Edge Intelligence and Scheduling on Heterogeneous Platforms

Weisong Shi and Yongtao Yao

Wayne State University

weisong@wayne.edu, yongtaoyao@wayne.edu
http://thecarlab.org

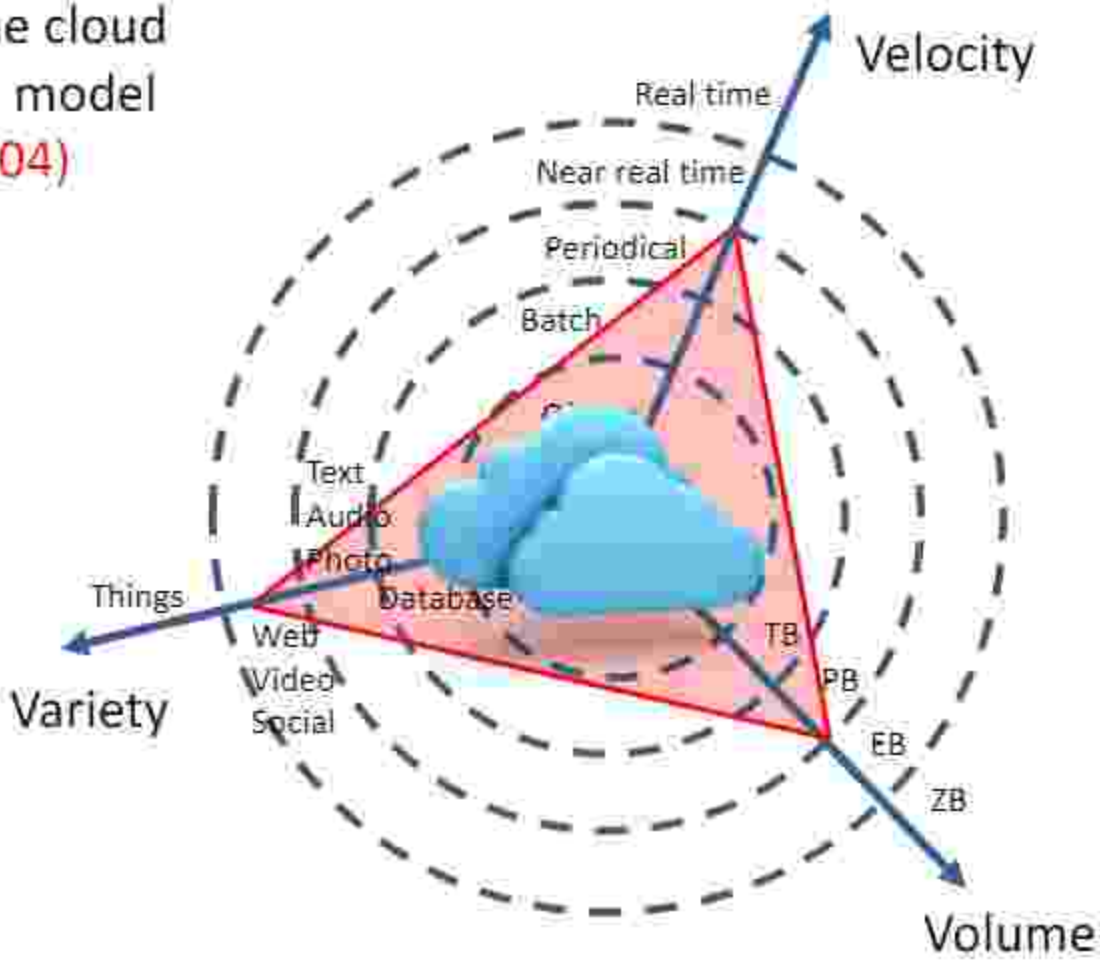
Roadmap

- ☞ Why Edge Computing?
- Edge Intelligence
- Edge Computing in CAVs

Digital Infrastructure 1.0 (05-15)

Key Features

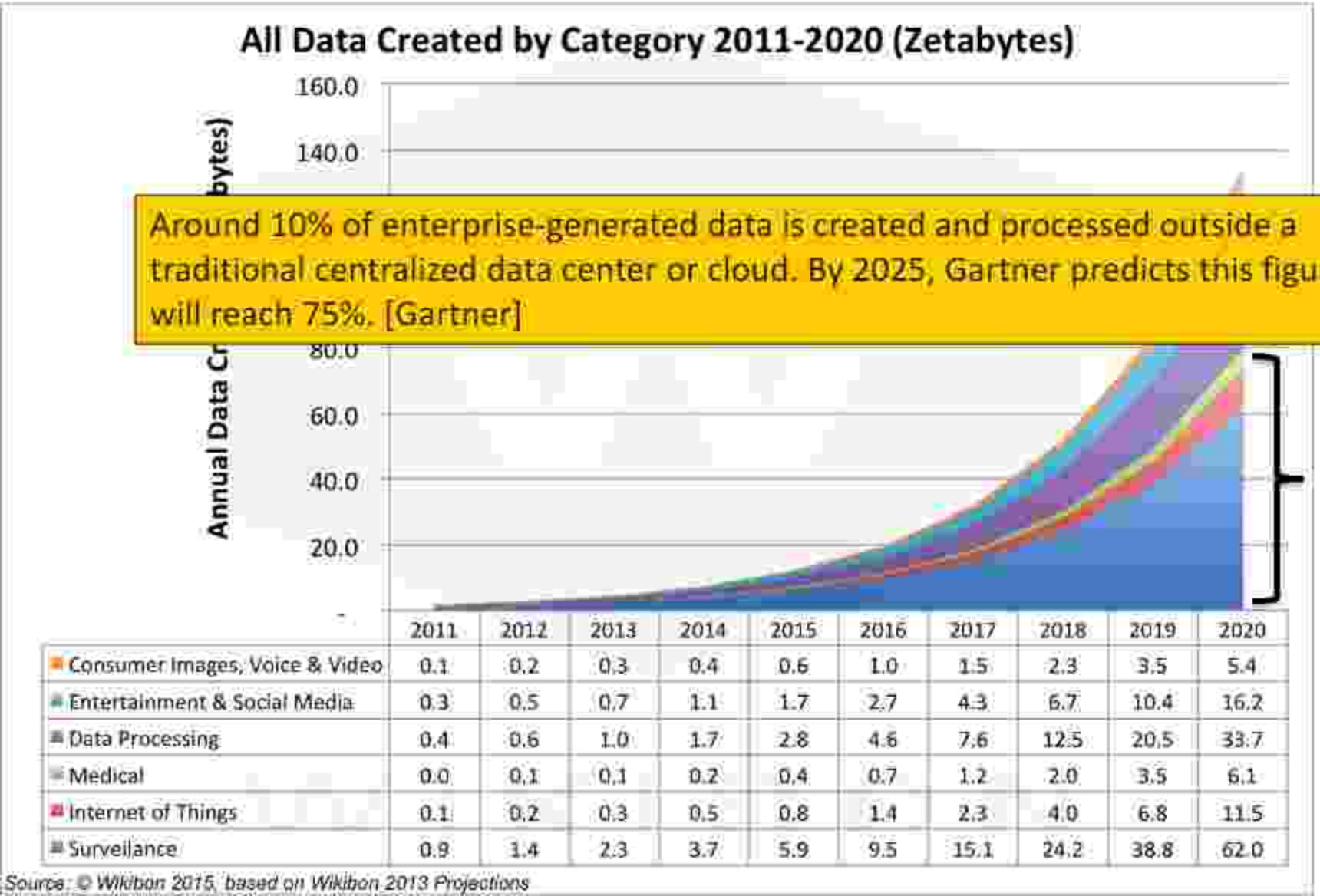
- Push the data to the cloud
- Popular processing model
  MapReduce (OSDI'04)



Edge Computing (2015 -)

- Not new, an early example is CDN
- Push from Cloud providers
  - Reduce latency, e.g., 30ms
  - Improve availability
  - Save bandwidth
- Pull from Internet of Things
  - Real time context computing
  - Resource constraints
  - Security/privacy requirements

