

# TinyML and Efficient Deep Learning Computing

6.S965 (Fall 2022)

**Song Han**

[songhan@mit.edu](mailto:songhan@mit.edu)



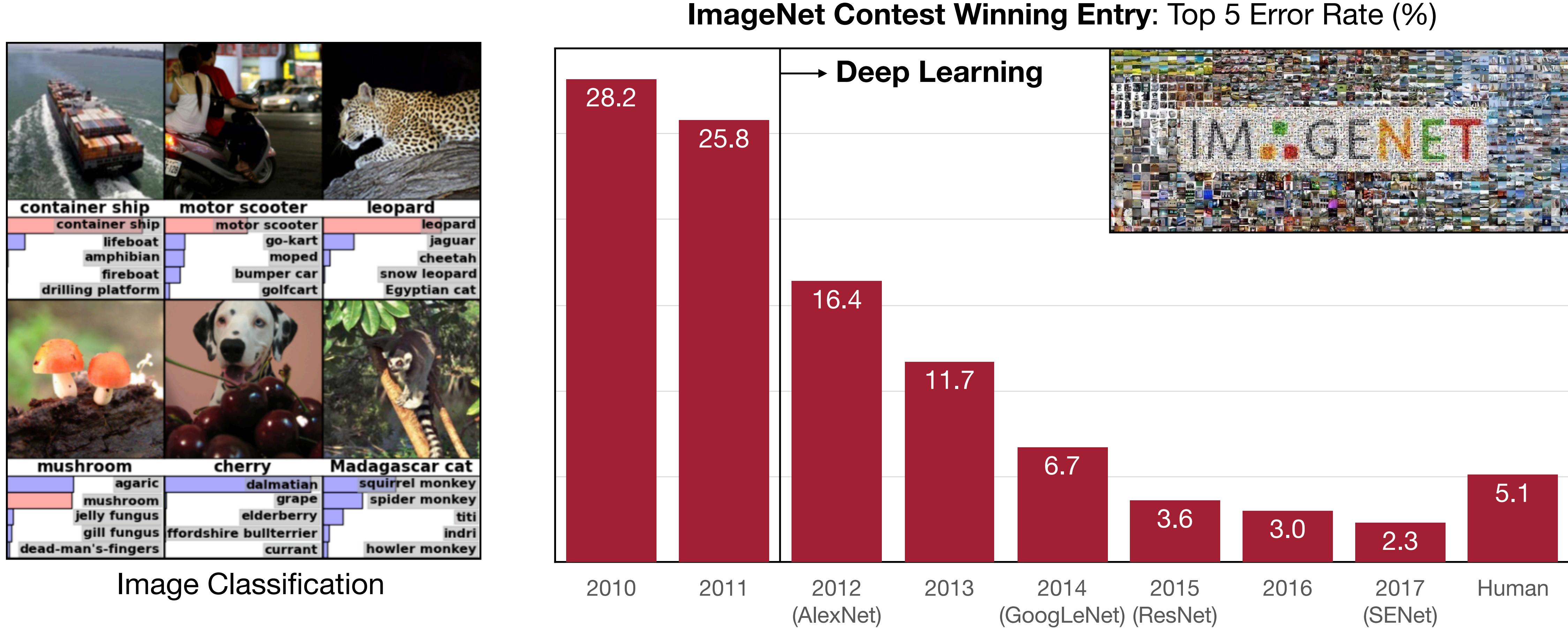
# Deep Learning is Everywhere



Image source: [1](#), [2](#), [3](#)

# Deep Learning for Image Classification

# DNNs achieve super-human classification accuracy on ImageNet



# Deep Learning for Games

AlphaGo masters the game of Go with DNNs & tree search



AlphaGo (Nature 2016)



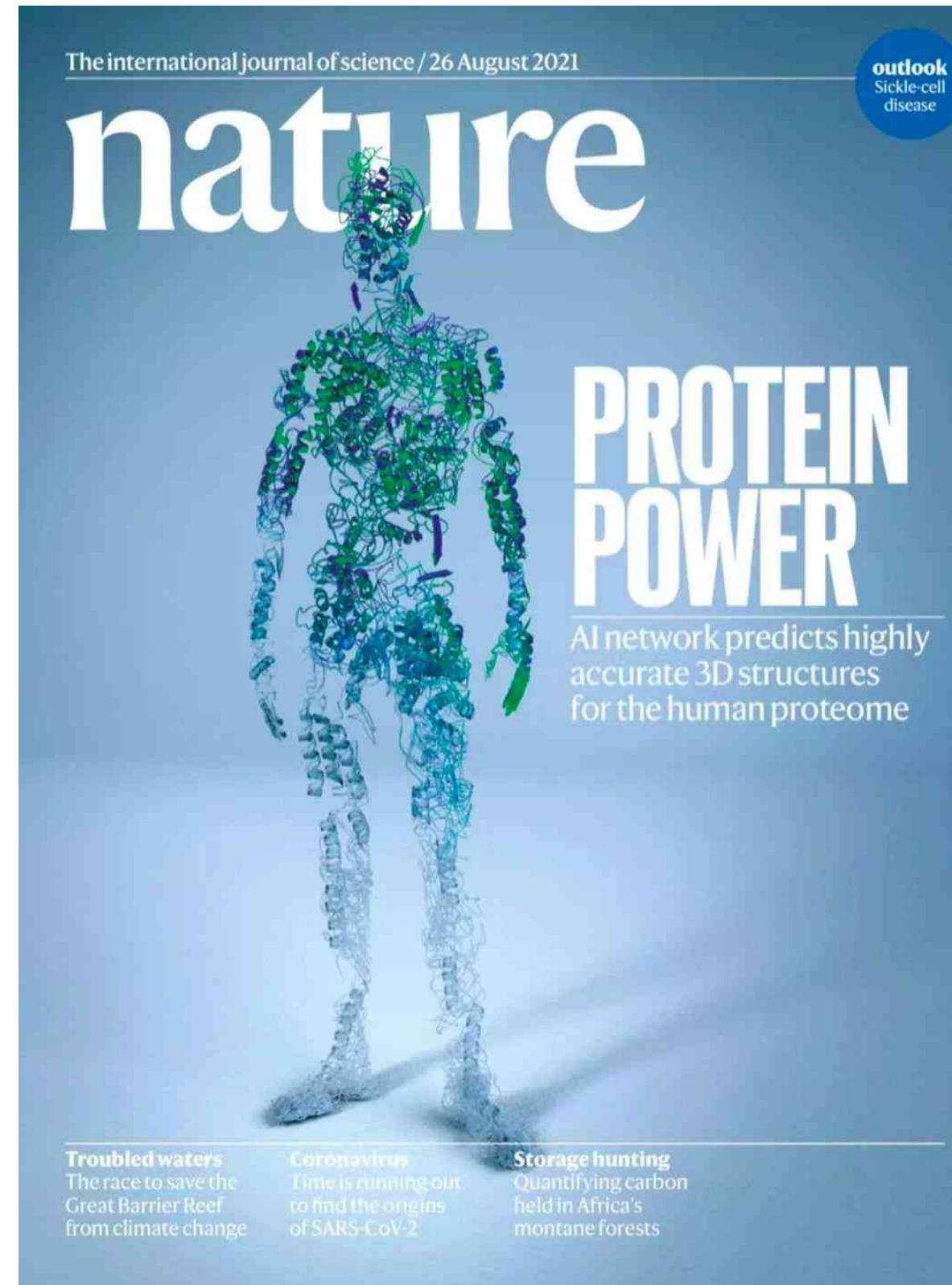
AlphaGo versus Lee Sedol (4-1)

**Compute: 1920 CPUs and 280 GPUs (\$3000 electric bill per game)**

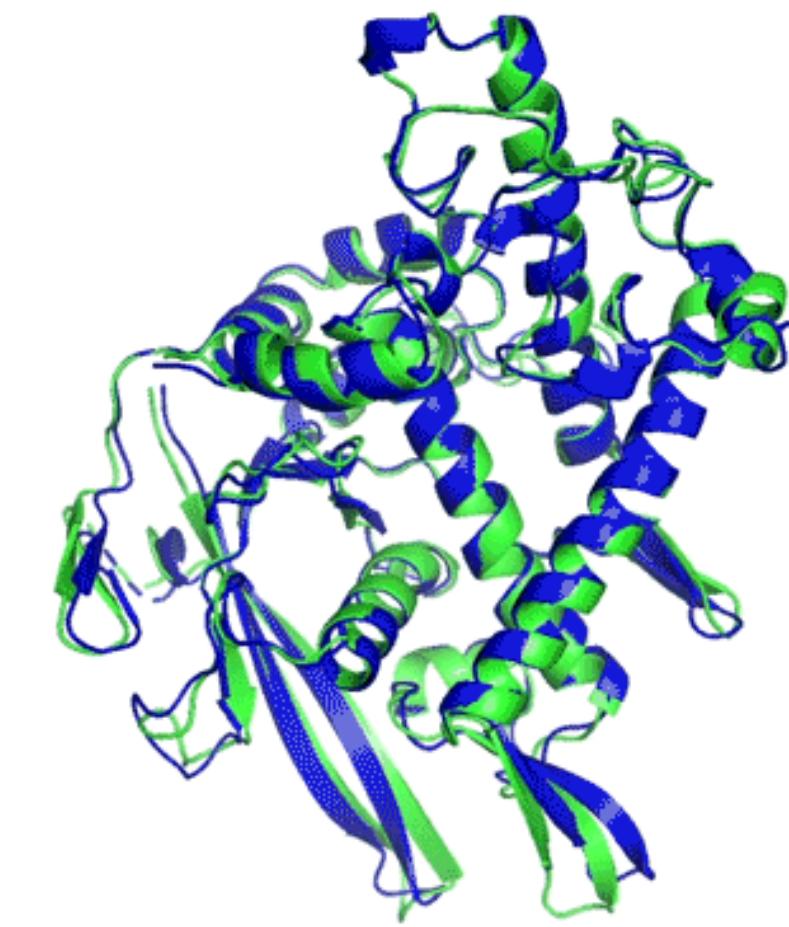
AlphaGo: <https://www.deepmind.com/research/highlighted-research/alphago>

# Deep Learning for Scientific Discovery

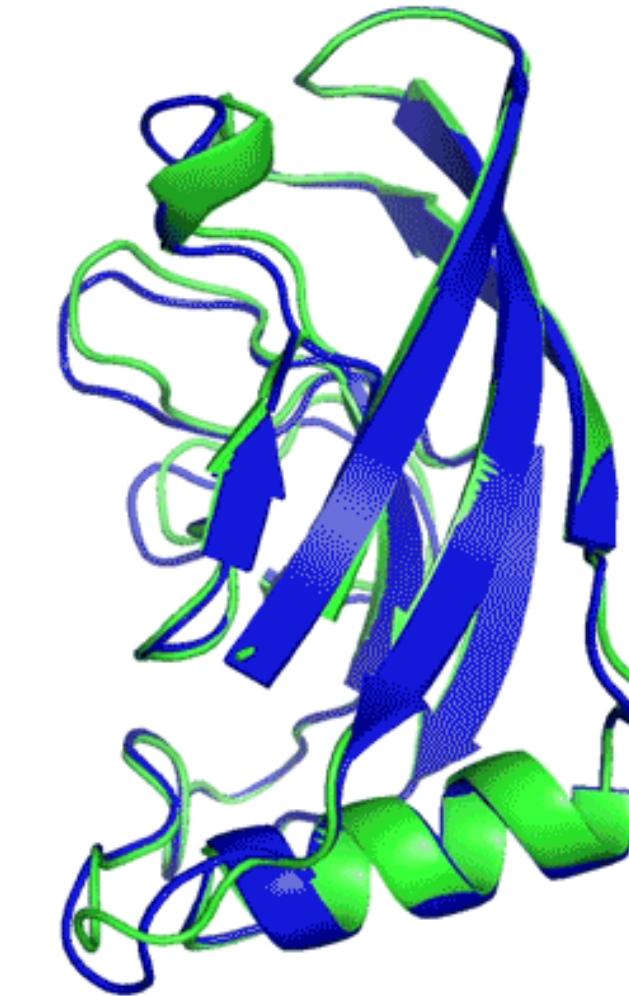
AlphaFold reveals the structure of the protein universe



AlphaFold (Nature 2021)



T1037 / 6vr4  
90.7 GDT  
(RNA polymerase domain)



T1049 / 6y4f  
93.3 GDT  
(adhesin tip)

- Experimental result
- Computational prediction

Compute: 16 TPUv3s (128 TPUv3 cores) for a few weeks

AlphaFold 2: <https://www.deepmind.com/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>

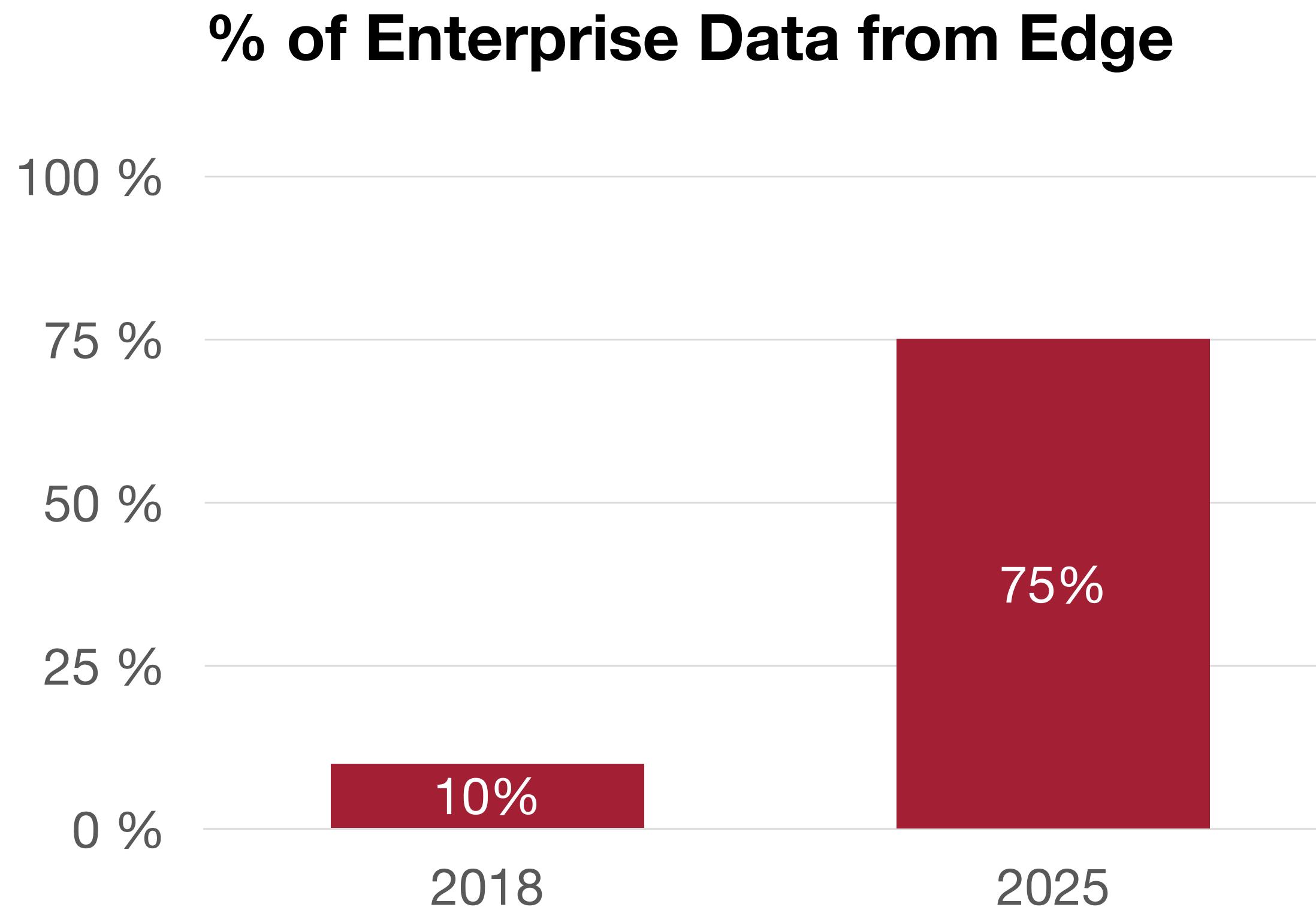
# AI is Quickly Coming to the Edge



Image source: [1](#), [2](#), [3](#)

# AI is Quickly Coming to the Edge

Privacy, latency, cost

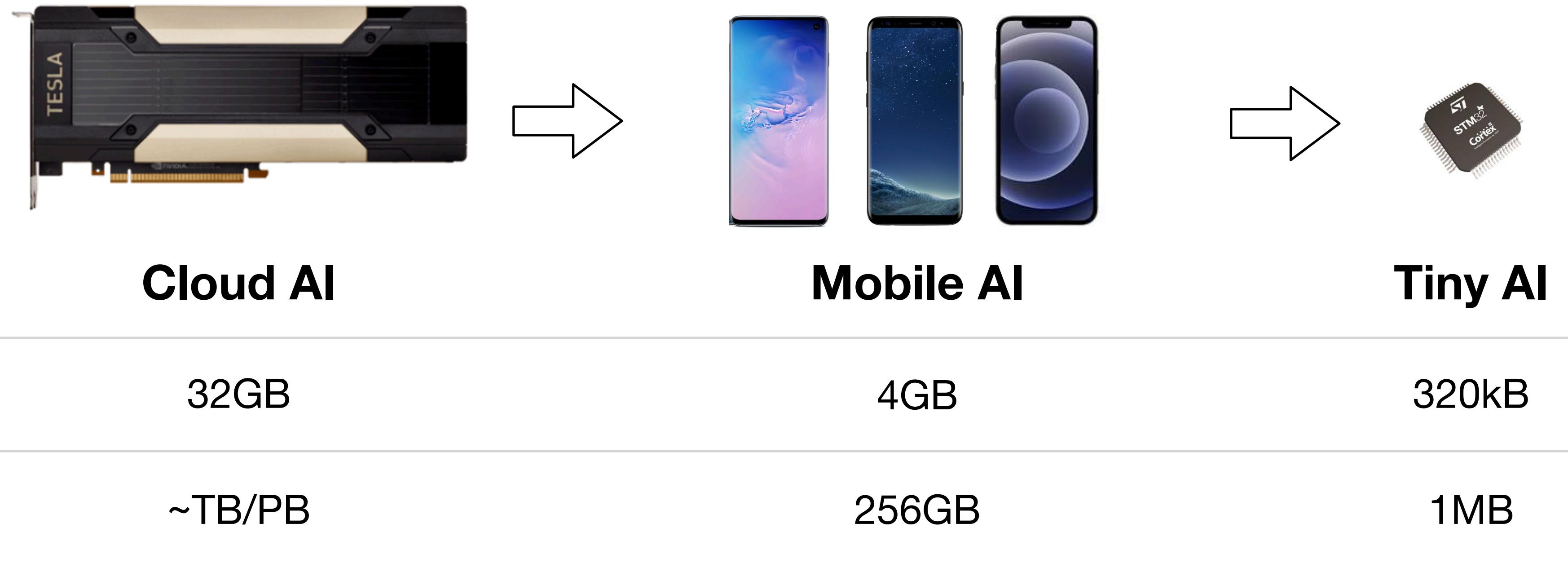


## Number of Edge Devices 2021

- 15 Billion Mobile phones
- 1.4 Billion Cars
- 770 Million Security cameras
- 15 Million Robots

# AI is Quickly Coming to the Edge

Edge AI devices are different from cloud AI processors



# Deep Learning for Image Classification

DNNs achieve super-human classification accuracy on ImageNet

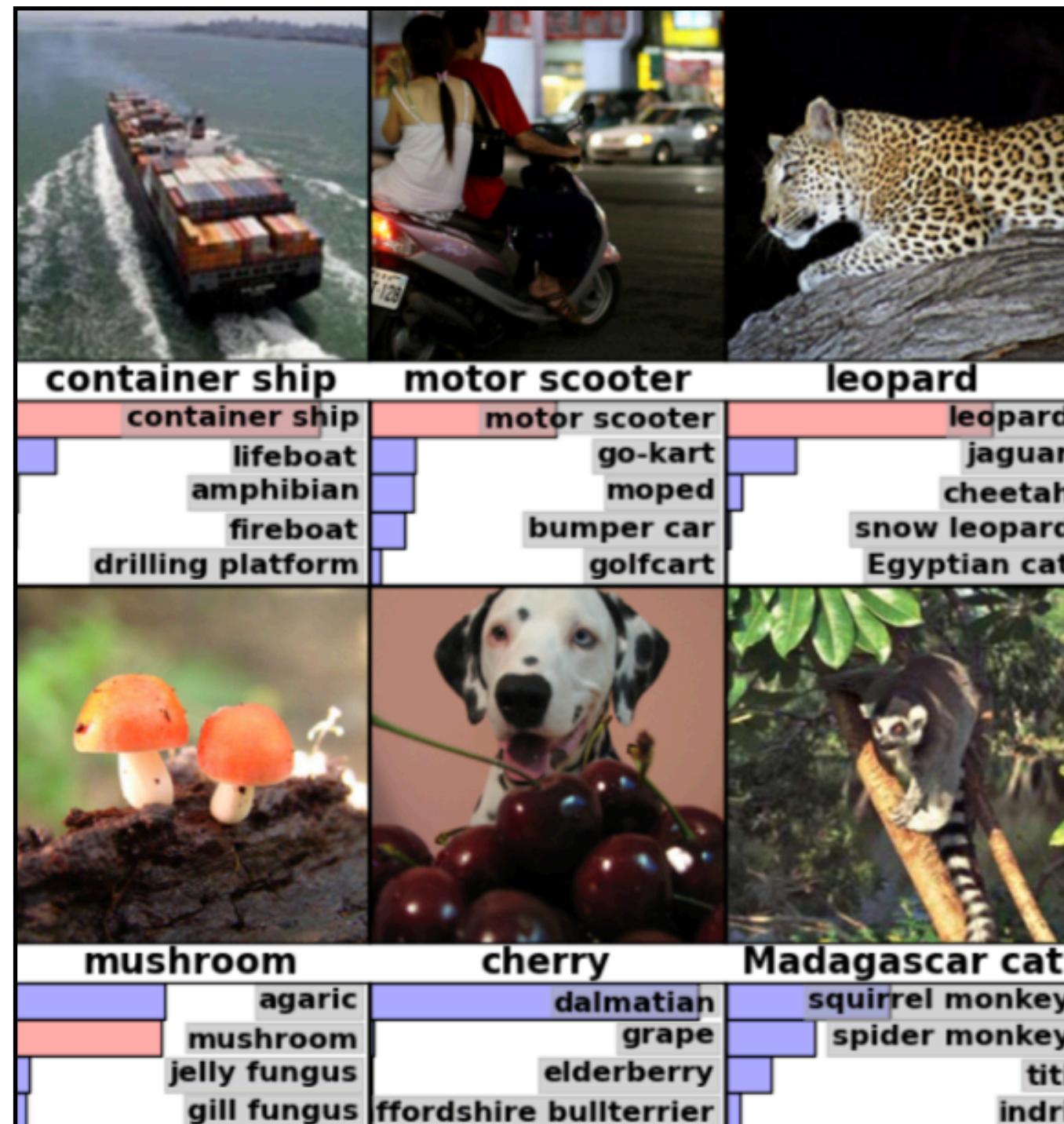
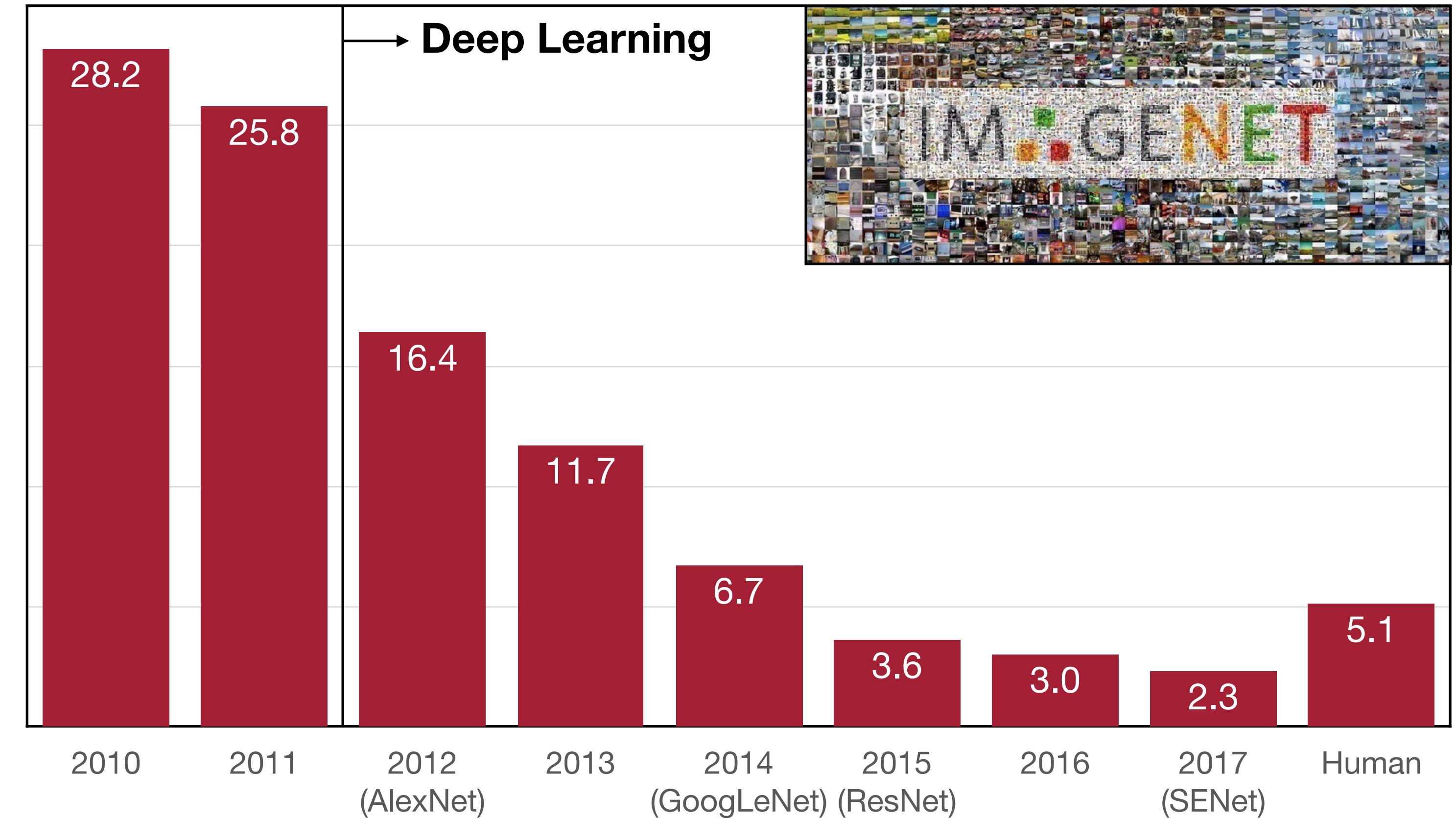


