AI@EDGE

Applying Artificial Intelligence on Edge devices using Deep Learning with Embedded optimizations

VLAIO TETRA HBC.2019.2641

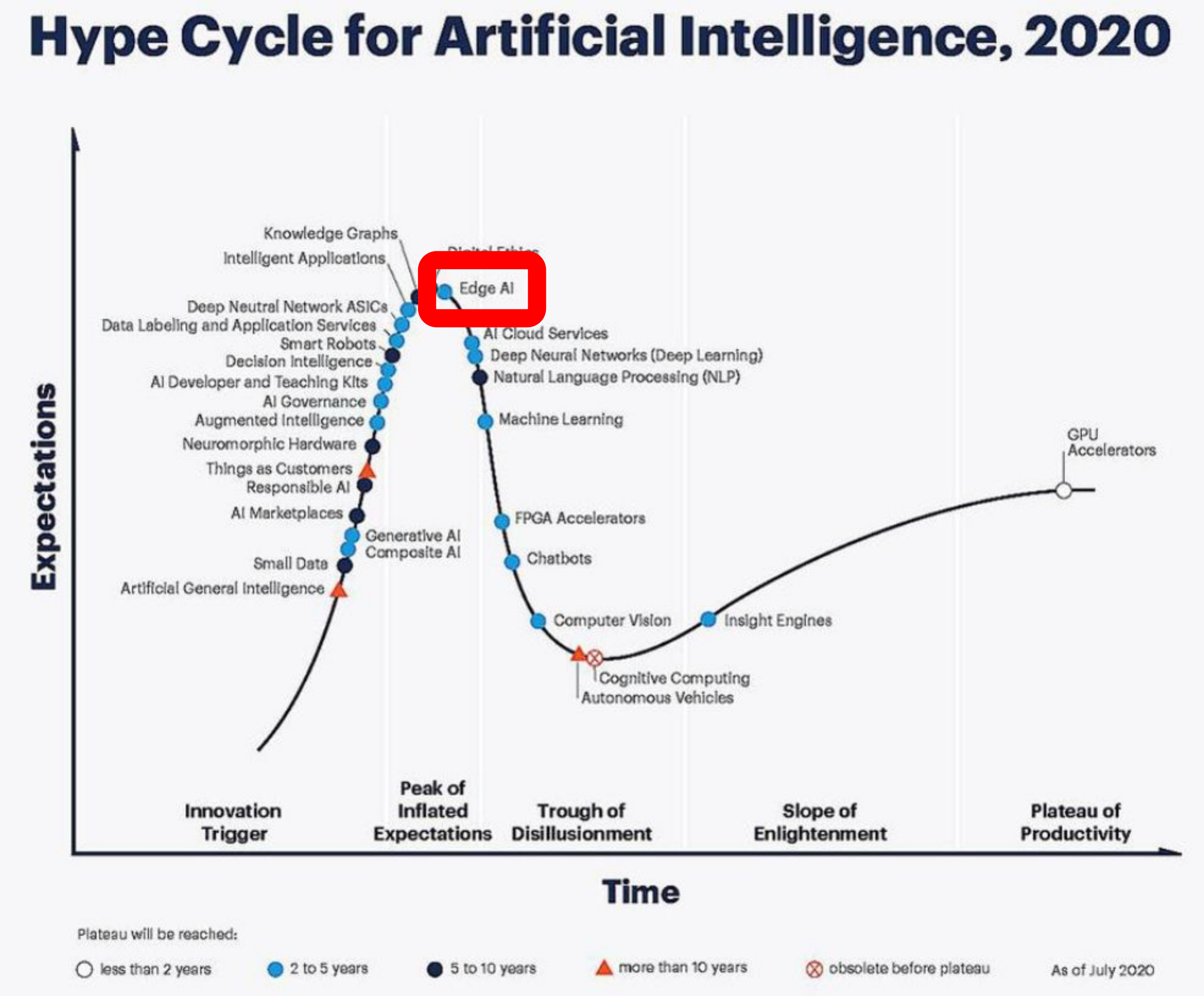
User group meeting 2 28-01-2020

ai-edge.be

iot-incubator.be www.eavise.be

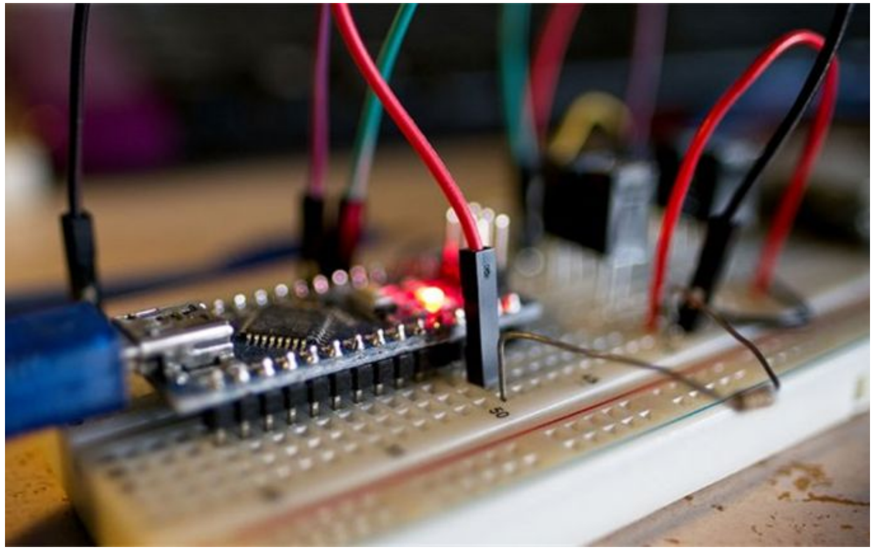
Agenda

1. Introduction
2. Use cases by the user group
3. Platforms
4. Current work
5. Workshop
6. Networking


Hype Cycle for Artificial Intelligence, 2020
Expectations
Knowledge Graphs
Intelligent Applications
Edge AI
Deep Neutral Network ASICs
Data Labeling and Application Services
Smart Robots
Decision Intelligence
AI Developer and Teaching Kits
AI Governance
Augmented Intelligence
Neuromorphic Hardware
Things as Customers
Responsible AI
AI Marketplaces
Generative AI
Composite AI
Small Data
Artificial General Intelligence
AI Cloud Services
Deep Neural Networks (Deep Learning)
Natural Language Processing (NLP)
Machine Learning
FPGA Accelerators
Chatbots
Computer Vision
Cognitive Computing
Autonomous Vehicles
Insight Engines
GPU Accelerators
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
Time
Plateau will be reached:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau
As of July 2020

Gartner

AI@EDGE in a nutshell



AI ->

Deep Learning Inference

Trained model -> Target system

EDGE ->

Low-cost small embedded systems

Microcontrollers & System processors

AI@EDGE overview

WP1: Exploration (3 mm)

- WP 1.1: study of frameworks for low-cost embedded systems
- WP 1.2: study of optimisation techniques for Deep Learning on embedded systems
- WP 1.3: query @ user group

WP2: Proof of concept (6 mm)

- WP 2.1: selection hardware & frameworks
- WP 2.2: collect & annotate data
- WP 2.3: implementation
- WP 2.4: test & validate

WP3: Industrial Case studies (18 mm)

- WP 3.1: gather functional & non-functional requirements
- WP 3.2: select & operationalize hardware and framework
- WP 3.3: implementation
- WP 3.4: optimisation
- WP 3.5: test & validate

WP4: Valorisation (9 mm)

- WP 4.1: overview of hardware & frameworks on website
- WP 4.2: manual with best-practices
- WP 4.3: hands-on workshop
- WP 4.4: scientific publications
- WP 4.5: final symposium

Project planning

- New project member (VIVES): Jonas Lannoo started 01/10/2020
- Project has been extended with 3 months (standard COVID extension) -> end date 31/05/2022
- Milestones have been postponed
  - M1 Proof of concept application: Q2 2021
  - M4 Workshop: Summer 2021
  - M2 Industrial use cases: Q1 2022
  - M3 Manual of best practices: Q1 2022
  - M5 Publications: end of project