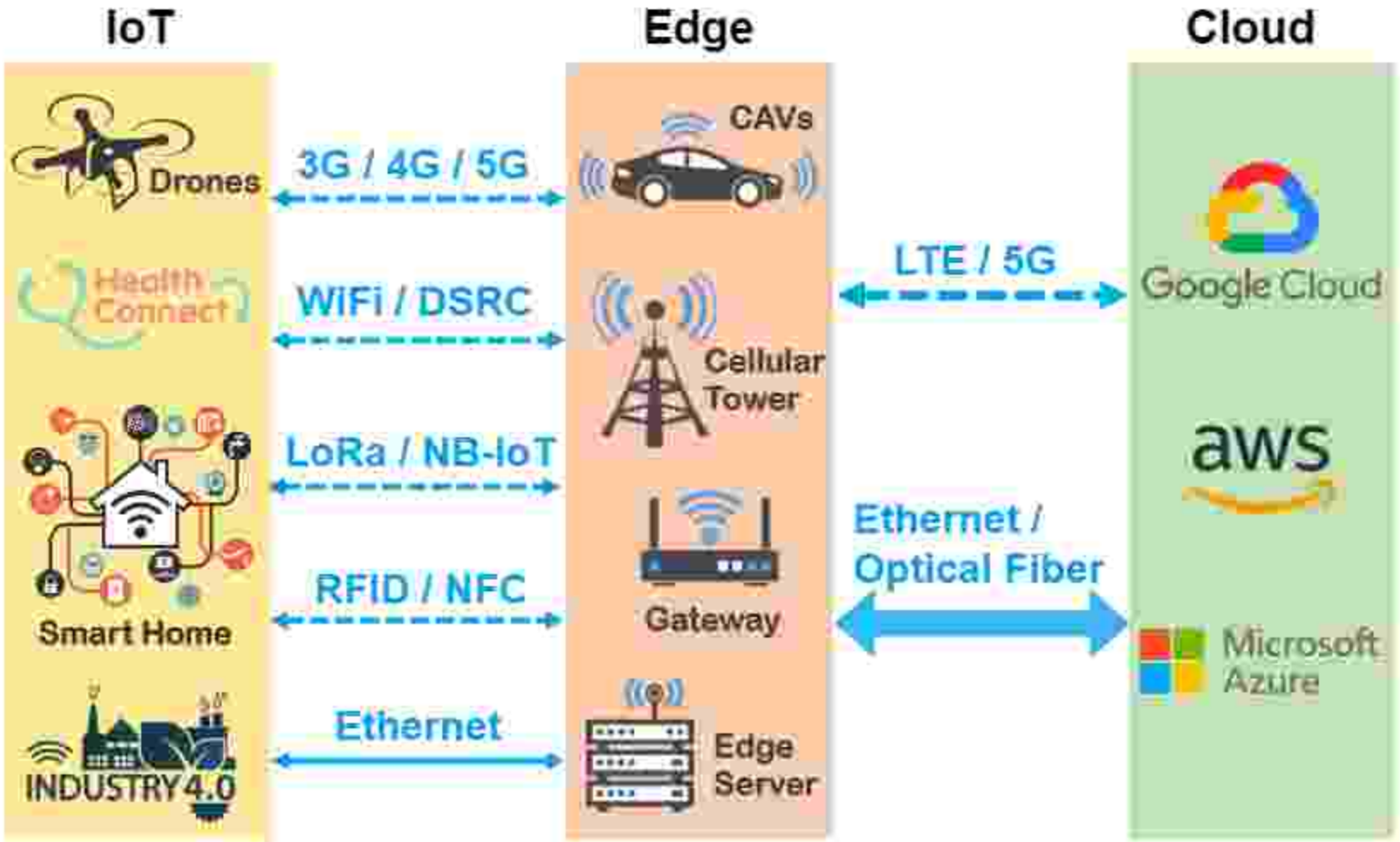
Data Trend: Edge



	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Consumer Images, Voice & Video	0.1	0.2	0.3	0.4	0.6	1.0	1.5	2.3	3.5	5.4
Entertainment & Social Media	0.3	0.5	0.7	1.1	1.7	2.7	4.3	6.7	10.4	16.2
Data Processing	0.4	0.6	1.0	1.7	2.8	4.6	7.6	12.5	20.5	33.7
Medical	0.0	0.1	0.1	0.2	0.4	0.7	1.2	2.0	3.5	6.1
Internet of Things	0.1	0.2	0.3	0.5	0.8	1.4	2.3	4.0	6.8	11.5
Surveillance	0.9	1.4	2.3	3.7	5.9	9.5	15.1	24.2	38.8	62.0

Source: © Wikibon 2015, based on Wikibon 2013 Projections

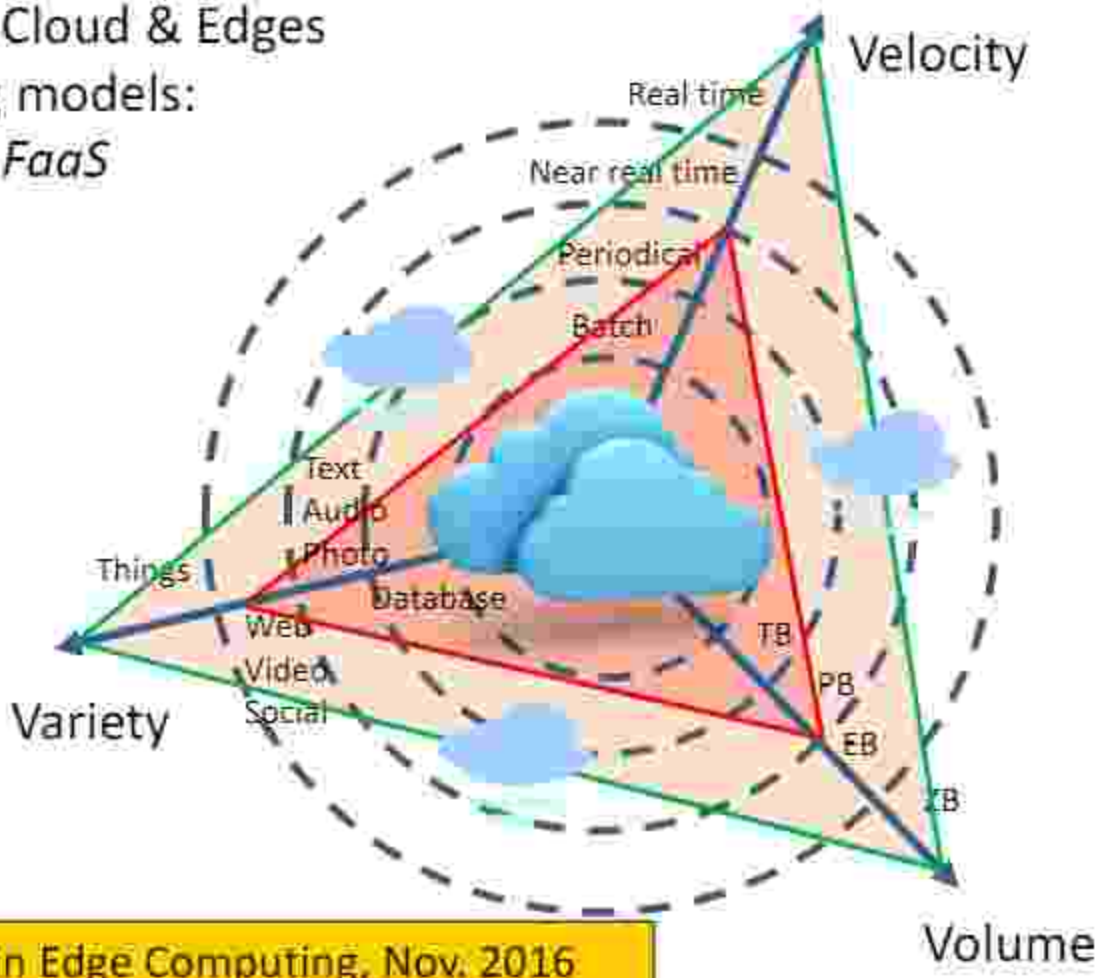
The 3-Tier Edge Compute Model



Digital Infrastructure 2.0 (15-25)

Key Features

- Data distributed at Cloud & Edges
- Possible processing models:
  Cloudlet, Firework, FaaS



NSF Workshop on Grand Challenges in Edge Computing, Nov. 2016
http://iot.eng.wayne.edu/edge/program.php

