ghostnet: More features from cheap operations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Efficient Transformers

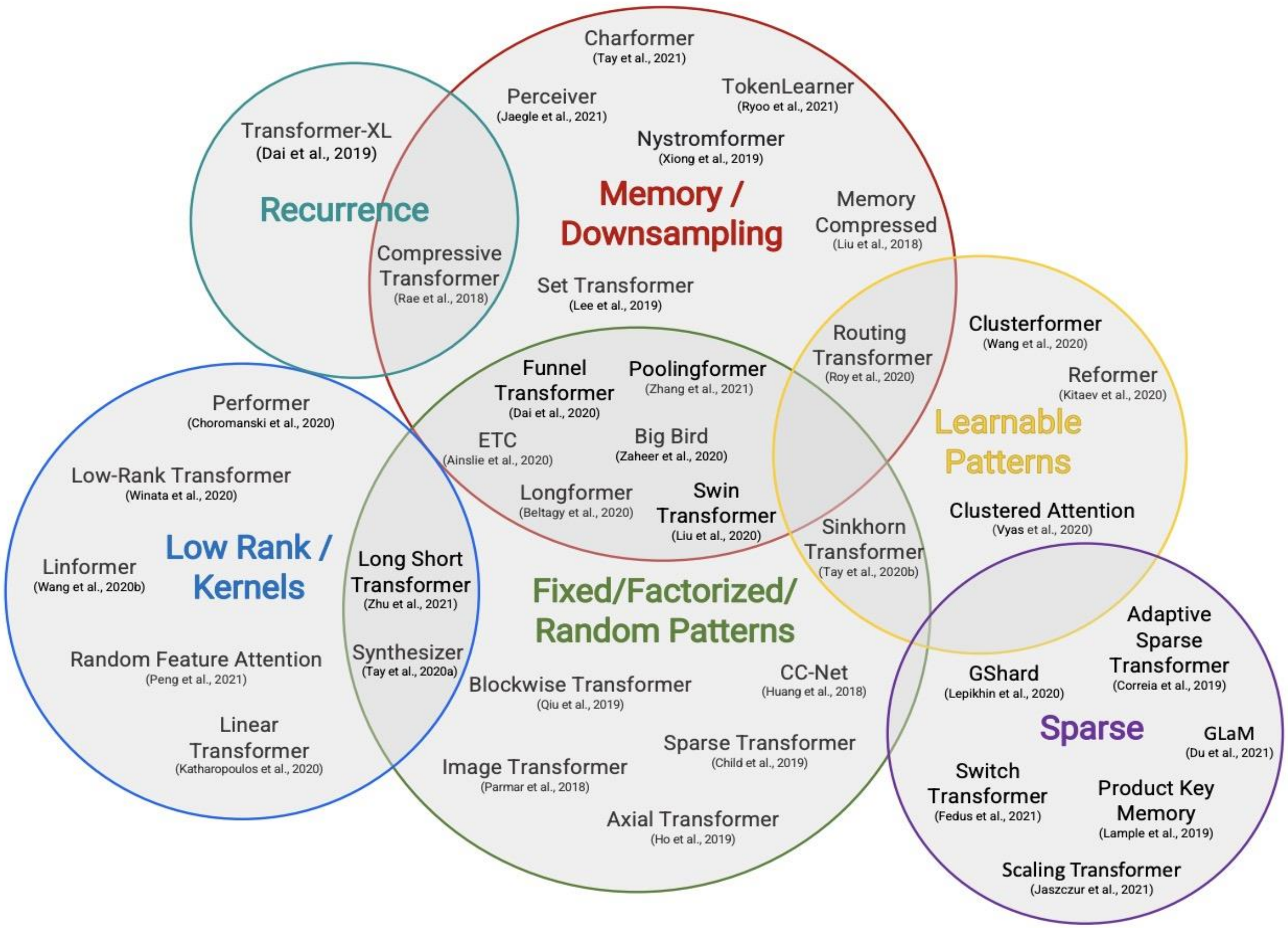


Figure 2: Taxonomy of Efficient Transformer Architectures.

Efficient Transformers: A Survey
https://arxiv.org/pdf/2009.06732.pdf

Efficient Transformers

January 30, 2024

PyTorch 2.2: FlashAttention-v2 integration, AOTInductor

by Team PyTorch

We are excited to announce the release of PyTorch® 2.2 (release note)! PyTorch 2.2 offers ~2x performance improvements to *scaled_dot_product_attention* via FlashAttention-v2 integration, as well as *AOTInductor*, a new ahead-of-time compilation and deployment tool built for non-python server-side deployments.