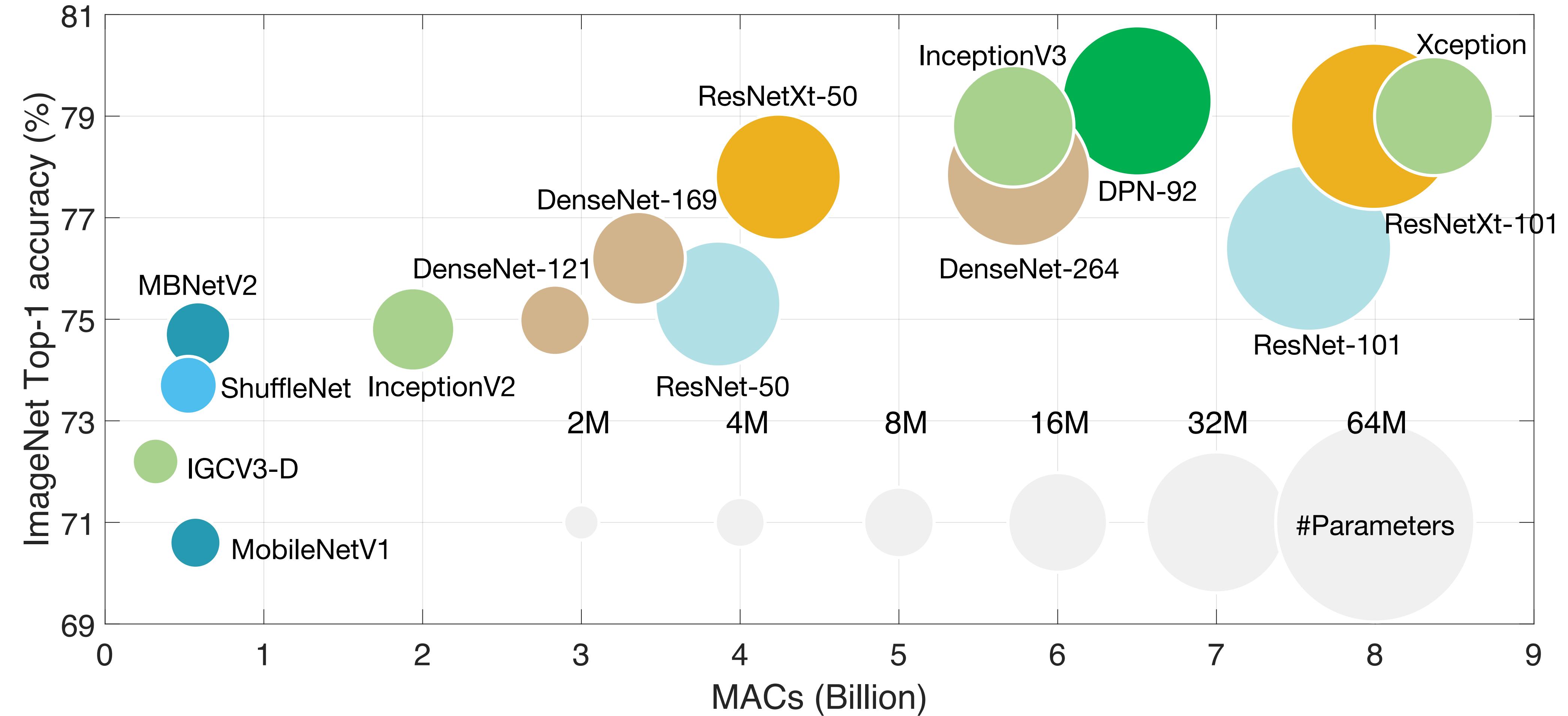
Image Classification

ImageNet Contest Winning Entry: Top 5 Error Rate (%)



# Deep Learning for Image Classification

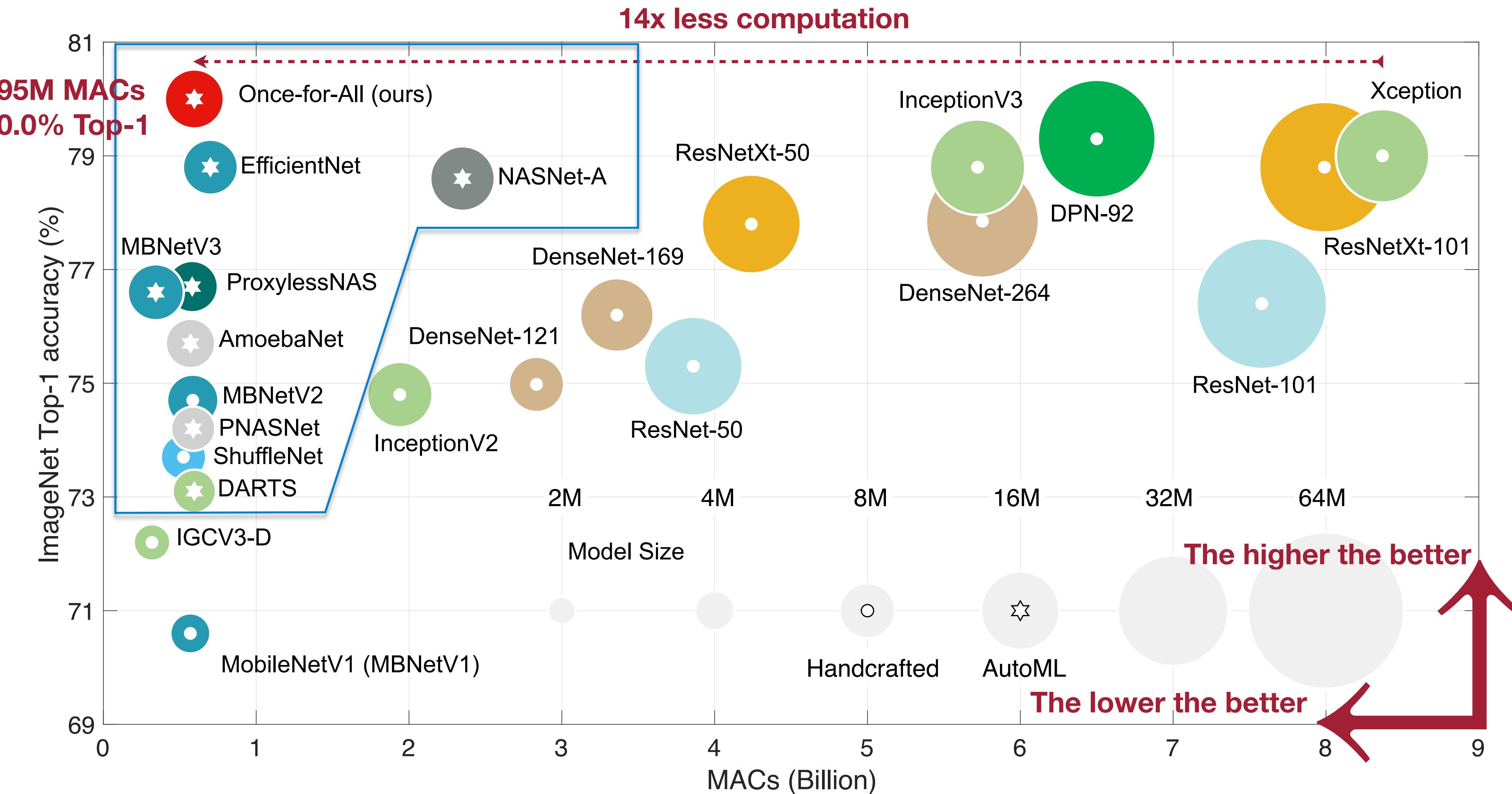
Computational cost of DNNs is growing



Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey [Deng et al., IEEE 2020]

# Efficient Deep Learning for Image Classification

Neural architecture search reduces the computational cost



Once-for-All: Train One Network and Specialize it for Efficient Deployment [Cai et al., ICLR 2020]

# Efficient Deep Learning for Image Classification

Efficient deep learning enables daily life application on mobile phones

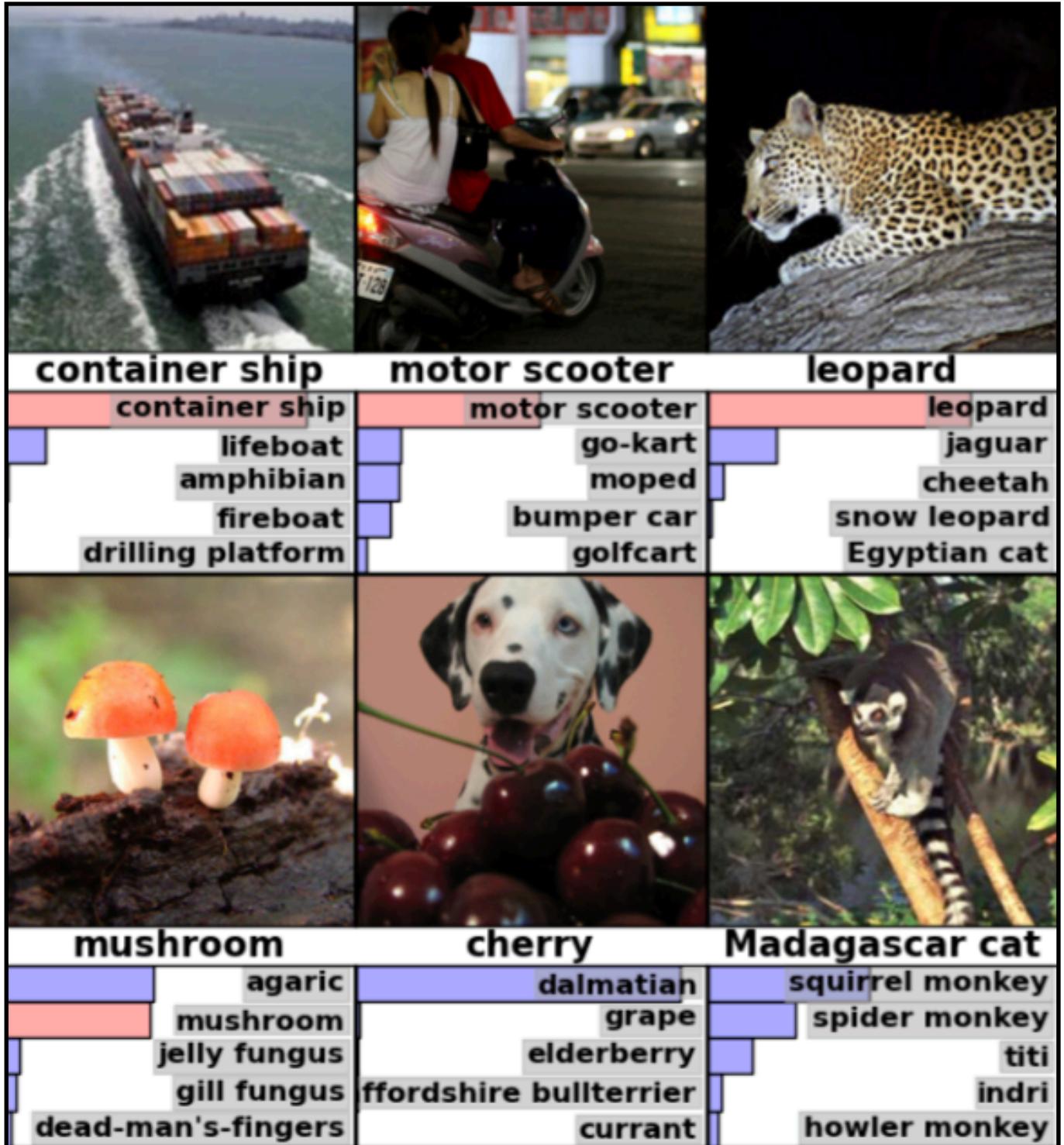


Image Classification

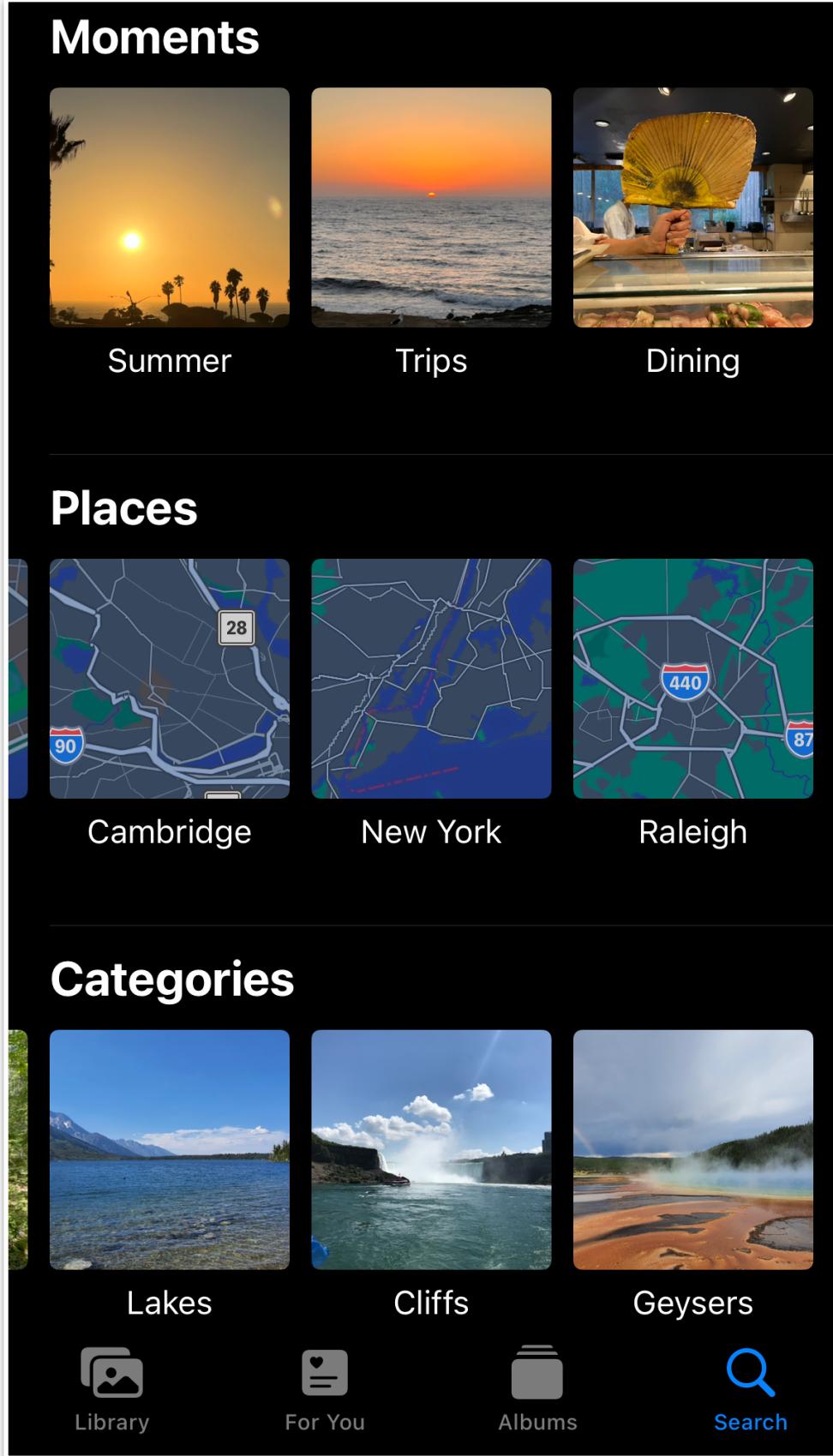
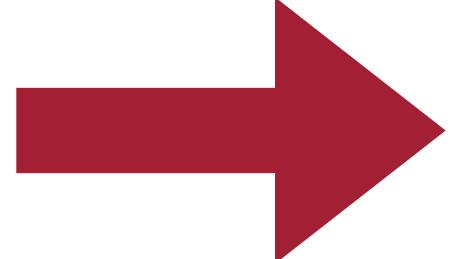
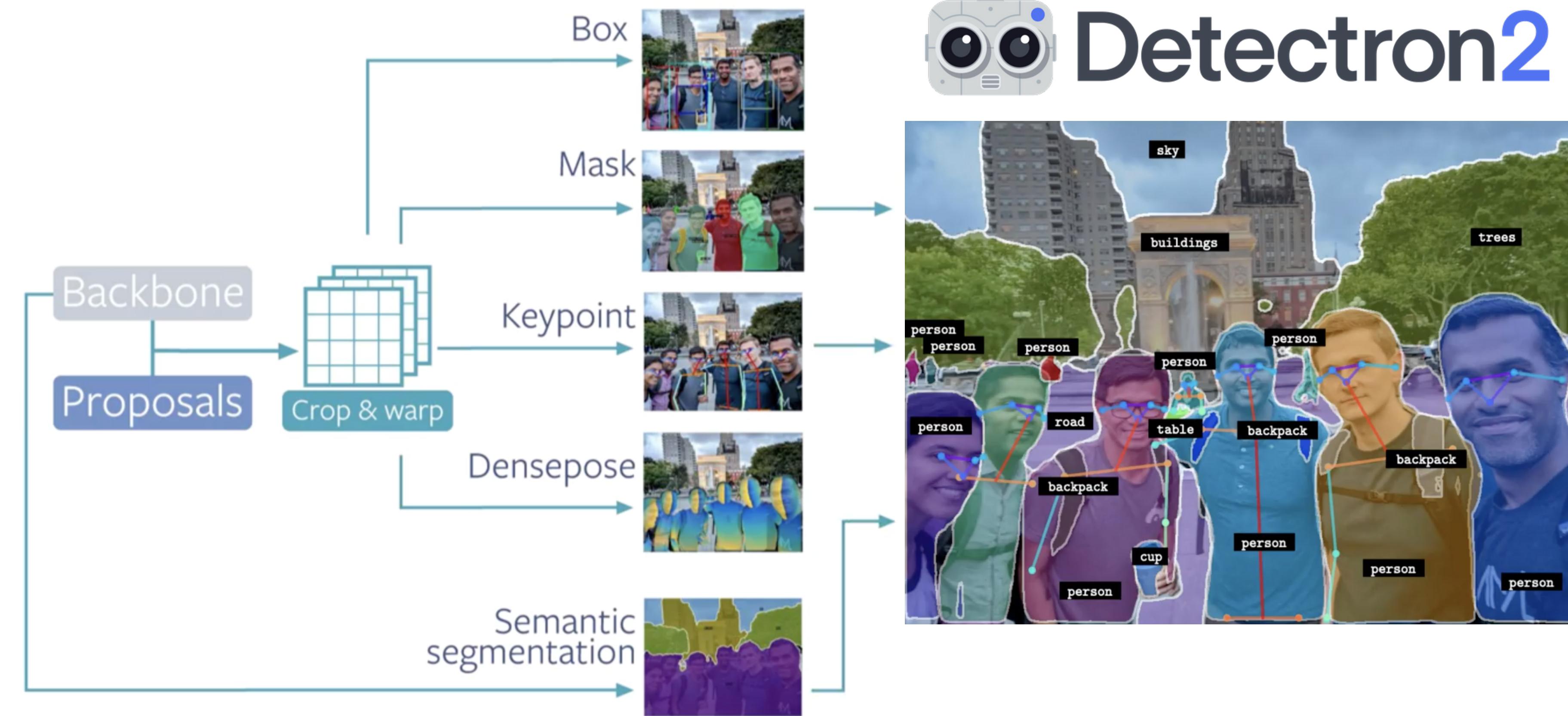


Photo Tags (on iPhone)

# Deep Learning for Image Recognition

## Detectron2: detection, segmentation, keypoint and pose estimation

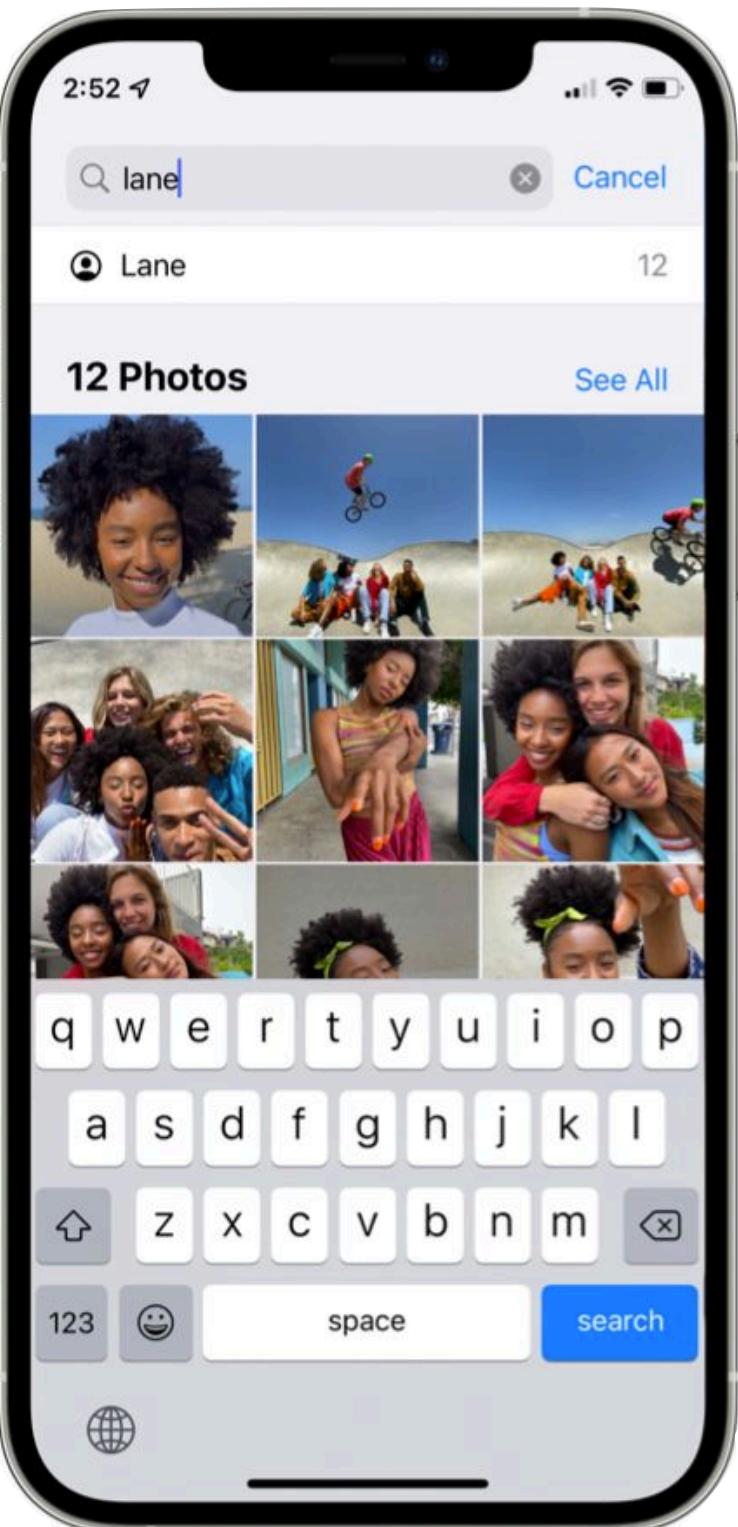
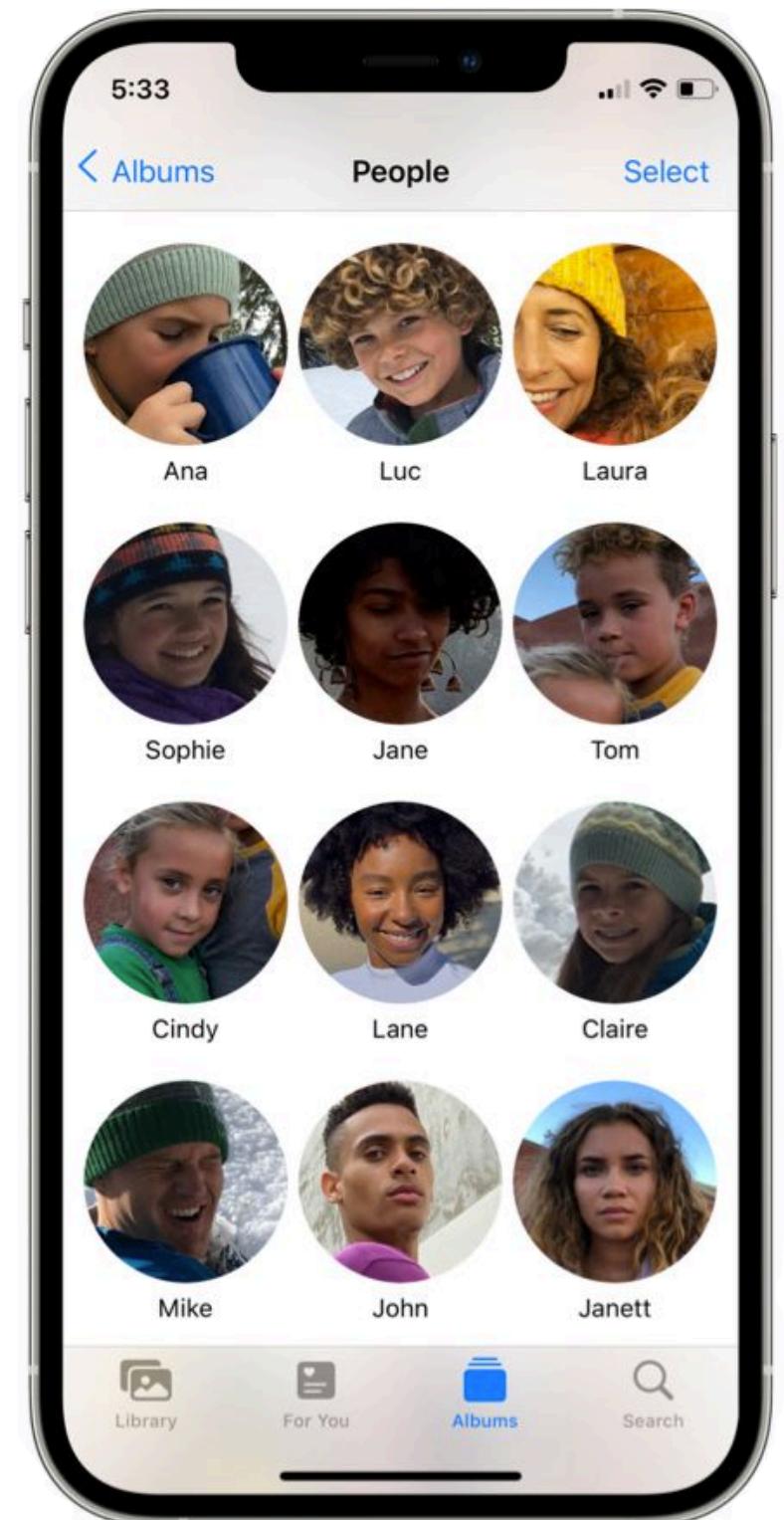
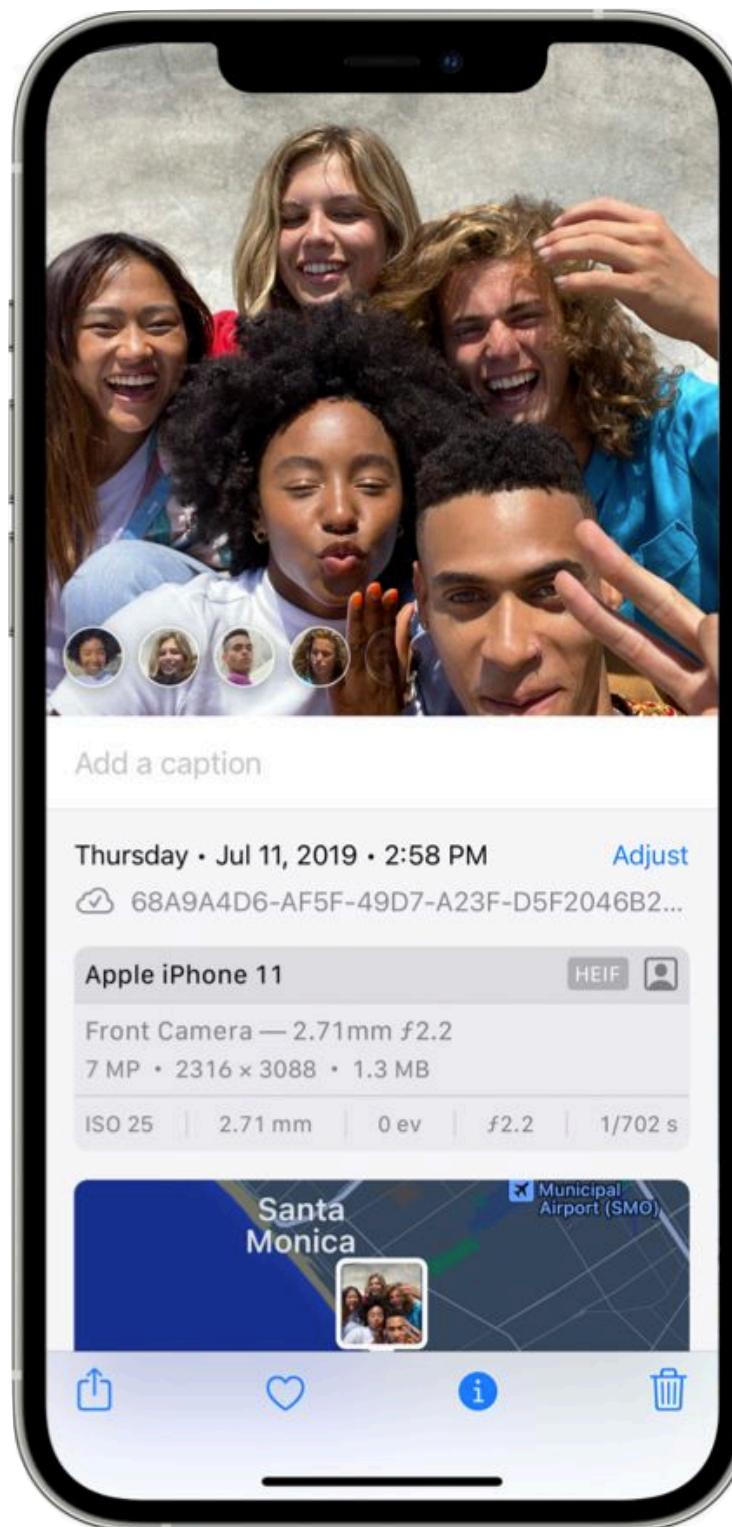