User group use cases

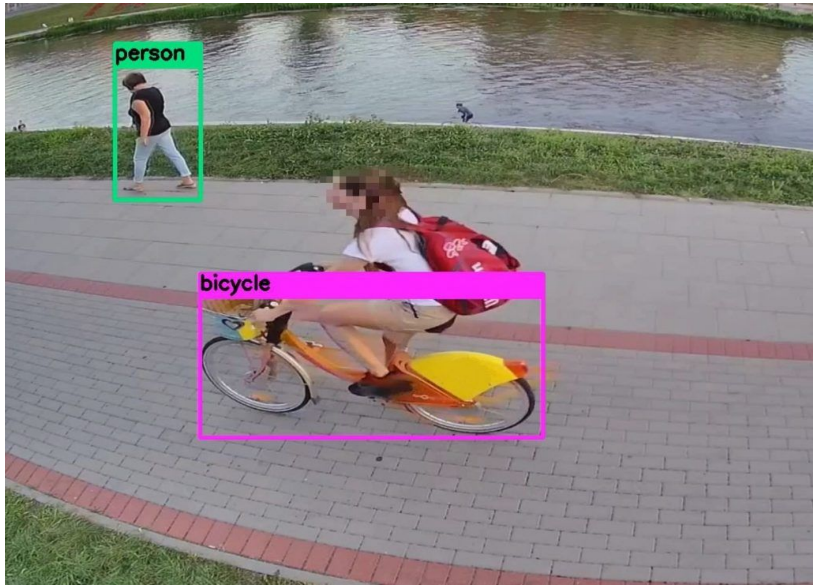
- 1st round of meetings
  - Melexis
  - Digipolis
  - TML
  - Scioteq
  - Picanol
  - E.D.&A.
  - Sensotec
- 2nd round of meetings
  - 6WOLVES
  - Edgise
  - QMineral
  - DP Technics

Melexis

- Use case: Low resolution thermal image sensor (32x24 pixels)
- Dataset: small dataset recorded by Melexis, semi automatic annotation.
- R-CNN baseline model trained on dataset
- Goal: fit neural network on a microcontroller <=8€
- See: current work

Digipolis

- Use case: Traffic camera analysis
- Dataset: Privacy issues, standard data sets are not suited, angle of traffic cameras
- Goal: Recognise cyclists,
  pedestrians, busses



TML

- Use case: Telraam camera image analysis
- Dataset: Available and annotated
- Goal:
    - Detect pedestrians, (motor)cyclists, cars, heavy vehicles
    - Detect their speed & direction
    - Cost constraint
    - Power constraint

ScioTeq

- Use case 1: Aircraft 6D pose estimation
- Dataset: un-annotated camera frames + 6D pose
- Goal:
  - Determine plane 6D pose from runway images

- Use case 2: Flight display error detection
- Dataset: un-annotated display frames
- Goal:
  - Detect errors in displayed numbers

Picanol

- No specific use case
- Use hardware for industrial environment (+50 ˚C)
- Interests:
  - Measurement and analysis using DL of:
    - Oil temperature
    - Control box environment
    - Humidity
    - Process timings
    - Pressure
    - Wire detection (specific for weaving)
  - Using vision

E.D.&A.

- Use case: Capacitive touch sensor induction furnace
  - Effectiveness under stress (moisture, dirt, EMI …)
- Dataset: Available from master thesis
- Goal:
  - Reproduce using NN
  - Better performance
  - STM Devkit
  - < 8 ms inference time

Sensotec

- Use case: Wordprediction & correction in the browser
- Dataset: Available
- Goal:
  - Continuation of master thesis
  - Improving system using NN
  - Generalisable for other languages?
  - Recognition of written letters
  - Target: Chromebook

Proposal

- Target microcontroller:
  - Case 1: Thermal image sensor
  - Case 2: Capacitive touch
- Target single board computer:
  - Case 3: Traffic Analysis
- Target browser:
  - Case 4: Word prediction

- 2 additional use cases to be determined later in the project

User group interaction

Questions / remarks?

Other use cases / data sets?

On which non-functional requirements should we focus?
(Accuracy? Size? Power consumption? Other?)

Platforms

Hardware

- STMicroelectronics (STM32)
- Arduino
- Kendryte
- Raspberry Pi (& industrial variants)
- Nvidia Jetson Nano

Software

- Tensorflow Lite
- Edge impulse
- TFLite on Mbed

Hardware platforms

STM Development boards with

- Cortex M0+, M3, M4, M7 (32-216MHz)

Arduino Development board

- Nano Sense BLE 33 (Cortex M4, 64MHz, 1MB Flash)

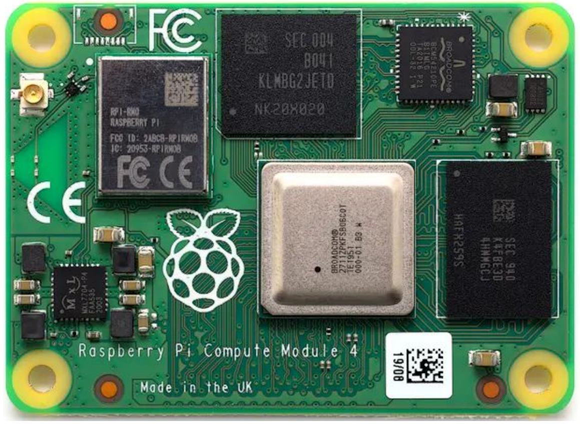
Kendryte K210

- RISC-V Dual Core 64-bit, hardware accelerator

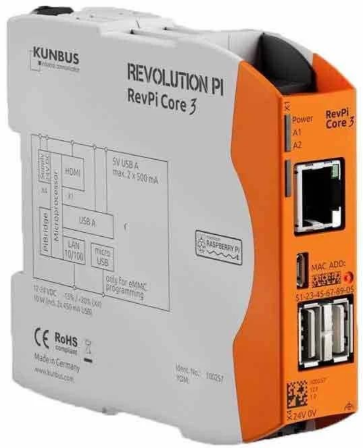
Raspberry Pi 4 SBC

- Quad core Cortex-A72, 1.5GHz, 1-2-4-8GB RAM

Hardware platforms



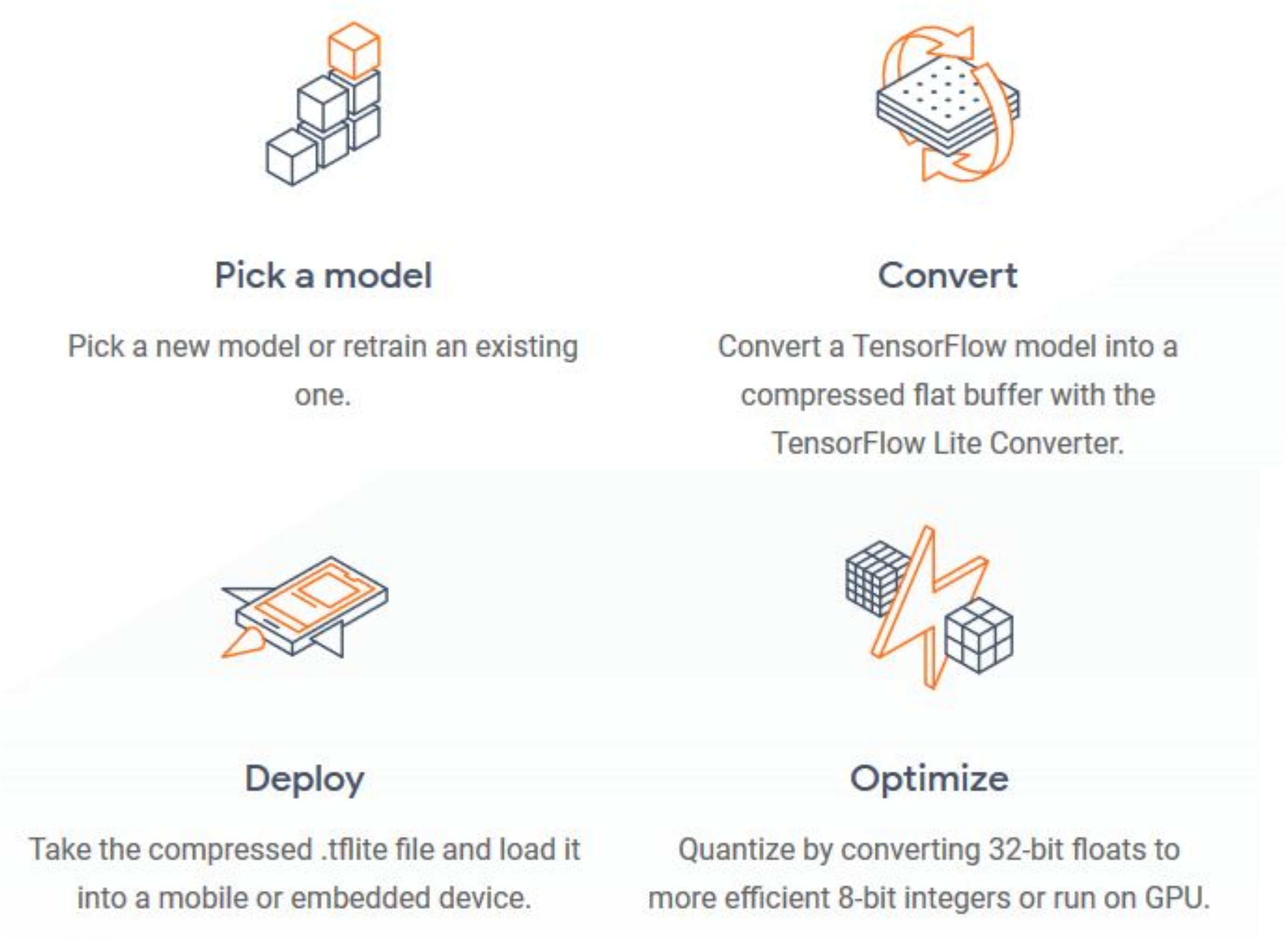
Raspberry Pi 4 Compute module
- Same specs as RPi 4, with industrial support



