How to Run Audio and Vision AI Algorithms at Ultra-Low Power

Presenter:

Deepak Mital

Sr. Director, Architecture

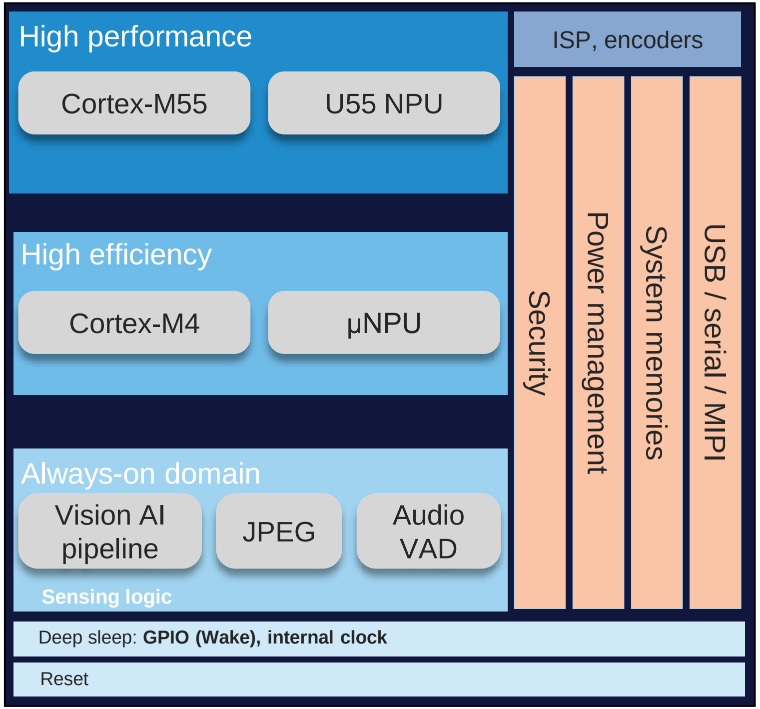
Synaptics Incorporated

Problem statement

- Many IoT applications do not require “continuous maximum” compute
- Continuous monitoring results in battery drain
- Examples:
  - Security camera: Turn on main processing for actual detection only when confirmed necessary
  - Human presence detection (HPD) and identification to turn device on: Run HPD detection and identification algorithm only when detected “potential” presence
  - Predictive maintenance: Enable advanced detection only when initial metrics are met
  - Shoplift prevention: Enable detailed analytics only when “potential” threat detected

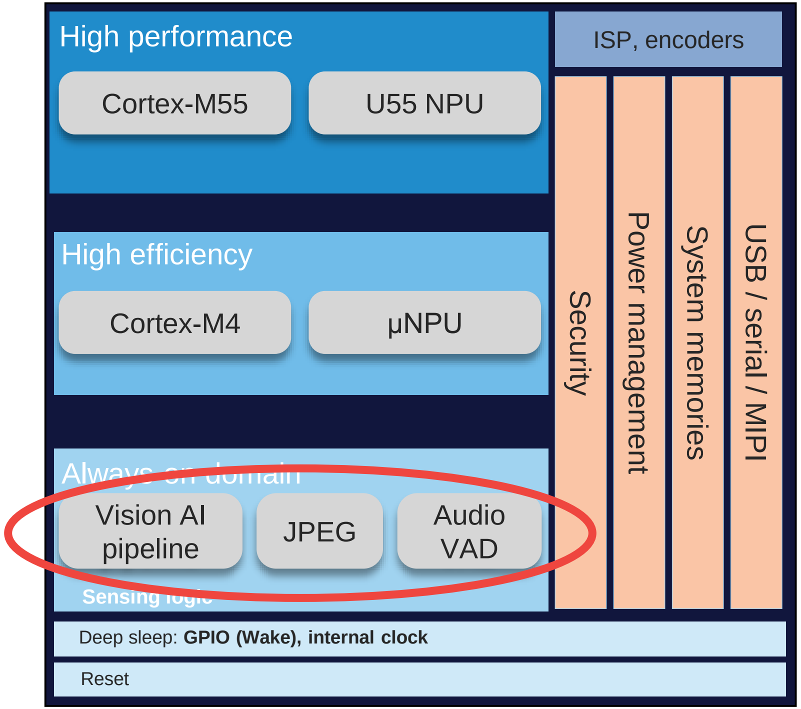
Solution

- Multistage hardware: Capable of running Audio and Video AI algorithms
- Highly efficient AI models with different KPIs for each stage
- Tight orchestration of software to invoke each stage