Detection, Segmentation, Keypoint & Pose Estimation

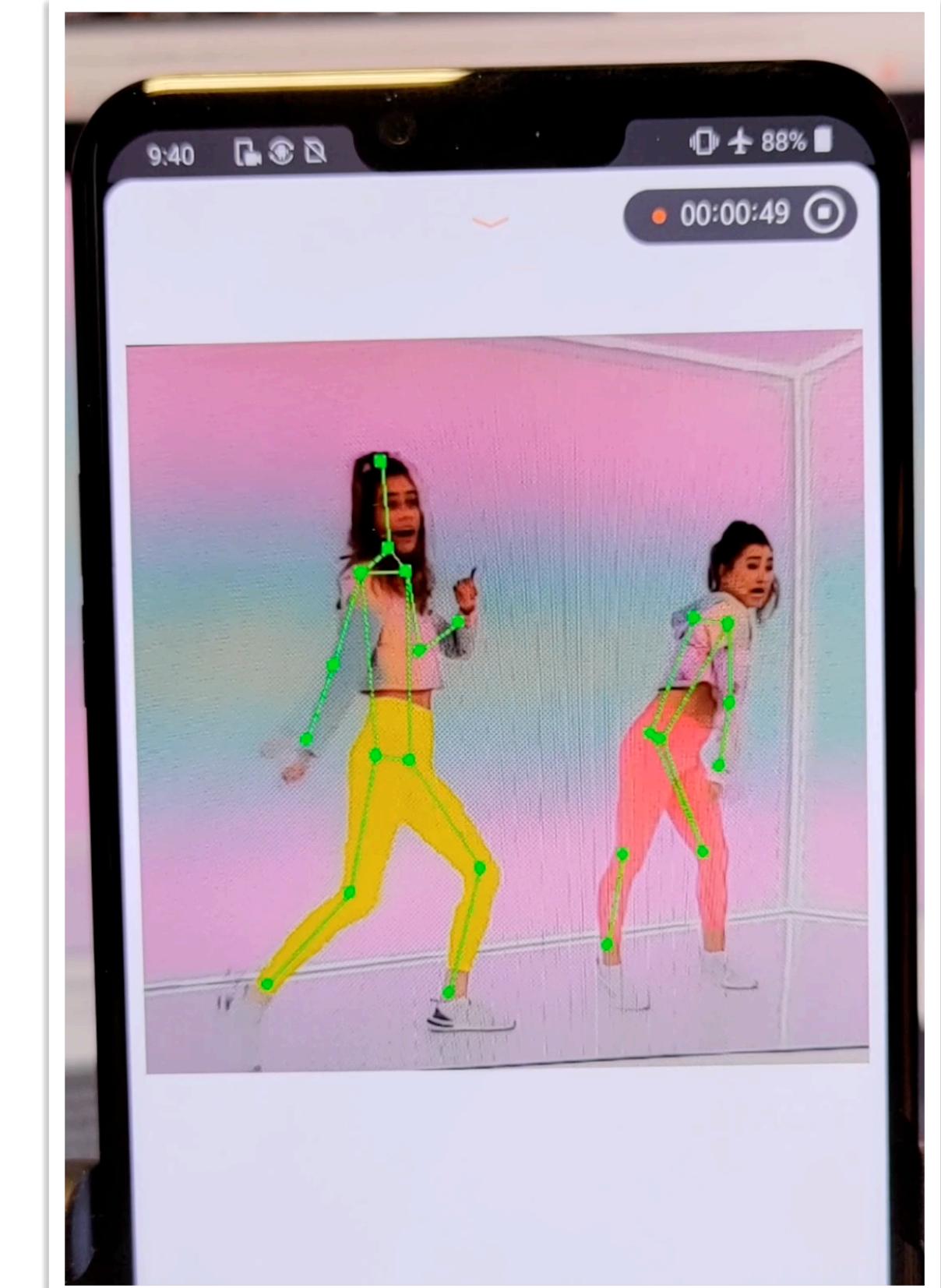
Detectron2: <https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/>

# Efficient Deep Learning for Image Recognition

Efficient deep learning enables daily life application on mobile phones



People Recognition (on iPhone)

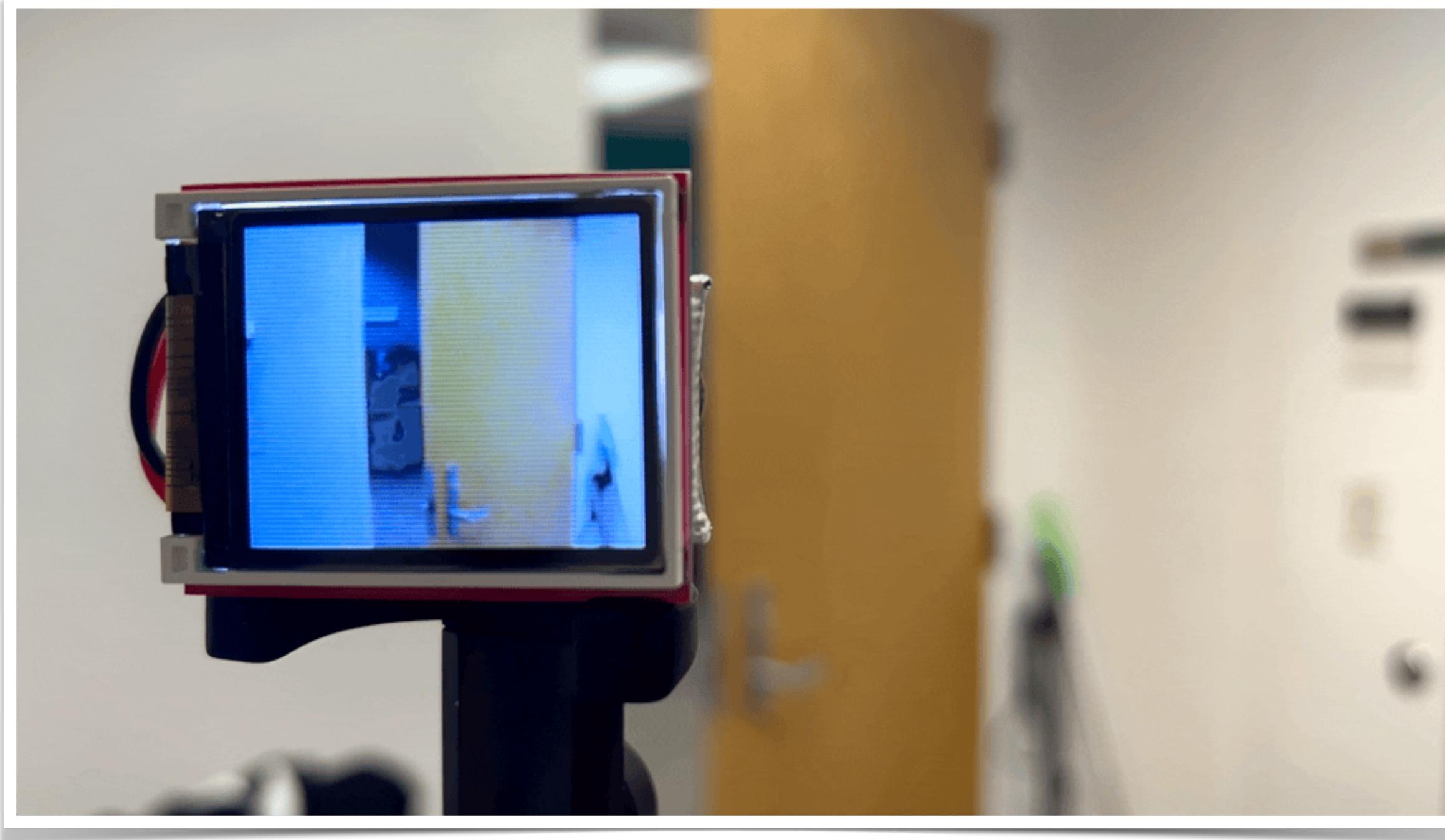


On-Device Pose Estimation

Lite Pose: Efficient Architecture Design for 2D Human Pose Estimation [Wang et al., CVPR 2022]  
[Recognizing People in Photos Through Private On-Device Machine Learning \[Apple, 2021\]](#)

# Efficient Deep Learning for Image Recognition

## MCUNet: tiny machine learning on IoT devices



Facial Mask Detection



Person Detection

MCUNet: Tiny Deep Learning on IoT Devices [Lin et al., NeurIPS 2020]

# Deep Learning for Image Generation

DALL·E 2 creates realistic images/art from a natural language description

Teddy bears, mixing sparkling chemicals as mad scientists, as a 1990s Saturday morning cartoon



A bowl of soup, as a planet in the universe, as a 1960s poster

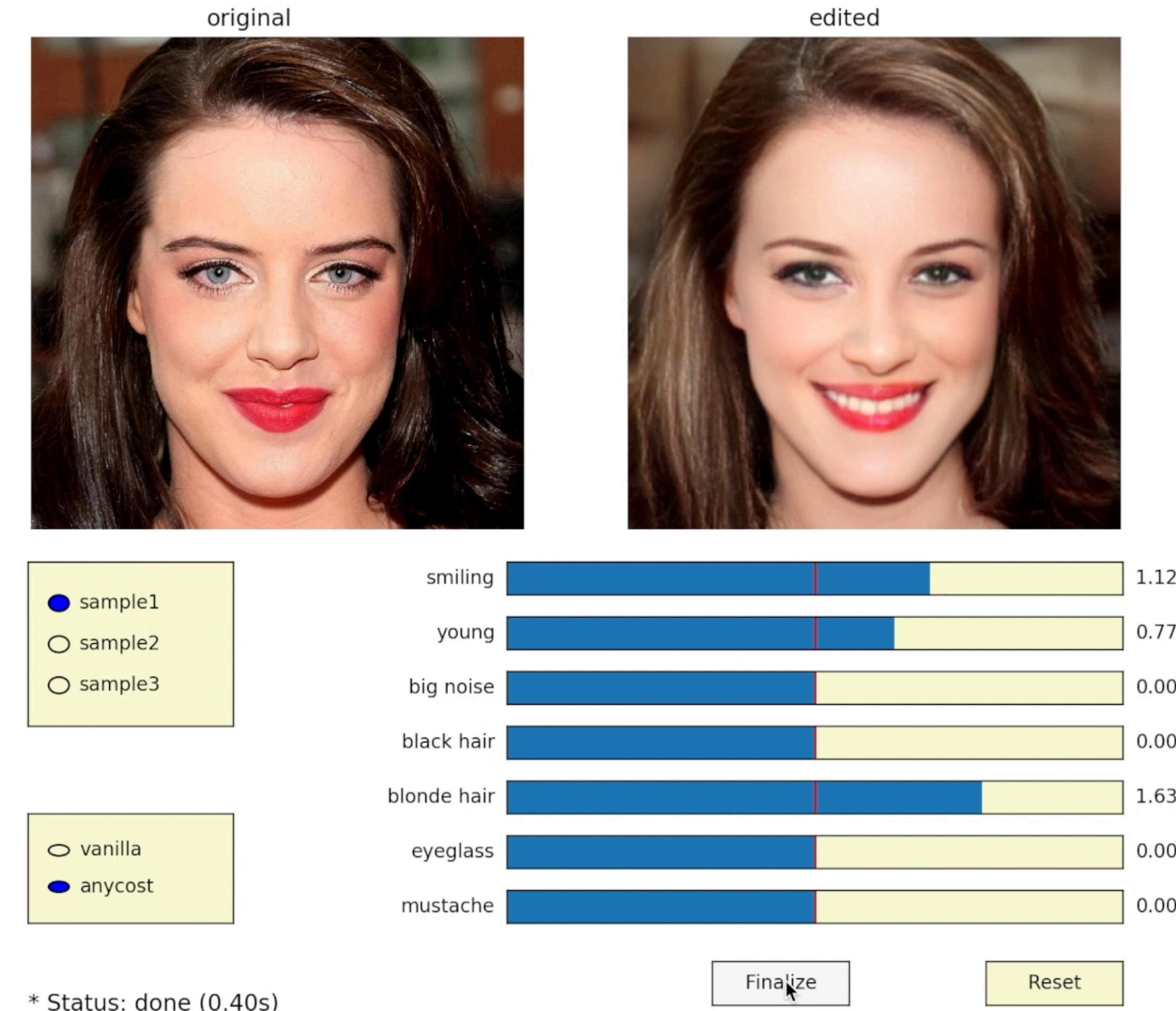


Compute: 256 GPUs for 2 weeks (>\$100k)

DALL·E 2: <https://openai.com/dall-e-2/>

# Efficient Deep Learning for Image Generation

## AnycostGAN: interactive image synthesis and editing on a laptop



Anycost GANs for Interactive Image Synthesis and Editing [Lin et al., CVPR 2021]

# Deep Learning for Text Translation

Neural machine translation bridges the language barrier

The screenshot shows the Google Translate mobile application interface. At the top, there are language selection bars for English (DETECTED), English, Spanish, French, and Chinese (Simplified). Below these, there are two text input fields. The left field contains the following text:

Have you found it difficult to deploy neural networks on mobile devices and IoT devices? Have you ever found it too slow to train neural networks? This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices. Topics cover efficient inference techniques, including model compression, pruning, quantization, neural architecture search, distillation; and efficient training techniques, including gradient compression and on-device transfer learning; followed by application-specific model optimization techniques for videos, point cloud and NLP; and efficient quantum machine learning. Students will get hands-on experience implementing deep learning applications on microcontrollers, mobile phones and quantum machines with an open-ended design project related to mobile AI.

The right field displays the translated text in Chinese (Simplified):

您是否发现很难在移动设备和物联网设备上部署神经网络？你有没有发现训练神经网络太慢了？本课程深入探讨有效的机器学习技术，这些技术可在资源受限的设备上实现强大的深度学习应用。主题涵盖高效推理技术，包括模型压缩、剪枝、量化、神经架构搜索、蒸馏；和高效的训练技术，包括梯度压缩和设备迁移学习；其次是针对视频、点云和 NLP 的特定应用模型优化技术；和高效的量子机器学习。学生将通过与移动 AI 相关的开放式设计项目获得在微控制器、手机和量子机器上实施深度学习应用程序的实践经验。

Below the translated text, there is a transcription in Pinyin: Nín shìfǒu fāxiàne hěn nán zài yídòng shèbèi hé wù liánwǎng shèbèi shàng bùshǔ shénjīng wǎngluò? Nǐ yǒu méiyǒu fāxiàne xùnliàn shénjīng wǎngluò tài mǎnle? Běn kèchéng shēnrù tàn tǎo yóuxiào de jīqì xuéxí jishù, zhèxiē jishù kě zài zīyuán shòu xiàn de

At the bottom of the right panel, there is a "Show more" link and a "Share" icon.

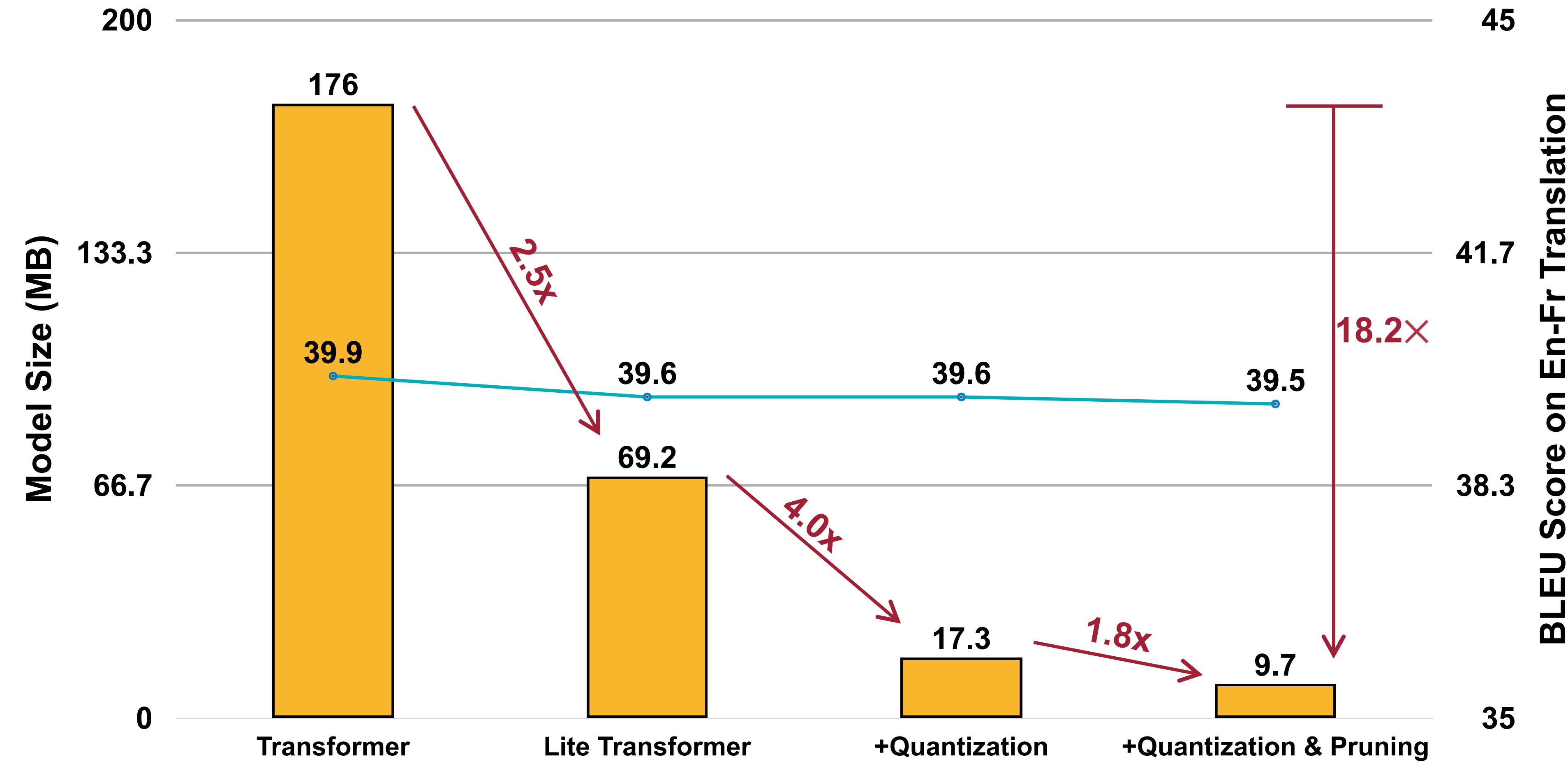


Conversation Translation  
(on iPhone)

Google Translate: <https://translate.google.com/>

# Efficient Deep Learning for Text Translation

Lite Transformer reduces the model size with pruning and quantization



Lite Transformer with Long-Short Range Attention [Wu et al., ICLR 2020]

# Deep Learning for Language Modeling

## GPT-3 language model

Have you found it difficult to deploy neural networks on mobile devices and IoT devices?

Yes, it can be difficult to deploy neural networks on mobile devices and IoT devices due to their limited resources.

Have you ever found it too slow to train neural networks?

Yes, training neural networks can be slow, especially when training on large datasets.

This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices.

In this course, you will learn how to:

- Understand the challenges of deploying neural networks on resource-constrained devices
- Explore efficient neural network architectures that are well-suited for resource-constrained devices
- Implement efficient neural network training algorithms that can speed up training
- Optimize neural network inference for resource-constrained devices
- Deploy efficient neural networks on mobile and IoT devices

6.S965

Logistics Schedule

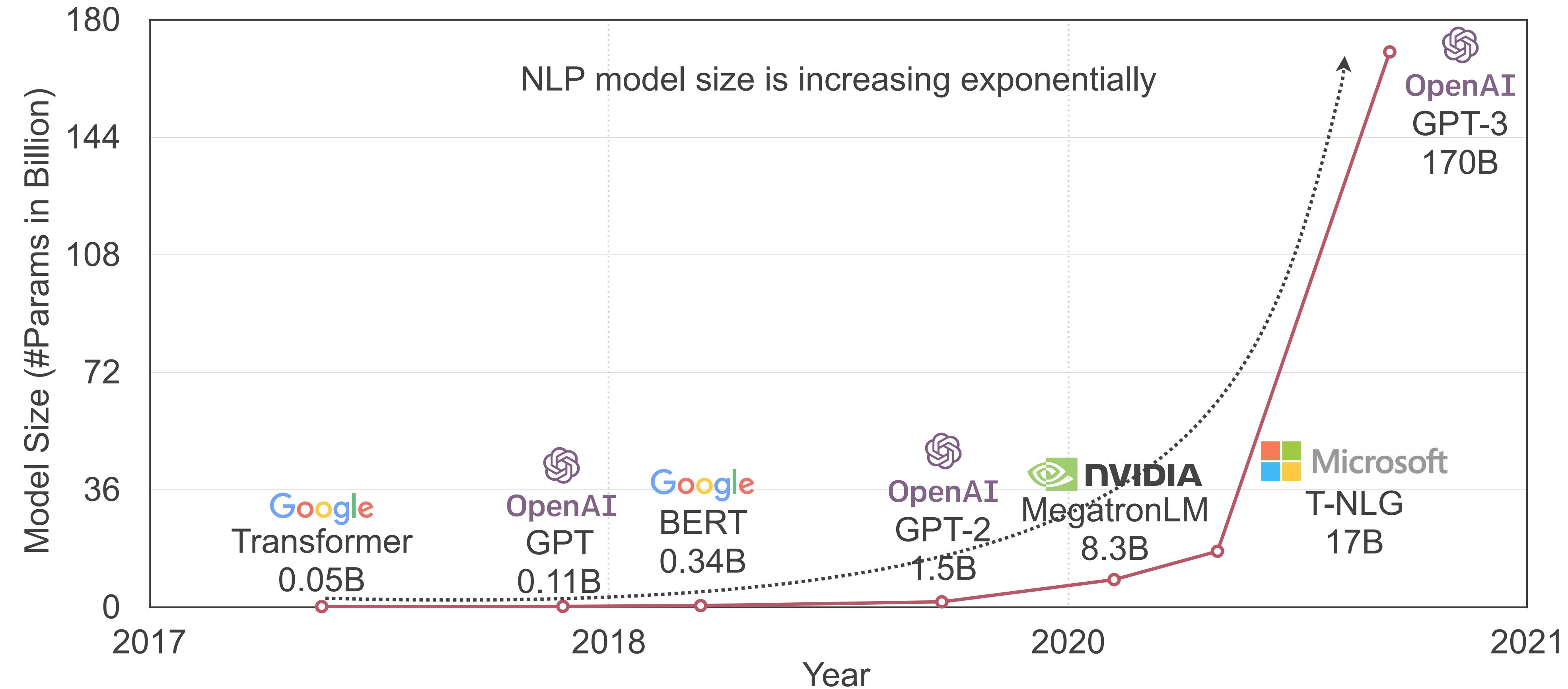
### TinyML and Efficient Deep Learning

6.S965 • Fall 2022 • MIT

Have you found it difficult to deploy neural networks on mobile devices and IoT devices? Have you ever found it too slow to train neural networks? This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices. Topics cover efficient inference techniques, including model compression, pruning, quantization, neural architecture search, and distillation; and efficient training techniques, including gradient compression and on-device transfer learning; followed by application-specific model optimization techniques for videos, point cloud, and NLP; and efficient quantum machine learning. Students will get hands-on experience implementing deep learning applications on microcontrollers, mobile phones, and quantum machines with an open-ended design project related to mobile AI.

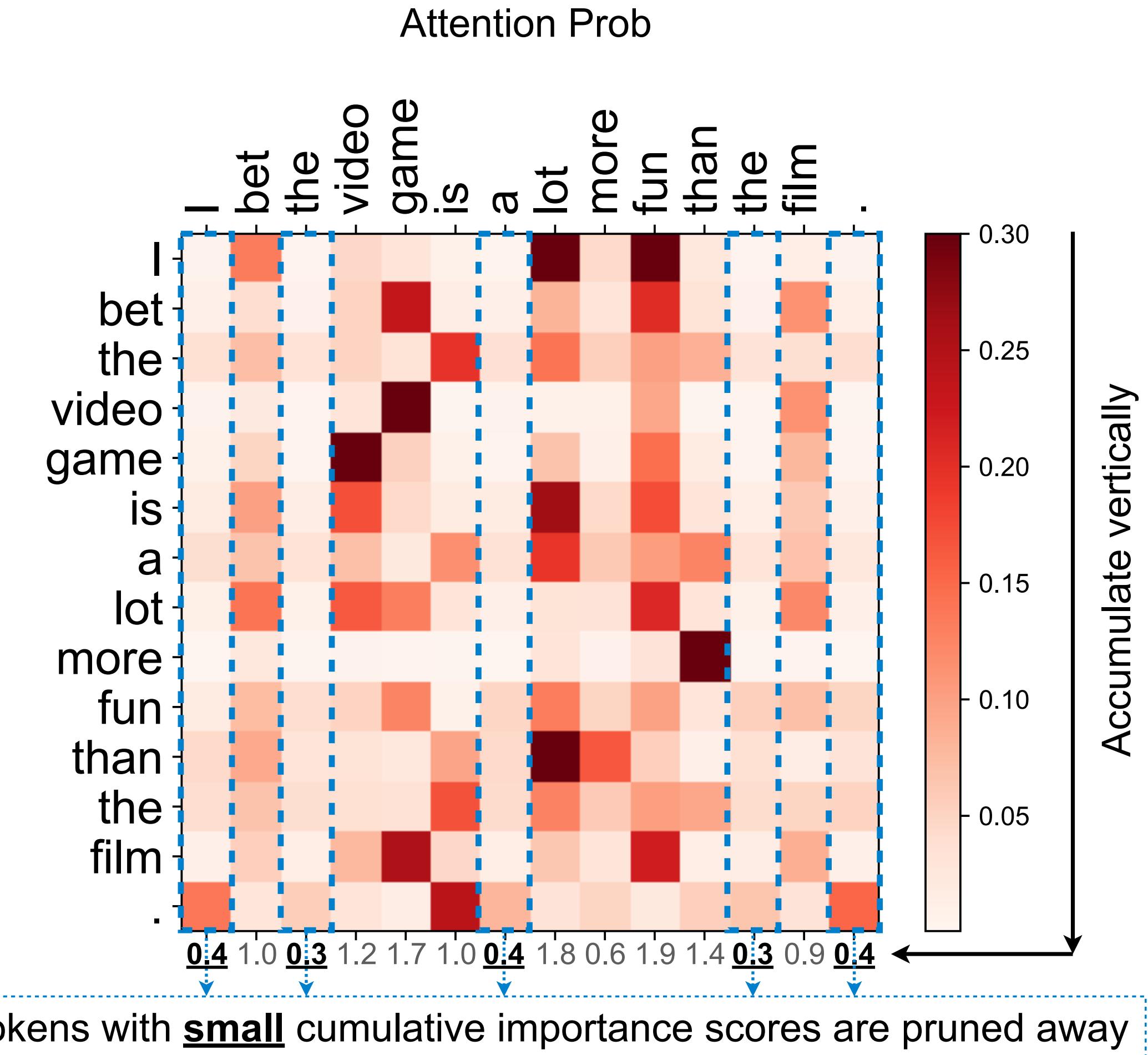
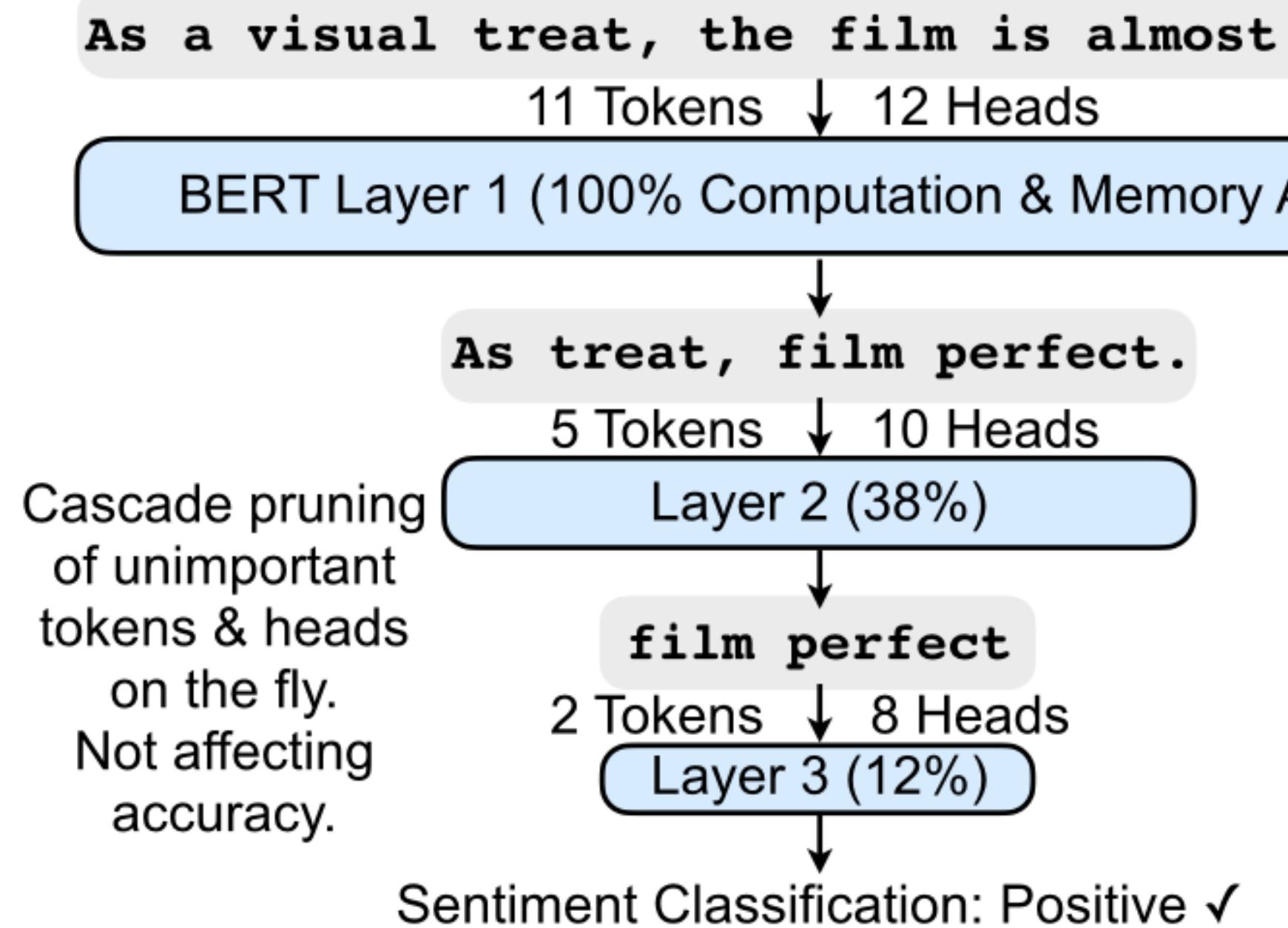
# Deep Learning for Language Modeling