Revolution Pi 3+ from KUNBUS
- Industrial variant

Nvidia Jetson Nano
- Quad-core ARM A57, 1.43GHz,
  4GB RAM + 128 Core GPU

TensorFlow Lite



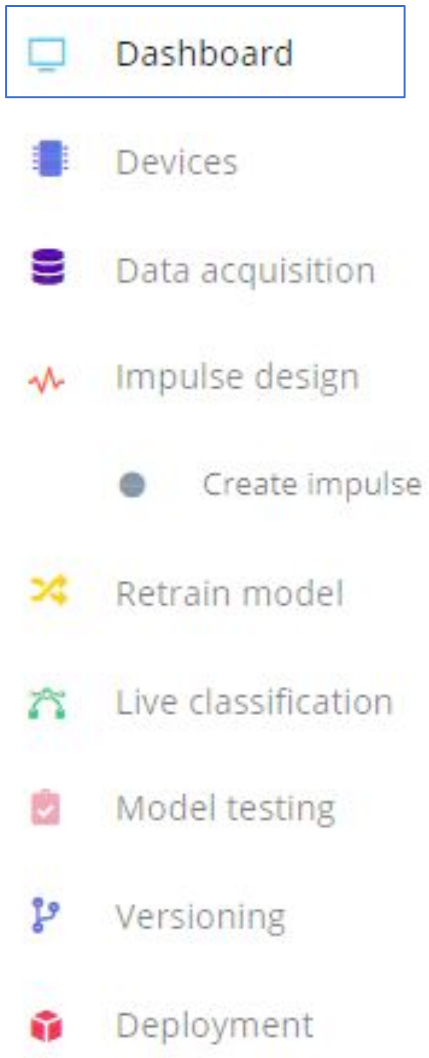
How it works

Used in:

- Edge Impulse
- TFLite for Mbed

Platforms - Edge Impulse

- Online embedded machine learning tool
- TFLite (micro) as backbone
- Project based, versioning and sharing available
- Large community for support
- Extensive documentation available online



Platforms - Edge Impulse

EDGE IMPULSE

Collecting data:

Device integration

Connect a fully supported development board

The best way to get started with real hardware from Nordic, Arduino, OpenMV, ST, Eta Compute and Himax - fully supported by Edge Impulse.

Use your mobile phone

Use your mobile phone to capture movement, audio or images, and even run your trained model locally. No app required.

Data from any device with the data forwarder

Capture data from any device or development board over a serial connection, in 10 lines of code.

Upload data

Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, CBOR or JSON format.

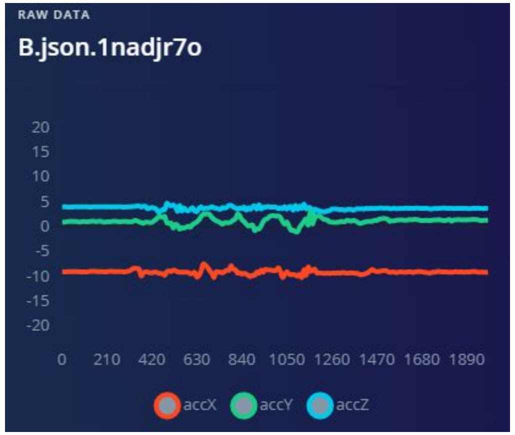
Integrate with your cloud

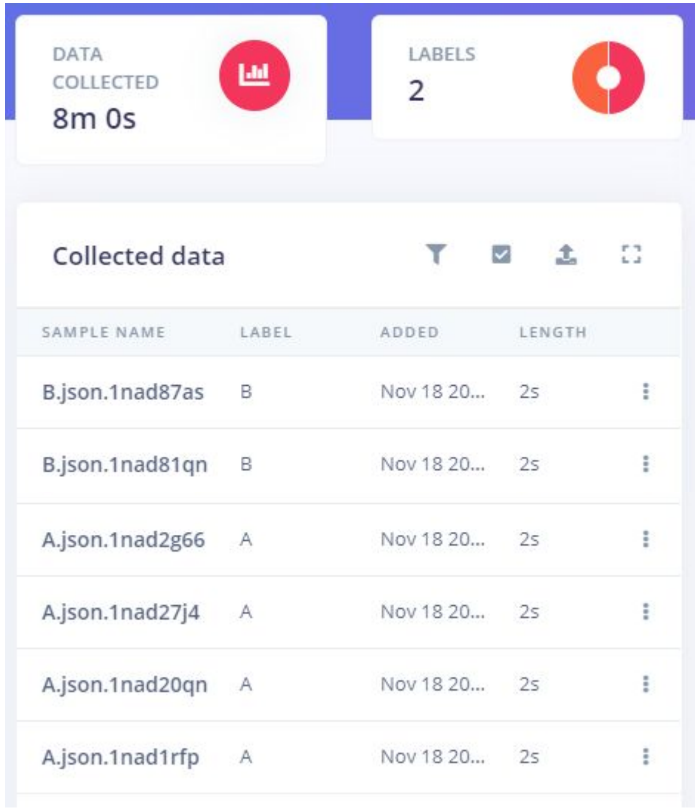
The enterprise version of Edge Impulse integrates directly with the data stored in your cloud platform.

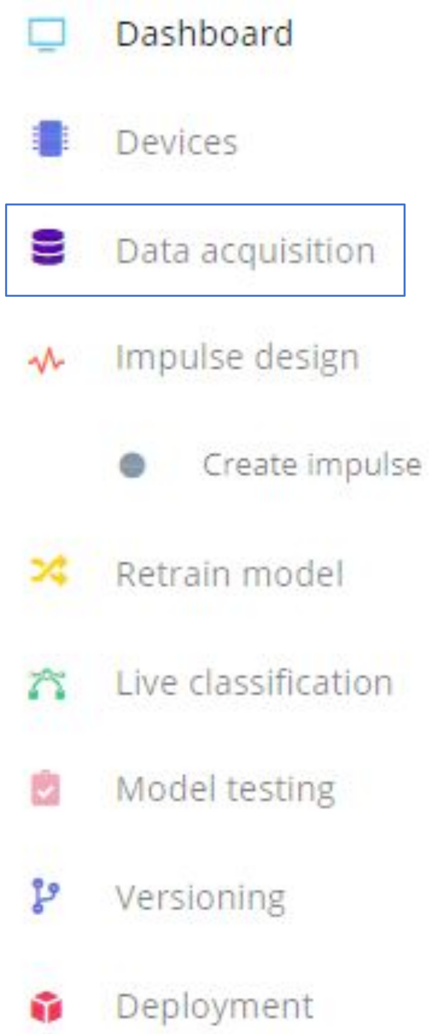
- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

