DASYA

www.dasya.dk
@dasyaITU

RAD

rad.itu.dk

IT UNIVERSITY OF COPENHAGEN

www.itu.dk

data processing at the edge: *from satellites to earth*

Pınar Tözün

Associate Professor, IT University of Copenhagen

pito@itu.dk, pinartozun.com, @pinartozun

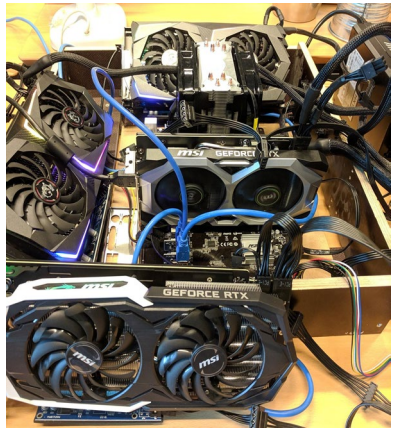
INDEPENDENT RESEARCH FUND DENMARK

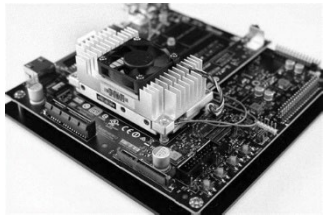
novo nordisk foundation

MDM 2024

hardware scales for data systems

large **tiny**

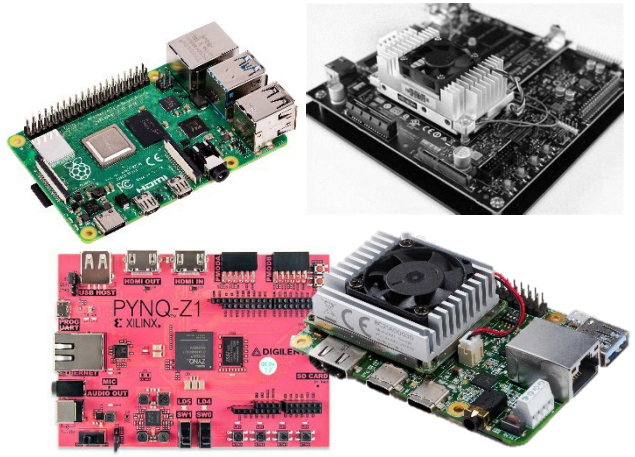

- fewer hardware resources (less processing power, memory …)
- consumes less power per unit of time
- cheaper to buy

➔can we utilize these hardware well?

➔can we do more with less?

hardware scales for data systems

large ⟷ **tiny**

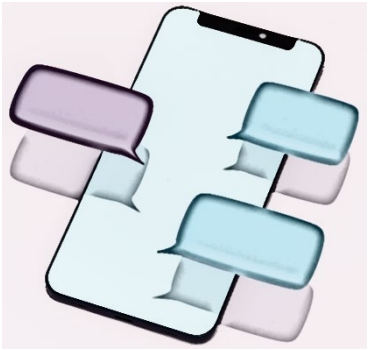


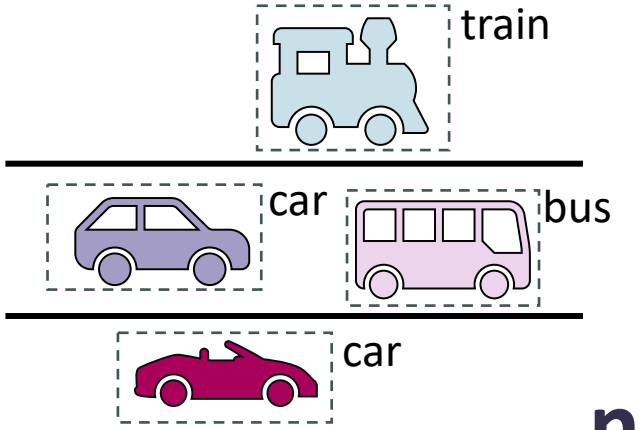
"Reaching the Edge of the Edge: Image Analysis in Space", DEEM 2024

➔can we utilize these hardware well?

➔can we do more with less?

data processing @ the edge





traditional-approach

- do (most) data processing in the cloud

cannot satisfy

- low-latency & real-time applications
- poor / non-existing connectivity
- legal restrictions & privacy

need for efficient & complex data processing closer to data sources; *at the edge*!

DISCO: Danish student CubeSat organization

https://discosat.dk/

collaboration across Danish universities

goal: education & research

- students learn how to build small satellites
- researchers experiment with ML models for Earth observation

use-case:

- observation of landmasses (especially snow, ice …) in the Arctic

small satellites

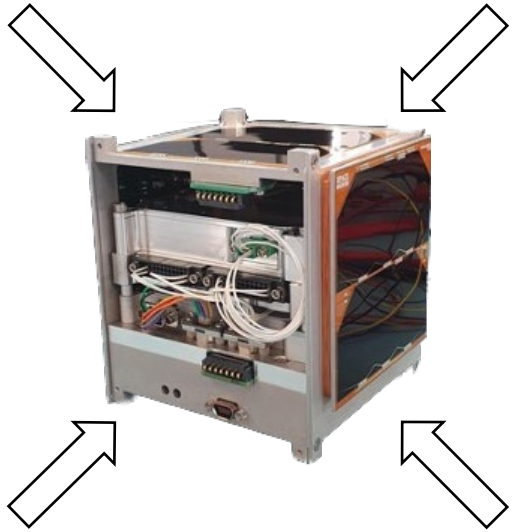
benefits

$$

⇩

$

reduced cost



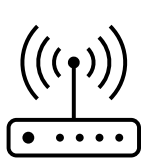
shrinking
&
standardization

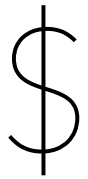
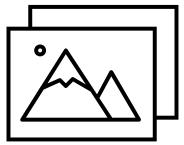
e.g., CubeSats
10cm cube

compromises

reduced power
generation

network on small satellites



uhf	9600 bit/s	low-power	low	0.05 / day*
s-band	10 Mbit/s	high-power	high	49.1 / day*

