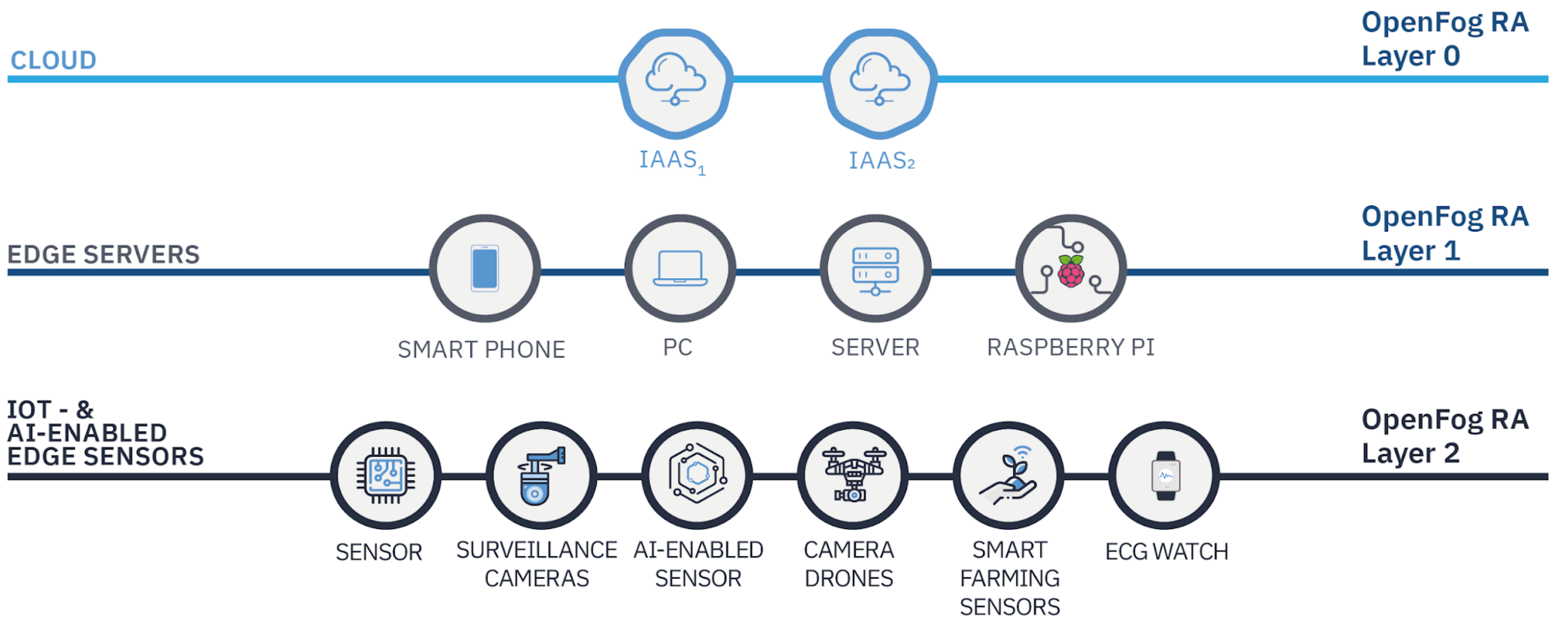
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

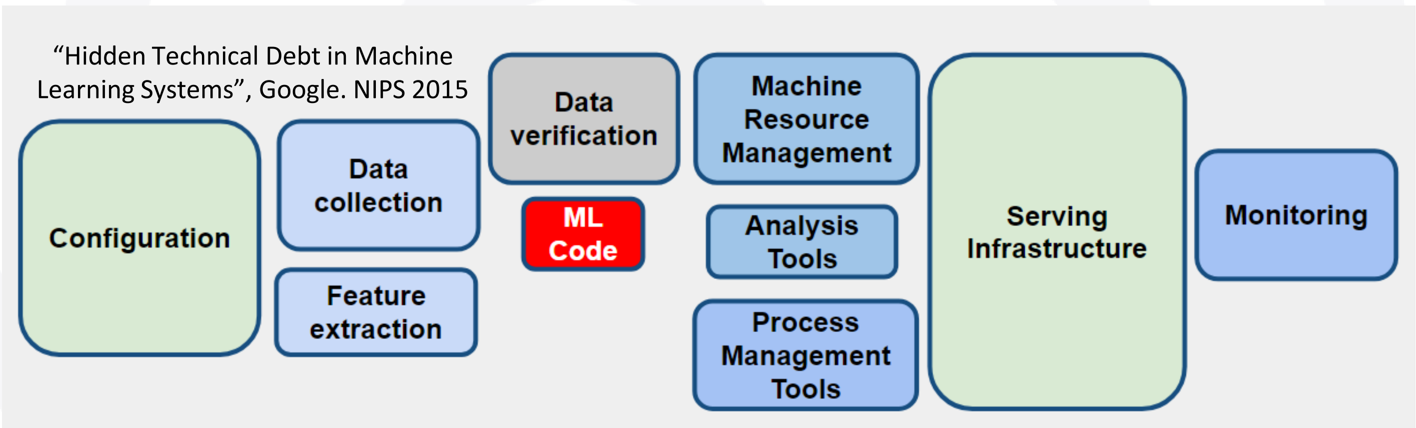
Federated Learning for Privacy Preserving Machine Learning

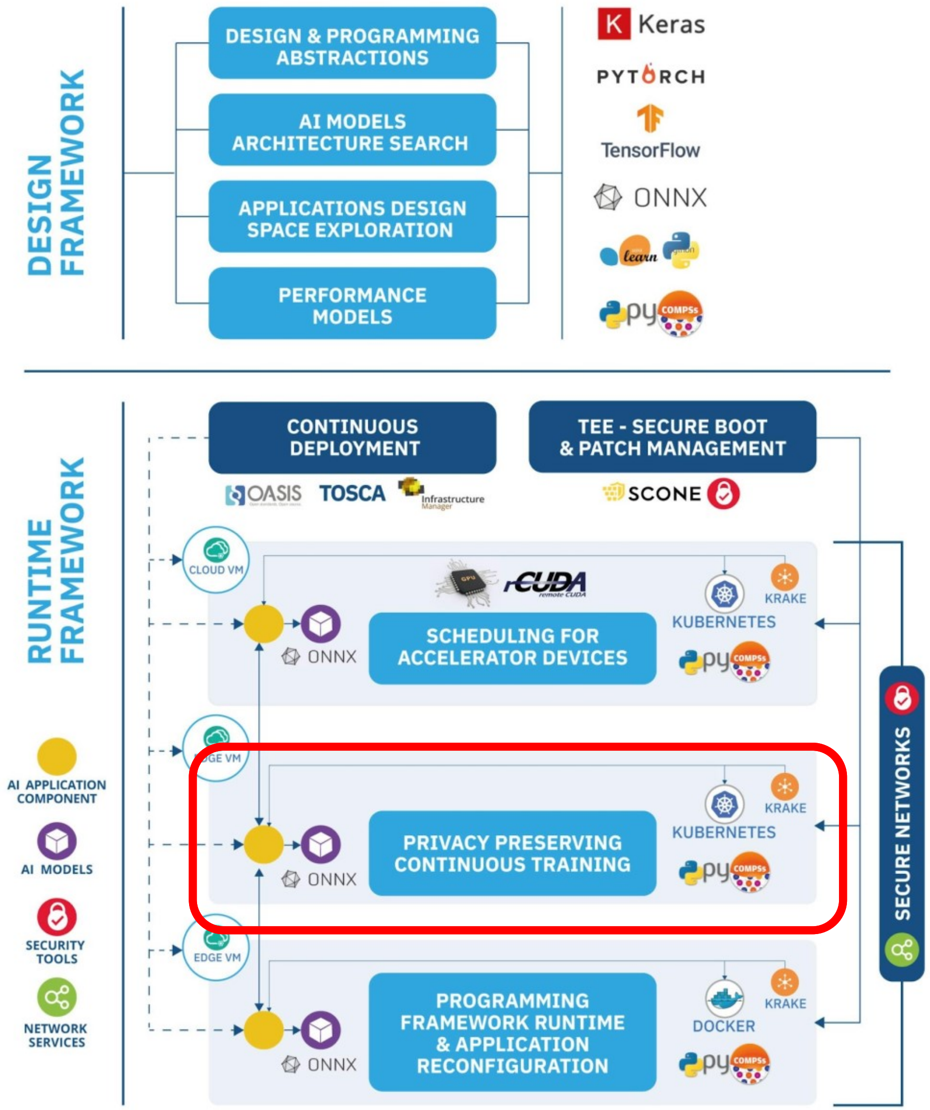
Matteo Matteucci, Politecnico di Milano

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

Federated Learning and AI-SPRINT





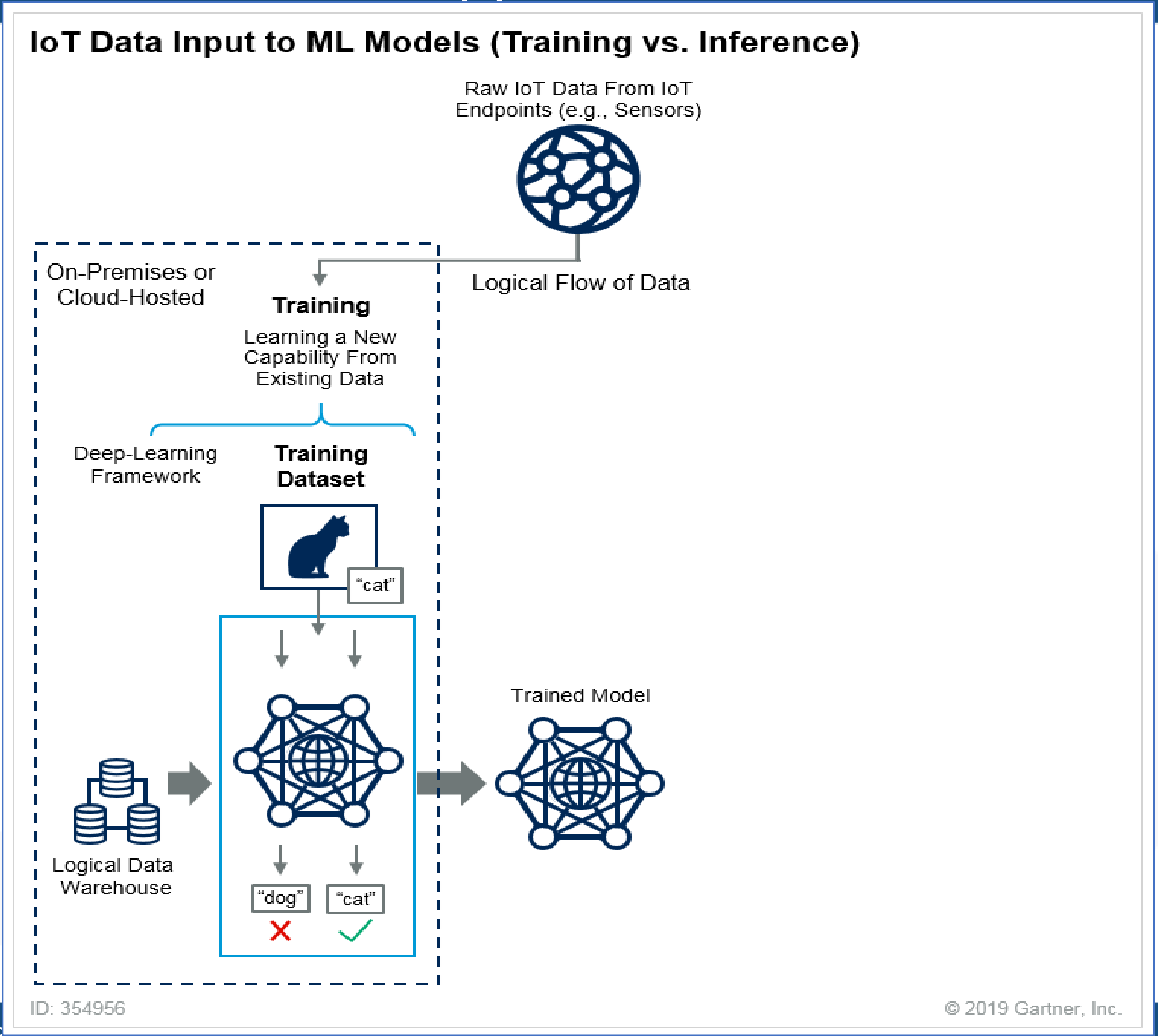


Where Does AI Happens?

- Data is born at the edge
- Pros of processing directly at the edge:
  - Low latency
  - Communication
  - Energy efficiency
  - Privacy
- Compliance to GDPR and privacy regulation laws

P. Kairouz *et al.*, “Advances and Open Problems in Federated Learning,” *arXiv:1912.04977 [cs, stat]*, Mar. 2021

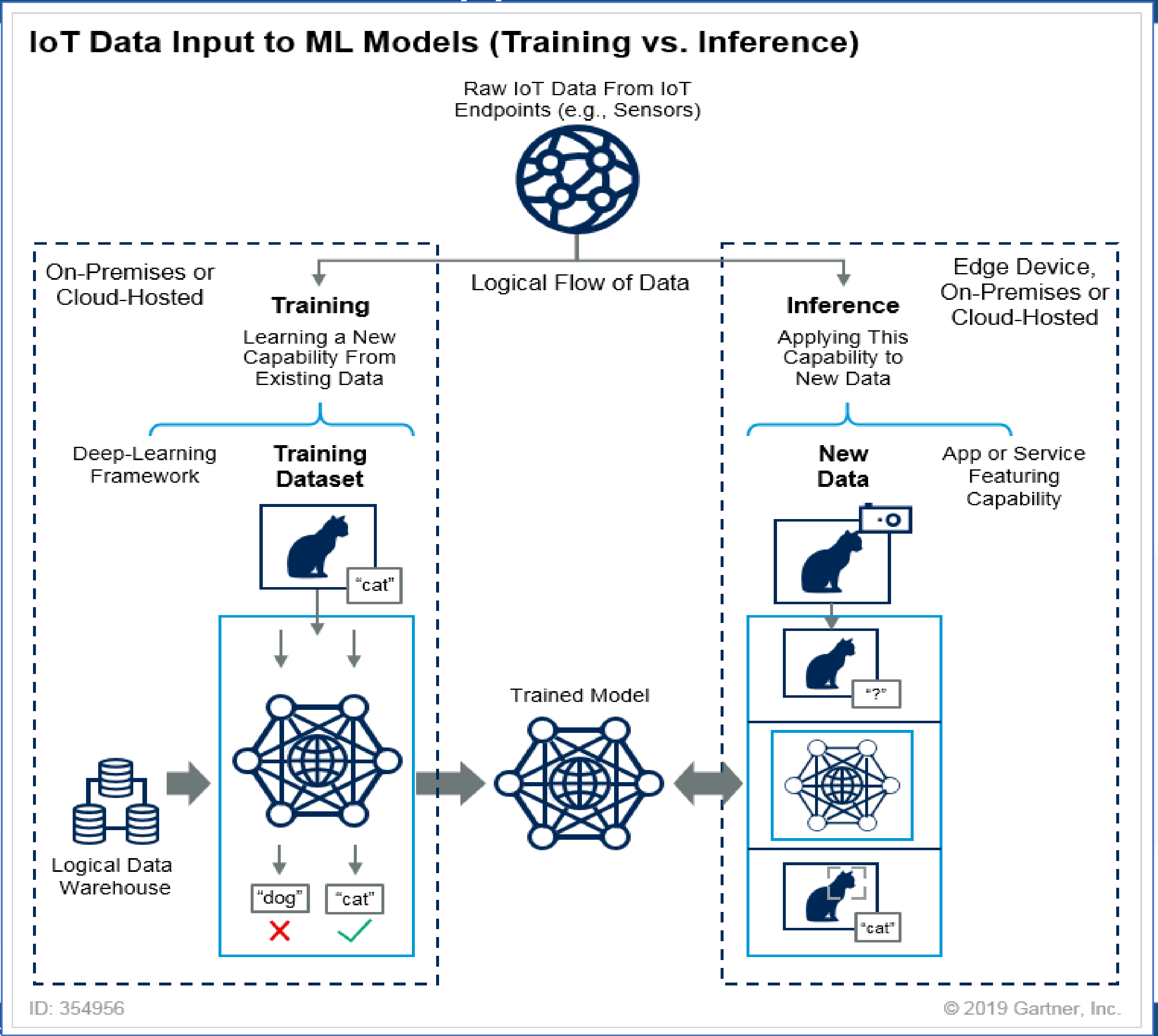
Where Does AI Happens?



Modern models are trained offline on the cloud and deployed on the field for inference on new data

Where Does AI Happens?

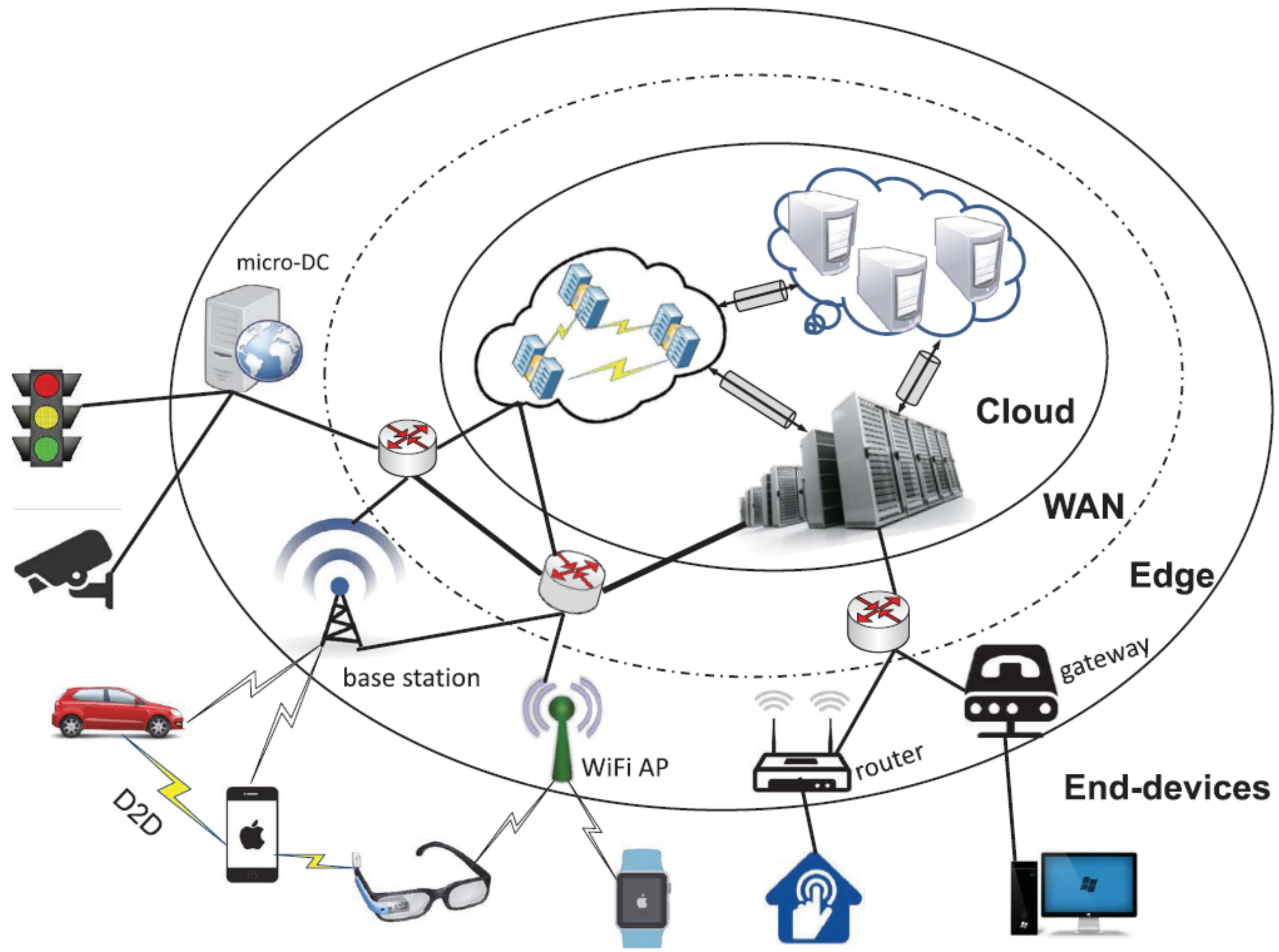
Where Does AI Happens?



Modern models are trained offline on the cloud and deployed on the field for inference on new data

FALSE

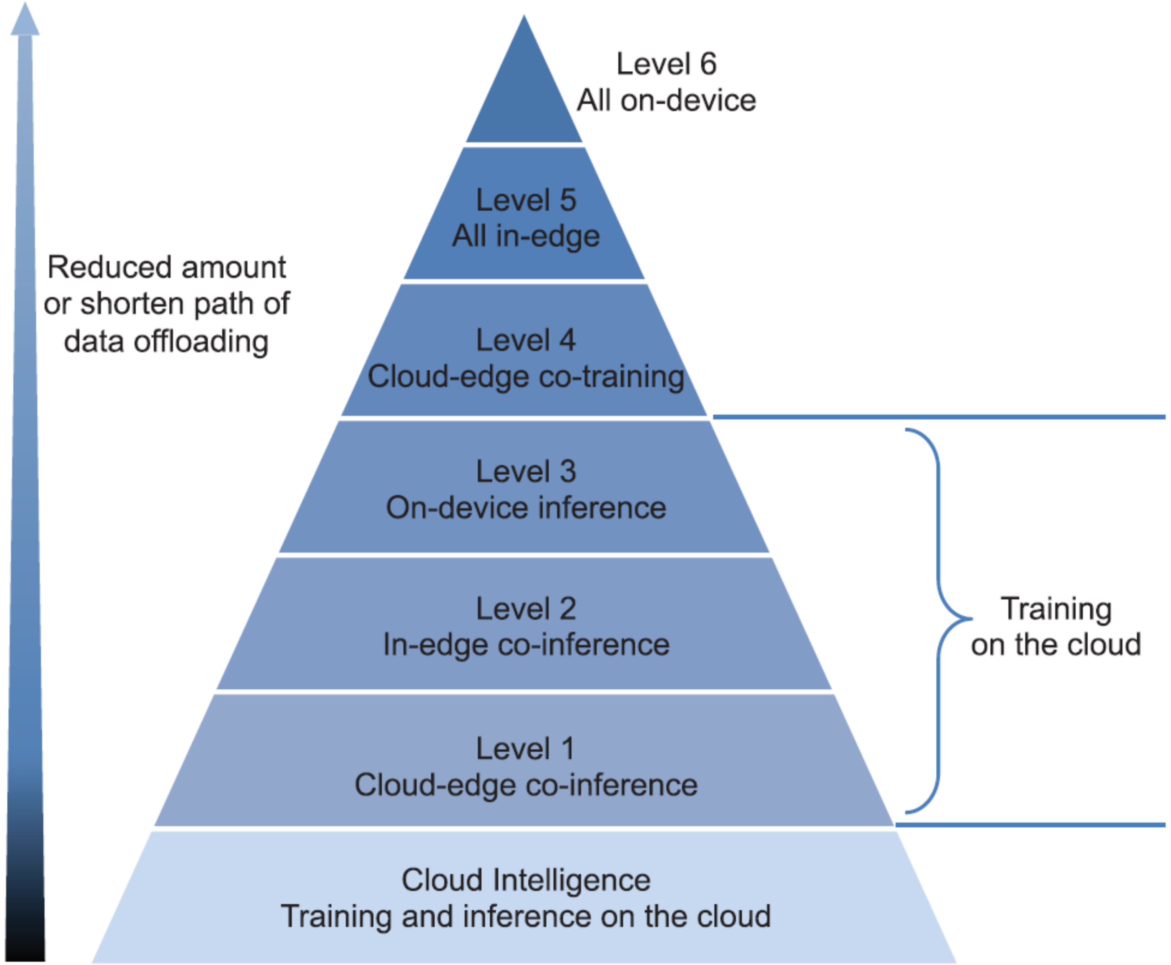
Where Does AI Happens?



Modern models are trained offline on the cloud and deployed on the field for inference on new data

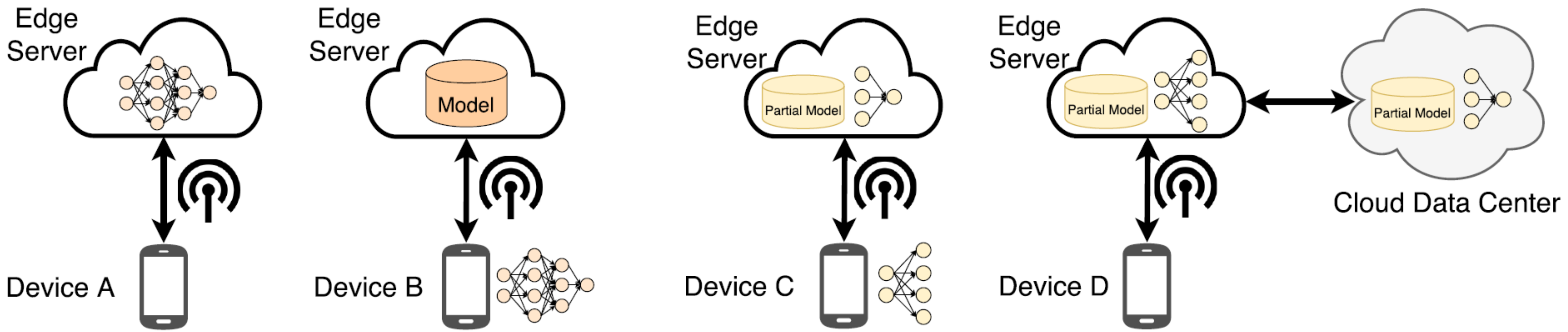
FALSE

The Edge Intelligence Paradigm



Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. Zhi Zhou, et al., Proceedings of IEEE. 2019

Model Inference on the Edge



Edge-based Device-based Edge-Device Edge-Cloud

Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. Zhi Zhou, et al., Proceedings of IEEE. 2019

Model Inference on the Edge

Technology	Highlights
Model Compression	• Weight pruning and quantization to reduce storage and computation
Model Partition	• Computation offloading to the edge server or mobile devices • Latency- and energy-oriented optimization
Model Early-Exit	• Partial DNNs model inference • Accuracy-aware
Edge Caching	• Fast response towards reusing the previous results of the same task
Input Filtering	• Detecting difference between inputs, avoiding abundant computation
Model Selection	• Inputs-oriented optimization • Accuracy-aware
Support for Multi-Tenancy	• Scheduling multiple DNN-based task • Resource-efficient
Application-specific Optimization	• Optimizations for the specific DNN-based application • Resource-efficient

Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. Zhi Zhou, et al., Proceedings of IEEE. 2019

Adantages of Training on the Edge

Assumptions	Very Conservative estimate	Less Conservative estimate
Fleet size	100	125
Duration of data collection	1 working year / 8h	1.25 working year / 10h
Volume of data generated by a single car	1TB / h	1.5TB / h
Data reduction due to preprocessing	0.0005	0.0008
Research team size	30	40
Proportion of the team submitting jobs	20%	30%
Target training time	7 days	6 days
Number of epochs required for convergence	50	50
Calculations		
Total raw data volume	203.1 PB	595.1 PB
Total data volume after preprocessing	104 TB	487.5 TB
Training time on a single DGX-1 Volta system (8 GPUs)	166 days (Inception V3) 113 days (ResNet 50)21 days (AlexNet)	778 days (Inception V3) 528 days (ResNet 50) 194 days (AlexNet)
Number of machines (DGX-1 with Volta GPUs) required to achieve target training time for the team	142 (Inception V3) 97 (ResNet 50) 18 (AlexNet)	1556 (Inception V3) 1056 (ResNet 50) 197 (AlexNet)

Source: NVIDIA https://devblogs.nvidia.com/training-self-driving-vehicles-challenge-scale/

Model Training on the Edge

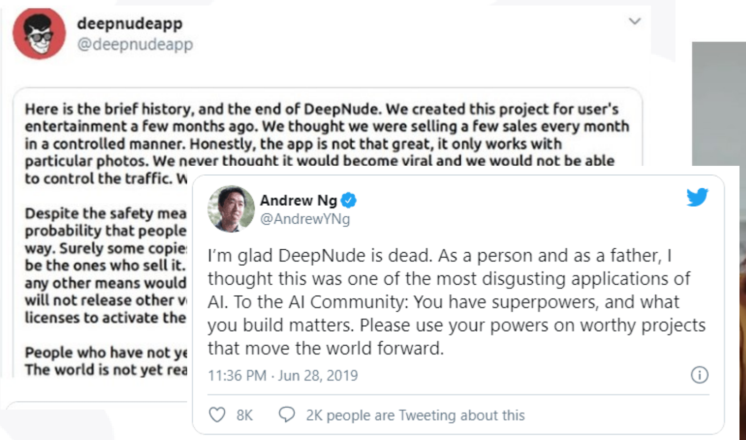
Technology	Highlights
Federated Learning	• Leave the training data distributed on the end devices • Train the shared model on the server by aggregating locally-computed updates • Preserve privacy
Aggregation Frequency Control	• Determine the best trade-off between local update and global parameter aggregation under a given resource budget • Intelligent communication control
Gradient Compression	• Gradient quantization by quantizing each element of gradient vectors to a finite-bit low precision value • Gradient sparsification by transmitting only some values of the gradient vectors
DNN Splitting	• Select a splitting point to reduce latency as much as possible • Preserve privacy
Knowledge Transfer Learning	• First train a base network (teacher network) on a base dataset and task and then transfer the learned features to a second target network (student network) to be trained on a target dataset and task • The transition from generality to specificity
Gossip Training	• Random gossip communication among devices • Full asynchronization and total decentralization • Preserve privacy

Cen d

Edge In 019

Why is this a Big Concern?

- “The enormous data that companies feed into AI-driven algorithms are susceptible to data breaches as well.“
- “AI may generate personal data […] created without the permission of the individual.”


deepnudeapp
@deepnudeapp

Here is the brief history, and the end of DeepNude. We created this project for user's entertainment a few months ago. We thought we were selling a few sales every month in a controlled manner. Honestly, the app is not that great, it only works with particular photos. We never thought it would become viral and we would not be able to control the traffic. W

Despite the safety mea
probability that people
way. Surely some copie
be the ones who sell it.
any other means would
will not release other v
licenses to activate the

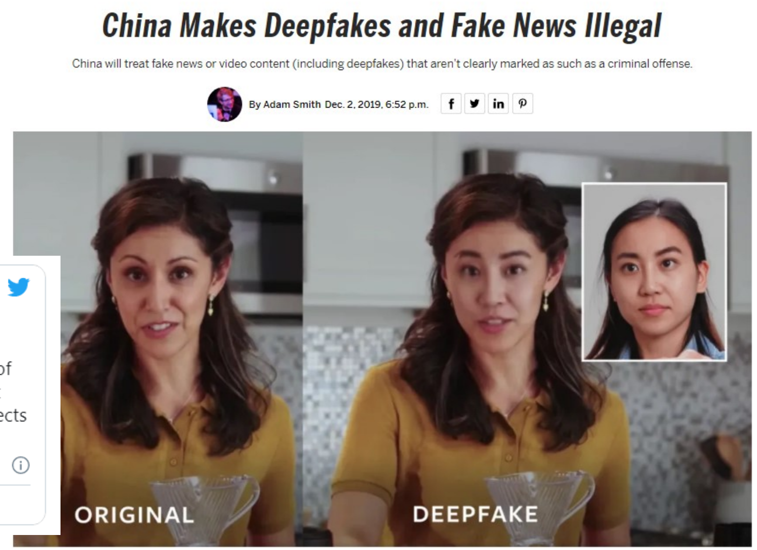
People who have not ye
The world is not yet rea

Andrew Ng
@AndrewYNg

I'm glad DeepNude is dead. As a person and as a father, I thought this was one of the most disgusting applications of AI. To the AI Community: You have superpowers, and what you build matters. Please use your powers on worthy projects that move the world forward.