https://pytorch.org/blog/pytorch2-2/

EfficientViT-SAM

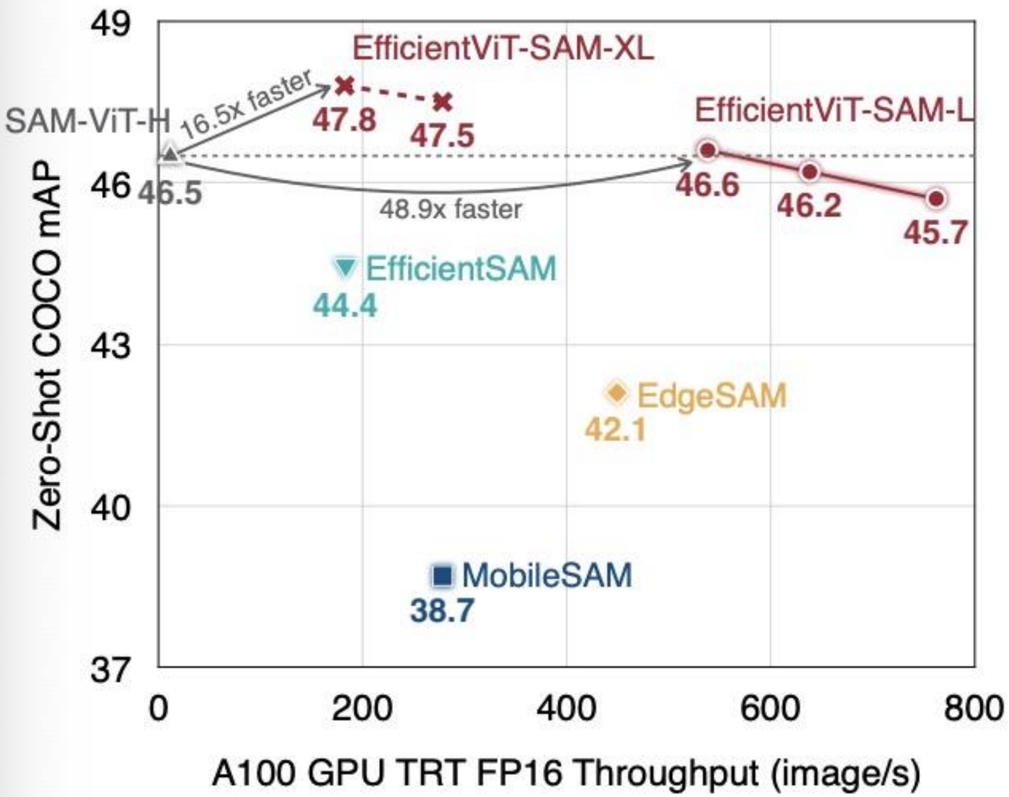


Figure 1. **Throughput vs. COCO Zero-Shot Instance Segmentation mAP.** EfficientViT-SAM is the first accelerated SAM model that matches/outperforms SAM-ViT-H's [1] zero-shot performance, delivering the SOTA performance-efficiency trade-off.

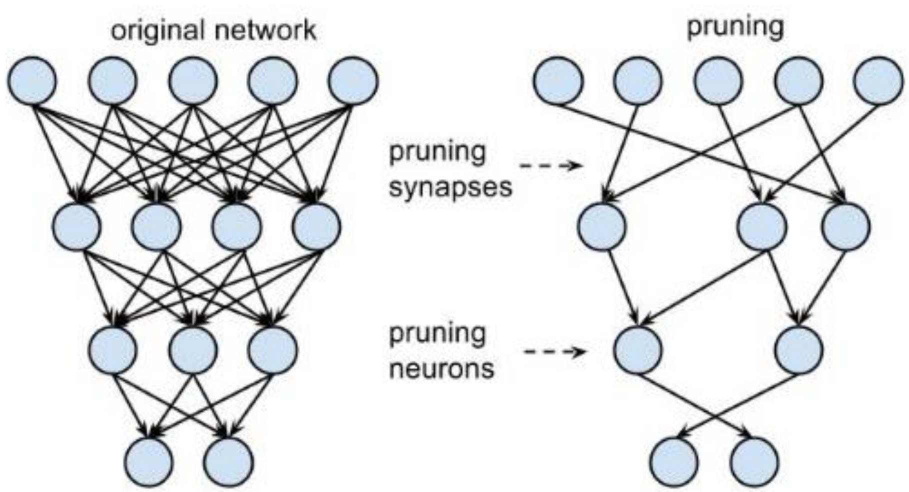
EfficientViT-SAM: Accelerated Segment Anything Model Without Performance Loss: https://arxiv.org/pdf/2402.05008.pdf

EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction https://arxiv.org/pdf/2205.14756.pdf

EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention https://arxiv.org/pdf/2305.07027.pdf

Network Pruning

- Remove weights/synapses “close to zero”
- **Retrain** to maintain accuracy
- Repeat



Sparse Network

Credit: Song Han

Network Pruning

- **Unstructured Pruning** methods prune individual parameters.
- Doing so produces a sparse neural network which, although smaller in terms of parameter count, may not be arranged in a fashion conducive to speed enhancements using modern libraries and hardware.
- This is also called **Weight Pruning** as we set individual weights in the weight matrix to zero.

https://blog.paperspace.com/neural-network-pruning-explained/

Network Pruning

- **Structured Pruning** methods consider parameters in groups, removing entire neurons, filters, or channels to exploit hardware and software optimized for dense computation.
- This is also called **Unit/Neuron Pruning**, as we set entire columns in the weight matrix to zero, in effect deleting the corresponding output neuron.

https://blog.paperspace.com/neural-network-pruning-explained/

Network Pruning

(a) Test Errors on CIFAR-10

Model	Test error (%)	Parameters	Pruned	FLOPs	Pruned
VGGNet (Baseline)	6.34	20.04M	-	7.97×10^8	-
VGGNet (70% Pruned)	**6.20**	2.30M	88.5%	3.91×10^8	51.0%
DenseNet-40 (Baseline)	6.11	1.02M	-	5.33×10^8	-
DenseNet-40 (40% Pruned)	**5.19**	0.66M	35.7%	3.81×10^8	28.4%
DenseNet-40 (70% Pruned)	5.65	0.35M	65.2%	2.40×10^8	55.0%
ResNet-164 (Baseline)	5.42	1.70M	-	4.99×10^8	-
ResNet-164 (40% Pruned)	**5.08**	1.44M	14.9%	3.81×10^8	23.7%
ResNet-164 (60% Pruned)	5.27	1.10M	35.2%	2.75×10^8	44.9%

