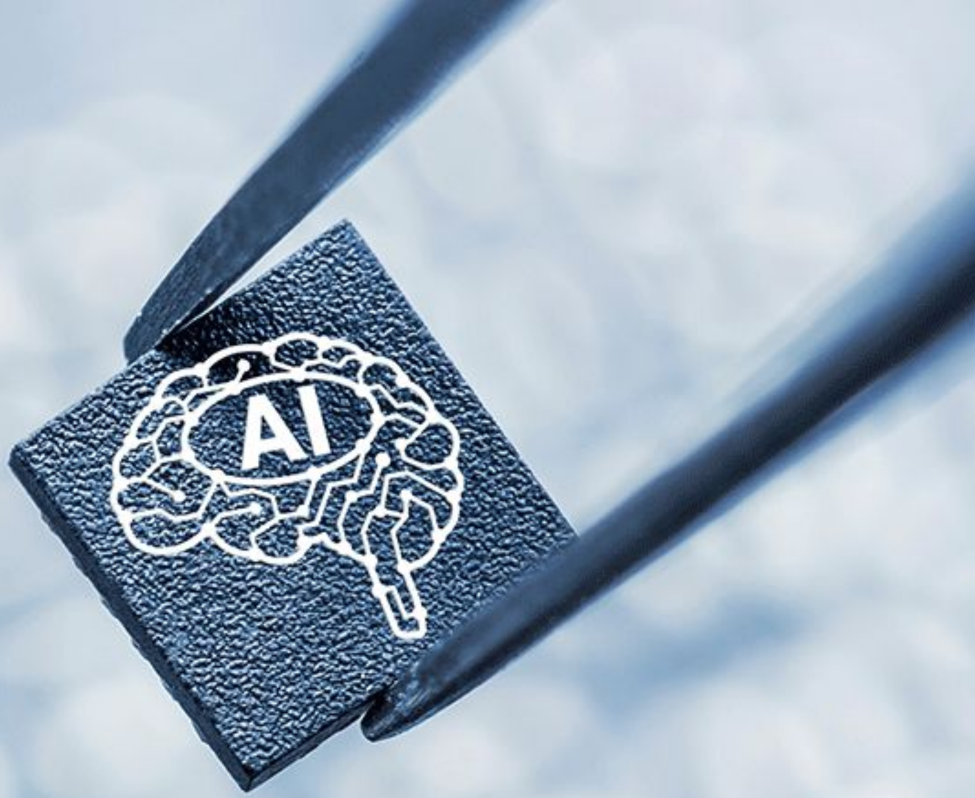
Final Symposium TETRA AI@EDGE

Artificial Intelligence, not in the Cloud, but on a microcontroller!



VLAIO TETRA HBC.2019.2641

ai-edge.be

iot-incubator.be www.eavise.be

hogeschool vives KU LEUVEN

Agenda

1. Project introduction
2. Project results
3. Academic use-cases
4. Industrial use-cases
5. Testimony from the industry
6. Conclusion
7. Use-case exhibition & reception

Introduction

Tools that we all know…

Google

DeepL

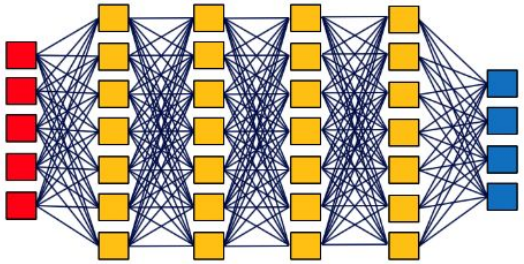
They use AI!

Introduction

How are they created?

Huge dataset

Storage in the Cloud



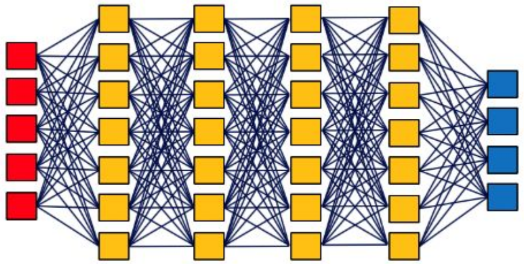
Training of AI network

Introduction

How do we use them?

Application

Storage in
the Cloud



Dog!

Run it on the
AI network

Introduction

Useful! But are there any issues?

- Latency
- Network requirement
- Data security/privacy
- Cost
- Energy

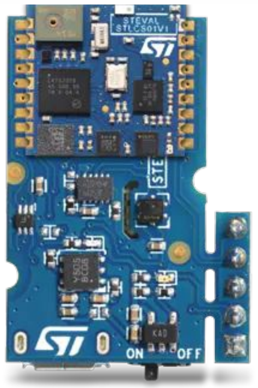
Introduction

AI on embedded devices = EDGE

- Single board computers
- Microcontrollers
- Microprocessors

Advantages

- Low-cost
- Low-power
- Low-latency
- Local data / privacy

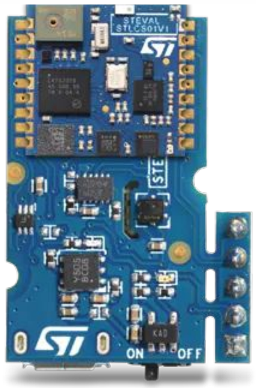


Introduction

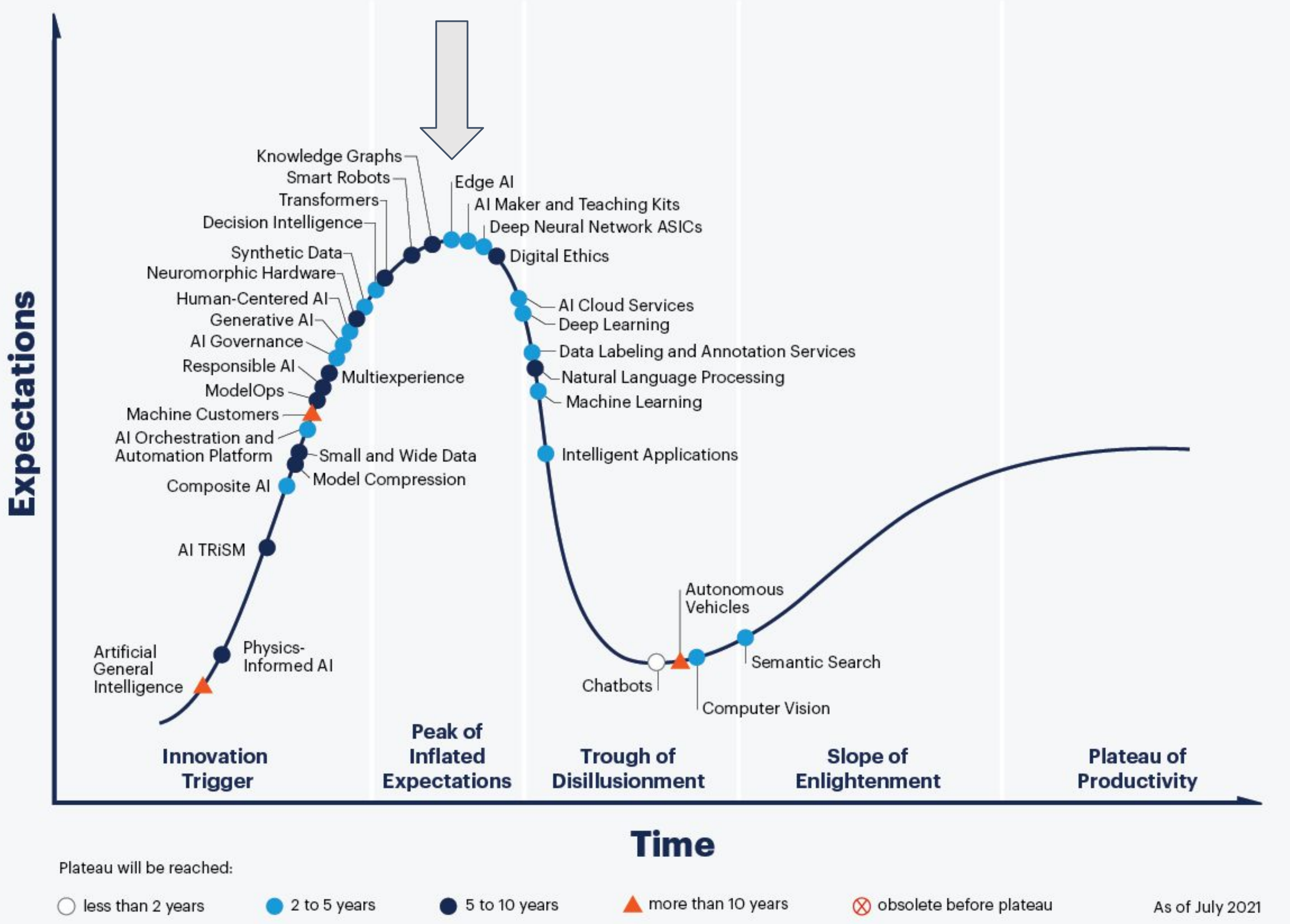
Disadvantages/challenges?

Limited resources:

- Memory
- Accuracy
- Computing power
- Development



Hype Cycle for Artificial Intelligence, 2021



gartner.com

Source: Gartner

AI@EDGE

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

Project partners

IoT Incubator

VIVES campus Brugge

https://iot-incubator.be/


KU Leuven campus De Nayer

https://eavise.be/



User Group Members

Duo LarbitsSisters
Larbitslab

Research questions

1. Identify the possibilities and application for Deep Learning on low-cost embedded devices.
2. What are the restrictions of the hardware?
3. What are the available software libraries and frameworks and how are they used?
4. What about the accuracy of the models?
5. What is the trade-off between efficiency and quality?
6. Which optimisation techniques can be used?
7. What about the energy usage of the system?
8. How can privacy and latency be improved by making decisions locally and autonomously?

Research questions

1. Identify the possibilities and application for **Deep Learning** on low-cost embedded devices.
2. What are the **restrictions** of the hardware?
3. What are the available **software libraries** and **frameworks** and how are they used?
4. What about the **accuracy** of the models?
5. What is the trade-off between **efficiency** and **quality**?
6. Which **optimisation techniques** can be used?
7. What about the **energy usage** of the system?
8. How can **privacy** and **latency** be improved by making decisions locally and autonomously?

Project goals

1. Overview of frameworks and hardware
2. Manual with best-practices, optimisation techniques
3. Create 6 industrial use-cases as a reference and inspiration
4. Create 2 proof-of-concept cases to organise hands-on workshops to experience Deep Learning on low-cost embedded hardware

Workplan

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 of the user group

WP2: Proof of concept (6 mm)

- WP 2.1: selection hardware and frameworks
- WP 2.2: collect and annotate data
- WP 2.3: implementation
- WP 2.4: test and validation

WP3: Industrial case studies (18 mm)

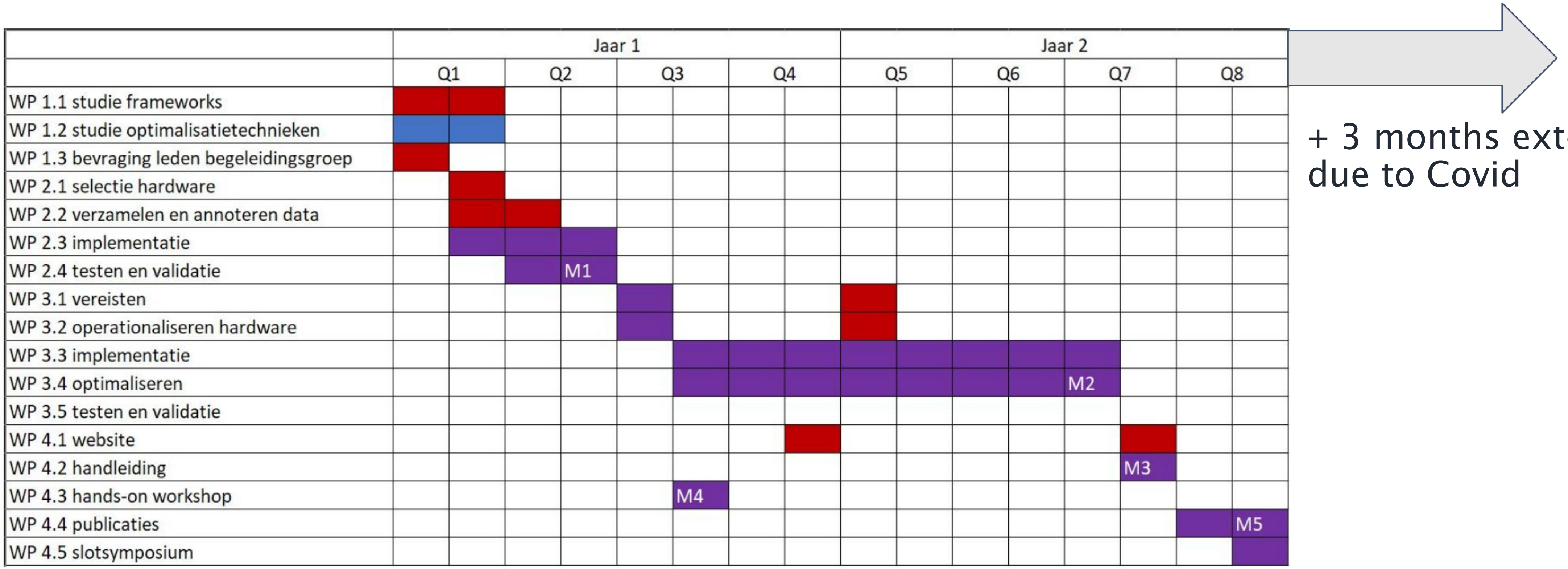
- WP 3.1: gather functional and non-functional requirements
- WP 3.2: select and operationalise hardware and framework
- WP 3.3: implementation
- WP 3.4: optimisation
- WP 3.5: test and validation

WP4: Valorisation (9 mm)