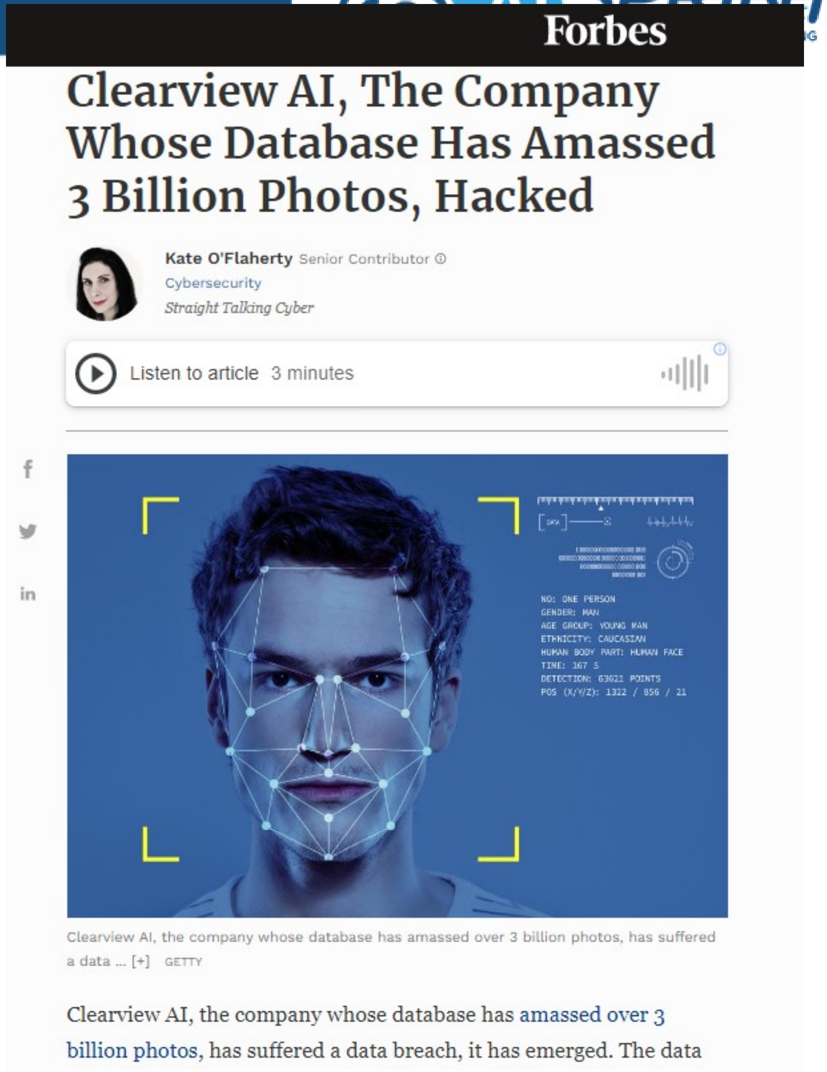
11:36 PM · Jun 28, 2019

8K 2K people are Tweeting about this


China Makes Deepfakes and Fake News Illegal

China will treat fake news or video content (including deepfakes) that aren't clearly marked as such as a criminal offense.

By Adam Smith Dec. 2, 2019, 6:52 p.m.


Forbes

Clearview AI, The Company Whose Database Has Amassed 3 Billion Photos, Hacked

Kate O'Flaherty Senior Contributor
Cybersecurity
Straight Talking Cyber

Listen to article 3 minutes

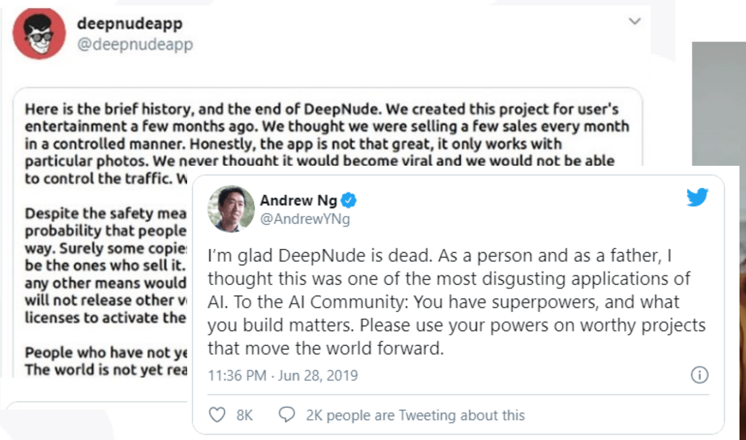
Clearview AI, the company whose database has amassed over 3 billion photos, has suffered a data ... [+] GETTY

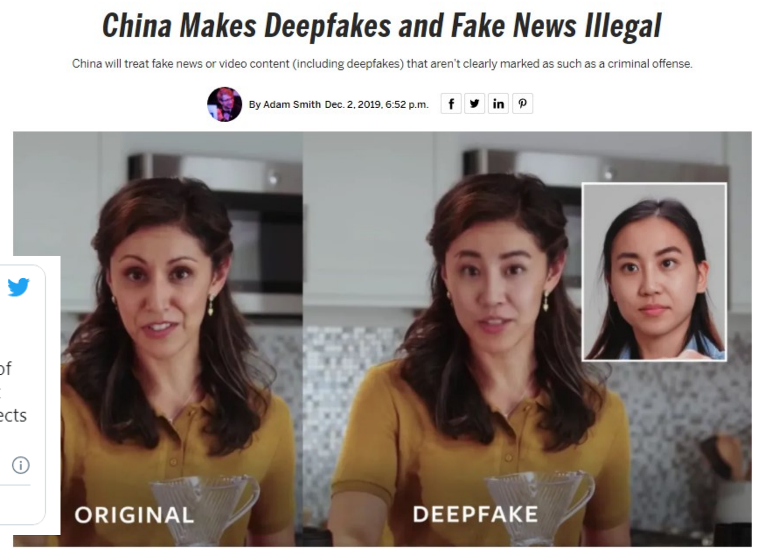
Clearview AI, the company whose database has amassed over 3 billion photos, has suffered a data breach, it has emerged. The data

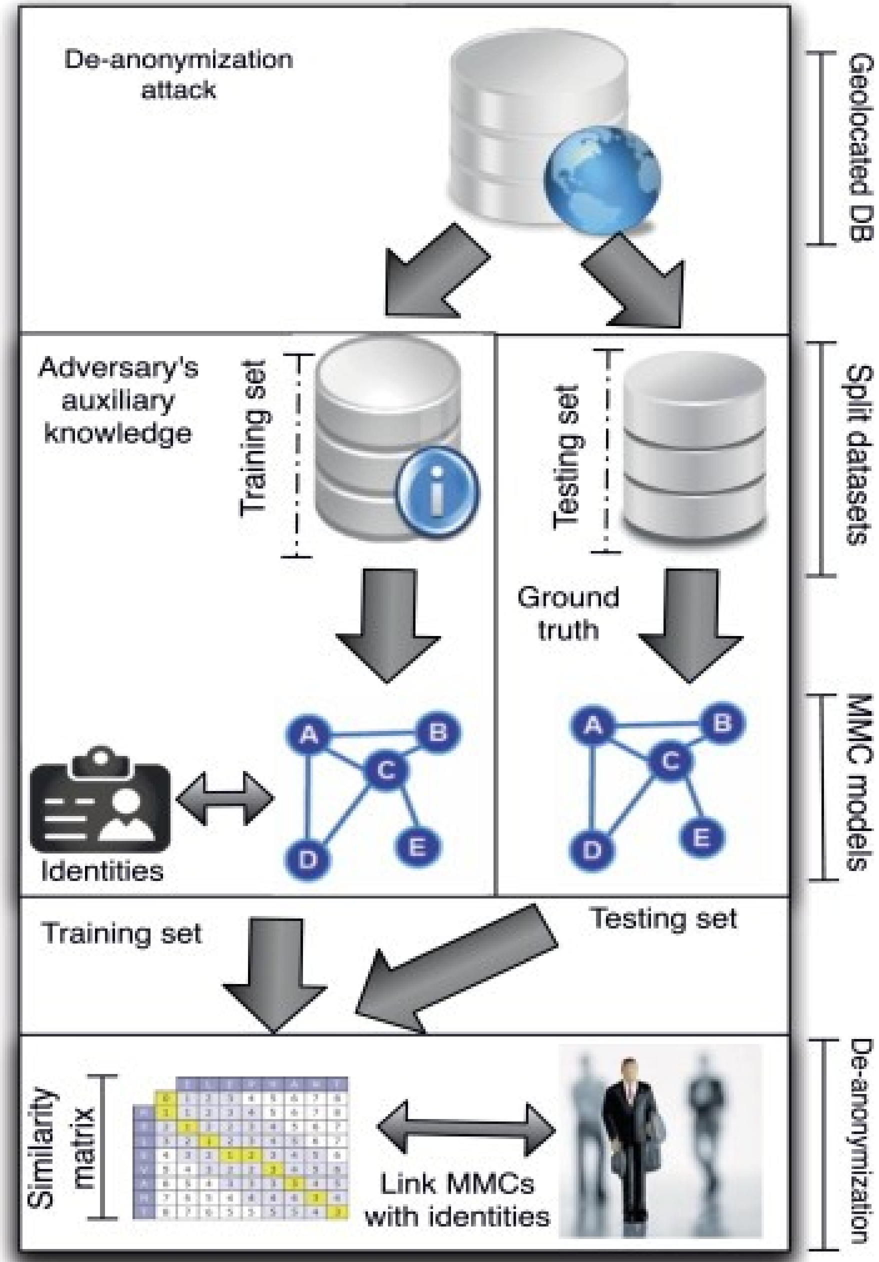
The Social Impact of Artificial Intelligence and Data Privacy Issues by Shree Das, 08 September 2020

Why is this a Big Concern?

- “The enormous data that companies feed into AI-driven algorithms are susceptible to data breaches as well.“
- “AI may generate personal data […] created without the permission of the individual.”







Are you entitled to use those data?

AI SPRINT
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

GPDP GARANTE PER LA PROTEZIONE DEI DATI PERSONALI

L'Autorità · Temi · Normativa e provvedimenti · News e comunicazione

Home / Stampa e comunicazione / Comunicato stampa / Riconoscimento facciale: Sari Real Time non è conforme alla normativa sulla privacy

Riconoscimento facciale: Sari Real Time non è conforme alla normativa sulla privacy

Ascolta

Riconoscimento facciale: Sari ... privacy

Non è favorevole il parere d... sull'utilizzo del sistema Sari ... sistema, oltre ad essere priv... automatizzato dei dati biometrici per il riconoscimento facciale a fini di sicurezza, realizzerebbe per come è progettato una forma di sorveglianza indiscriminata/di massa.

FINANCIAL TIMES

US · COMPANIES · TECH · MARKETS · CLIMATE · OPINION · WORK & CAREERS · LIFE & ARTS · HOW TO SPEND IT

Microsoft Corp + Add to myFT

Microsoft quietly deletes largest public face recognition data set

Stanford and Duke universities also remove facial recognition data

Facial recognition technology is demonstrated at an exhibition in Fujian province, China © Reuters

Search jobs · Sign in · Search · International edition

The Guardian For 200 years

Royal Free breached UK data law in 1.6m patient deal with Google's DeepMind

Information Commissioner's Office rules record transfer from London hospital to AI company failed to comply with Data Protection Act

'We underestimated the complexity of the NHS and of the rules around patient data' - DeepMind. Photograph: Alamy Stock Photo

London's Royal Free hospital failed to comply with the Data Protection Act when it handed over personal data of 1.6 million patients to DeepMind, a Google subsidiary, according to the Information Commissioner's Office.

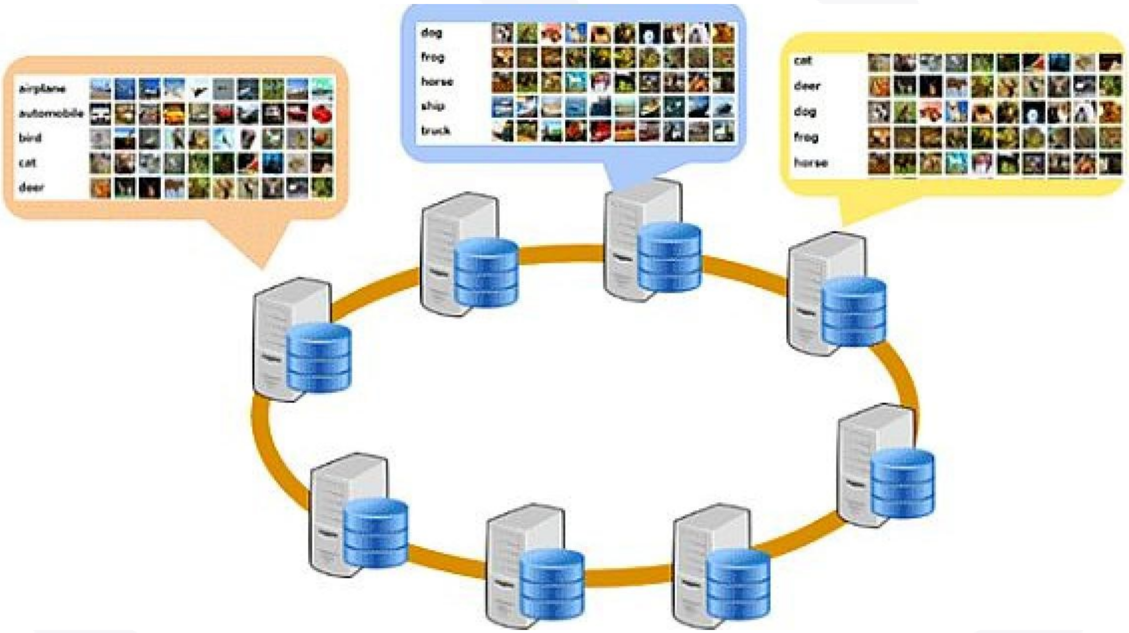
Federated Learning (FL)

"Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider.

Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective."

P. Kairouz et al., "Advances and Open Problems in Federated Learning," arXiv:1912.04977 [cs, stat], Mar. 2021

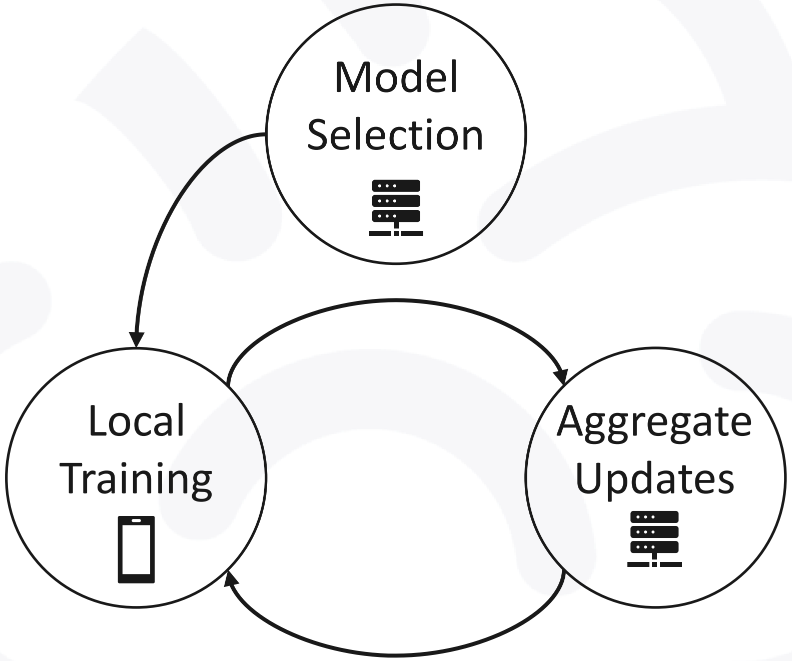
A new paradigm



- FL is fundamentally different from distributed machine learning, where:
  - Data are stored in a network of powerful cloud machines
  - Data can be shuffled and balanced across clients
  - Any client has access to any part of the dataset
  - Computation is the bottleneck
  - Typically, 1-1000 clients

A. Willing, "Asynchronous distributed deep learning technology," eenewseurope.com, Aug. 2020

The general FL pipeline



- Model Selection (server)

Define and initialize a global ML model, then send it to the clients

- Local Training (clients)

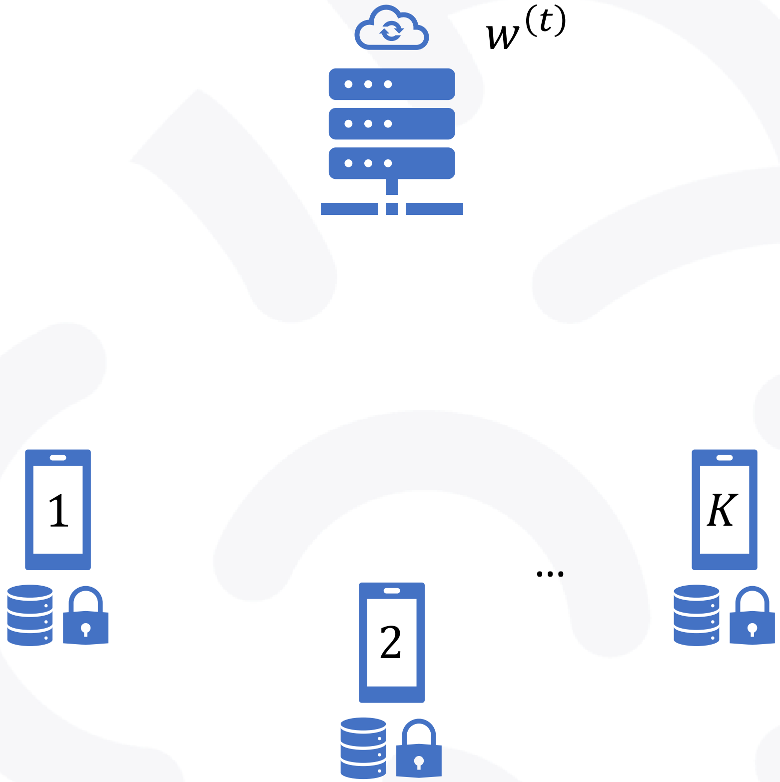
Train the global model on private data, then send the updated model back to the server

- Aggregate Updates (server)

Combine the local updates into a single, new, global model, then repeat the process

A. Hard et al., "Federated Learning for Mobile Keyboard Prediction," arXiv:1811.03604 [cs], Feb. 2019

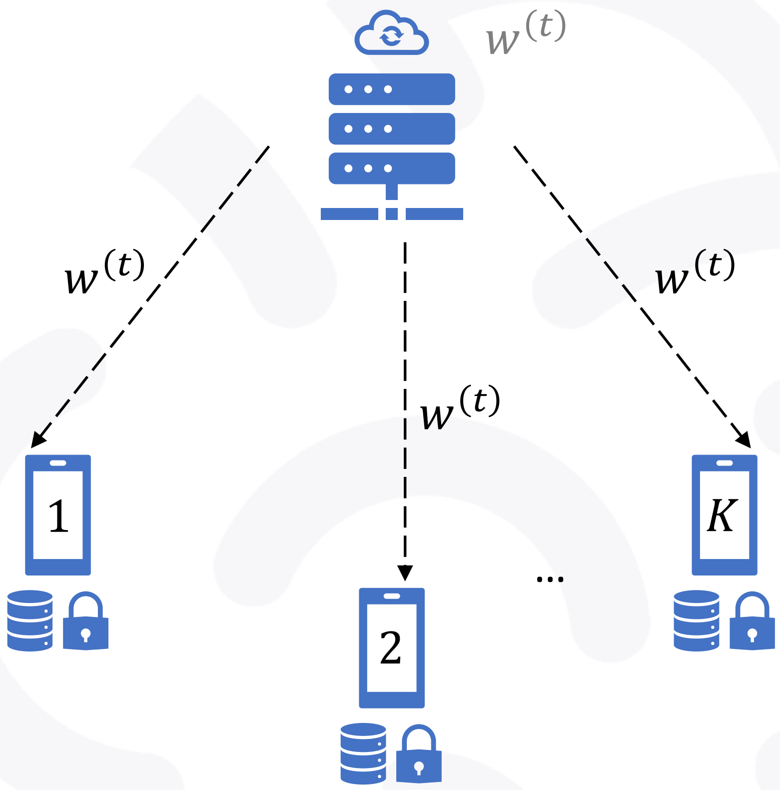
Federated Averaging



1. Select a random set of K clients

H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629 [cs], Feb. 2017

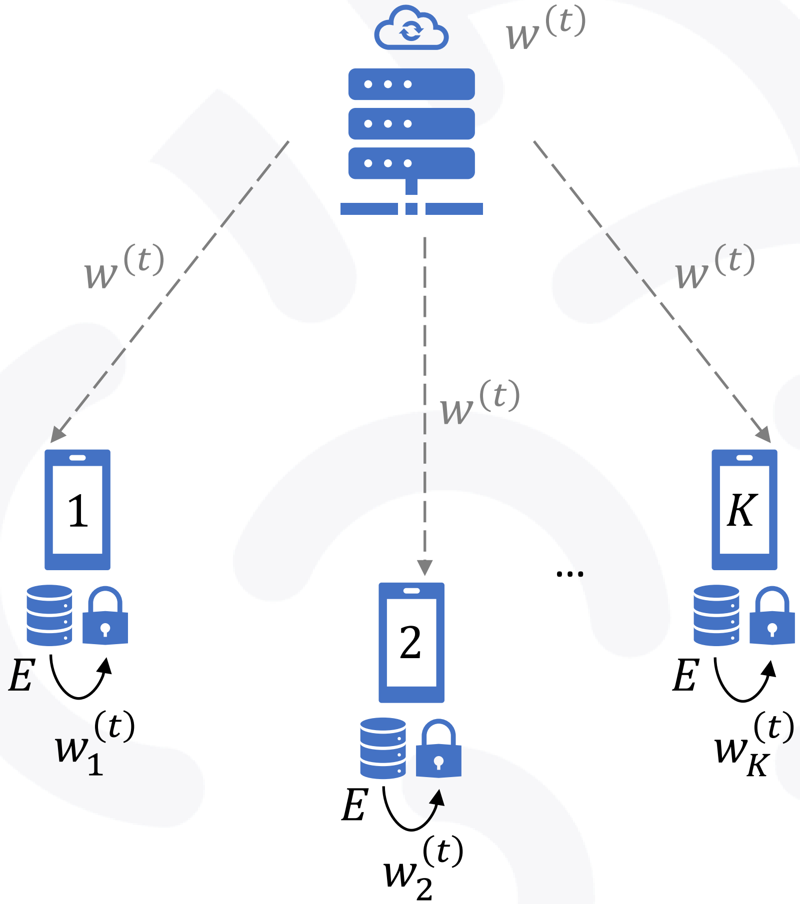
Federated Averaging



1. Select a random set of K clients
2. Broadcast $w^{(t)}$

H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629 [cs], Feb. 2017

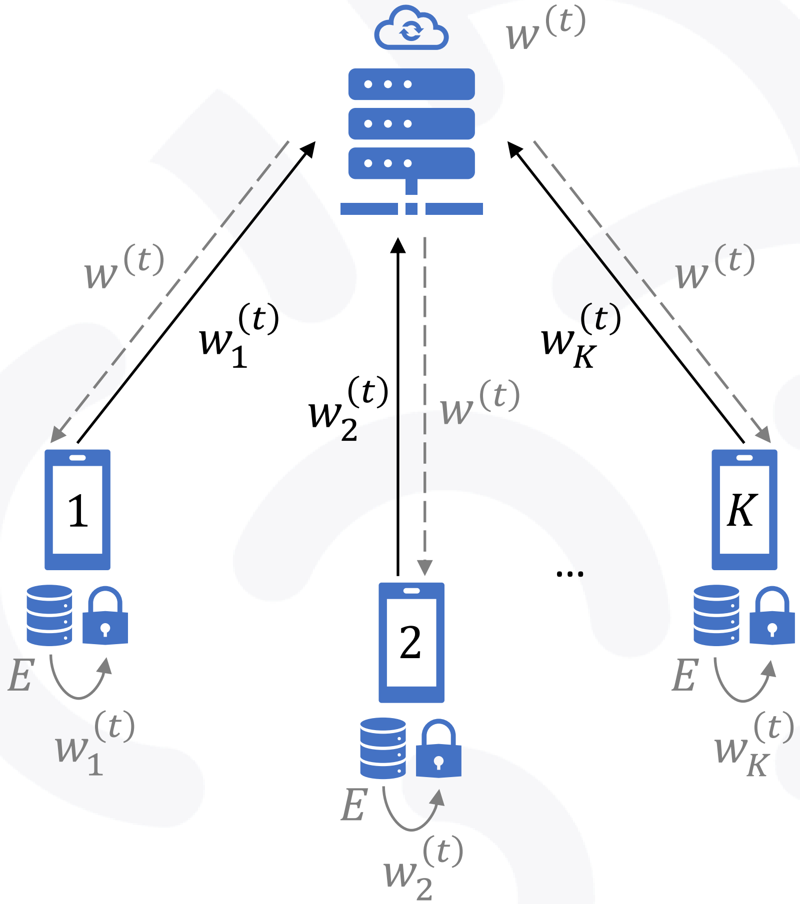
Federated Averaging



1. Select a random set of K clients
2. Broadcast $w^{(t)}$
3. Perform E iterations of SGD locally as $w_k^{(t)} \leftarrow w_k - \eta \nabla \mathcal{L}(w; b)$

H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629 [cs], Feb. 2017

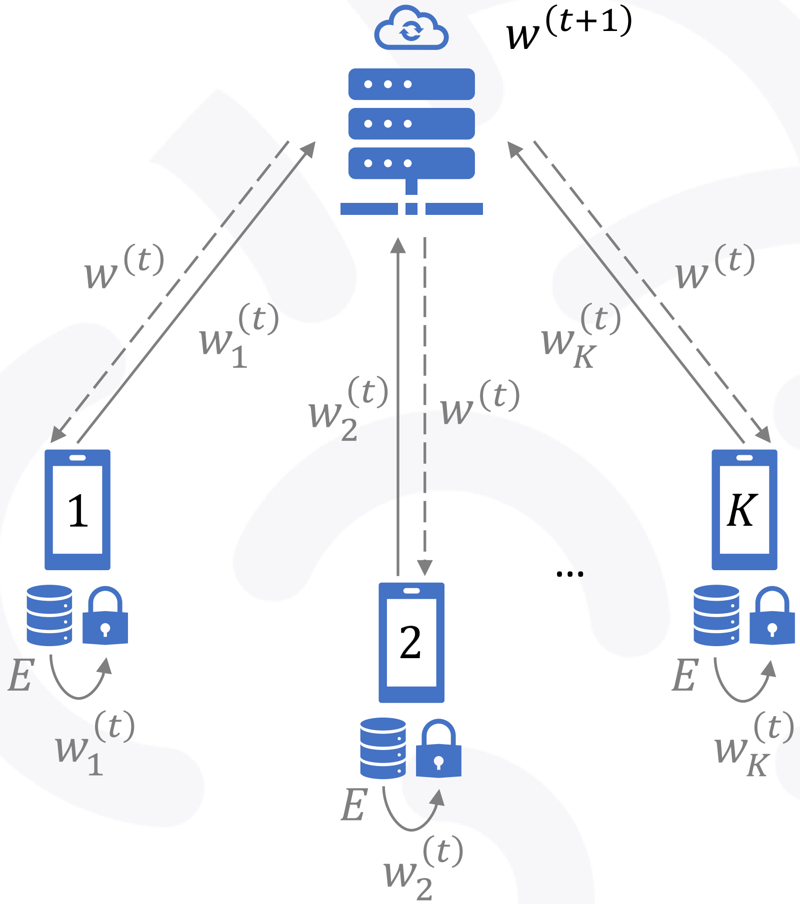
Federated Averaging



1. Select a random set of K clients
2. Broadcast $w^{(t)}$
3. Perform E iterations of SGD locally as $w_k^{(t)} \leftarrow w_k - \eta \nabla \mathcal{L}(w; b)$
4. Send $w_k^{(t)}$ back to the server

H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629 [cs], Feb. 2017

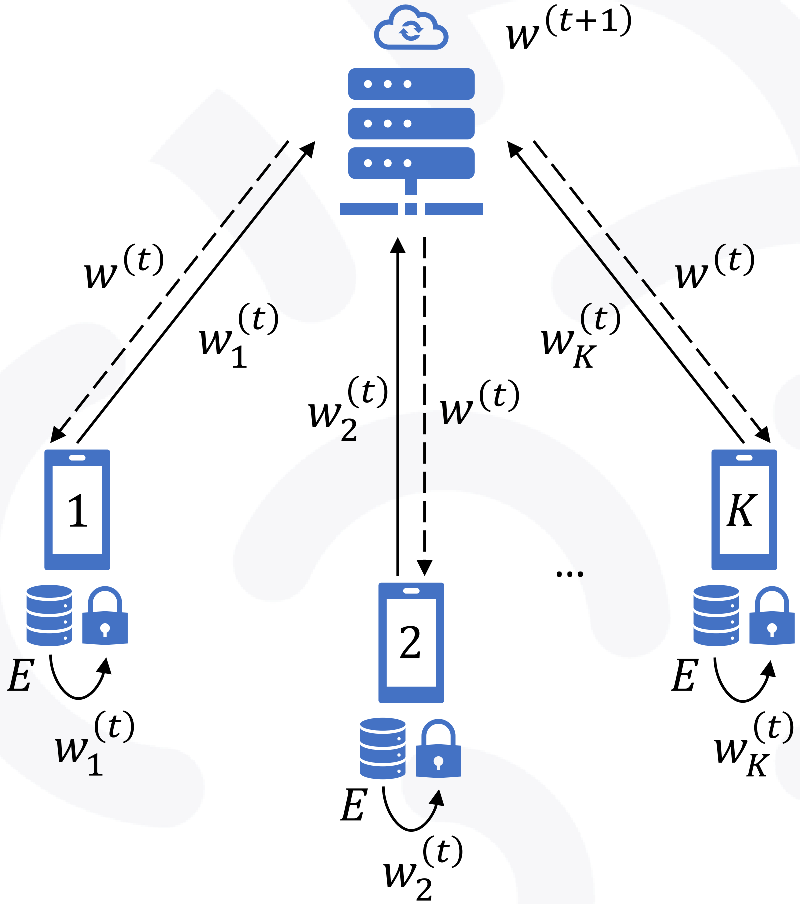
Federated Averaging



1. Select a random set of K clients
2. Broadcast $w^{(t)}$
3. Perform E iterations of SGD locally as $w_k^{(t)} \leftarrow w_k - \eta \nabla \mathcal{L}(w; b)$
4. Send $w_k^{(t)}$ back to the server
5. Aggregate updates as $w^{(t+1)} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t)}$

H. B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” arXiv:1602.05629 [cs], Feb. 2017

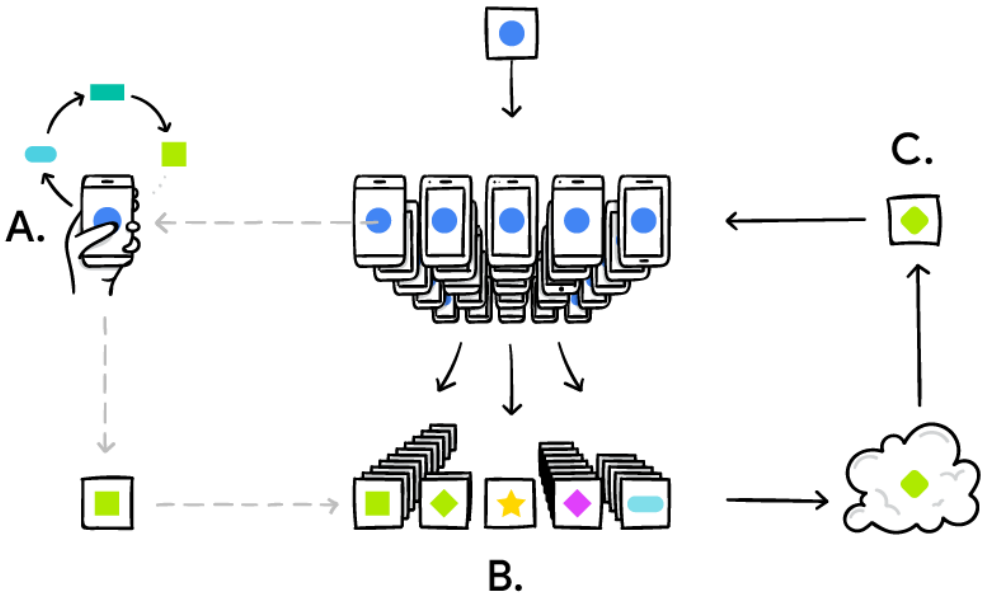
Federated Averaging



1. Select a random set of K clients
2. Broadcast $w^{(t)}$
3. Perform E iterations of SGD locally as $w_k^{(t)} \leftarrow w_k - \eta\nabla\mathcal{L}(w;b)$
4. Send $w_k^{(t)}$ back to the server
5. Aggregate updates as $w^{(t+1)} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t)}$
6. If not converged, go to 1.