(b) Test Errors on CIFAR-100

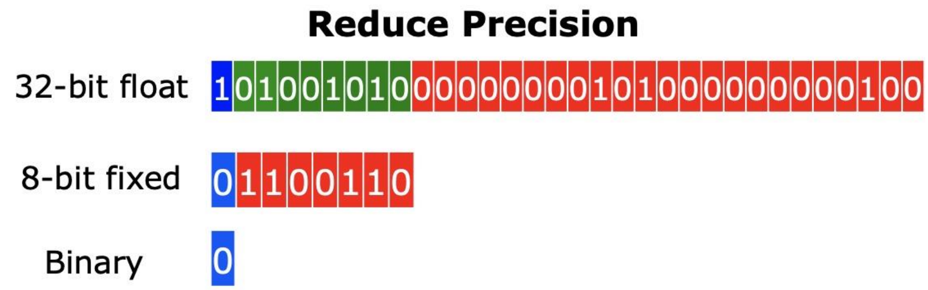
Model	Test error (%)	Parameters	Pruned	FLOPs	Pruned
VGGNet (Baseline)	26.74	20.08M	-	7.97×10^8	-
VGGNet (50% Pruned)	**26.52**	5.00M	75.1%	5.01×10^8	37.1%
DenseNet-40 (Baseline)	25.36	1.06M	-	5.33×10^8	-
DenseNet-40 (40% Pruned)	**25.28**	0.66M	37.5%	3.71×10^8	30.3%
DenseNet-40 (60% Pruned)	25.72	0.46M	54.6%	2.81×10^8	47.1%
ResNet-164 (Baseline)	23.37	1.73M	-	5.00×10^8	-
ResNet-164 (40% Pruned)	**22.87**	1.46M	15.5%	3.33×10^8	33.3%
ResNet-164 (60% Pruned)	23.91	1.21M	29.7%	2.47×10^8	50.6%

Liu, Zhuang, et al. "Learning efficient convolutional networks through network slimming." *Proceedings of the IEEE international conference on computer vision*. 2017.

WHAT IS THE STATE OF NEURAL NETWORK PRUNING
https://arxiv.org/pdf/2003.03033.pdf

Network Quantization

- Quantization for deep learning is the process of approximating a neural network that uses floating-point numbers by a neural network of low bit width numbers.
- Network quantization dramatically reduces both the memory requirement and computational cost of using neural networks.
- We assume that we have the trained model parameters θ, stored in floating point precision. In quantization, the goal is to reduce the precision of both the parameters (θ), as well as the intermediate activation maps to low-precision, with minimal impact on the generalization power/accuracy of the model.

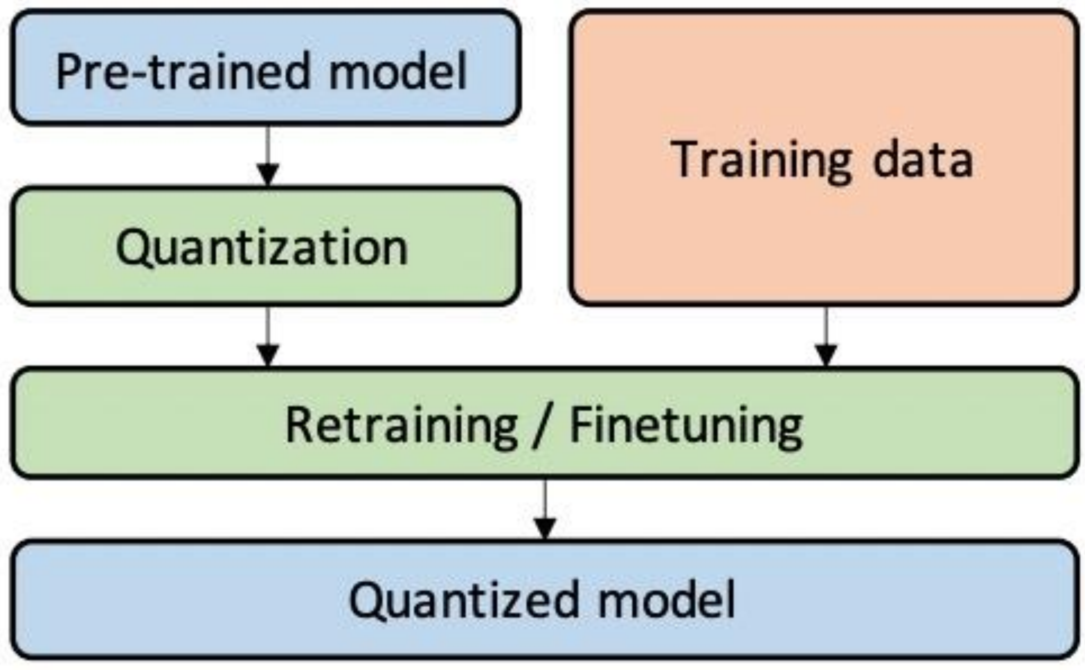


Network Quantization

- Quantization methods can be roughly divided into two categories:
  - quantization aware training (QAT)
  - post-training quantization (PTQ)

Network Quantization

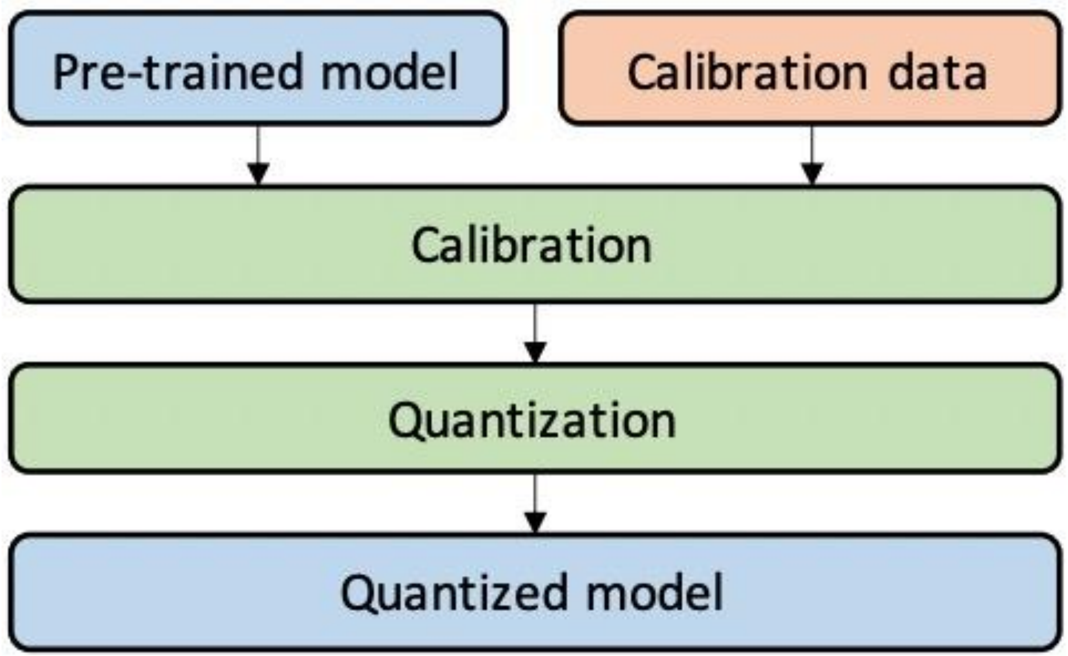
- Quantization aware training (QAT)
  - In QAT, a pre-trained model is quantized and then finetuned using training data to adjust parameters and recover accuracy degradation