Platforms - Edge Impulse

- Data acquisition
- Preview data
- Split training/test

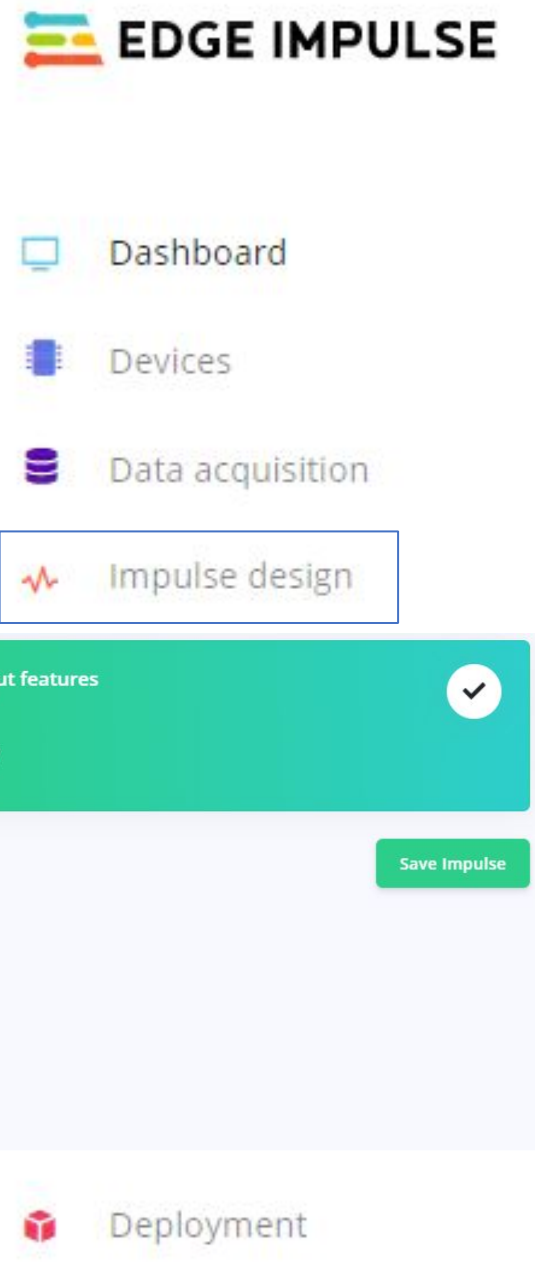


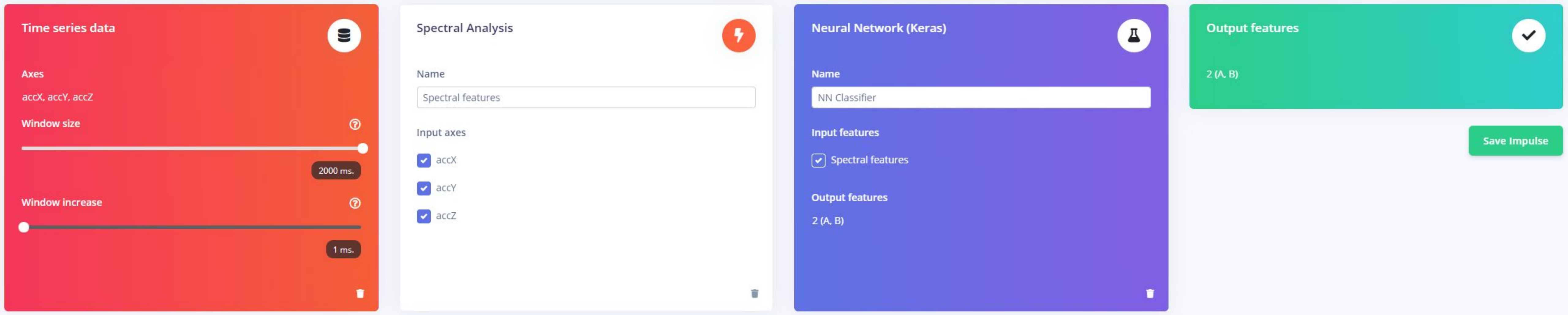




Platforms - Edge Impulse

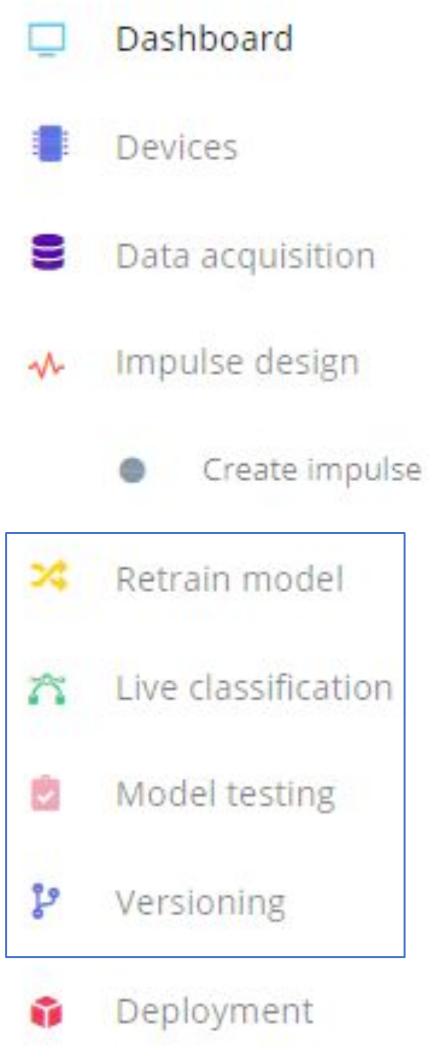
Creating the network (impulse design)





Platforms - Edge Impulse

- Retraining model after changes
- Live classification using your devices (API)
- Check model with test-data
- Store configurations using versioning