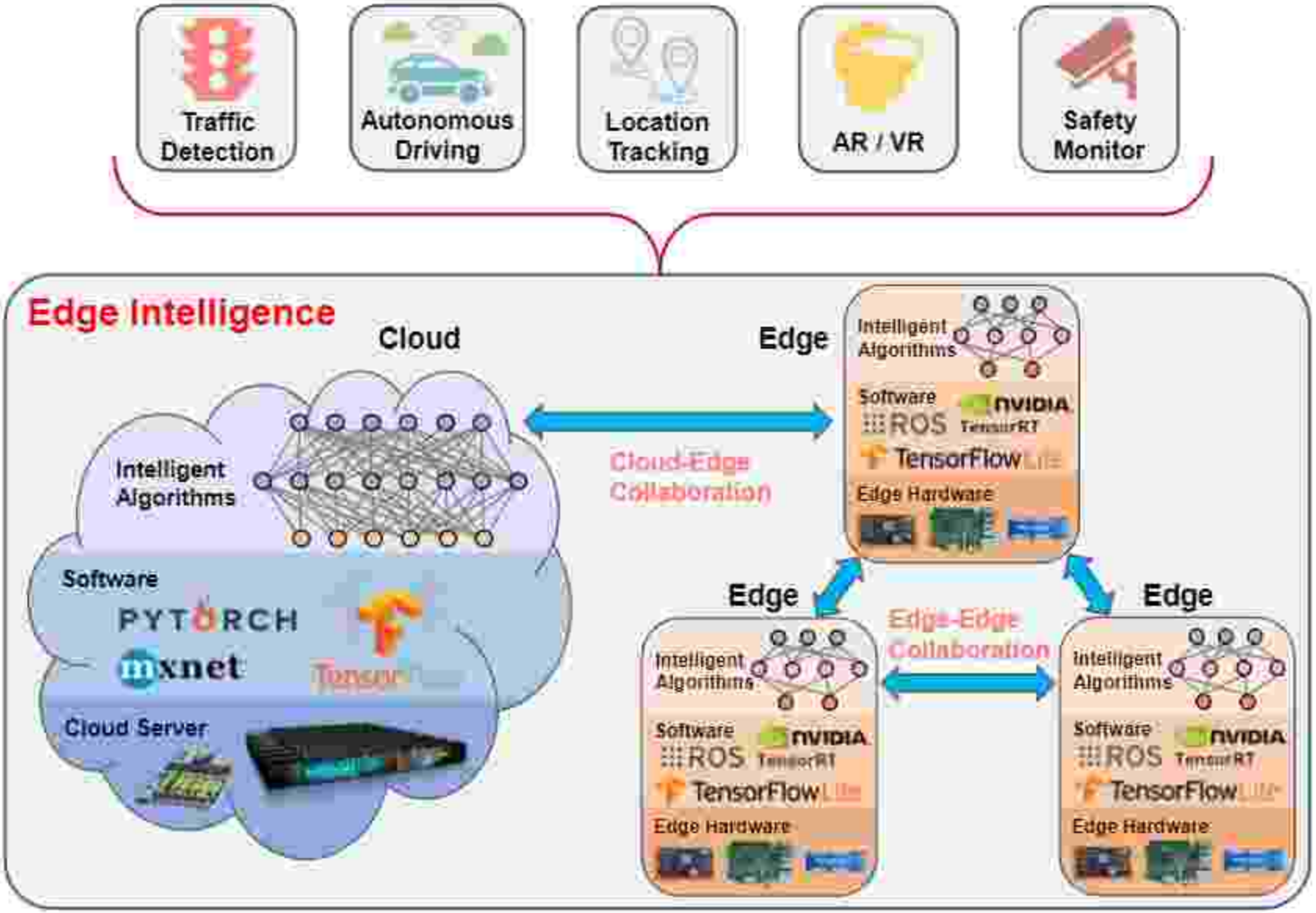
Roadmap

- Why Edge Computing?