Gholami, Amir, et al. "A survey of quantization methods for efficient neural network inference." arXiv preprint arXiv:2103.13630 (2021).

Network Quantization

- Post-Training Quantization (PTQ)
  - In PTQ, a pre-trained model is calibrated using calibration data (e.g., a small subset of training data) to compute the clipping ranges and the scaling factors. Then, the model is quantized based on the calibration result.



Gholami, Amir, et al. "A survey of quantization methods for efficient neural network inference." arXiv preprint arXiv:2103.13630 (2021).

Network Quantization

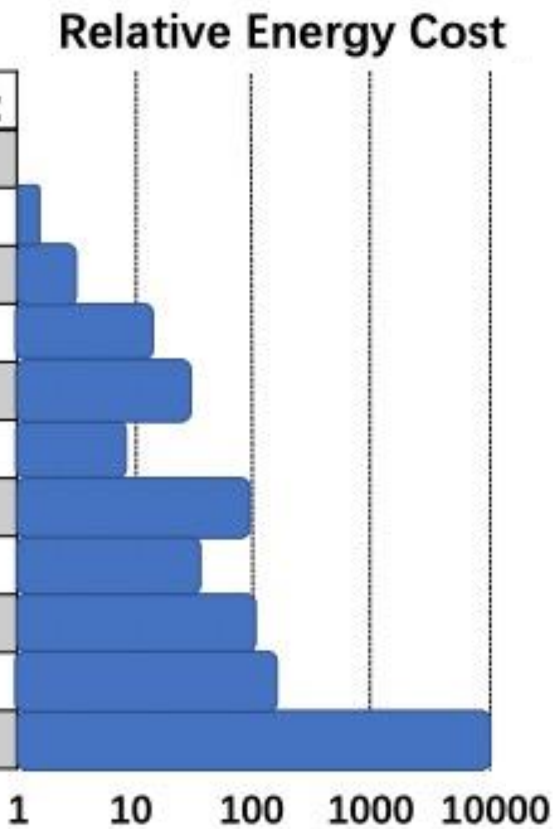
- Quantization methods can be roughly divided into two categories:
  – quantization aware training (QAT)
  – post-training quantization (PTQ)

- QAT methods usually achieve better results than PTQ methods. PTQ methods are simpler and add quantization to a given network model without any training process.

Network Quantization

- Lower precision provides exponentially better energy efficiency

Operation:	Energy(pJ):
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FP Mult	1.1
32b FP Mult	3.7
32b SRAM Read (8kb)	5.0
32b DRAM Read	640



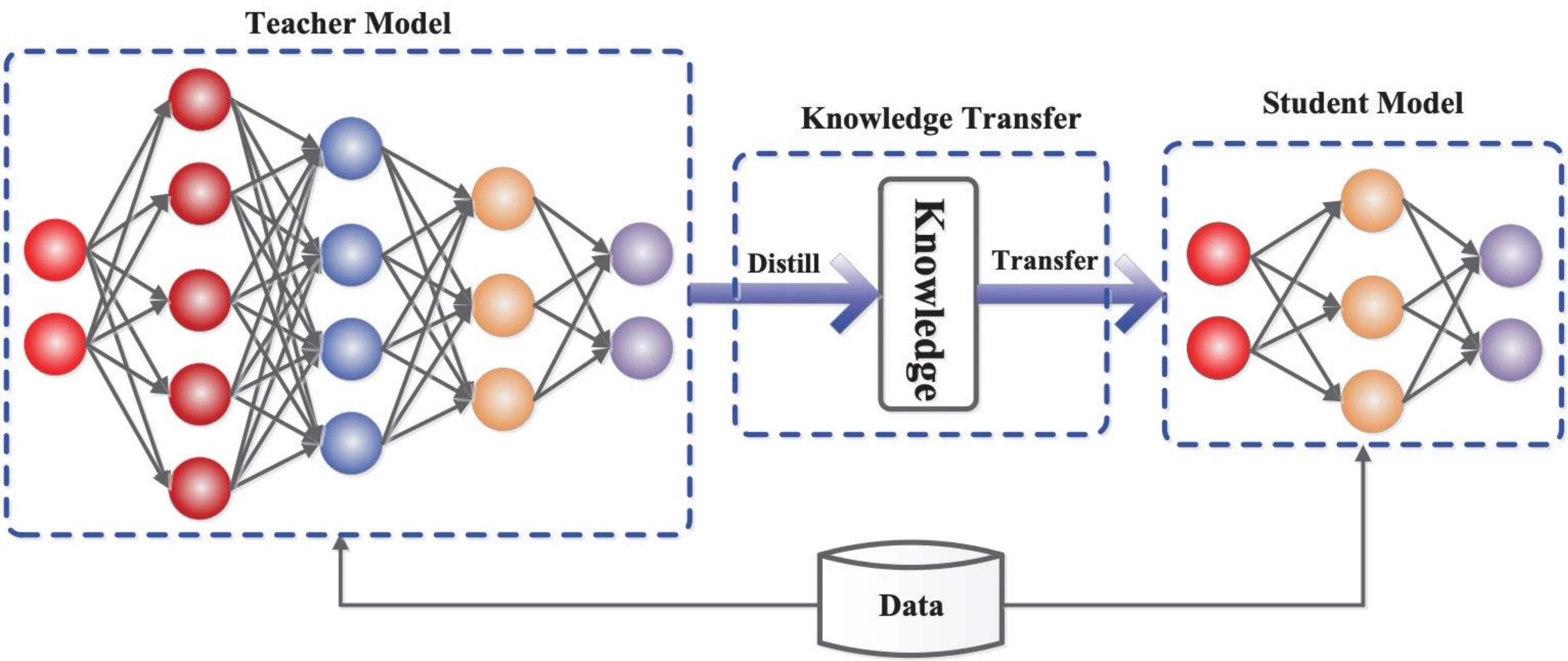
Comparison of the corresponding energy cost for different precision for 45nm technology.