H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv:1602.05629 [cs], Feb. 2017

FL and the Google Keyboard



A. Each client computes a step of stochastic gradient descent locally on private data
B. The server collects the gradients and performs an aggregated update on the previous model
C. The new model is broadcasted to the clients and the process repeats

A. Hard et al., “Federated Learning for Mobile Keyboard Prediction,” arXiv:1811.03604 [cs], Feb. 2019

AI SPRINT
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

Federated Learning for Privacy Preserving Machine Learning

Matteo Matteucci, Politecnico di Milano

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

Federated Learning Dimensions

- Federated Learning
  - Network Topology → Centralized, Clustered, Fully-decentralized
  - Data Availability → Cross-silo, Cross-device
  - Data Partition → Horizontal, Vertical, Transfer Learning
  - Open-Source Framework → Tensorflow Federated, PySift, FATE

V. Mothukuri *et al.*, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021

Network Topology

Centralized Federated Learning

- Trusted third party to monitor and manage the learning process
- All clients directly communicate to the central server
- Aggregation occurs on the server

V. Mothukuri *et al.*, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021

Network Topology

Centralized Federated Learning

- Trusted third party to monitor and manage the learning process
- All clients directly communicate to the central server
- Aggregation occurs on the server

Clustered Federated Learning

- Trusted third party to monitor and manage the learning process
- Clients are clustered according to their data distribution or system constraints
- Aggregation occurs on the server, but follows the clustering prescriptions

V. Mothukuri *et al.*, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021

Network Topology

Centralized Federated Learning

- Trusted third party to monitor and manage the learning process
- All clients directly communicate to the central server
- Aggregation occurs on the server

Clustered Federated Learning

- Trusted third party to monitor and manage the learning process
- Clients are clustered according to their data distribution or system constraints
- Aggregation occurs on the server, but follows the clustering prescriptions

Fully-decentralized Federated Learning

- Peer-to-peer topology, no trusted third party
- A trusted P2P protocol substitutes the role of the central server
- Aggregation occurs on the client
- Blochckain-based update ledger

V. Mothukuri *et al.*, “A survey on security and privacy of federated learning,” *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021

Data Availability

Distributed Machine Learning

- Data stored in a network of powerful cloud machines
- Data can be shuffled and balanced across clients
- Any client has access to any part of the dataset
- Computation is the bottleneck
- Typically, 1-1000 clients

P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *arXiv:1912.04977 [cs, stat]*, Mar. 2021

Data Availability

Distributed Machine Learning

- Data stored in a network of powerful cloud machines
- Data can be shuffled and balanced across clients
- Any client has access to any part of the dataset
- Computation is the bottleneck
- Typically, 1-1000 clients

Cross-Silo Federated Learning

- Data stored in edge devices with high computational power (institutions)
- Data never leave the client
- Data can be accessed only by the owner and data samples are never explicitly shared
- Computation or communication can be the bottleneck
- Typically, 2-100 clients

P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *arXiv:1912.04977 [cs, stat]*, Mar. 2021

Data Availability

Distributed Machine Learning

- Data stored in a network of powerful cloud machines
- Data can be shuffled and balanced across clients
- Any client has access to any part of the dataset
- Computation is the bottleneck
- Typically, 1-1000 clients

Cross-Silo Federated Learning

- Data stored in edge devices with high computational power (institutions)
- Data never leave the client
- Data can be accessed only by the owner and data samples are never explicitly shared
- Computation or communication can be the bottleneck
- Typically, 2-100 clients

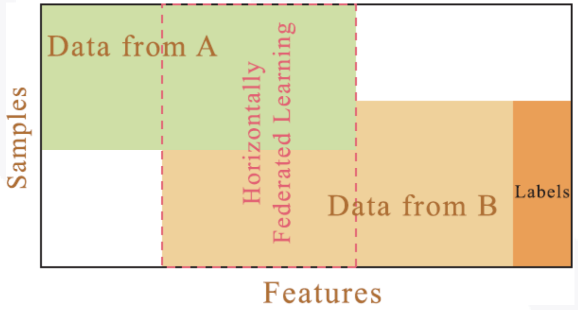
Cross-Device Federated Learning

- Data stored in edge devices with low computational power (end-users)
- Data never leave the client
- Data can be accessed only by the owner and data samples are never explicitly shared
- Communication is the bottleneck
- Up to 10^6 clients

P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *arXiv:1912.04977 [cs, stat]*, Mar. 2021

Horizontal Federated Learning

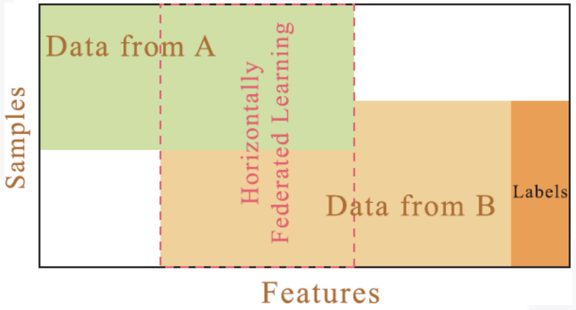
- Features overlap a lot
- Users overlap a little
- Example: same service provider in different regions



C. Zhang *et al.*, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021

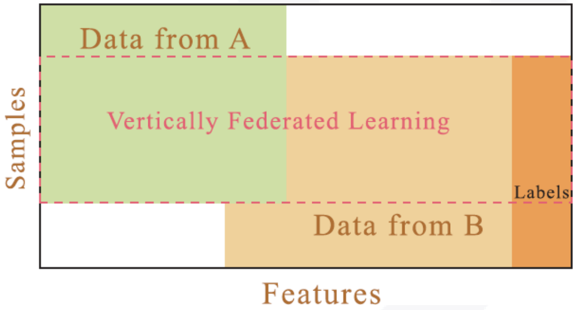
Horizontal Federated Learning

- Features overlap a lot
- Users overlap a little
- Example: same service provider in different regions



Vertical Federated Learning

- Features overlap a little
- Users overlap a lot
- Example: two different institutions, e.g., a bank and a store in the same region

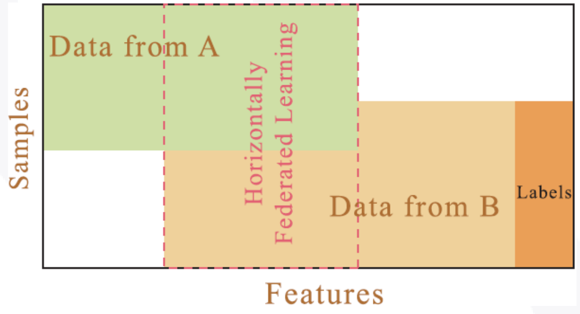


C. Zhang *et al.*, “A survey on federated learning,” *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021

Data Partition

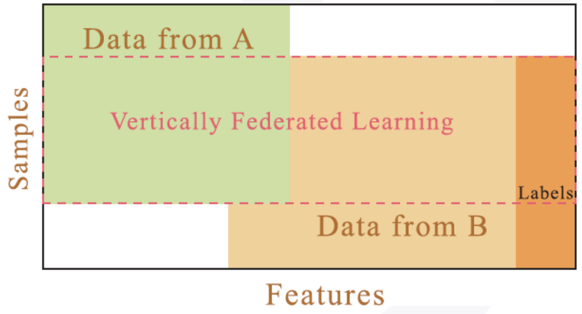
Horizontal Federated Learning

- Features overlap a lot
- Users overlap a little
- Example: same service provider in different regions



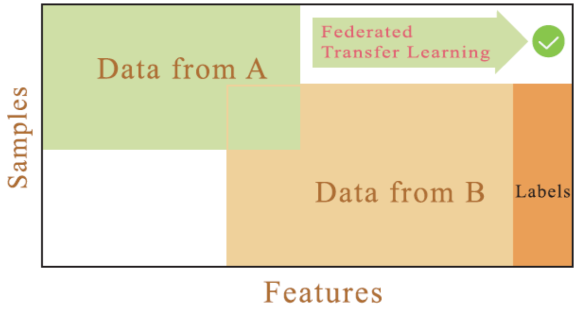
Vertical Federated Learning

- Features overlap a little
- Users overlap a lot
- Example: two different institutions, e.g., a bank and a store in the same region



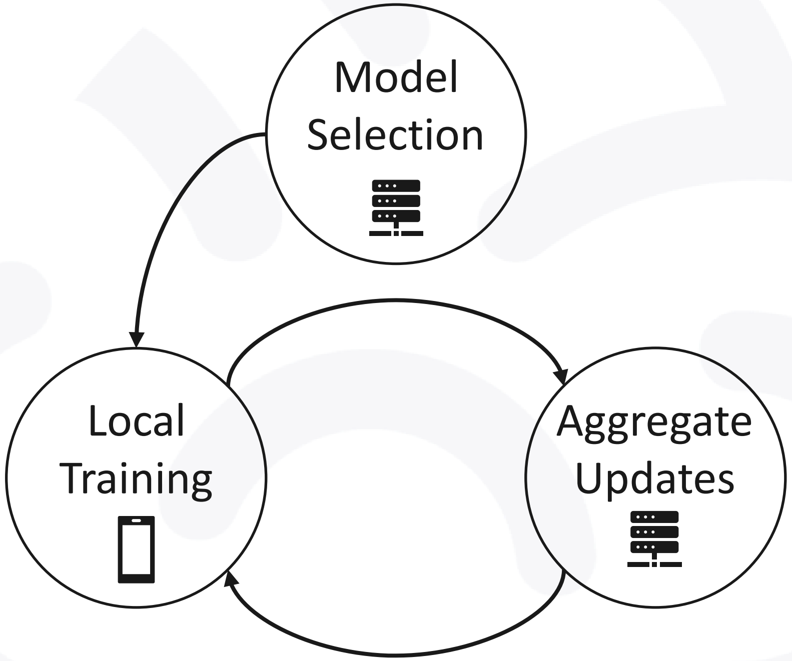
Federated Transfer Learning

- Features overlap a little
- Users overlap a little
- Example: two different institutions in different regions



C. Zhang *et al.*, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021

The general FL pipeline



- Model Selection (server)

Define and initialize a global ML model, then send it to the clients

- Local Training (clients)

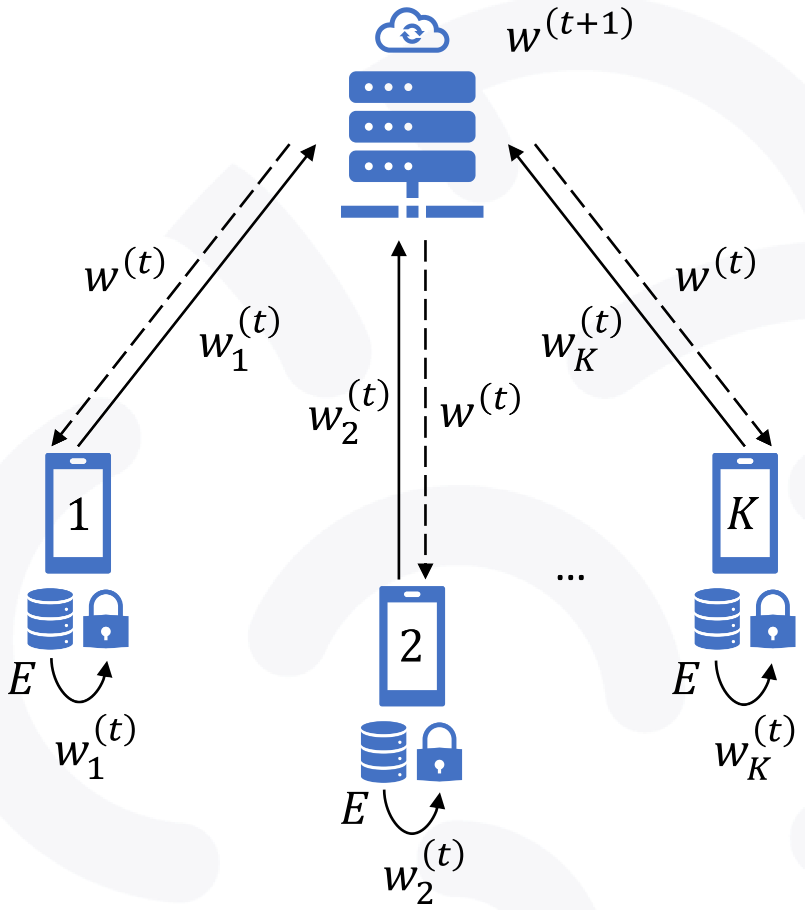
Train the global model on private data, then send the updated model back to the server

- Aggregate Updates (server)

Combine the local updates into a single, new, global model, then repeat the process

A. Hard et al., "Federated Learning for Mobile Keyboard Prediction," arXiv:1811.03604 [cs], Feb. 2019

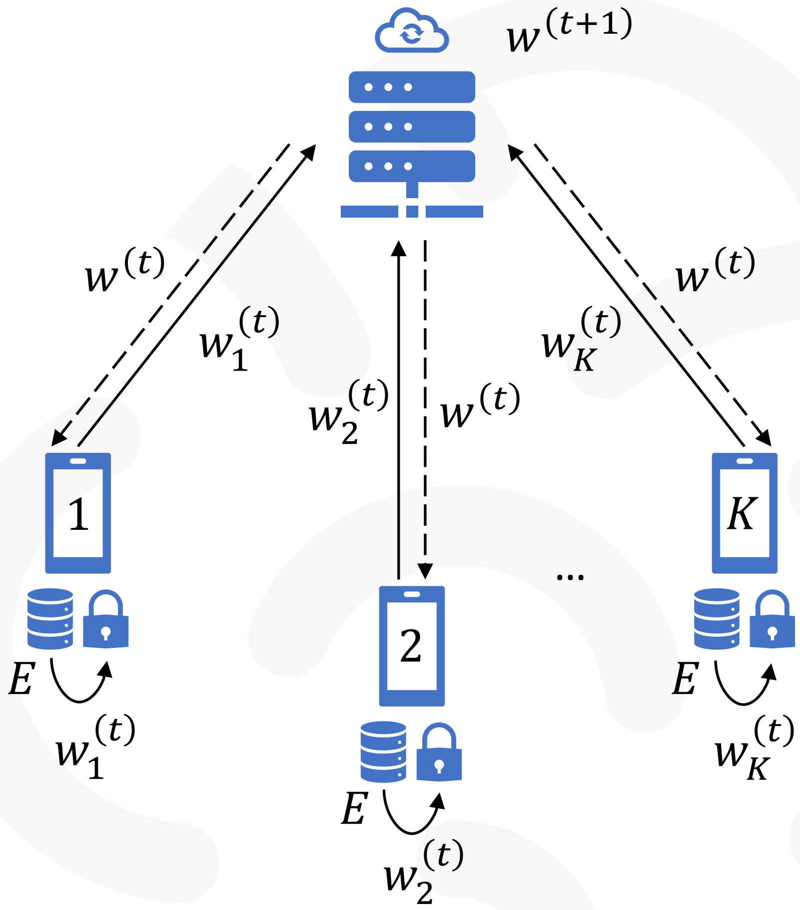
Statistical and System Heterogeneity



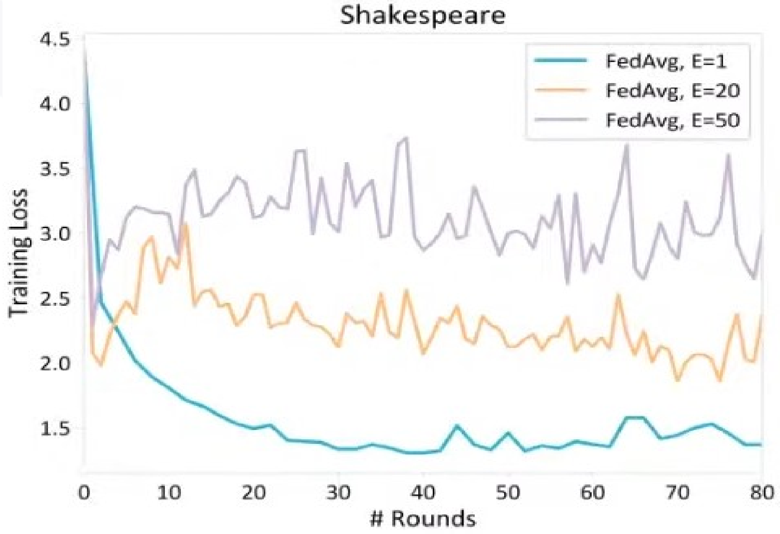
- Federated Averaging
- ✓ Simple and easy to understand
- ✓ Works well in practice
- ✗ Can diverge in heterogeneous settings

V. Smith, "On Heterogeneity in Federated Settings," Ep. 3 of Stanford MLSys Seminar Series, Oct. 2020

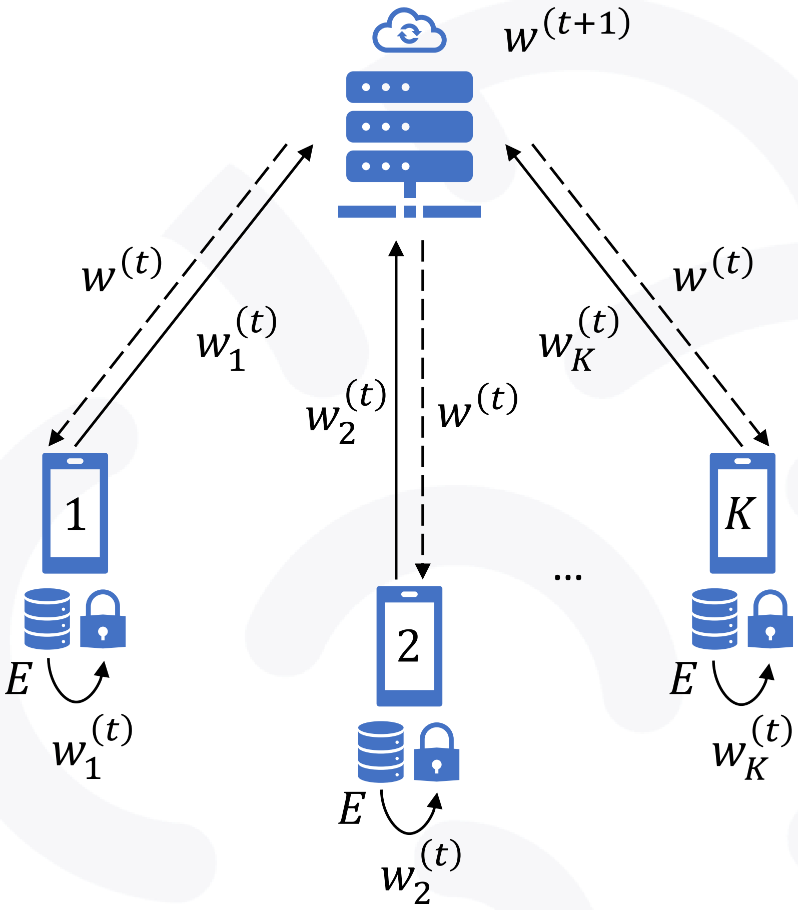
Statistical and System Heterogeneity



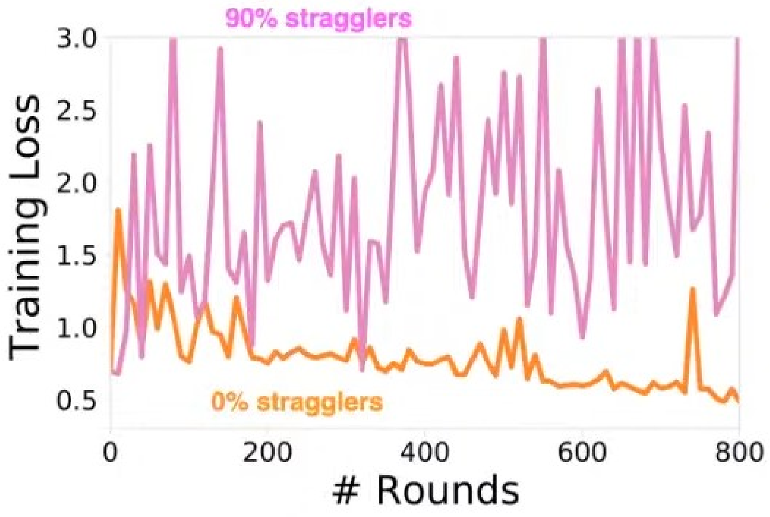
- Federated Averaging
- ✗ Statistical heterogeneity



V. Smith, "On Heterogeneity in Federated Settings," Ep. 3 of Stanford MLSys Seminar Series, Oct. 2020



- Federated Averaging
- ✗ System heterogeneity



V. Smith, "On Heterogeneity in Federated Settings," Ep. 3 of Stanford MLSys Seminar Series, Oct. 2020

Statistical and System Heterogeneity

New local update:

$$\min_{w} \underbrace{F_k(w)}_{\text{Original local objective}} + \underbrace{\frac{\mu}{2}\left\|w - w^{(t)}\right\|^2}_{\text{Regularization term to discourage big changes}}$$

- FedProx
- ✓ Statistical heterogeneity: encourage well-behaved updates using a regularization term
- ✓ System heterogeneity: allow for incomplete updates after a fixed ΔT

T. Li et al., "Federated Optimization in Heterogeneous Networks," arXiv:1812.06127 [cs, stat], Apr. 2020

Statistical and System Heterogeneity

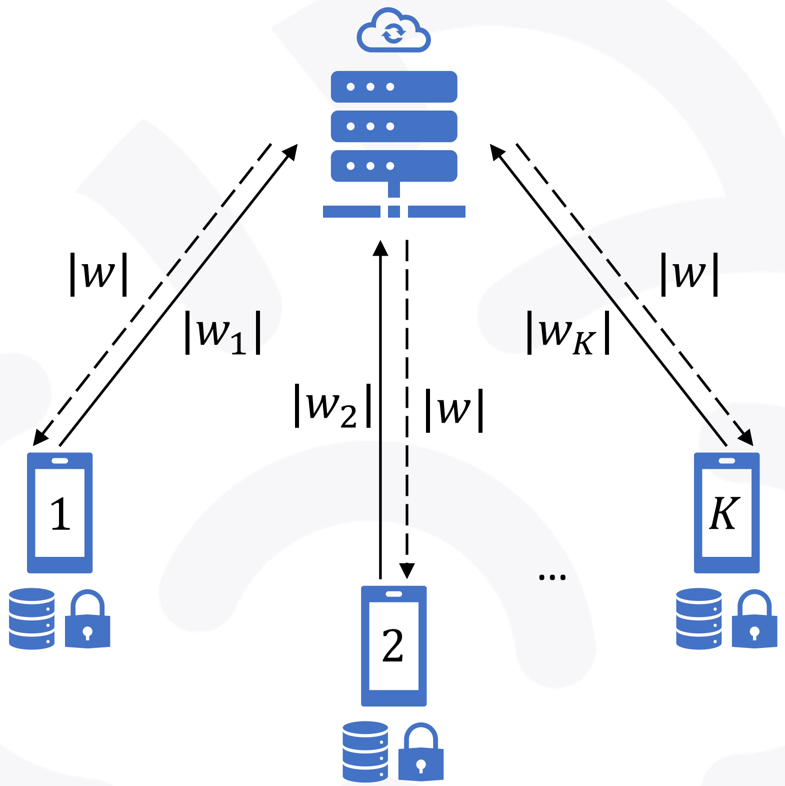
New local update:

$$\min_{w} F_k(w) + \cancel{\frac{\mu}{2}\left\|w - w^{(t)}\right\|^2}$$

- FedProx
- ✓ Statistical heterogeneity: encourage well-behaved updates using a regularization term
- ✓ System heterogeneity: allow for incomplete updates after a fixed ΔT
- ✓ Generalizes FedAvg ($\mu = 0$)

T. Li *et al*., "Federated Optimization in Heterogeneous Networks," *arXiv:1812.06127 [cs, stat]*, Apr. 2020

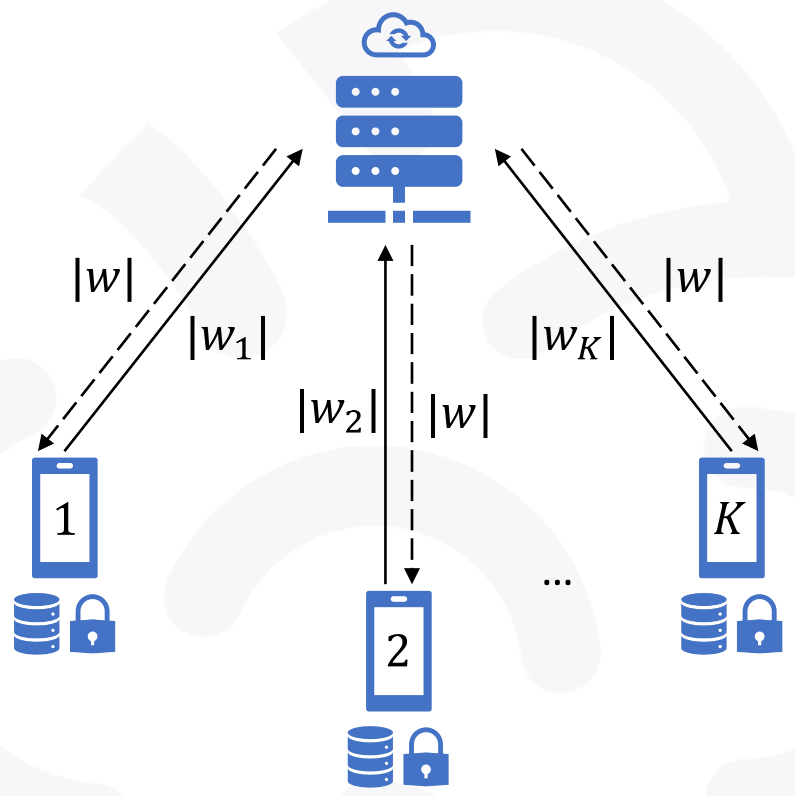
Reduce the Communication Costs



- Quantization

Reduce the number of bits required for the update with discretization

W. Y. B. Lim *et al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020



- Quantization

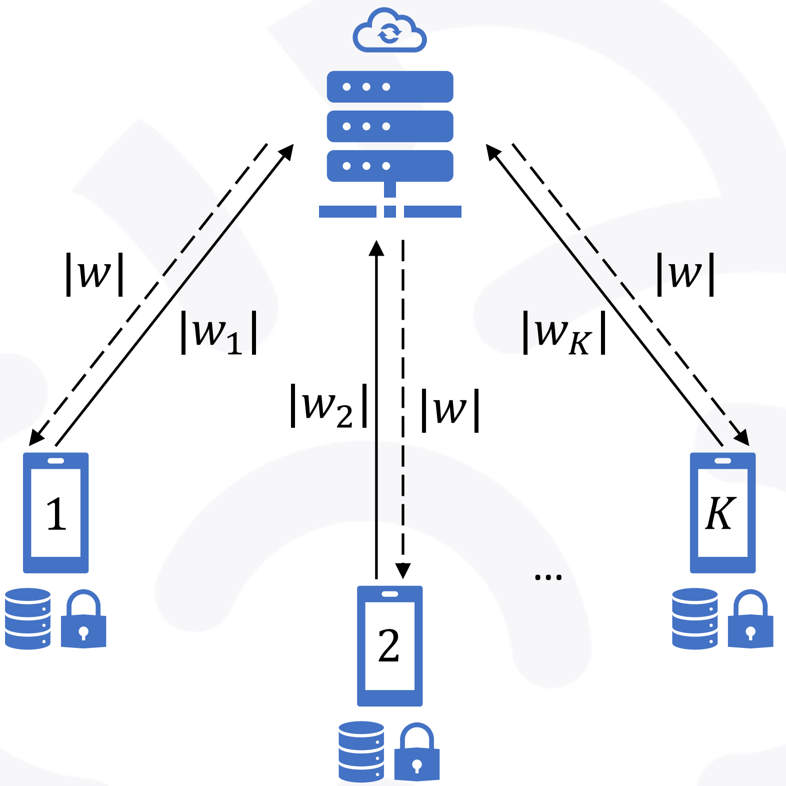
Reduce the number of bits required for the update with discretization

- Less Parameters

Select and design tiny ML models to be trained in the federation

W. Y. B. Lim *et al.*, “Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020

Reduce the Communication Costs



- Quantization

Reduce the number of bits required for the update with discretization

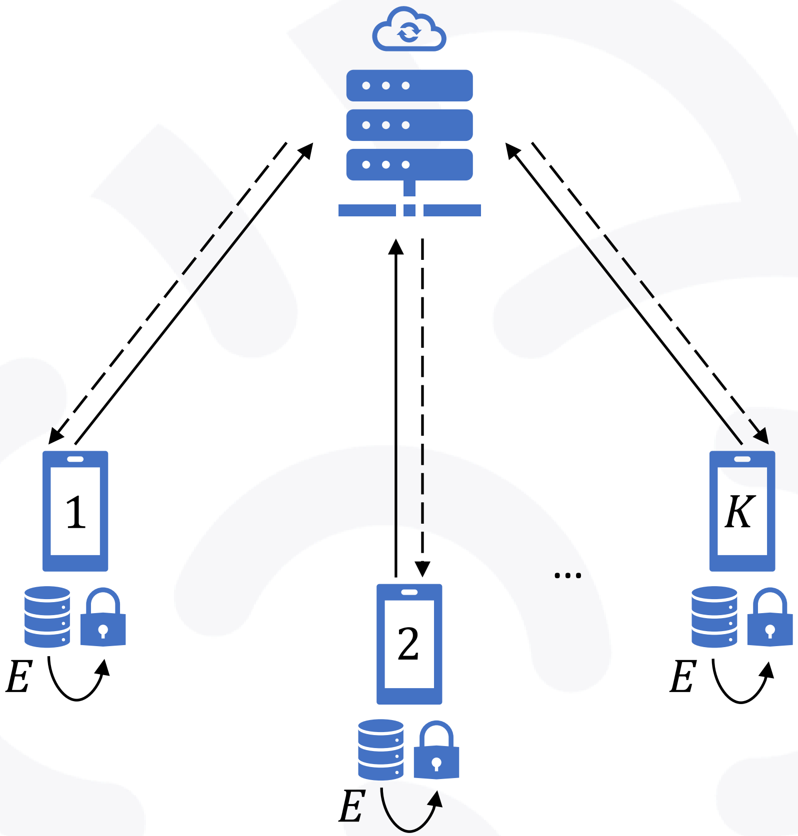
- Less Parameters

Select and design tiny ML models to be trained in the federation

- Importance-based Updating

Selectively send model weights using attention-based importance metrics and dropout

W. Y. B. Lim *et al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020



- Increase local computation

By increasing E, the learning process involves less iterations; this, however, may make convergence harder

W. Y. B. Lim *et al.*, “Federated Learning in Mobile Edge Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020

AI SPRINT

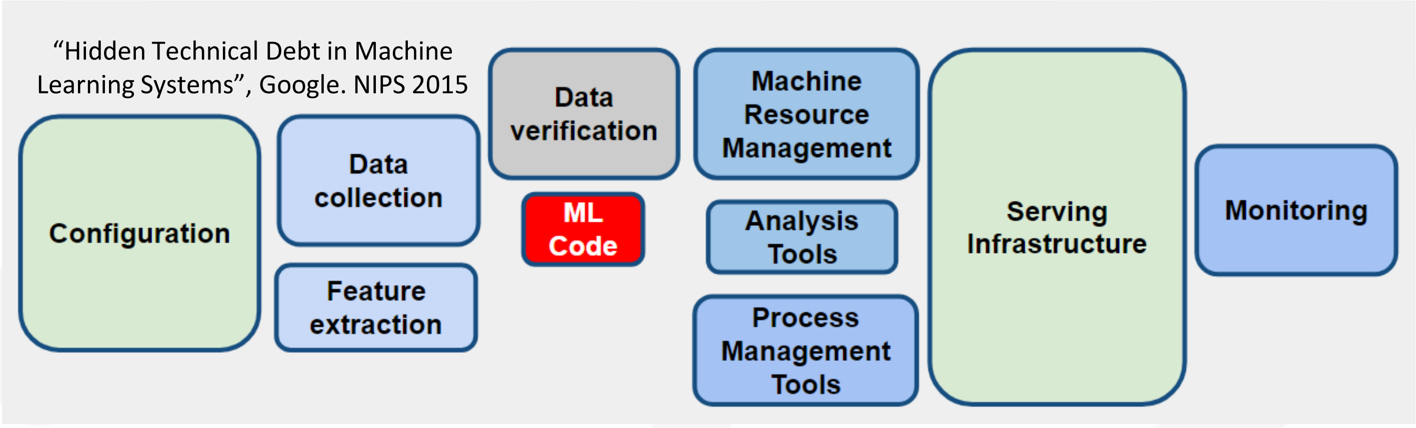
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

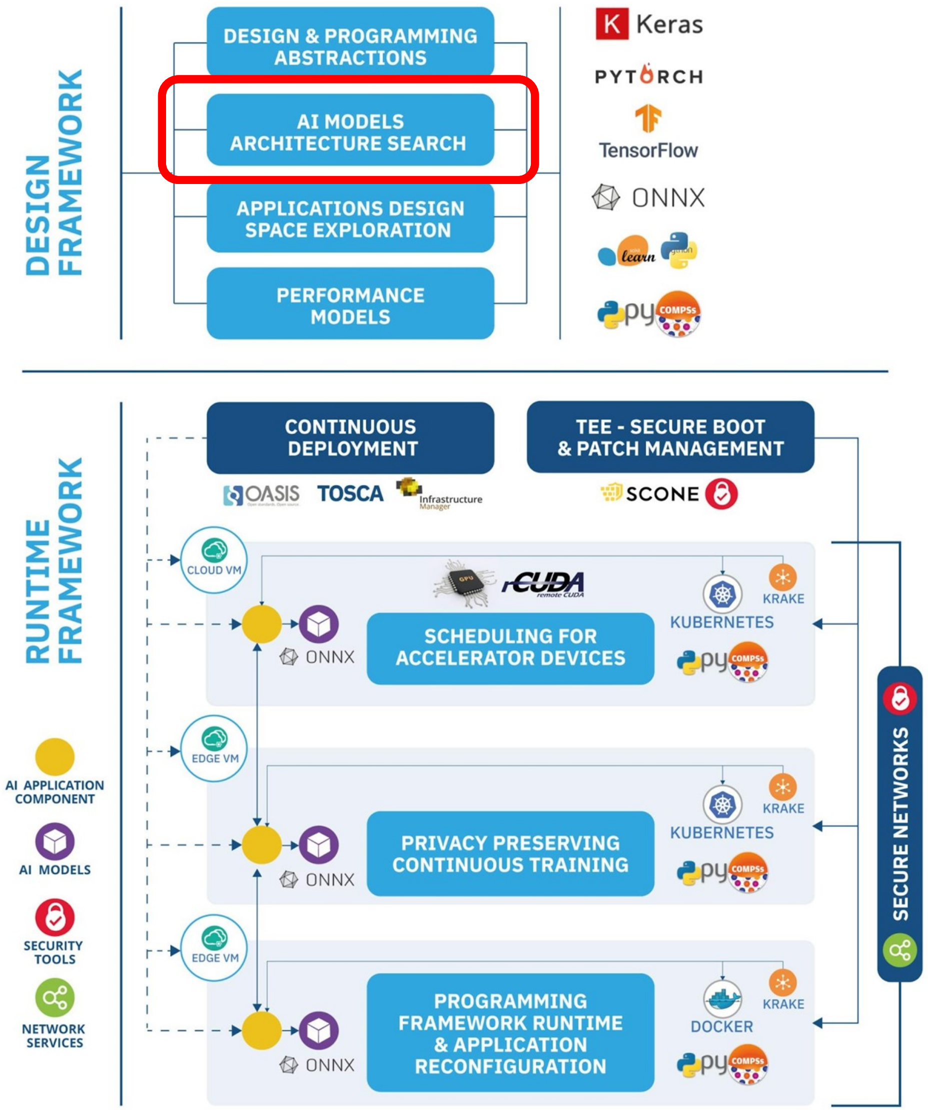
Neural Architecture Search (NAS)

Matteo Matteucci, Politecnico di Milano

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

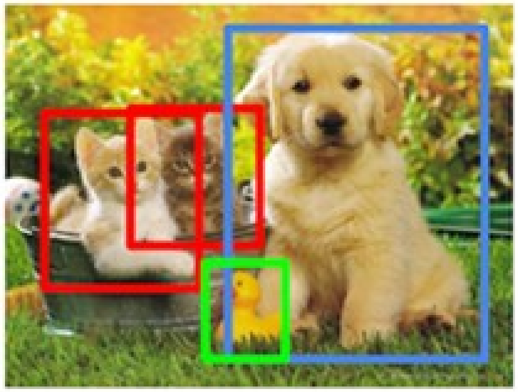
Neural Architecture Search and AI-SPRINT





What is computer vision and why do we care?