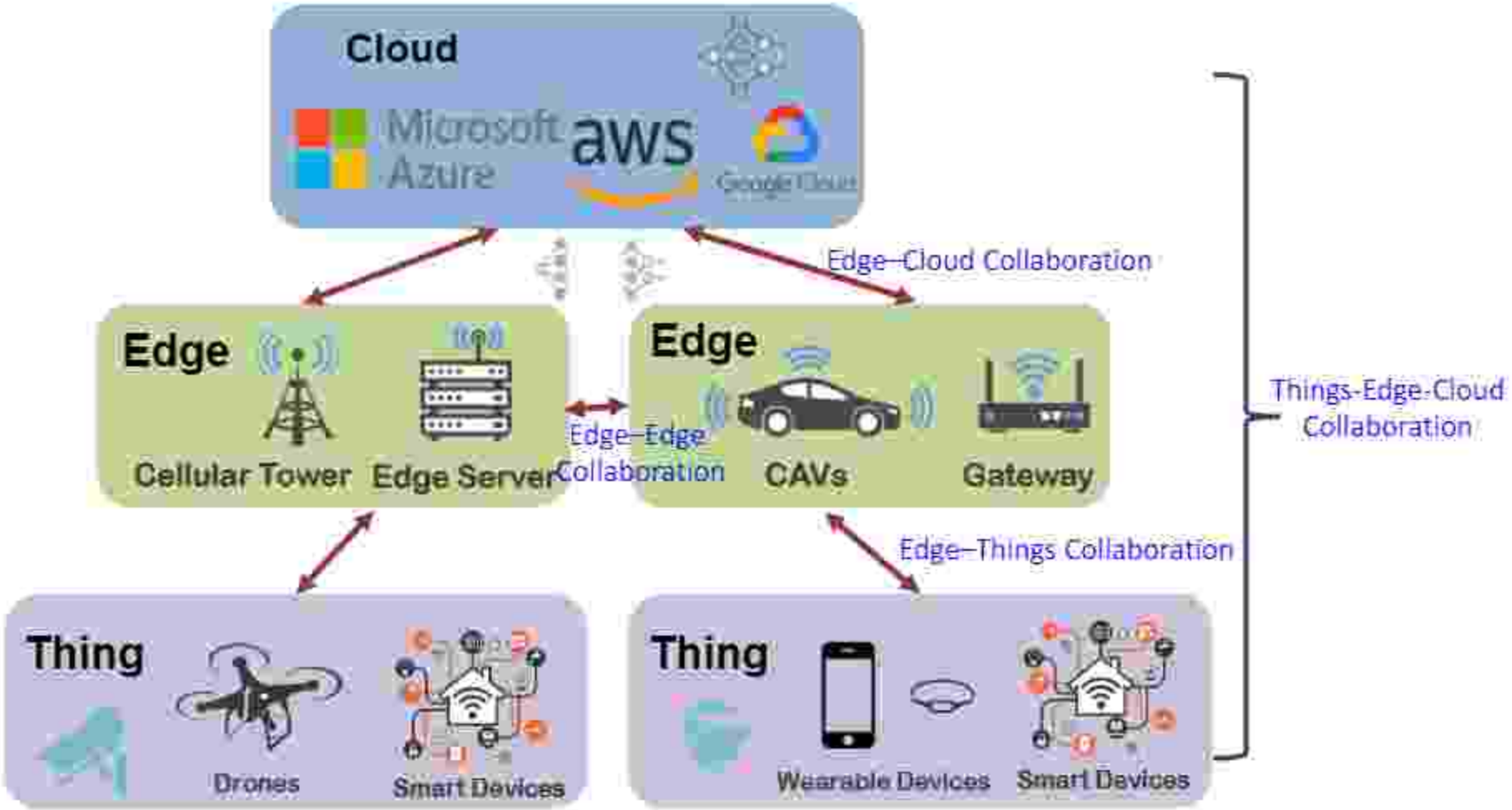
☞ Edge Intelligence

- Edge Computing in CAVs

Edge Intelligence

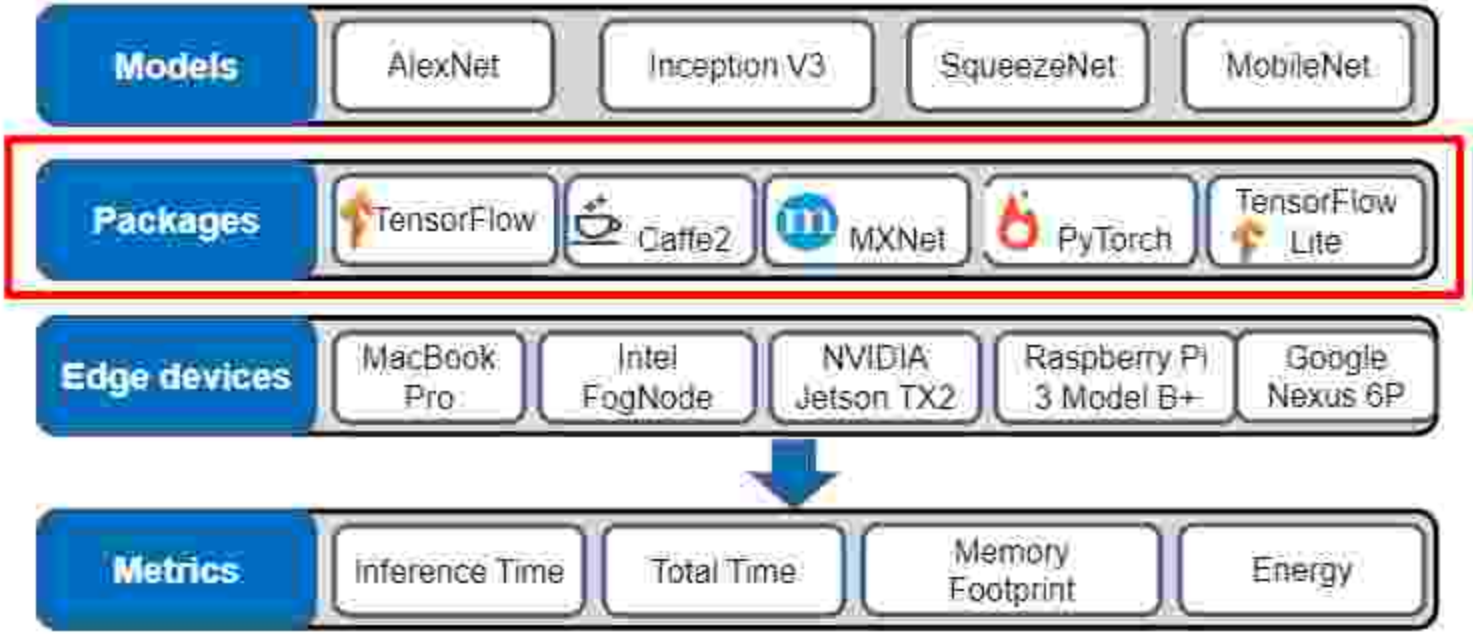


Collaborative Edge Intelligence



Quyuan Luo, Shihong Hu, Changle Li, Guanghui Li, and Weisong Shi, *Resource Scheduling in Edge Computing: A Survey*, to appear **IEEE Communications Surveys and Tutorials**, August 2021.

Performance Comparison (HotEdge'18)



pCAMP-Observations

	Model	Inference time	Total time	Memory footprint	Energy
Macbook	AlexNet	MXNet	PyTorch	PyTorch	PyTorch
	InceptionV3	MXNet	MXNet	MXNet	MXNet
	SqueezeNet	Caffe2	Caffe2	MXNet	Caffe2
	MobileNet	MXNet	MXNet	MXNet	MXNet
FogNode	AlexNet	TensorFlow	MXNet	PyTorch	MXNet
	InceptionV3	TensorFlow	TensorFlow	MXNet	TensorFlow
	SqueezeNet	Caffe2	Caffe2	Caffe2	Caffe2
	MobileNet	MXNet	MXNet	MXNet	MXNet
Jetson TX2	AlexNet	PyTorch	PyTorch	MXNet	PyTorch
	InceptionV3	MXNet	MXNet	MXNet	MXNet
	SqueezeNet	MXNet	MXNet	Caffe2	MXNet
	MobileNet	MXNet	MXNet	MXNet	MXNet

legends

- TensorFlow
- Caffe2
- PyTorch
- MXNet

Roadmap

- Why Edge Computing?
- Edge Intelligence
- ☞ Edge Computing in CAVs

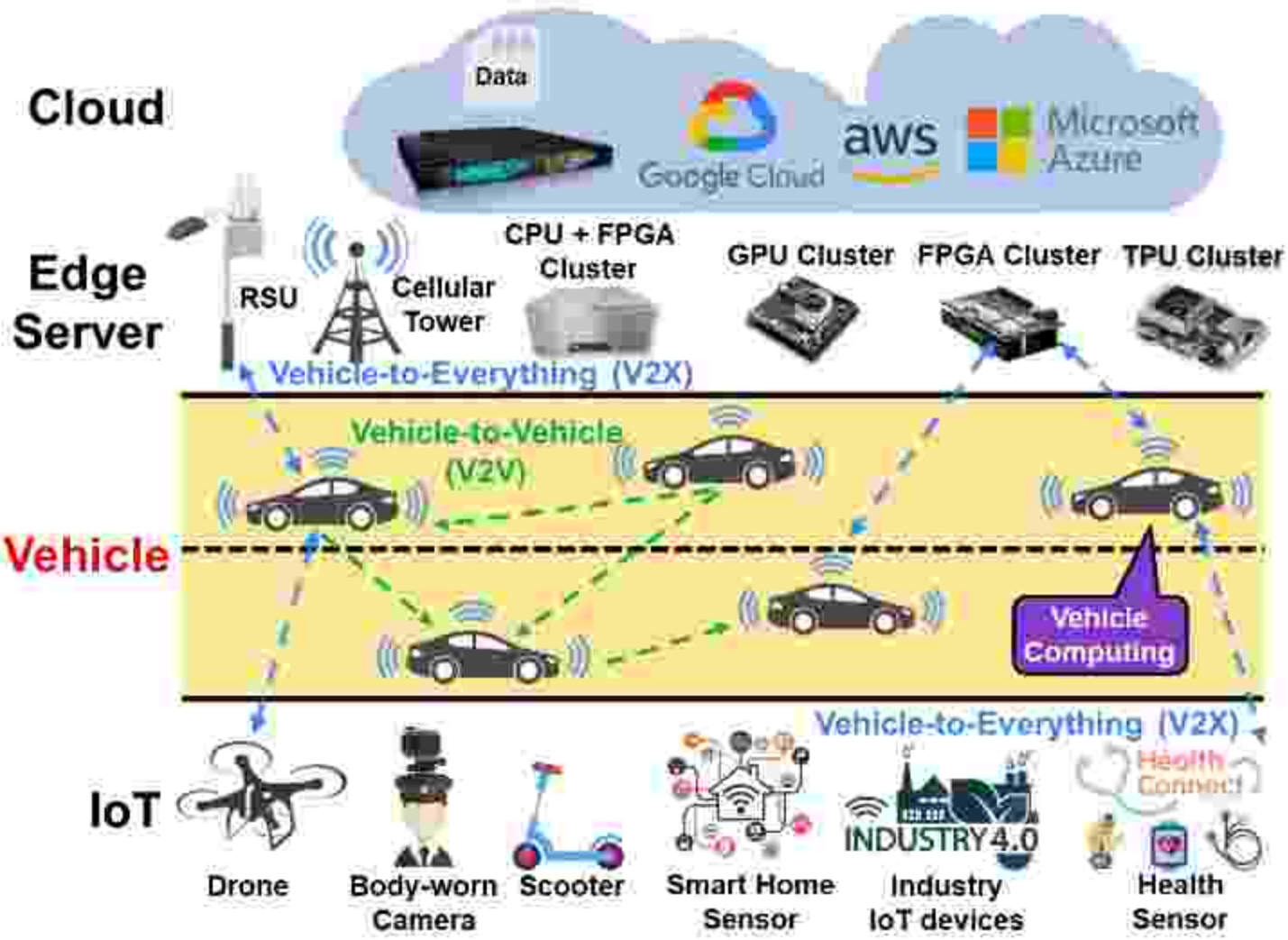
The Emergence of Vehicle Computing

(Internet Computing, 2021)