Gholami, Amir, et al. "A survey of quantization methods for efficient neural network inference." arXiv preprint arXiv:2103.13630 (2021).

Knowledge Distillation

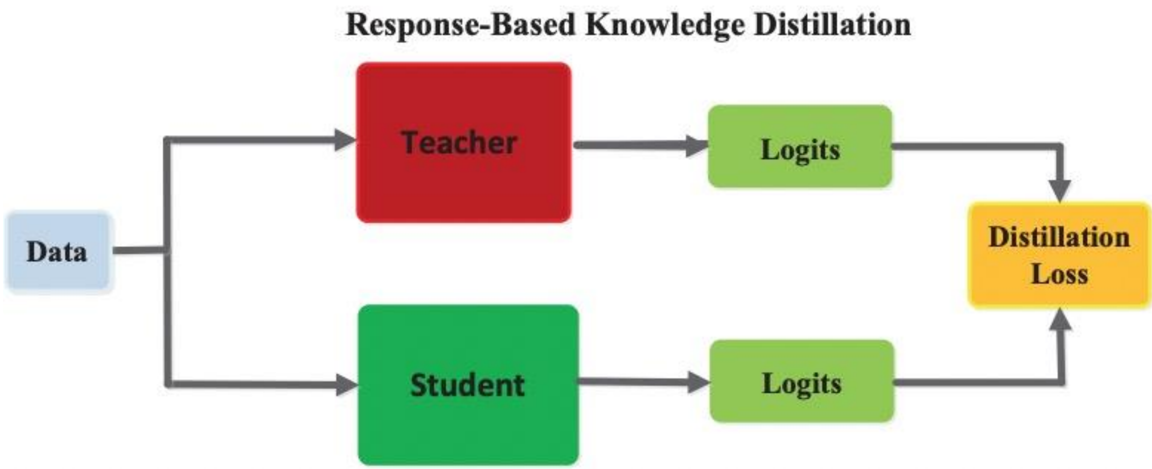
- Knowledge distillation is a process of distilling or transferring the knowledge from a (set of) large, cumbersome model(s) to a lighter, easier-to-deploy single model, without significant loss in performance.

Knowledge Distillation



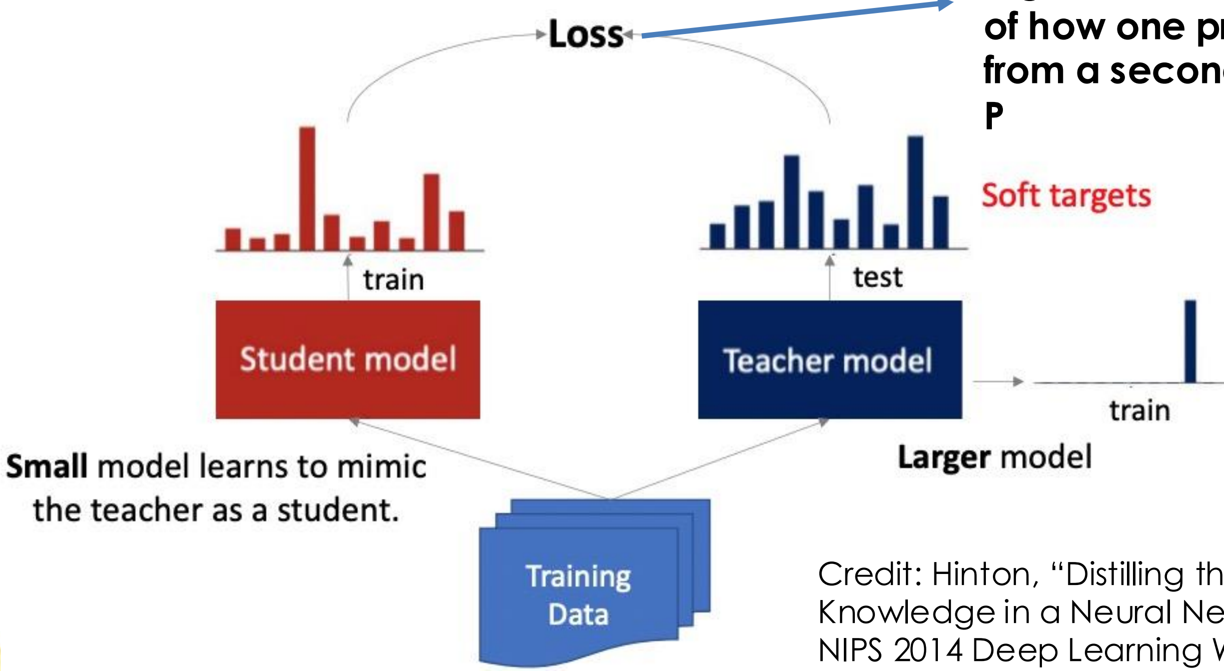
Gou, Jianping, et al. "Knowledge distillation: A survey." International Journal of Computer Vision 129.6 (2021): 1789-1819.

Knowledge Distillation



Gou, Jianping, et al. "Knowledge distillation: A survey." International Journal of Computer Vision 129.6 (2021): 1789-1819.