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

Gantt chart

Project timeline: March ‘19 - May ‘22



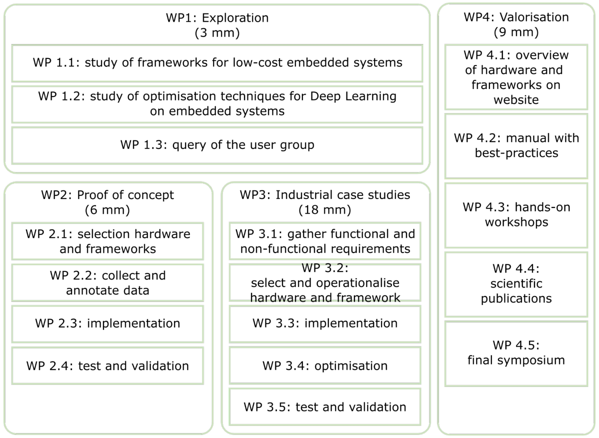
+ 3 months extension due to Covid

hogeschool vives KU LEUVEN

AI@EDGE

Project Results

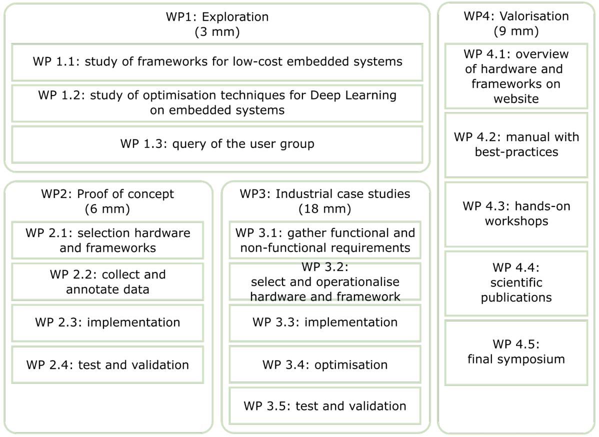
Project results (WP2)



- Proof-of-concepts (WP2):
  - **Goal:** diverse and more generalised case studies
    ⇒ should be beneficial for all companies
    ⇒ discussed with user group after initial exploration (WP1)

- Should contain the entire workflow
  - HW ⇒ collecting data ⇒ implementation ⇒ quantisation (if needed) ⇒ validation

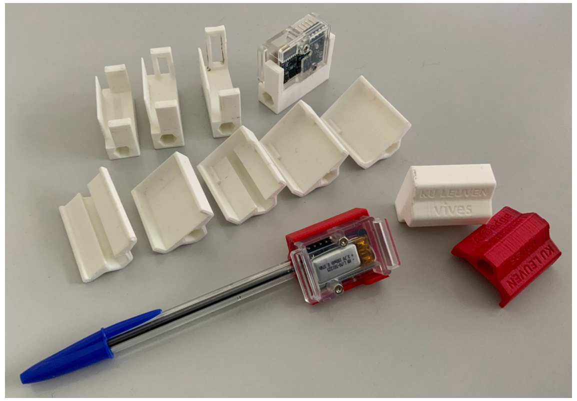
Project results (WP2)



- Crystallized as three *academic* use-cases
  - AB writing (accelerometer)
  - Car classification (computer vision)
  - People counting in rooms (seat detection)

- Each *academic* use-case resulted in a documented workshop
  - Repeated multiple times
  - Increasing level of complexity
  - Education/students highly involved (e.g. record data, testing workshop)

Project results (WP2)



- Academic use-case **AB writing**:
  - Detection of <u>handwritten</u> letters/symbols/numbers on small microcontroller
  - High-level: through Edge Impulse

- Resulted in first series of hands-on embedded workshops
- Given twice: 9 companies, 55 participants
- 09/12/2021 & 18/05/2022
- New session planned in September '22

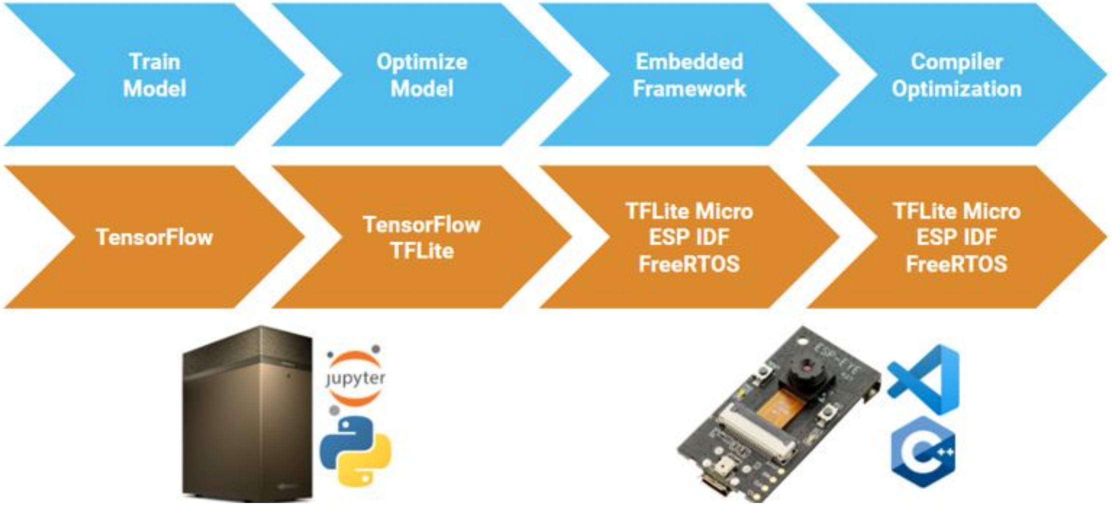
hogeschool vives KU LEUVEN AI@EDGE

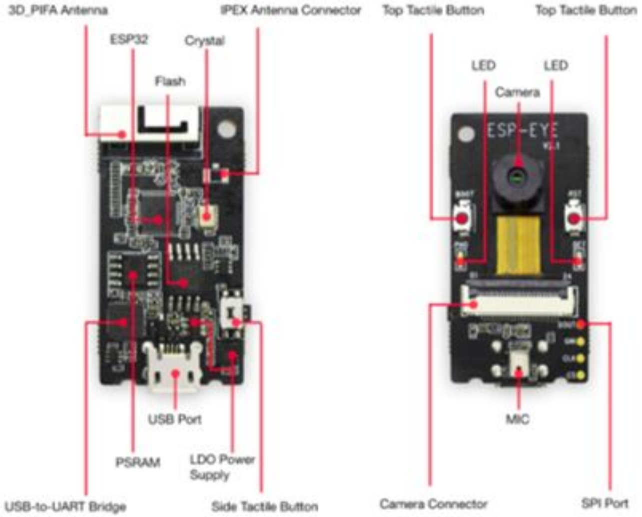
Project results (WP2)

- Academic use-case **Car classification**:
  - Computer vision task
  - Low-level programming on the MCU
  - Embedded optimizations (quantization) & implementation

- Resulted in second series of hands-on embedded workshops
- 11 companies, 20 participants
- 22/04/2022
- New session planned in September ‘22

Project results (WP2)





ESP32 MCU

Xtensa Dual-Core 32-bit LX6
240 MHz Clock
512 kB RAM
36 GPIO
WIFI stack
Bluetooth stack
$ 6 - 12

2 MP color camera
4 MB External SPI Flash
8 MB External SPI PSRAM
$ 20

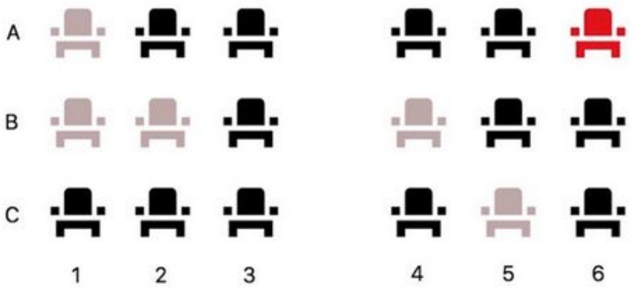
hogeschool vives KU LEUVEN

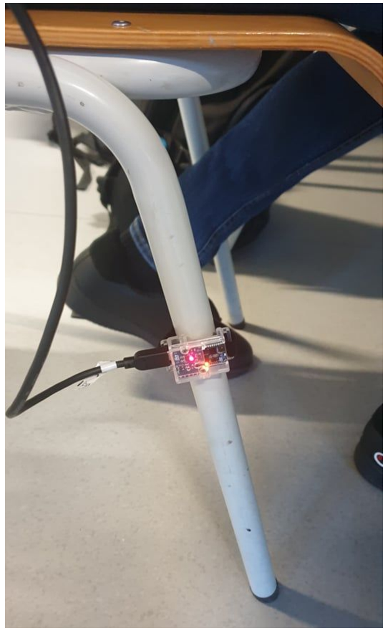
Project results (WP2)

- A down-scaled version of this workshop was developed as STEM workshop (targeted towards secondary schools)
- Automated 3D printed garage door for matchbox cars (see further)

- 29/03/2022 AM
- 02/04/2022 AM
- 02/04/2022 PM
- 03/06/2022 AM

Project results (WP2)





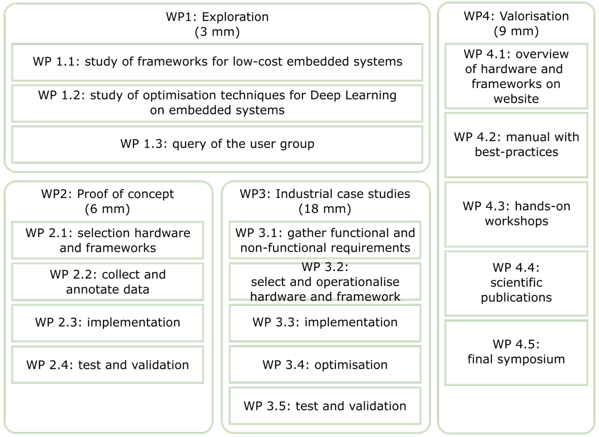
- Academic use-case: **people counting**
- Based on accelerometer on seat
- For educational course “AI Edge Computing”

- Summarization WP2:

  ⇒ Developed 4 generic documented use-cases with accompanying workshop/educational course

  ⇒ Reached ~180 individuals

Project results (WP3)



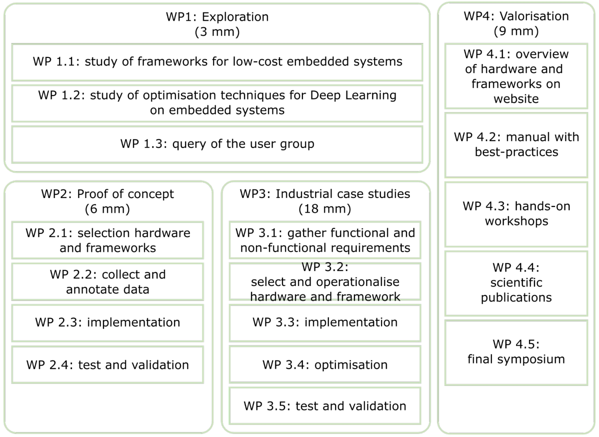
- *Industrial* use-cases (WP3):
  - Specific use-cases for individual companies
  - Highly diverse topics:
    - Traffic monitoring, thermal person detection, capacitive touch sensors, medical sector, AI in art exhibition,...

- A total of five *industrial* use-cases were investigated and documented

Project results (WP3)

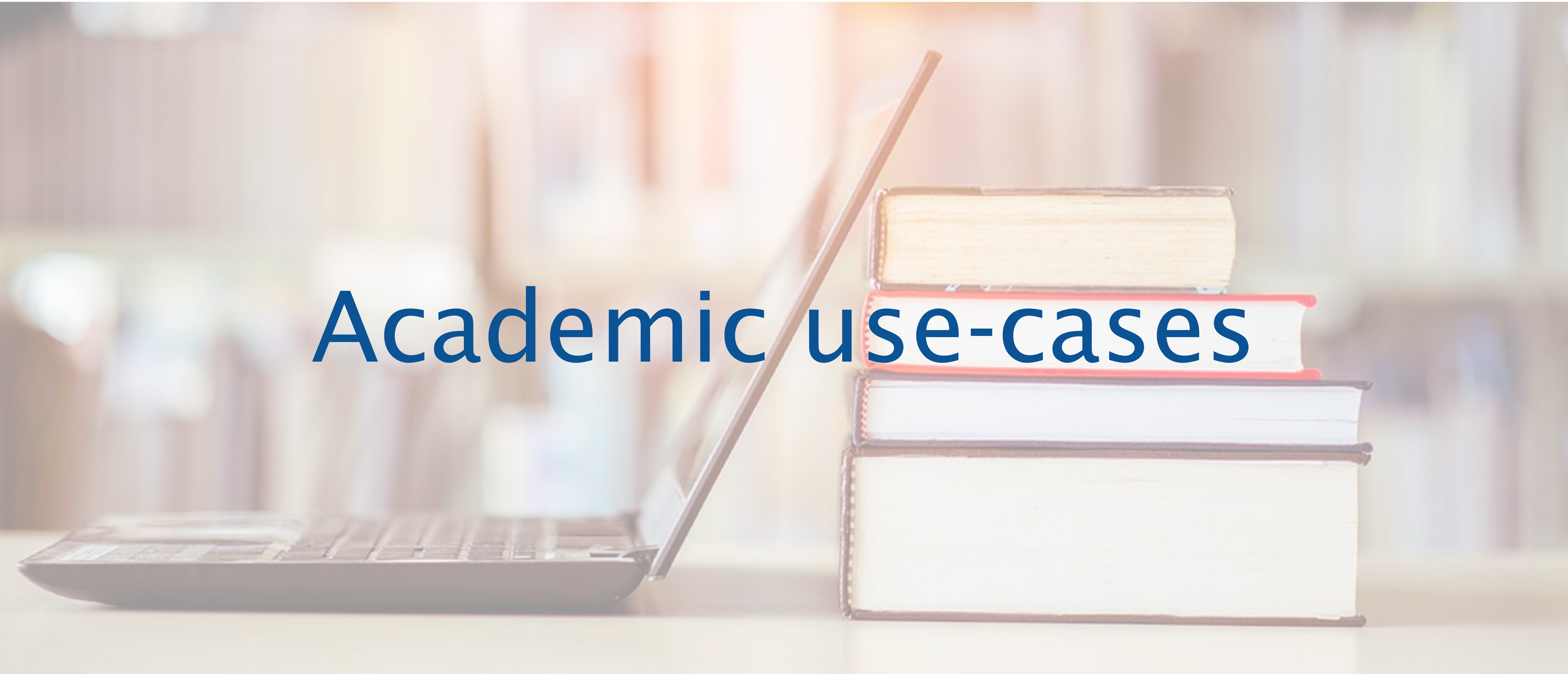
- **Melexis:** Person detection in low-resolution thermal sensors on low-cost hardware
- **E.D&A.:** AI to optimize key-press detection with a capacitive sensor (induction heater)
- **TML:** Traffic counting (i.e. road user detection and tracking) on RPi
- **6Wolves/Yogalife:** evaluation of effectiveness of fitness exercises
- **Artists Duo - LarbitsSisters:** development of AI-driven autonomous robot