How many animals are there in this image?



CAT, DOG, DUCK

Multi-label image classification

Classify multiple objects

Is this a dog?

Image classification

Classify object

Where are the animals in this image?

Object Detection

Bounding Box

Which pixels belong to which object?

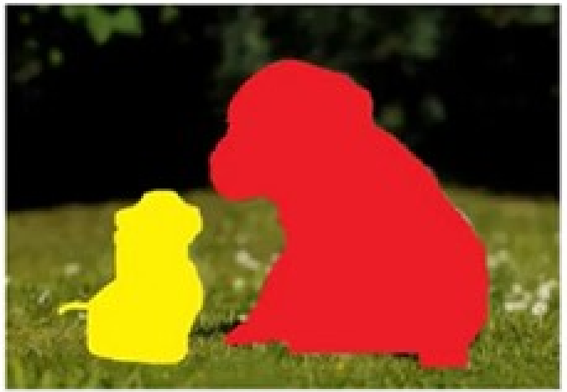
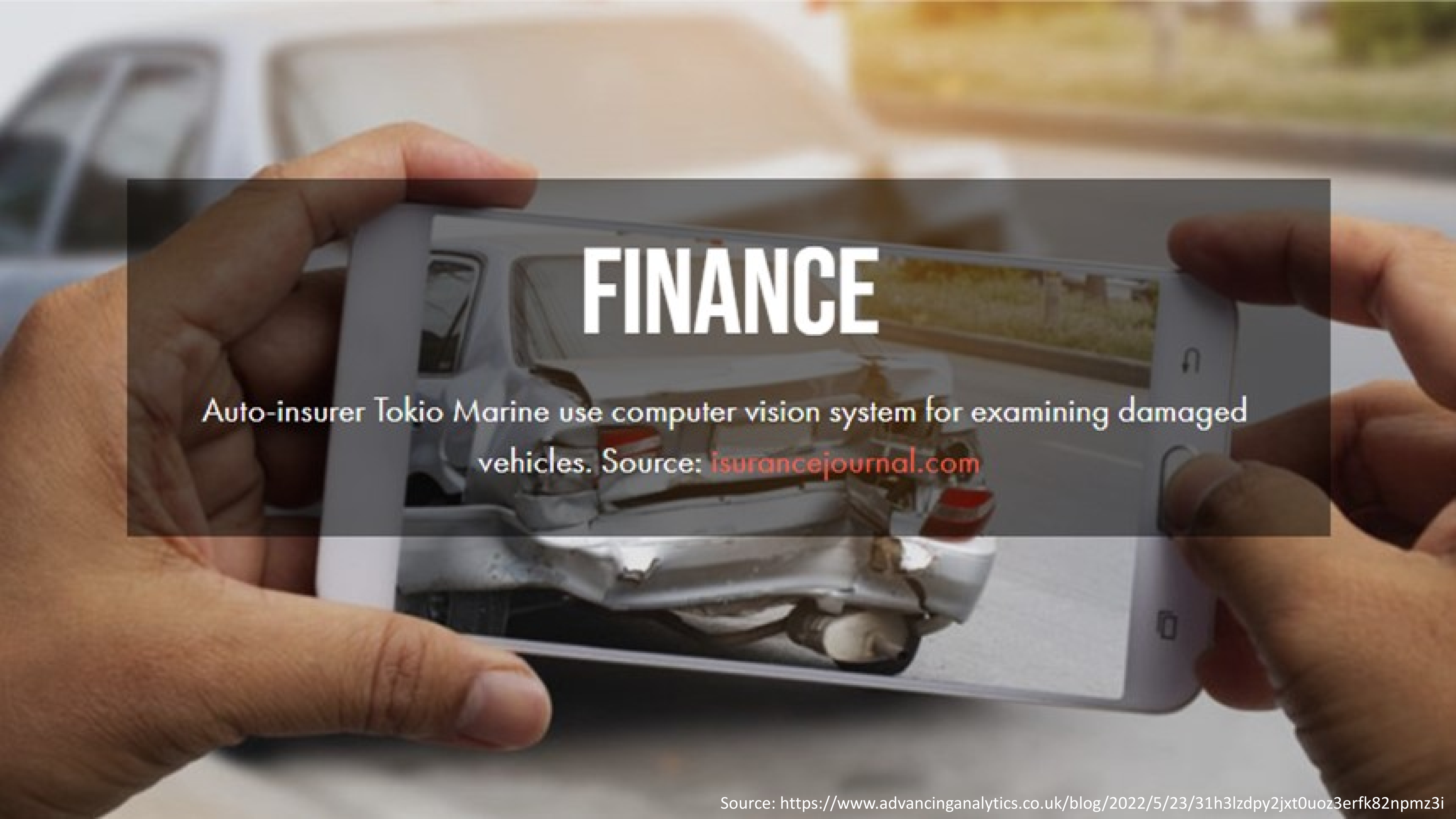


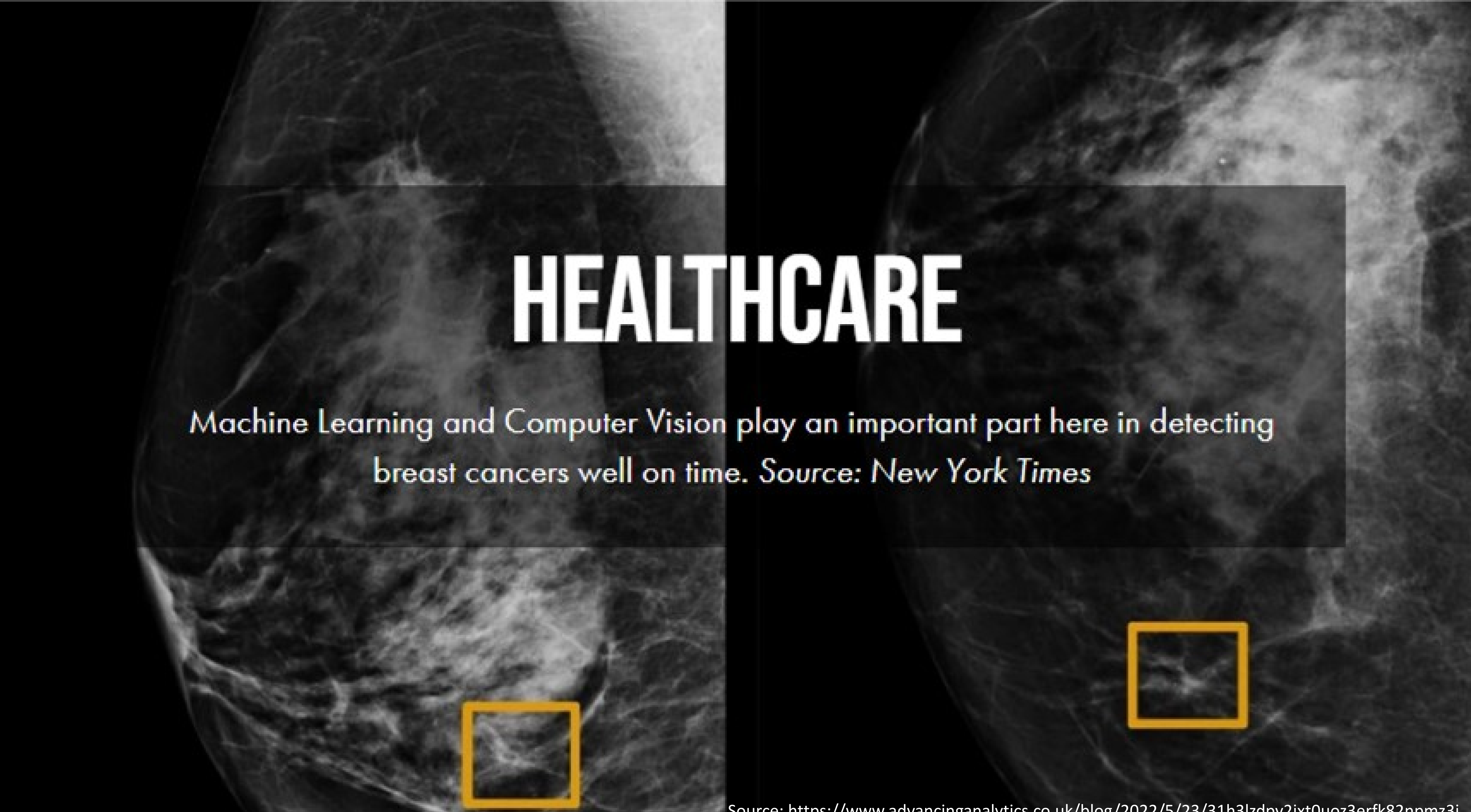
Image Segmentation

Outline of the object

Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

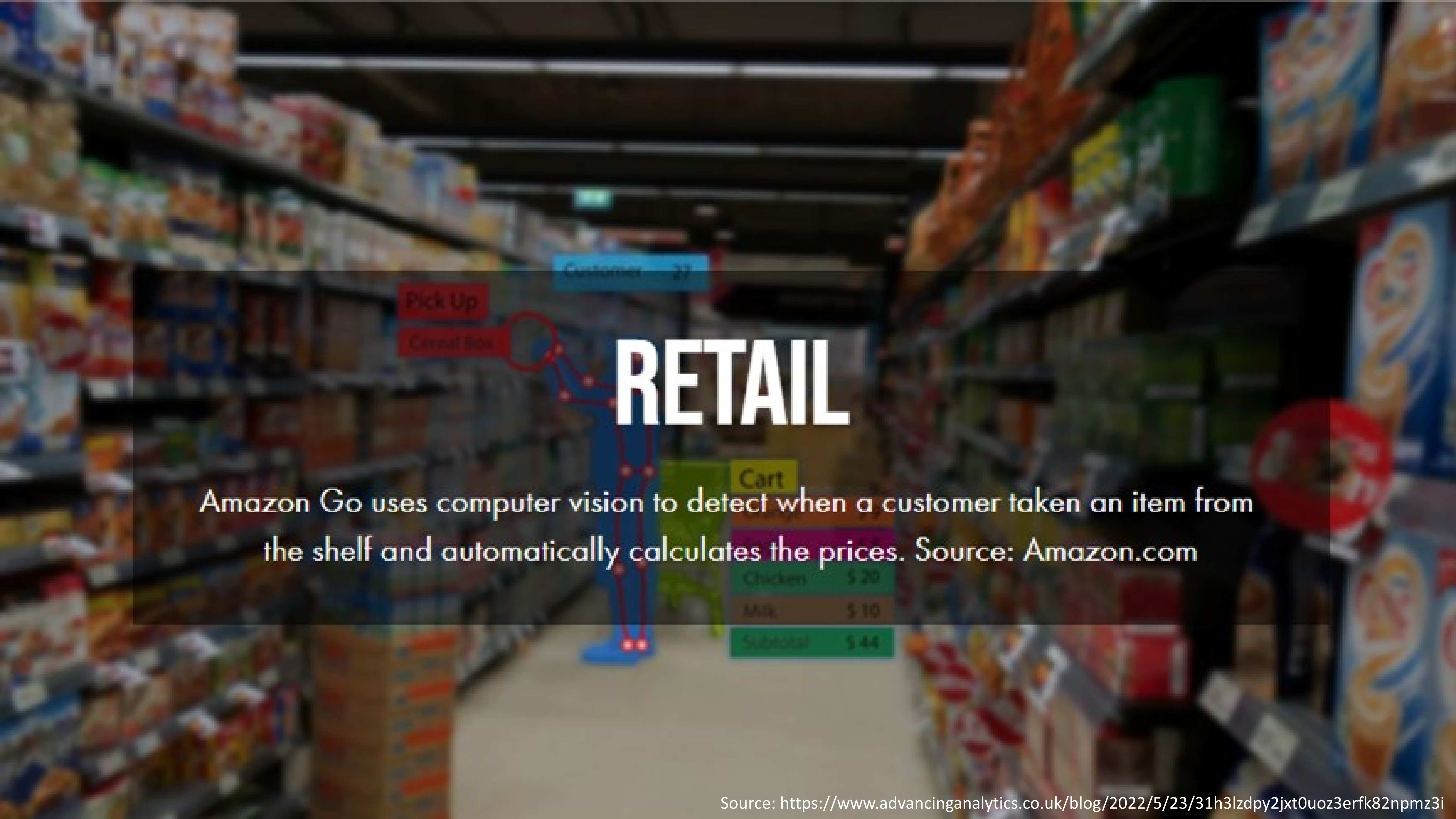


Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i



Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

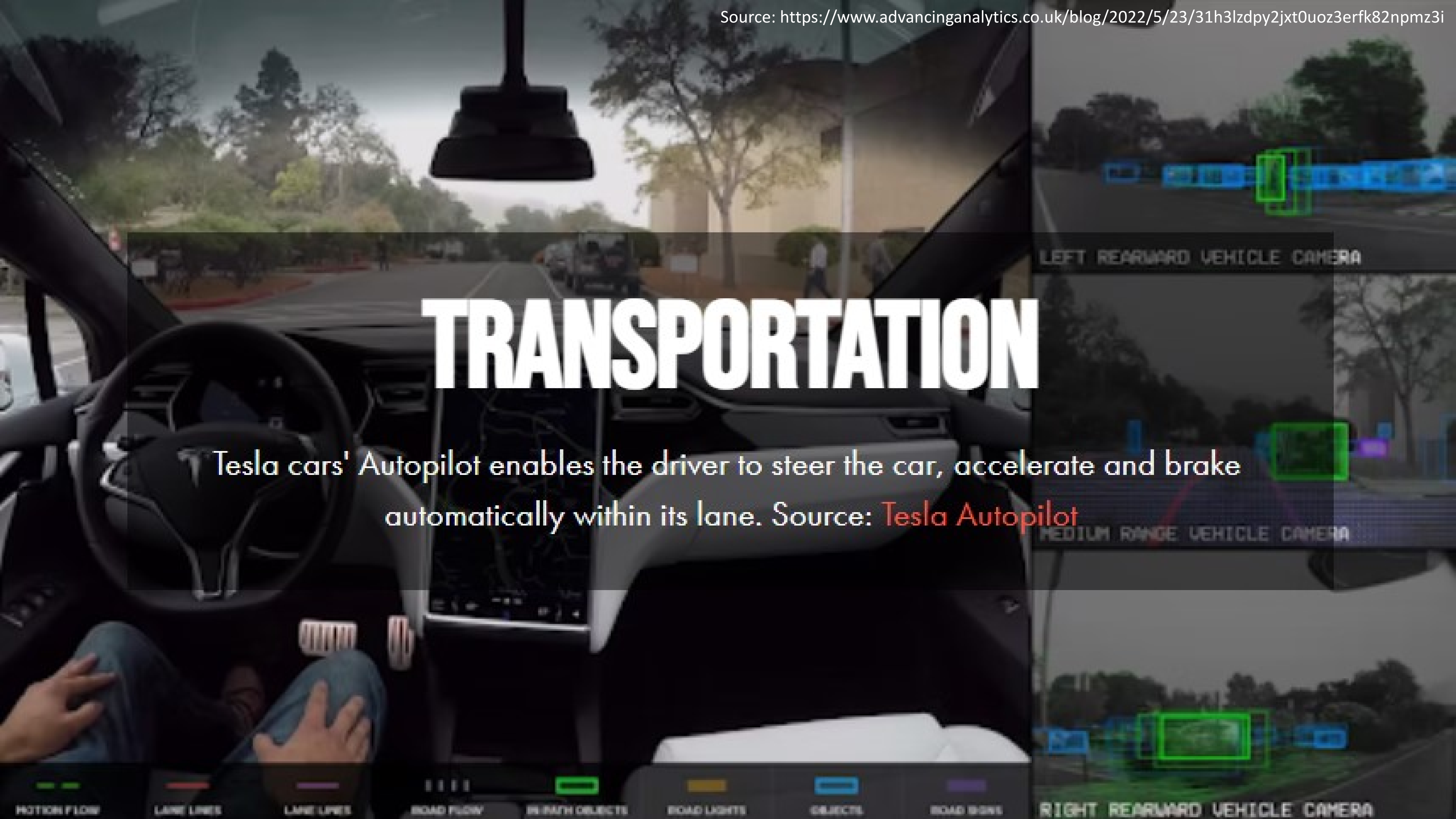


RETAIL

Amazon Go uses computer vision to detect when a customer taken an item from the shelf and automatically calculates the prices. Source: Amazon.com

Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

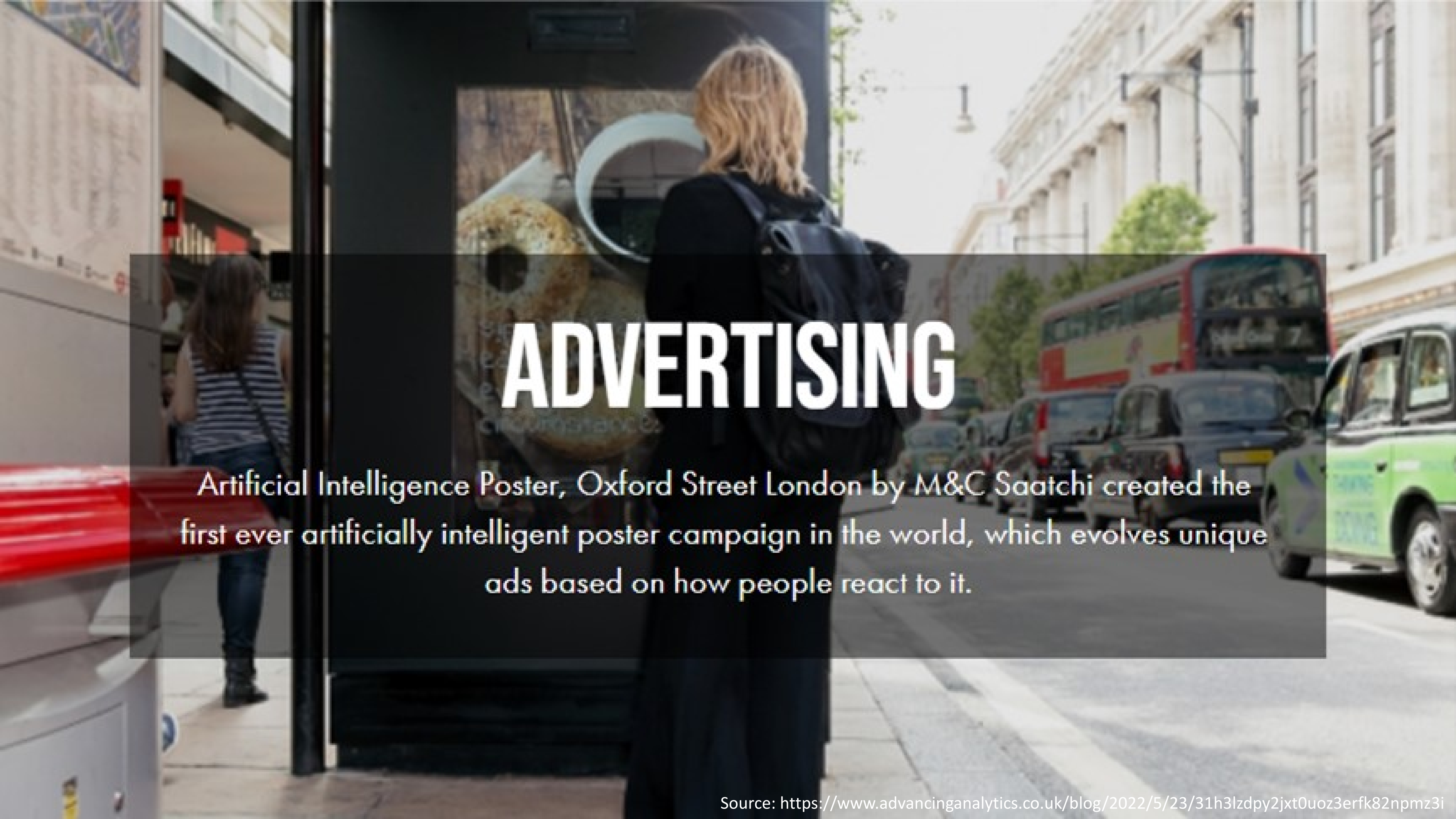
Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i



TRANSPORTATION

Tesla cars' Autopilot enables the driver to steer the car, accelerate and brake automatically within its lane. Source: Tesla Autopilot

Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

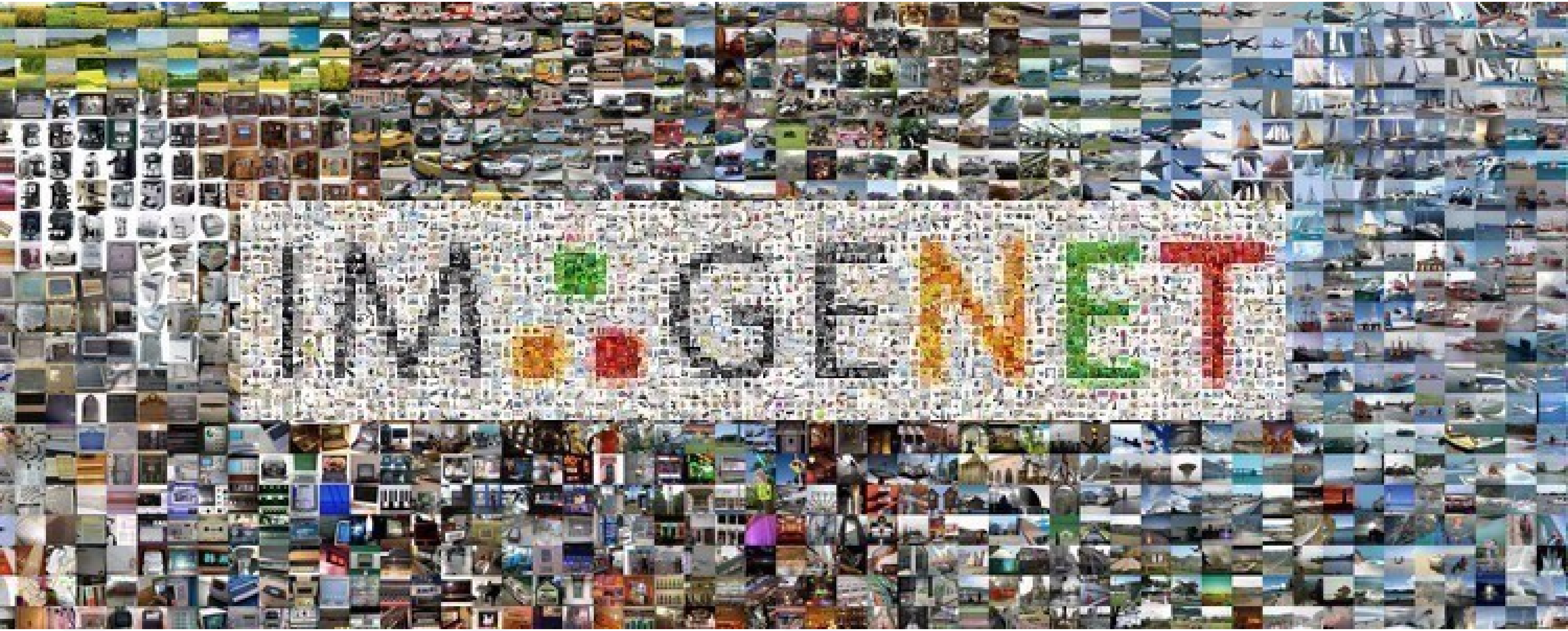


ADVERTISING

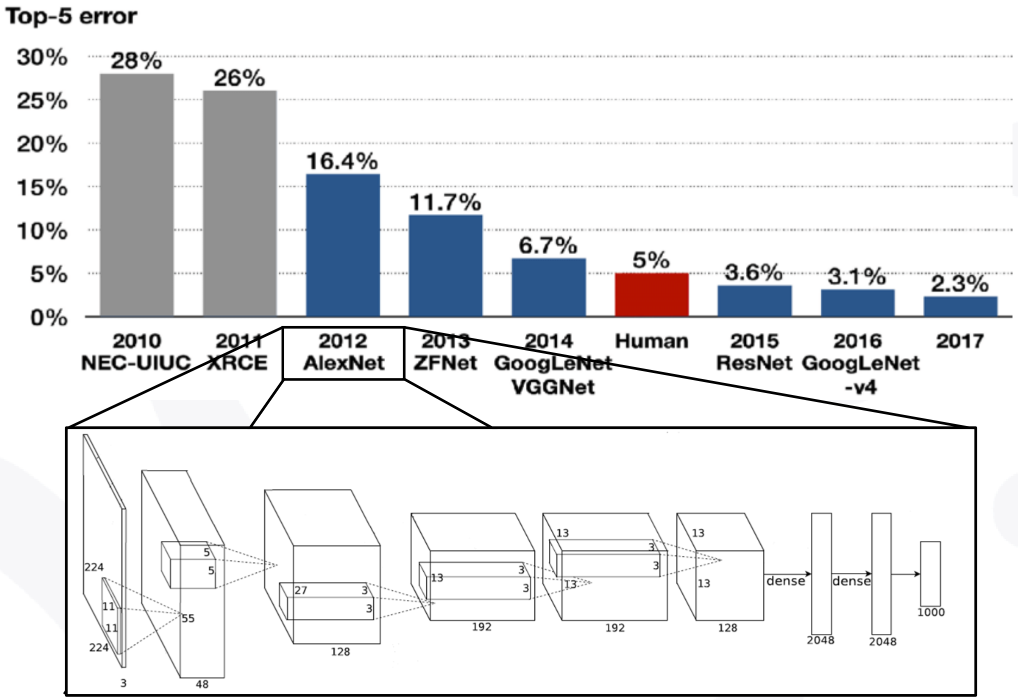
Artificial Intelligence Poster, Oxford Street London by M&C Saatchi created the first ever artificially intelligent poster campaign in the world, which evolves unique ads based on how people react to it.

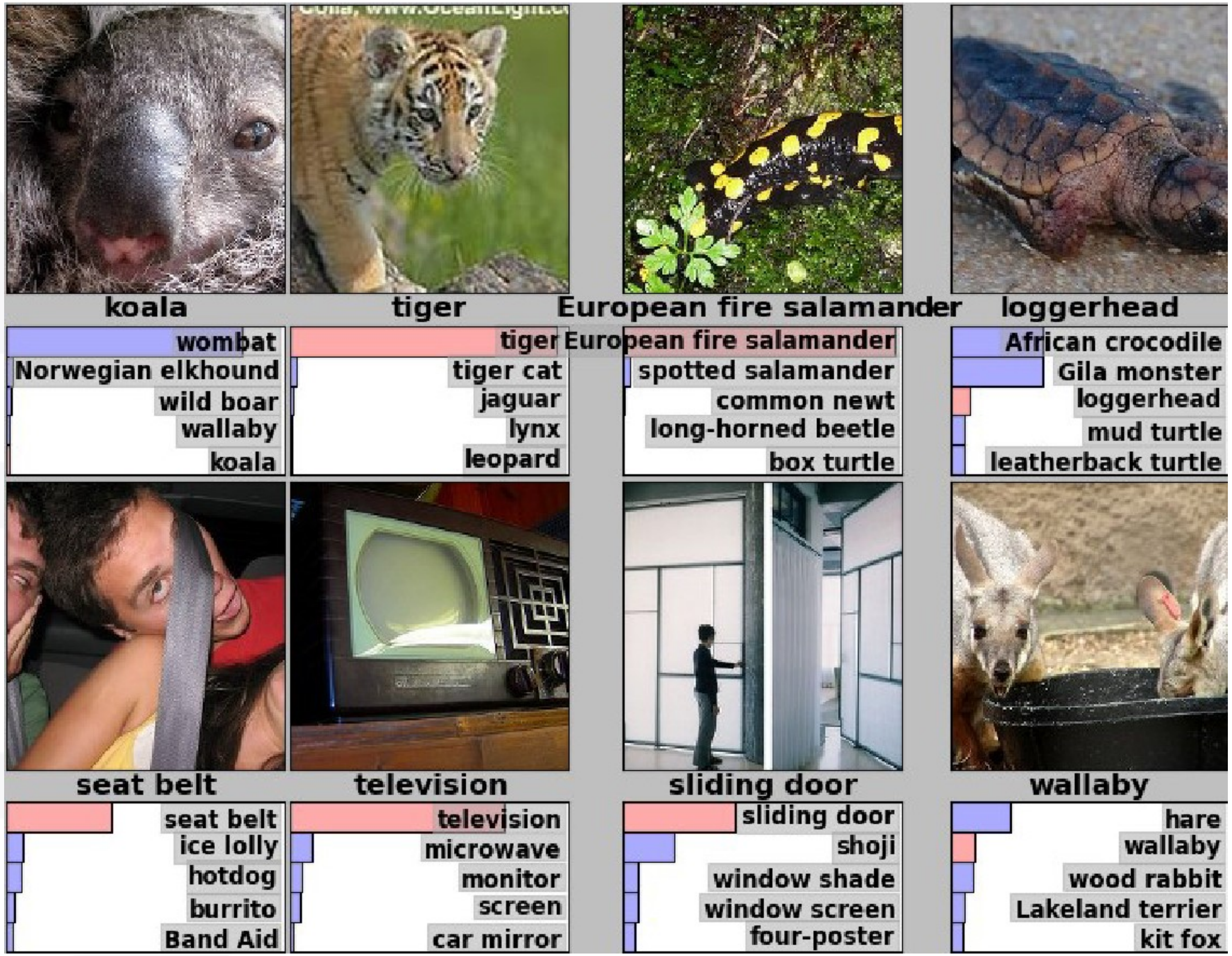
Source: https://www.advancinganalytics.co.uk/blog/2022/5/23/31h3lzdpy2jxt0uoz3erfk82npmz3i

It all started with ImageNet!