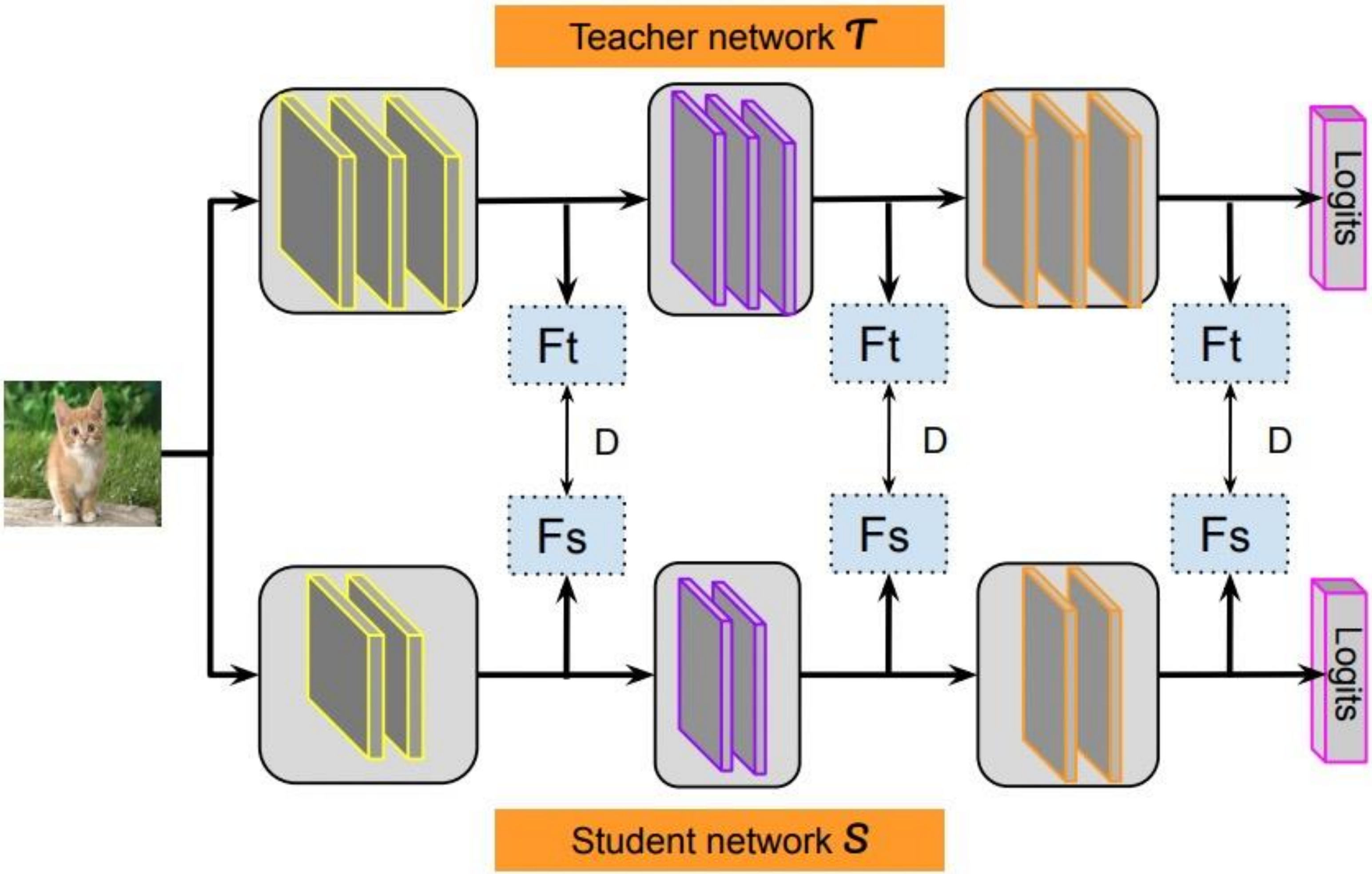
e.g., the Kullback–Leibler divergence: a measure of how one probability distribution Q is different from a second, reference probability distribution P



Credit: Hinton, "Distilling the Knowledge in a Neural Network" at NIPS 2014 Deep Learning Workshop

Knowledge Distillation

Feature-based knowledge distillation



Wang, Lin, and Kuk-Jin Yoon. "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