Platforms - Edge Impulse

EDGE IMPULSE

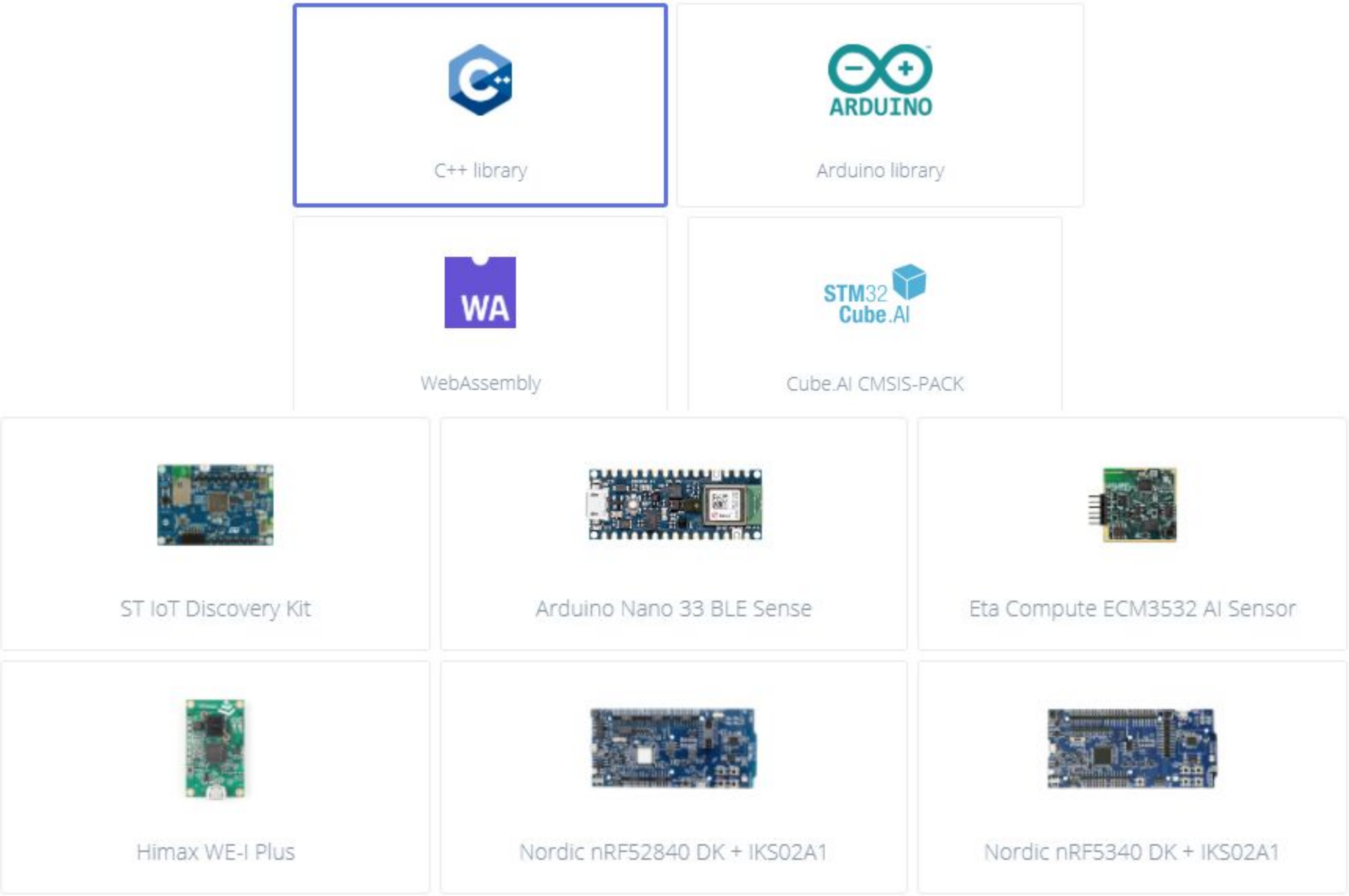
Deployment

- To library
- As binary for device

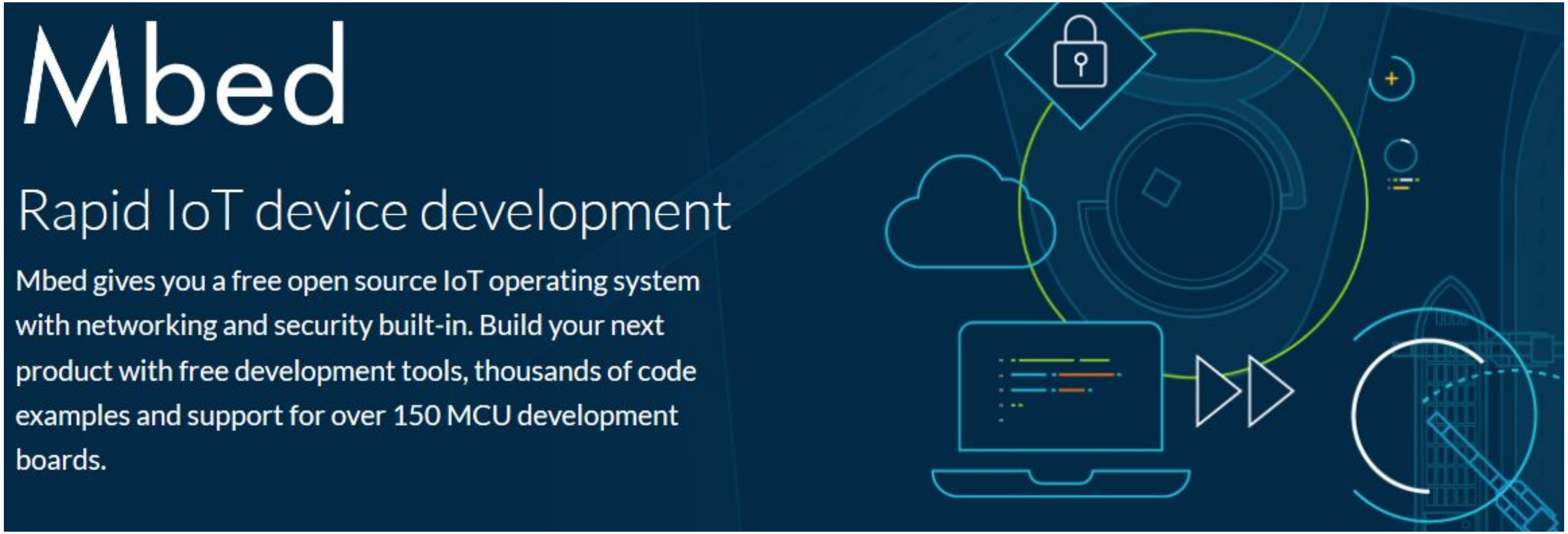
+ Optimisation

  Quantizing model

Quantized (int8) ⭐ Currently selected This optimization is recommended for best performance.	RAM USAGE 1.5K ROM USAGE 15.3K	LATENCY 1 ms ACCURACY 16.97%
Unoptimized (float32) Click to select	RAM USAGE 1.5K ROM USAGE 17.6K	LATENCY 1 ms ACCURACY 16.4%



Mbed platform/ecosystem



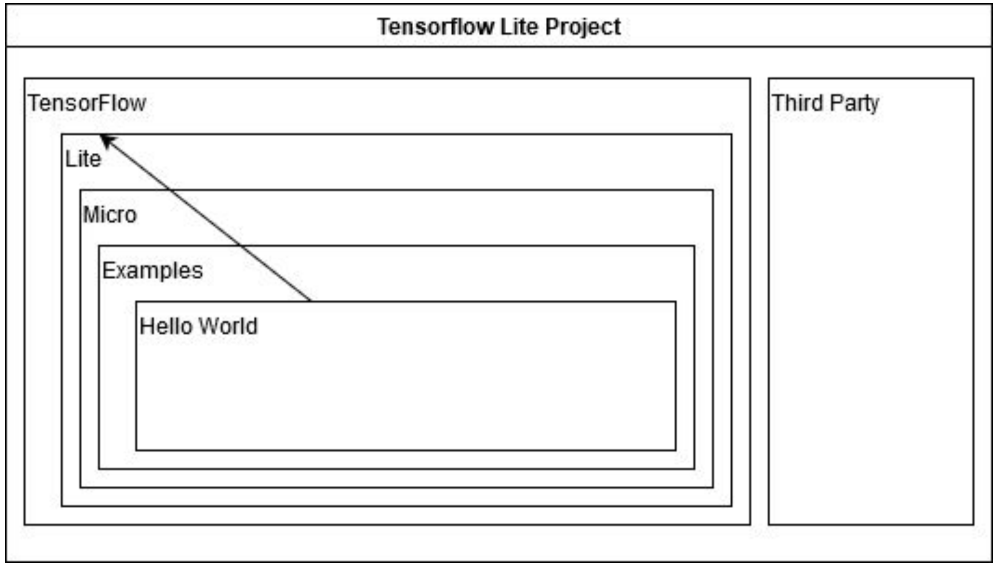
Mbed OS: RTOS or baremetal

Mbed compiler + tools (IDE, CLI,...)