Model size of language models is growing exponentially



# Efficient Deep Learning for Language Modeling

SpAtten accelerates language models by pruning redundant tokens

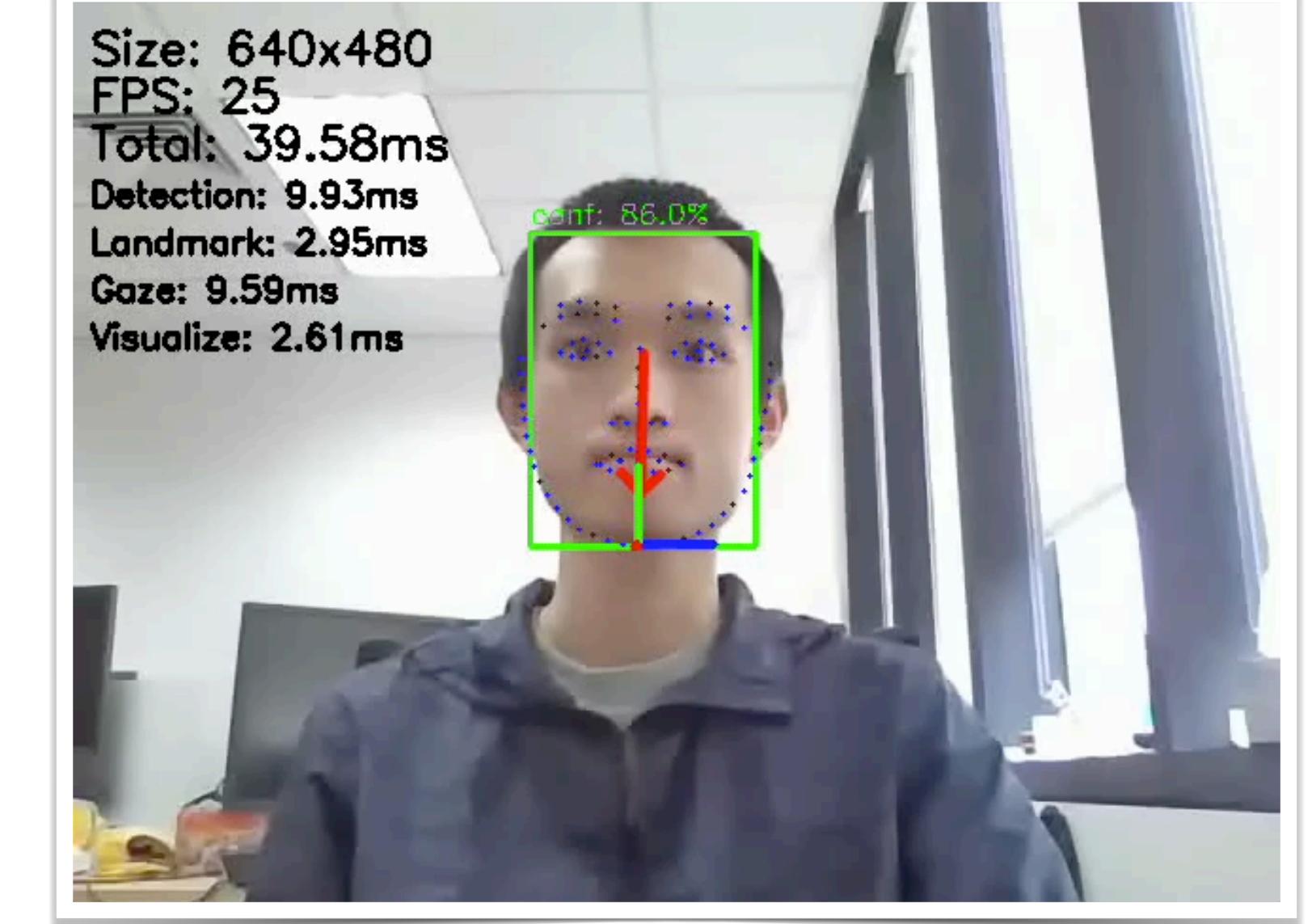


# Deep Learning for Computational Photography

Computational photography helps capture more sensational shots



Size: 640x480  
FPS: 25  
Total: 39.58ms  
Detection: 9.93ms  
Landmark: 2.95ms  
Gaze: 9.59ms  
Visualize: 2.61ms



Facial Landmarking and Gaze Estimation  
(on Raspberry Pi)

<https://www.theverge.com/2019/9/10/20850694/apple-iphone-11-pro-max-camera-specs-features-triple-three-slofies>

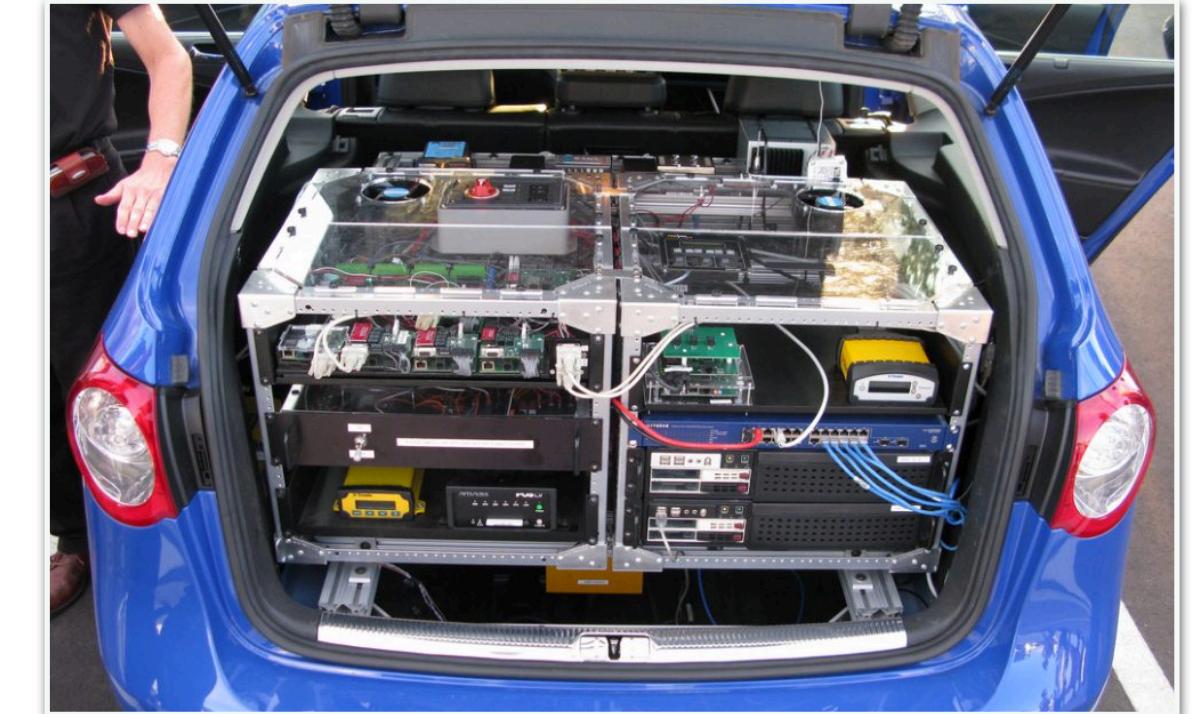
# Deep Learning for Autonomous Driving

Deep learning helps machine perceive the surrounding environment



Waymo Driver

Waymo Driver: <https://www.youtube.com/watch?v=2CVInKMz9cA>



A whole trunk of workstation

# Efficient Deep Learning for Autonomous Driving

Fast-LiDARNet accelerates 3D perception with algorithm/system co-design

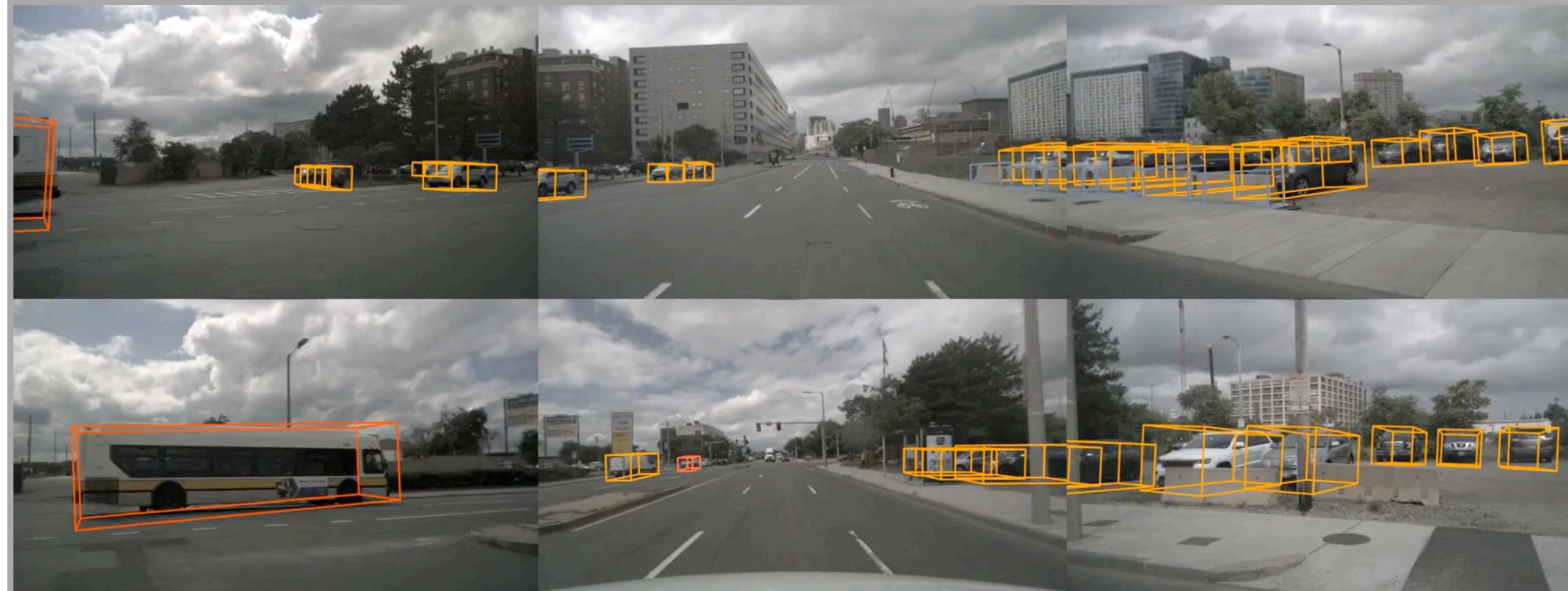


Efficient and Robust LiDAR-Based End-to-End Navigation [Liu et al., ICRA 2021]

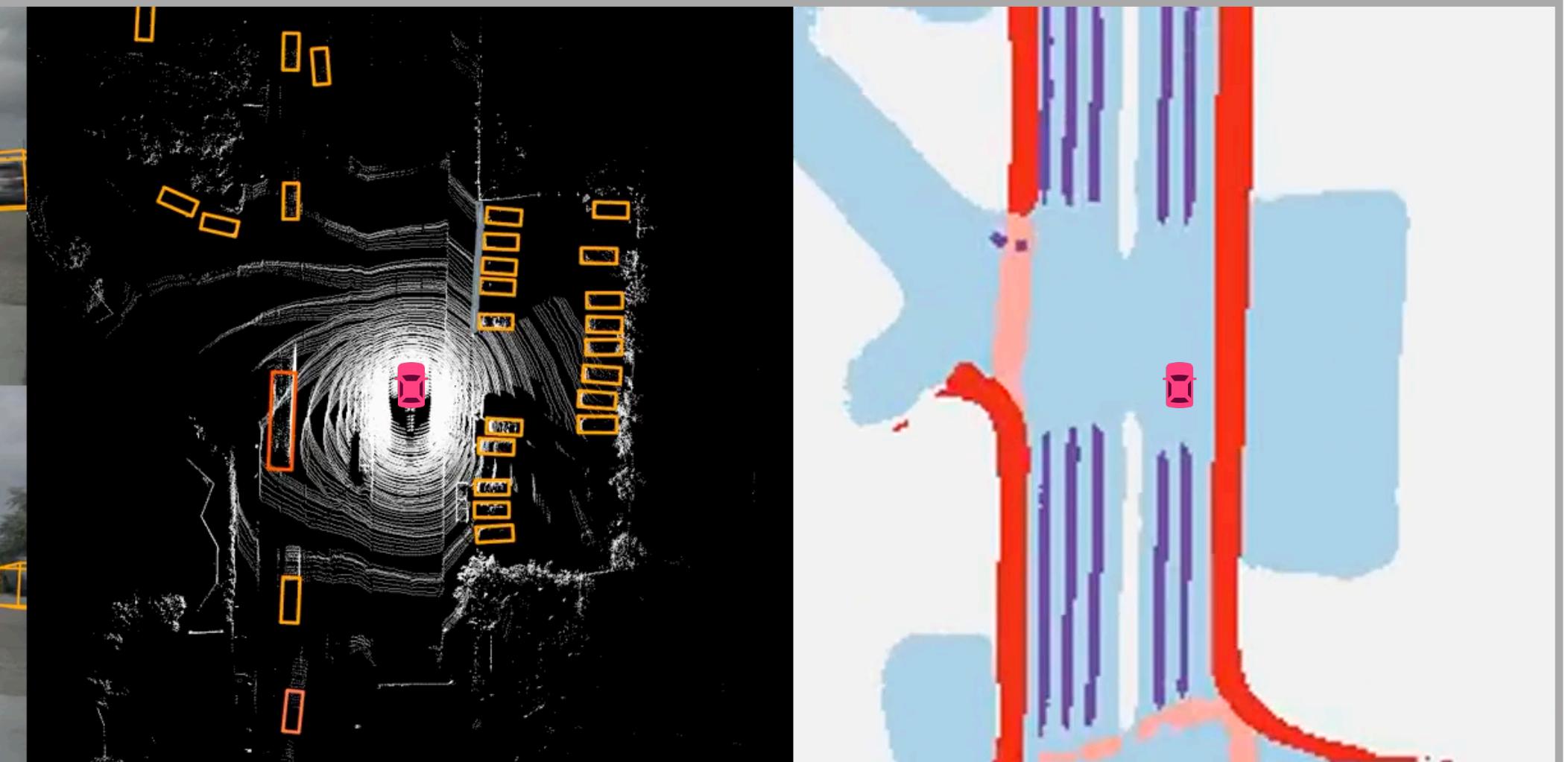
# Efficient Deep Learning for Autonomous Driving

**BEVFusion takes multi-modal sensory inputs and supports multiple tasks**

3D Object Detection



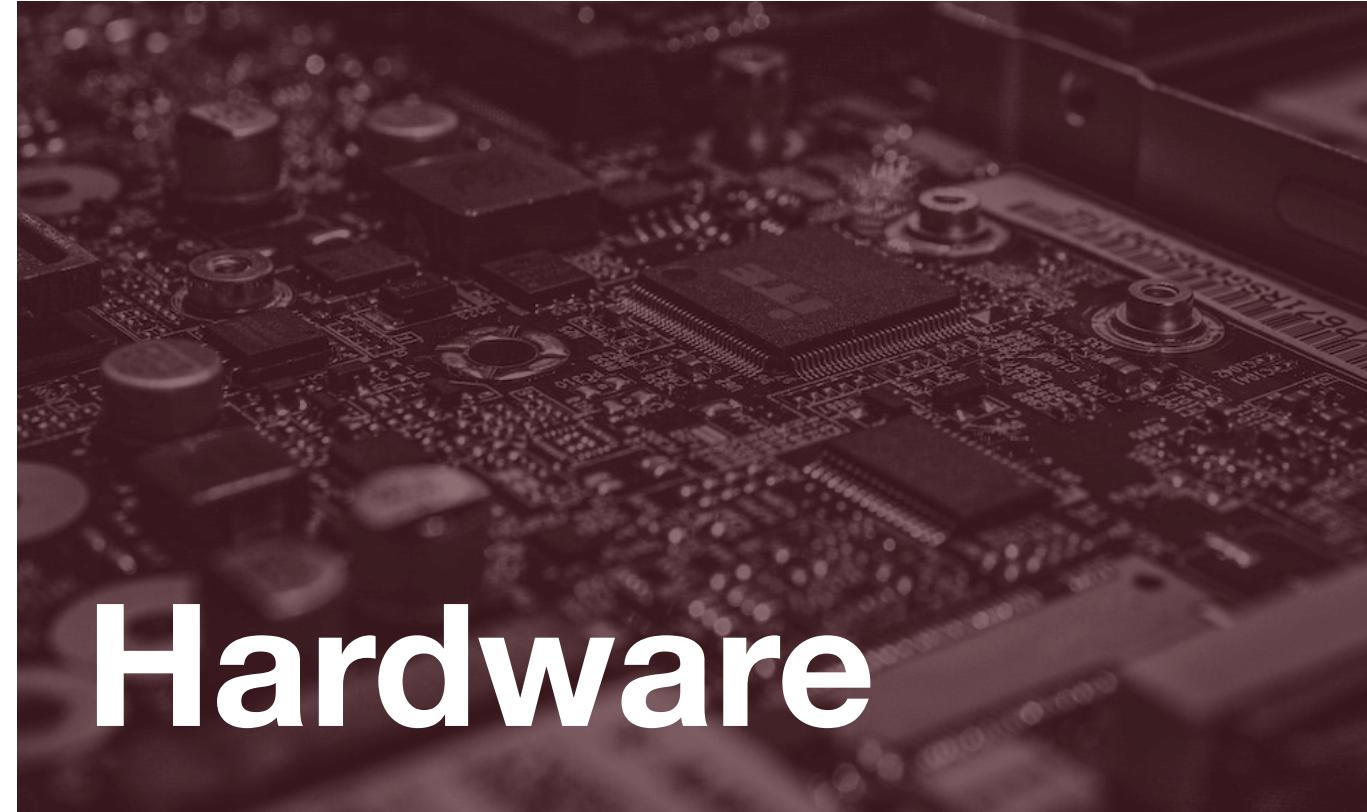
BEV Map Segmentation



□ : Car   □ : Truck   □ : Pedestrian   □ : Barrier   □ : Drivable Area   □ : Lane Divider   □ : Walkway   □ : Crosswalk

BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation [Liu et al., Arxiv 2022]

# Deep Learning: Three Pillars



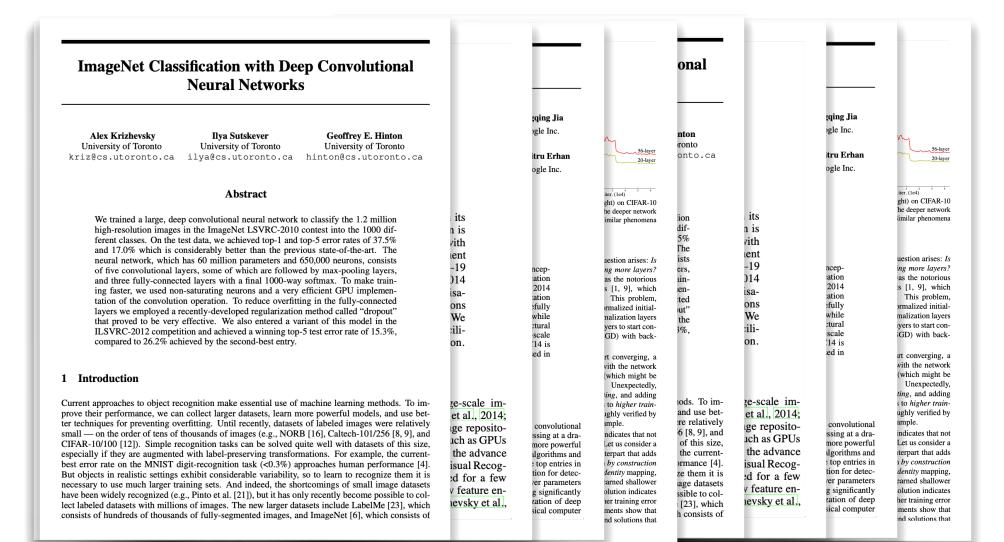
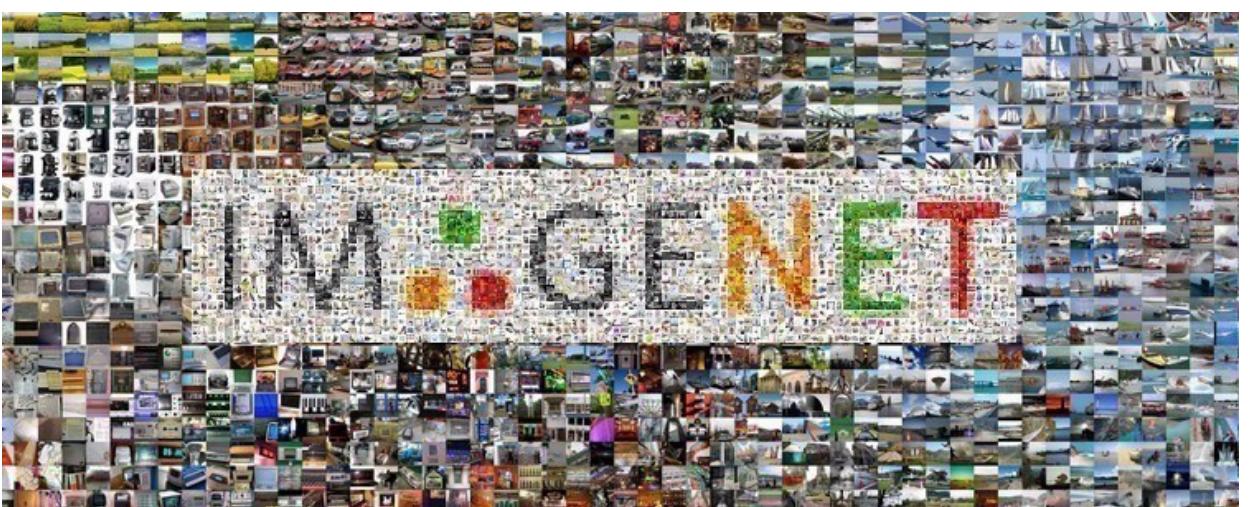
## Hardware