Knowledge Distillation

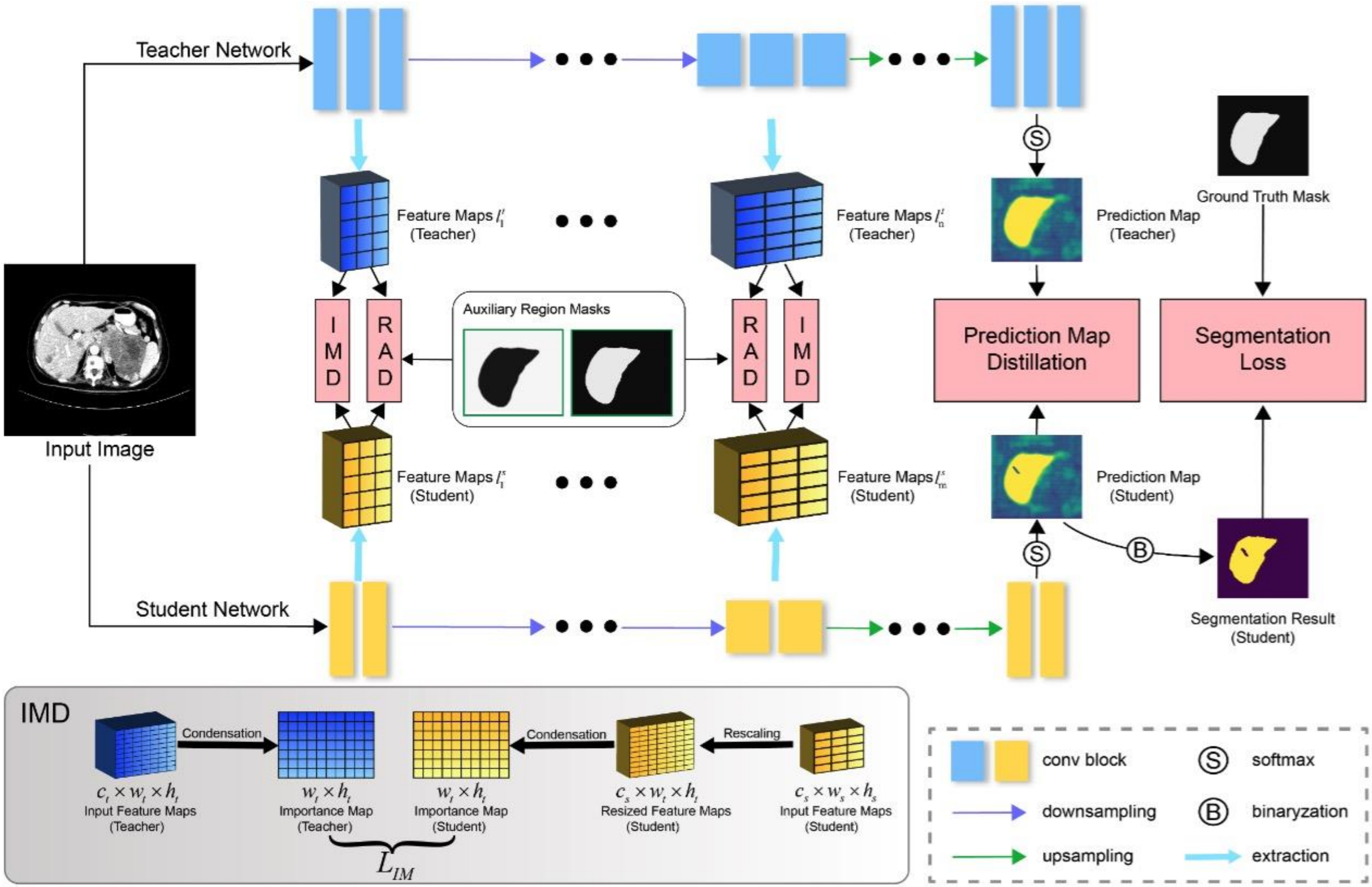
Table 5 Performance comparison of different knowledge distillation methods on CIFAR10. Note that ↑ indicates the performance improvement of the student network learned by each method comparing with the corresponding baseline model.

Offline Distillation				
Methods	Knowledge	Teacher (baseline)	Student (baseline)	Accuracies
FSP (Yim et al., 2017)	RelK	ResNet26 (91.91)	ResNet8 (87.91)	88.70 (0.79 ↑)
FT (Kim et al., 2018)	FeaK	ResNet56 (93.61)	ResNet20 (92.22)	93.15 (0.93 ↑)
IRG (Liu et al., 2019g)	RelK	ResNet20 (91.45)	ResNet20-x0.5 (88.36)	90.69 (2.33 ↑)
SP (Tung and Mori, 2019)	RelK	WRN-40-1 (93.49)	WRN-16-1 (91.26)	91.87 (0.61 ↑)
SP (Tung and Mori, 2019)	RelK	WRN-40-2 (95.76)	WRN-16-8 (94.82)	95.45 (0.63 ↑)
FN (Xu et al., 2020b)	FeaK	ResNet110 (94.29)	ResNet56 (93.63)	94.14 (0.51 ↑)
FN (Xu et al., 2020b)	FeaK	ResNet56 (93.63)	ResNet20 (92.11)	92.67 (0.56 ↑)
AdaIN (Yang et al., 2020a)	FeaK	ResNet26 (93.58)	ResNet8 (87.78)	89.02 (1.24 ↑)
AdaIN (Yang et al., 2020a)	FeaK	WRN-40-2 (95.07)	WRN-16-2 (93.98)	94.67 (0.69 ↑)
AE-KD (Du et al., 2020)	FeaK	ResNet56 (—)	MobileNetV2 (75.97)	77.07 (1.10 ↑)
JointRD (Li et al., 2020b)	FeaK	ResNet34 (95.39)	plain-CNN 34 (93.73)	94.78 (1.05 ↑)
TOFD (Zhang et al., 2020a)	FeaK	ResNet152 (—)	ResNeXt50-4 (94.49)	97.09 (2.60 ↑)
TOFD (Zhang et al., 2020a)	FeaK	ResNet152 (—)	MobileNetV2 (90.43)	93.34 (2.91 ↑)
CTKD (Zhao et al., 2020a)	RelK, FeaK	WRN-40-1 (93.43)	WRN-16-1 (91.28)	92.50 (1.22 ↑)
CTKD (Zhao et al., 2020a)	RelK, FeaK	WRN-40-2 (94.70)	WRN-16-2 (93.68)	94.42 (0.74 ↑)

Gou, Jianping, et al. "Knowledge distillation: A survey." International Journal of Computer Vision 129.6 (2021): 1789-1819.

Case Study (Medical Imaging)

- Efficient Medical Image Segmentation Based on Knowledge Distillation



Qin, Dian, et al. "Efficient medical image segmentation based on knowledge distillation." *IEEE Transactions on Medical Imaging* 40.12 (2021): 3820-3831.