The 4-Tier **Vehicle Computing** Paradigm

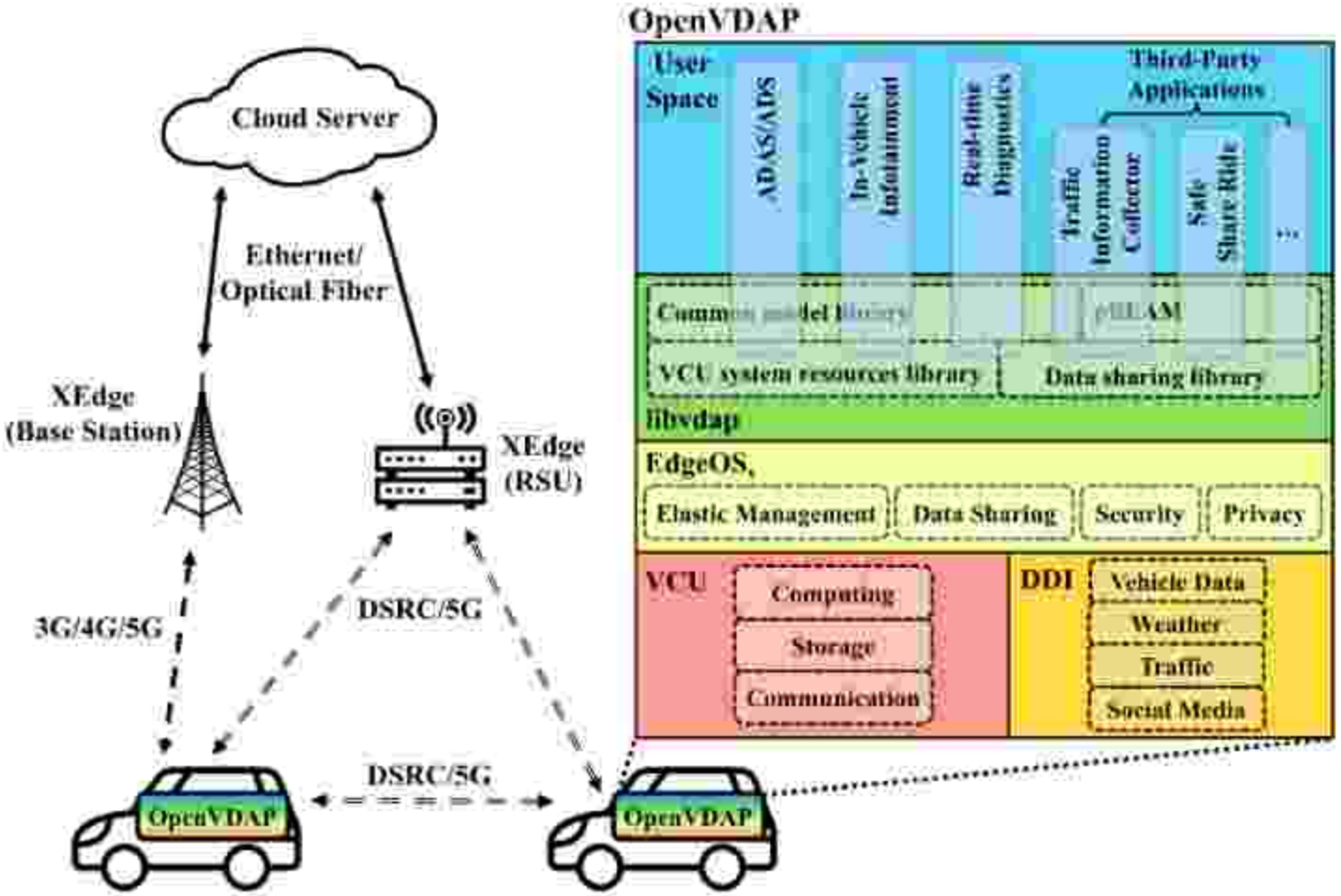
Computing on CVs based on data from:

- In-vehicle sensors,
- Surrounding **connected devices**



OpenVDAP (ICDCS'18)

- **Open** **V**ehicular **D**ata **A**nalytics **P**latform

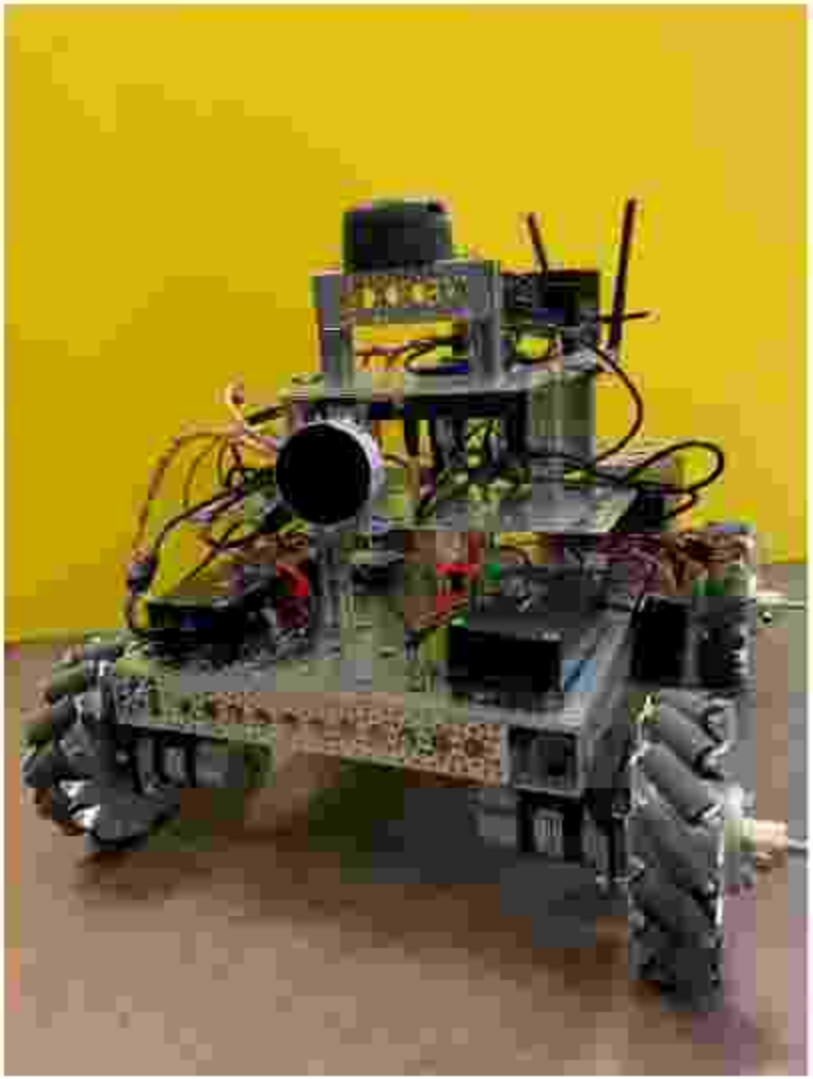


Research Activities

- Benchmarking/Platforms for CAVs
  - CAVBench/ADBench/HydraOne/HydraMini/Zebra
- Machine learning on heterogeneous platforms
  - pBEAM, pCAMP, DIME, CLONE, HydraView
- Operating systems for vehicle
  - OpenVDAP, HydraOS, E2M
- Collaborative Cloud-Edge analytics
  - Firework,, EdgeCompression, VEC, NLUBroker
- Communication optimization
  - STREMS, CHA, 4C
- Privacy and Security
  - EdgeMask, AC4AV
- Safety Applications
  - SafeShareRide, AAA, AutoVAPS
  - OpenEdgeMap, EVBatteryGuard