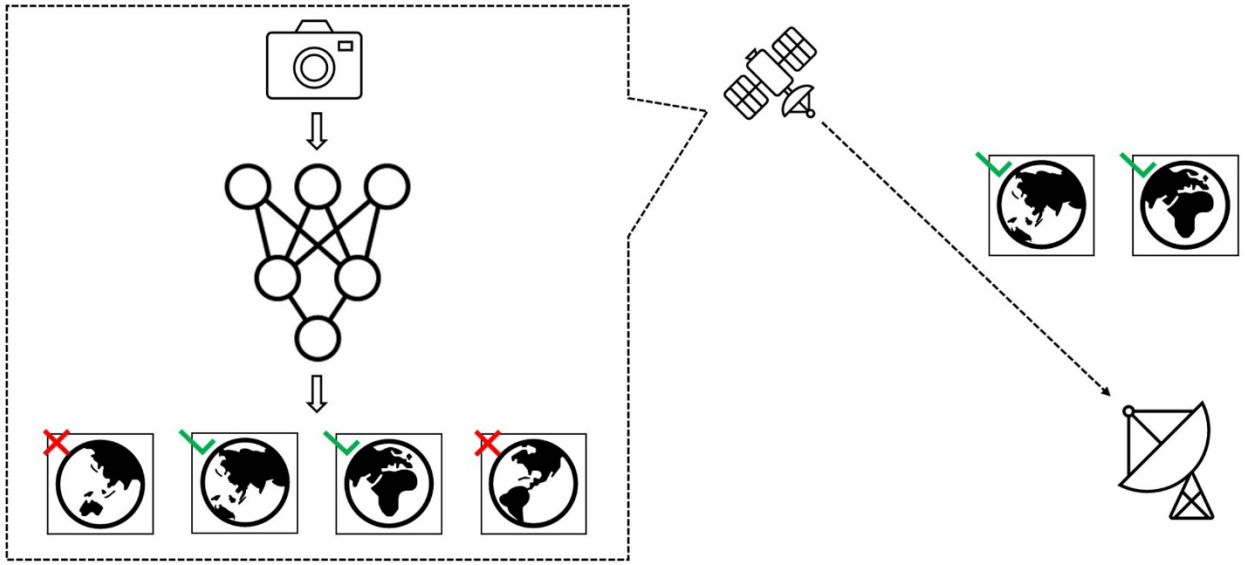
* based on 4512x4512px image & short-lived connection

cannot send all the images to Earth!

image filtering on satellites

in DISCO: ML-based image classification

➔ **send only the relevant images to ground & minimize data movement!**



our task: build the image processing unit on the satellite

➔ **determine the edge device that can satisfy the *requirements* of this task!**

image processing unit requirements

real-time imaging

< 4.42 s latency

Arctic region

< 71.74 s latency

Greenland

< 270 s latency

---

max 49.1 images can be transferred per day!

min 320 images captured per day.

---

max 5 watts

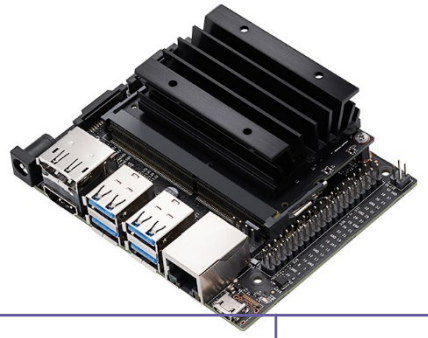
average 2 watts

***mass:* 150grams**

***dimensions:* 10x70x80mm**

---

flexible software upload!

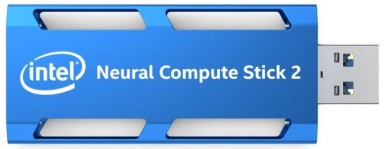



ARM Cortex-M7	Jetson Nano	Toradex Verdin	CoralAI Micro	CoralAI Mini	CoralAI USB Stick	Neural Compute Stick
ARM Cortex-M7 @300MHz	ARM A57 @1.43GHz	ARM Cortex-A53 @1.8GHz, ARM Cortex-M7 @800MHz	ARM Cortex-M7 @800MHz, ARM Cortex-M4 @400MHz	ARM Cortex-A35 @1.5GHz	Raspberry Pi 3 BCM2837 ARM @1.2GHz	
384KB SRAM, 32KB FRAM	4GB	4GB	64MB	2GB	1GB	
none	128-core Maxwell GPU	NPU (2.25 TOPS)	CoralAI Edge TPU (4 TOPS)			Intel Movidius Myriad X VPU

general-purpose ⟷ **higher specialization for ML**

hardware setup



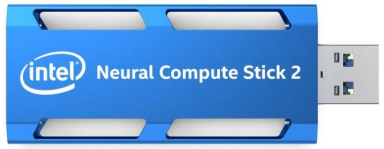



	ARM Cortex-M7	Jetson Nano	Toradex Verdin	CoralAI *	Neural Compute Stick 2
framework	TensorFlow Lite for Microcontrollers	TensorRT	TensorFlow Lite with OpenVX backend	TensorFlow Lite	OpenVino

each require separate optimizations!

hardware setup



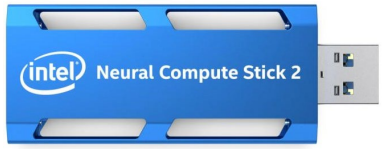



	ARM Cortex-M7	Jetson Nano	Toradex Verdin	CoralAI *	Neural Compute Stick 2
framework	TensorFlow Lite for Microcontrollers	TensorRT	TensorFlow Lite with OpenVX backend	TensorFlow Lite	OpenVino
quantization	8bit (to fit the device memory)	16bit	8bit (only supports 8bit ints)	8bit (only supports 8bit ints)	16bit (only supports 16bit floats)

each require separate optimizations!

hardware setup

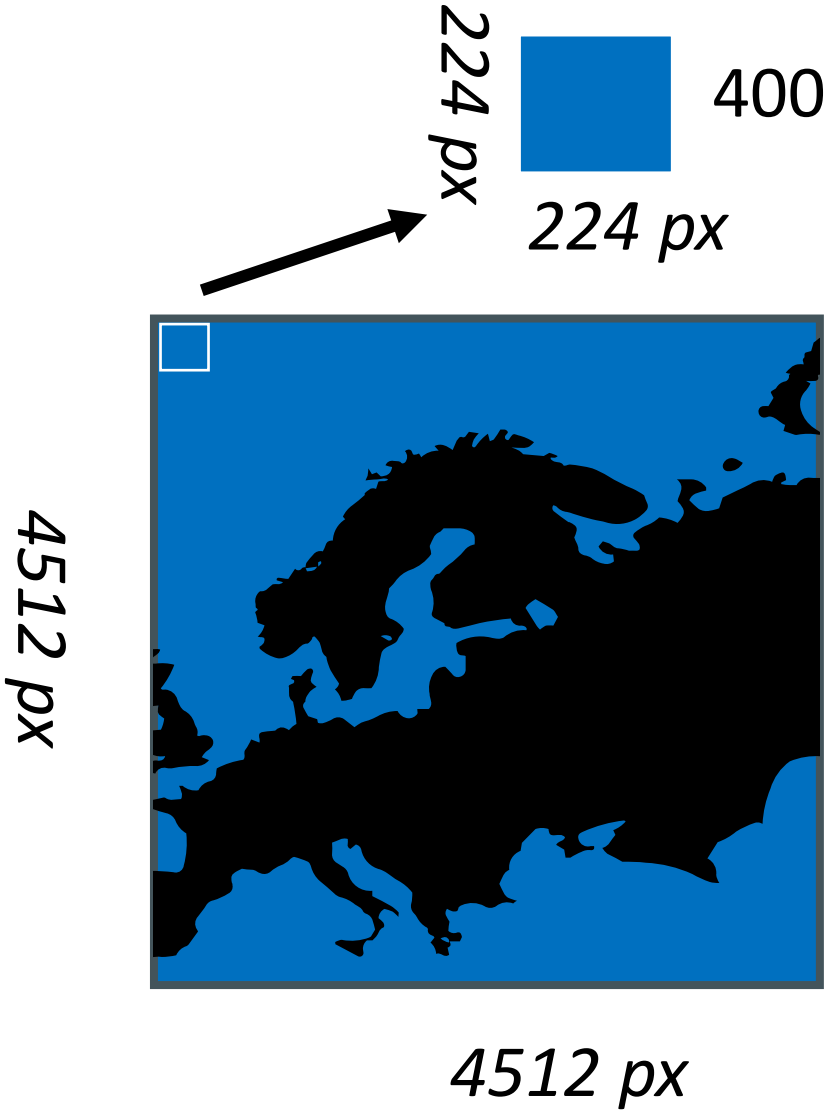


	ARM Cortex-M7	Jetson Nano	Toradex Verdin	CoralAI *	Neural Compute Stick 2
framework	TensorFlow Lite for Microcontrollers	TensorRT	TensorFlow Lite with OpenVX backend	TensorFlow Lite	OpenVino
quantization	8bit (to fit the device memory)	16bit	8bit (only supports 8bit ints)	8bit (only supports 8bit ints)	16bit (only supports 16bit floats)
batching	not enough memory to do batching	batch size per inference = 16	doesn't support batching	doesn't support batching	number of concurrent inference requests = 4

each require separate optimizations!