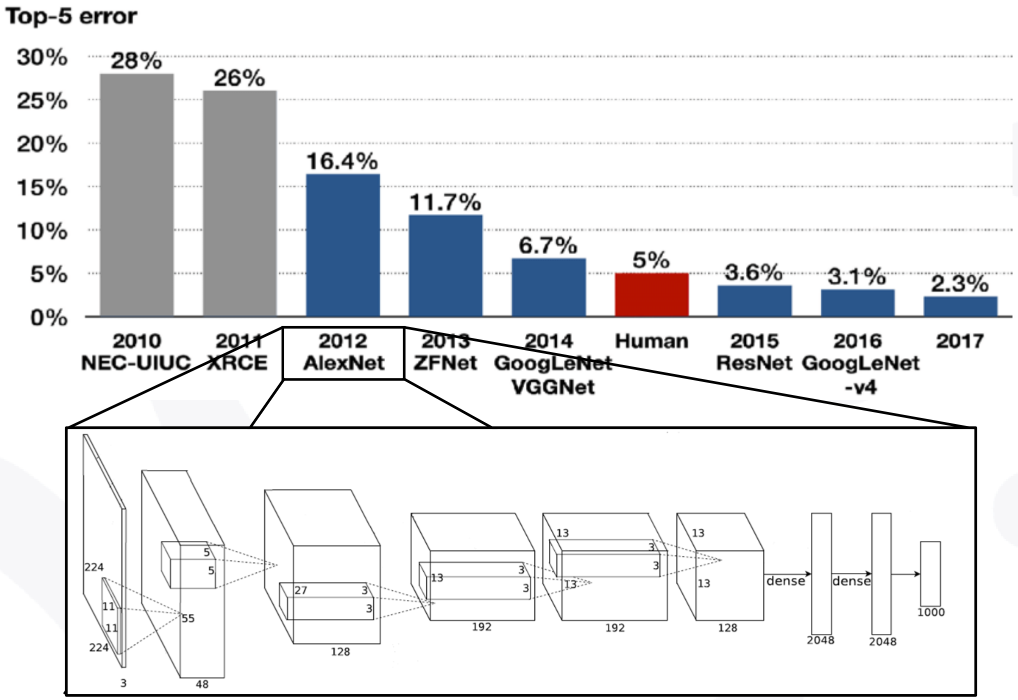


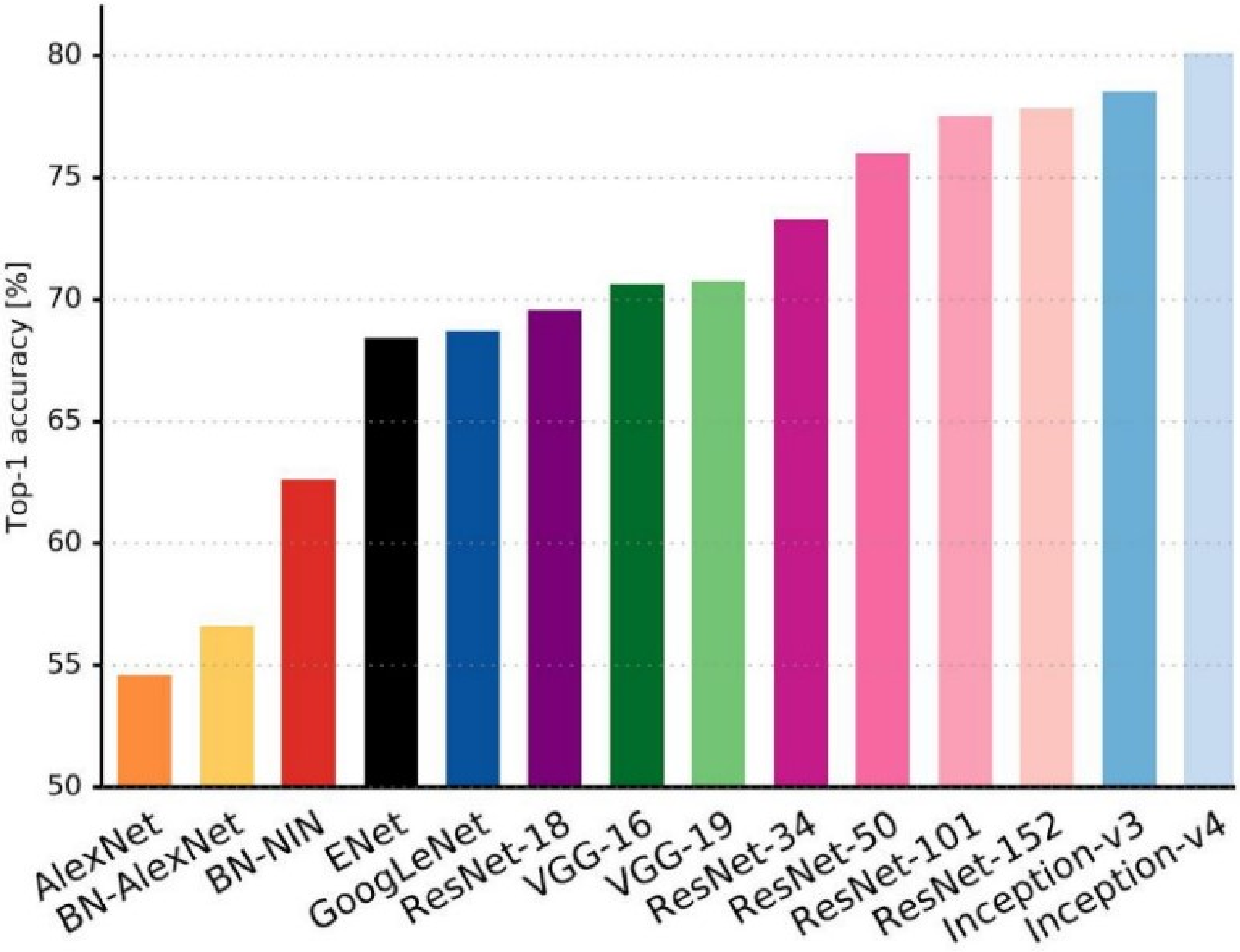
It all started with ImageNet!





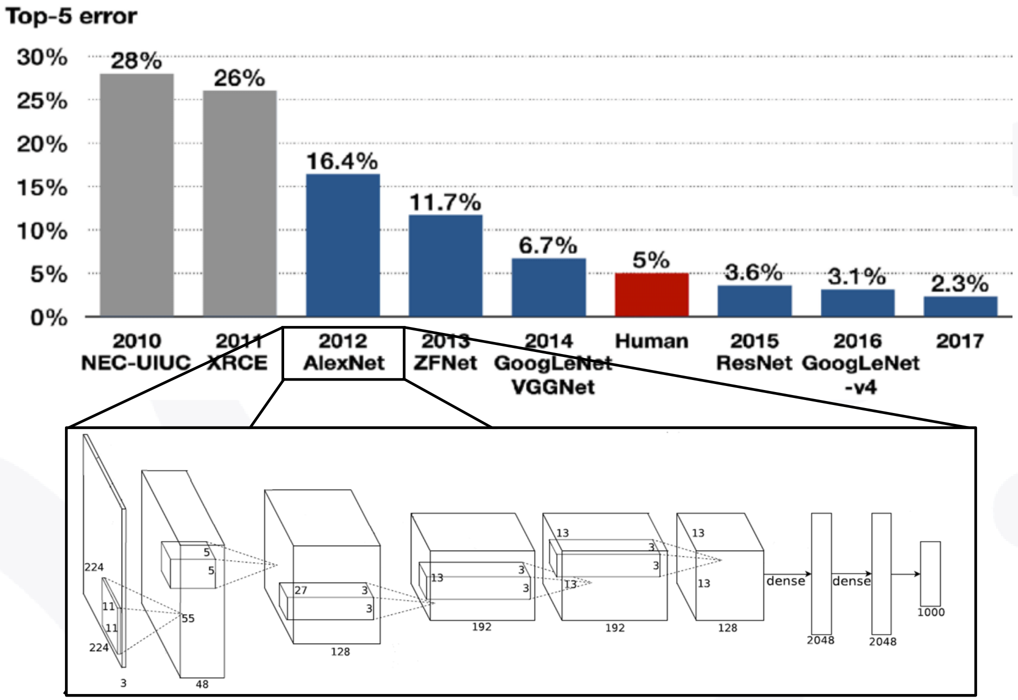
It all started with ImageNet!

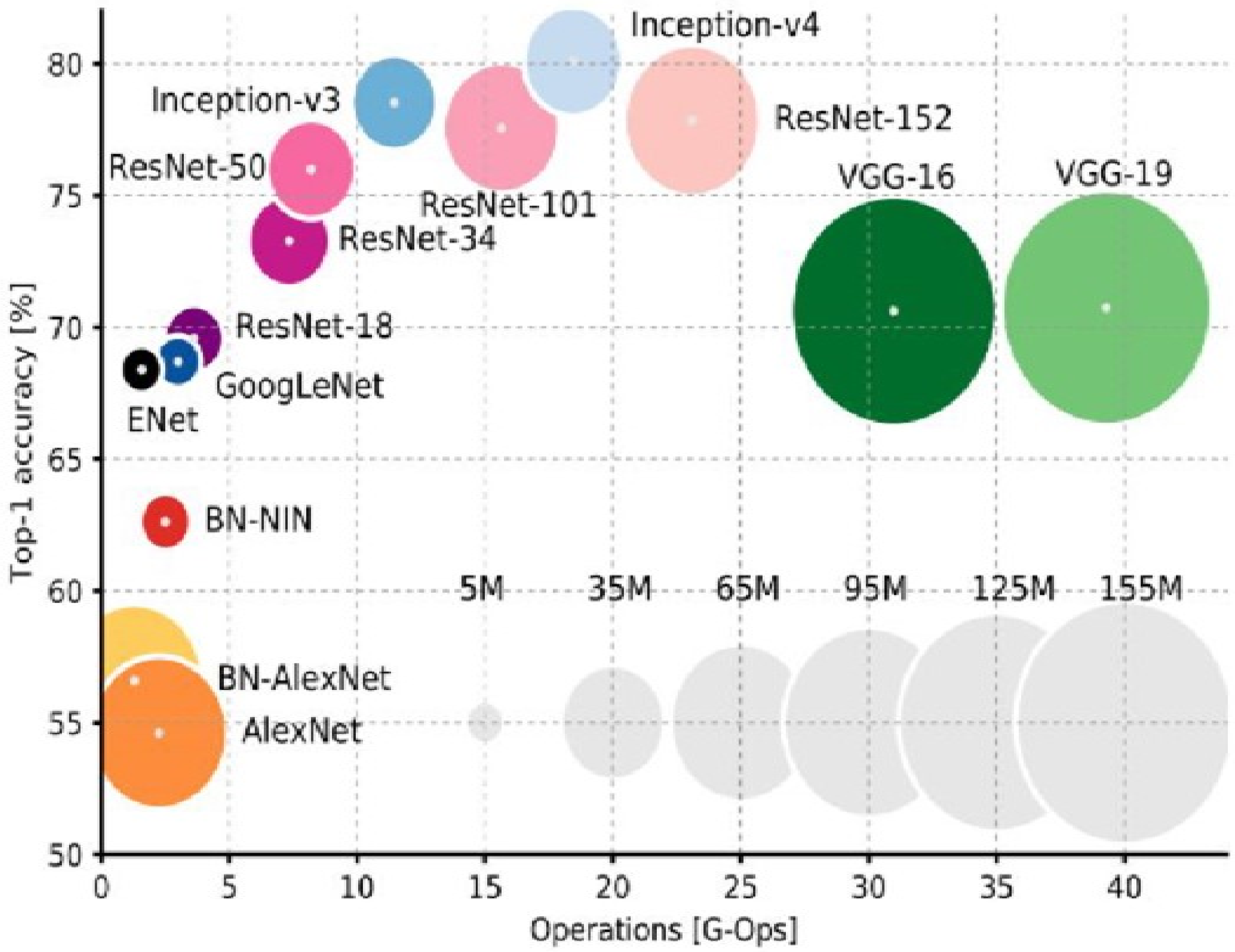




Canziani et al. (2017)

It all started with ImageNet!

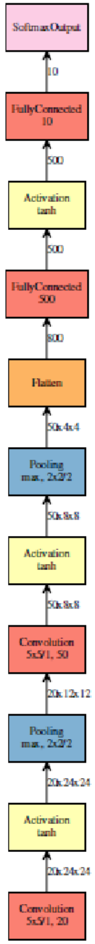




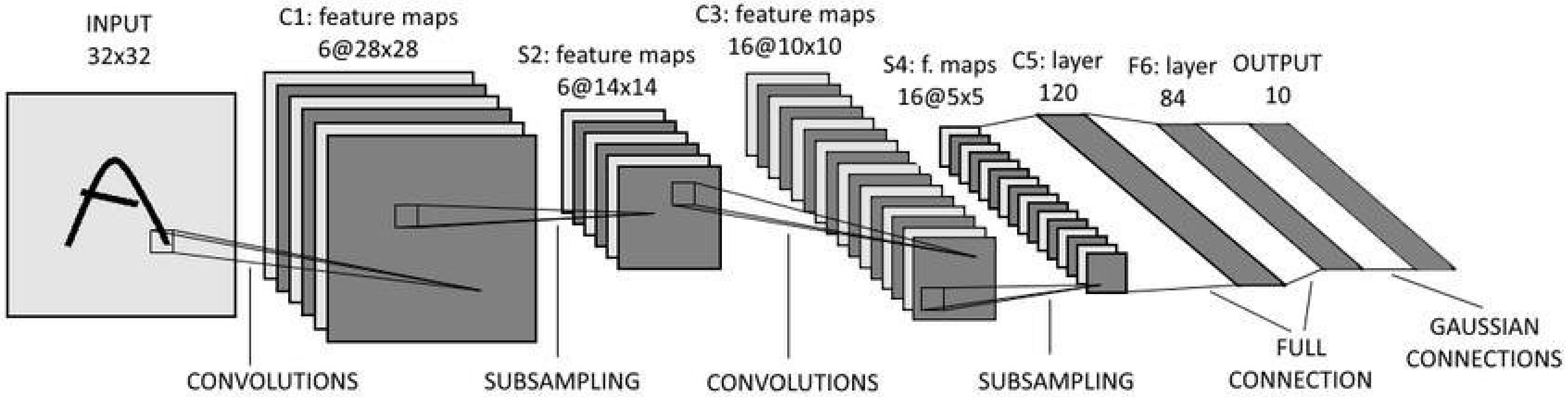
Canziani et al. (2017)

Neural networks keep growing!

LeNet (1998)



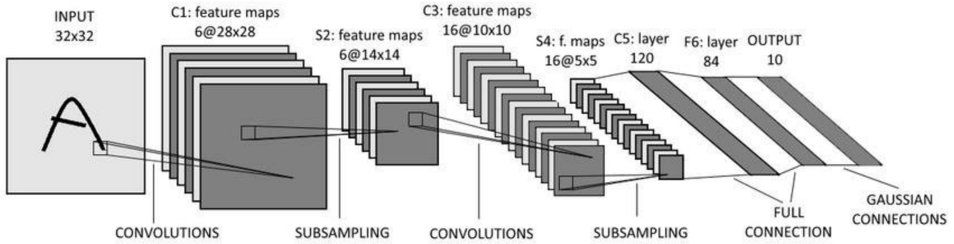
2 convolutional layers
2 fully connected layers



Neural networks keep growing!

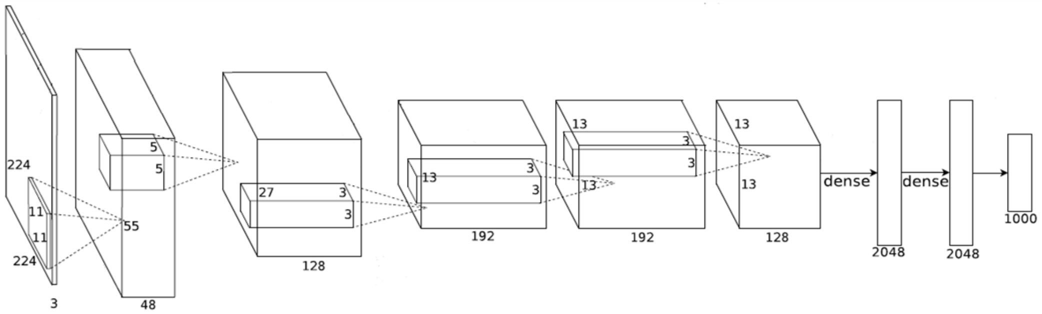
LeNet (1998)

2 convolutional layers
2 fully connected layers



AlexNet (2012)

5 convolutional layers
3 fully connected layers

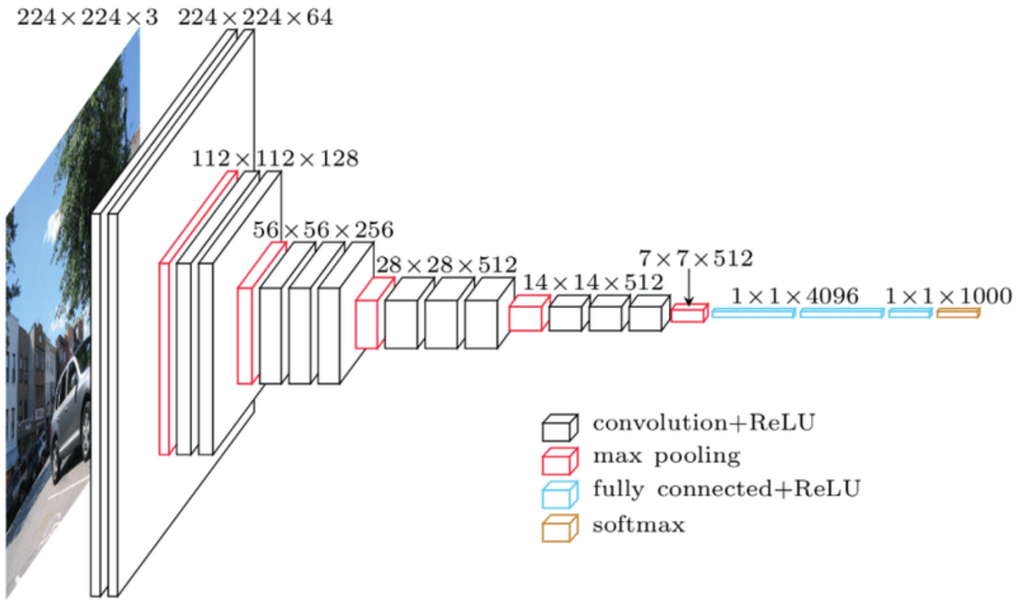


Neural networks keep growing!

LeNet (1998)

AlexNet (2012)

VGGNet-M (2013)

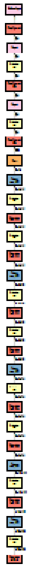


Neural networks keep growing!

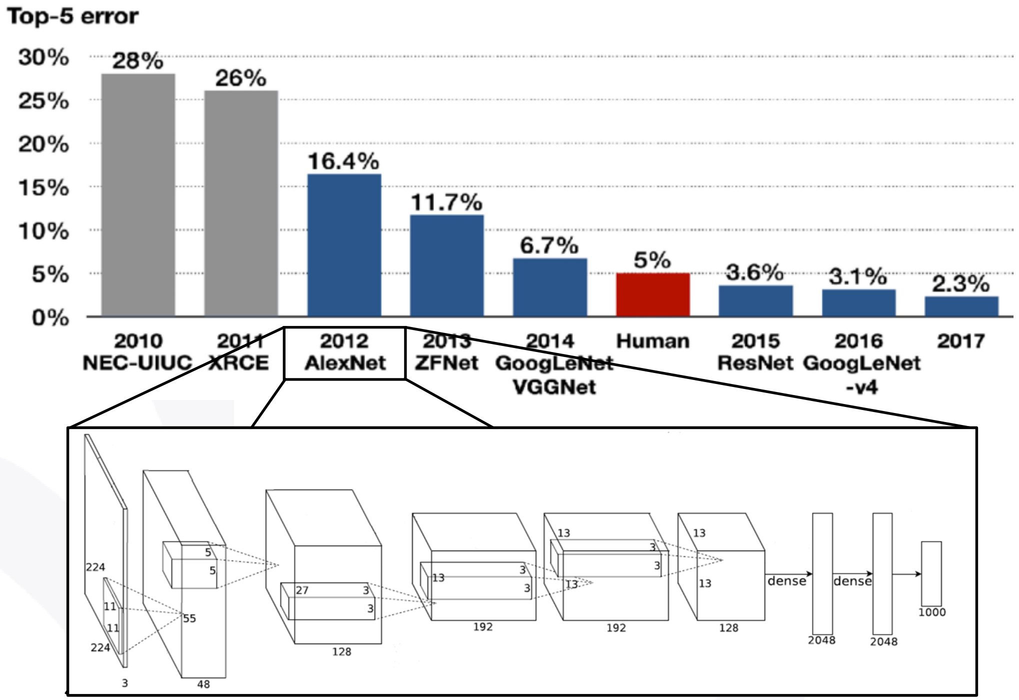
LeNet (1998)

AlexNet (2012)

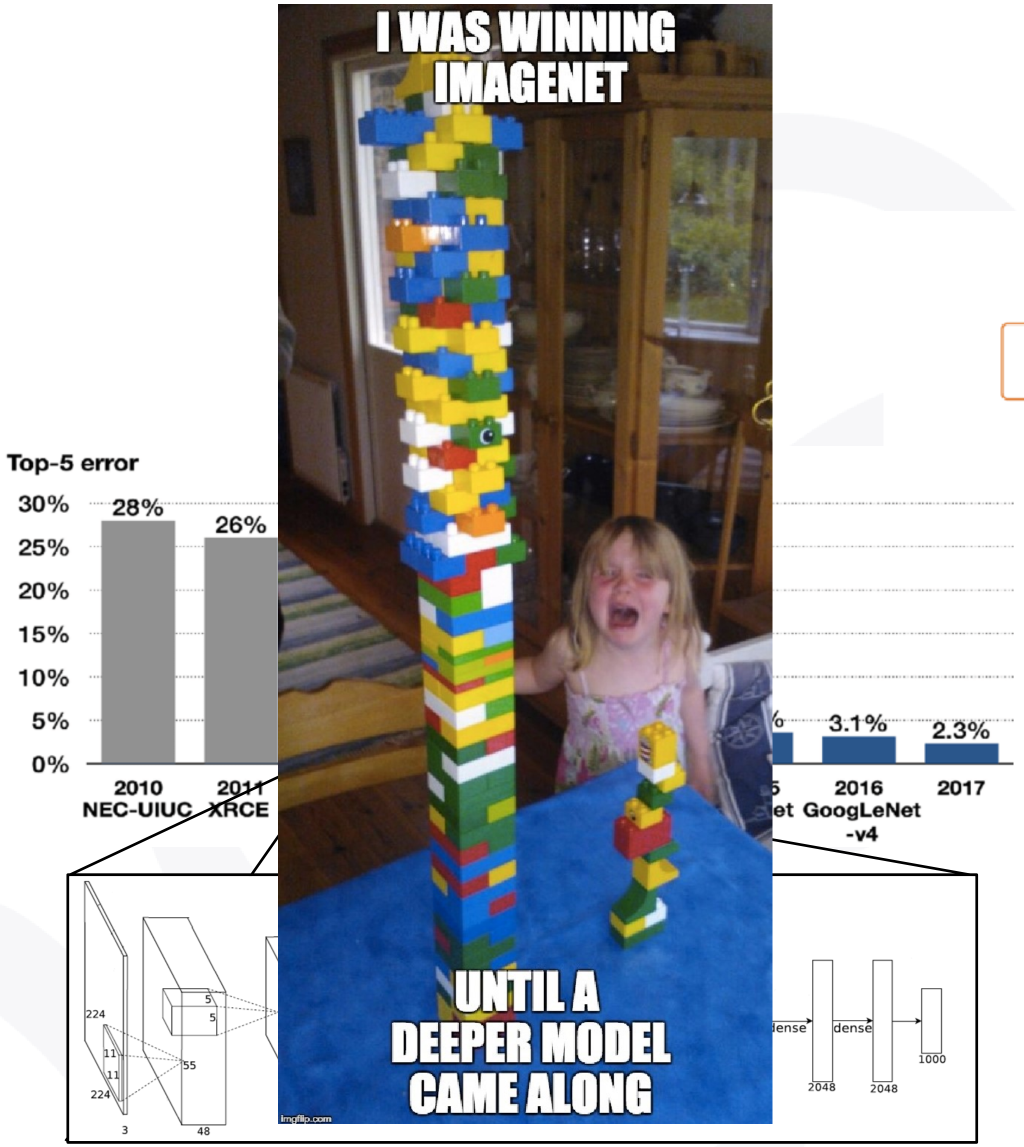
VGGNet-M (2013)

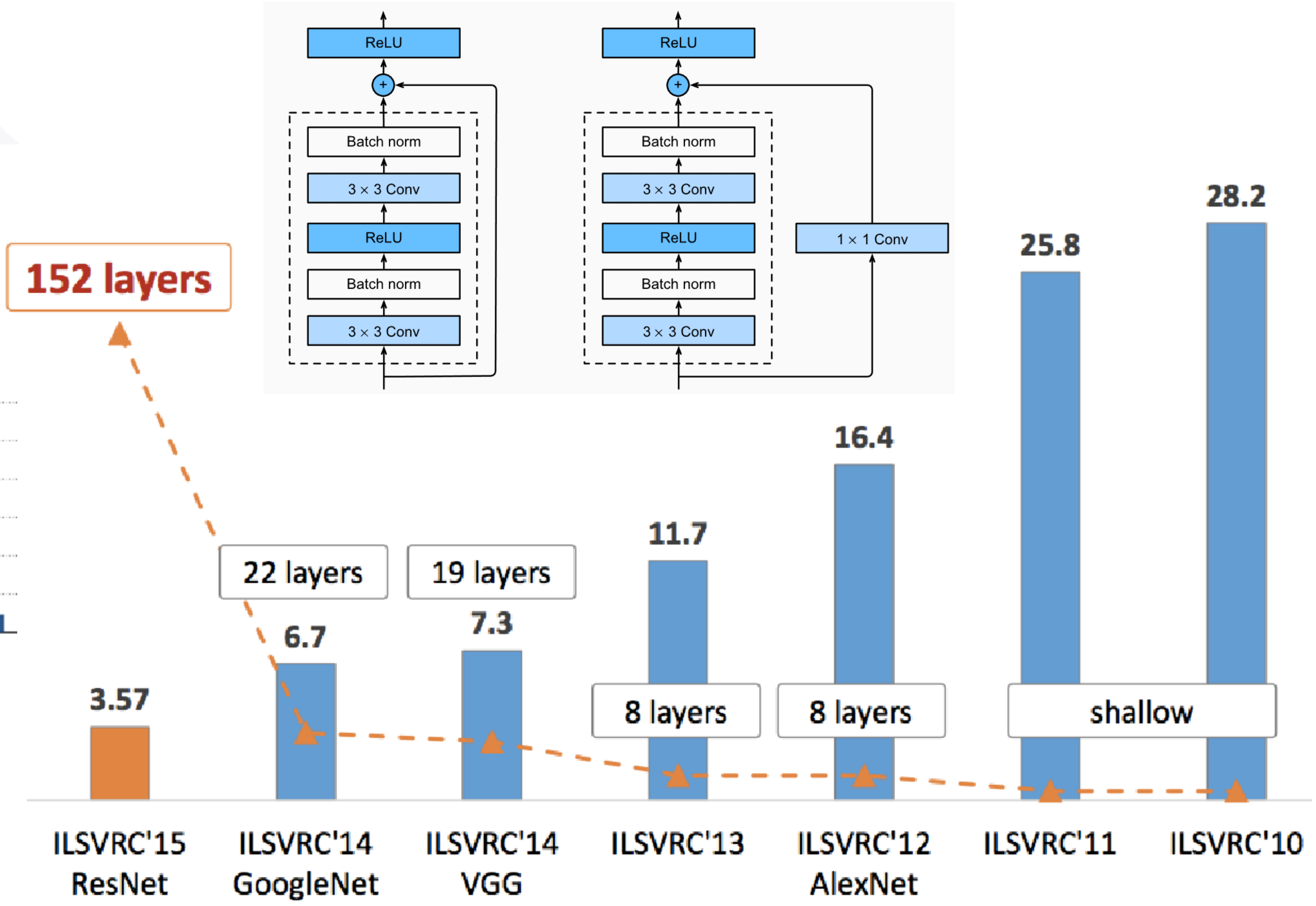


GoogLeNet (2014)



Neural networks keep growing!





He et al. (2015) and Szegedy et al. (2017)

Neural networks keep growing!

AI SPRINT — AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

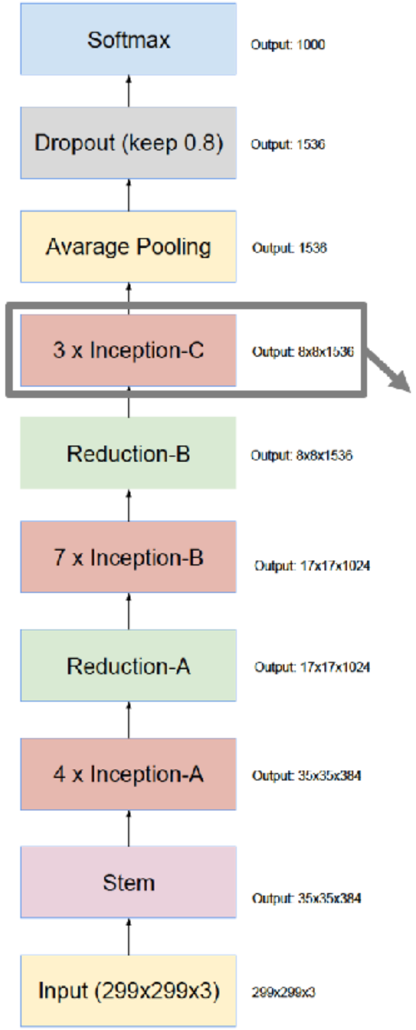
arXiv 2016
Inception-ResNet model
C. Szegedy et al.
GoogleNet -> Inception -> Inception_v4
Top-5: 3.1

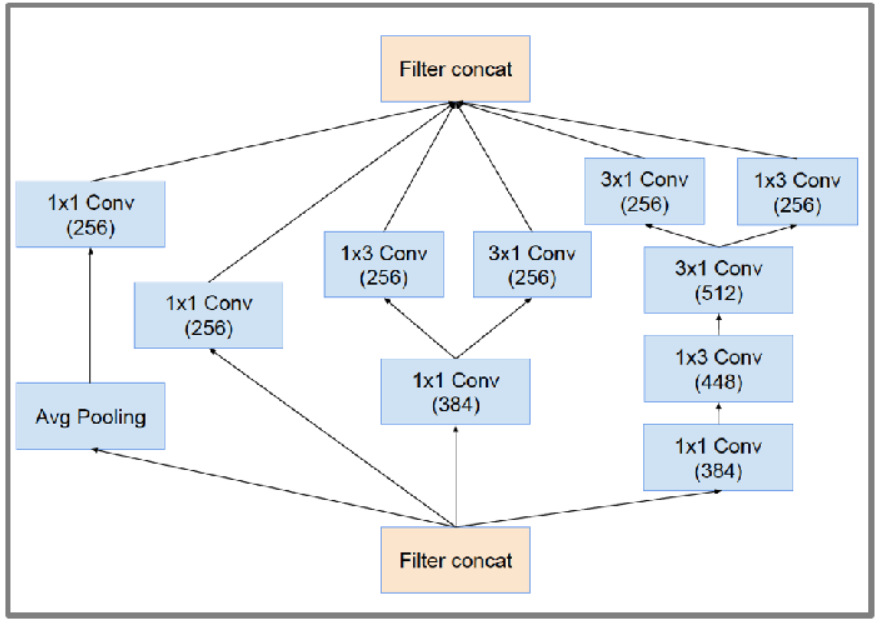
CVPR 2016
Residual Network
K. He et al.
VGG -> PReLU (20 layers) -> ResN
Top-5: 4.49

ICLR 2015, CVPR 2015
VGG, GoogLeNet
K. Simonyan; C. Szegedy
Classification top-5 error: 9.33, 9.15

2012 - 2014
Overfeat, Clarifi, Net-in-Ne
Variants of AlexNet

NIPS 2012
AlexNet
A. Krizhevsky et al.
Milestone. First time to apply CNN in large-scale dataset (top-5: 15.3).





Zoom-in description of Inception-C block.

Compared with the original GoogleNet, it has more convolution outputs with smaller filter size before feature concatenation.

Inception-v4 network

Neural networks keep growing!