The HydraOne Platform

Research Platforms

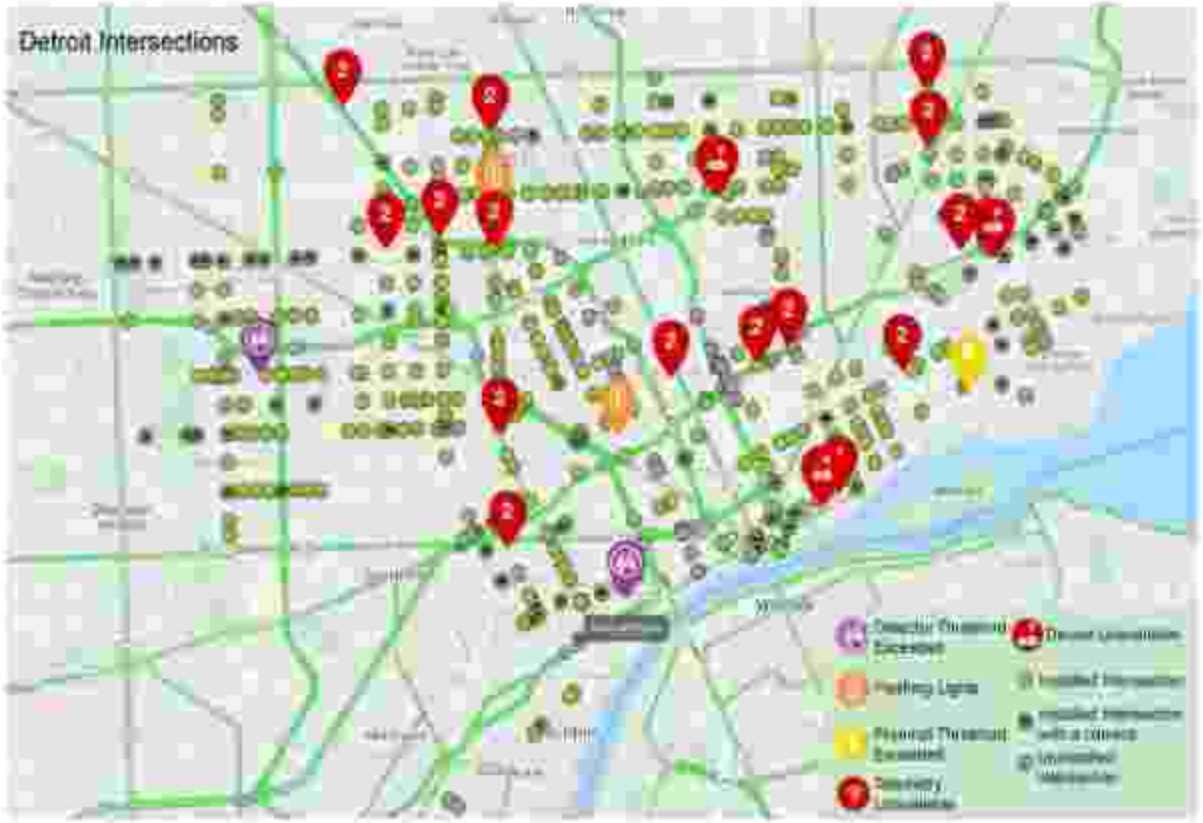


Road-Side Unit: Equinox (HotEdge'20)

LTE, DSRC, WiFi, 5G

Hydra (OBU)

Ongoing Collaboration



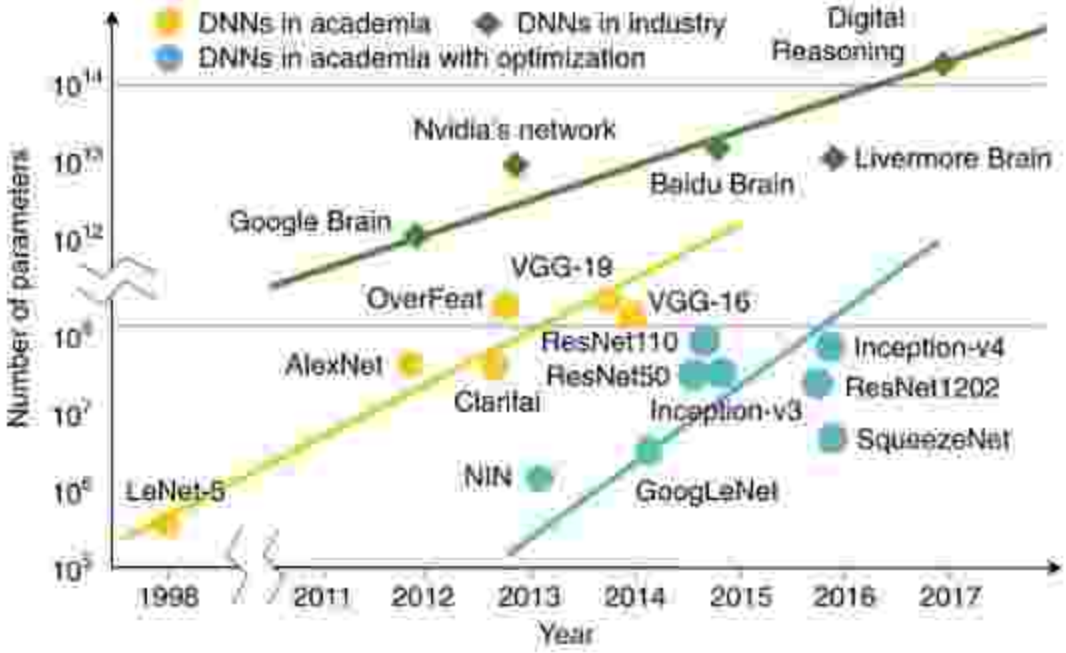
Edge Latency-Aware Model Scheduling on Heterogeneous Platforms

Part II

Motivation

Background

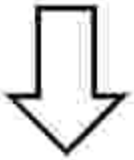
- Explosive growth in **model size**
- Increasing **#models** and **#devices**
- Collaboration of multiple oneDNN models on diverse edge devices



How to concurrently & efficiently deploy and execute the **codependent models** on **heterogeneous devices**?

Previous Work

- **One-to-One**
  - An optimal DNN architecture to best fit a particular target device
  - A single specific application scenario



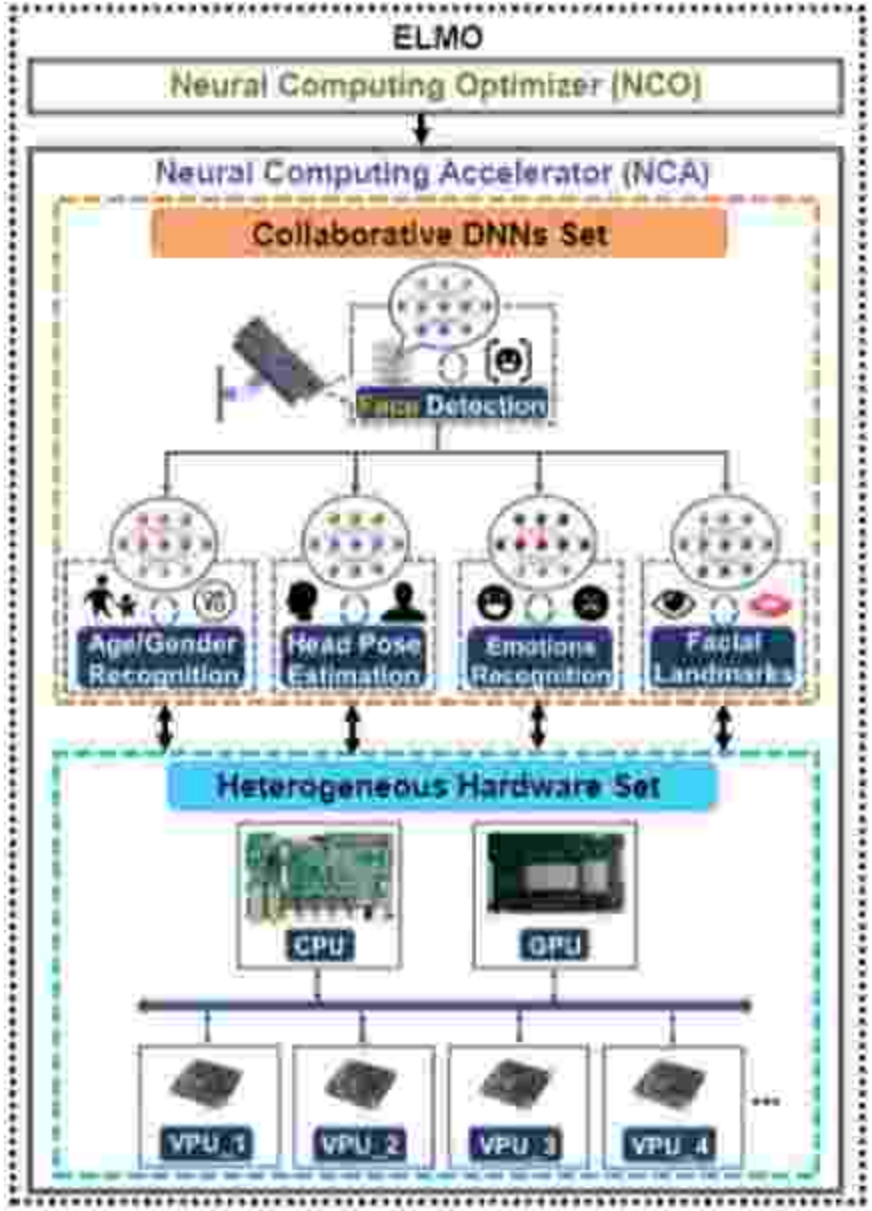
- **Issue**
  - Too **resource demanding** (case-by-case deployment)
  - **Not practical enough** considering the involvement of multi-models and diverse devices at the same time