## Data



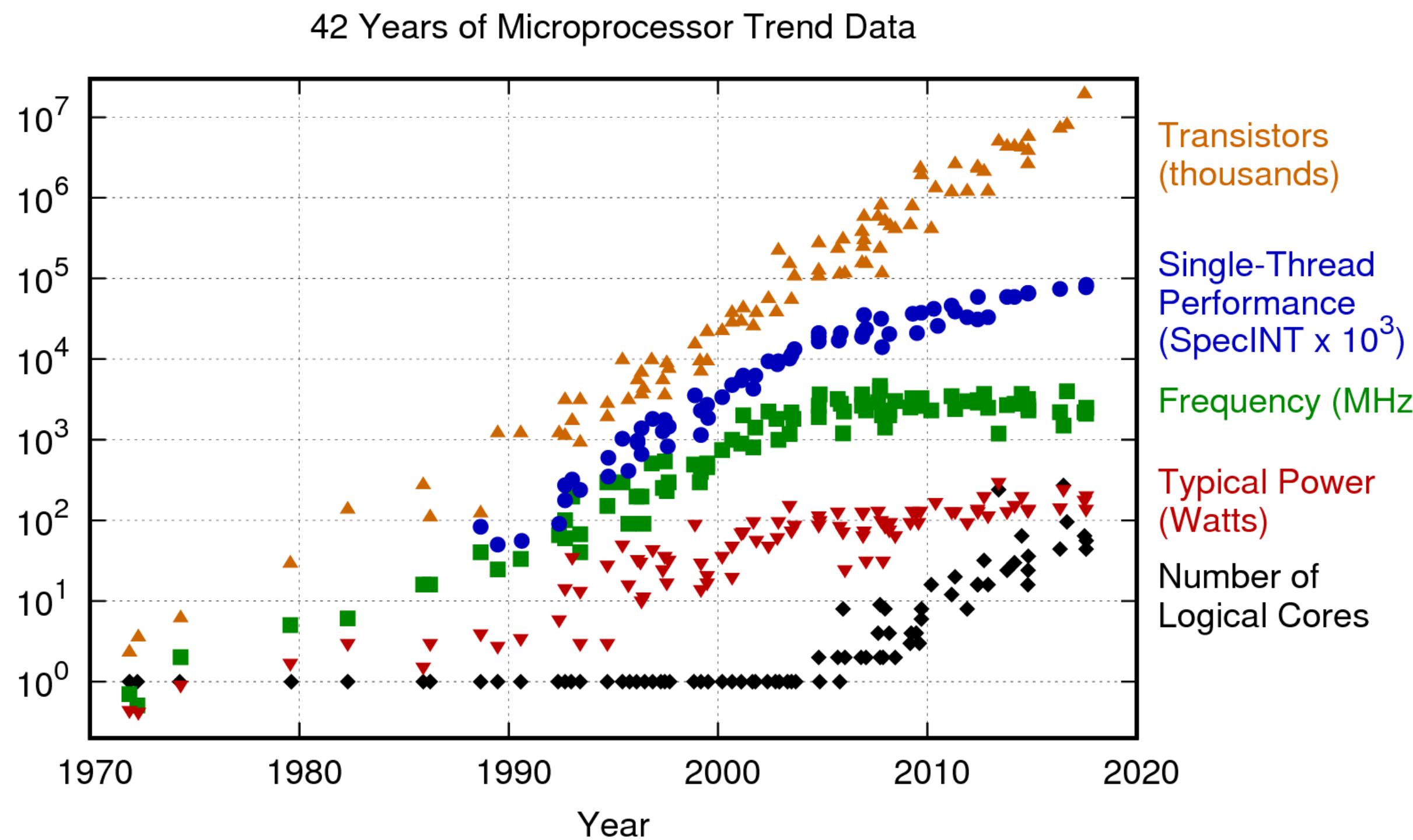
## Algorithm



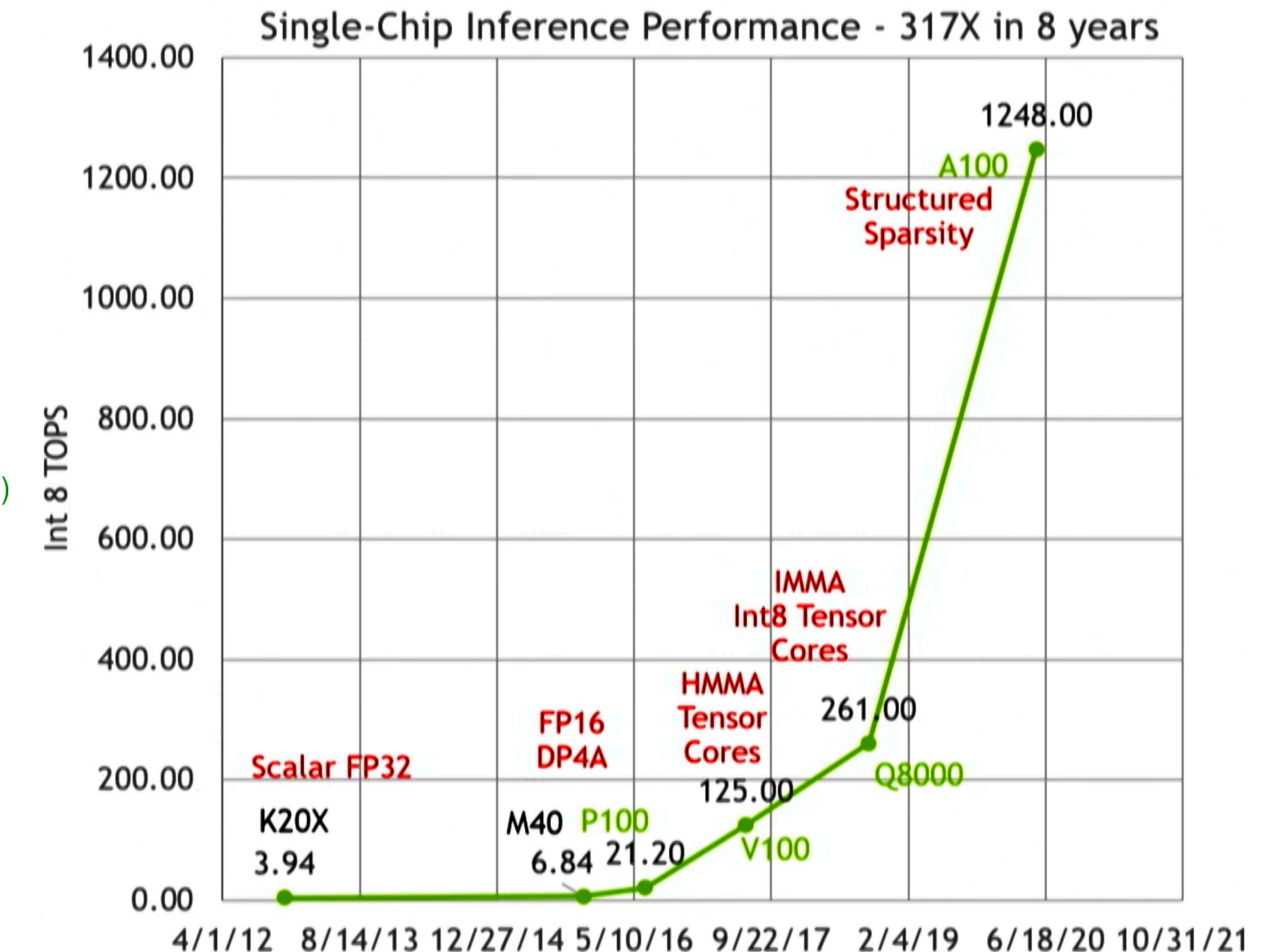
ImageNet Classification with Deep Convolutional Neural Networks

# Architectural support for quantization/pruning brings tremendous improvement

**FP32 => FP16 => Int8; dense => sparse**



Data source: M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten, K. Rupp

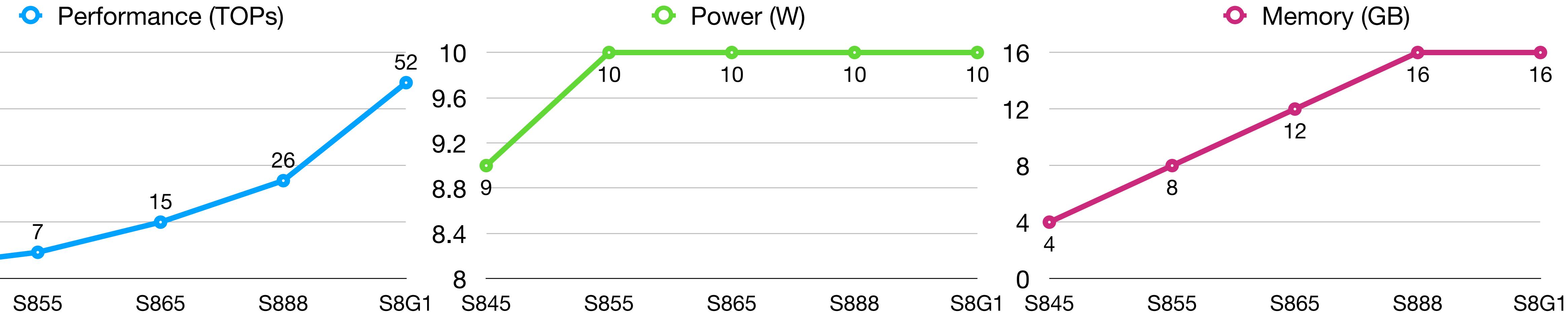


GPUs, Machine Learning, and EDA — Bill Dally

# Edge AI Hardware

## Qualcomm Hexagon DSP

- Qualcomm Hexagon is a family of digital signal processor (DSP) products by Qualcomm. It is designed to deliver performance with low power over a variety of applications

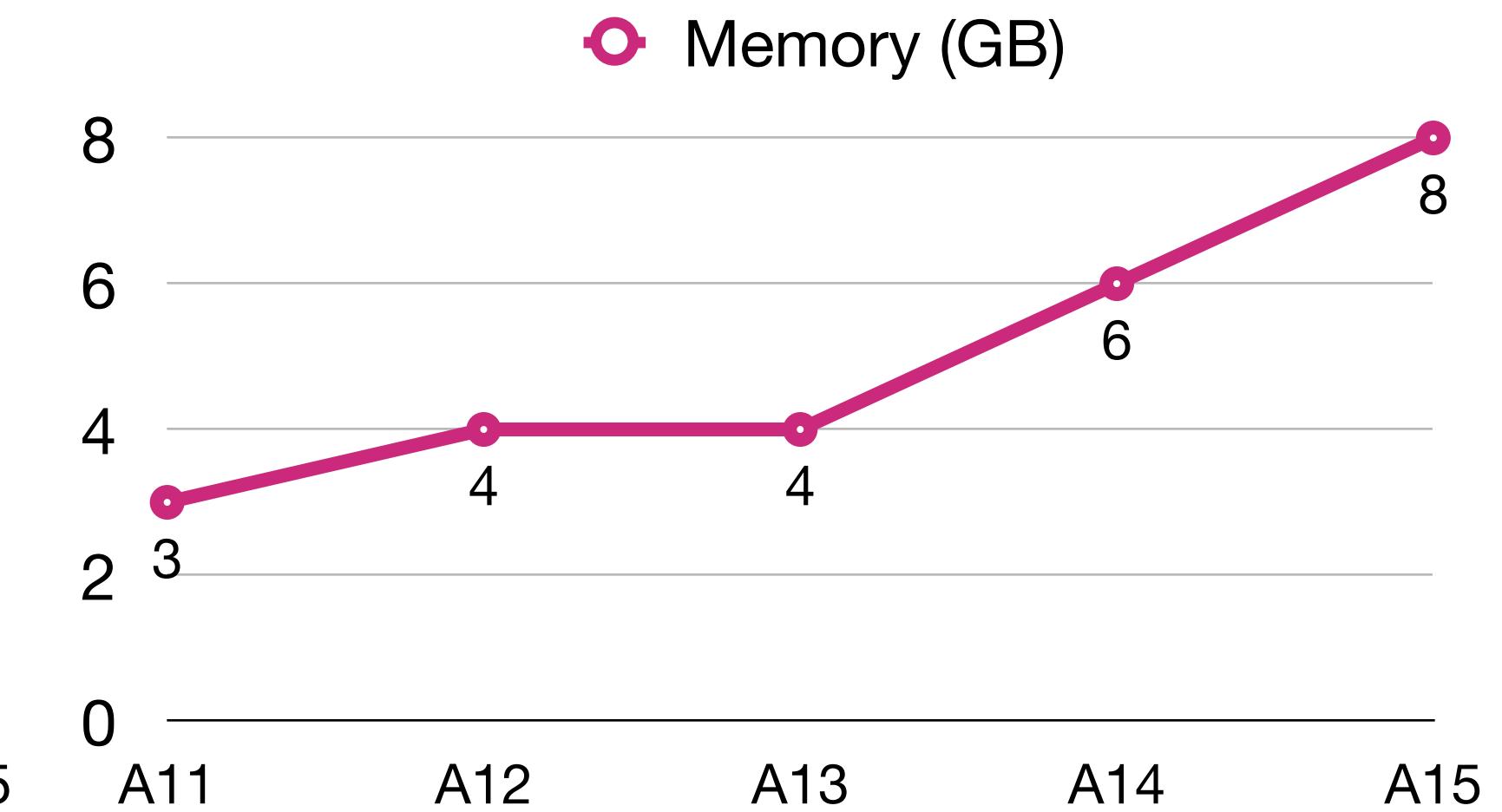
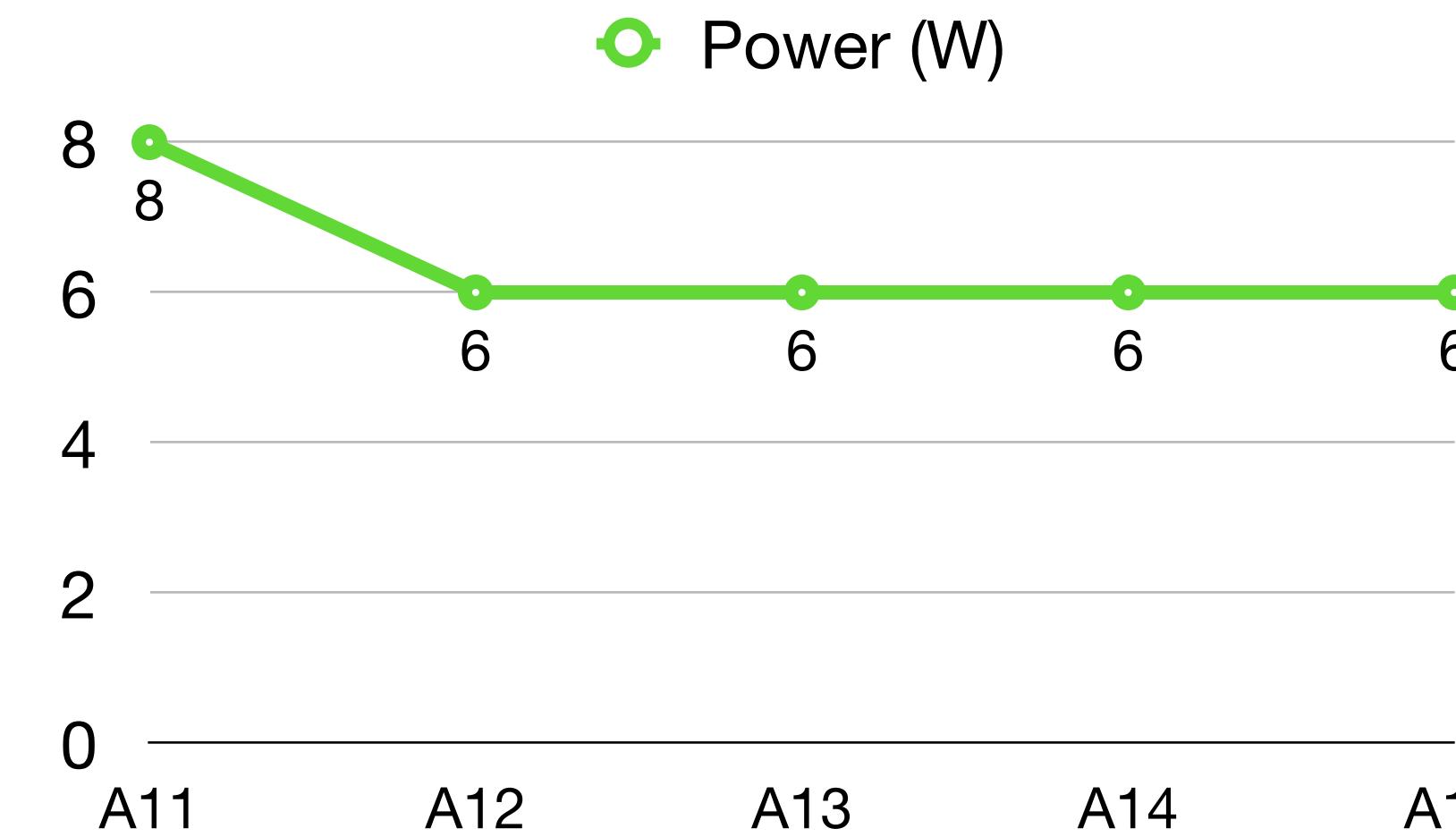
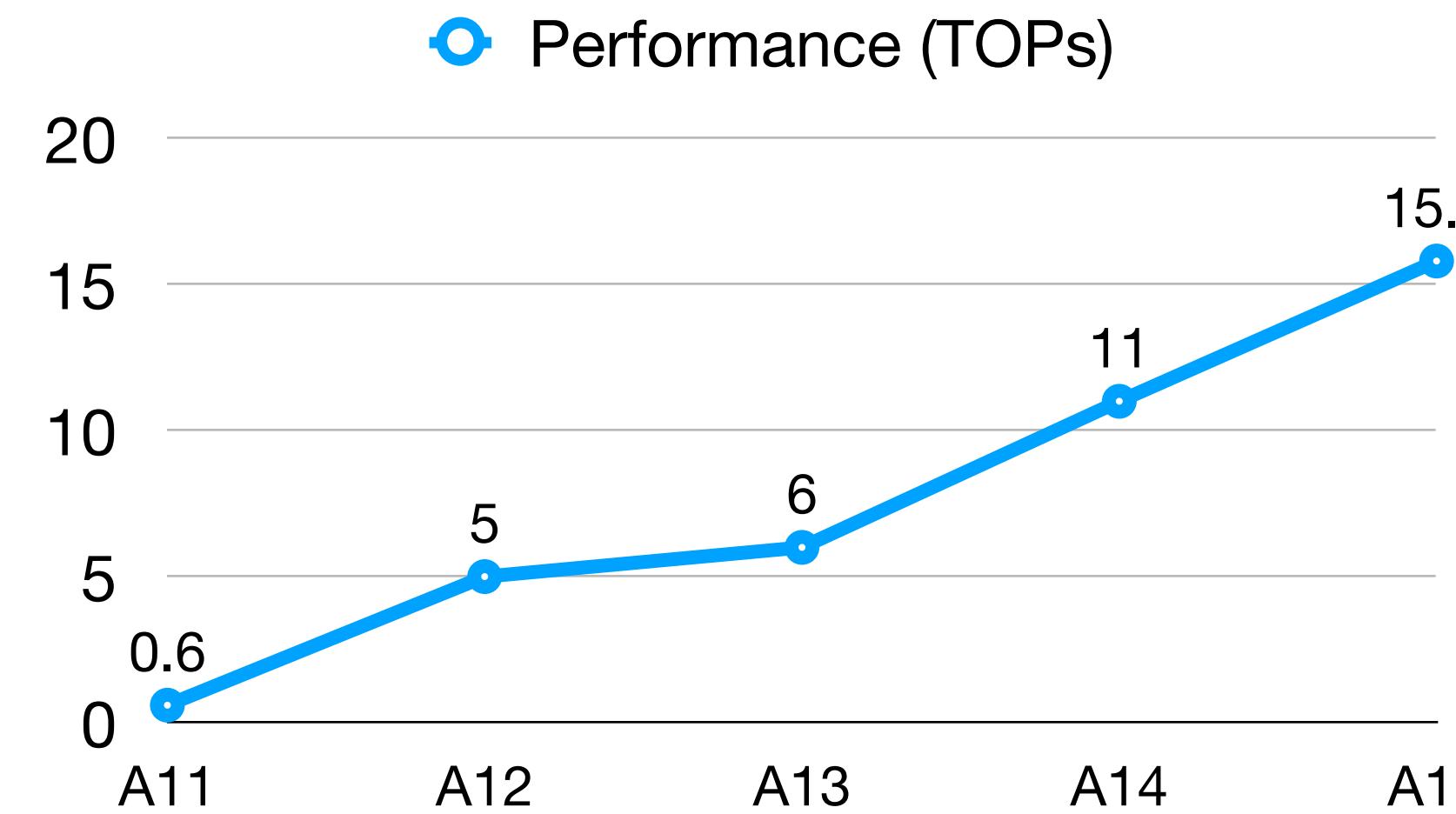


[https://en.wikipedia.org/wiki/List\\_of\\_Qualcomm\\_Snapdragon\\_processors](https://en.wikipedia.org/wiki/List_of_Qualcomm_Snapdragon_processors)

# Edge AI Hardware

## Apple Neural Engine

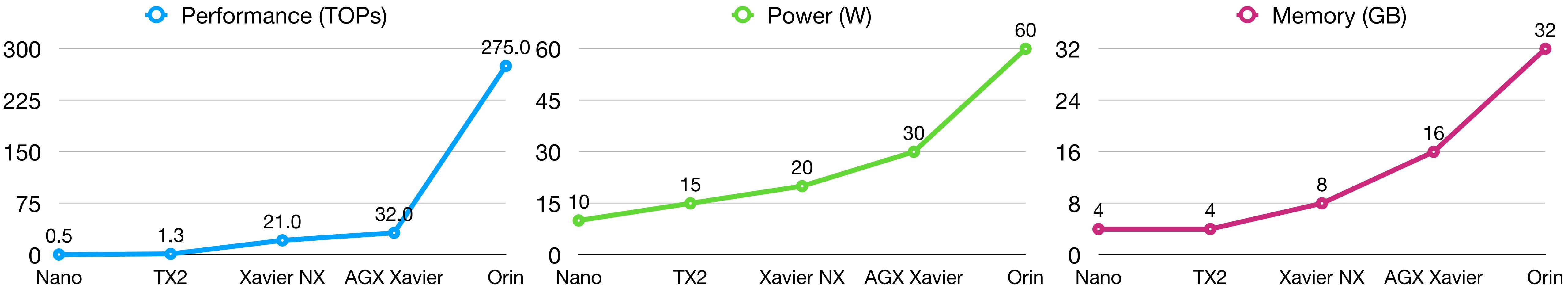
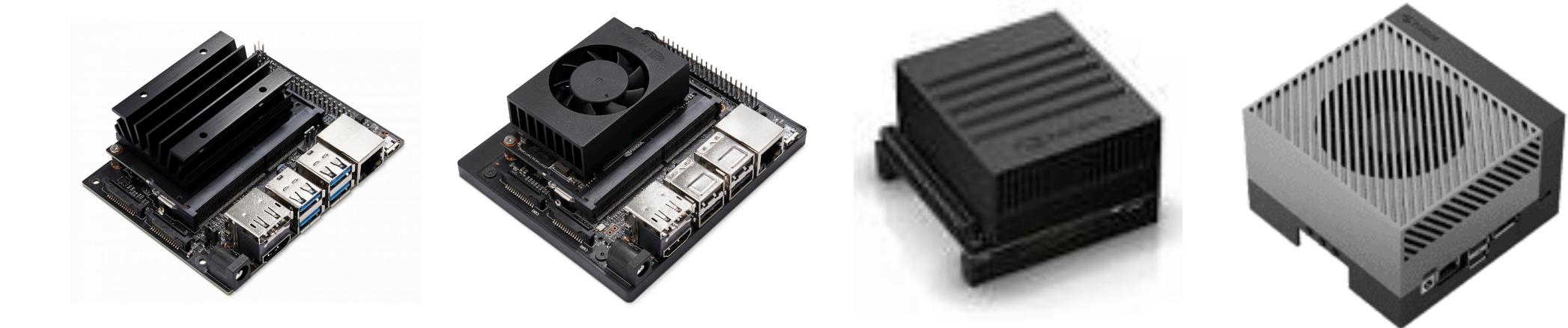
- The Apple Neural Engine (ANE) is an energy-efficient and high-throughput engine for ML inference on Apple silicon.



# Edge AI Hardware

## Nvidia Jetson

- NVIDIA Jetson is a complete System on Module (SOM) that includes a GPU, CPU, memory, power management, high-speed interfaces, and more.

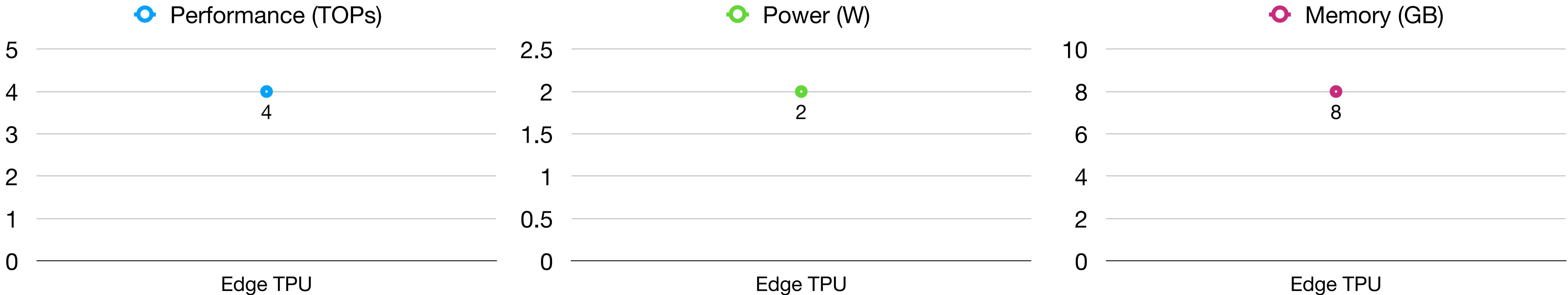


NanoReview. <https://connecttech.com/jetson/jetson-module-comparison/>

# Edge AI Hardware

## Tensor Processing Unit

- Tensor Processing Unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google for neural network machine learning, using Google's own TensorFlow software.