Project results (WP4)



- Valorisation (WP4):
  - Four **user group meetings**
  - Numerous **workshops** (see above)
  - Final **symposium** (today)
  - Several **educational courses** were involved:
    - AI Edge (Computing), Embedded AI, Smart Embedded Electronics
  - **Publications**:
    ⇒ One bachelor thesis
    ⇒ Master thesis: Deep mobile product recognition: applying deep learning on a smartphone
    ⇒ PhD thesis M. Vandersteegen
  - **Project website (https://ai-edge.be/)**: Documentation and guidelines

Academic use-cases

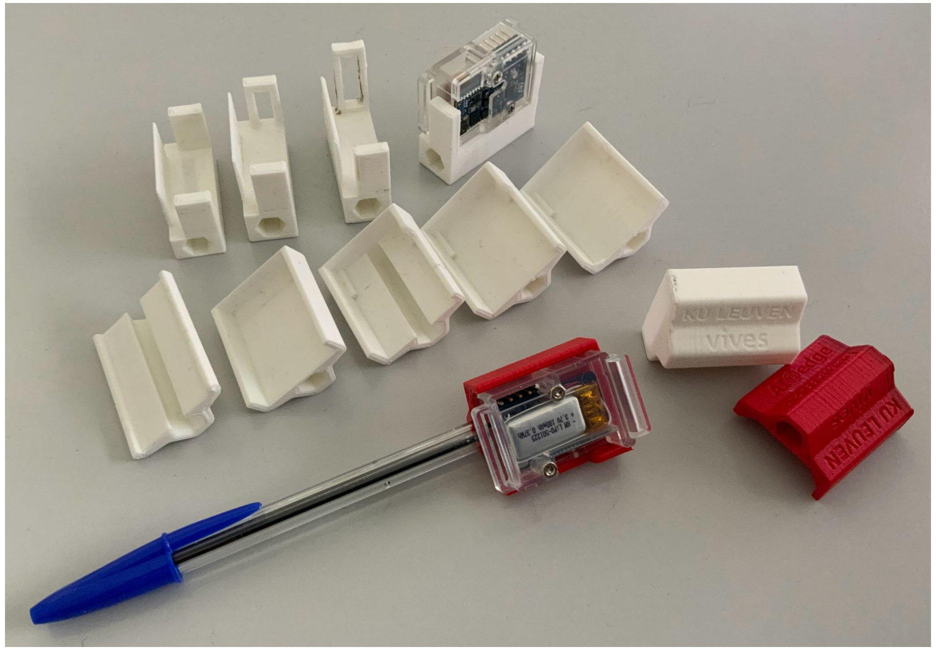


AB-Writing

Goal: Detection of <u>handwritten</u> letters/symbols/numbers

Challenges:

- Using accelerometer
- Small microcontroller



Approach:

- STM Sensortile
- 3D printed housing
- Mounted on a pen/pencil

STM32L476JGY (Cortex M4)
80 MHz
128 KB RAM
1 MB Flash

AB-Writing

EDGE IMPULSE

- Step 1: Framework → Edge Impulse

  Ease of use, no code
  Tools to import and annotate data
  High level AI model generation
  Automatic quantisation
  Deployment to different hardware

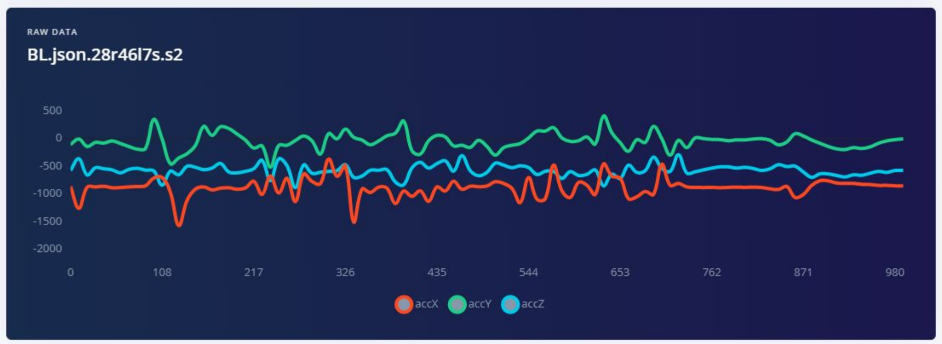
AB-Writing

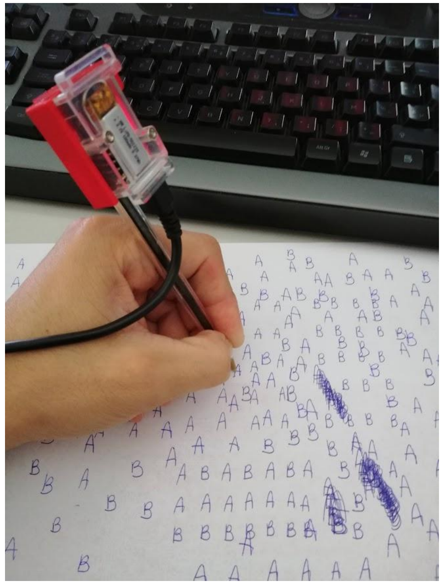
- Step 2: Data acquisition & annotation

Data from colleagues, students, workshop attendees

Dataset classes:

Symbol, hand (L/R), idle





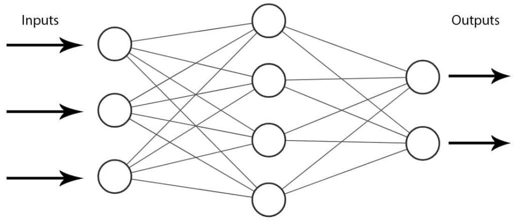
AB-Writing

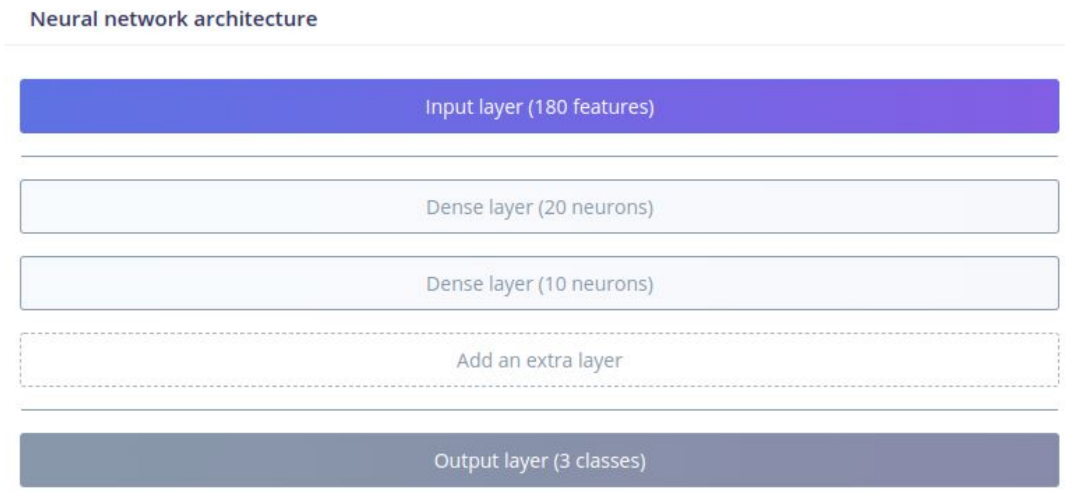
- Step 3: Implementation

Preprocessing: none/raw (DL approach)

Analysis of network size & layers

- FFNN + x hidden layers
- Time domain input
- Online training





AB-Writing

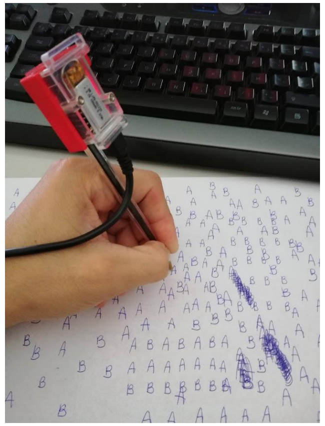
EDGE IMPULSE

- Step 4: Testing & validation

Validate model with test data

Deployment on µC

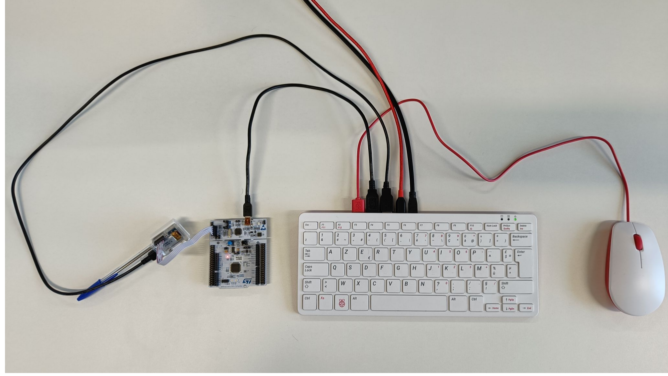
	AL	AR	BL	BR	IDLE
AL	61.1%	16.7%	16.7%	0%	5.6%
AR	0%	63.2%	5.3%	31.6%	0%
BL	9.5%	0%	81.0%	9.5%	0%
BR	5.9%	29.4%	5.9%	58.8%	0%
IDLE	0%	0%	0%	0%	100%
F1 SCORE	0.69	0.62	0.79	0.57	0.99



Live classification →

```
run_classifier returned: 0
Predictions (DSP: 0 ms., Classification: 1 ms., Anomaly: 0 ms.):
O: 0.00000
X: 0.00000
idle: 0.99609
run_classifier returned: 0
Predictions (DSP: 0 ms., Classification: 1 ms., Anomaly: 0 ms.):
O: 0.00000
X: 0.00000
idle: 0.99609
run_classifier returned: 0
Predictions (DSP: 0 ms., Classification: 1 ms., Anomaly: 0 ms.):
O: 0.00000
X: 0.00000
idle: 0.99609
```

AB-Writing



Result embedded in hands-on workshops

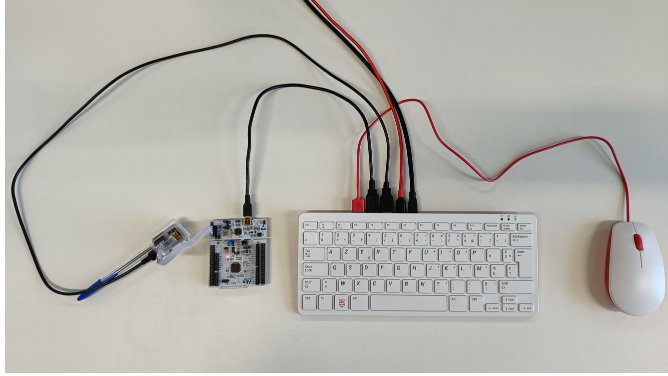
Full workflow:

Data collection → inference

Hardware: Raspberry Pi 400 + µC

Dataset growing with workshops

AB-Writing - Conclusions



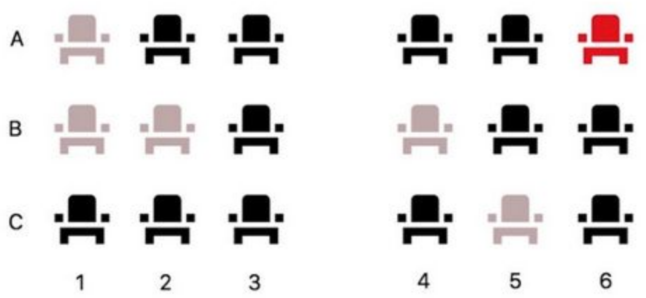
Lessons learned:

- Small datasets give poor results
- Preprocessing the raw data can improve results

Outtakes:

- Simple NN training can be done on a RPi
- Edge Impulse for quick non-code prototyping

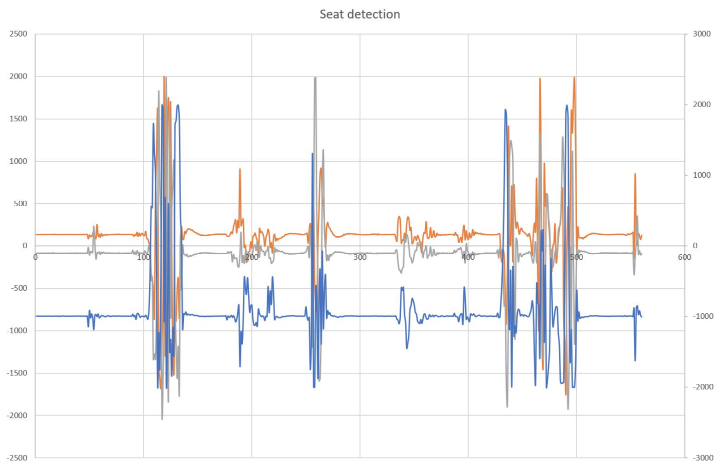
Seat Detection



Goal: Count number of people in a room

Challenges:

- Prevent false positive (eg, cleaning personnel will move all seats)
- Large interference from nearby movements
- Accelerometer, gyroscope, magnetometer?