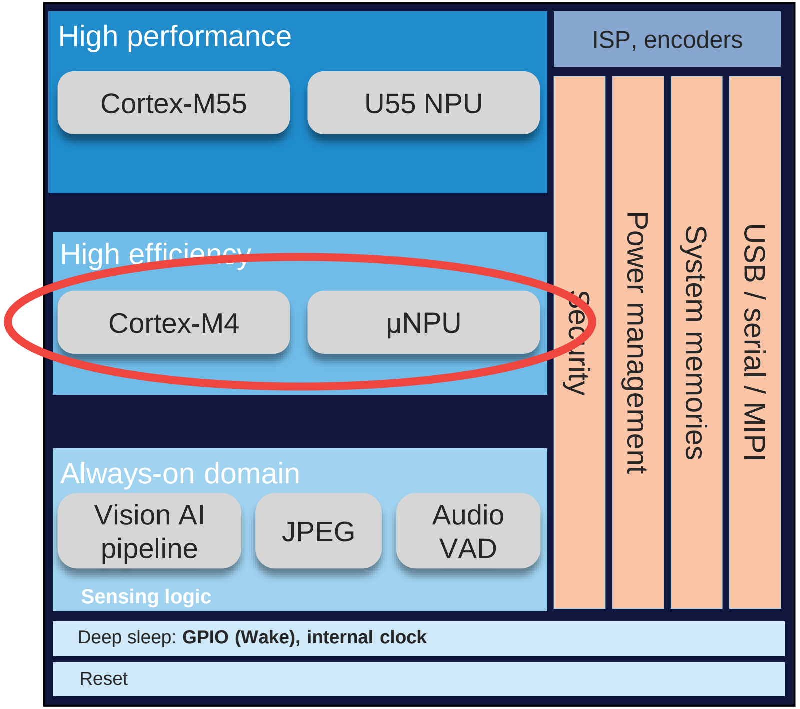
Solution – Stage 1

- Ultra-low power: Microwatts hardware, always on
- Sound detection
- Image change detection
- Critical model requirements are for very few false negatives
  - False negatives will render device unresponsive



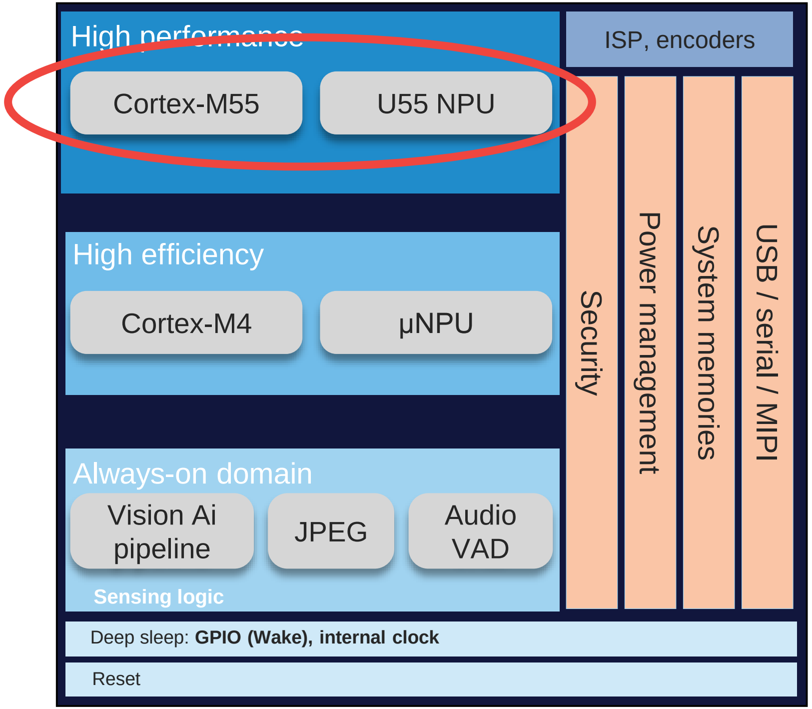
Solution – Stage 2

- Mid- to low power – 10s of microwatts hardware, activated by stage 1 via software
- AI algorithms (example):
  - Wake-word detection
  - Human presence detection
- Critical model requirements are for very few false negatives and false positives
  - False negatives will render device unresponsive
  - False positives will increase power consumption