data & full-image inference



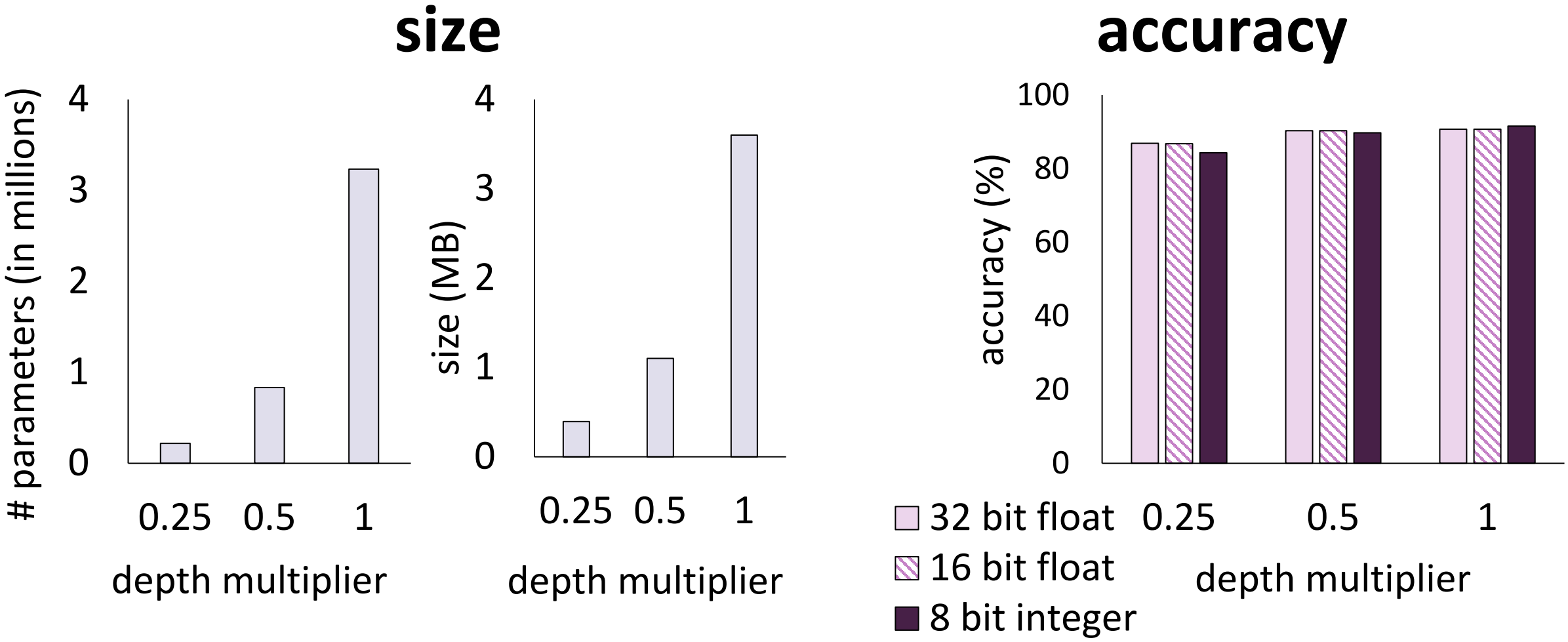
per-image inference latency

= time to infer 400 patches

performance benefits

- parallelizing per-image inference
- sending only the relevant parts of the whole image