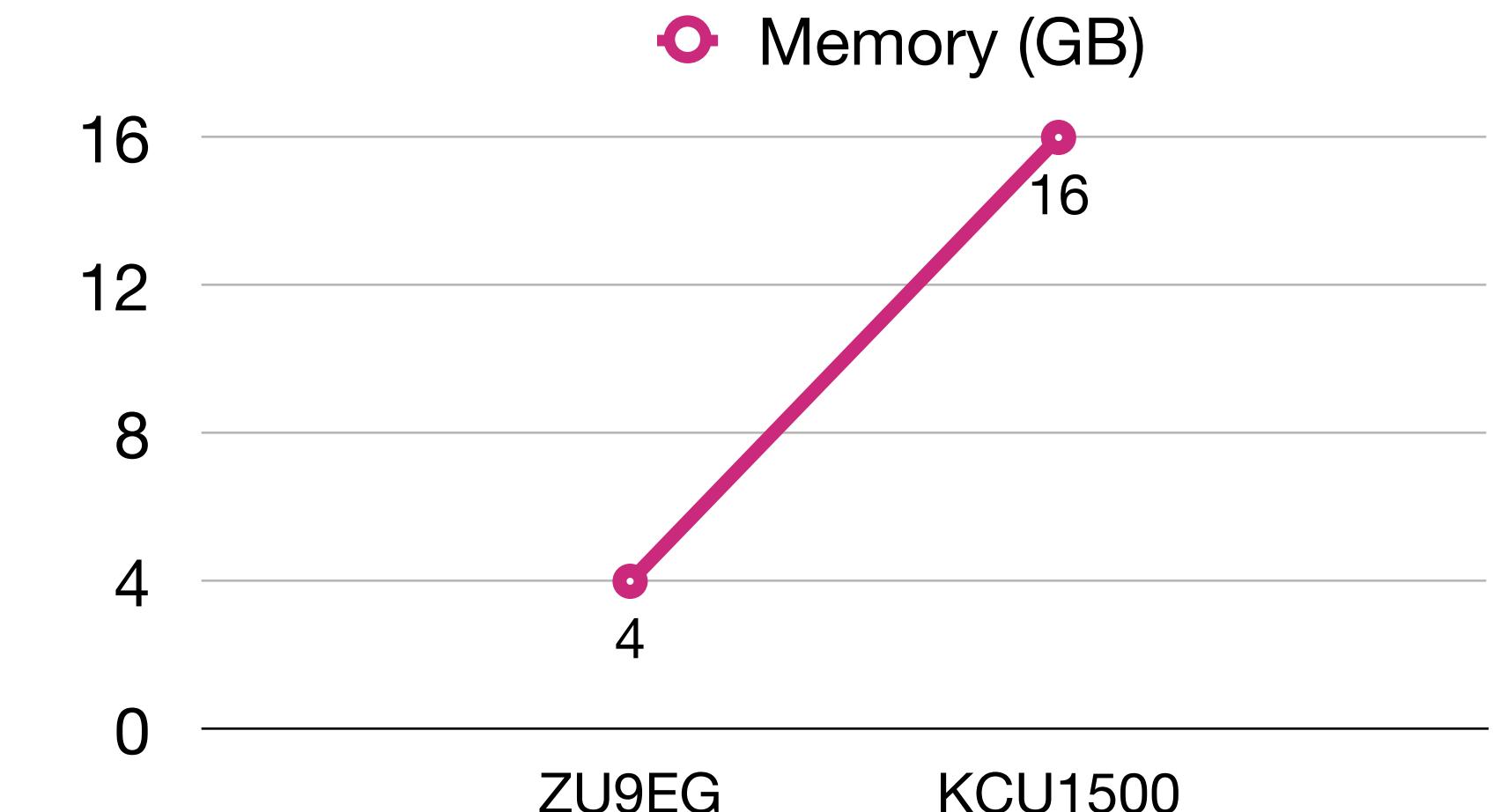
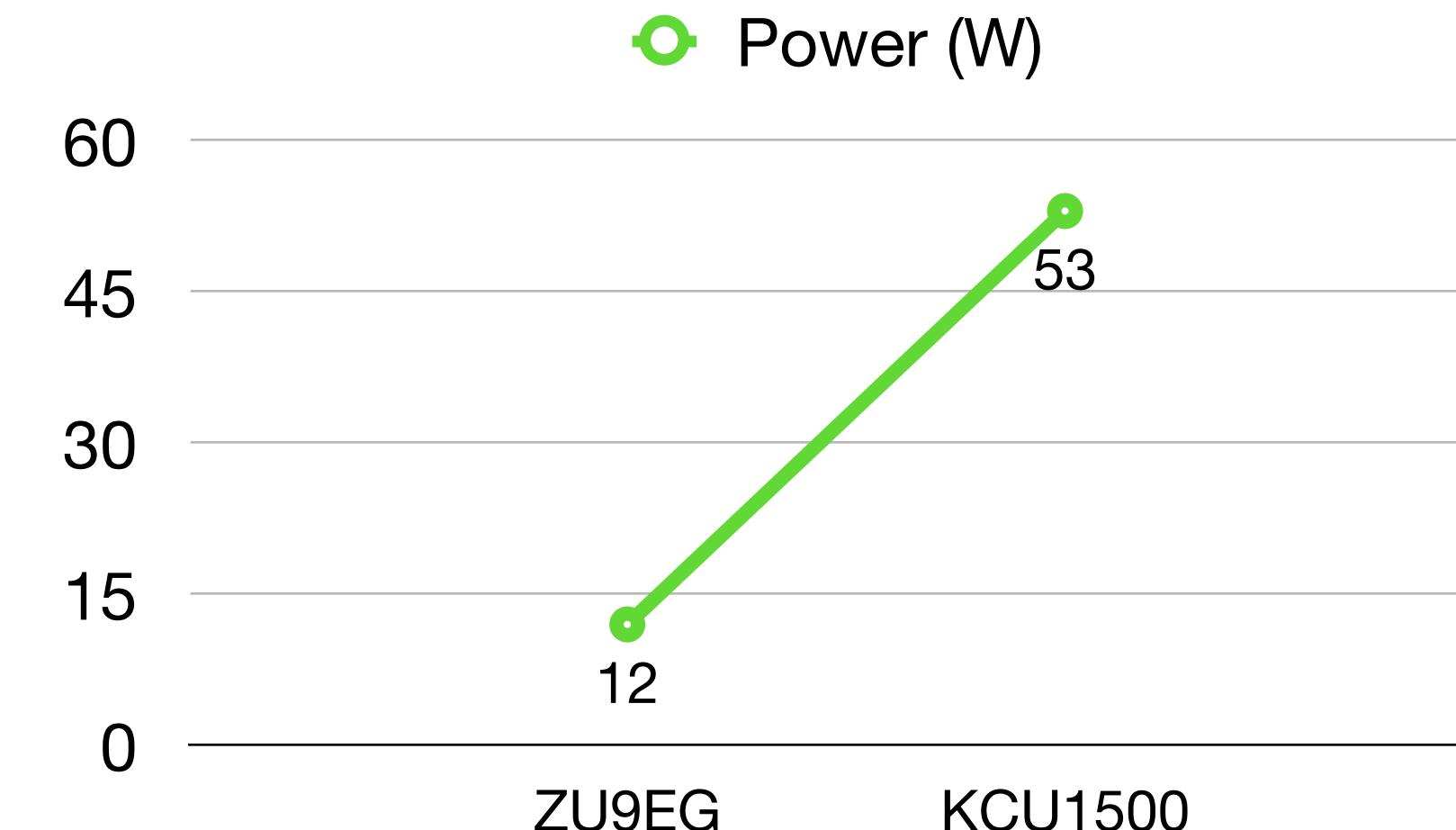
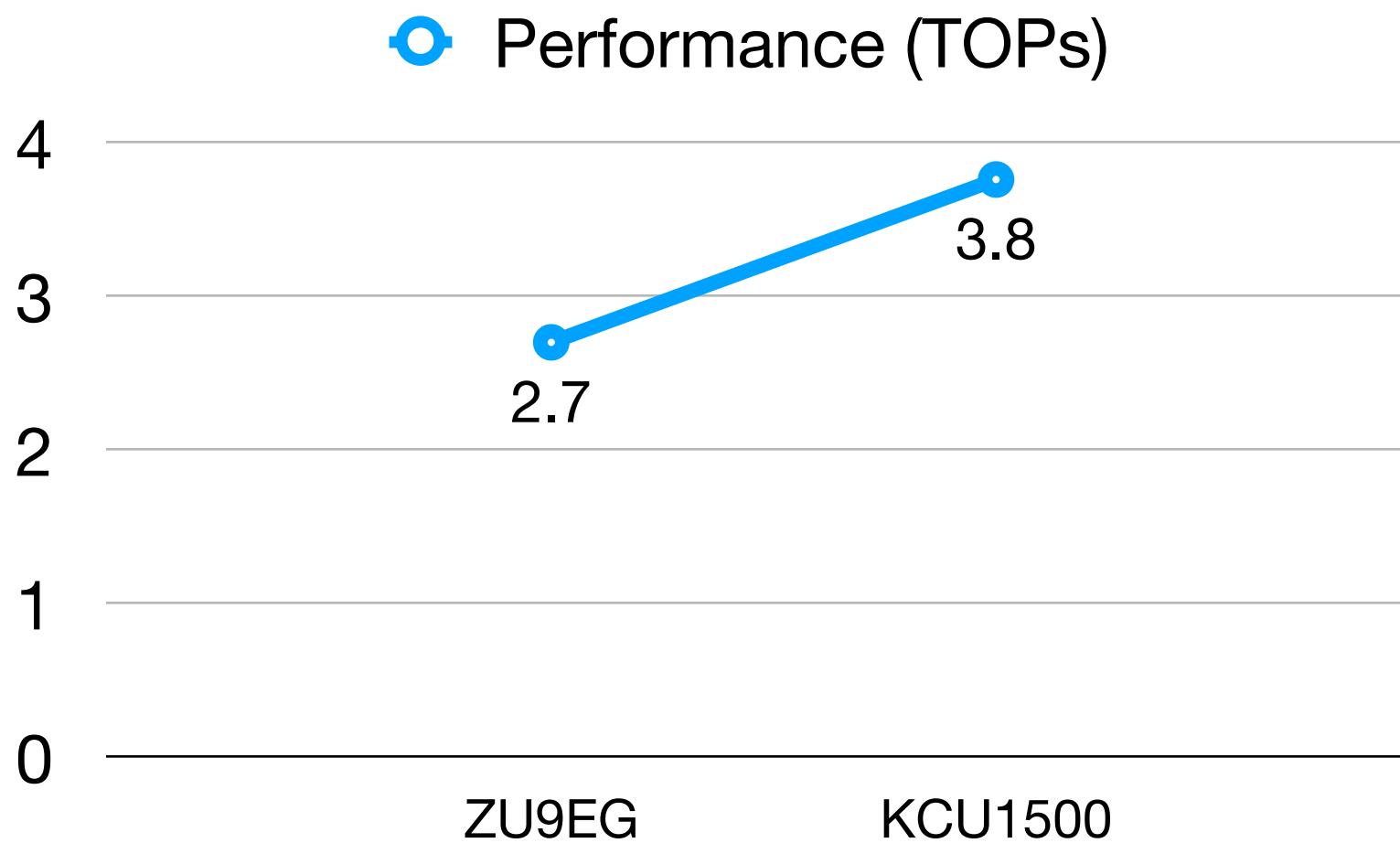


Tensor Processing Unit. [https://en.wikipedia.org/wiki/Tensor\\_Processing\\_Unit](https://en.wikipedia.org/wiki/Tensor_Processing_Unit)

# Edge AI Hardware

## FPGA-based Accelerators

- Field Programmable Gate Arrays (FPGA) delivers higher performance compared to a fixed-architecture AI accelerator like a GPU due to efficiency of custom hardware acceleration.

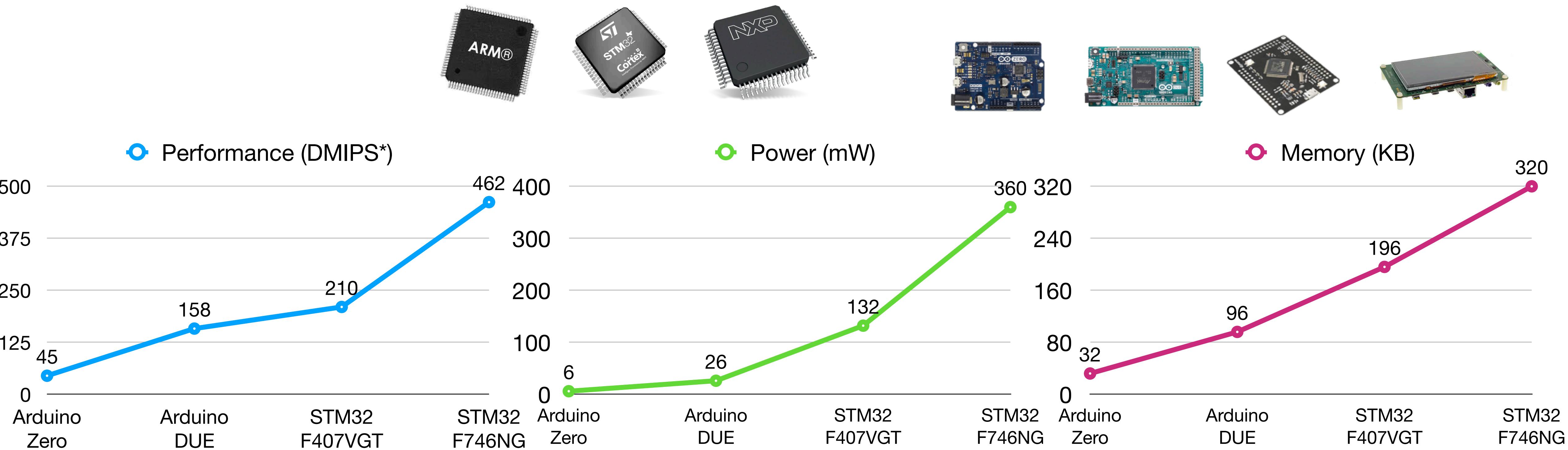


Neural Network Accelerator Comparison. <https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator.html>

# Edge AI Hardware

## Microcontrollers (MCU)

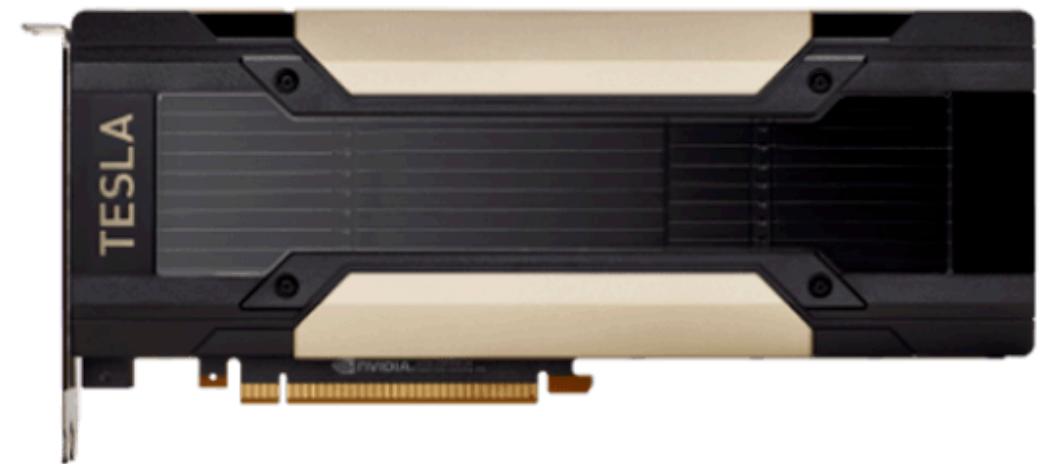
- A microcontroller is a compact integrated circuit designed for embedded systems. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.



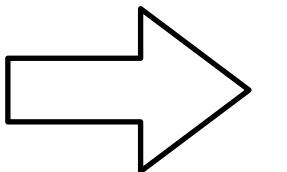
\* Dhrystone Million Instructions Per Second (DMIPs) is an index for integer computation.

# Edge AI Hardware

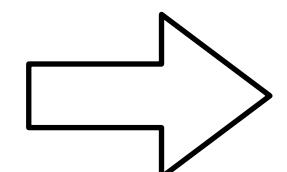
Edge AI devices still have huge gap to cloud processors



Cloud AI



Mobile AI



Tiny AI

---

Memory (Activation)

32GB

4GB

320kB

---

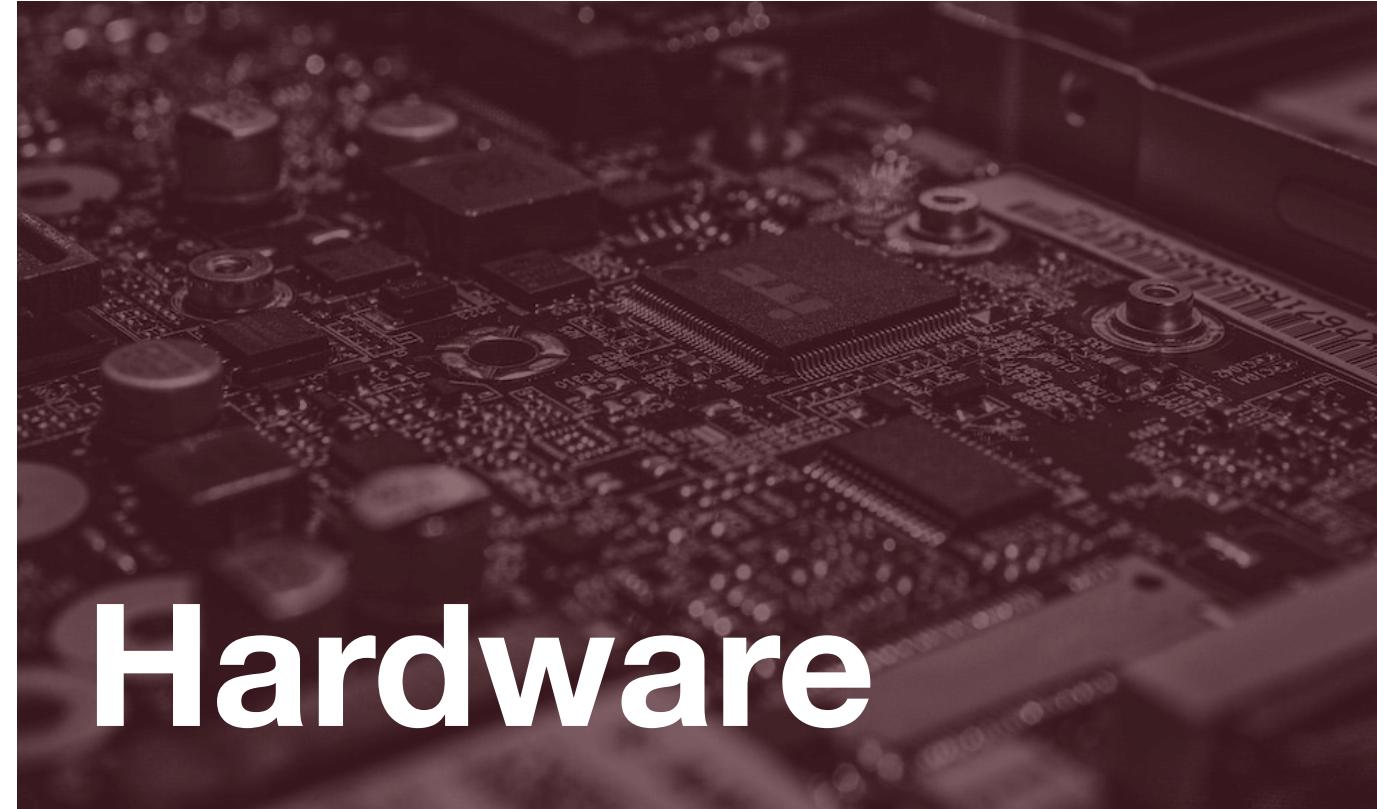
Storage (Weights)

~TB/PB

256GB

1MB

# Deep Learning: Three Pillars



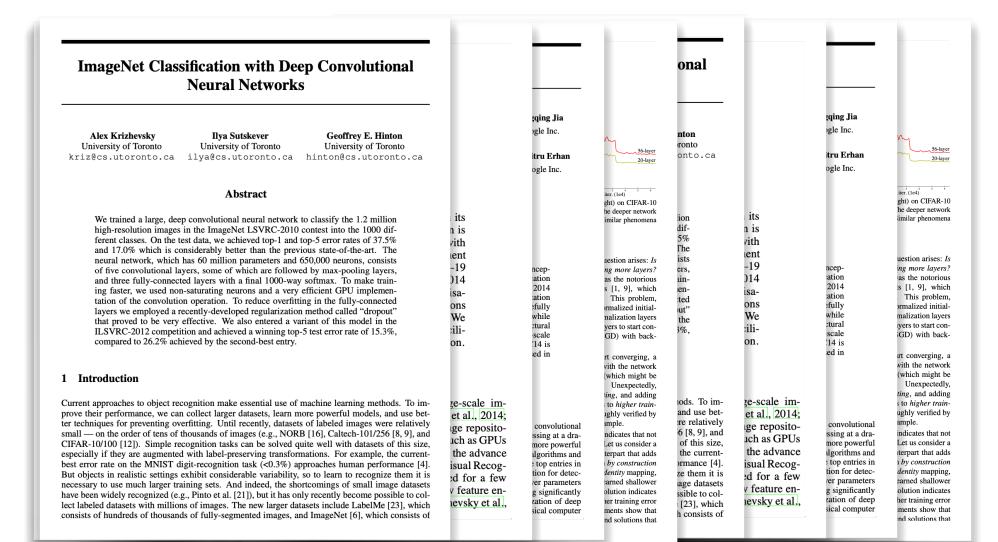
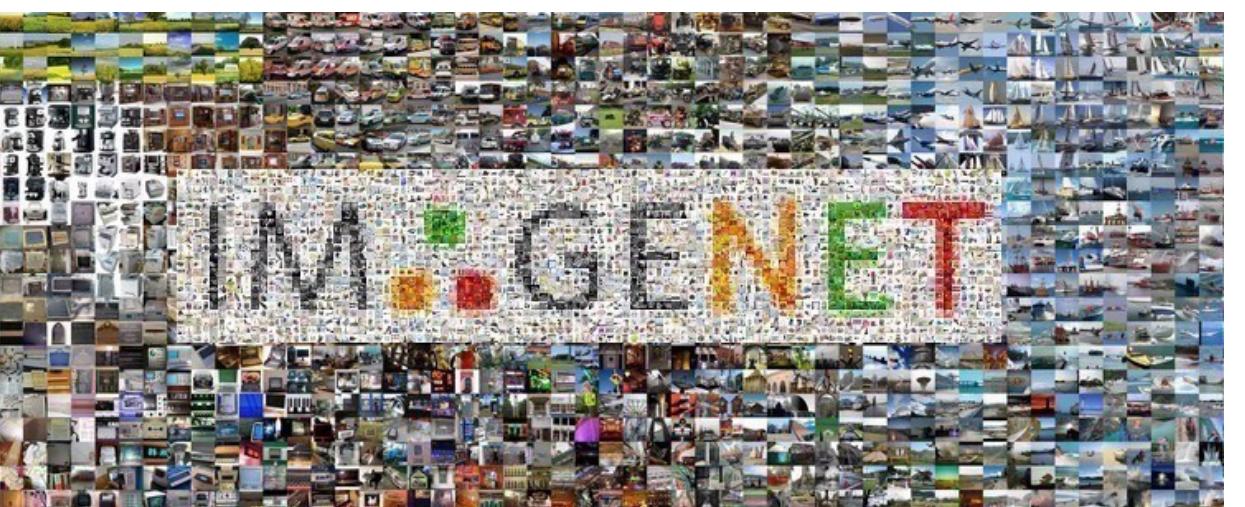
Hardware



Data



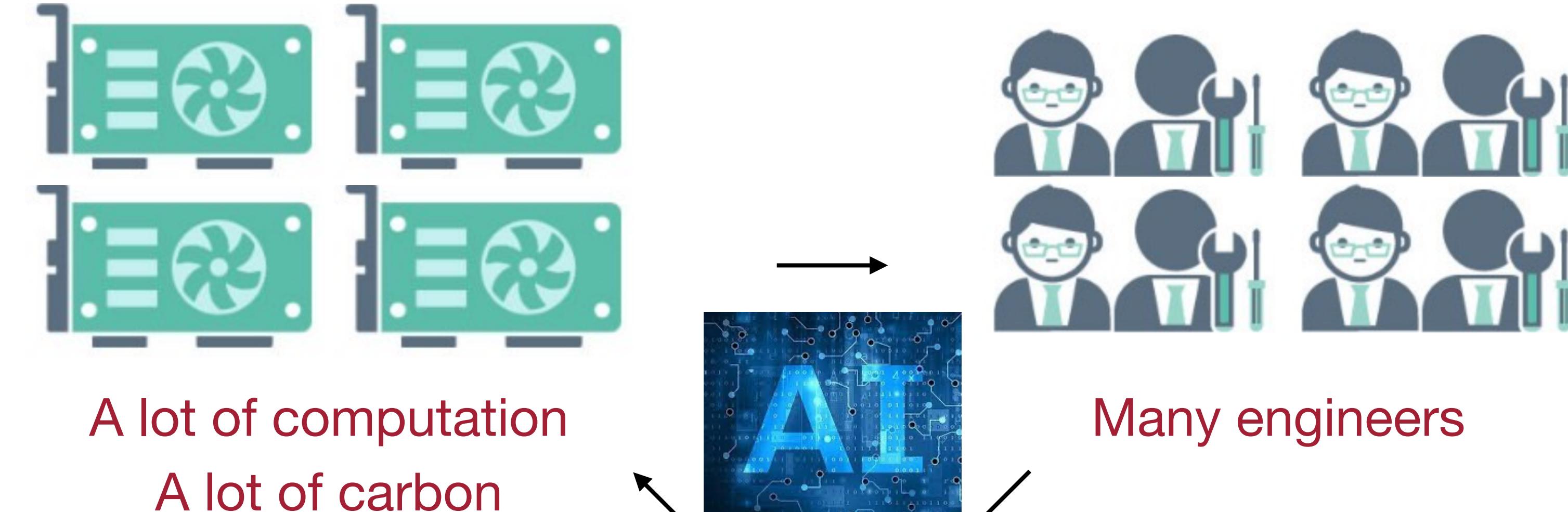
Algorithm



ImageNet Classification with Deep Convolutional Neural Networks

# Demand for Efficient Deep Learning

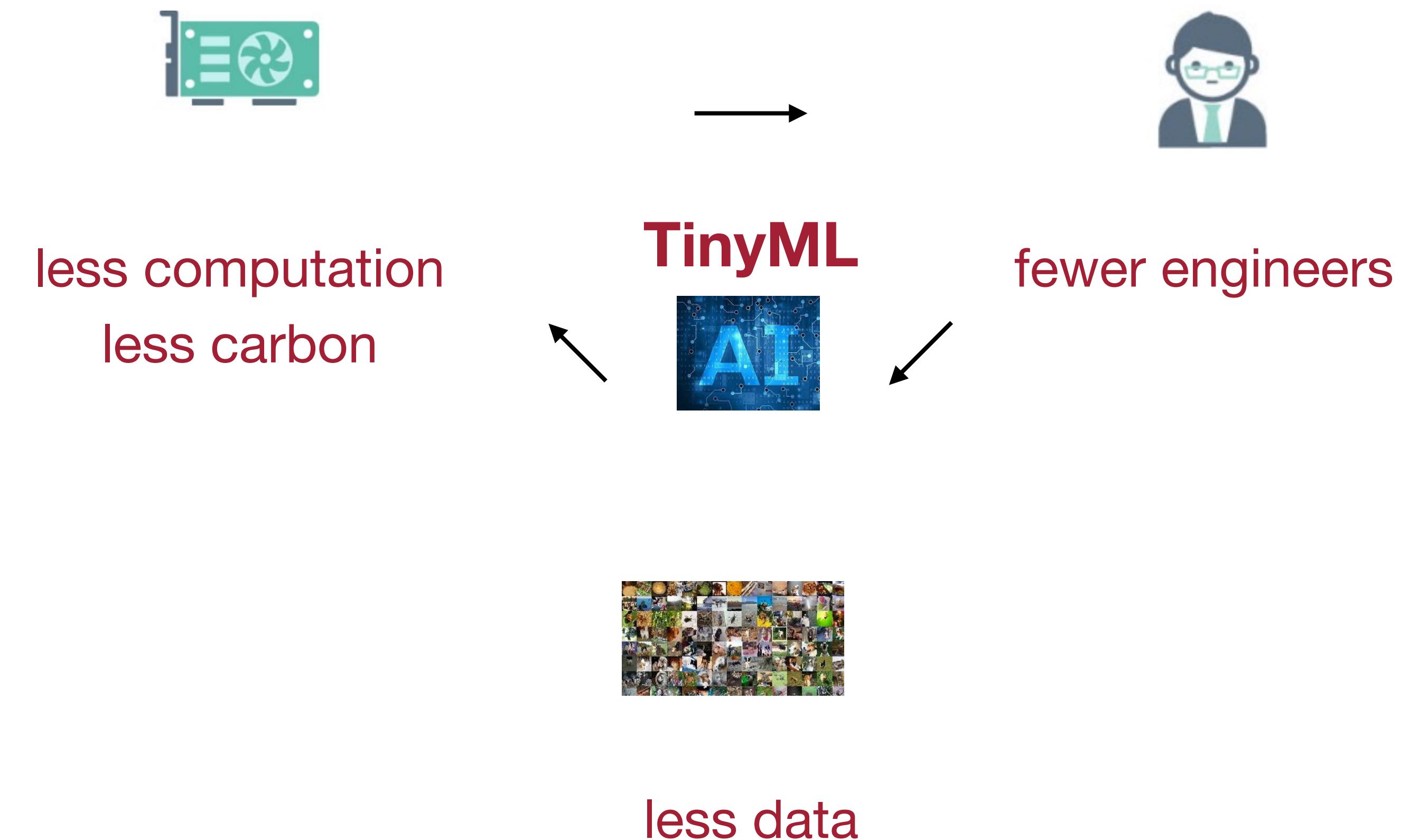
Big computation, engineer and data



A lot of data

# Demand for Efficient Deep Learning

Tiny computation, engineer and data



# Course Overview

# Course Outline

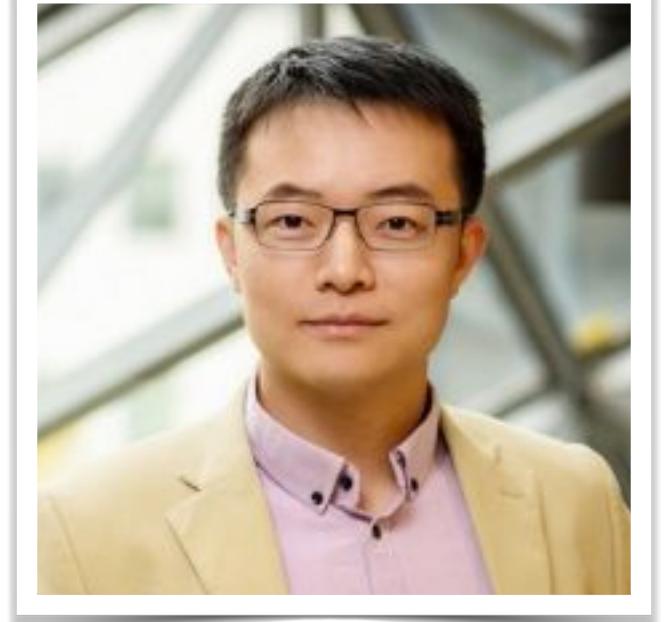
- **Basics of Deep Neural Networks (DNNs)**
- **Efficient Inference of Neural Networks**
  - Pruning, Quantization, Neural Architecture Search, Knowledge Distillation
- **Efficient Training of Neural Networks**
  - Distributed Training, Gradient Compression, On-Device Learning, Transfer Learning
- **System Support for Efficient Deep Learning Computing**
- **Application-Specific Optimization of Deep Learning Computing**
  - Point Cloud Recognition, Video Understanding, GANs, Transformers
- **Quantum Machine Learning**

# Takeaways

- Know the key efficiency metrics for deep learning computing
- Be familiar with how to speed up inference and training of neural networks on resource-constraint platforms
- Understand the tradeoffs between different optimization techniques
- Be aware of recent research trends and industry practice
- Get hands-on experience implementing deep learning applications on microcontrollers, mobile phones, and quantum machines with an open-ended design project related to mobile AI
- Internship opportunities

# Course Staff and Contact Info

- **Instructors (office hours by request)**
  - Song Han ([songhan@mit.edu](mailto:songhan@mit.edu))
- **Teaching Assistants (office hours & location announced on course website)**
  - Zhijian Liu ([zhijian@mit.edu](mailto:zhijian@mit.edu))
  - Yujun Lin ([yujunlin@mit.edu](mailto:yujunlin@mit.edu))
- **Lecture Schedule**
  - TR 3:30 PM - 5:00 PM, Room 36-153
- **Course Website**
  - <https://efficientml.ai>
- **Video**
  - <https://live.efficientml.ai>



6.S965 Logistics Schedule

## TinyML and Efficient Deep Learning

6.S965 • Fall 2022 • MIT

Have you found it difficult to deploy neural networks on mobile devices and IoT devices? Have you ever found it too slow to train neural networks? This course is a deep dive into efficient machine learning techniques that enable powerful deep learning applications on resource-constrained devices. Topics cover efficient inference techniques, including model compression, pruning, quantization, neural architecture search, and distillation; and efficient training techniques, including gradient compression and on-device transfer learning; followed by application-specific model optimization techniques for videos, point cloud, and NLP; and efficient quantum machine learning. Students will get hands-on experience implementing deep learning applications on microcontrollers, mobile phones, and quantum machines with an open-ended design project related to mobile AI.