Solution – Stage 3

- High performance, activated by Stage 2 via software
- AI algorithms (example):
  - Person identification
  - Object detection
- Critical model requirements are for very high performance at low power
  - Slow run times will increase power consumption

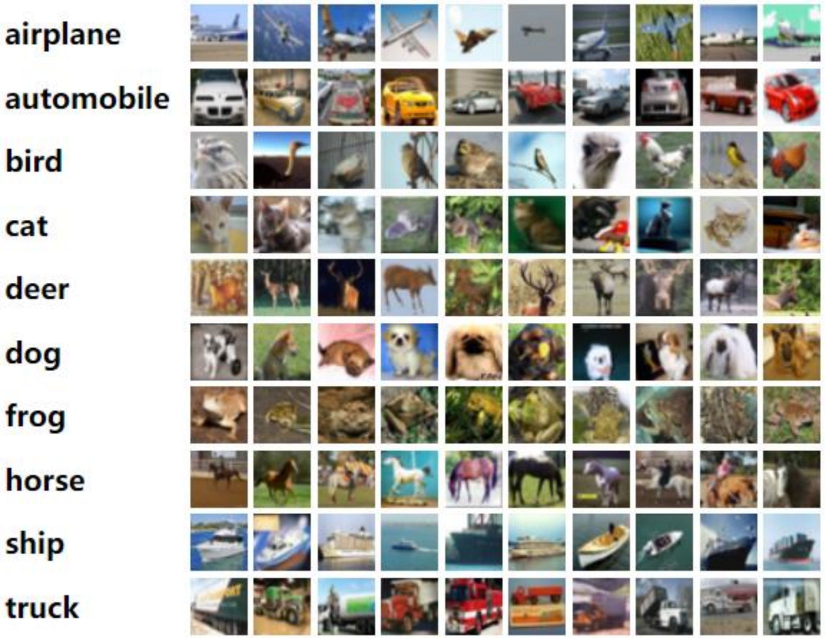


AI models

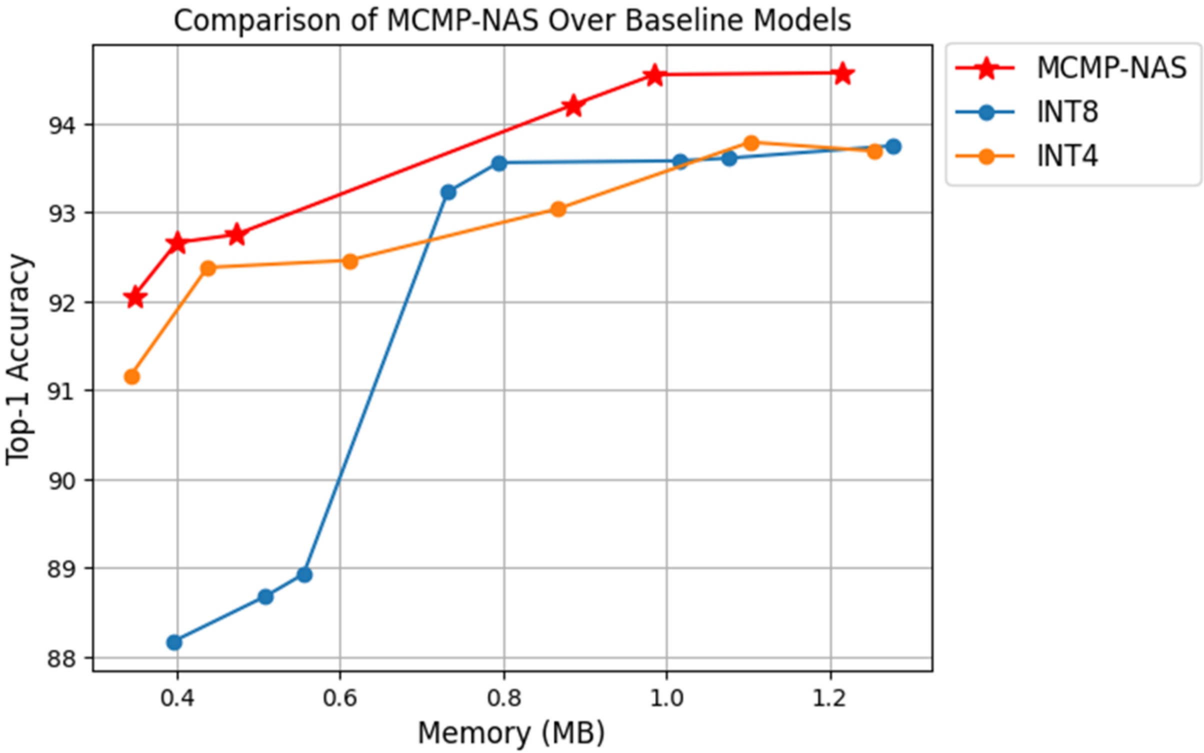
- Different requirements for AI models at each stage
- Need AI models optimized for different KPIs: accuracy, performance, and size
- NAS-based model generation architecture where the models are purpose built for the constrained silicon
- Primary factors affecting inference KPI
  - Model architecture design
  - Model quantization
- Approach: Jointly optimize model architecture and quantization under memory constraints