model – pretrained MobileNetV1

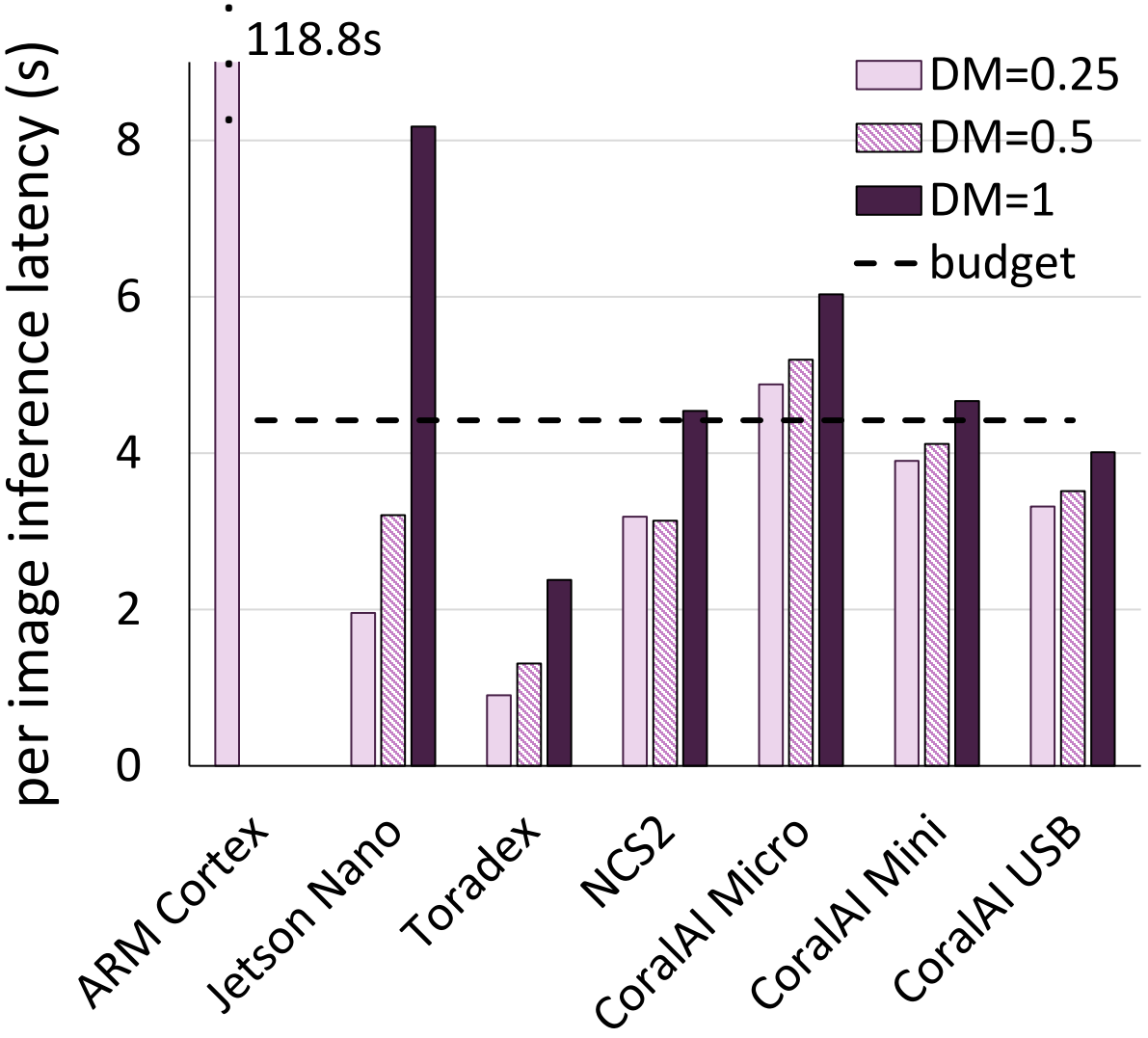


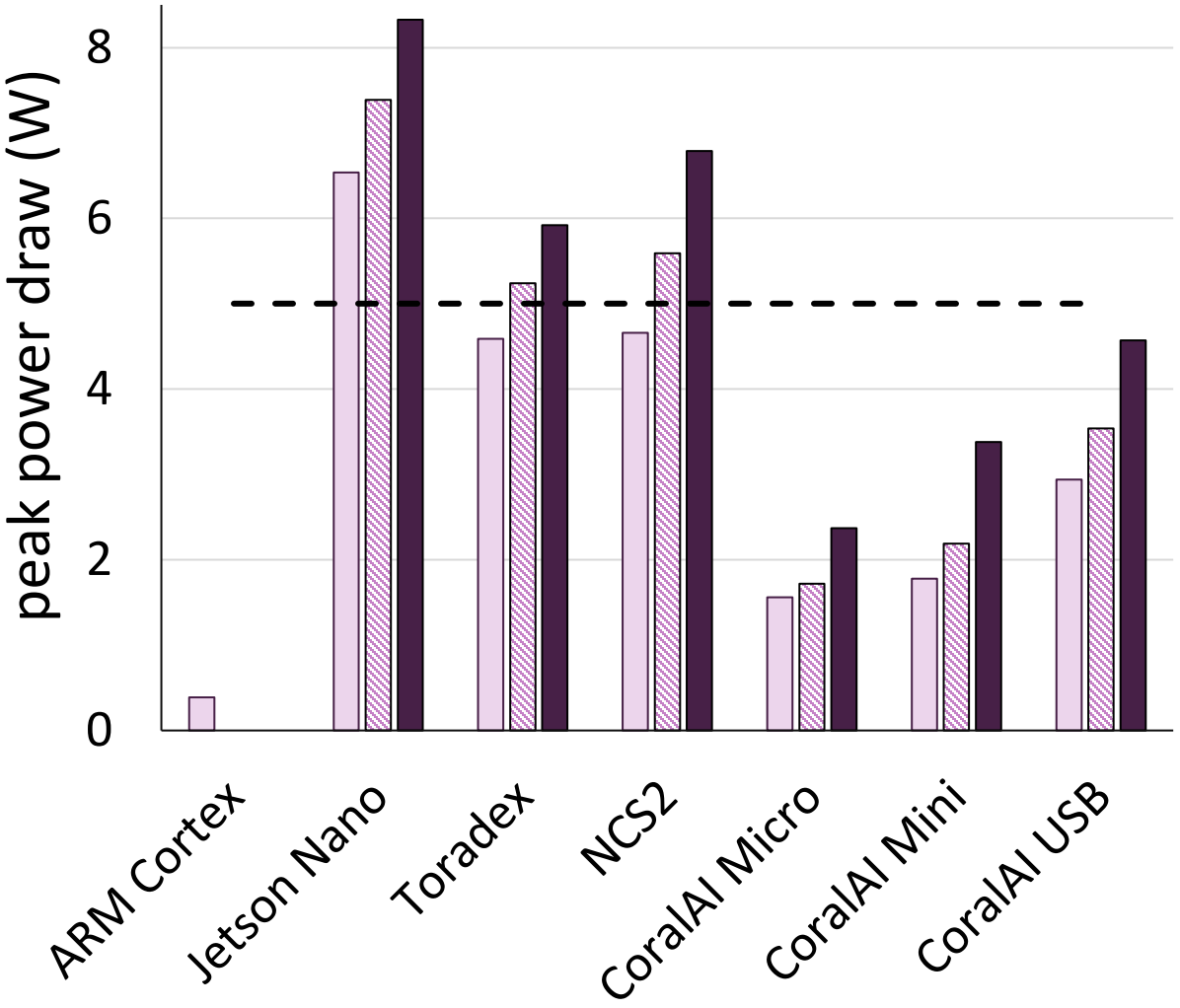
model size impacts resource consumption & updatability.

quantization does not have big impact on accuracy.