Case Study (Medical Imaging)

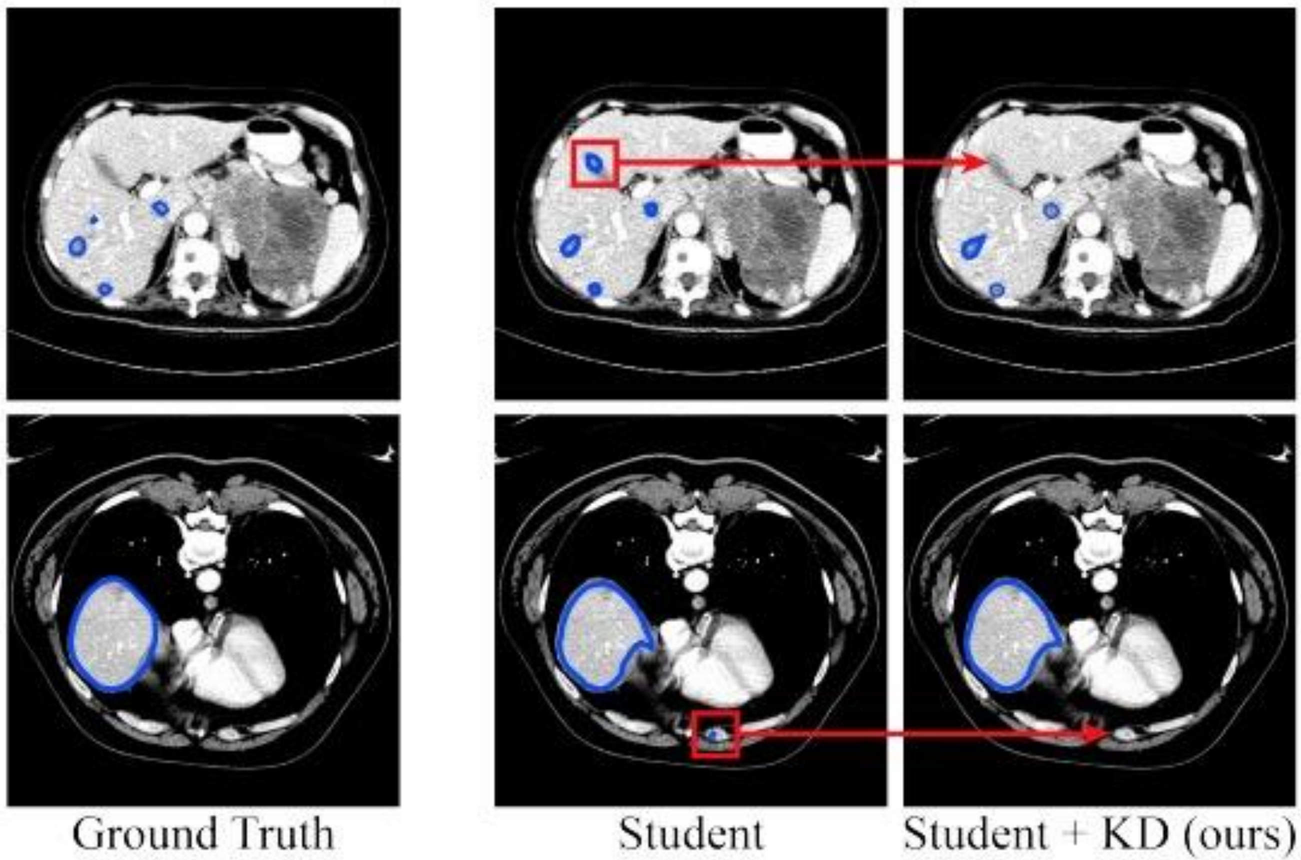
- Efficient Medical Image Segmentation Based on Knowledge Distillation

Method	Liver Tumor Dice	Liver Dice	Kidney Tumor Dice	Kidney Dice	#Params (M)
Teachers					
T1: RA-UNet	0.685 ± 0.004	0.960 ± 0.001	0.745 ± 0.003	0.970 ± 0.001	22.1
T2: PSPNet	0.640 ± 0.005	0.959 ± 0.001	0.659 ± 0.007	0.968 ± 0.002	46.7
T3: UNet++	0.669 ± 0.003	0.949 ± 0.001	0.644 ± 0.007	0.943 ± 0.002	20.6
Students and their performances distilled from different teachers by our approach					
ENet	0.574 ± 0.005	0.952 ± 0.001	0.521 ± 0.015	0.939 ± 0.001	0.353
ENet + T1 (ours)	$\mathbf{0.652 \pm 0.005}$	$\mathbf{0.959 \pm 0.001}$	0.676 ± 0.007	0.965 ± 0.001	
ENet + T2 (ours)	0.635 ± 0.003	0.958 ± 0.001	0.599 ± 0.009	$\mathbf{0.967 \pm 0.001}$	
ENet + T3 (ours)	0.634 ± 0.004	0.953 ± 0.001	0.648 ± 0.008	0.941 ± 0.001	
MobileNetV2	0.540 ± 0.003	0.921 ± 0.002	0.516 ± 0.009	0.945 ± 0.001	2.2
MobileNetV2 + T1 (ours)	0.595 ± 0.004	0.932 ± 0.002	$\mathbf{0.684 \pm 0.006}$	0.952 ± 0.001	
MobileNetV2 + T2 (ours)	0.590 ± 0.006	0.927 ± 0.002	0.678 ± 0.003	0.949 ± 0.001	
MobileNetV2 + T3 (ours)	0.589 ± 0.002	0.924 ± 0.001	0.679 ± 0.005	n/a	
ResNet18	0.464 ± 0.008	0.934 ± 0.001	0.435 ± 0.005	0.933 ± 0.001	11.2
ResNet18 + T1 (ours)	0.508 ± 0.004	0.943 ± 0.001	0.582 ± 0.008	0.939 ± 0.001	
ResNet18 + T2 (ours)	0.491 ± 0.004	0.946 ± 0.001	0.551 ± 0.005	0.941 ± 0.001	
ResNet18 + T3 (ours)	0.508 ± 0.006	0.935 ± 0.001	0.450 ± 0.009	0.934 ± 0.001	

Qin, Dian, et al. "Efficient medical image segmentation based on knowledge distillation." *IEEE Transactions on Medical Imaging* 40.12 (2021): 3820-3831.

Case Study (Medical Imaging)

- Efficient Medical Image Segmentation Based on Knowledge Distillation



Liver segmentation

Qin, Dian, et al. "Efficient medical image segmentation based on knowledge distillation." *IEEE Transactions on Medical Imaging* 40.12 (2021): 3820-3831.

Thank you!

Question?