Approach:

- Small microcontroller: STM Sensortile
- Accelerometer

Seat Detection

Problem solved by students: Course “AI Edge Computing”

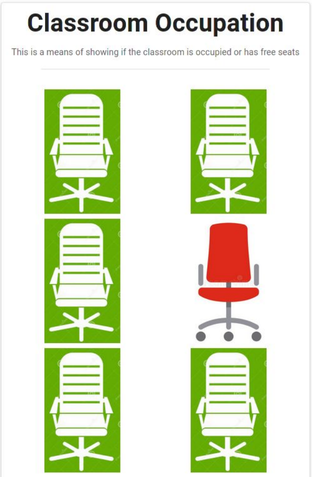
Two teams: Kortrijk vs Brugge

Result: two approaches

1. Static seat detection
2. Dynamic seat detection

Seat Detection

1. Static seat detection



Static => seat moving
or not moving

Slight vibration when seated

Team Brugge
https://ai-edge-raport.netlify.app/

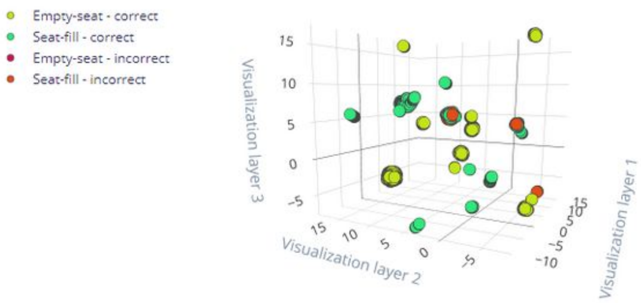
ACCURACY
93.5%

LOSS
0,33

Confusion matrix (validation set)

	EMPTY-SEAT	SEAT-FILL
EMPTY-SEAT	99.9%	0.1%
SEAT-FILL	15.7%	84.3%
F1 SCORE	0.95	0.91

Feature explorer (full training set)



On-device performance

INFERENCING TIME
9 ms.

PEAK RAM USAGE
2,3K

FLASH USAGE
54,9K

hogeschool vives KU LEUVEN AI@EDGE

Seat Detection

2. Dynamic seat detection

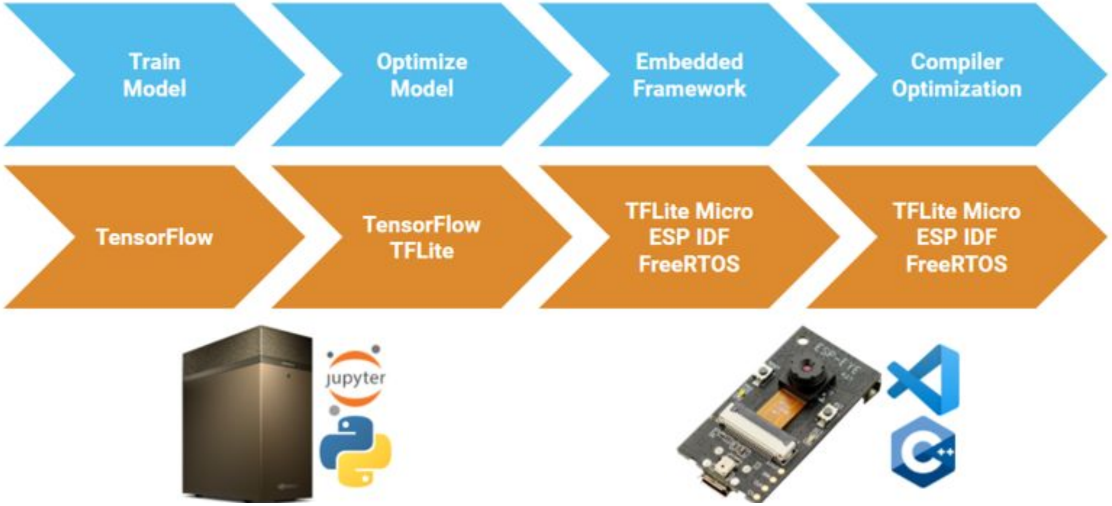
Dynamic = movement detection of seat

Forward & backwards sliding

Team Kortrijk:
https://github.com/VIVES-AI-edge-computing/seat-detection-team-kortrijk/tree/main/report/docs/src/guide

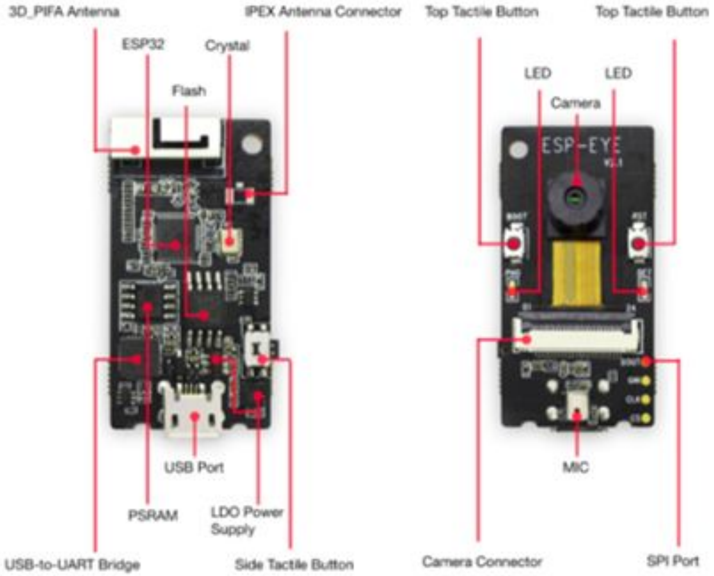
Car Detection

Workshop Embedded AI Optimization



hands-on workshop on 22/04/2022

-> repeat on 15/09/2022 (PUC summer school Kortrijk)



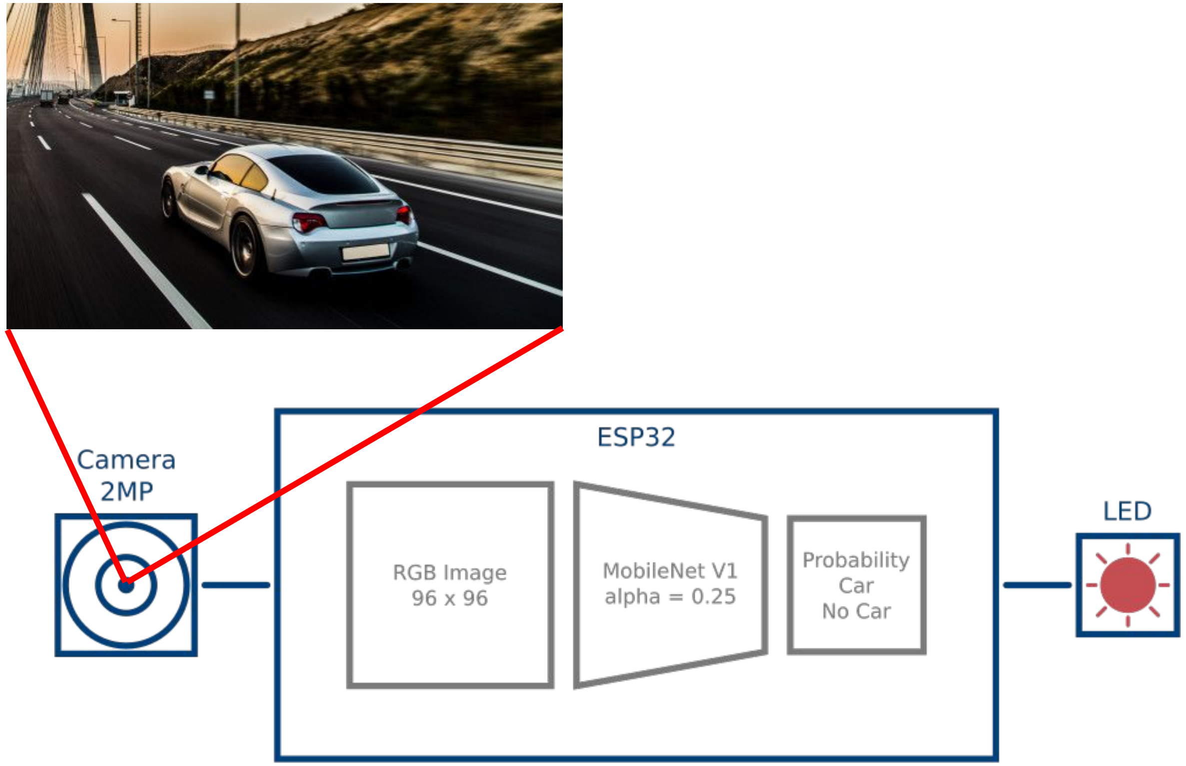
ESP32 MCU

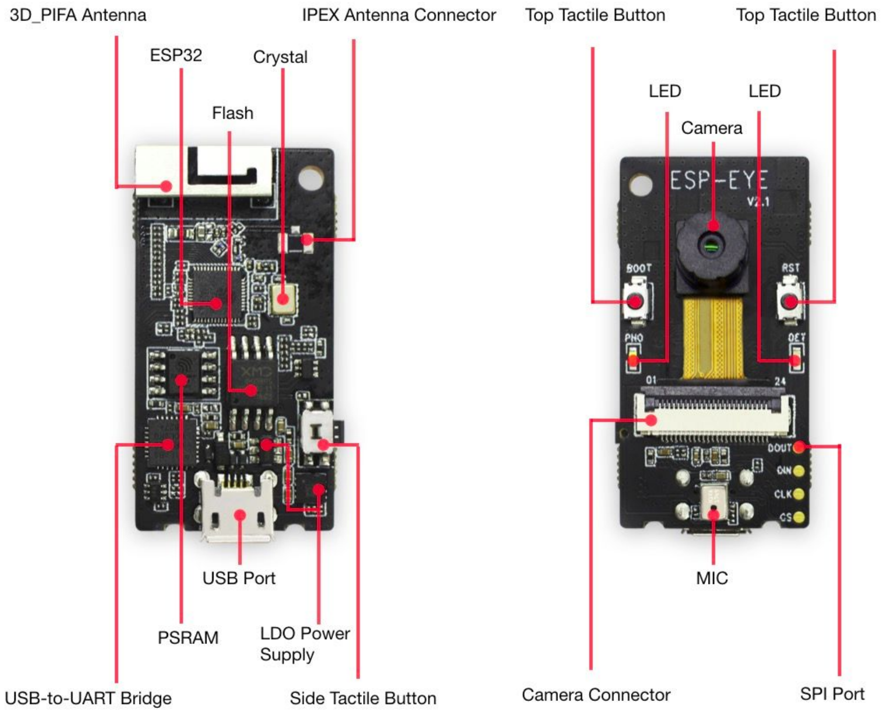
Xtensa Dual-Core 32-bit LX6
240 MHz Clock
512 kB RAM
36 GPIO
WIFI stack
Bluetooth stack
$ 6 - 12

2 MP color camera
4 MB External SPI Flash
8 MB External SPI PSRAM
$ 20

hogeschool vives KU LEUVEN

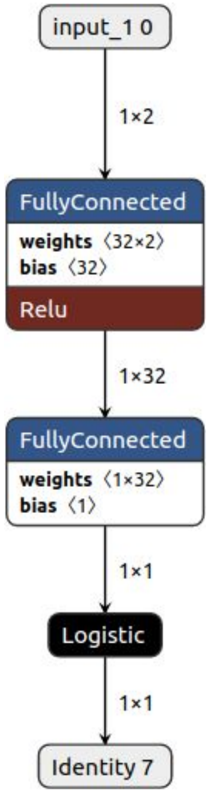
Workshop Embedded AI Optimization





Workshop Embedded AI Optimization

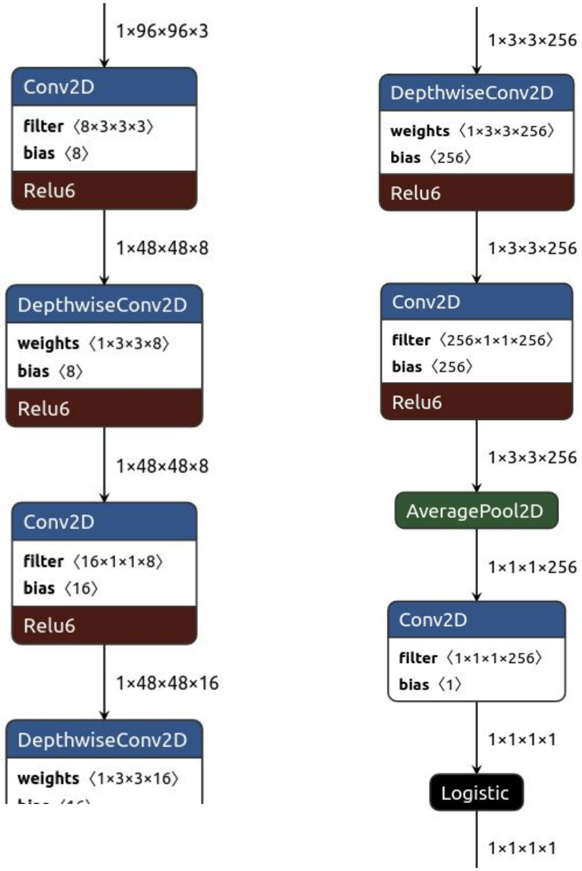
Toy example:
XOR gate



Define and Train Model	Convert Model	Deploy
TensorFlow	TensorFlow TFLite	TFLite Micro ESP IDF