TensorFlow lite micro for Mbed ecosystem

TensorFlow generator tool using make

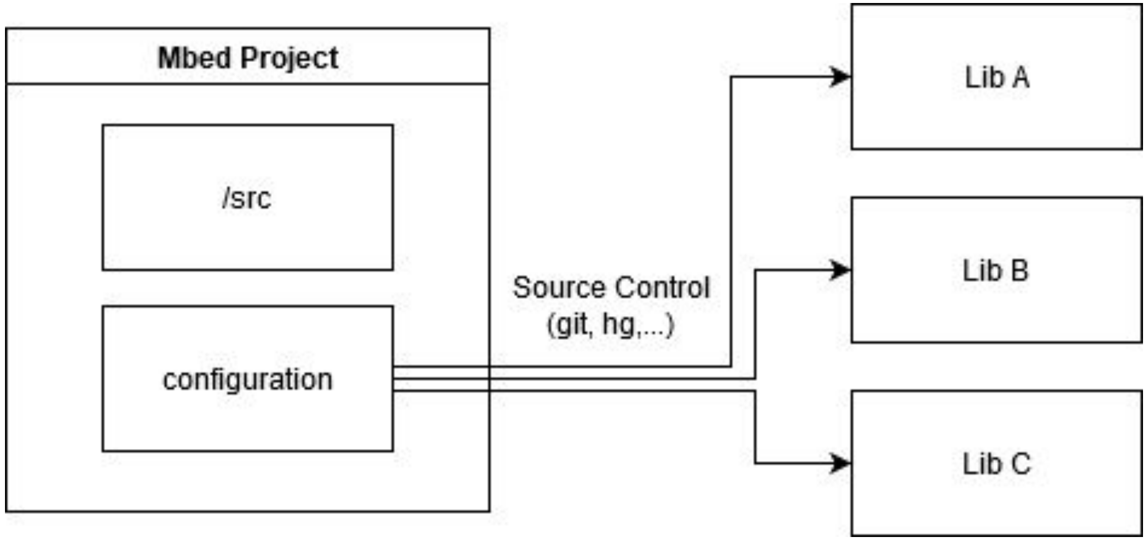
- Inside out project structure
- Applications lives inside the TensorFlow project
- Hard to update or extend
- Hard to implement in existing projects
- Enforces to use Google TensorFlow design style/rules



TensorFlow lite micro for Mbed ecosystem

Typical Mbed project structure:

- Project source in /src directory
- Dependencies are managed in .lib files
  - Only contain source control origin + explicit version (eg GitHub)
- Lightweight projects
- Easy to update



TensorFlow lite micro for Mbed ecosystem

TensorFlow Lite Micro as Library (for mbed)

Easy integration (Mbed add command)

Easy updates (Mbed update command)

https://github.com/sillevl/tensorflow-lite-micro-mbed

- TensorFlow generated project
- Excluded application specific files
- Fix #include paths

Example: https://github.com/sillevl/tensorflow-lite-micro-hello-world-mbed
Hello World application for mbed using TensorFlow Lite as library

TensorFlow Lite Docker Helper

TensorFlow is developed in the Linux ecosystem

Hard to use in a Windows environment

--> Docker container helper to generate projects

- Docker container containing:
  - TensorFlow project
  - Linux build tools
  - mbed build tools

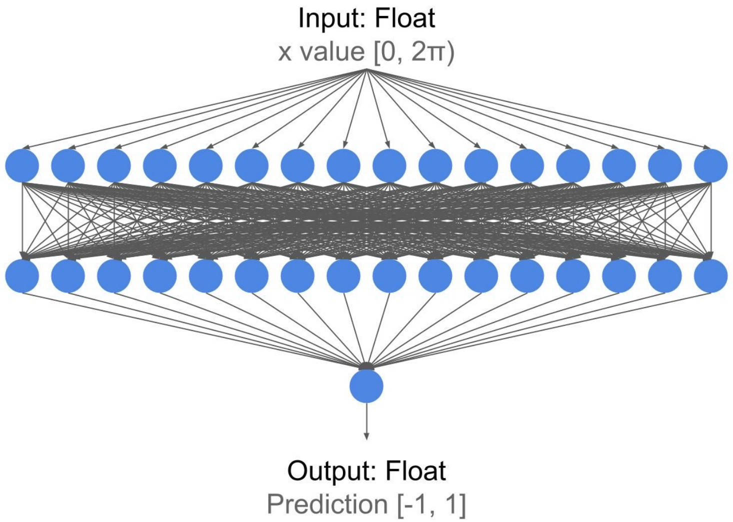
Generate new projects on windows

https://github.com/sillevl/tensorflow-lite-micro-docker-mbed-helper

Benchmark

Tensorflow Lite Micro Hello World example

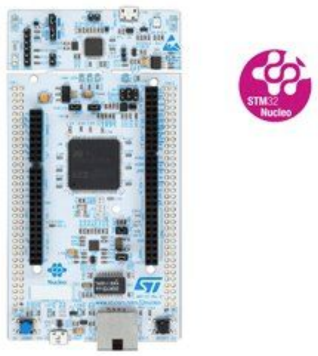
- Model that replicates a sine function
- Absolute basics example
- 3-layer, fully connected neural network with a single, floating point input and a single, floating point output



Benchmark targets

mbed-os (v6.6.0) with mbed-cli
GCC (v9.3.1)
1000 iterations

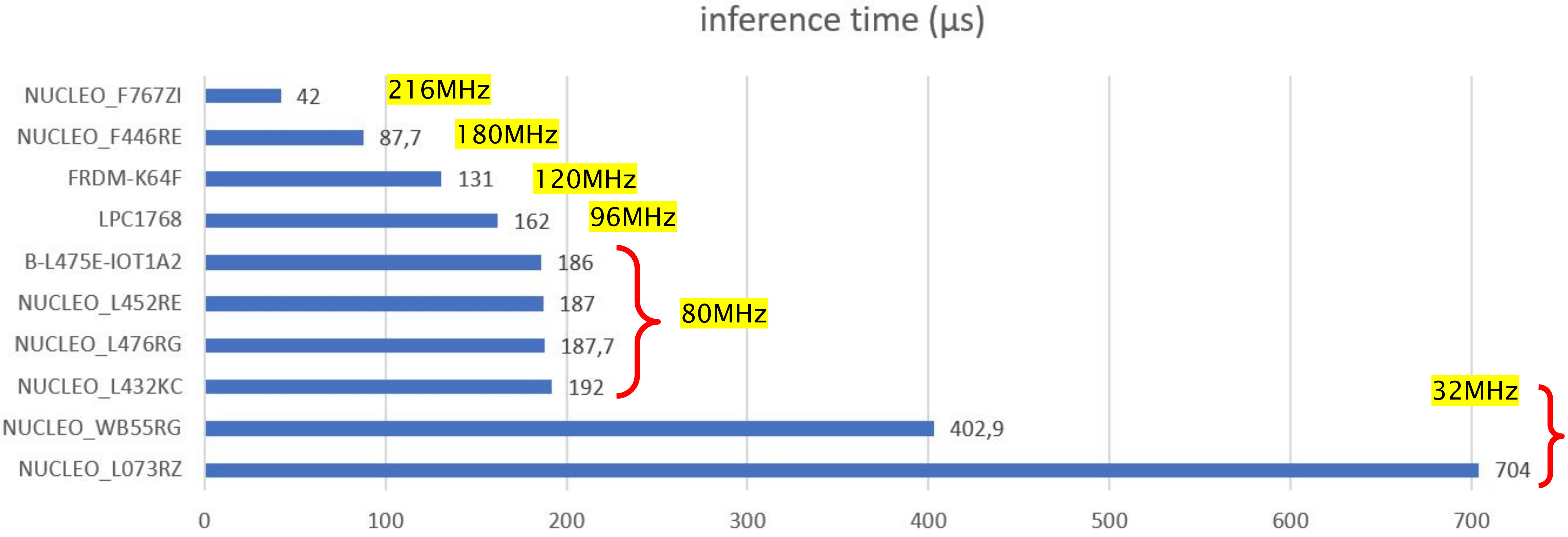
- Cortex-M0+
  - STM32L073RZ @ 32MHz
- Cortex-M3
  - LPC1768 @ 96Mhz
- Cortex-M4
  - STM32F446RE @ 180Mhz
  - STM32L476RG, STM32L432KC, STM32L452RE, STM32L4S5VI @ 80 Mhz
  - K64F @ 120Mhz
- Cortex-M7
  - STM32F767ZI @ 216Mhz