---

- **Time:** Tuesday/Thursday 3:30-5:00 pm
- **Location:** [36-153](#)
- **Discussion:** [Piazza](#)
- **Homework submission:** [Canvas](#)
- **Online lectures:** The lectures will be recorded and made available on [YouTube](#).
- **Contact:** Students should ask all course-related questions on [Piazza](#). For external inquiries, personal matters, or emergencies, you can email us at [6s965-fall2022-staff@mit.edu](mailto:6s965-fall2022-staff@mit.edu).

# Course Requirements and Materials

- **Pre-requisites**
  - 6.1910[6.004] (Computation Structures)
  - 6.3900[6.036] (Introduction to Machine Learning)
- **We will use Python, PyTorch and C/C++**
  - PyTorch website: <https://pytorch.org/>
- **Course Textbook/Readings (course website)**
  - Selected papers published in past 3 to 5 years
- **Course Handouts (course website)**
- **Homework submission (on Canvas)**

# Related Classes

- **Advances in Computer Vision** (6.8300[6.869] / 6.8301[6.819])
- **Machine Learning** (6.7900[6.867])
- **Hardware Architecture for Deep Learning** (6.5930[6.825] / 6.5931[6.812])

# Assignments and Grading

- **Grading**
  - Scribe Duties: 5%
  - 4 Labs: 60% (15% each)
  - Paper Reading: 10%
  - Final Project: 25%
- All assignments are due by **11:59PM on the due date** (submitted online)
- **Late Policy for Labs**
  - *You will be allowed 6 total homework late days without penalty for the entire semester.*
  - The allowed late days are counted by day (i.e., each new late day starts at 12:00 am ET).
  - Once the allowed late days are exceeded, the penalty is 50% per late day.
  - The homework is worth zero credit 2 days after exceeding the late day limit.

# Labs

**60% of grade (15% each)**

- Lab 0 - Getting Started (Due Sep 15, not counted to the final grades)
- Lab 1 - Pruning (Due Oct 4)
- Lab 2 - Quantization (Due Oct 18)
- Lab 3 - Neural Architecture Search (Due Nov 1)
- Lab 4 - Deploy TinyML Models on Microcontrollers (Due Nov 17)

# Scribe Duties

## 5% of grade

- Each student is required to scribe for a few of lectures.
- During your assigned lectures, take detailed notes in collaboration with your fellow scribes. After the lecture, the scribe team converts the notes into markdown.
- TAs will release an example scribe by the next lecture.
- As long as your scribe notes are complete and accurate, you will be awarded full credit.

# Paper Review Presentation

## 10% of grade

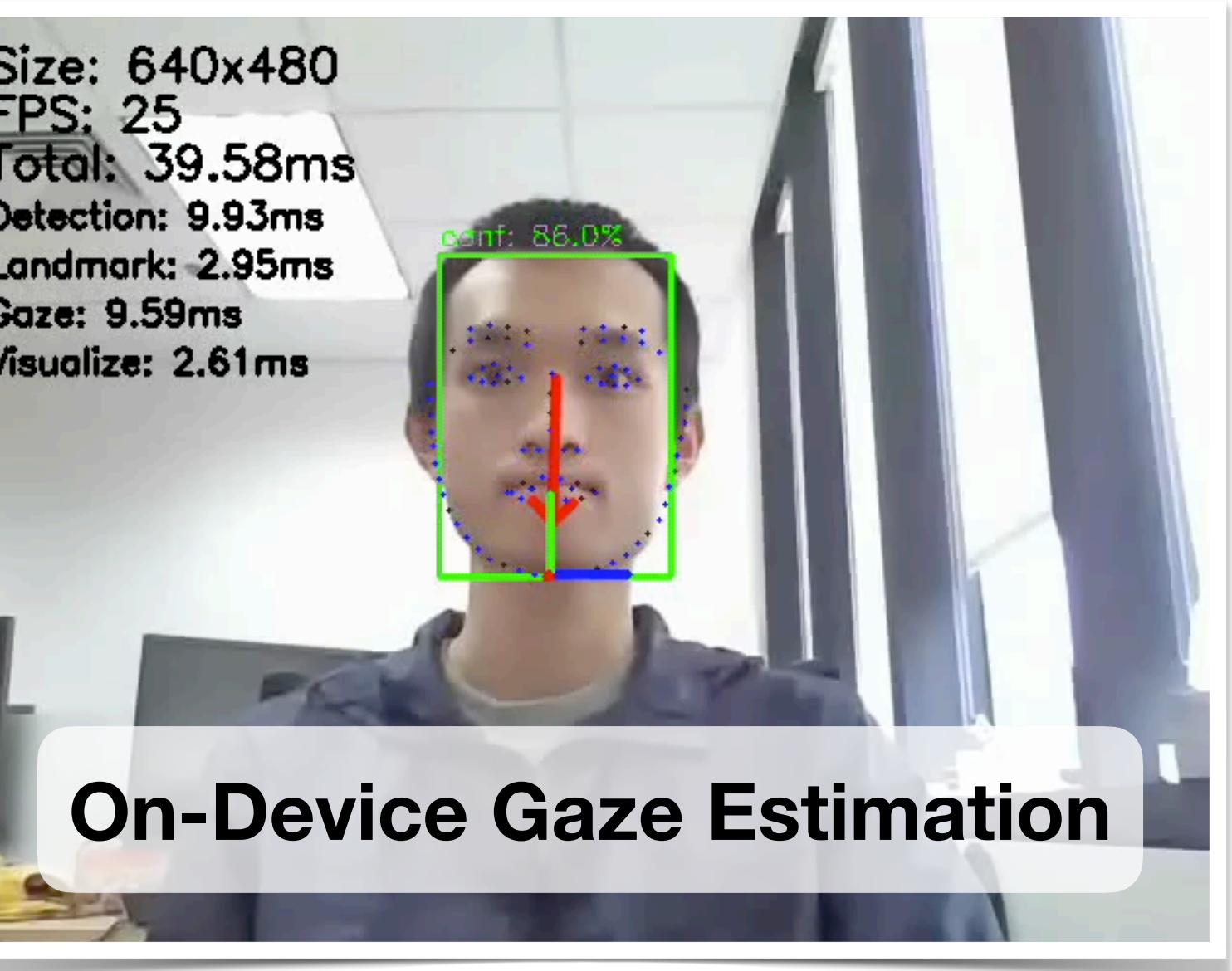
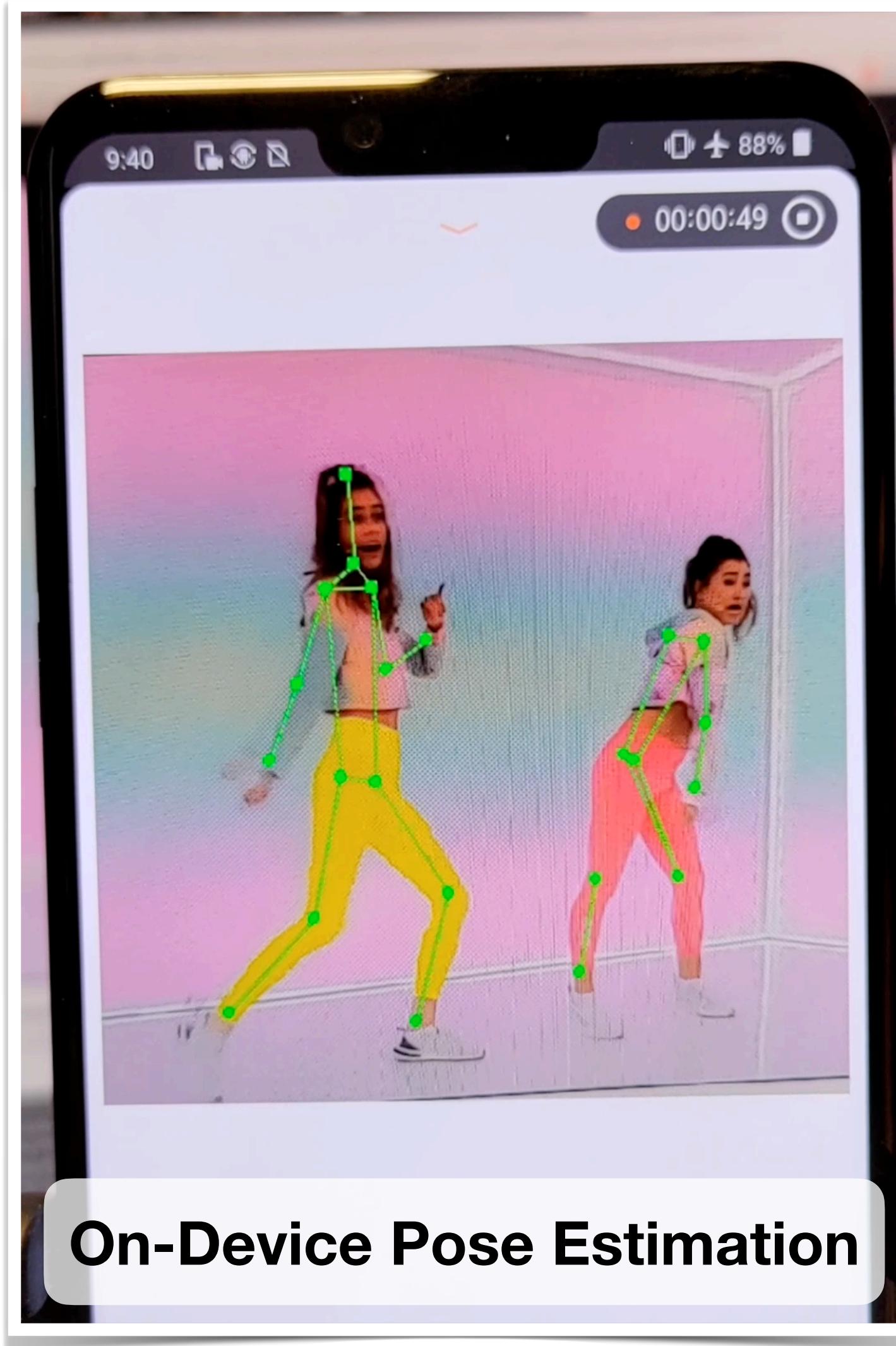
- Every paper will have 2-3 students acting as reviewer team.
- Each team is required to present the paper in the class:
  - The team will give overview of paper, including background, contributions, methods, and key evaluation results. (5 - 7 min)
  - Each student will give strength/weakness of paper. (1.5 min for each student)
  - The team will answer questions from other students in the class.
- Papers will be assigned by Oct 6.
- Paper Review Presentation is on Oct 20.

# Final Project

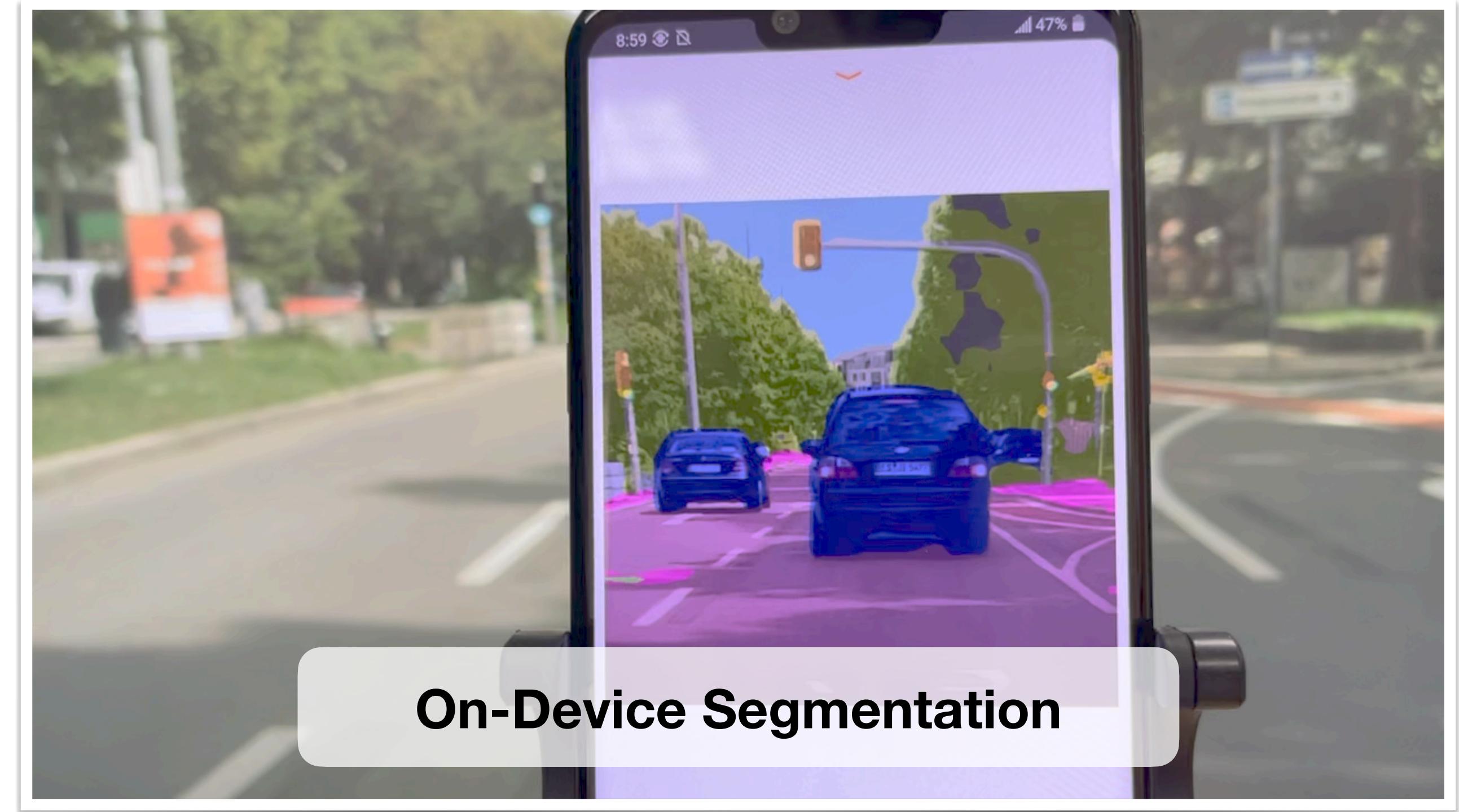
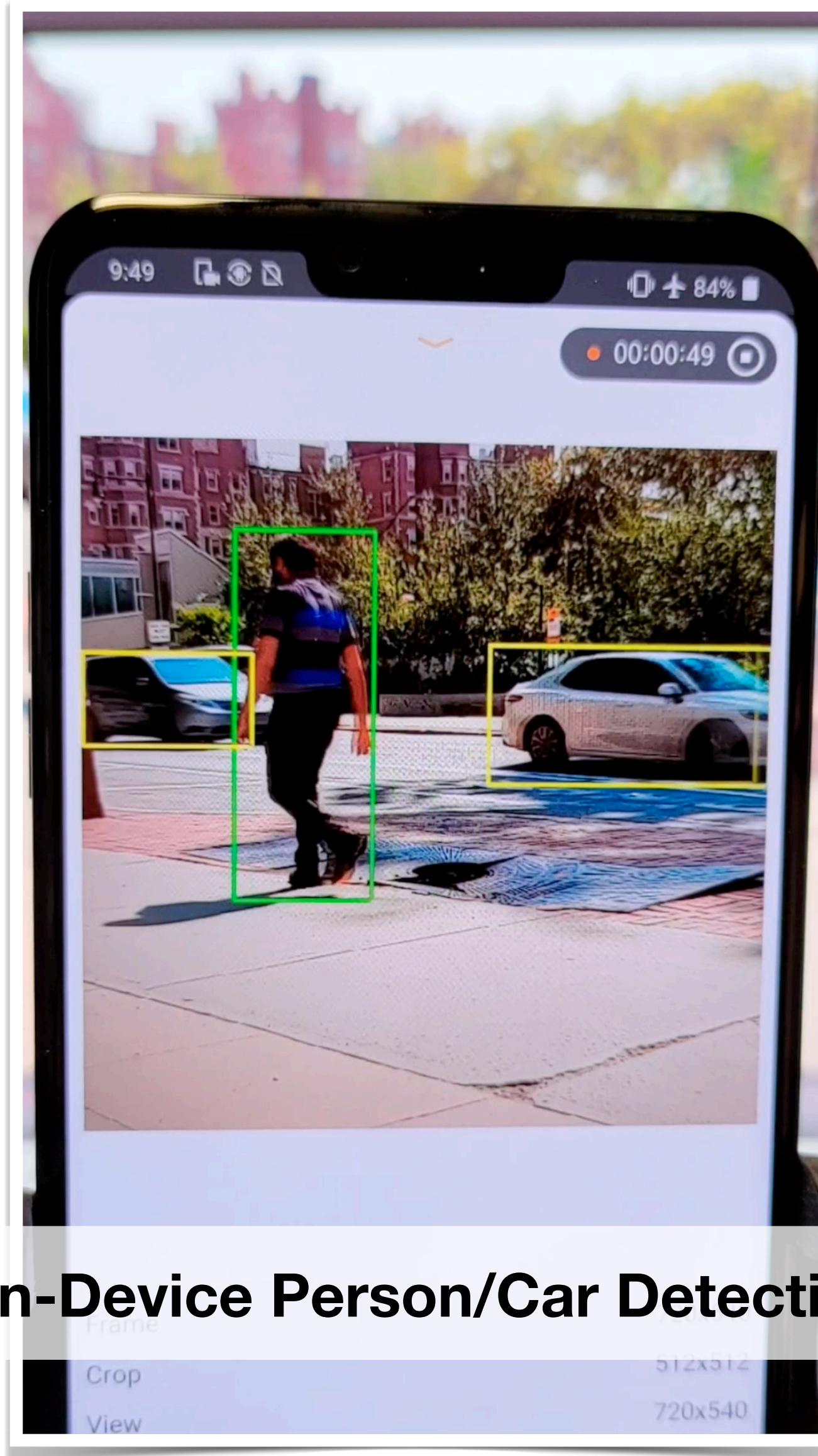
**25% of grade**

- **Project**
  - Choose from a list of suggested projects, or:
  - Propose your own project (requires formal proposal and approval by course staff)
- **You're welcome to use tools from labs**
  - PyTorch, TinyEngine
- **Teams of two or three**
- **Schedule**
  - Oct 6 - List of projects released
  - Oct 25 - Submit project selection (or proposal)
  - Dec 6 and Dec 8 - Project Presentation
  - Dec 13 - Project Report Due

# Example Final Projects



# Example Final Projects



# Encourage Feedback

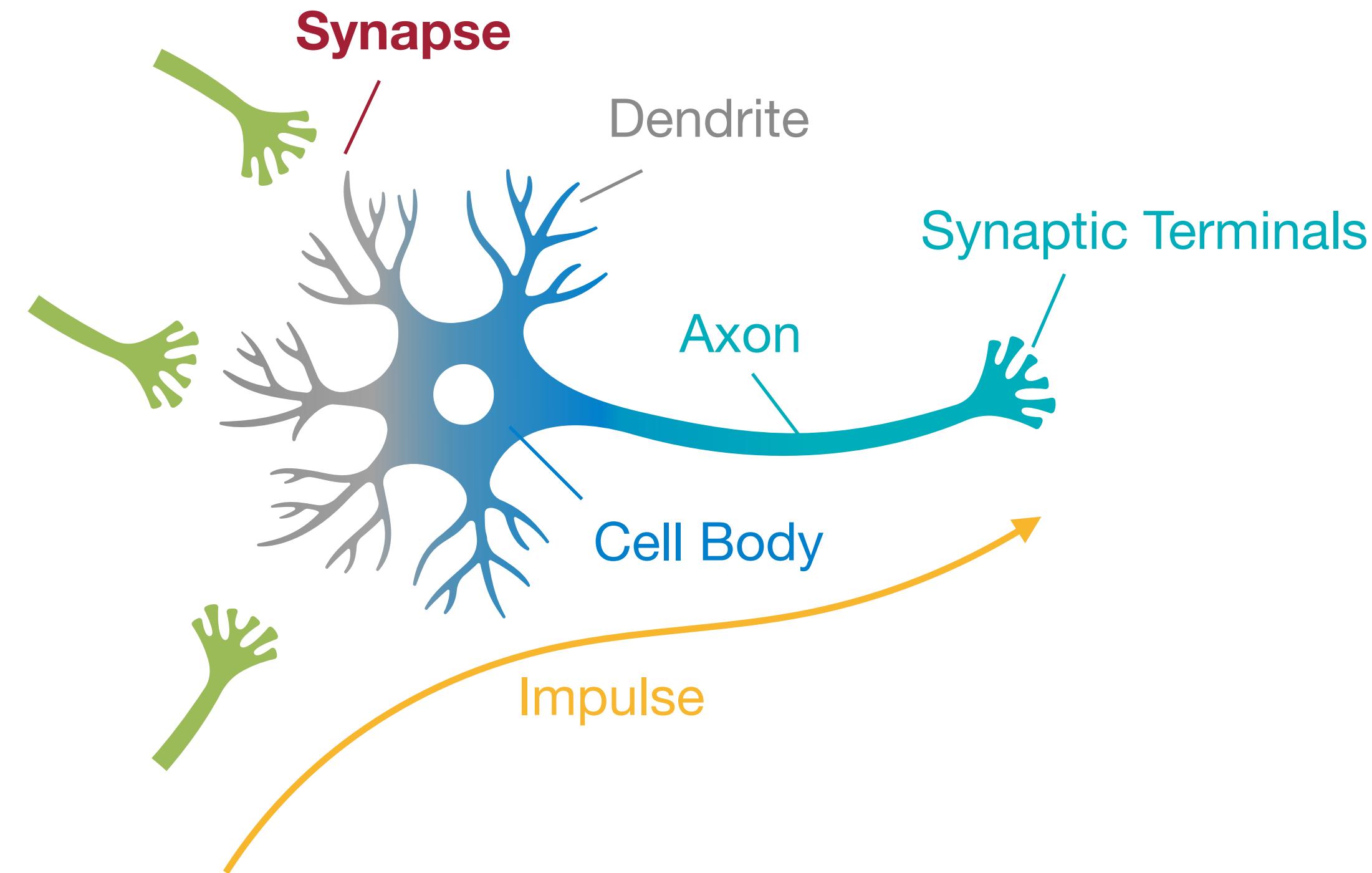
We appreciate everyone being actively involved in the class!

- Participation bonus credit will be capped at 4%:
  - **Piazza participation**
    - The top ~10 contributors to Piazza will get 3%
  - **Completing mid-semester evaluation**
    - Around the middle of the semester, we will send out a survey to help us understand how the course is going, and how we can improve. Completing it is worth 1%.
  - **Karma point**
    - Any other act that improves the class, which a TA or instructor notices and deems worthy: 1%.

# What is Deep Learning

# How Does the Brain Work?

## Neuron and synapse

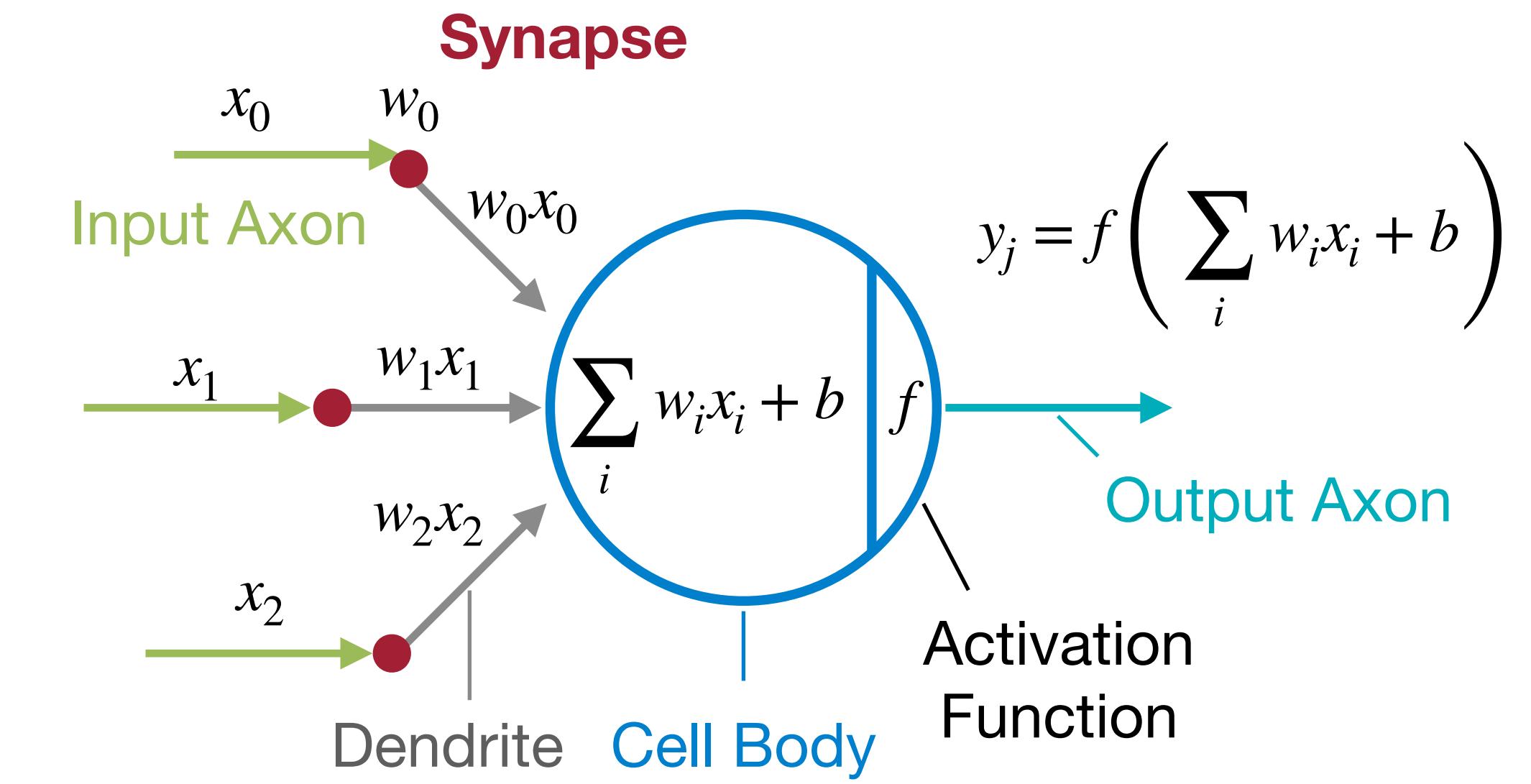
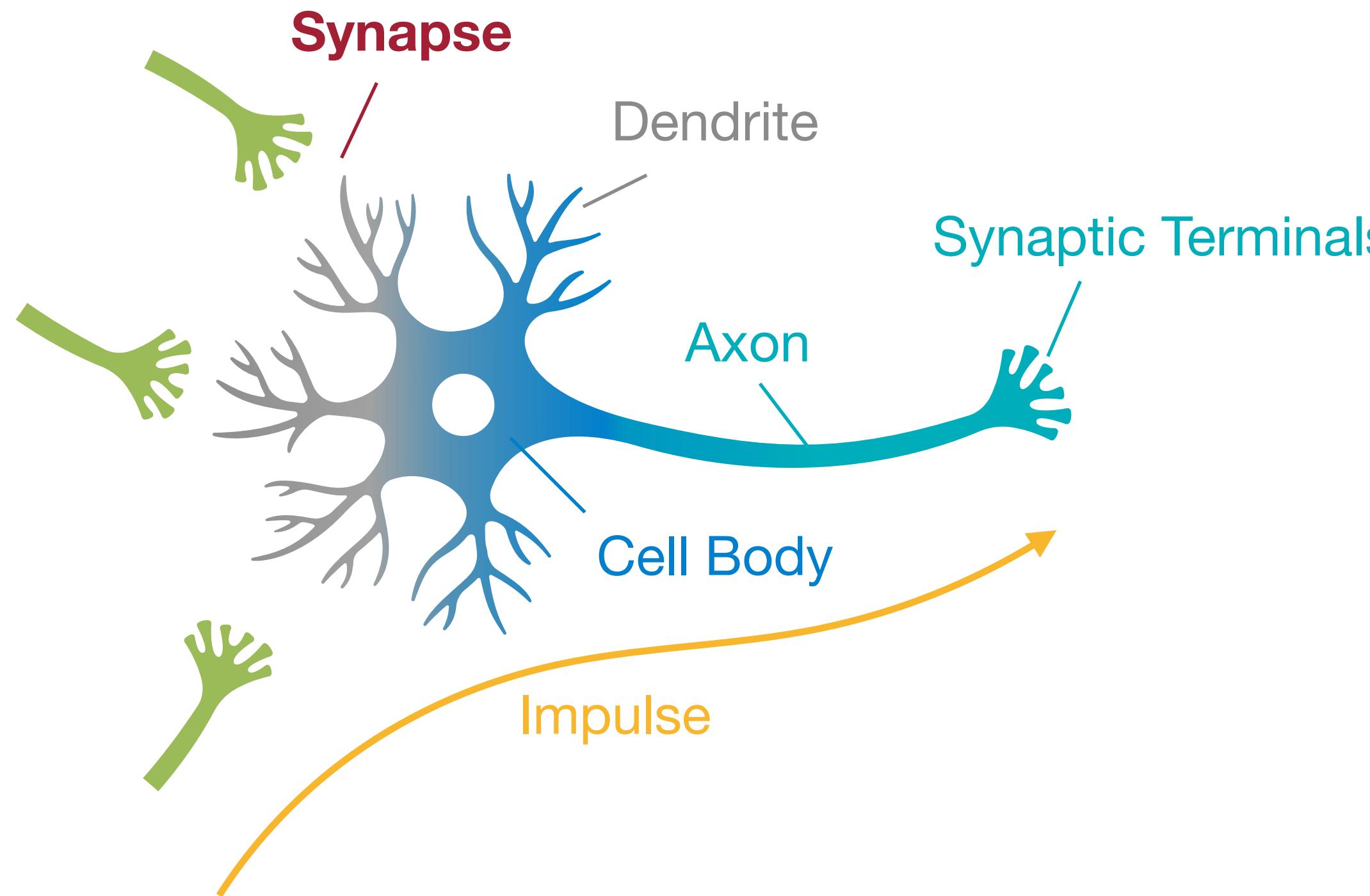


- The basic computational unit is the **neuron**.
  - 86B neurons in human brain
- A neuron receives *input* from other neurons through **dendrite**.
- Once input exceeds a critical level, the neuron discharges a spike - an electrical pulse traveling from the cell body down the **axon** towards the next neuron(s).
- Neurotransmitters, which are chemicals, are released from the first neuron to the second via **synapse**.
  - Each neuron has on average 7000 synapses.
- **Synaptic weights** control the extent to which the signal from one neuron is passed on to the other.