latency & power draw

DM = depth multiplier

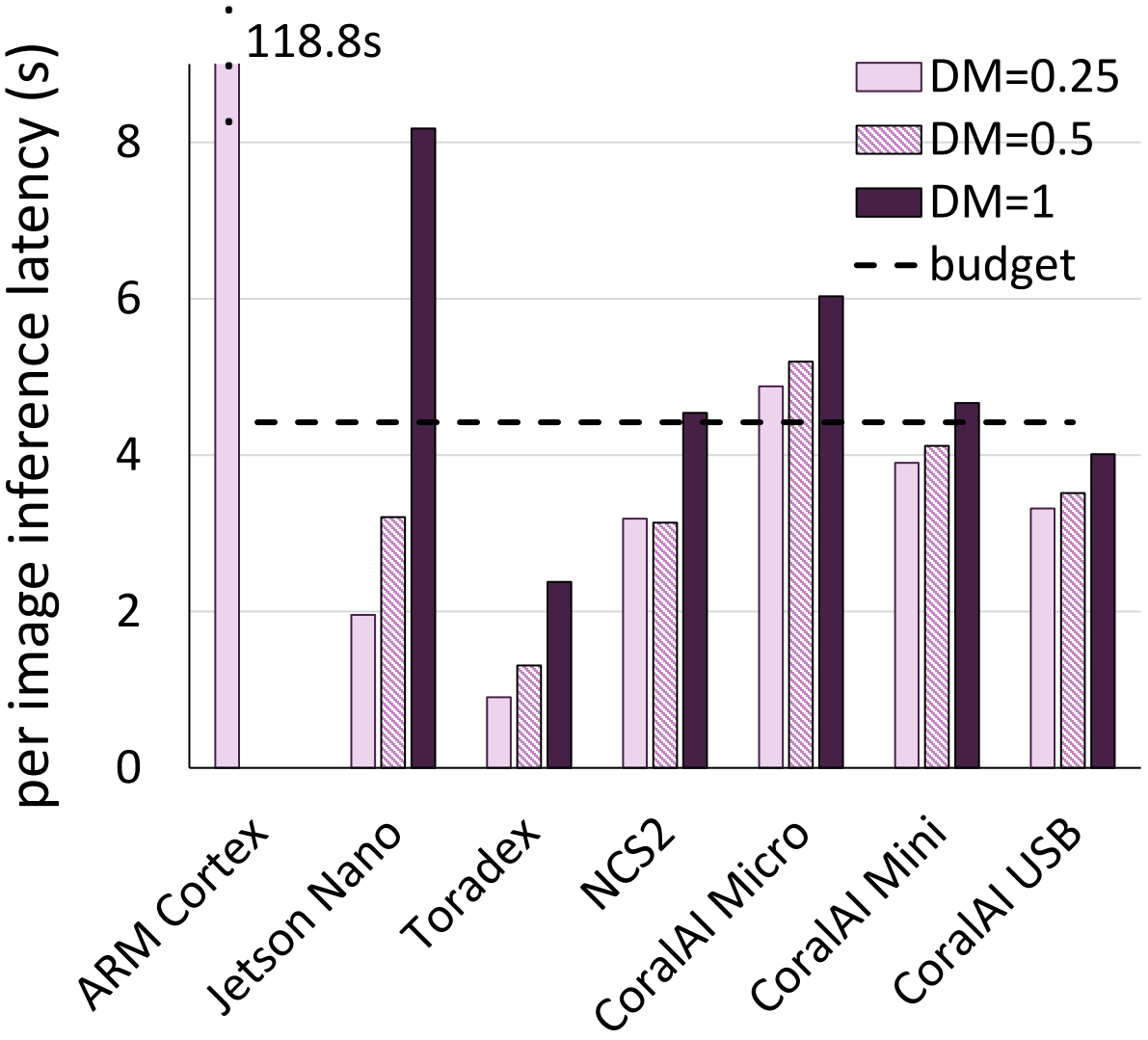


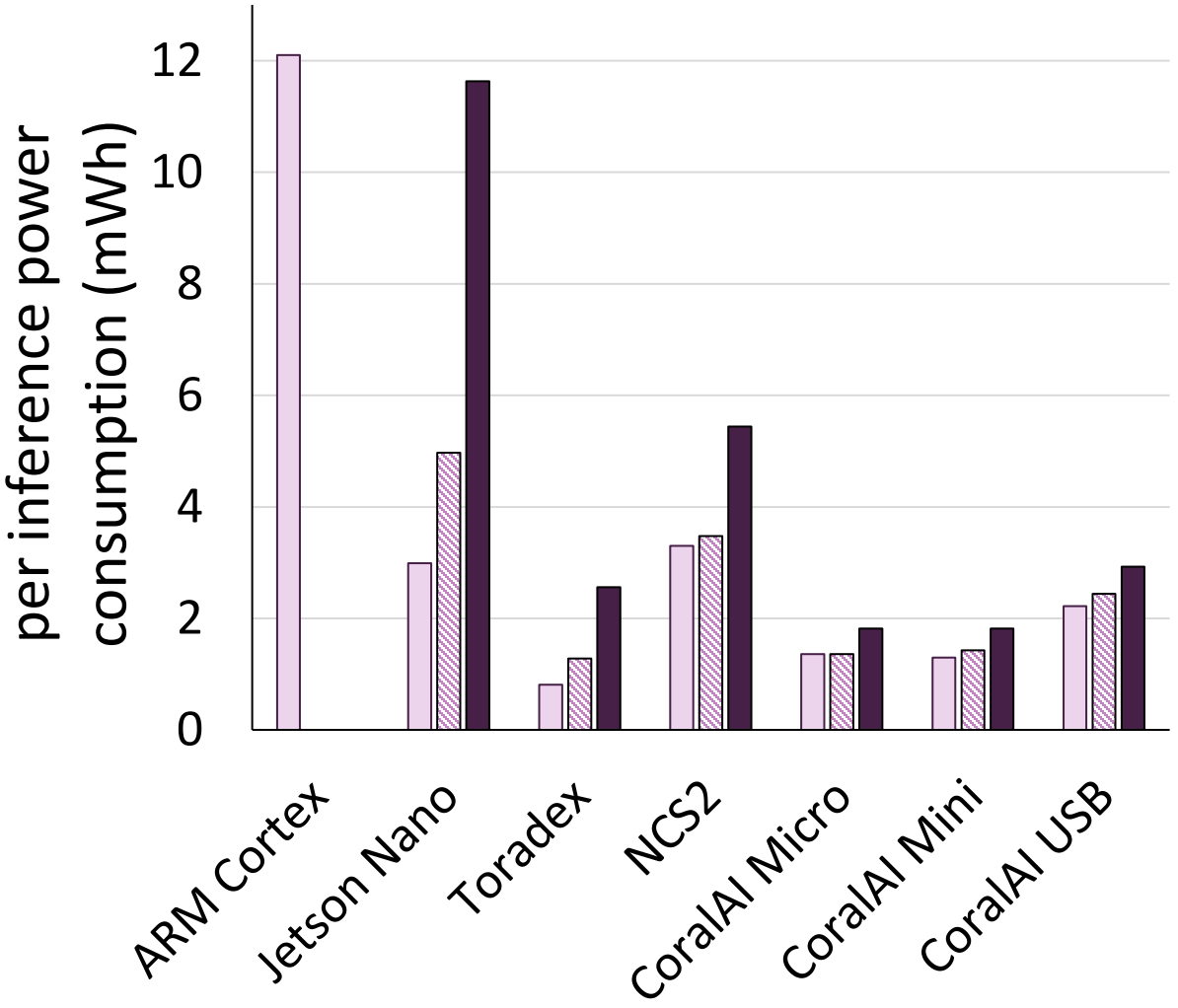


Jetson & NCS2 have low-latency for smaller models, Toradex is best.
faster devices fail the peak power budget.