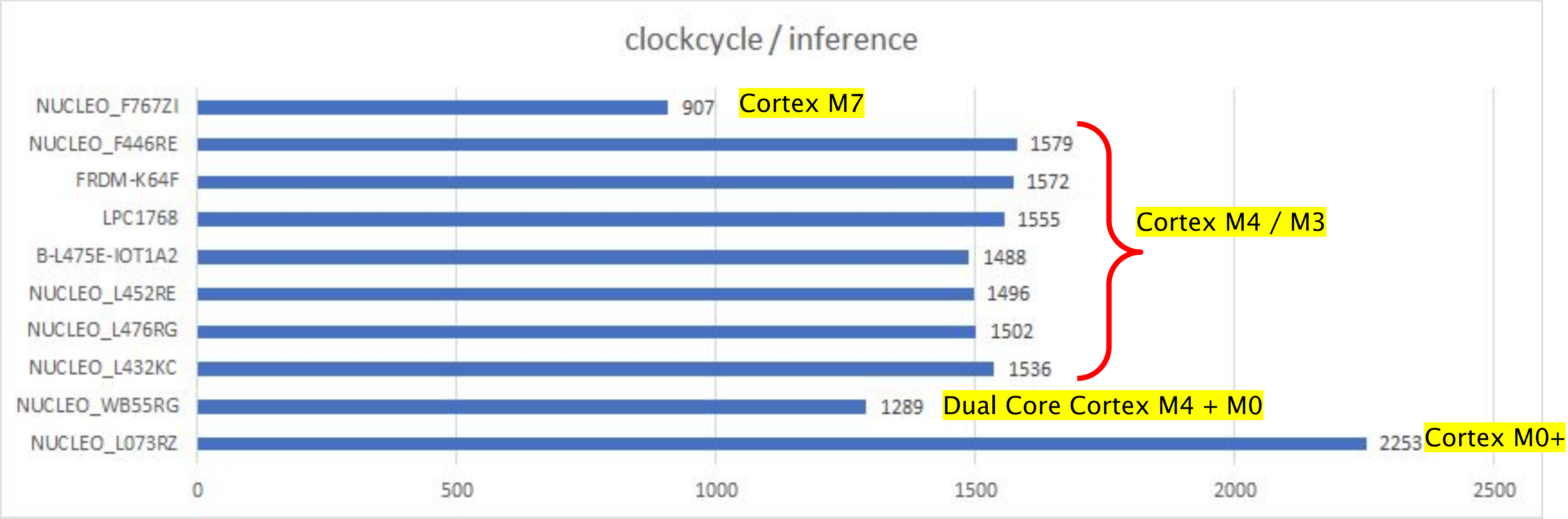
Benchmark results

Board	inference time (µs)	Manufacturer	CPU	Family	CPU clock (MHz)	inference frequency (Hz)	inference frequency per Mhz CPU clock	clock cycles per inference
NUCLEO_L073RZ	704	ST	STM32L073RZ	M0+	32	1420	44,4	2253
NUCLEO_WB55RG	402,9	ST	STM32WB55RG	M4	32	2482	77,6	1289
NUCLEO_L432KC	192	ST	STM32L432KC	M4	80	5208	65,1	1536
NUCLEO_L476RG	187,7	ST	STM32L476RG	M4	80	5328	66,6	1502
NUCLEO_L452RE	187	ST	STM32L452RE	M4	80	5348	66,8	1496
B-L475E-IOT1A2	186	ST	STM32L4S5VI	M4	80	5376	67,2	1488
LPC1768	162	NXP	LPC1768	M3	96	6173	64,3	1555
FRDM-K64F	131	NXP	MK64F	M4	120	7634	63,6	1572
NUCLEO_F446RE	87,7	ST	STM32F446RE	M4	180	11403	63,3	1579
NUCLEO_F767ZI	42	ST	STM32F767ZI	M7	216	23810	110,2	907

Benchmark inference time



Benchmark cpu speed



Benchmark conclusions

- Exact same codebase (model, implementation, compiler, toolchain)
- Inference time scales inversely proportional with CPU clock speed
- ARM family has a significant impact on inference time

User group interaction

Questions / remarks?

Which platforms (hardware / software) are relevant for you?

Do you need more detailed information on the benchmark?

Hardware accelerators (e.g. Kendryte) and FPGA’s are out of scope of this project. Would it be interesting to take these platforms in consideration?

Current work

- Industrial use cases
  - Thermal Image Sensor
  - Traffic Analysis

- Proof of concept use cases
  - Seat detection
  - A/B writing

- Workshop

Melexis use case

Thermal Image sensor

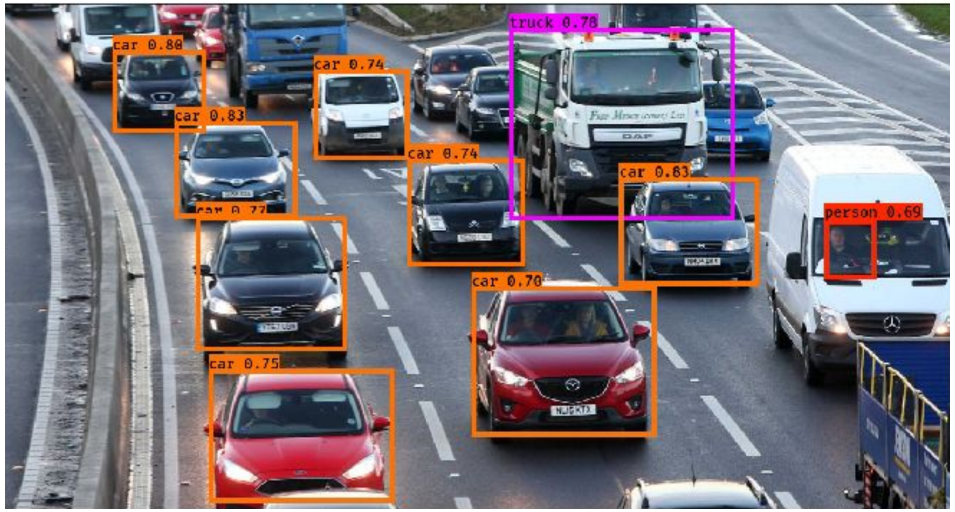
Object Detection on Raspberry Pi

Many object detectors already exist:

YOLO (TinyYOLO), SSD (MobileNet backbone),...

But… very slow on CPU-only device

TensorFlow Lite to the rescue!
Ideal for “small” devices



Results on Raspberry Pi

Many models available online:

- SSD + MobileNetv1
- Quantized and trained on MS COCO

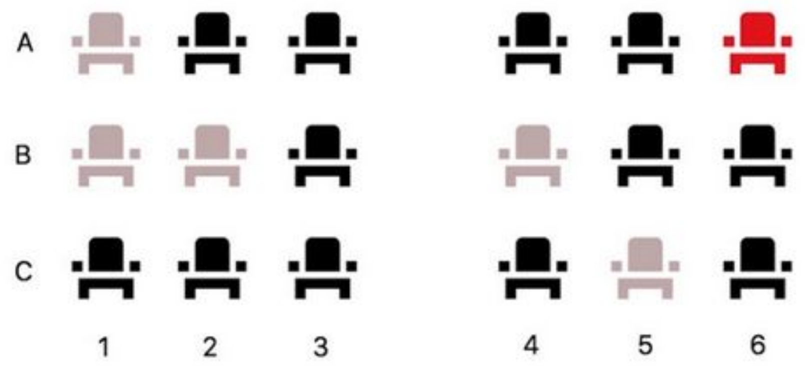
Results on Raspberry Pi :