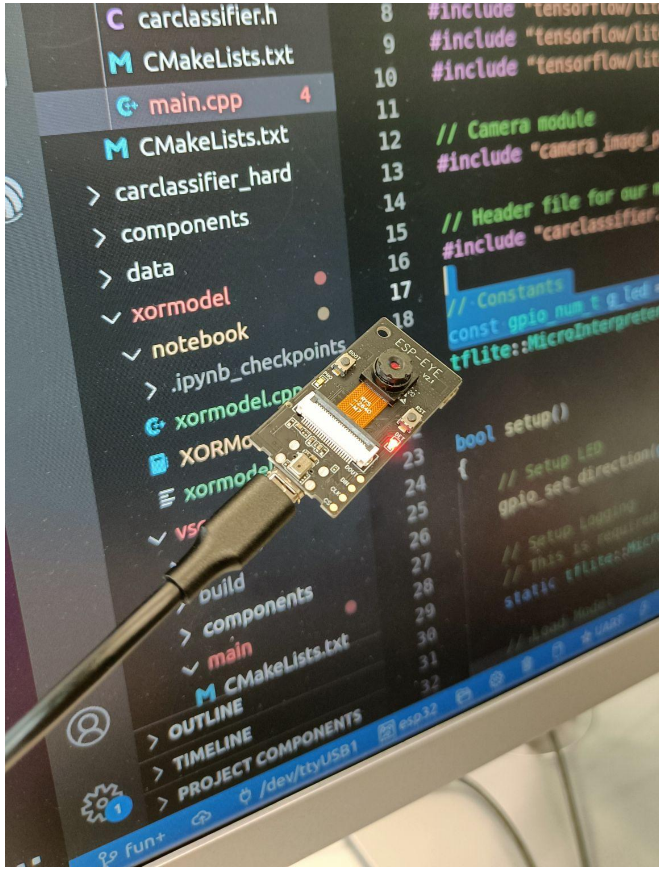
Workshop Embedded AI Optimization

Car classifier
(MobileNetV1 α=0.25)



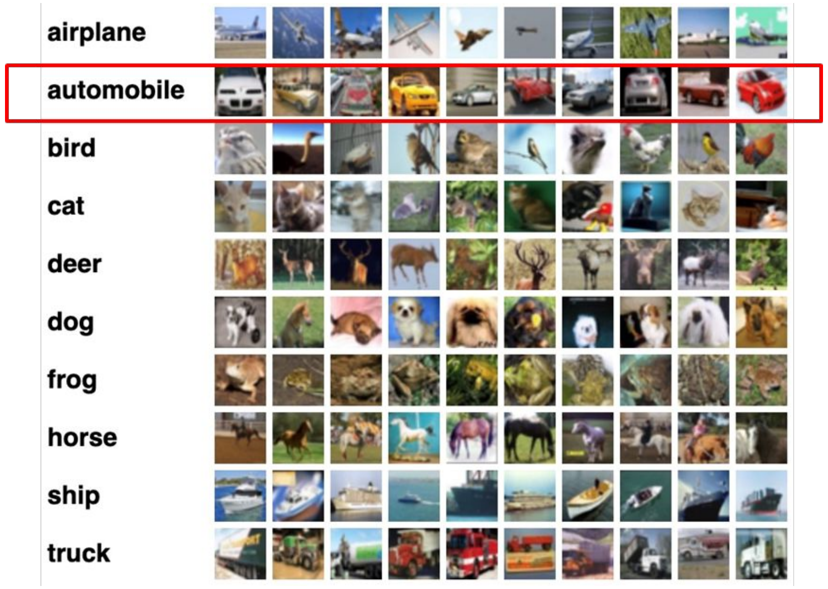
- Load pre-trained Model → Convert and quantize Model (8-bit) → Deploy
- TensorFlow → TensorFlow TFLite → TFLite Micro ESP IDF

Workshop Embedded AI Optimization



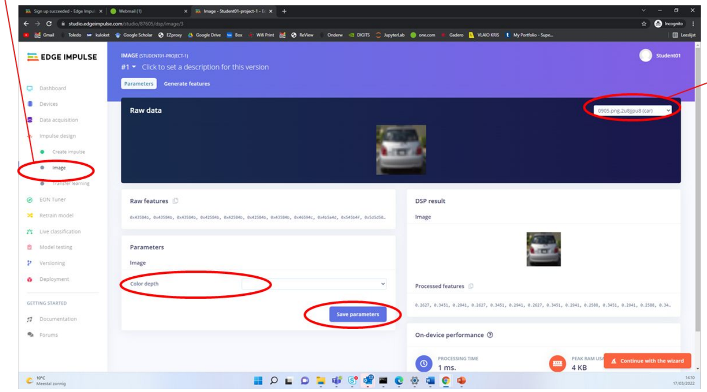
Workshop STEM (secondary school)

- Hands-on embedded Deep Learning experience for youth
  - collecting data (CIFAR10 + custom)
  - training model (MobileNetV2)
  - evaluation
  - deployment on RPi
  - test with real toy garage



EDGE IMPULSE

In het menu, klik op "Image"



Hier kan je je beelden bekijken

5x30 auto-foto's met verschillende standpunten, belichting en achtergronden

150 foto's met andere objecten dan auto's, of lege achtergrond

hogeschool vives KU LEUVEN

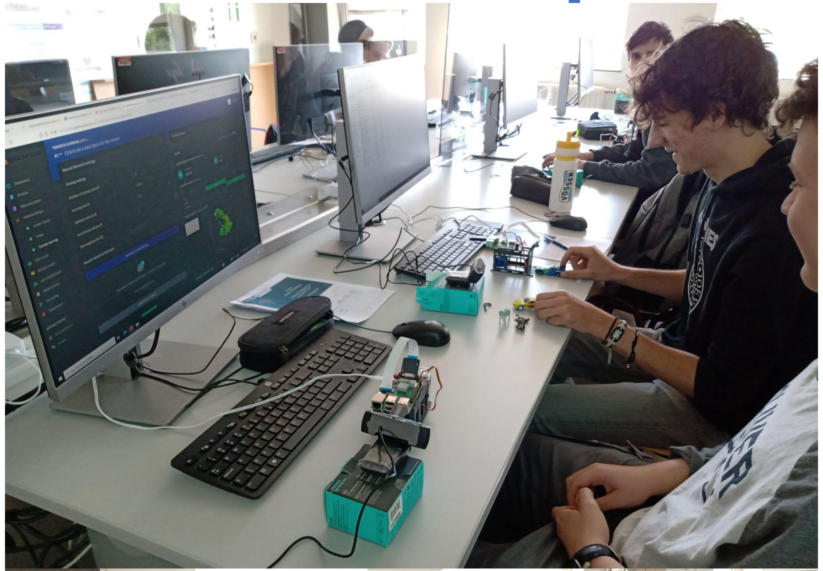
Workshop STEM (secondary school)

Workshop booked:

- 29/03/2022 AM
- 02/04/2022 AM
- 02/04/2022 PM
- 03/06/2022 AM
- ...

Lessons learned in STEM workshop

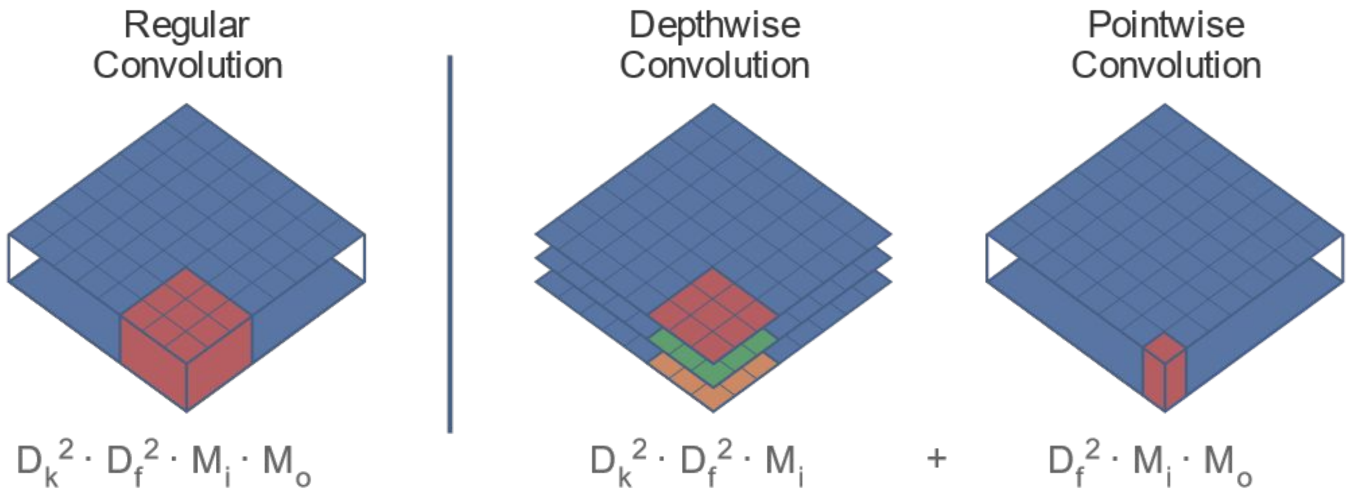
- Methodology of deep learning
- Need for labeled datasets
  - large enough & representative
  - train/test split
- Evaluation of classifier
  - validation/test accuracy
  - confusion matrix
  - real-life test
- Improve performance
  - training duration (epochs)
  - image resolution
  - model size
  - dataset
- Overfitting



hogeschool vives KU LEUVEN

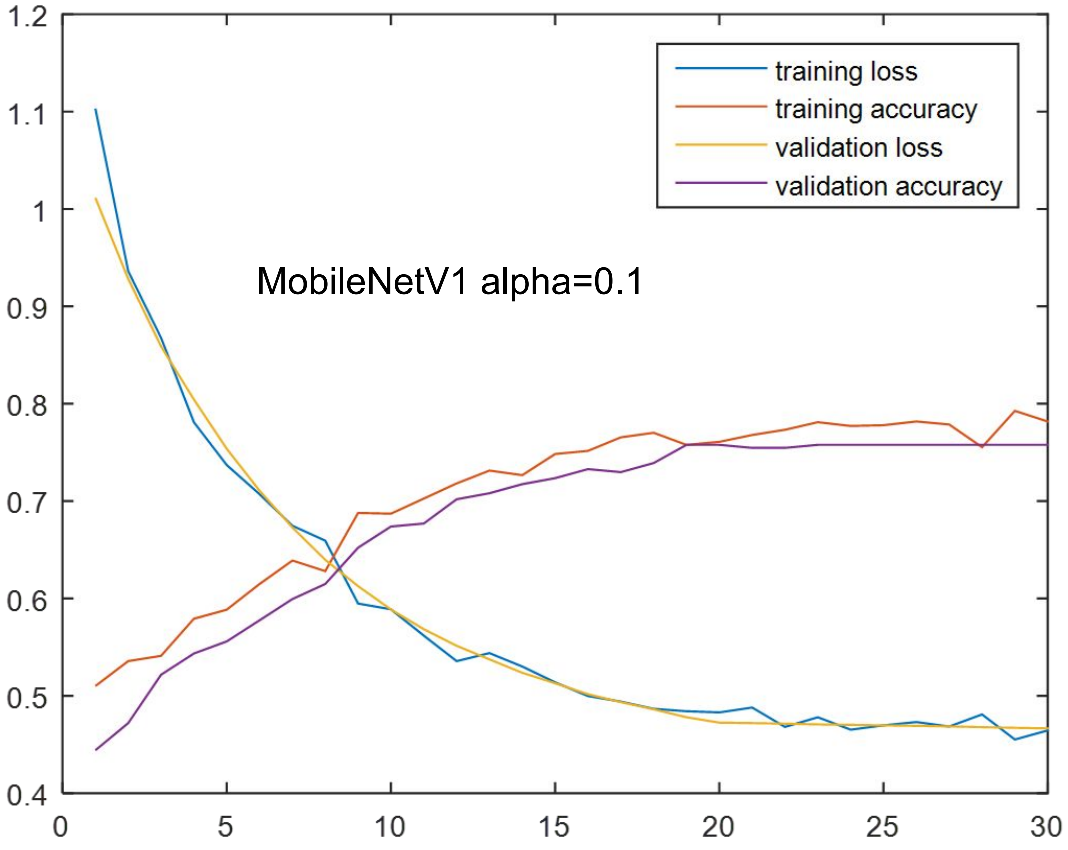
Efficient CNN Models

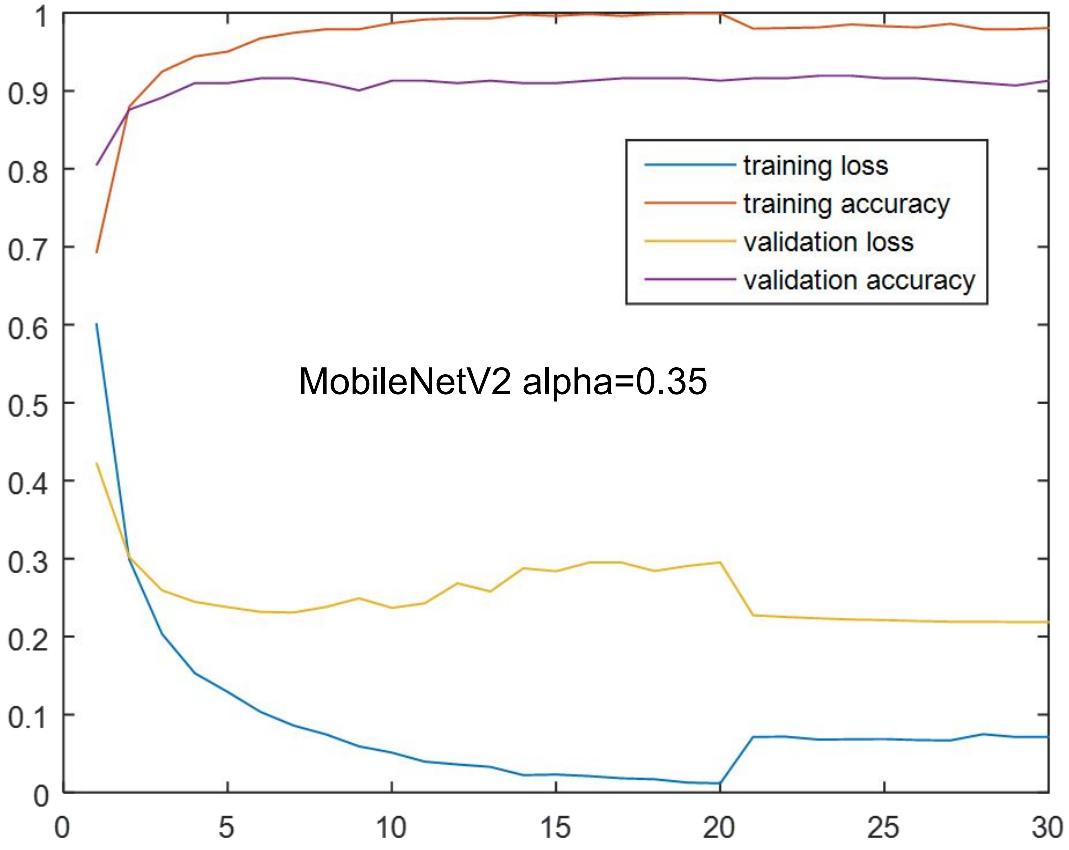
model	RAM	ROM	inference time on Raspberry PI
MobileNetV1 96x96 alpha=0.1	66.1K	108K	26 ms
MobileNetV2 96x96 alpha=0.35	346.6K	575.5K	103 ms



hogeschool vives KU LEUVEN AI@EDGE

Looking at training graphs





Looking at evaluation dashboard

Lab book STEM workshop

dataset	image resolution	model	nb of epochs	validation accuracy (float32)	validation accuracy (quantized int8)	test accuracy (float 32)
CIFAR10 1000+1000	32x32	MobileNetV1 alpha=0.1	20	73.6%	67.4%	55.38%
CIFAR10 1000+1000	32x32	MobileNetV1 alpha=0.1	40	73.9%	72.7%	58.72%
CIFAR10 1000+1000	32x32	MobileNetV2 alpha=0.35	20	85.4%	86.3%	81.54%
CIFAR10 1000+1000	96x96	MobileNetV2 alpha=0.35	20	**90.7%**	51.9%	**90.51%**