Slide Inspiration: [Nic Schraudolph and Fred Cummins](#)

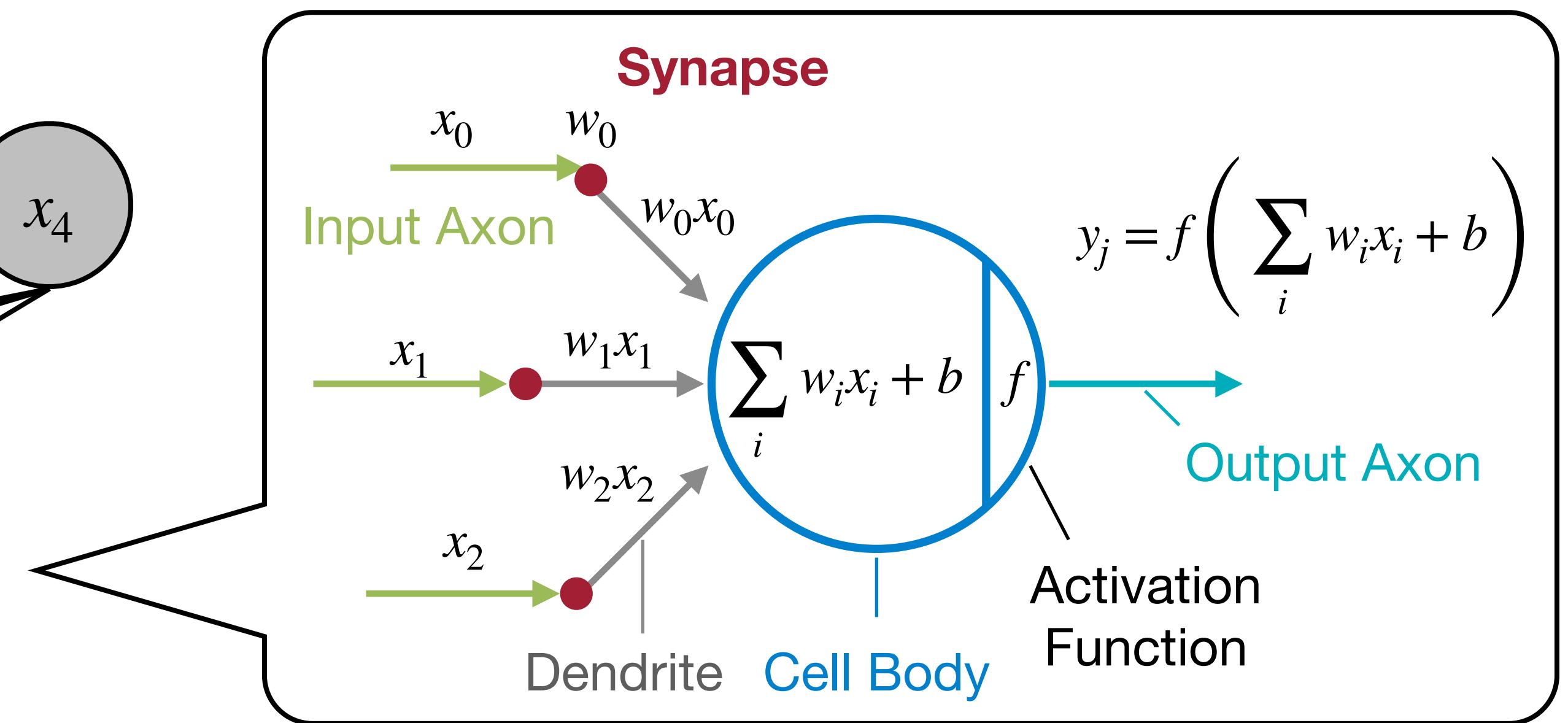
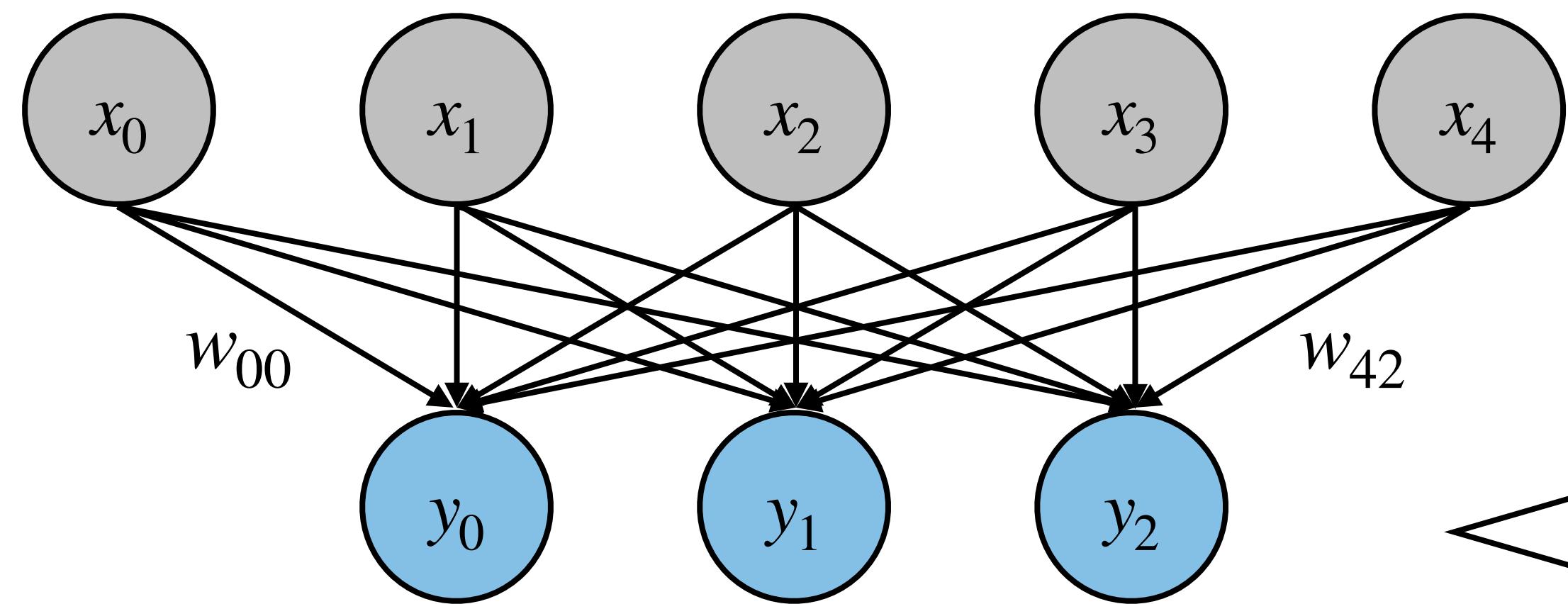
# Neuron and Synapse

From neuroscience to computer science

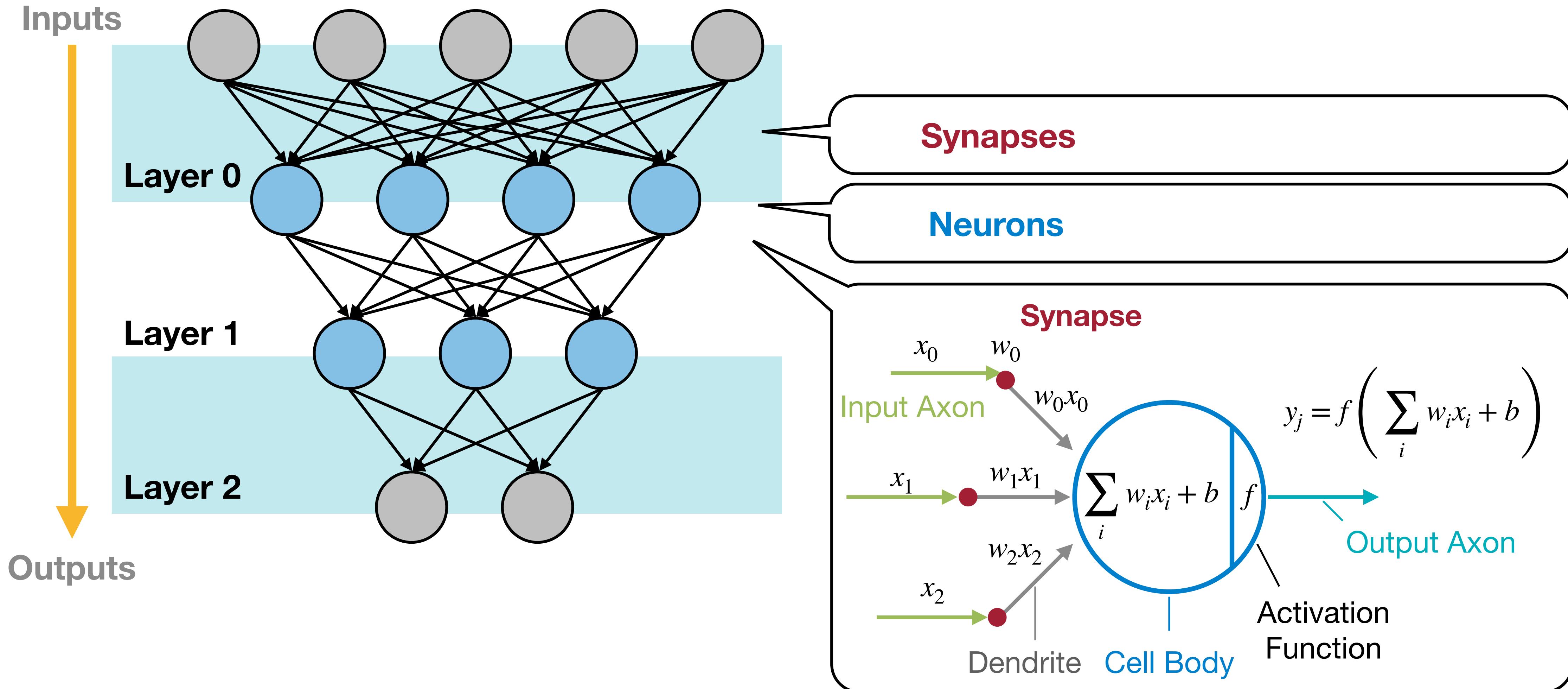


Slide Inspiration: [Ruohan Gao](#)

# Neural Network

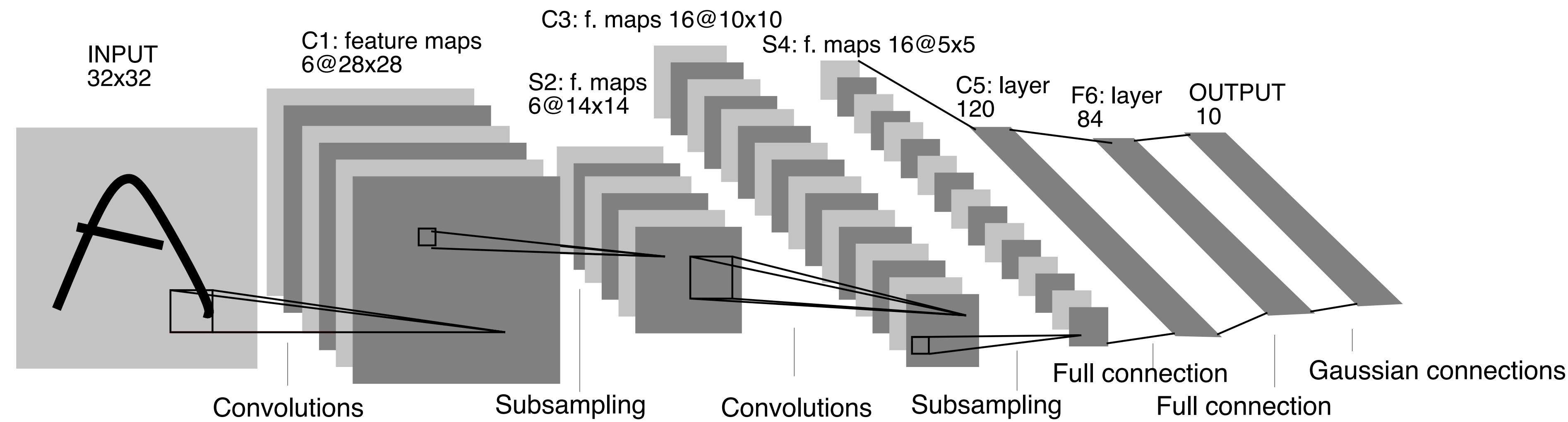


# Deep Neural Network



# Deep Neural Network

Neural Networks become deeper, larger, and more complicated.

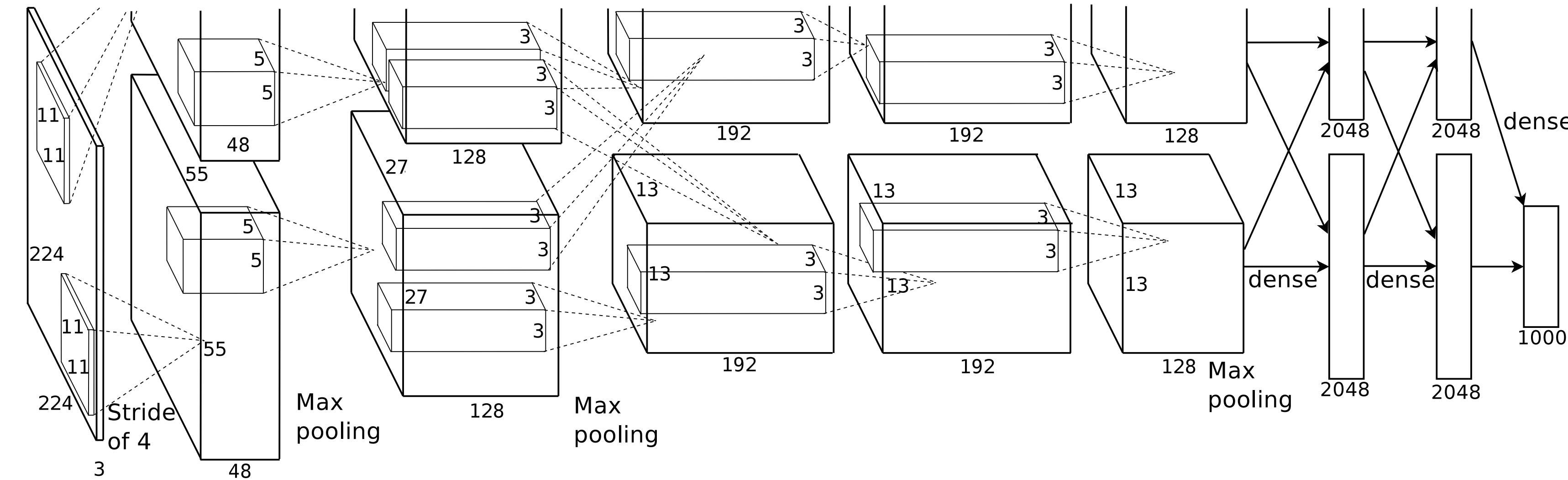


LeNet-5

Gradient-Based Learning Applied to Document Recognition [LeCun et al., the IEEE 1998]

# Deep Neural Network

Neural Networks become deeper, larger, and more complicated.



AlexNet

ImageNet Classification with Deep Convolutional Neural Networks [Krizhevsky et al., NeurIPS 2012]

# Deep Neural Network

Neural Networks become deeper, larger, and more complicated.

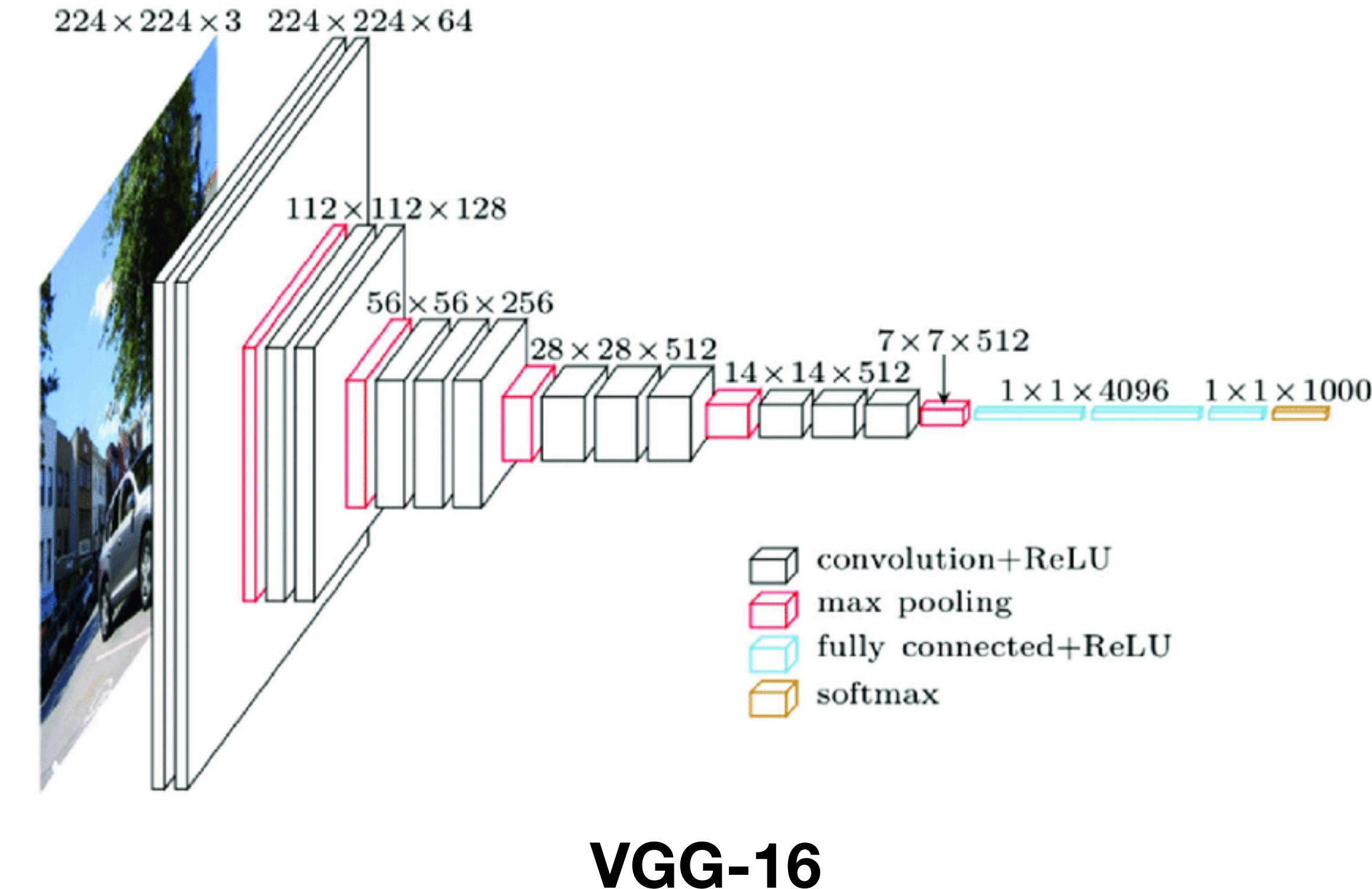
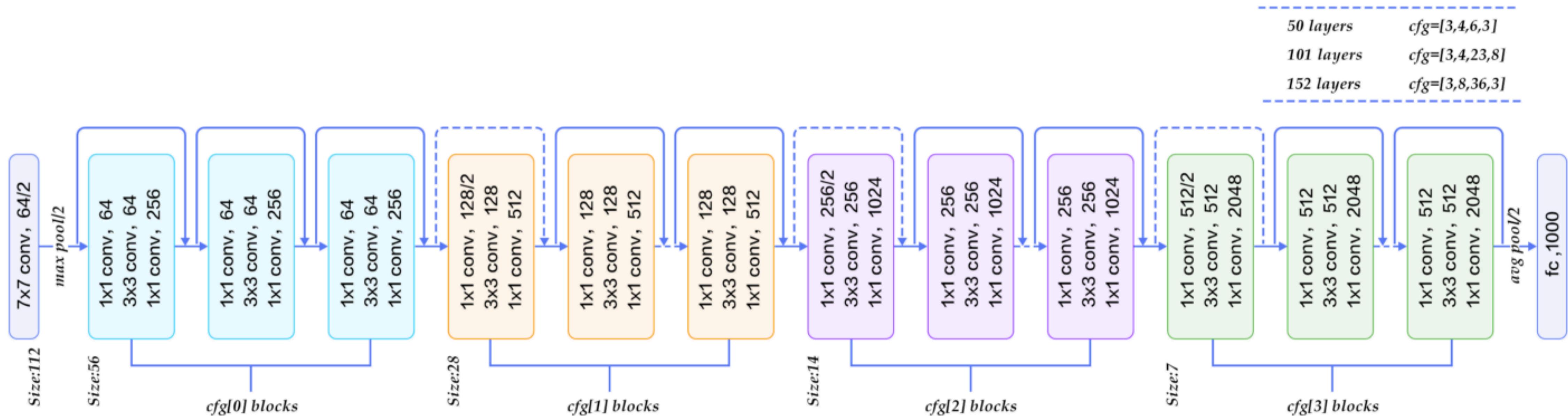


Image Source: 1

Very Deep Convolutional Networks for Large-Scale Image Recognition [Simonyan et al., ICLR 2015]

# Deep Neural Network

Neural Networks become deeper, larger, and more complicated.



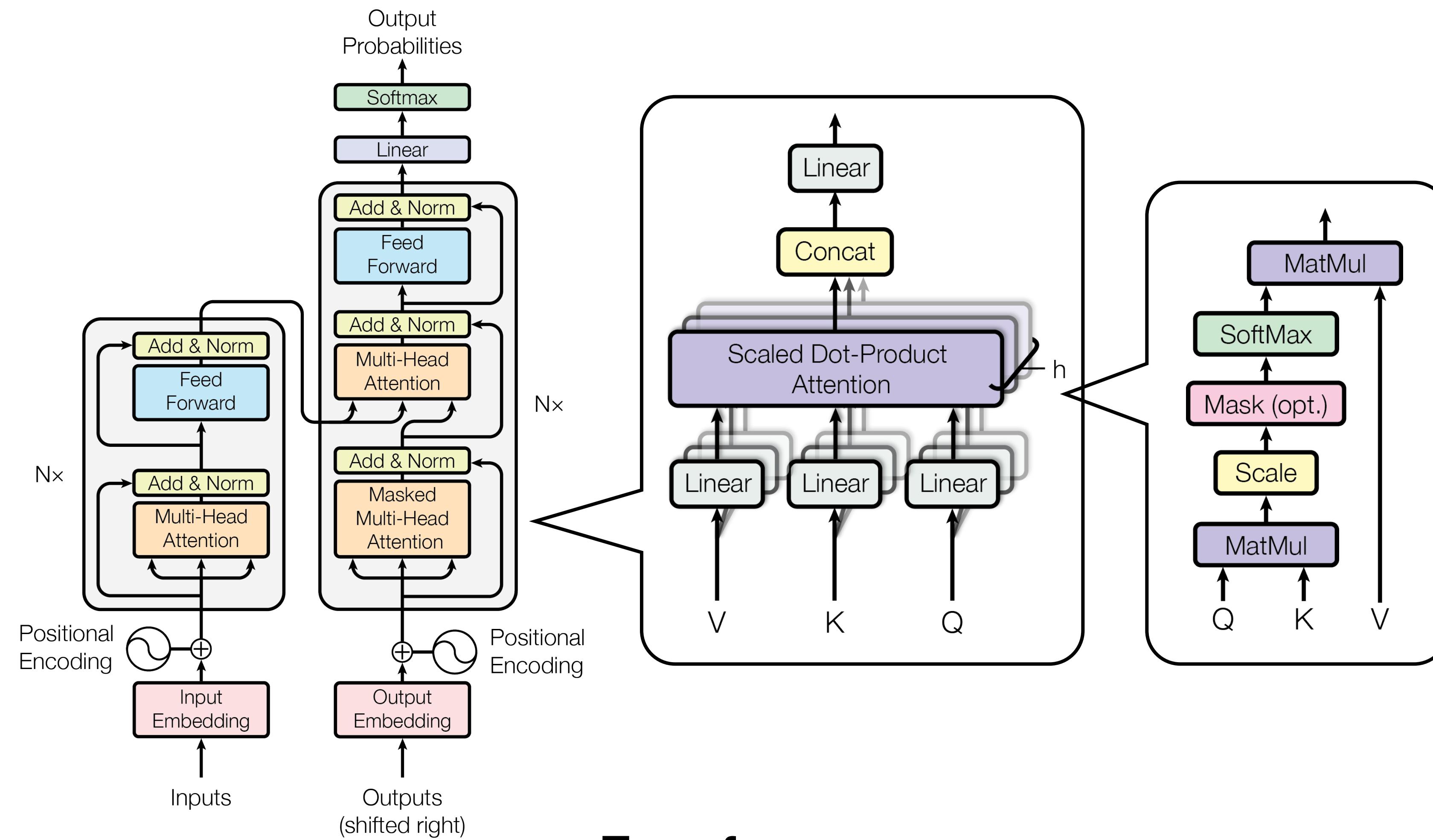
ResNet-50

Image Source: 1

Deep Residual Learning for Image Recognition [He et al., CVPR 2016]

# Deep Neural Network

Neural Networks become deeper, larger, and more complicated.



Attention is All You Need [Vaswani et al., NeurIPS 2017]

# In the Next Lecture

We will:

1. Review the **terminology of neural networks**

- Neuron, Synapses, Activation, Feature, Weight, Parameter, etc

2. Review **popular building blocks** in a neural network

- Fully-Connected, Convolution, Grouped Convolution, Depthwise Convolution, Pooling, Normalization

3. Review **classic neural networks'** architecture

- AlexNet, VGG-16, ResNet-50, MobileNetV2

4. Introduce **popular efficiency metrics** for neural networks

- MACs, FLOPs, FLOPS, #Parameters, Model Size, etc