Innovation

- **Many-to-Many**
  - Scheduling multiple collaborative DNNs on a group of heterogeneous edge devices (named ELMO)
  - Based on **oneDNN** & **OpenVINO**

- **Key contributions**
  - Pioneer that points out the importance of this new research direction, i.e., Many-to-Many
  - **Accelerate** the inference of multiple models on the heterogeneous edge devices by up to **25.33%**

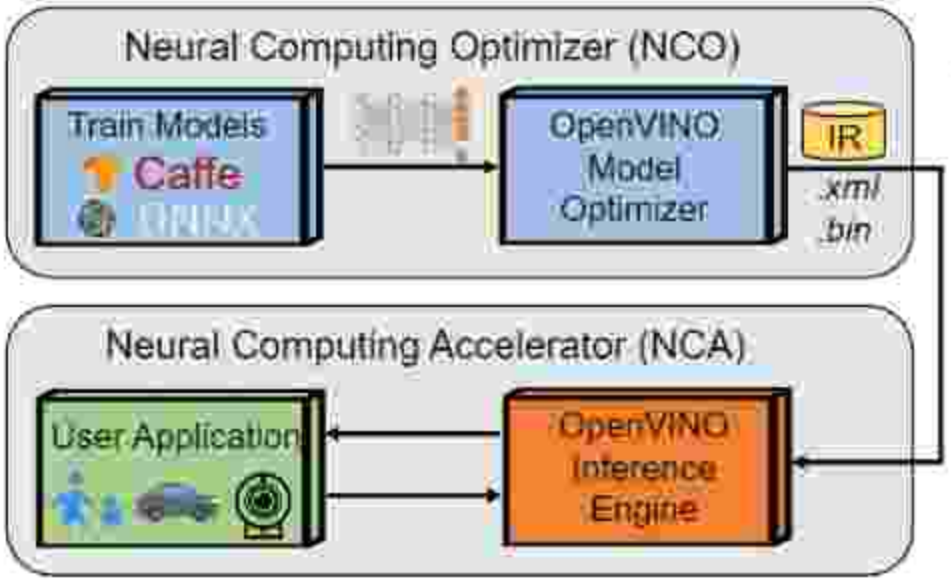
ELMO Framework



- **Two Key Components**
  - **Neural Computing Optimizer (NCO)**
    - Train, optimize and transform DNNs to the *hardware-specific format*
  - **Neural Computing Accelerator (NCA)**
    - Multiple collaborative DNNs *scheduling* on heterogeneous edge devices

Core Components of ELMO

- **NCO**
  - Deploy OpenVINO Model Optimizer
  - Convert models trained from ***different*** *ML frameworks* into a ***unified*** Intermediate Representation *(IR) format*
    - **.xml file**: network structure of the model
    - **.bin file**: weights and biases of the model

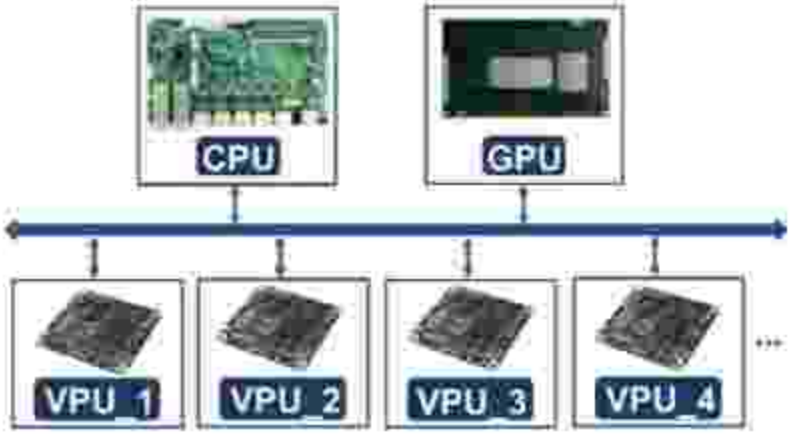


- **NCA**
  - Deploy OpenVINO Inference Engine
  - Takes the corresponding input data and calls DNN models for inference
  - Supports hardware devices such as CPU, GPU, FPGAs, VPU

Experimental Platform

- **Intel Fog Reference Design**
  - One **CPU** and one **GPU**
  - Support scalable Intel MyriadX **VPUs**

(a)

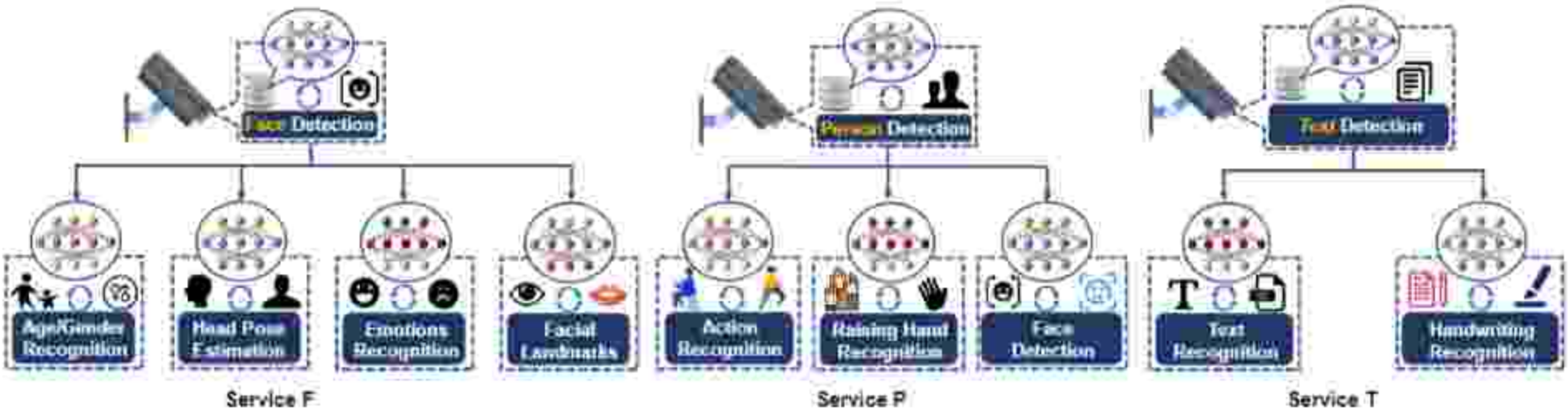


(b)

	Core	Computation Power / GFLOPS
CPU	Xeon(R) E3-1275	33.0
GPU	HD Graphics P530	768.0
VPU	NCE + Shave	358 / NCE × 2, 22.4 / Shave × 16

Three Model Groups

- **Image and text recognition**
  - Human face (service F)
  - Person (Service P)
  - Text recognition (service T)



Single model FPS

Experiment Observations

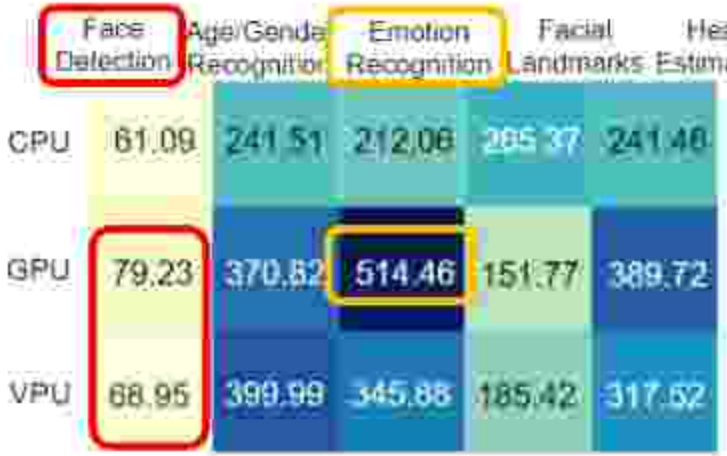
Computing power (GFLOPS): VPU > GPU > CPU