Lab book STEM workshop

dataset	image resolution	model	nb of epochs	validation accuracy (float32)	validation accuracy (quantized int8)	test accuracy (float 32)
CIFAR10 1000+1000	32x32	MobileNetV1 alpha=0.1	20	73.6%	67.4%	55.38%
CIFAR10 1000+1000	32x32	MobileNetV1 alpha=0.1	40	73.9%	72.7%	58.72%
CIFAR10 1000+1000	32x32	MobileNetV2 alpha=0.35	20	85.4%	86.3%	81.54%
CIFAR10 1000+1000	96x96	MobileNetV2 alpha=0.35	20	**90.7%**	51.9%	**90.51%**
custom application-specific dataset 150+150	96x96	MobileNetV1 alpha=0.1	20	74.5%	72.5%	76.19%
custom application-specific dataset 150+150	96x96	MobileNetV1 alpha=0.1	40	82.4%	76.5%	87.30%
custom application-specific dataset 150+150	96x96	MobileNetV2 alpha=0.35	20	**100%**	**100%**	**100%**

Scientific conclusions STEM workshop

- AI model training test can be easily done without coding
  - online services exist, like e.g. Edge Impulse
  - including model optimization for embedded devices
- Performance increase possibilities:
  - hyperparameter tuning (training cycles, learning rate)
  - model size
  - image resolution
  - case-specific dataset
- Small models are less likely to overfit than large models on limited datasets

Industrial use-cases

TinyML person detection

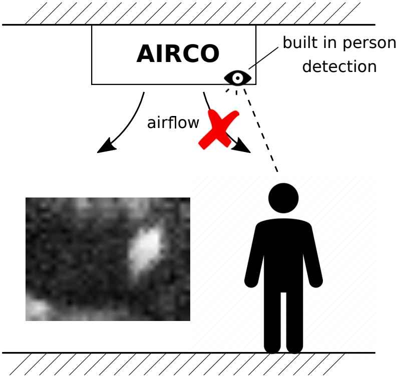
with a low resolution thermal imager

Maarten Vandersteegen
KU Leuven - EAVISE

KU LEUVEN eavise

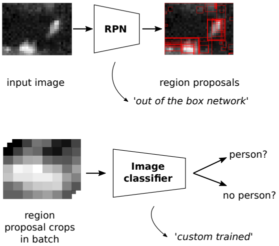
Usecase

- 32×24 pixels
- CPU of $\approx$ 8€



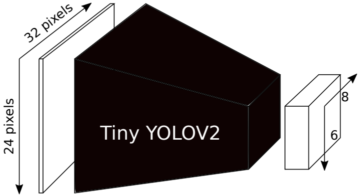
Initial Research

Melexis



- Two-stage R-CNN
- 88.56% AP
- Not deployable on MCU

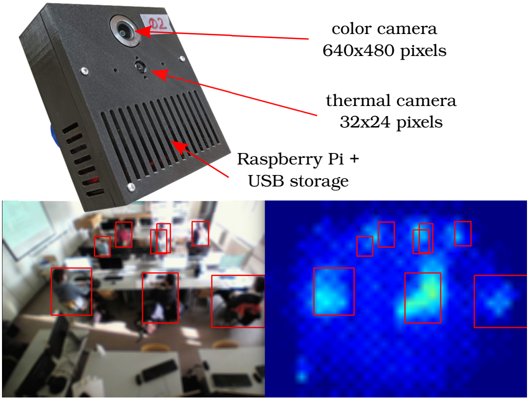
Ours



- Modified Tiny YOLOV2
- **98.86%** AP
- **67.42ms** on Cortex-M7 after compression

RECORDING A REPRESENTATIVE DATASET

1. Record videos with custom recorder
2. Run object tracking offline (Mask-RCNN)
3. Manual edit tracks in annotation tool
4. Copy bounding boxes to thermal image plane



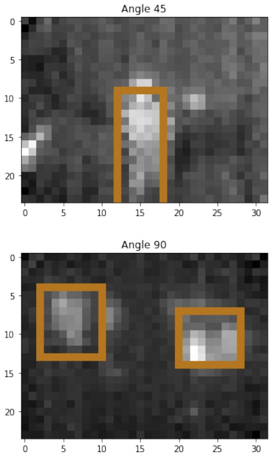
DATASET SUMMARY

Initial dataset:

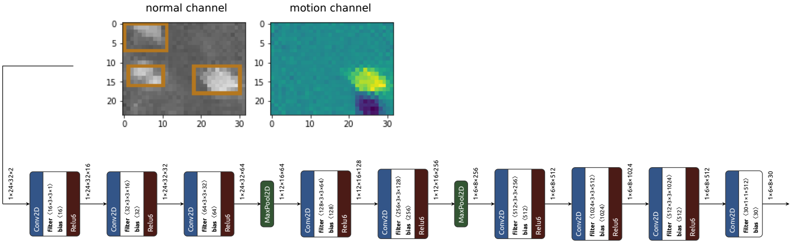
- 1 locations (room)
- 1 camera viewpoints
- 8k annotated video frames

New dataset:

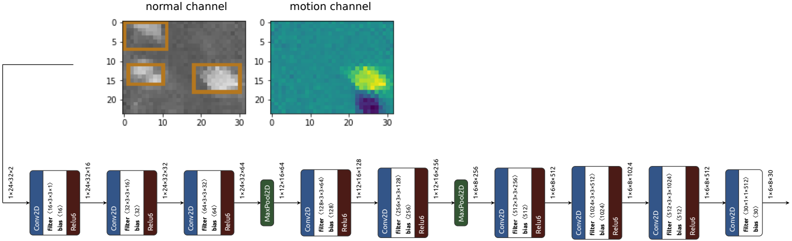
- 10 different recording locations (rooms)
- 26 different camera viewpoints
- 90k annotated video frames



Model and Results



Model and Results



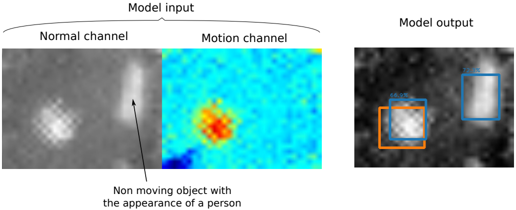
AP results on validation set

Camera angle	Single image	Single image + diff image
90	74% AP	76% AP
45 + 90	68% AP	TODO

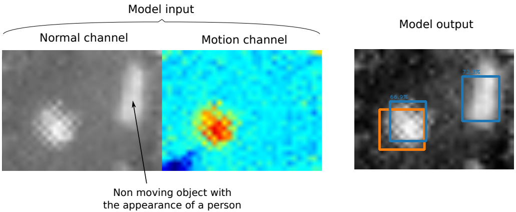
F1-score results on test set

Camera angle	Ours	Melexis std software
90	74%	88%
45	48%	62%

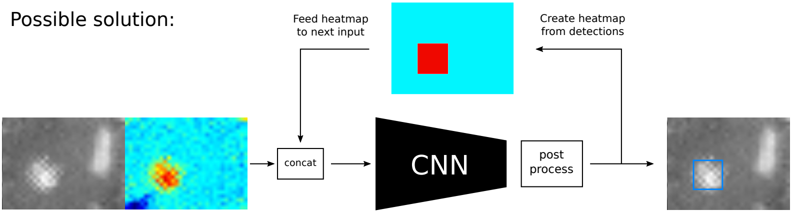
What is wrong?



WHAT IS WRONG?

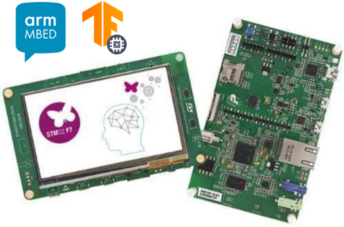


Possible solution:



Compression steps:

- Replace each conv with a depthwise separable conv
- Apply L2-norm based channel pruning for several iterations
- Apply post-training quantization to 8-bit



COMPRESSION AND DEPLOYMENT

Compression steps:

- Replace each conv with a depthwise separable conv
- Apply L2-norm based channel pruning for several iterations
- Apply post-training quantization to 8-bit

Results model 90 degree camera angle

Configuration	Valid acc (% AP)	#params	MACs	Inference time Cortex-M7
Regular	80%	11M	600M	Does not fit
Regular + pruned + quant	75.7%	141k (÷ 78)	22M (÷ 27)	220ms
Mobile	78%	1.3M (÷ 8.5)	68M (÷ 8.8)	Does not fit
Mobile + pruned + quant	75%	**43k (÷ 255)**	**5M (÷ 120)**	85ms (real-time)

Conclusion & future work

Conclusion:

- Acquired a larger dataset
- Proposed and trained a working object detector
- Successful optimization for real-time performance on Cortex-M7
- Still less accurate compared to existing software of Melexis

Future work:

- Implement long-term memory mechanism

E.D.&A. - Induction Heater

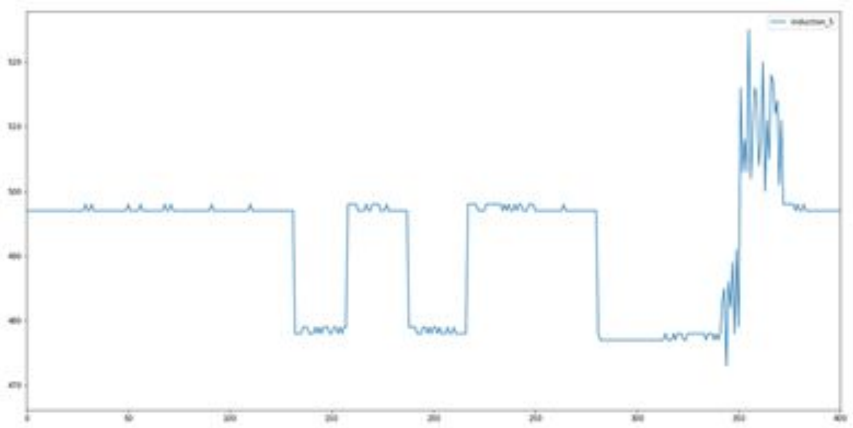
- Buttons with capacitive touch sensor
- Classical touch sense algorithm
  - Interference from induction radiation
  - Water or other contaminations on the button surface
- Can an AI algorithm detect button presses?
- Can AI make the sense algorithm more robust?

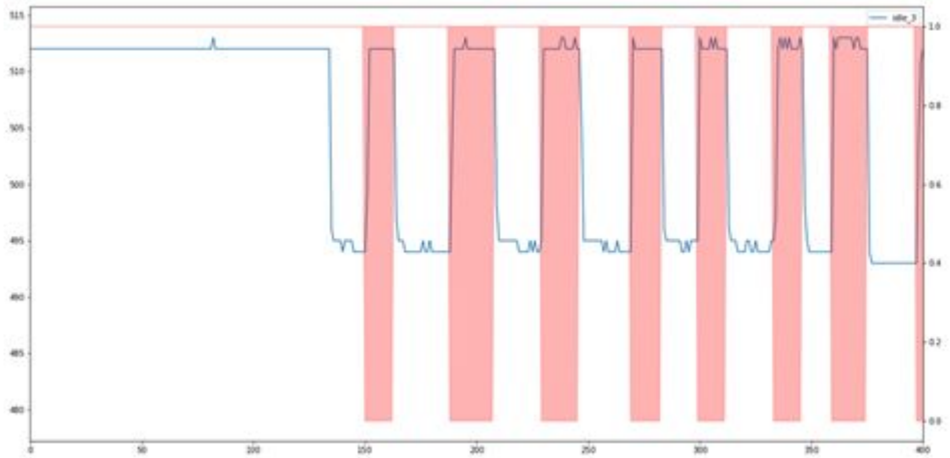
E.D.&A. - Context

```
In [6]: for dataset in datasets:
            print(dataset["name"], len(dataset["data"]))

cal 249
idle_0 7472
idle_1 7410
idle_2 7544
idle_3 7590
idle_4 7504
idle_6 7632
idle_7 7321
induction_0 7377
induction_1 7443
induction_2 7515
induction_3 7482
induction_4 7567
induction_5 7540
induction_6 7624
induction_7 7610
induction_8 7331
power_0 11167
val_0 12504
val_1 12557
val_2 22285
val_3 22438
```

- Intern @ ED&A collected data in 2019 of different situations
  - Automatic mechanical finger to label samples
  - Data collected with different induction heater settings and water levels on the buttons
  - Collected idle data (no touches)
  - Collected button presses
  - 201162 samples @ 13Hz => +4 hours