Multi-precision NAS search range for classification

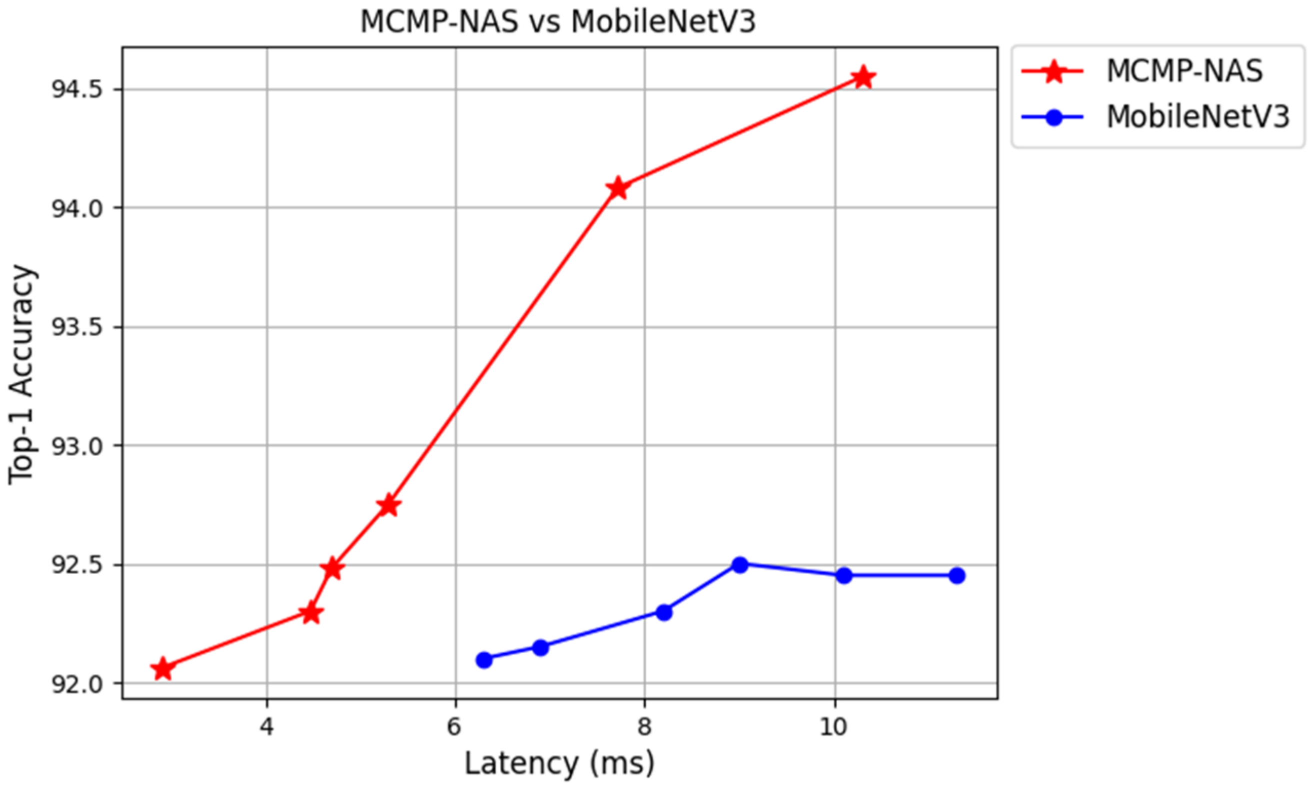
- Resolution – [28x28 – 32x32]
- Kernel size – [3x3, 5x5, 7x7]
- Depth – [2, 3, 4]
- Width (channel expansion factor) – [2, 3, 4]
- **Mixed-precision** quantization parameters – [4 bit, 6 bit, 8 bit]