latency & power draw

DM = depth multiplier

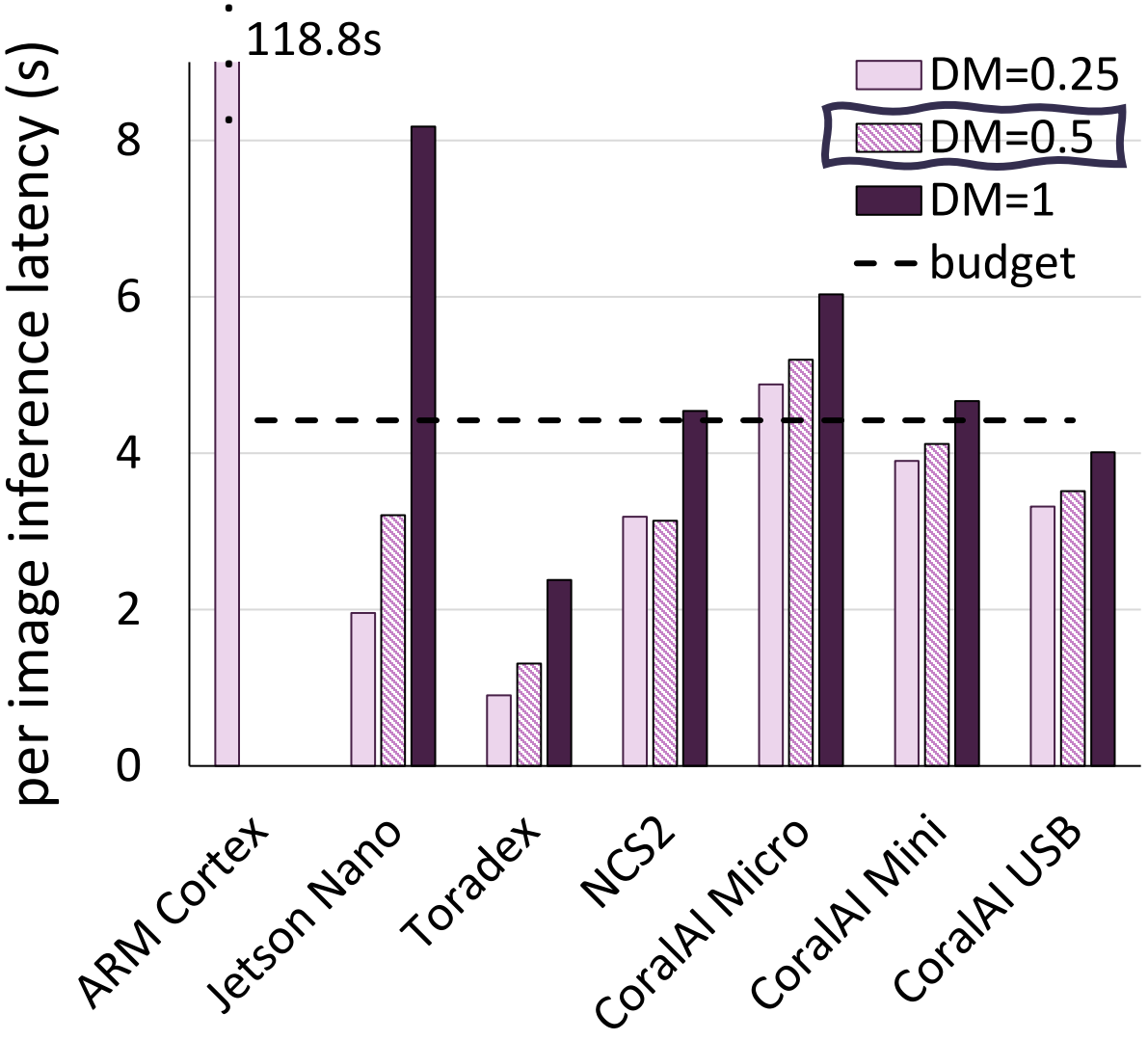


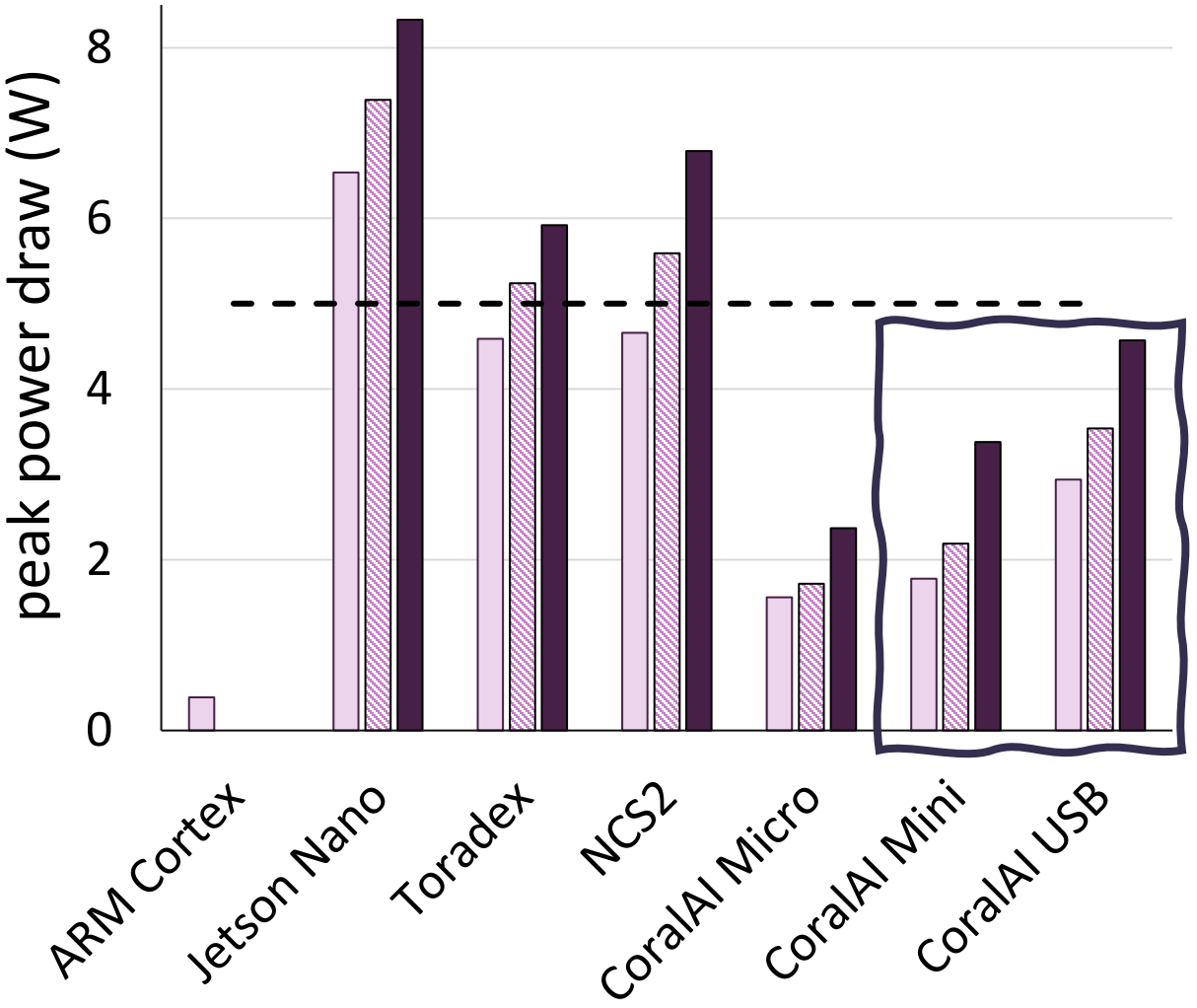


ARM-based microcontroller draws little power per unit time but per inference power need is higher than the rest!