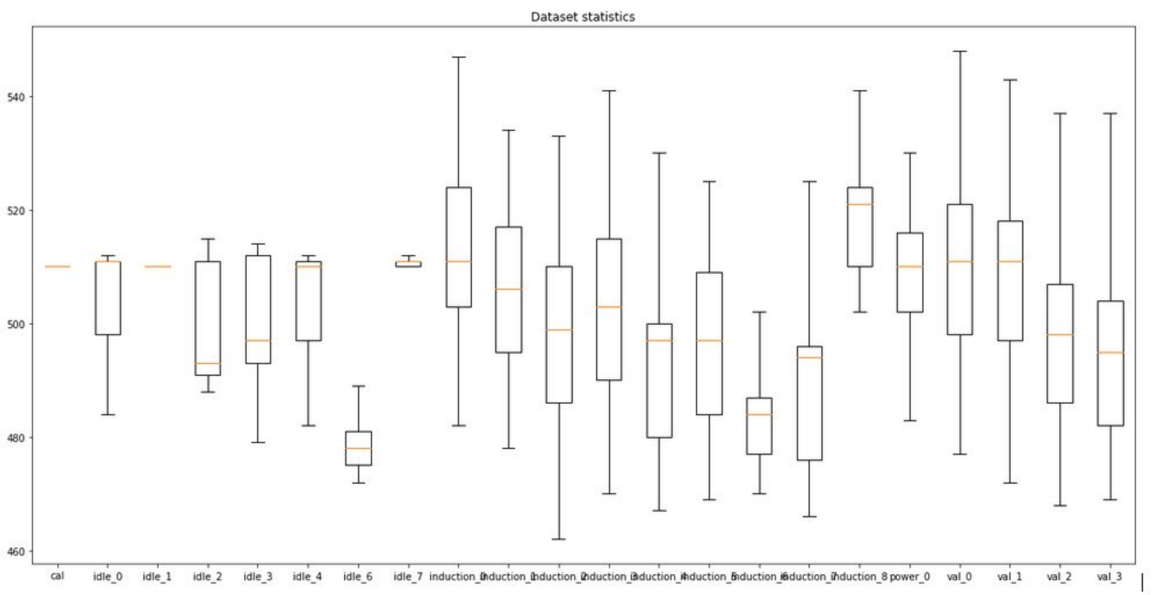




E.D.&A.
Data analyses

- Parsed and preprocessed original data which consists out of split .txt log files captured with Putty (UART)

- Statistical analysis on the data
  - Did not reveal any useful information or insights into the dataset



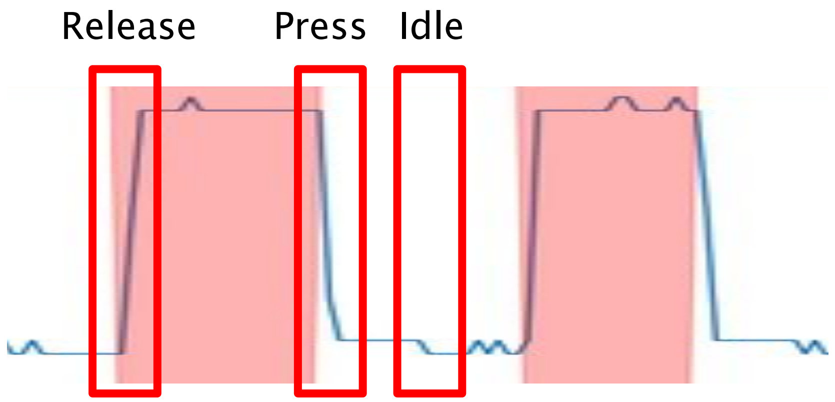
classical algorithm output
timestamp
sensor value left button
sensor value right button

```
E,990,425,514
E,031,425,513
E,073,425,514
E,115,425,514
G,155,425,513
G,195,425,514
G,236,425,514
G,275,425,514
G,315,425,514
G,355,425,512
G,395,426,514
G,435,425,513
G,475,425,513
```