CIFAR-10 classification – Mixed vs 8- or 4-bit precision



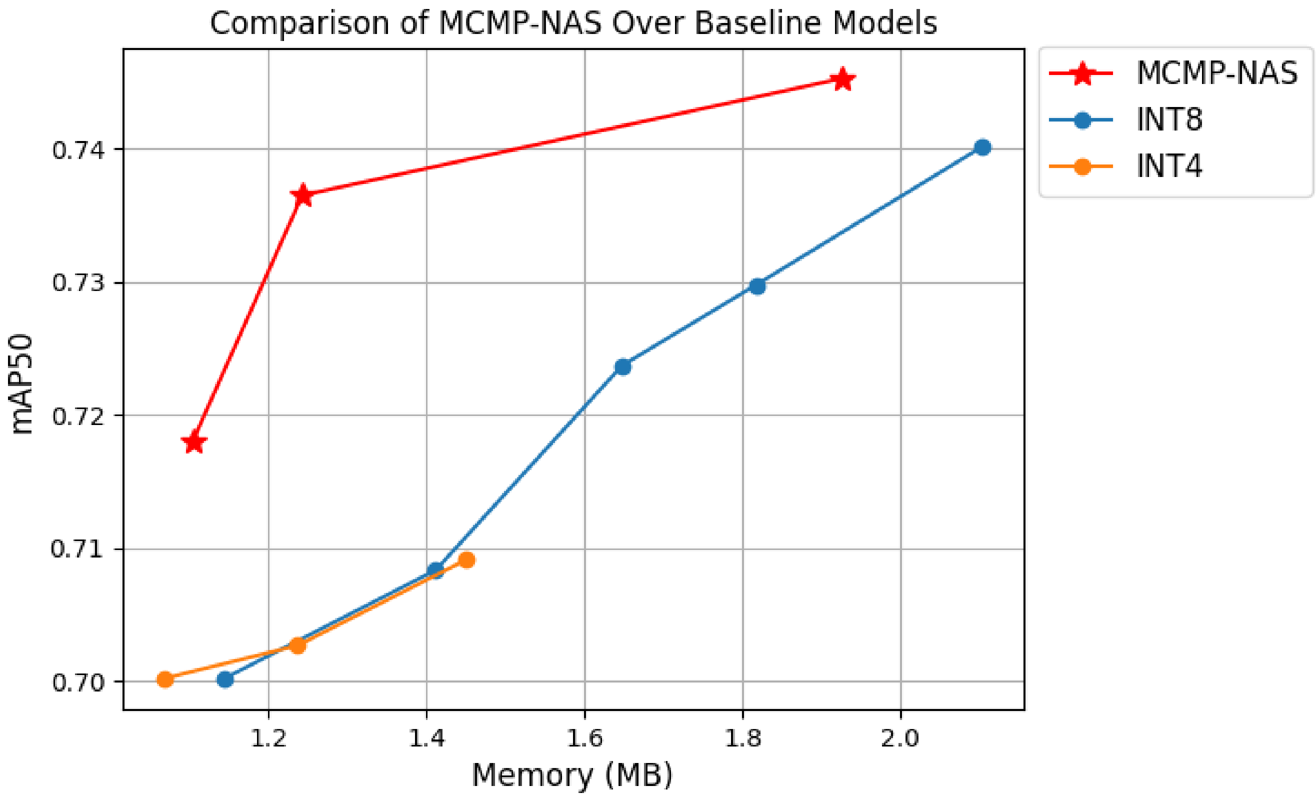
CIFAR-10 classification comparison



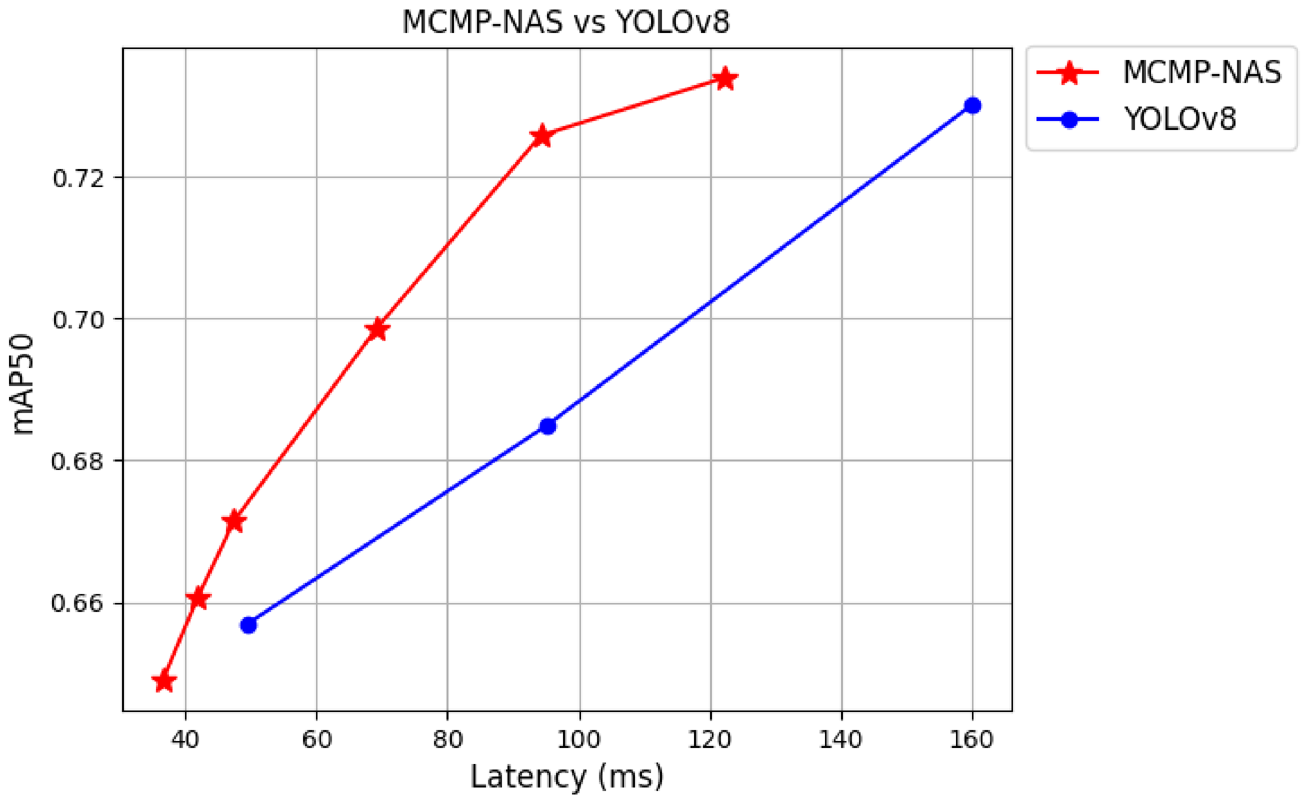
Object detection dataset

- Resolution – [320x240 – 640x480]
- Kernel size – [3x3, 5x5, 7x7]
- Depth – [2, 3, 4]
- Width (channel expansion factor) – [2, 3, 4]
- **Mixed-precision** quantization parameters – [4 bit, 6 bit, 8 bit]

COCO person detection – Mixed vs 8- or 4-bit precision



COCO person detection comparison



Segmentation run on Stage 3

- Model development stage KPI:
  - COCO Instance Mask mAP: 0.636
  - Latency: 92.19 ms
  - Resolution: 480x640 (VGA)
  - Weights: 1.57 M parameters
- Model run on hardware:
  - Inference time: 96 ms
  - Total frame time: 120 ms

Summary

- Building full applications running at ultra-low power requires high levels of integration of hardware and software
- Multiple levels of processing is needed to wake up silicon components as needed
  - Stage 2 and Stage 3 come out of deep sleep based on results from previous stage
- The low-power orchestration demands tight software integration
- Each stage requires AI models with different KPIs on accuracy, model size, and speed
  - Need to have NAS-based model generation/training software to enable the complete solution
- Solution enables battery-powered devices that are AI capable and can run for many months/years

Resources

Synaptics Astra embedded processors
https://www.synaptics.com/products/embedded-processors

Synaptics Astra evaluation Kit
https://synacsm.atlassian.net/servicedesk/customer/portal/543/group/563/create/6387

Synaptics Astra software
https://github.com/synaptics-astra