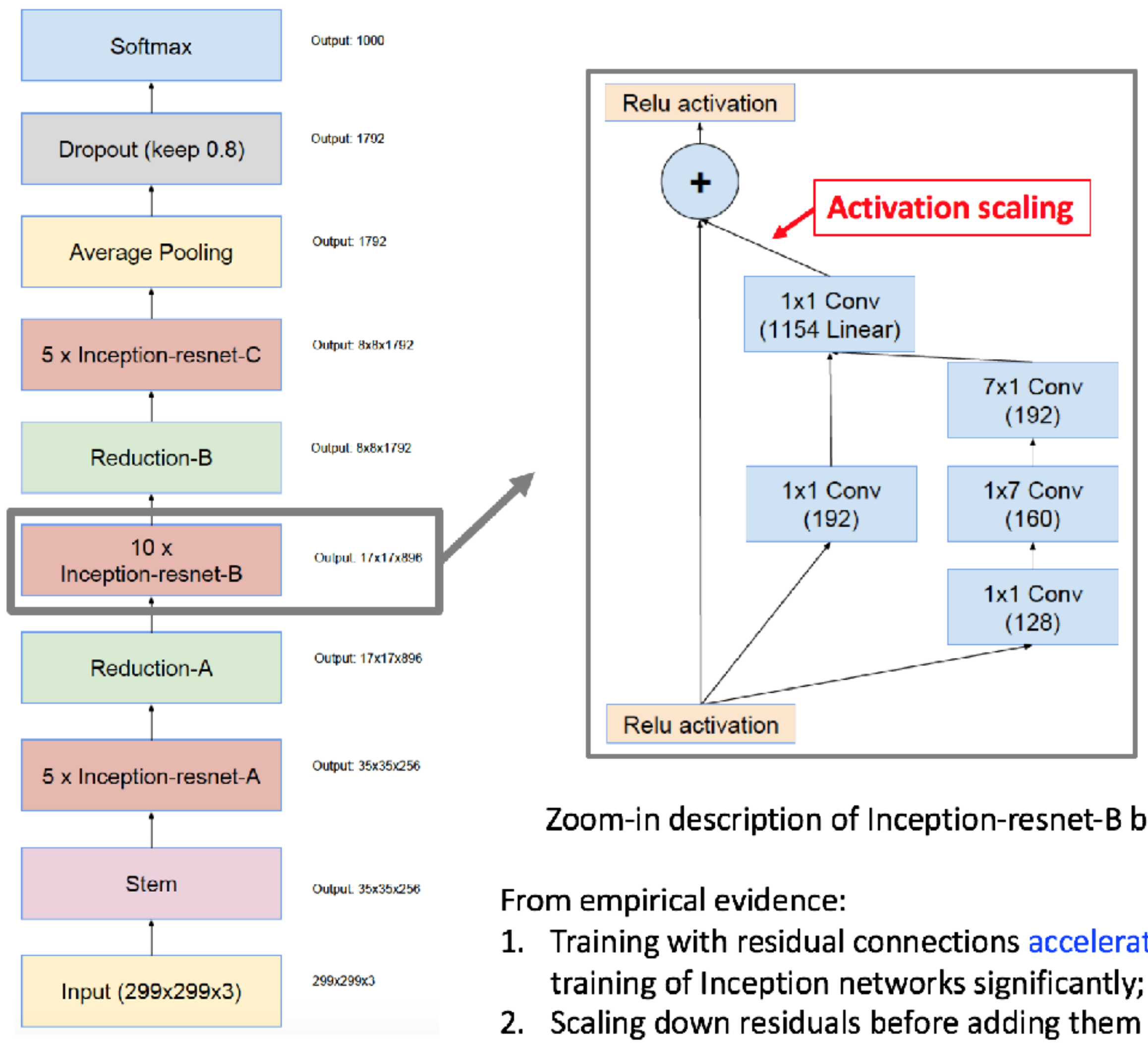
arXiv 2016
Inception-ResNet model
C. Szegedy et al.
GoogleNet -> Inception -> Inception_v4
Top-5: 3.1

CVPR 2016
Residual Network
K. He et al.
VGG -> PReLU (20 layers) -> ResNet (2…
Top-5: 4.49

ICLR 2015, CVPR 2015
VGG, GoogLeNet
K. Simonyan; C. Szegedy
Classification top-5 error: 9.33, 9.15

2012 - 2014
Overfeat, Clarifi, Net-in-Net, et…
Variants of AlexNet

NIPS 2012
AlexNet
A. Krizhevsky et al.
Milestone. First time to apply CNN in large-scale dataset (top-5: 15.3).



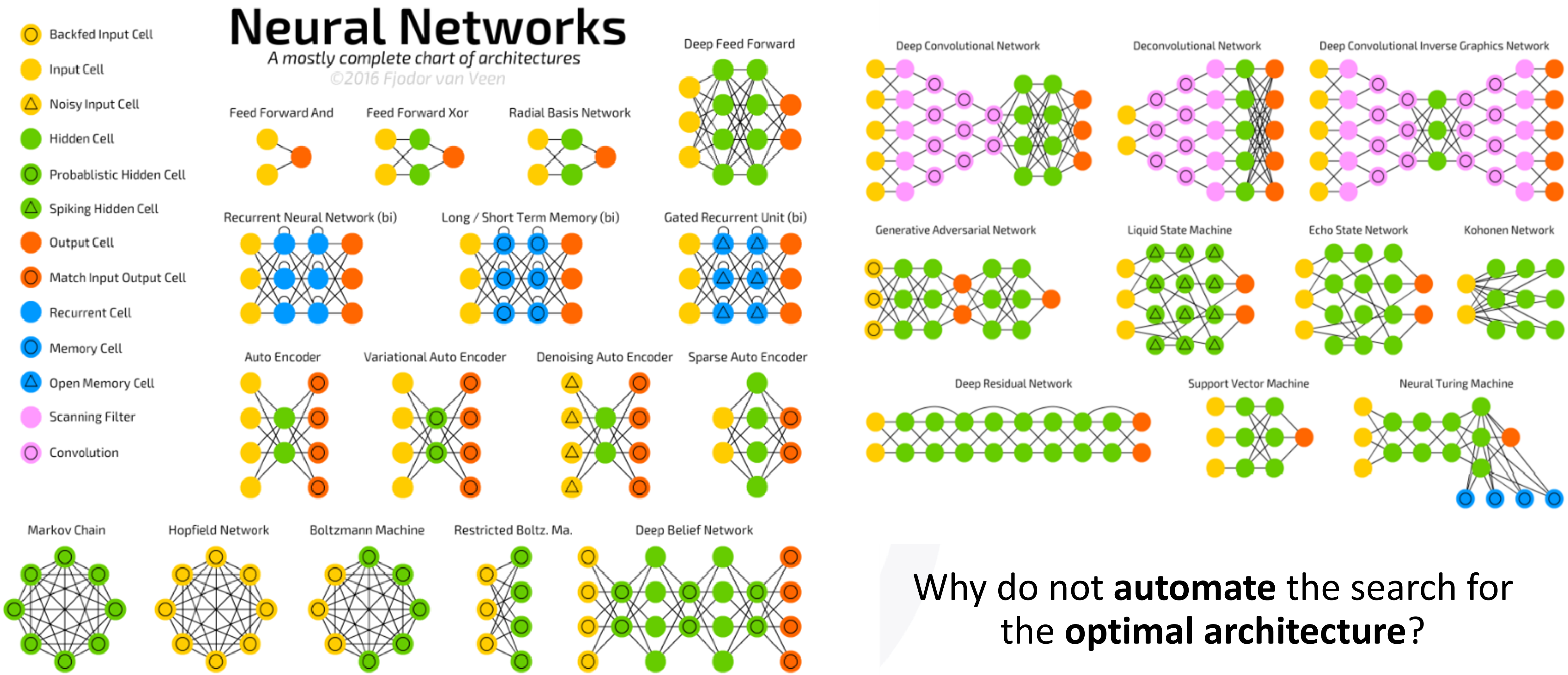
Inception-Resnet v2

Zoom-in description of Inception-resnet-B block.

From empirical evidence:

1. Training with residual connections accelerates the training of Inception networks significantly;
2. Scaling down residuals before adding them to the subsequent layer's activation stabilizes training.

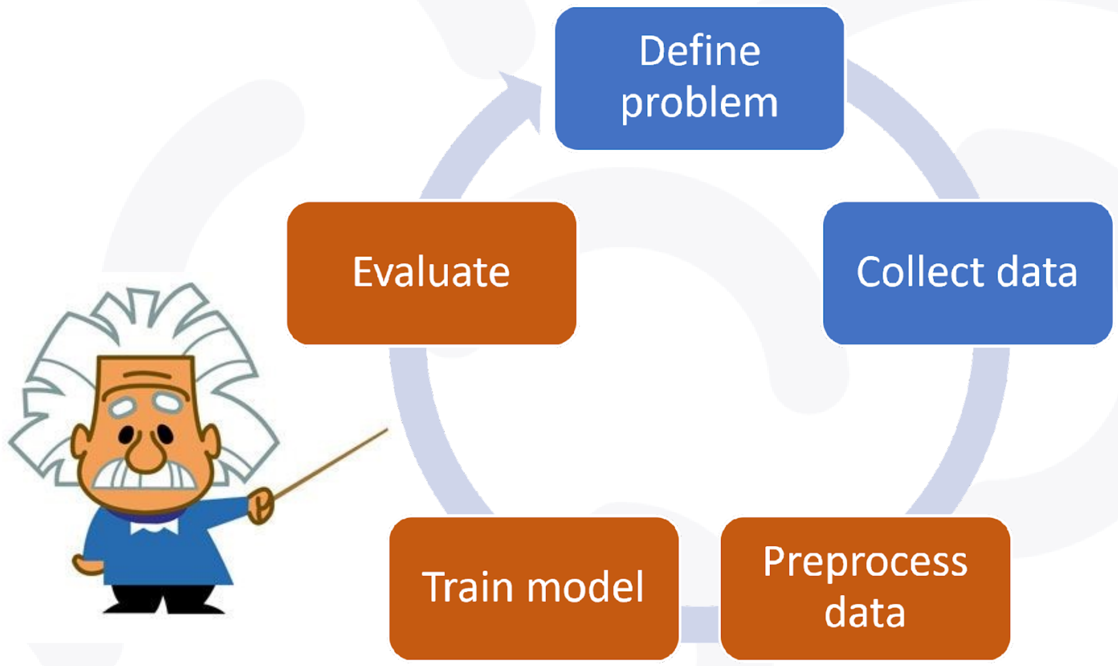
How good are humans in designing neural nets?



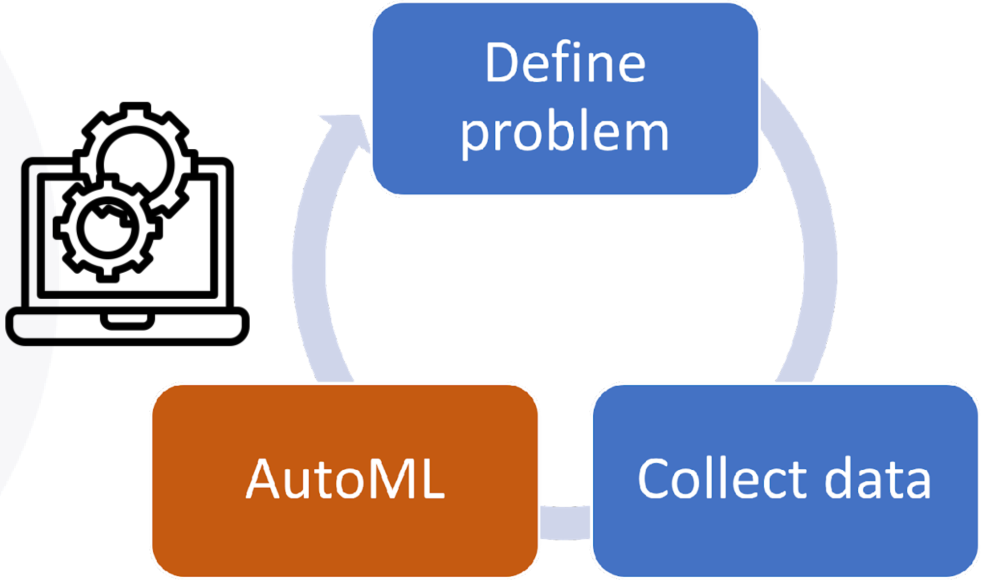
Why do not **automate** the search for the **optimal architecture**?

Automated Machine Learning (**AutoML**), refers to the use of automated processes and techniques to automate various stages of the machine learning pipeline

Traditional ML training workflow

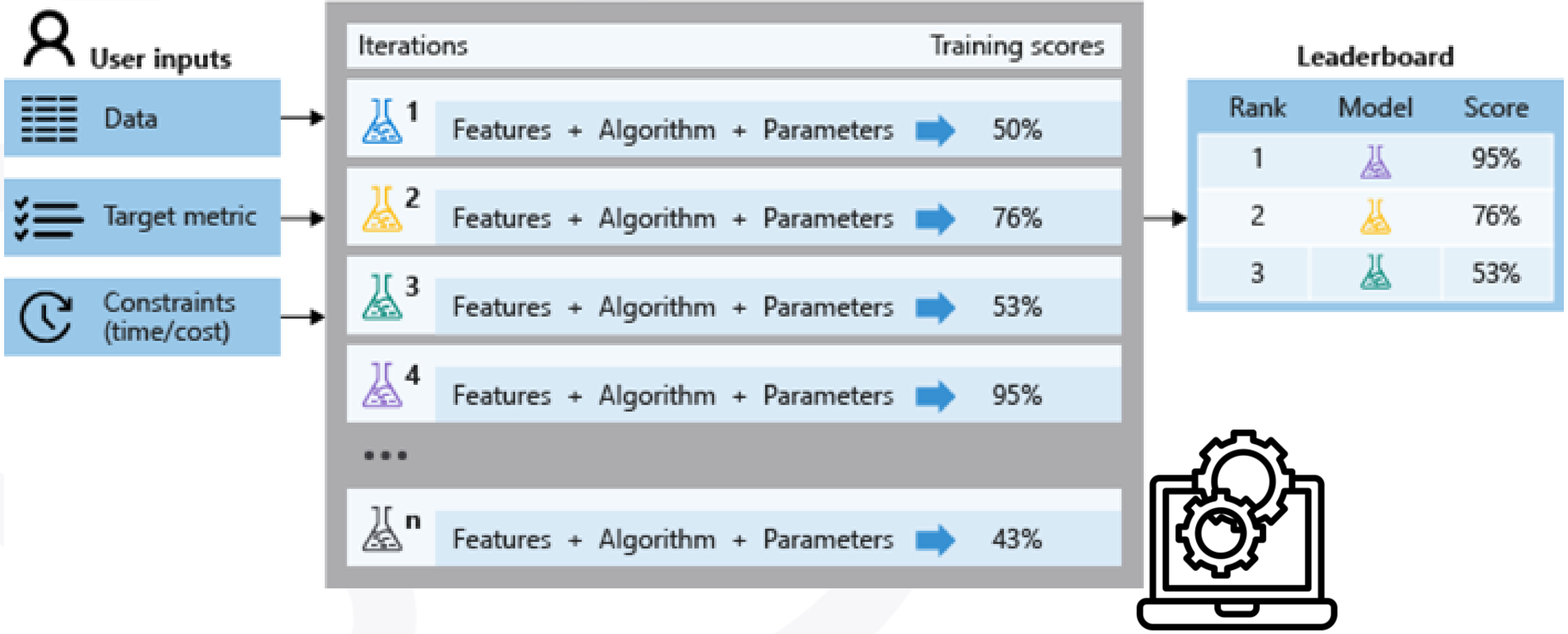


AutoML workflow



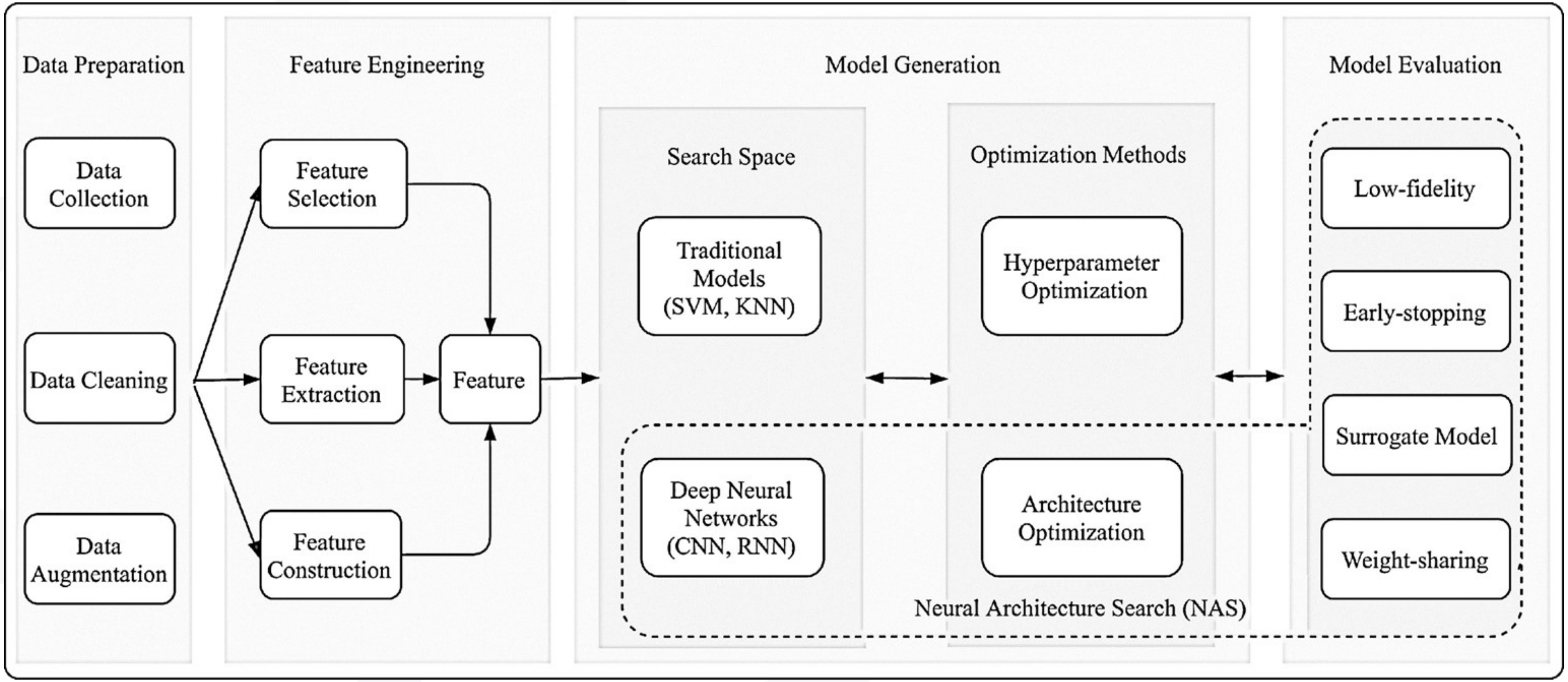
What is AutoML?

The goal of AutoML is to **simplify** and **accelerate** the process of developing machine learning models by reducing the manual effort from data scientists and AI/ML experts



What is AutoML?

The fundamental steps, especially for Computer Vision problems, is the design of the model. In Deep Learning this is called **Neural Architecture Search**



AI SPRINT

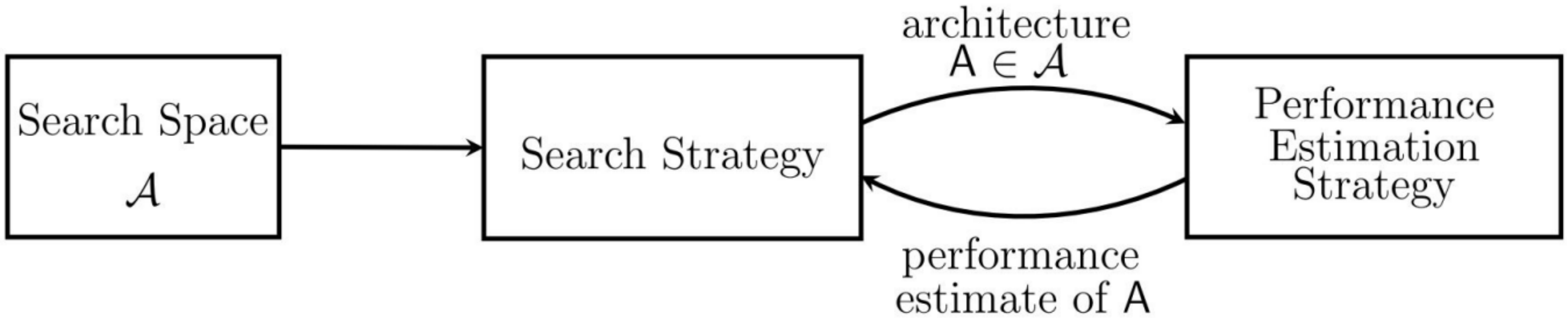
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

Neural Architecture Search (NAS)

Matteo Matteucci, Politecnico di Milano

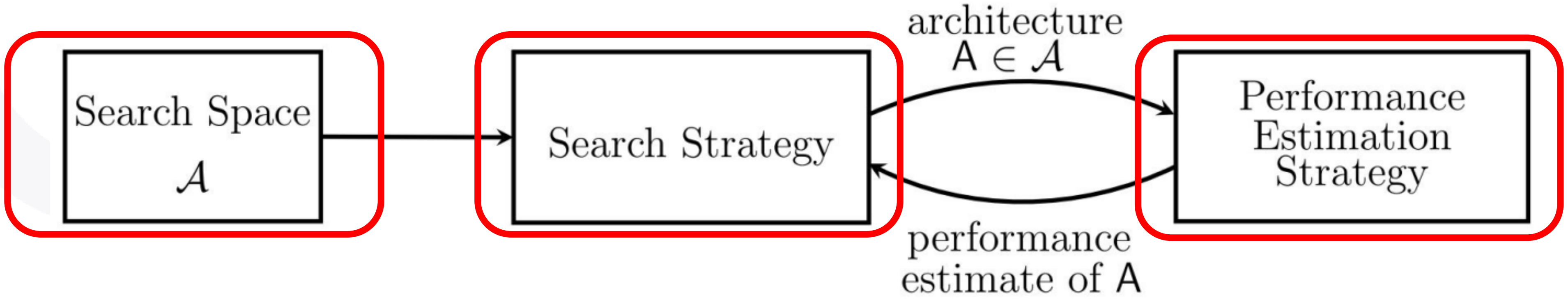
AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

Neural Architecture Search (**NAS**) is a technique within the field of machine learning that automates the process of **designing** and **discovering** optimal neural network architectures for a given task in the direction of **automatic model design.**



NAS works can be described as a composition of three ingredients:

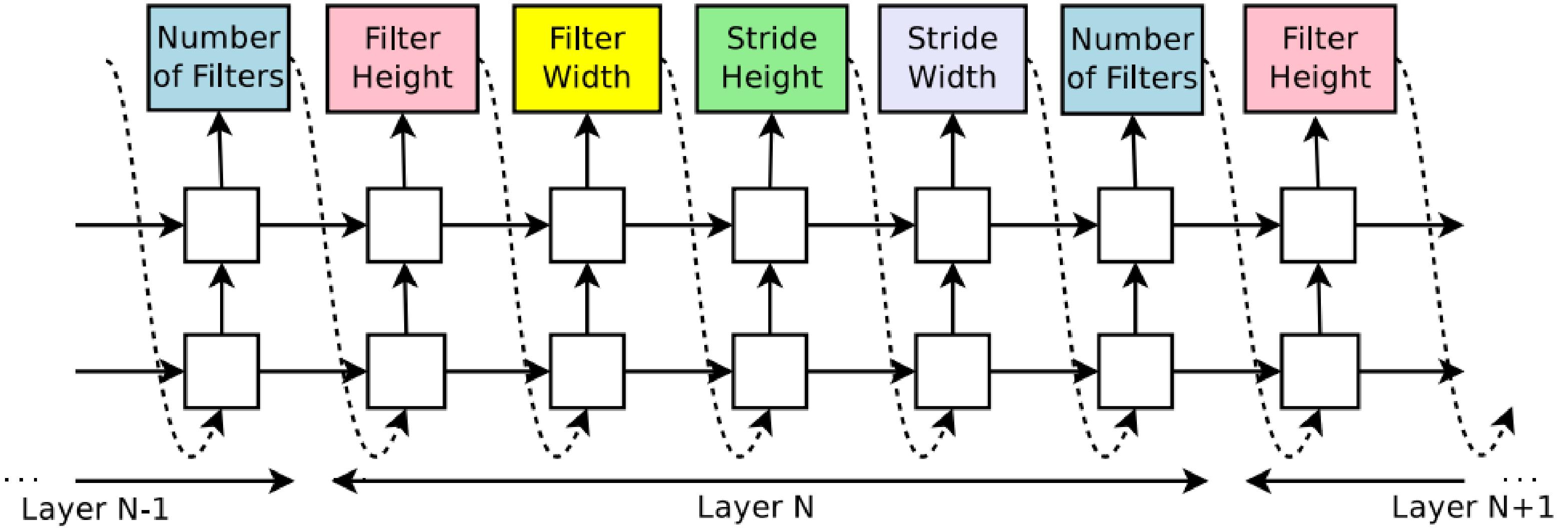
- The **search space** is the set of all possible architectures that can be found during the search process
- The **search strategy** defines how the algorithm explores the search space to find optimal architectures for the given task
- The **performance evaluation strategy** determines how to efficiently evaluate the quality of the architectures during the search process



Elsken et al. (2019)

“*Neural Architecture Search with Reinforcement Learning*” represents the first milestone in automating neural networks design.

The goal was to search for the whole neural network architecture for a given task



NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

Barret Zoph,* **Quoc V. Le**
Google Brain
{barretzoph,qvl}@google.com

ABSTRACT

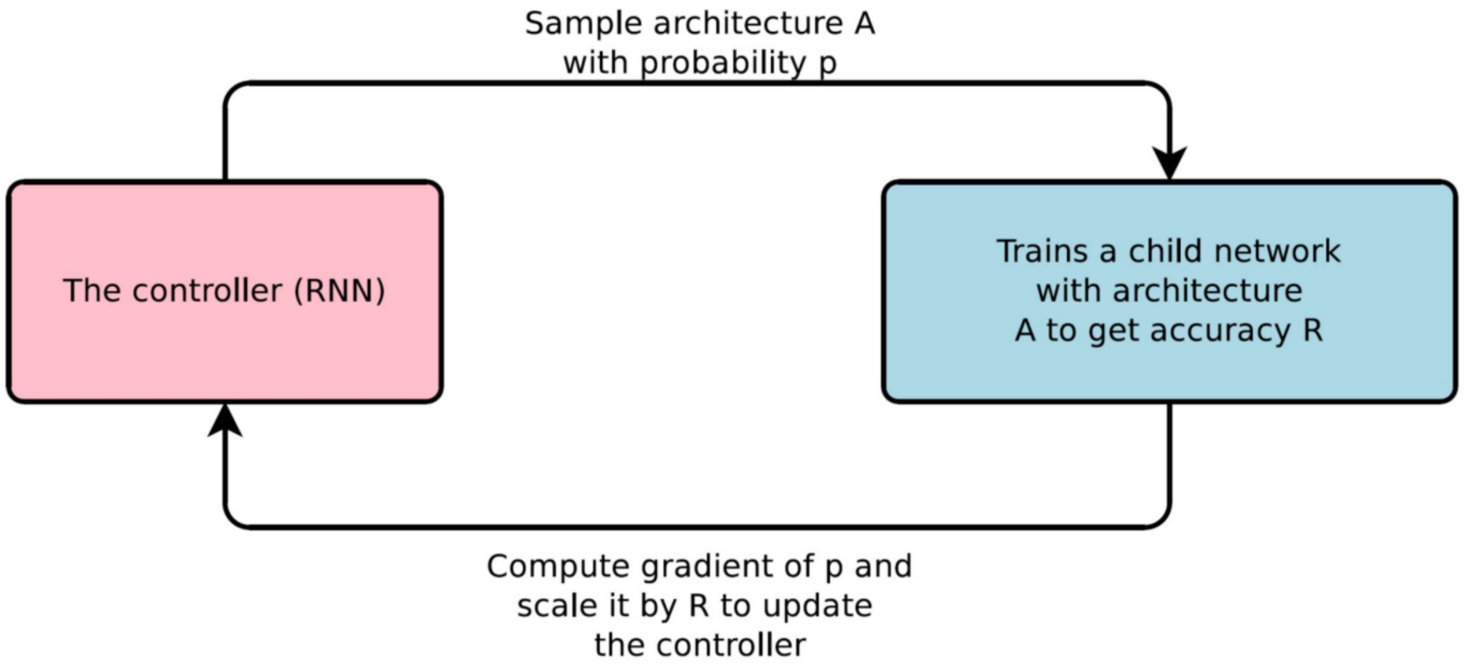
Neural networks are powerful and flexible models that work well for many difficult learning tasks in image, speech and natural language understanding. Despite their success, neural networks are still hard to design. In this paper, we use a recurrent network to generate the model descriptions of neural networks and train this RNN with reinforcement learning to maximize the expected accuracy of the generated architectures on a validation set. On the CIFAR-10 dataset, our method, starting from scratch, can design a novel network architecture that rivals the best human-invented architecture in terms of test set accuracy. Our CIFAR-10 model achieves a test error rate of 3.65, which is 0.09 percent better and 1.05x faster than the previous state-of-the-art model that used a similar architectural scheme. On the Penn Treebank dataset, our model can compose a novel recurrent cell that outperforms the widely-used LSTM cell, and other state-of-the-art baselines. Our cell achieves a test set perplexity of 62.4 on the Penn Treebank, which is 3.6 perplexity better than the previous state-of-the-art model. The cell can also be transferred to the character language modeling task on PTB and achieves a state-of-the-art perplexity of 1.214.

Zoph and Le (2017)

"Vanilla" Neural Architecture Search

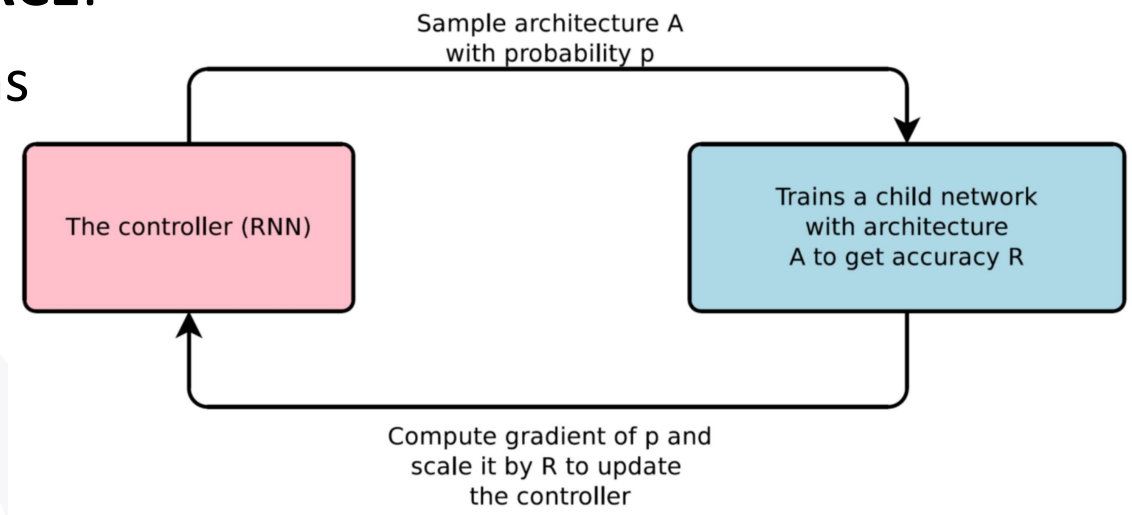
An RL-based controller proposing child model architectures for evaluation is included in the initial design of the NAS

The controller is implemented as an RNN that outputs a sequence of tokens of variable length, which are used for the configuration of a network architecture



The **controller** is trained as an RL task via **REINFORCE**:

- **Action Space**: The action space is a list of tokens for the definition of a child network that is predicted by the controller. The controller outputs an action, $a_{1:T}$, where T is the total number of tokens in the action space.
- **Reward**: The reward for training the controller is the accuracy of a child network that can be achieved at convergence R.
- **Loss**: NAS optimises the controller parameters w via REINFORCE loss. The goal is the maximization of the expected reward (high accuracy).