latency & power draw

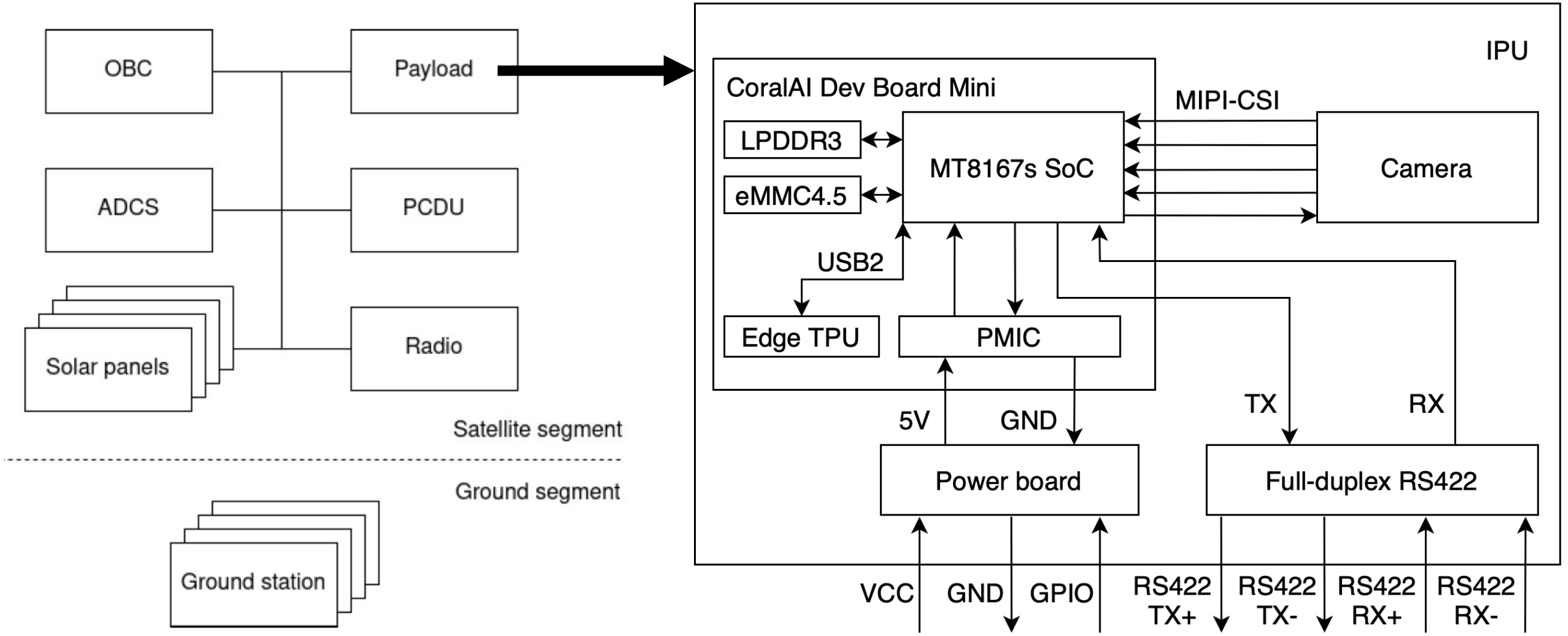
DM = depth multiplier



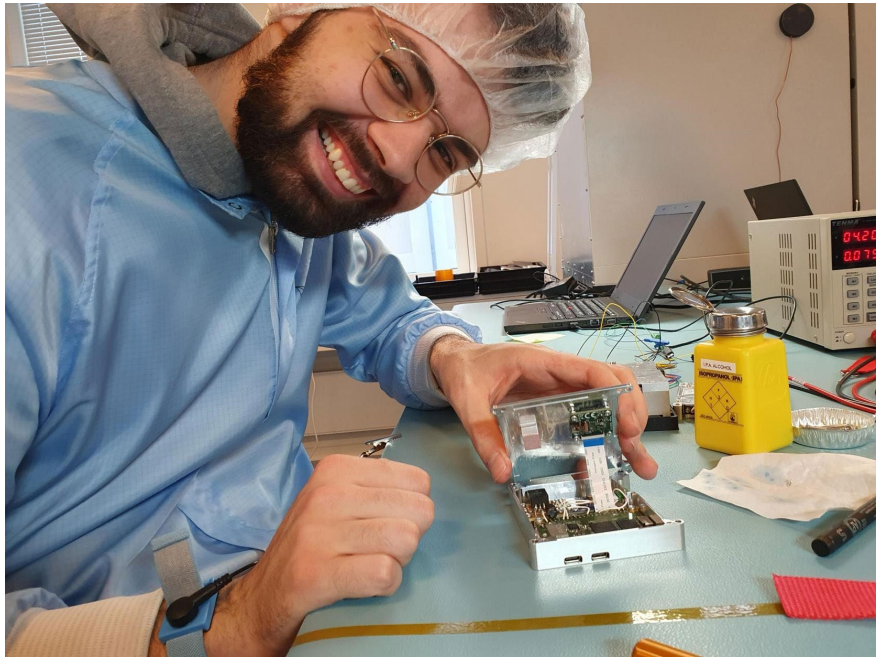


CoralAI Mini & USB satisfy both latency and power budget.

image processing unit = IPU



DISCO satellite



Students launch a satellite to test artificial intelligence in space

On April 14, students from ITU will contribute to writing space history. The satellite, DISCO-1, is launched into space and it carries a microcomputer to test artificial intelligence outside the atmosphere. The satellite is developed by the space program, DISCO, which is a collaboration between students from four Danish universities.

IT-Universitetet i København
April 15 ·

Så lykkedes det! 🚀🛰️✨

Satellitten DISCO-1, udviklet af danske studerende fra bl.a. ITU, blev her til morgen sendt ud i rummet med SpaceX' raket fra Californien.

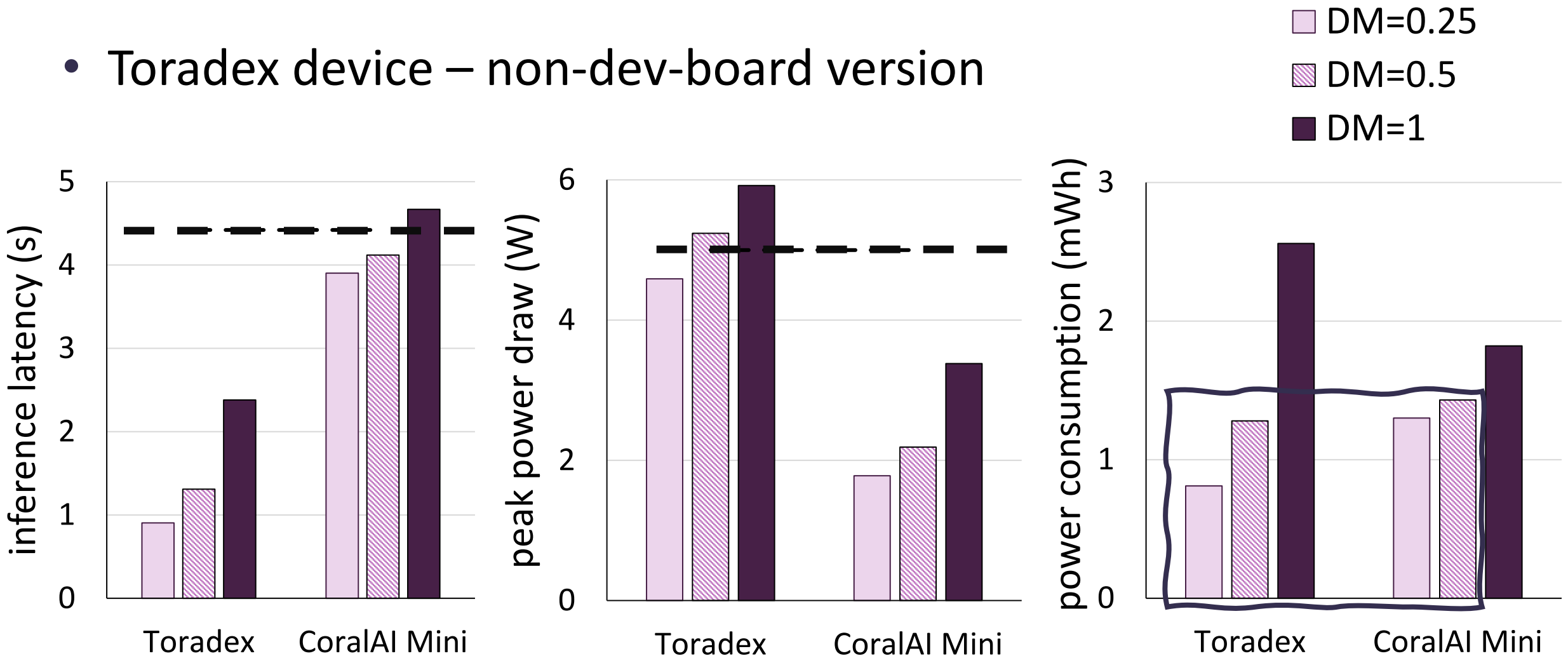
Satellitten indeholder en mikrocomputer, der skal teste kunstig intelligens i rummet. 😤