	Core	Computation Power / GFLOPS
CPU	Xeon(R) E3-1275	33.0
GPU	HD Graphics P530	768.0
VPU	NCE + Shave	358 / NCE × 2, 22.4 / Shave × 16

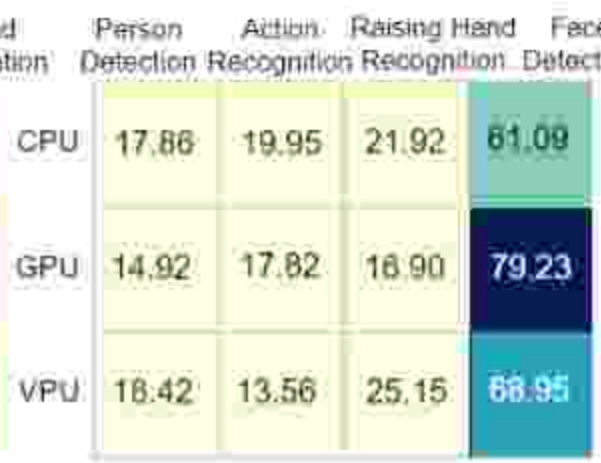
1. The computational complexity of model is ***not necessarily*** related to its processing efficiency on a given edge device.
2. The computation capability of an edge device ***cannot determine*** the processing efficiency for the inference task

Not the smallest

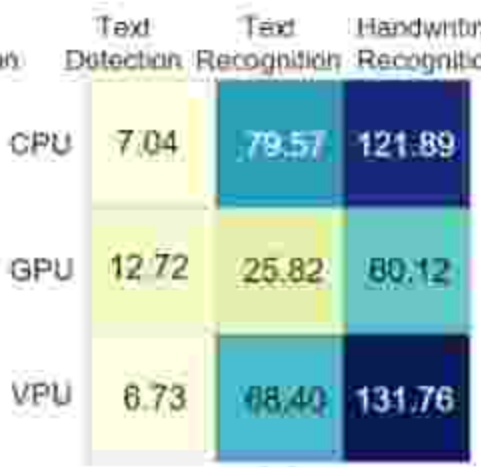
	DNN Model	GFLOPs
Service F	Face Detection	2.835
	Age/Gender Recognition	0.047
	Emotions Recognition	0.053
	Facial Landmarks	0.064
	Head Pose Estimation	0.021
Service P	Person Detection	7.140
	Action Recognition	8.225
	Raising Hand Recognition	7.138
	Face Detection	2.835
Service T	Text Detection	23.305
	Text Recognition	1.485
	Handwriting Recognition	0.792



(a)



(b)



(c)

Model-first Scheduling (MFS)

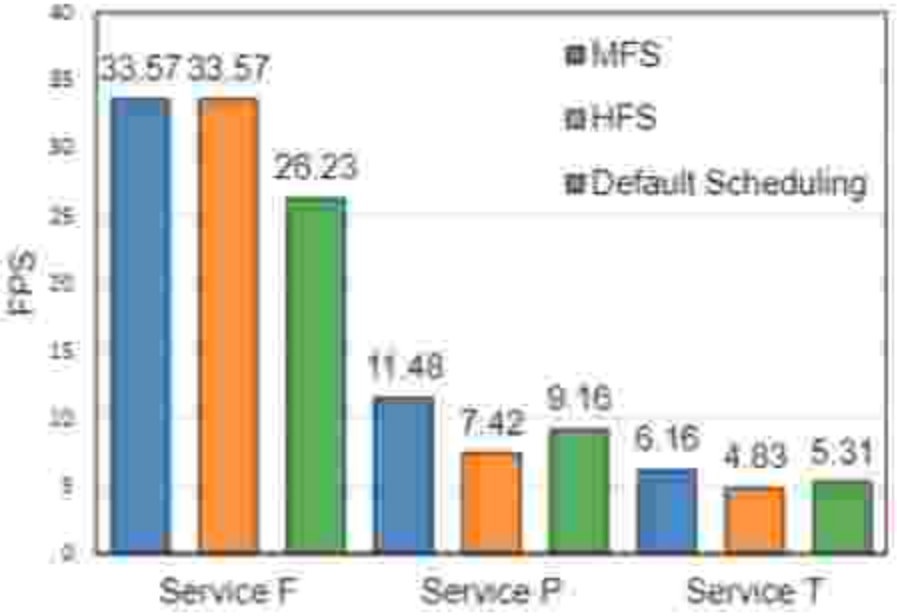
- **Worst-Case-First (Amdahl's Law)**
  - No matter how fast a single model can run, the overall inference speed of the whole service is highly dependent on where the slowest speed is

- **Two scheduling methods**
  - **Model-first Scheduling (MFS)**
    - Targets at finding an appropriate computation device for a specific model
  - **Hardware-first Scheduling (HFS)**
    - Targets at finding an appropriate model for an edge device

Single Service FPS after Scheduling

- Three sets of models will be assigned to their most suitable edge devices

	DNN Model	MFS	HFS
Service F	Face Detection	GPU	GPU
	Age/Gender Recognition	VPU	VPU
	Emotions Recognition	VPU	VPU
	Facial Landmarks	CPU	CPU
	Head Pose Estimation	VPU	VPU
Service P	Person Detection	VPU	VPU
	Action Recognition	CPU	CPU
	Raising Hand Recognition	GPU	VPU
	Face Detection	VPU	VPU
Service T	Text Detection	GPU	GPU
	Text Recognition	CPU	VPU
	Handwriting Recognition	VPU	VPU

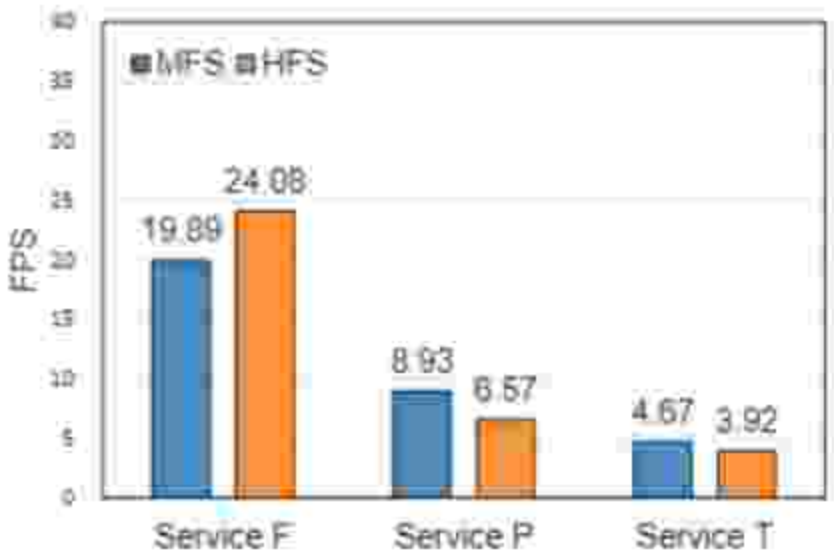


- Service F: MFS & HFS leads to the same FPS(33.57), **25.33% higher** than default FPS (26.23)
- Service P and Service V: ELMO improves FPS by 2.58%

Model Scheduling for Multiple Services

- Three sets of 12 models assigned to their most suitable edge devices
- Overall FPS for all services & models are calculated together

	DNN Model	MFS	HFS
Service F	Face Detection	GPU	GPU
	Age/Gender Recognition	VPU	VPU
	Emotions Recognition	VPU	VPU
	Facial Landmarks	CPU	CPU
	Head Pose Estimation	VPU	VPU
Service P	Person Detection	VPU	VPU
	Action Recognition	CPU	CPU
	Raising Hand Recognition	GPU	VPU
	Face Detection	VPU	VPU
Service T	Text Detection	GPU	GPU
	Text Recognition	CPU	VPU
	Handwriting Recognition	VPU	VPU



- Service F/P/V shows **better** FPS than default FPS scheduling

Conclusion

- Prove the importance of model scheduling for **multiple DNNs and heterogeneous edge devices** with diverse computation resources
- Key concept is **Worst-Case-First** for hardware-aware models scheduling
- **Two scheduling algorithms** + evaluation results of three oneDNN groups on **CPU, GPU and multiple VPUs**
- The effectiveness of ELMO on accelerating the co-inference of multi-models on the heterogeneous edge devices by up to **25.33%**