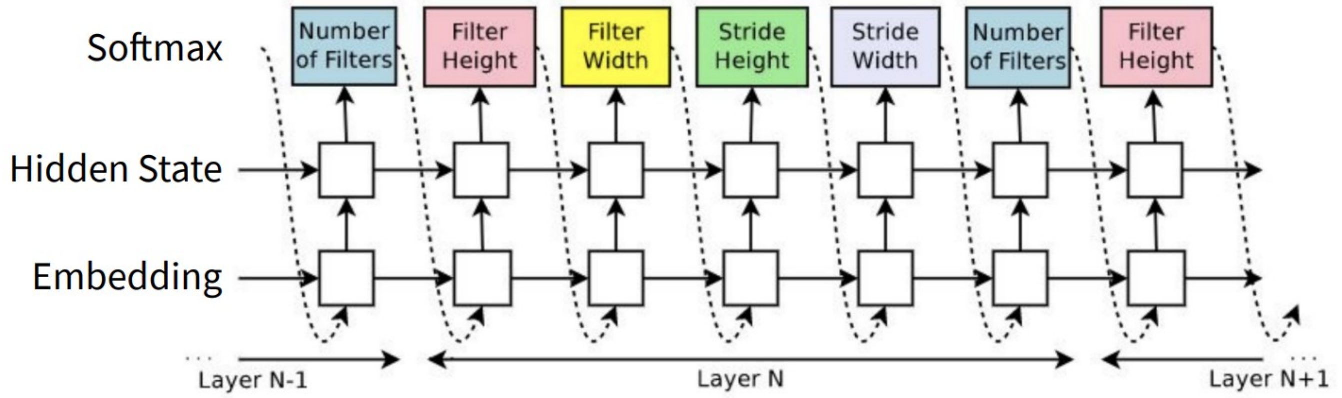
function REINFORCE
  Initialise θ arbitrarily
  for each episode $\{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**
    for $t = 1$ to $T - 1$ **do**
      $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$
    end for
  end for
  return θ
end function

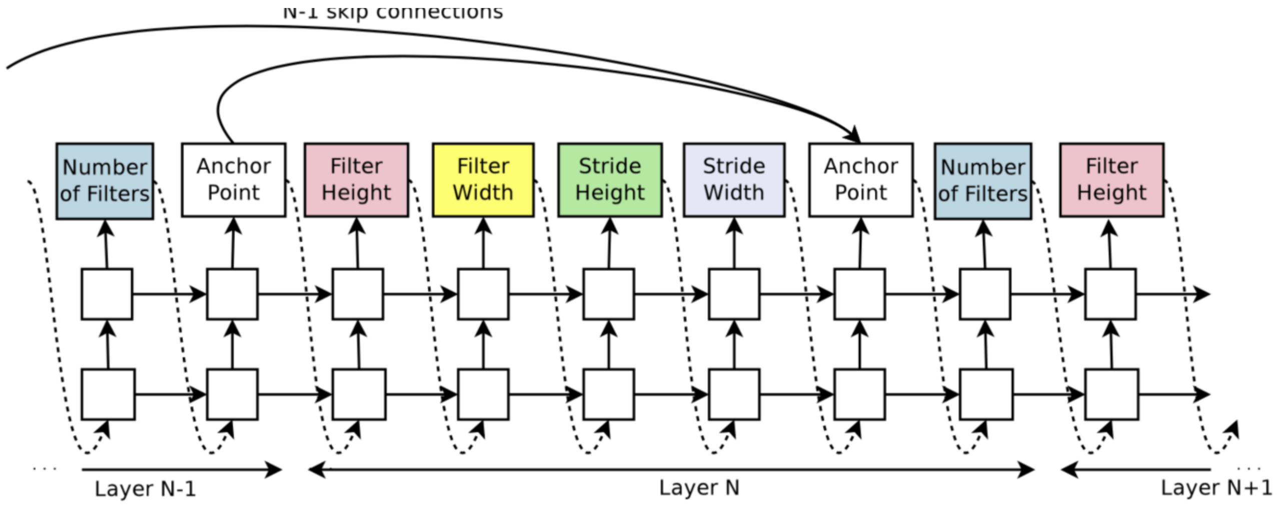
“Vanilla” Neural Architecture Search

The controller **samples** convolutional networks. It **predicts** filter height, width, stride height, stride width, and number of filters per layer.

Each prediction is made by a **softmax classifier**. Its score is then used as input for the next time step.

Skip connections added by means of anchor points too



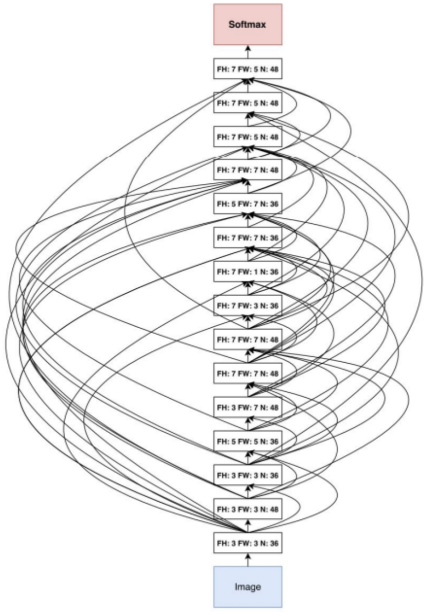


Zoph and Le (2017)

“Vanilla” Neural Architecture Search

The achieved results were impressive, with performance able to compete with the best human being designed architectures (the leaderboard is on CIFAR10 dataset)

Model	Depth	Parameters	Error rate (%)
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65



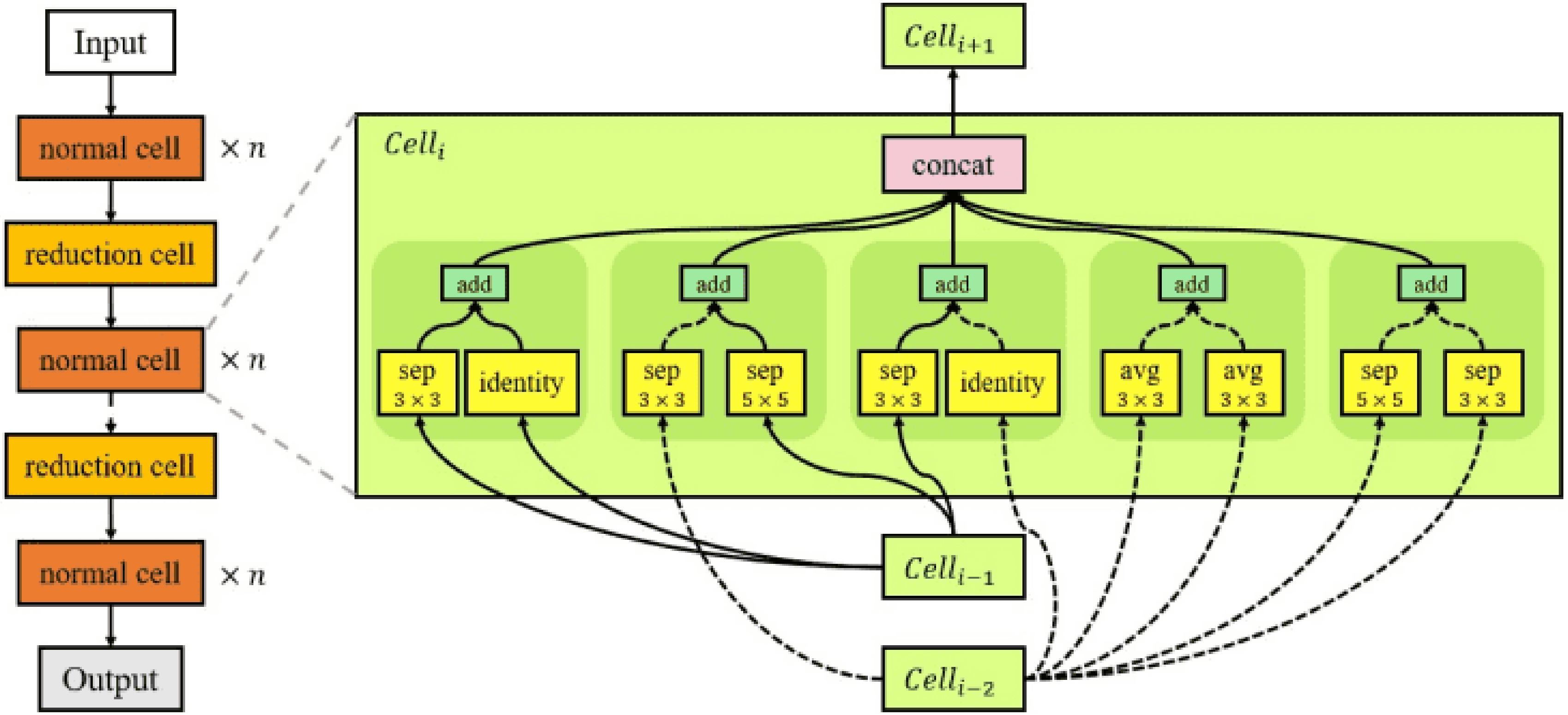
Best NAS Architecture on CIFAR-10

The cost involved the training of 12800 architectures from scratch until convergence, using 22400 GPU-days, and thus making the process ***not practical nor scalable*** ☹

Zoph and Le (2017)

NASNet

Inspired by the use of **repeating modules** in successful architectures (e.g. Inception, ResNet), the NASNet search space defines a convnet architecture by repeating several times the same **cell** containing **multiple operations** predicted by the NAS algorithm



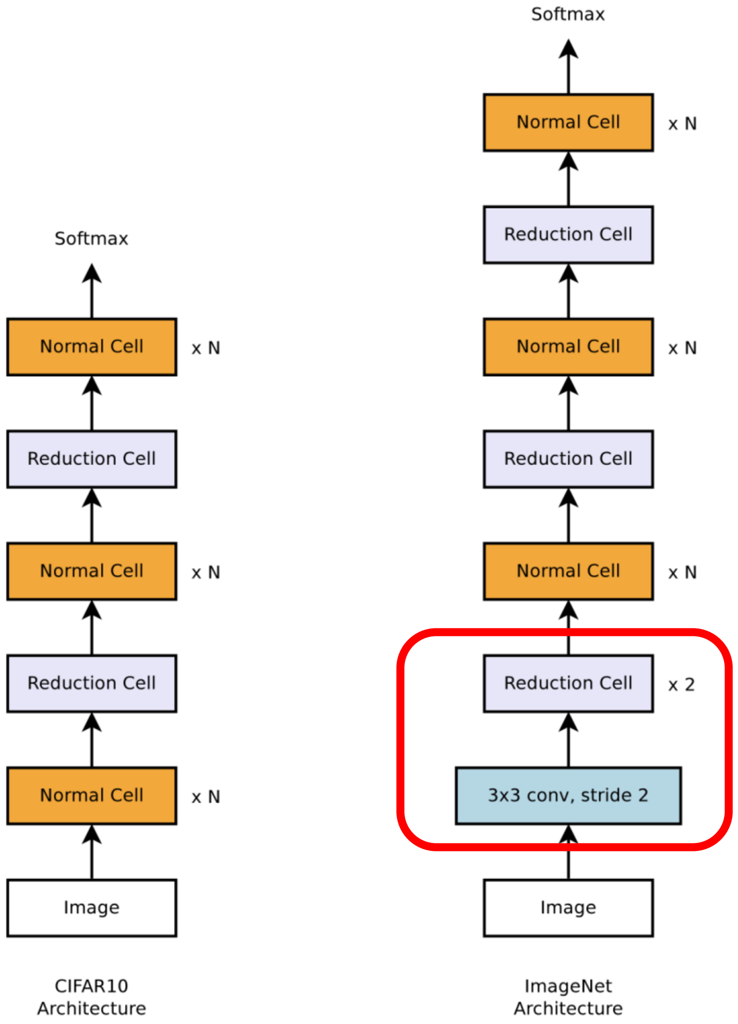
Zoph et al. (2018)

NASNet

The NASNet search space learns two types of cells for network construction:

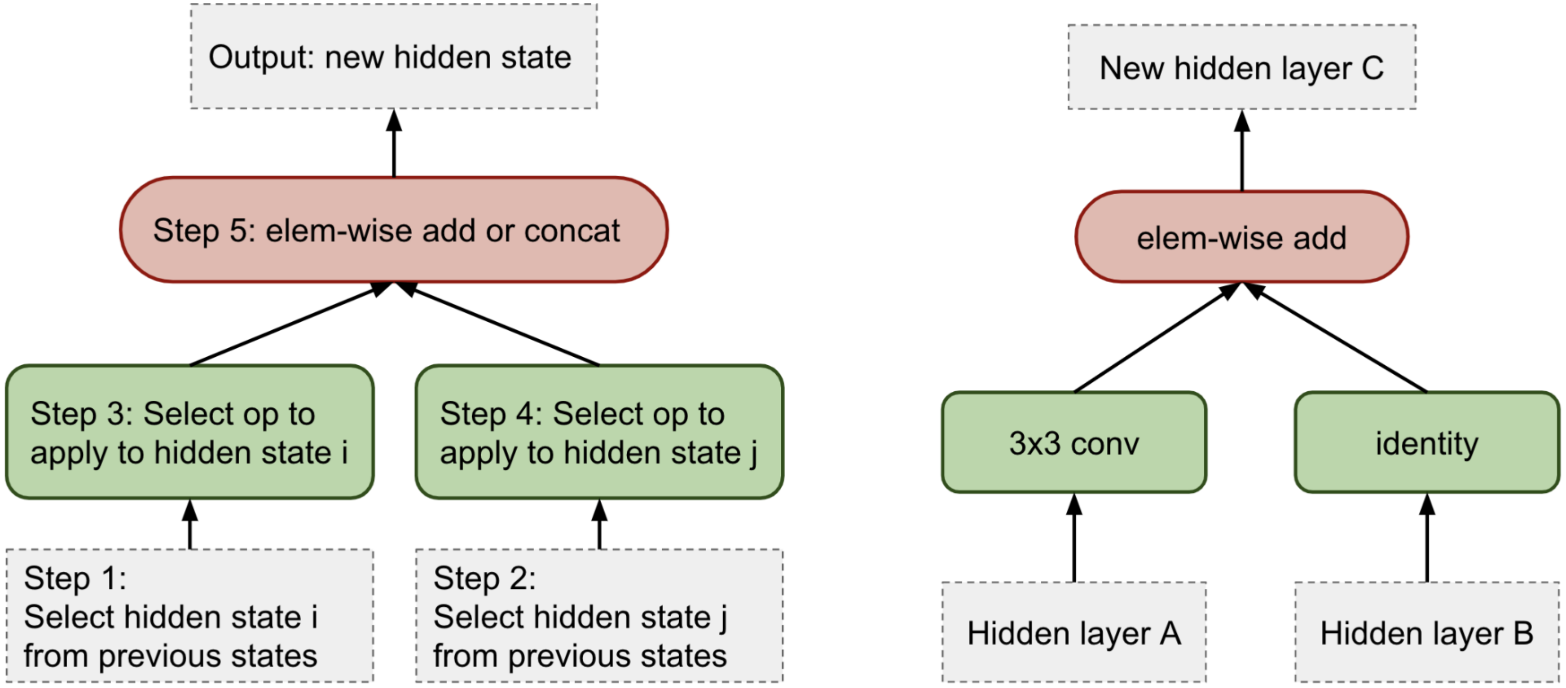
- **Normal Cell:** The input and output feature maps have the same dimension (like with convolutional layers).
- **Reduction Cell:** The output feature map has its width and height reduced by half (like with pooling layers).

Well-designed cell modules provide portability across datasets and easy scalability of model size by adjusting the number of cell repeats.



Zoph et al. (2018)

Predictors for each cell are made by ***B blocks*** (B = 5 in NASNet paper), where each block contains five predictive steps made by five different softmax classifiers corresponding to discrete selections of elements of a block.



(a) 5 discrete choices in each block

(b) A concrete example

Zoph et al. (2018)

NASNet

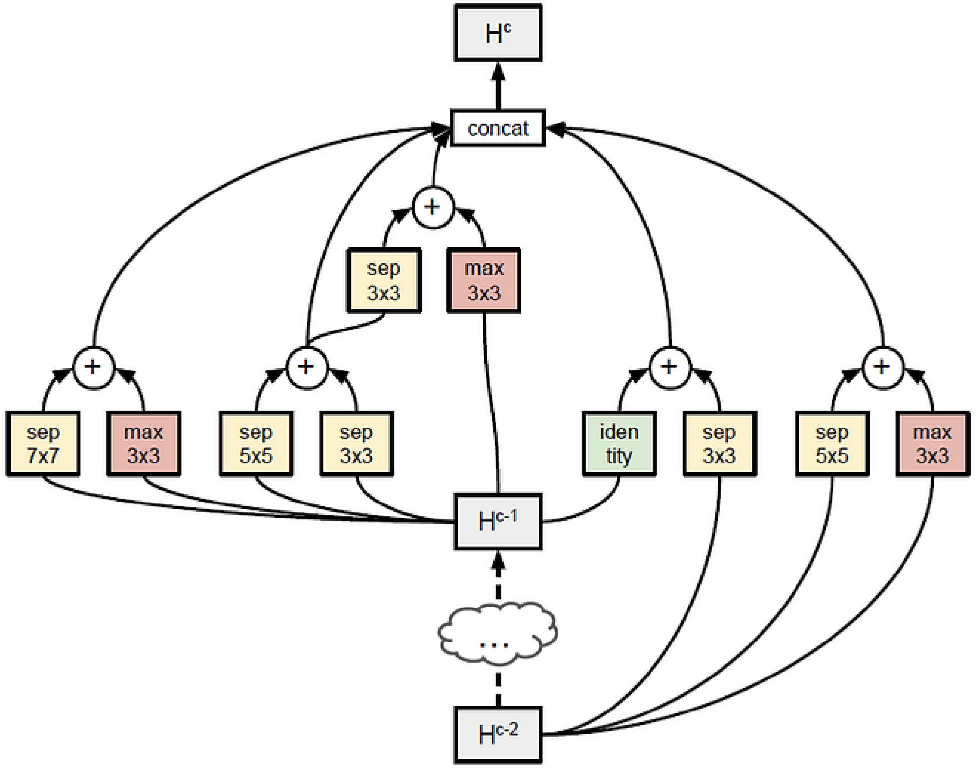
The major advantages of the NASNet search space can be summarized as:

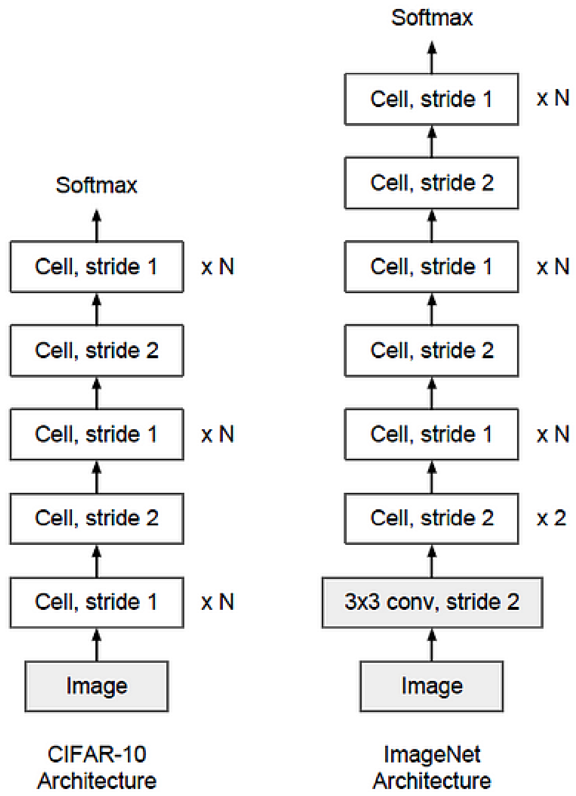
- The size of the search space is drastically reduced
- The cell-based architecture can be more easily applied to different datasets
- It provides strong evidence for a useful design pattern of repeated stacking of modules in architecture engineering (e.g., residual blocks in CNNs, multi-headed attention blocks in Transformers, etc.)

Search Method	Search Space	Search Strategy	Search Cost (GPU-days)	CIFAR10 Error	ImageNet Error (mobile)
NAS Zoph and Le (2017)	Global	REINFORCE	22400	3.65	-
NASNet Zoph et al. (2018)	Cell-based	PPO	**2000**	**3.41**	**26.0**

Zoph et al. (2018)

Progressive NAS (PNAS) frames the NAS problem as a progressive search for increasingly complex models via Sequential Model-based Bayesian Optimization (SMBO) as its search strategy instead of RL



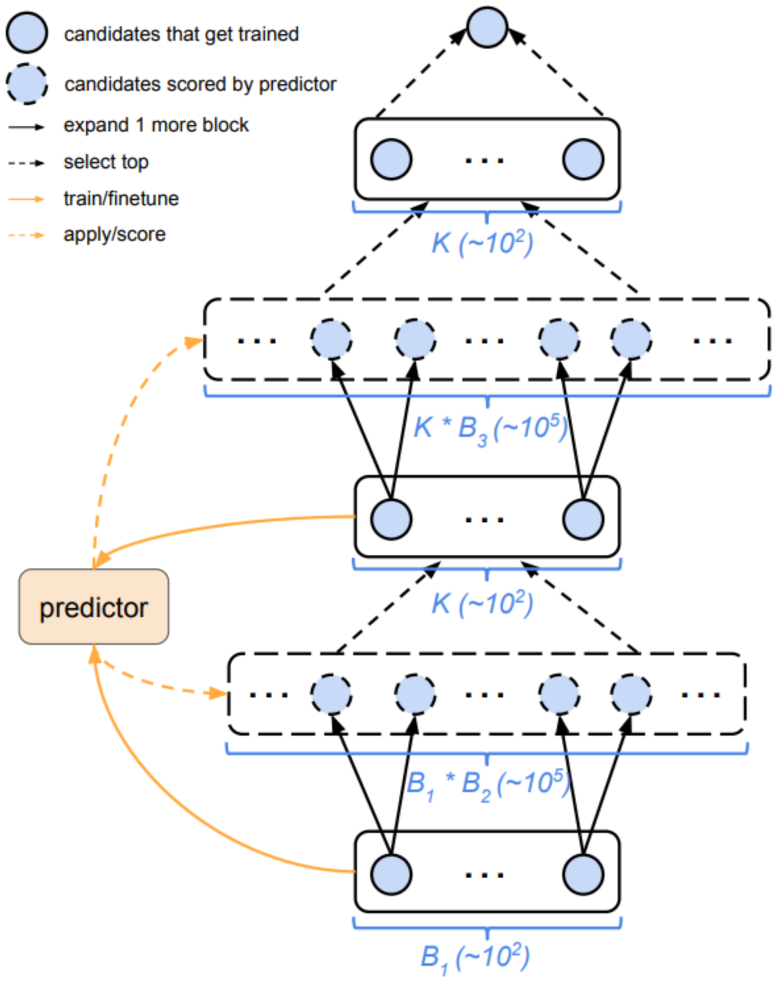


Liu et al. (2018)

Progressive NAS (PNAS)

PNAS makes use of the same search space of NASNet

- Each block is specified as a tuple of 5 elements, with PNAS considering only element-wise addition as the step 5
- Instead of setting the number of blocks B to a fixed number, **PNAS starts with B = 1**, a model with only one block in a cel **and gradually increases B**
- The performance on a validation set is used as feedback for the training of a **surrogate model** for the prediction of the performance of novel architectures. This **predictor** can then be used to prioritise which models to evaluate next
- The predictor is implemented with RNN model to handle different input sizes, accuracy, and sampling efficiency



Liu et al. (2018)

Neural Architecture Search Benchmark

Search Method	Search Space	Search Strategy	Search Cost (GPU-days)	CIFAR10 Error	ImageNet Error (mobile)
NAS Zoph and Le (2017)	Global	REINFORCE	22400	3.65	-
NASNet Zoph et al. (2018)	Cell-based	PPO	2000	**3.41**	26.0
PNAS Liu et al. (2018)	Cell-based	SMBO	**225**	**3.41**	**25.8**

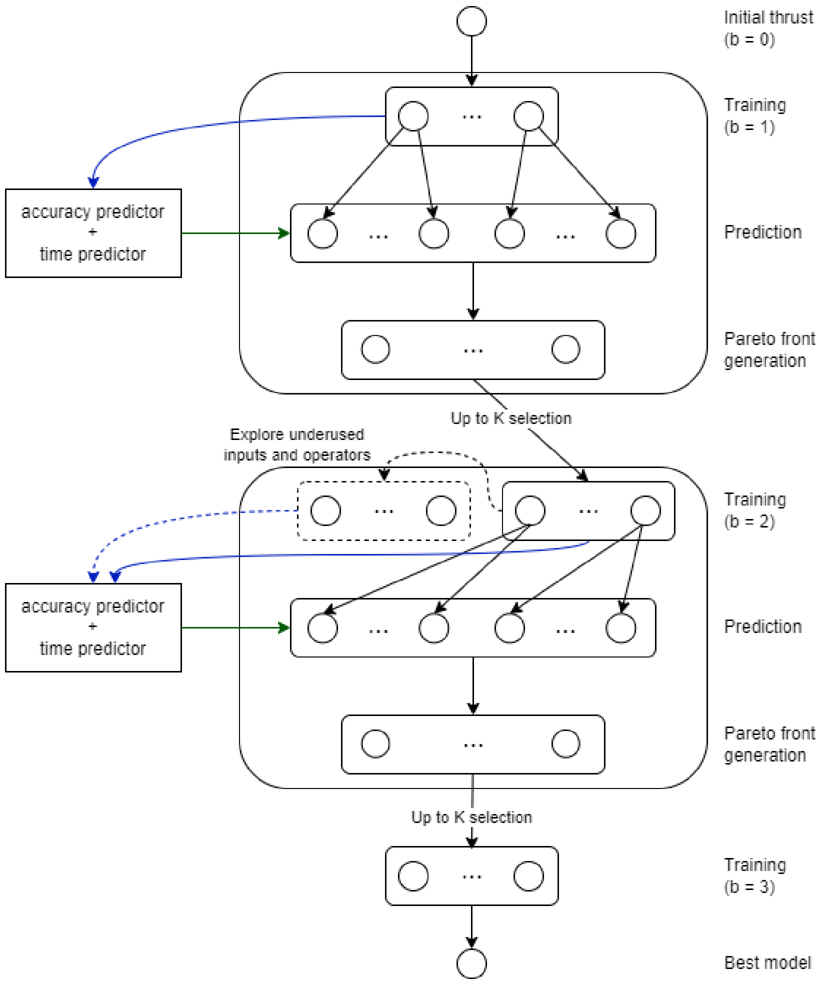
Pareto-Optimal Progressive NAS

Pareto-Optimal Progressive NAS (POPNAS) extends PNAS realising a **multi-objective** search between **accuracy** and **training time** of the researched architectures such that **Pareto optimality** is satisfied

POPNAS is based on two predictors:

- One predictor for accuracy (LSTM with Self-Attention)
- One predictor for training time (Catboost)

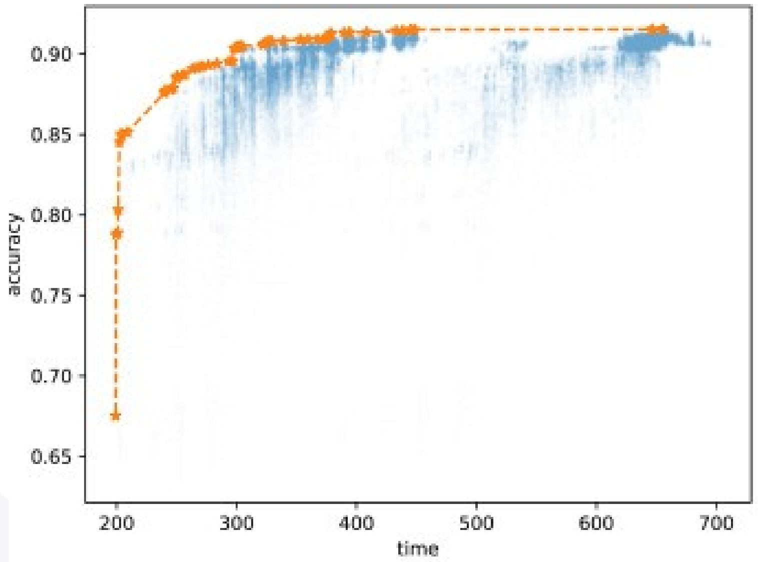
POPNAS can can adapt to both **image** and **time series classification** problems



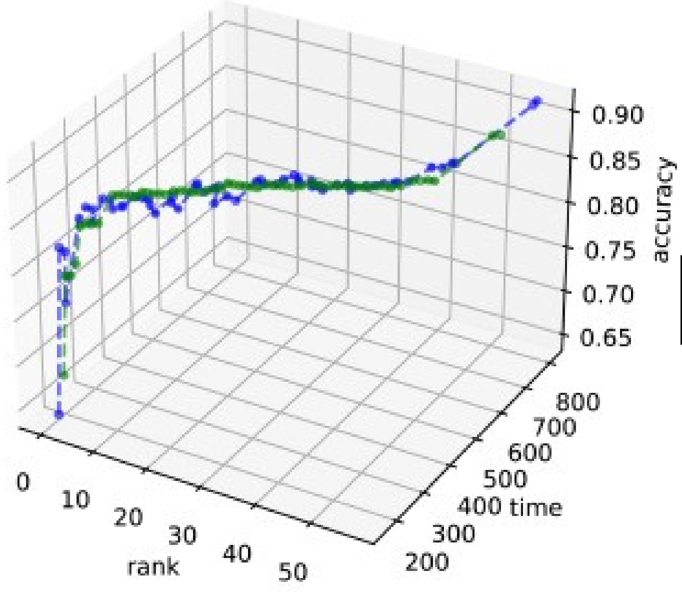
Falanti et al. (2023)

Pareto-Optimal Progressive NAS

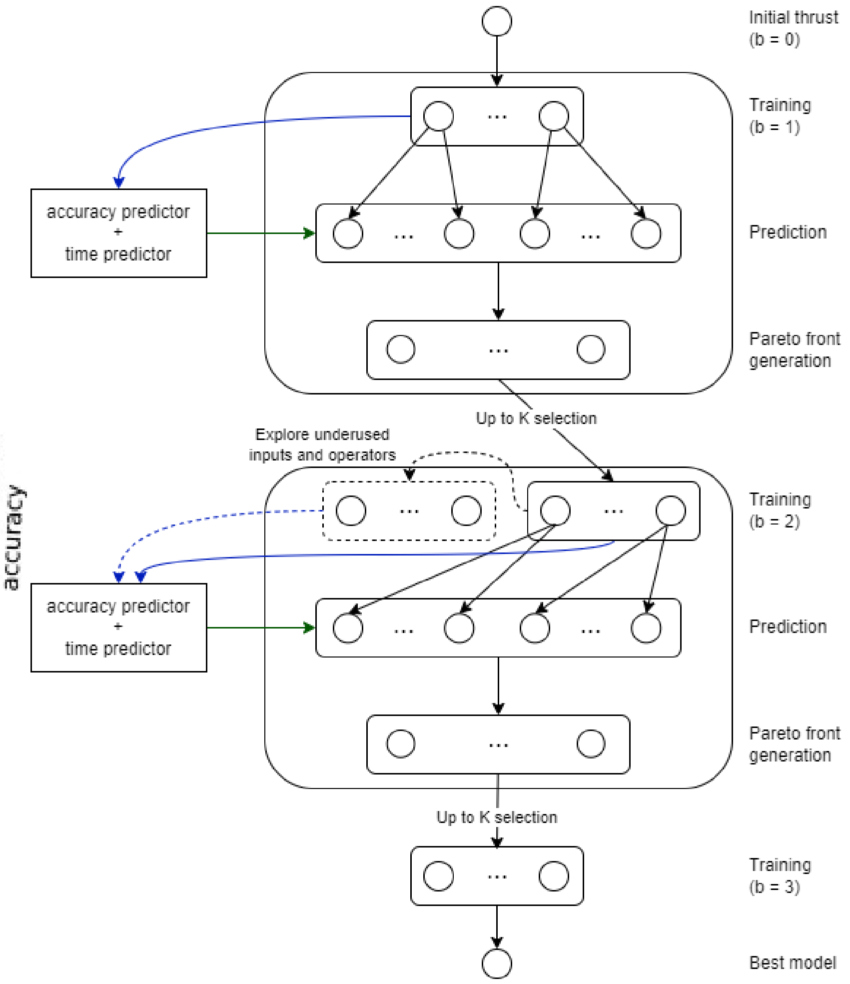
Pareto-Optimal Progressive NAS (POPNAS) extends PNAS realising a **multi-objective** search between **accuracy** and **training time** of the researched architectures such that **Pareto optimality** is satisfied



(a) Pareto front computed from predictions



(b) Predicted Pareto front vs real results



Falanti et al. (2023)

AI SPRINT

AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

Neural Architecture Search (NAS)

Matteo Matteucci, Politecnico di Milano

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

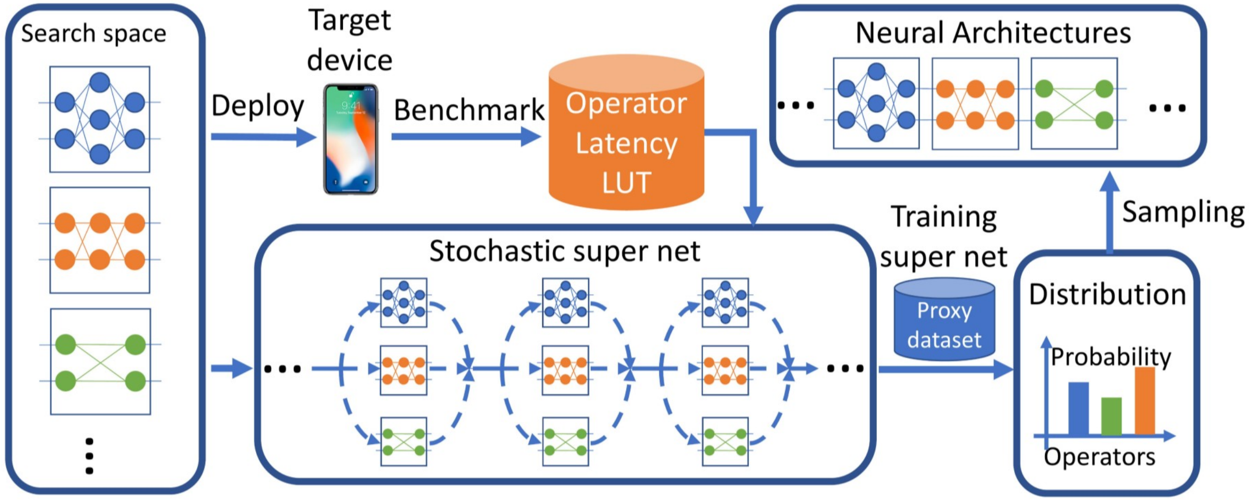
One-Shot NAS: Search + Evaluation

Since search and evaluation independently for a large population of child models is expensive, **one-shot architecture** search extends the idea of **weight sharing:**

- Combine learning of architecture generation with learning of weight parameters
- Treat child architectures as different **subgraphs of a supergraph** with shared weights between common edges in the **supergraph**

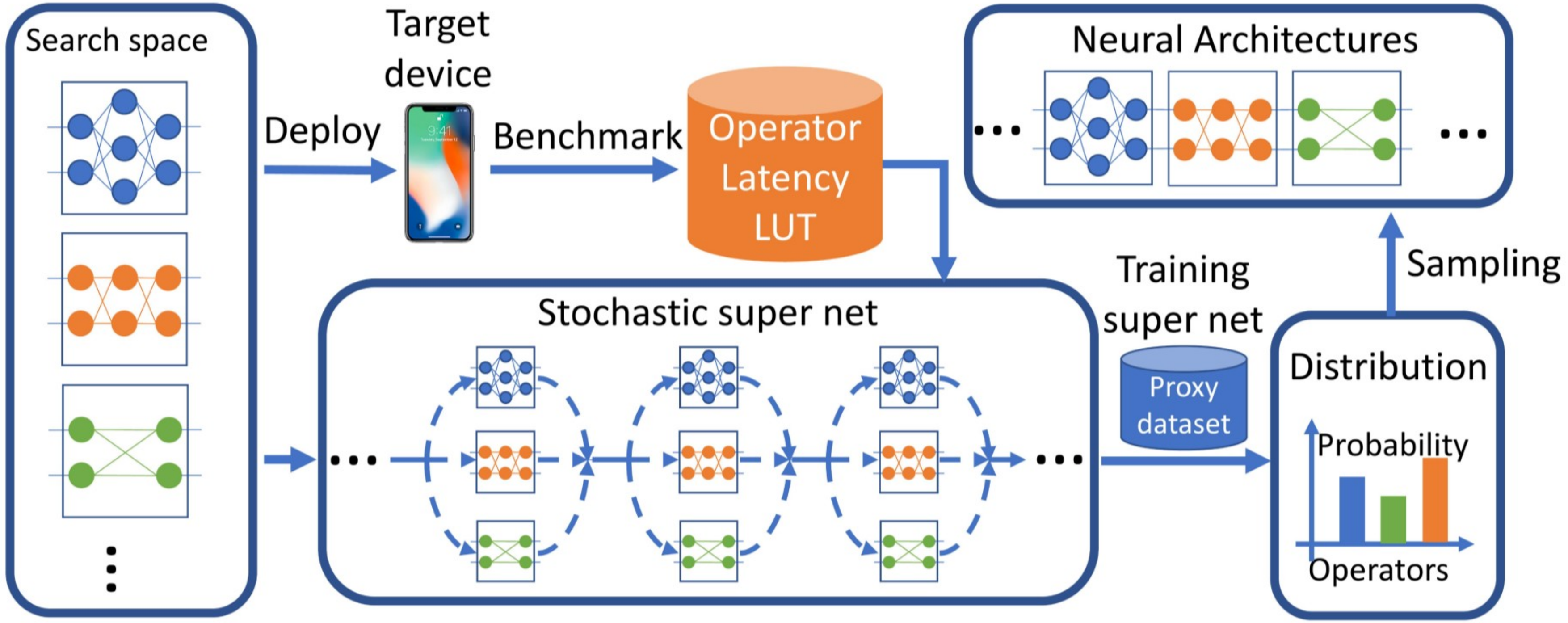
Once the one-shot model is trained, it is used to evaluate the performance of many different architectures that are randomly sampled by zeroing / removing operations.