Læs mere om projektet her 👉 https://www.itu.dk/.../Studerende-opsender-satellit-der...

📷 Julian Priest (CC BY-NC 3.0)

future steps – DISCO 2

- Toradex device – non-dev-board version



Toradex is the most energy-efficient for the small & mid-sized models!
non-dev-board version reduces the peak power.

future steps – DISCO 2

- Toradex device
- denoising / image stacking to improve image quality
- modular, robust, and parametrizable image processing pipeline
- power-aware scheduling
- data drift detection

future steps – data processing @ the edge

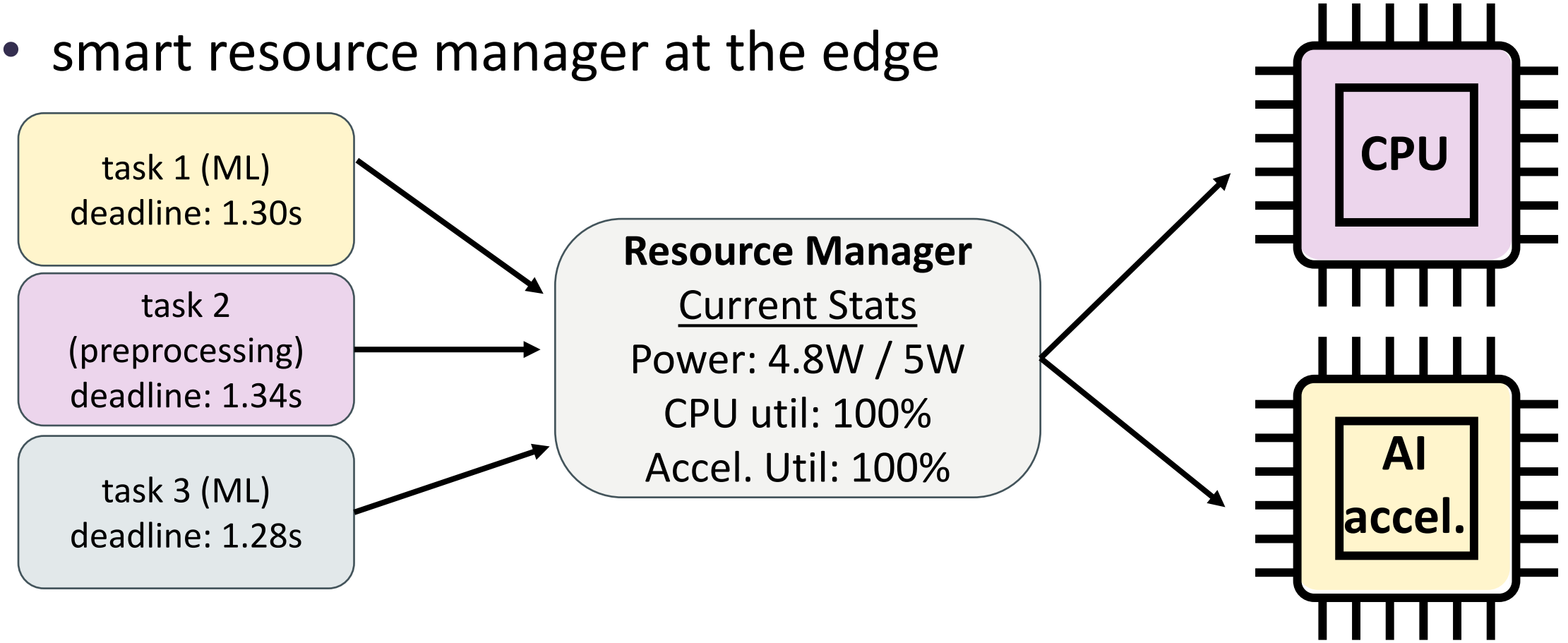
- broader benchmarking

need something that can appeal to different sizes of devices!

future steps – data processing @ the edge

adjust resources assigned to tasks, considering power & latency requirements!

- broader benchmarking
- smart resource manager at the edge



future steps – data processing @ the edge

- broader benchmarking
- smart resource manager at the edge
- model optimizations
  - e.g., quantization, pruning, operator fusion …

reduce model sizes with low impact on accuracy!

data processing @ the edge

- demand for more data analysis closer to the data source
  - reduces data movement & privacy concerns
  - helps with real-time decisions
- variety of edge devices to choose from offering increasingly powerful hardware but still resource-constrained
  - requires not just latency-efficient,
    but also energy-efficient data processing
- hardware specialization helps with latency & power budget
  - though, we need more flexibility

need for methods that can deal with resource management & program updates at the edge!

teamRAD - resource-aware data systems

rad.itu.dk

phd students

Ties Robroek

Ehsan Yousefzadeh-Asl-Miandoab

Robert Bayer

DISCO collaborator

Julian Priest

helpers

Sebastian Büttrich

Lottie Greenwood

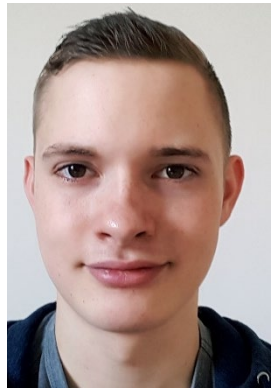
DASYA

Data-Intensive Systems and Applications

www.dasya.dk

@dasyaITU


















ITU MSc students who worked on DISCO

- Ívar Óli Sigurðsson
- Nicolaj Valsted
- Daniel Gjesse Kjellberg
- Jeppe Lindhard
- Nikolaj Sørensen
- Martin Pinholt
- …

data processing @ the edge

thank you!

- demand for more data analysis closer to the data source
  - reduces data movement & privacy concerns
  - helps with real-time decisions
- variety of edge devices to choose from offering increasingly powerful hardware but still resource-constrained
  - requires not just latency-efficient,
    but also energy-efficient data processing
- hardware specialization helps with latency & power budget
  - though, we need more flexibility

need for methods that can deal with resource management & program updates at the edge!