RPI 3B+ (1GB RAM) 0:38s per frame

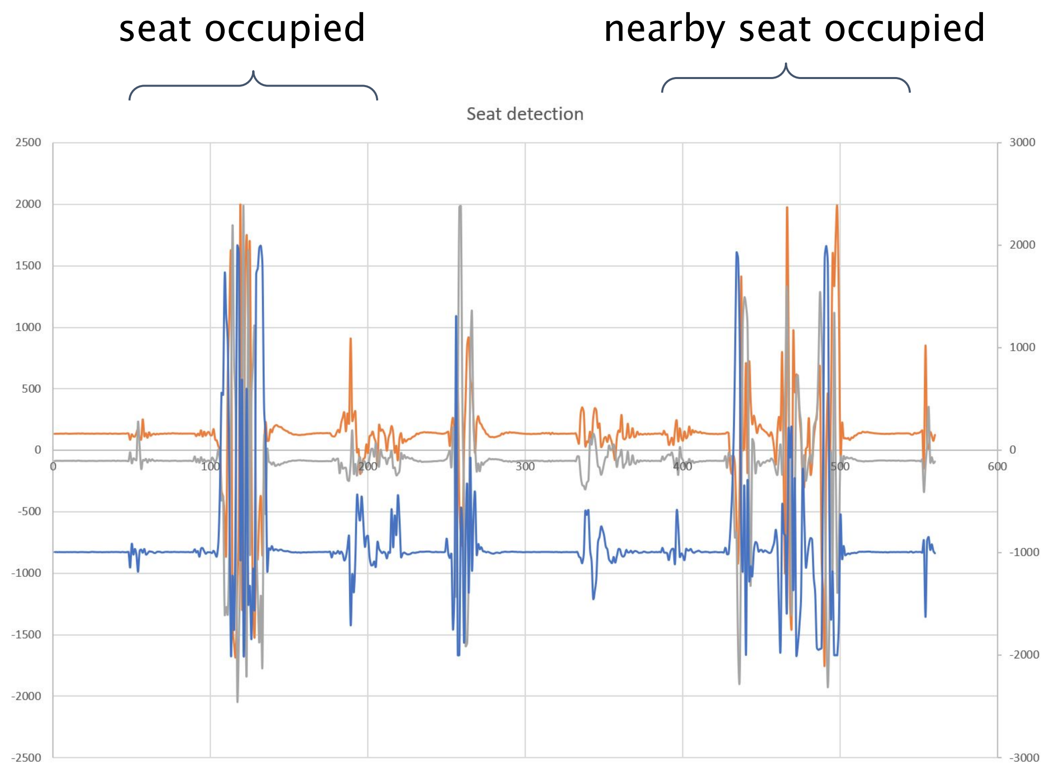
RPI 4 (4GB RAM) 0:19s per frame

multiple seconds without TF Lite

Seat detection case



- Seat detection
- Count number of people in a room
- Prevent false positive (eg, cleaning personnel will move all seats)
- Large inference from nearby movements
- Accelerometer, gyroscope, magnetometer?

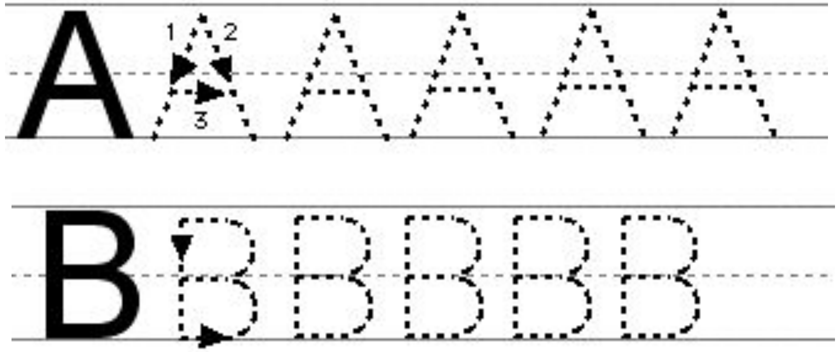


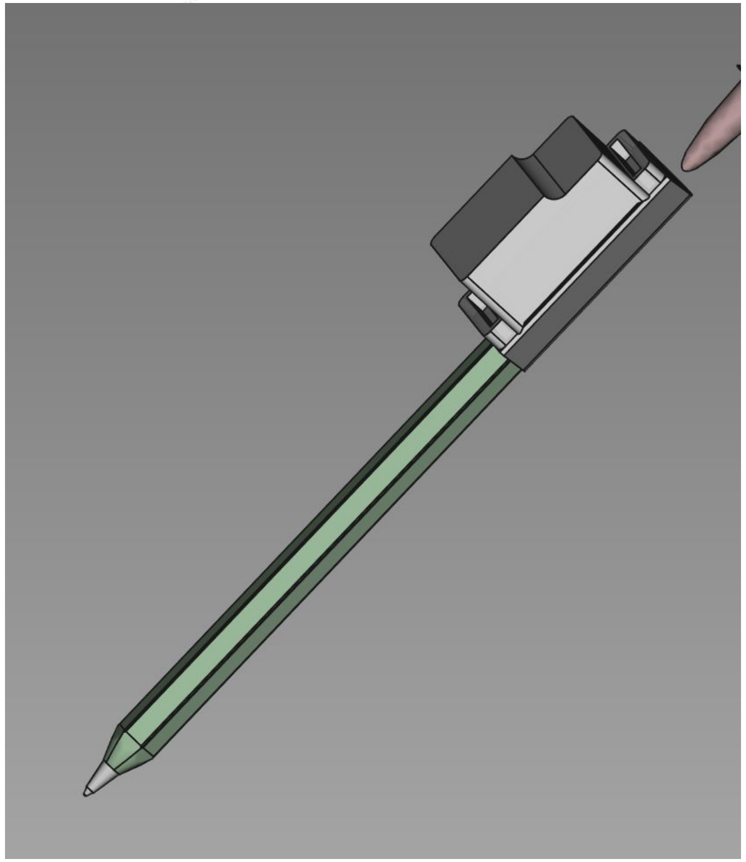
Workshop - “ABWriting”

Written letter recognition
ST Sensortile mounted on a pen
Using accelerometer data

- Collecting data
- Training NN
- Inference on the device

Wireless communication using BLE





ST SensorTile development kit

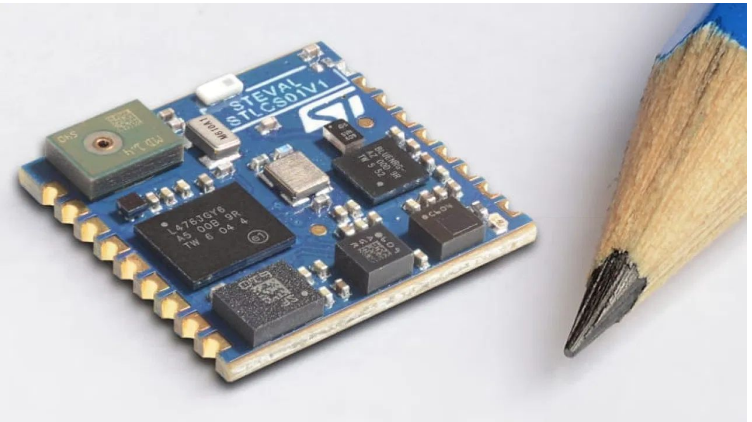
STM32L476JG microcontroller - Cortex M4

80MHz - 1MB Flash - 128KB SRAM

Bluetooth Low Energy

Sensors:

- Accelerometer
- Gyroscope
- Magnetometer
- Mems Microphone
- SD-Card
- Temperature
- Pressure



Tutorials about
- Magic Wand learning
- Image sensor & low-power
- Quantising, pruning with AIMET
- Industrial grade applications with Edge Impulse

Several keynotes, tinyTalks & breakout sessions

Free! Registration required.
https://www.tinyml.org/summit2021/

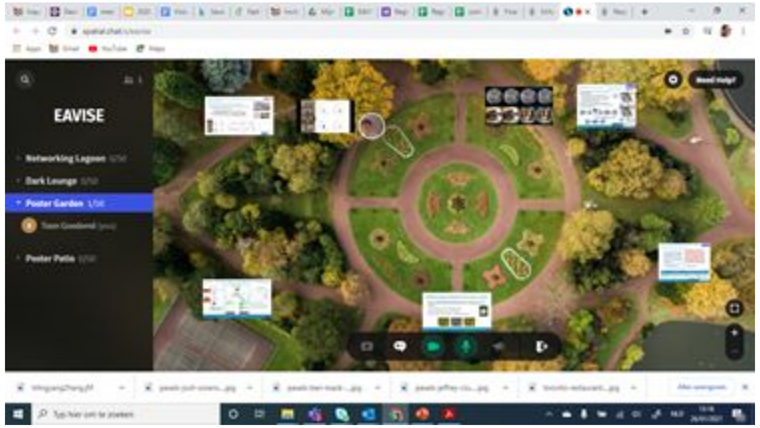
Administration

- Rules of procedure (reglement van orde)
- User Poll
- Next user group meeting: June 2021

Networking: https://spatial.chat/s/eavise

Access Procedure

- **Click** on the following link: https://spatial.chat/s/eavise
- Fill in your name in the *Full Name* box and your company in the *About* box
- Allow the browser to use your microphone and camera
- Check your audiovisual settings and click on *Join Space*



Getting around in the space

- **Drag** your avatar around to freely move through the space. If you are **close enough** to other people, you **can talk** to each other
- The space is split up in **two rooms**. You can **click on a name of a room** in the lefthand list to jump to that room.
- **Click on a name of another visitor** at the lefthand list to move quickly towards him or her and start a conversation.