This sampling process can be replaced by Reinforcement Learning or Evolutionary Algorithms



Liu et al. (2019)

Differentiable Architecture Search (**DARTS**) allows architecture parameters and weights to be jointly trained via gradient descent by introducing a continuous relaxation and softmax operators on each path in the search super-graph



Liu et al. (2019)

Differentiable Architecture Search

A cell is a directed acyclic graph (DAG) consisting of a topologically ordered sequence of N nodes. Each node has a latent representation $\boldsymbol{x_i}$ to be learned. Each edge $(\boldsymbol{i}, \boldsymbol{j})$ is associated with an operation $\boldsymbol{o}^{(i,j)} \in \mathbf{O}$ that transforms $\boldsymbol{x_j}$ to form $\boldsymbol{x_i}$:

$$\boldsymbol{x_i} = \sum_{j<i} \boldsymbol{o}^{(i,j)}(\boldsymbol{x_j})$$

DARTS relaxes the categorical choice of a particular operation as a softmax over all operations; architecture search is reduced to learning mixing probabilities $\alpha = \{\alpha^{(i,j)}\}$

$$\overline{\boldsymbol{o}}^{(i,j)}(\boldsymbol{x}) = \sum_{o \in \mathbf{O}} \frac{\exp\left(\boldsymbol{\alpha}_{ij}^{\boldsymbol{o}}\right)}{\sum_{o' \in \mathbf{O}} \exp(\boldsymbol{\alpha}_{ij}^{\boldsymbol{o}'})} \boldsymbol{o}(\boldsymbol{x})$$

with α_{ij} a $|\mathrm{O}|$ vector containing weights between nodes i and j over all operations

Liu et al. (2019)

A bilevel optimization arises because we want to optimize both the network weights w and the architecture representation α :

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\mathcal{L}}_{val}(\boldsymbol{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha})$$

$$s.t.\ \boldsymbol{w}^*(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{w}} \boldsymbol{\mathcal{L}}_{train}(\boldsymbol{w}, \boldsymbol{\alpha})$$

At step k, given the current architecture parameters $\boldsymbol{\alpha}_{k-1}$, we first optimise the weights $\boldsymbol{w}_k$ by moving $\boldsymbol{w}_{k-1}$ in the direction of minimizing, with a learning rate λ, training loss

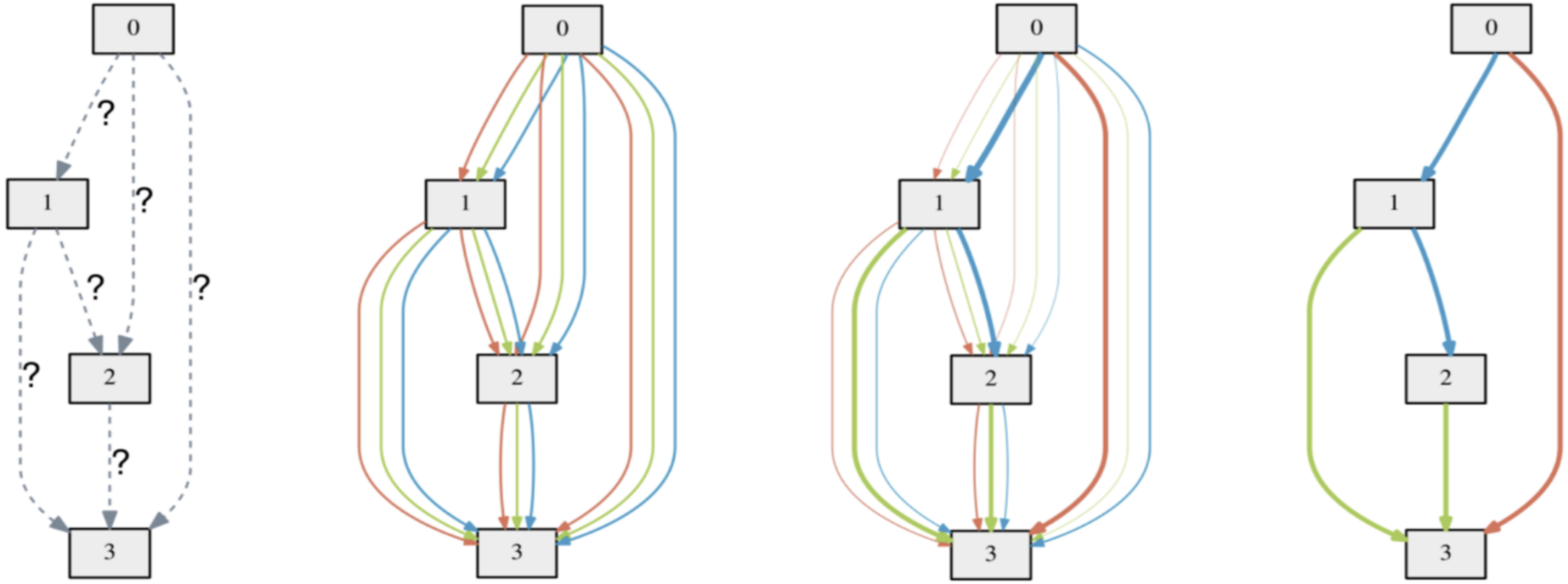
$$\min_{\boldsymbol{w}} \mathcal{L}_{train}(\boldsymbol{w}_{k-1}, \boldsymbol{\alpha}_{k-1})$$

Next, keeping the newly updated weights $\boldsymbol{w}_k$ we update mixing probabilities to minimize the validation loss after a single step of gradient descent with respect to the weights:

$$\boldsymbol{J}_\alpha = \boldsymbol{\mathcal{L}}_{val}(\boldsymbol{w}_k - \boldsymbol{\lambda}\nabla_w \boldsymbol{\mathcal{L}}_{train}(\boldsymbol{w}_k, \boldsymbol{\alpha}_{k-1}), \boldsymbol{\alpha}_k)$$

Liu et al. (2019)

DARTS finds an architecture with a low validation loss when its weights are optimized by gradient descent and one-step unrolled weights serve as a surrogate for $w^*(\alpha)$



(a) Initially unknown operations on the edges.

(b) Continuous relaxation by placing a mixture of operations on each edge.

(c) Bilevel optimization to jointly train mixing probabilities and weights.

(d) Finalized the model based on the learned mixing probabilities.

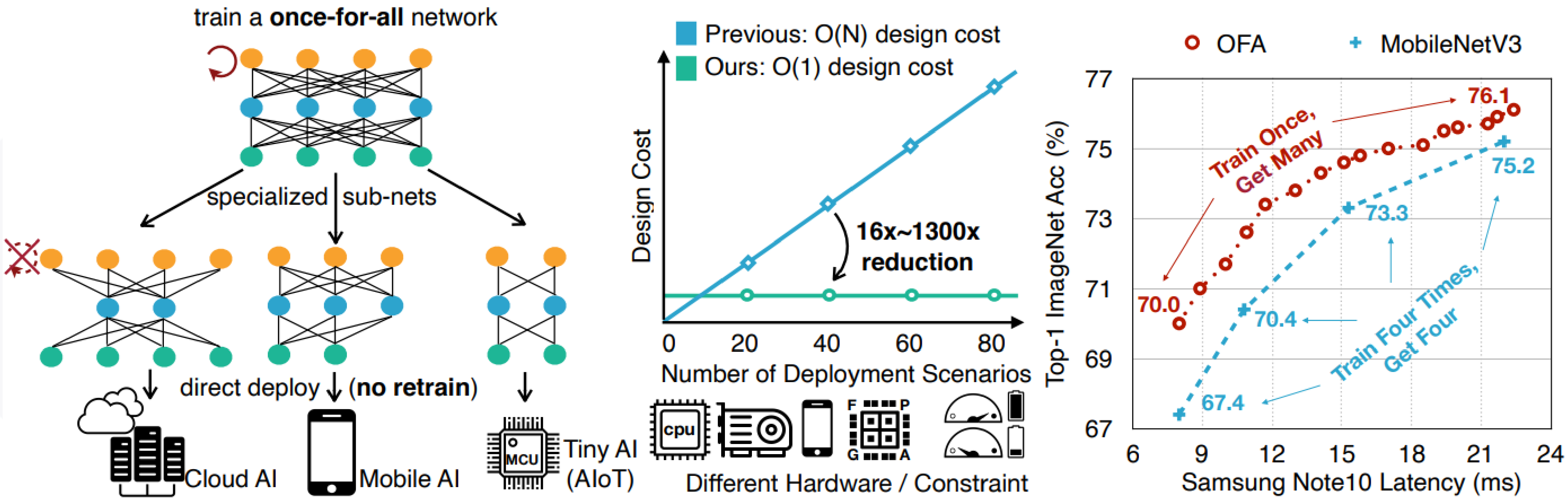
Liu et al. (2019)

Neural Architecture Search Benchmark

Search Method	Search Space	Search Strategy	Search Cost (GPU-days)	CIFAR10 Error	ImageNet Error (mobile)
NAS Zoph and Le (2017)	Global	REINFORCE	22400	3.65	-
NASNet Zoph et al. (2018)	Cell-based	PPO	2000	3.41	26.0
PNAS Liu et al. (2018)	Cell-based	SMBO	225	3.41	**25.8**
DARTS Liu et al. (2018)	Super-network	Weight Sharing + SGD	**4**	**3.00**	26.9

Once-for-All

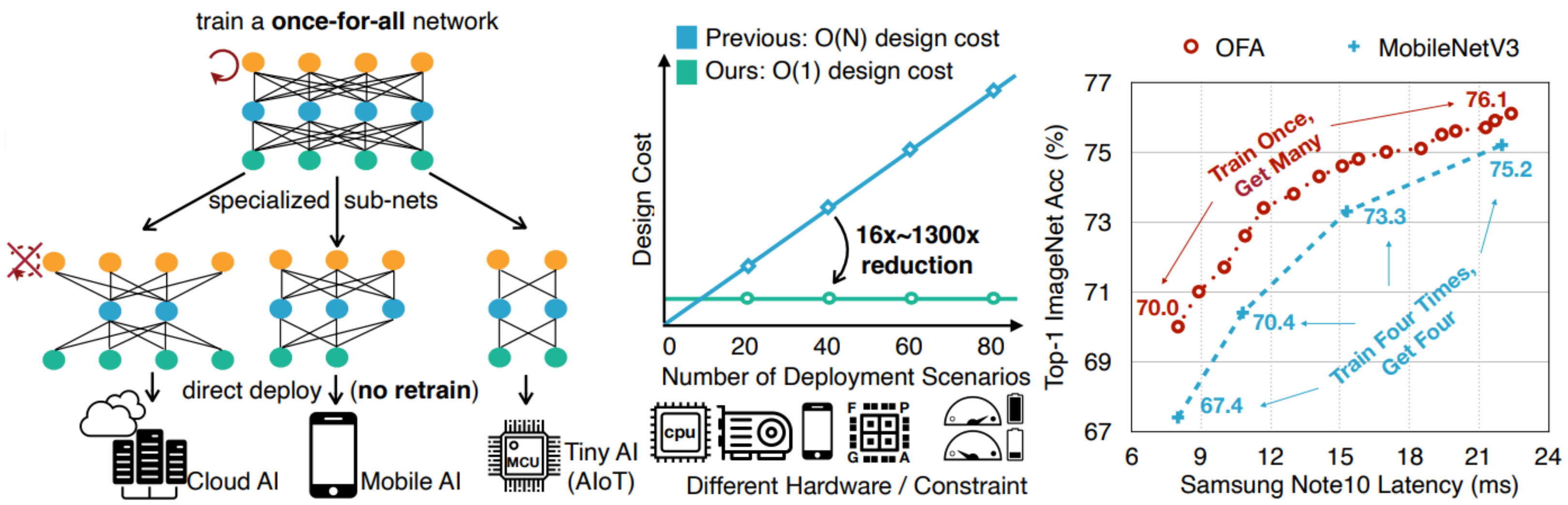
Once-for-All (OFA) builds on the ideas presented in DARTS, but aims to address the challenge of efficiently deploying neural network architectures on different hardware platforms with different computational requirements



Cai et al. (2020)

The key contribution of OFA is **architecture decoupling**; instead of optimizing both architecture & weights simultaneously, as in DARTS, OFA separates the process in two:

- OFA trains a single large neural network, referred to as the **super-network**, encompassing all possible sub-networks
- OFA derives specialized **sub-networks** from it by selecting appropriate paths based on specific hardware constraints or resource budgets.

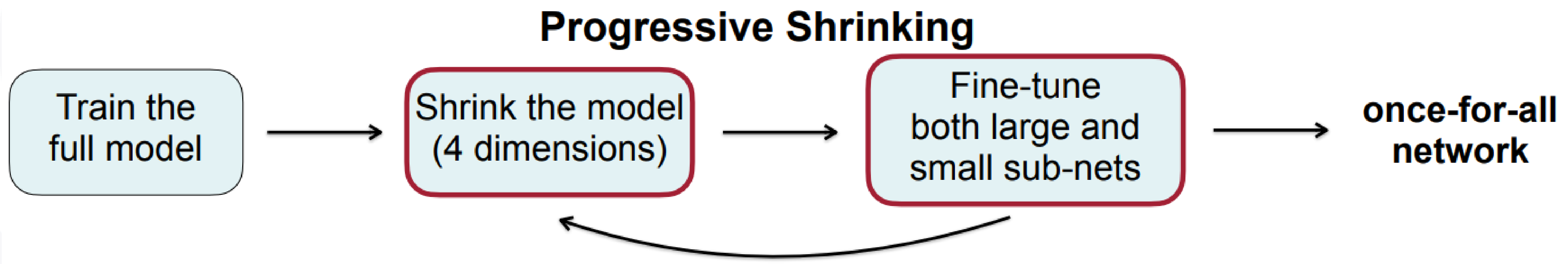


Cai et al. (2020)

OFA uses the **Progressive Shrinking (PS)** optimization strategy; this strategy not only enables the acquisition of excellent starting points for sub-network extraction but also concentrates the computational load into a single end-to-end training process

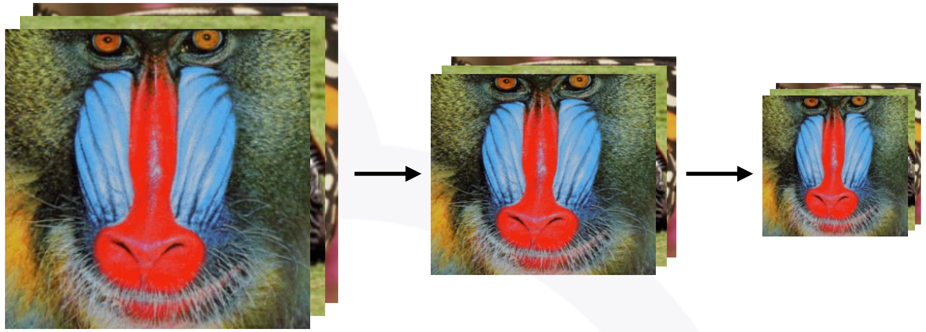
The algorithm begins by defining the maximal network, which includes all PS parameters set to their maximum values. Subsequently, the PS training steps and phases are executed sequentially

The PS algorithm is organized into **four elastic steps**, each comprising multiple phases.

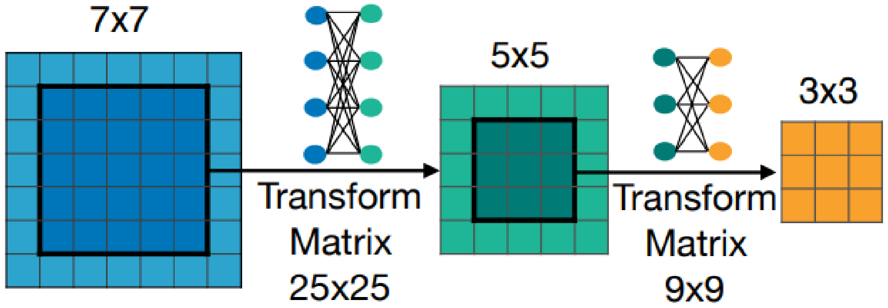


Cai et al. (2020)

The first step, **Elastic Resolution**, involves randomly varying the size of input images

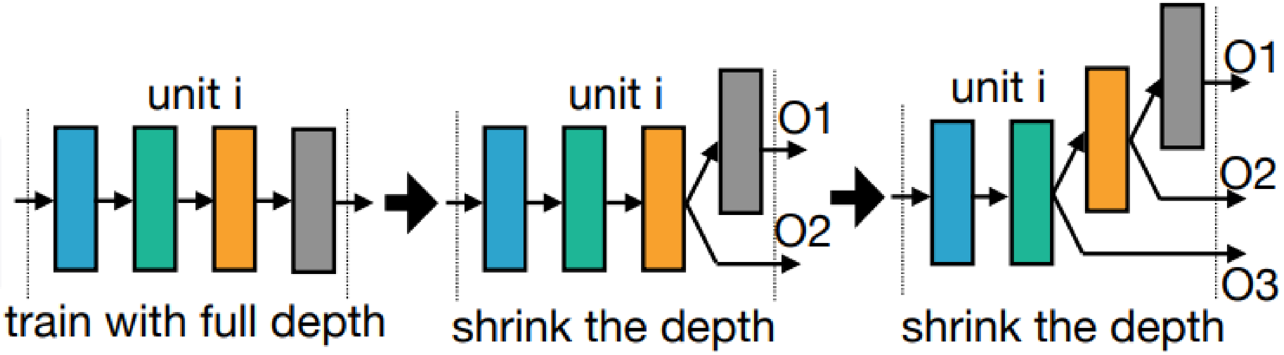


The second step, **Elastic Kernel Size**, gradually reduces the maximum kernel size for convolutional operators across the entire network

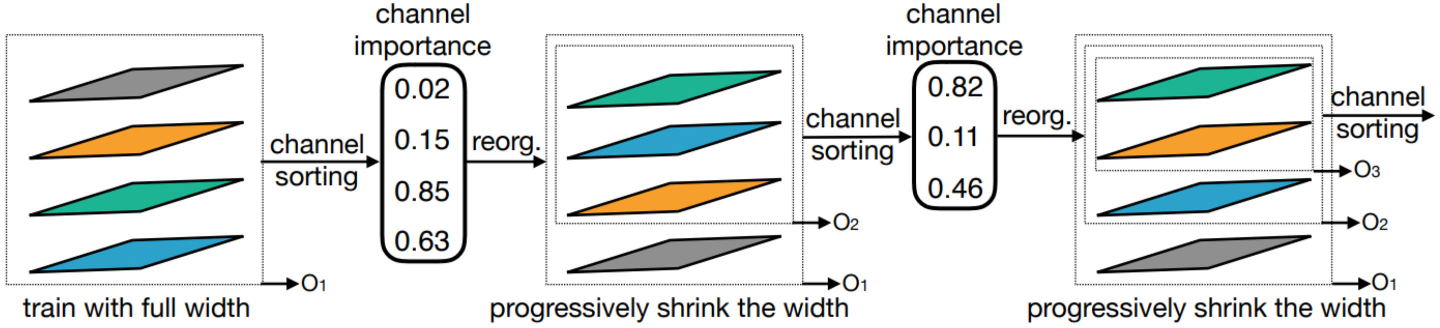


Cai et al. (2020)

The third step, **Elastic Depth**, progressively decreases the minimum depth achievable for sub-networks



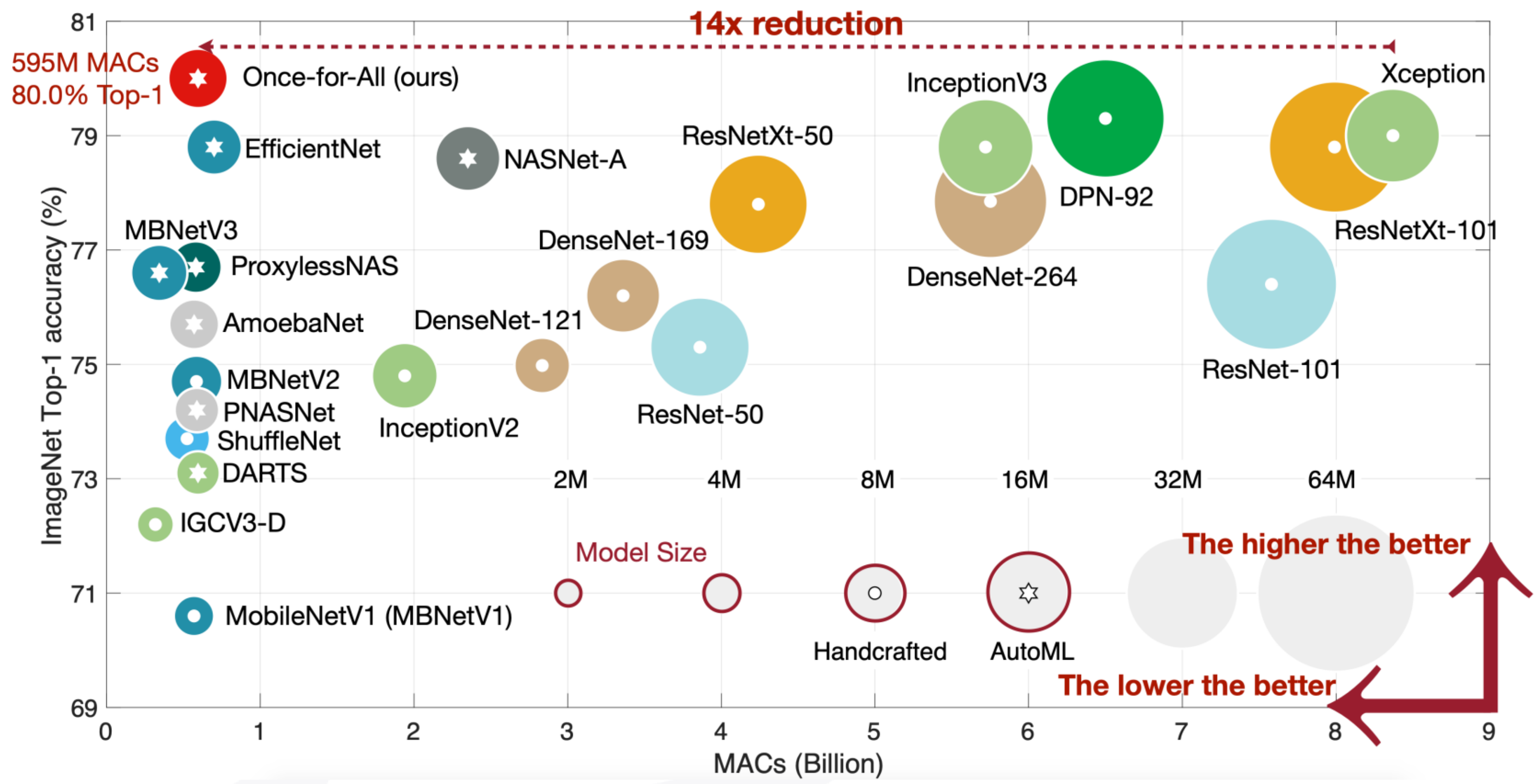
Finally, the fourth step, **Elastic Width**, aims to reduce the number of filters available for each convolutional layer



Cai et al. (2020)

Once-For-All

It is possible to sample **sub-networks** to extract their configuration encoding, and train surrogate models to enhance EA effectiveness in identifying the most suitable and performing sub-network according to different **hardware constraints**



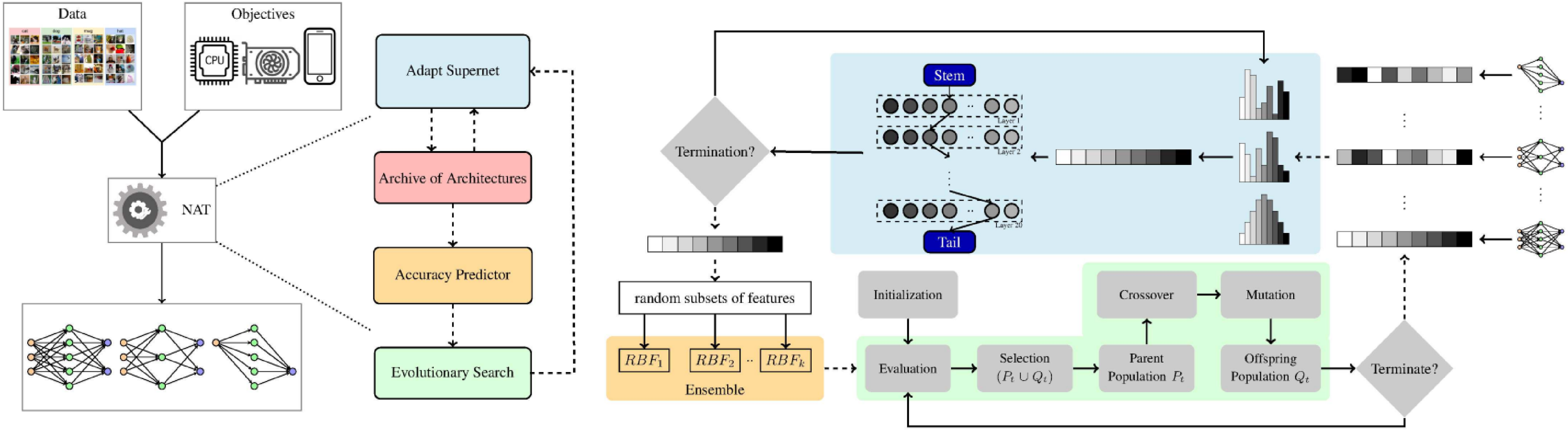
Cai et al. (2020)

Neural Architecture Search Benchmark

Search Method	Search Space	Search Strategy	Search Cost (GPU-days)	CIFAR10 Error	ImageNet Error (mobile)
NAS Zoph and Le (2017)	Global	REINFORCE	22400	3.65	-
NASNet Zoph et al. (2018)	Cell-based	PPO	2000	3.41	26.0
PNAS Liu et al. (2018)	Cell-based	SMBO	225	3.41	25.8
DARTS Liu et al. (2018)	Super-network	Weight Sharing + SGD	4	**3.00**	26.9
OFA Cai et al. (2020)	Super-network	Weight Sharing + EA	(50 +) **1.67**	-	**20.0**

Neural Architecture Transfer

Neural Architecture Transfer (NAT) builds on OFA framework as an adaptive post-processing replacement of the original sub-network extraction. NAT progressively transforms a pre-trained generic super-network into a task-specific super-network



Lu et al. (2021)

Neural Architecture Transfer

To speed-up the super-network adaptation process, NAT selectively fine-tunes only those parts of the super-network that correspond to sub-networks whose structures can be directly sampled from the current trade-off front distribution.

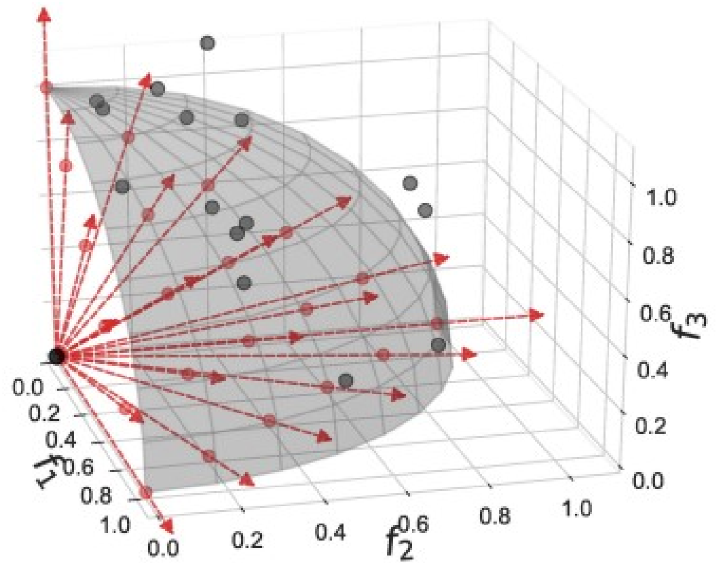
NAT's multi-objective evolutionary search is guided and accelerated by a performance prediction model updated online with only the best sub-network configurations

R	W	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	$\cdots$	L_{17}	L_{18}	L_{19}	L_{20}
		Stage 1				Stage 2					Stage 5			

R	192	$\cdots$	224	$\cdots$	256
E	0	$\cdots$	8	$\cdots$	16

W	1.0	1.2
E	0	1

L_i	Skip	$E=3, K=3$	$E=3, K=5$	$\cdots$	$E=6, K=7$
E	0	1	2	$\cdots$	9



Lu et al. (2021)

Neural Architecture Search Benchmark

Search Method	Search Space	Search Strategy	Search Cost (GPU-days)	CIFAR10 Error	ImageNet Error (mobile)
NAS Zoph and Le (2017)	Global	REINFORCE	22400	3.65	-
NASNet Zoph et al. (2018)	Cell-based	PPO	2000	3.41	26.0
PNAS Liu et al. (2018)	Cell-based	SMBO	225	3.41	25.8
DARTS Liu et al. (2018)	Super-network	Weight Sharing + SGD	4	3.00	26.9
OFA Cai et al. (2020)	Super-network	Weight Sharing + EA	(50 +) **1.67**	-	20.0
NAT Lu et al. (2021)	Super-network	Weight Sharing + EA	(50 +) 6.25	**1.60**	**19.5**

AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

Neural Architecture Search (NAS)

Matteo Matteucci, Politecnico di Milano

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**