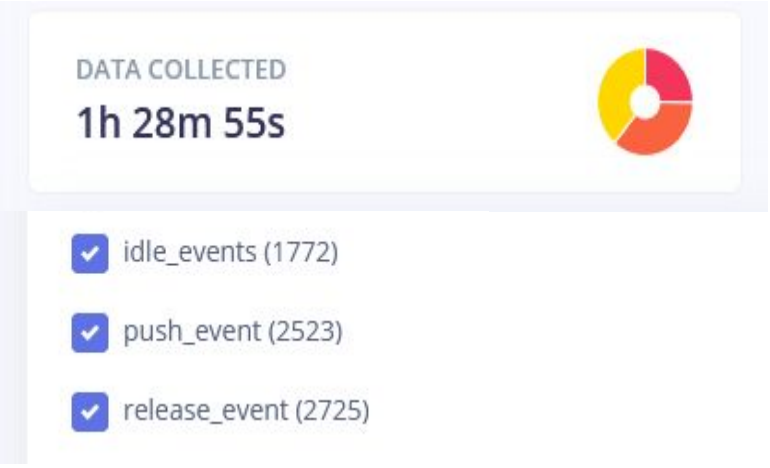
```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

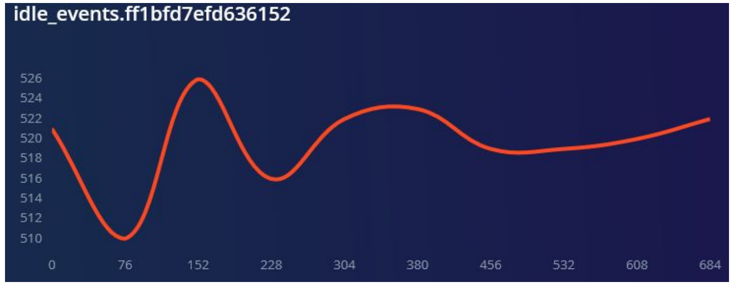
mechanical finger activation

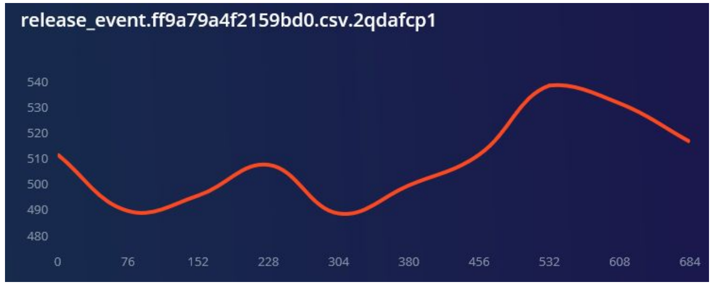
E.D.&A. - Datasets

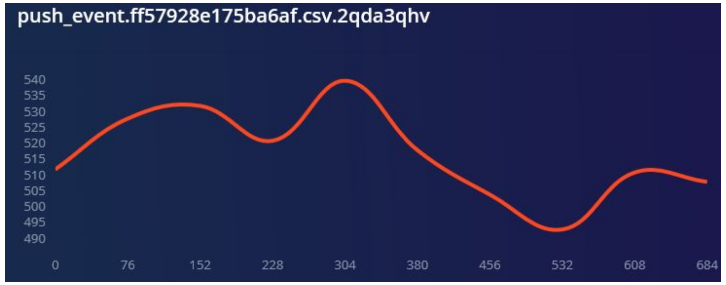


- Instead, focussed on the state changes of the mechanical finger labels
  - 1) Rising edge (release event)
  - 2) Falling edge (press event)
  - 3) Steady state (idle)
- Python script to detect events
  - Sliced 10 sensor data samples around each event
  - Corrected timestamps to Edge Impulse format
  - Each event saved to a separate CSV file

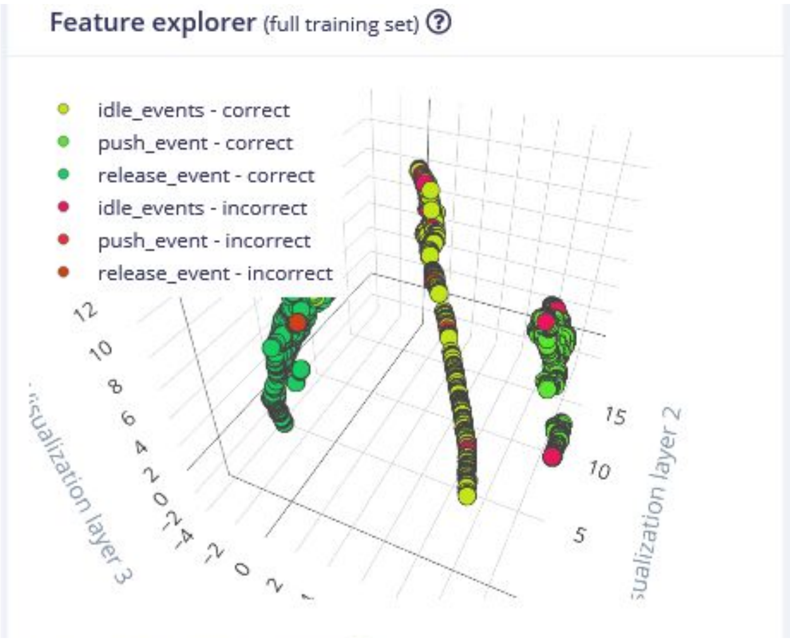


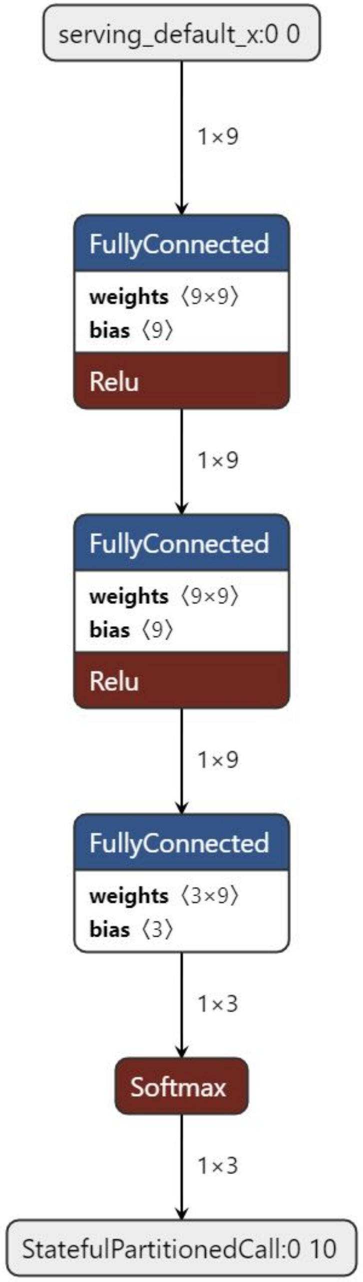






E.D.&A.
Edge Impulse





Model Model version: ? Quantized (int8)

Last training performance (validation set)

ACCURACY 97.2%

LOSS 0,12

Confusion matrix (validation set)

	IDLE_EVENTS	PUSH_EVENT	RELEASE_EVEN
IDLE_EVENTS	94.7%	3.3%	1.9%
PUSH_EVENT	0.6%	99.4%	0%
RELEASE_EVEN	3.2%	0%	96.8%
F1 SCORE	0.94	0.98	0.98

On-device performance ?

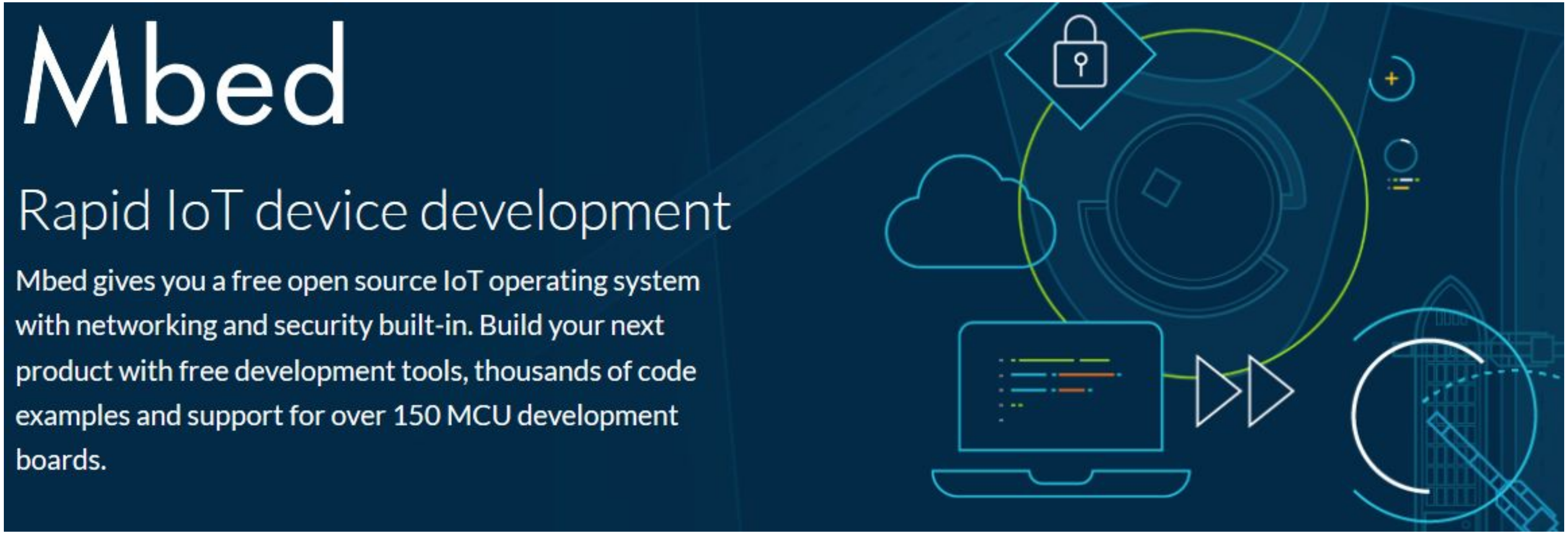
INFERENCIN... 1 ms.

PEAK RAM U... 1,7K

FLASH USAGE 18,5K

hogeschool vives KU LEUVEN AI@EDGE

E.D.&A. Mbed OS implementation



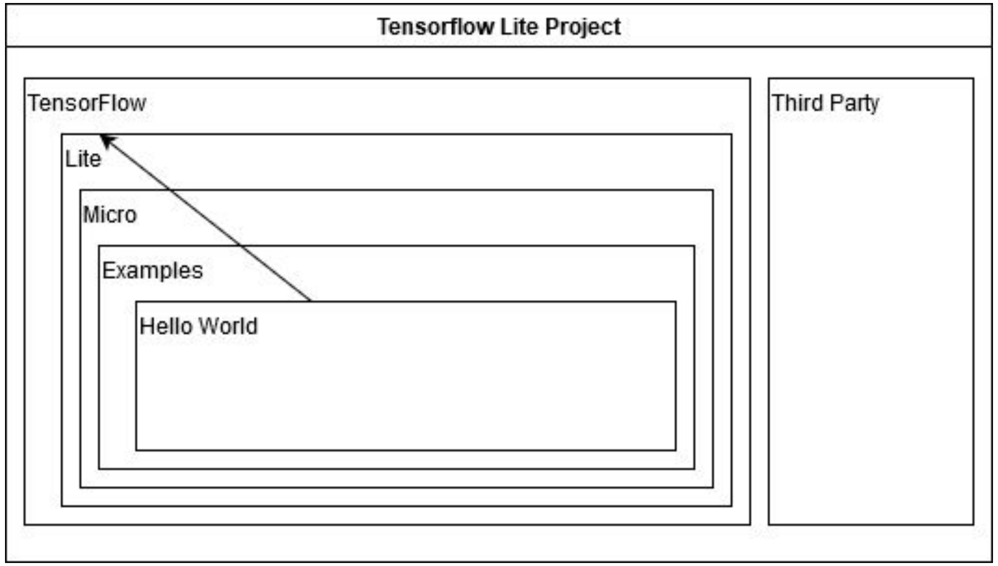
Mbed OS: RTOS or baremetal

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

TensorFlow lite micro for Mbed ecosystem

TensorFlow generator tool using make

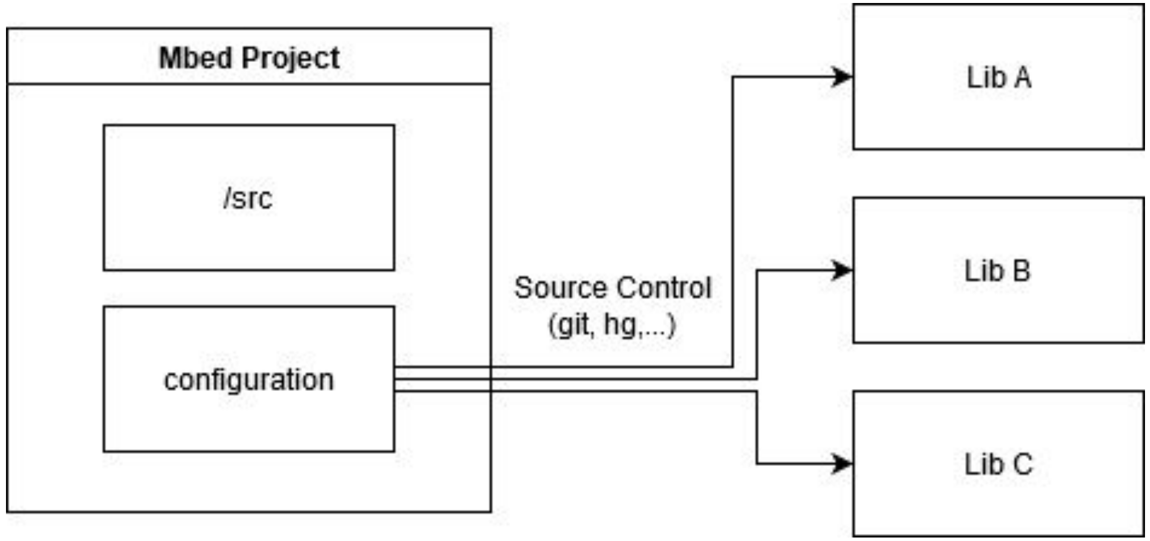
- Inside out project structure
- Applications lives inside the TensorFlow project
- Hard to update or extend
- Hard to implement in existing projects
- Enforces to use Google TensorFlow design style/rules



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

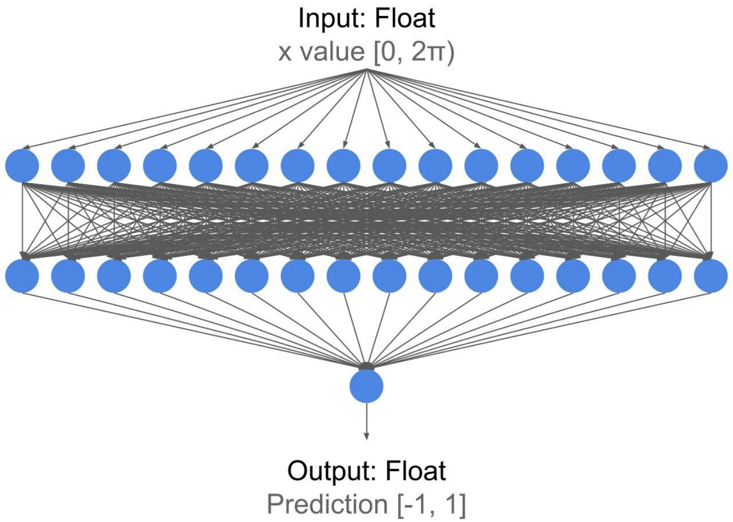
Generate new projects on windows

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

Mbed benchmark- Hello World

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



mbed benchmark targets

mbed-os (v6.6.0) with mbed-cli

GCC (v9.3.1)

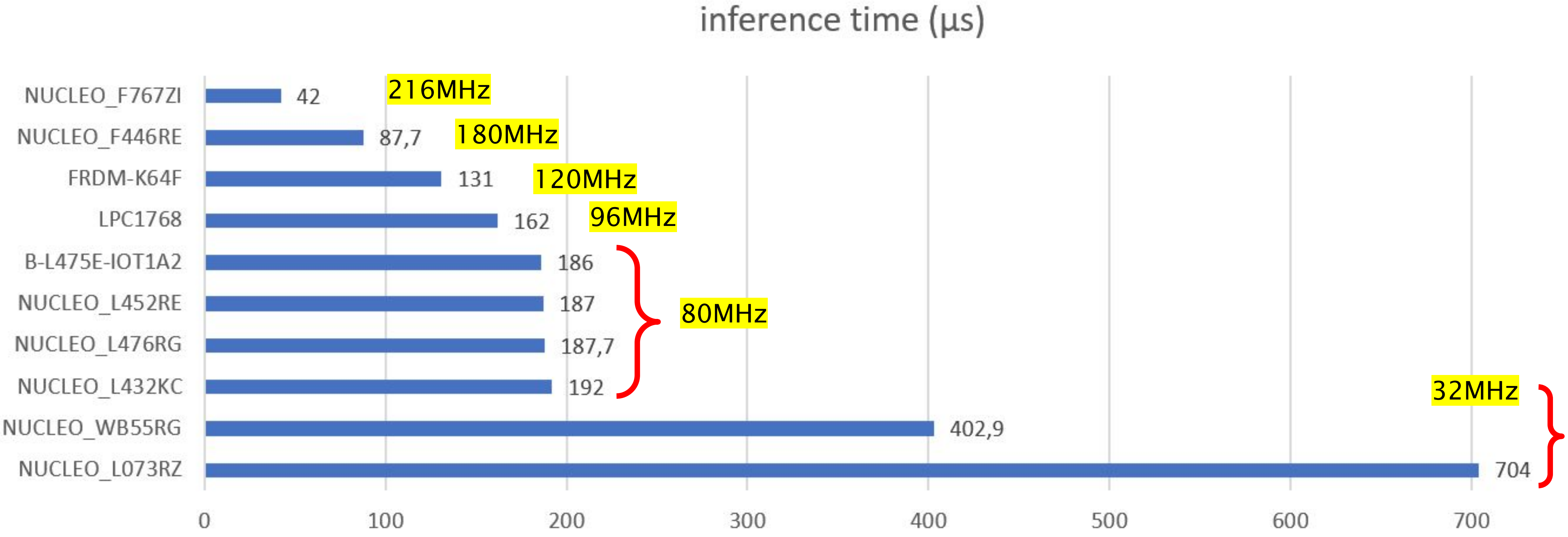
1000 iterations

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

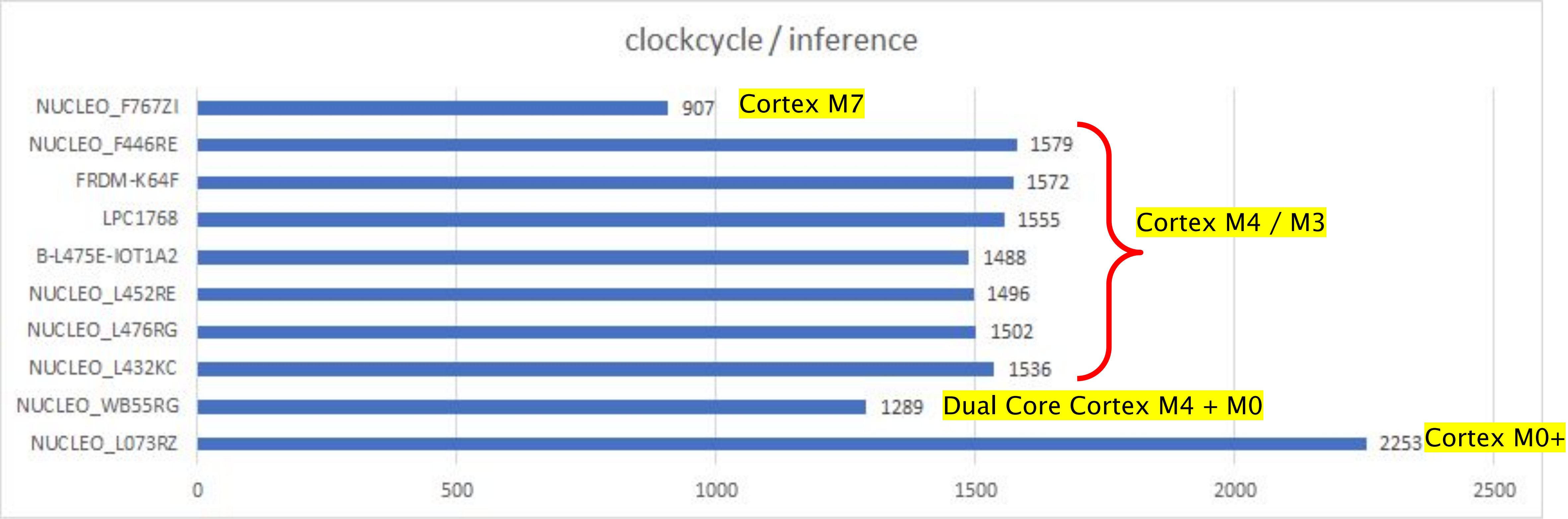
TF Lite mbed - benchmark results

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

TF lite mbed - inference times



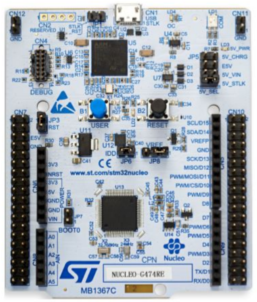
TF Lite mbed - cpu speeds



E.D.&A. Implementation

Test setup

- Target:
  STM32L476
  Cortex-M4 @ 80MHz
  1 MB Flash
  128 KB SRAM
- Tensorflow Lite for microcontrollers



UART

sensor value | classical algorithm | NN output | result

COM11 - PuTTY

```
[492](E:   released) push:   0, release:  99, idle:   0 release <---
[492](E:   released) push:   0, release:  99, idle:   0 release <---
[492](E:   released) push:   0, release:  97, idle:   2 release <---
[492](E:   released) push:   0, release:  91, idle:   8 release <---
[469](E:   released) push:   0, release:   1, idle:  98
[467](E:   released) push:  97, release:   0, idle:   2 ---> press
[467](E:   released) push:  99, release:   0, idle:   0 ---> press
[475](E:   released) push:  99, release:   0, idle:   0 ---> press
[491](E:   released) push:  50, release:   0, idle:  50
[492](E:   released) push:   0, release:  64, idle:  35
[492](E:   released) push:   0, release:  99, idle:   0 release <---
[492](E:   released) push:   0, release:  94, idle:   5 release <---
```

E.D.&A. - Optimizations

Benchmark:

Cortex Family	**Inference time**
Cortex-M4 @ 80 MHz	1.04 ms
Cortex-M0 @ 32 MHz	3.96 ms

- Tensorflow OpsResolver optimization needed to fit microcontroller flash

```
Elf2Bin: induction-touch-ai-mbed
| Module                                  |       .text |     .data |      .bss |
|-----------------------------------------|-------------|-----------|-----------|
| [fill]                                  |     246(-4) |     9(+0) |    29(+0) |
| [lib]\c.a                               |   52044(+0) |  2474(+0) |    58(+0) |
| [lib]\gcc.a                             |    3448(+0) |     0(+0) |     0(+0) |
| [lib]\m.a                               |    8100(+0) |     1(+0) |     0(+0) |
| [lib]\misc                              |     188(+0) |     4(+0) |    28(+0) |
| [lib]\nosys.a                           |      32(+0) |     0(+0) |     0(+0) |
| [lib]\stdc++.a                          |      40(+0) |     0(+0) |    16(+0) |
| mbed-os\drivers                         |    3318(+0) |     0(+0) |     0(+0) |
| mbed-os\hal                             |    1366(+0) |     8(+0) |   114(+0) |
| mbed-os\platform                        |    4602(+0) |   260(+0) |  1193(+0) |
| mbed-os\targets                         |   14412(+0) |     8(+0) |  1156(+0) |
| src\lib                                 |      82(+0) |     0(+0) |     0(+0) |
| src\main.o                              |    1226(+4) |     0(+0) | 10518(+0) |
| tensorflow-lite-micro-mbed\tensorflow   |  111100(+0) |    36(+0) |     8(+0) |
| Subtotals                               |  200204(+0) |  2800(+0) | 13120(+0) |
Total Static RAM memory (data + bss): 15920(+0) bytes
Total Flash memory (text + data): 203004(+0) bytes
```

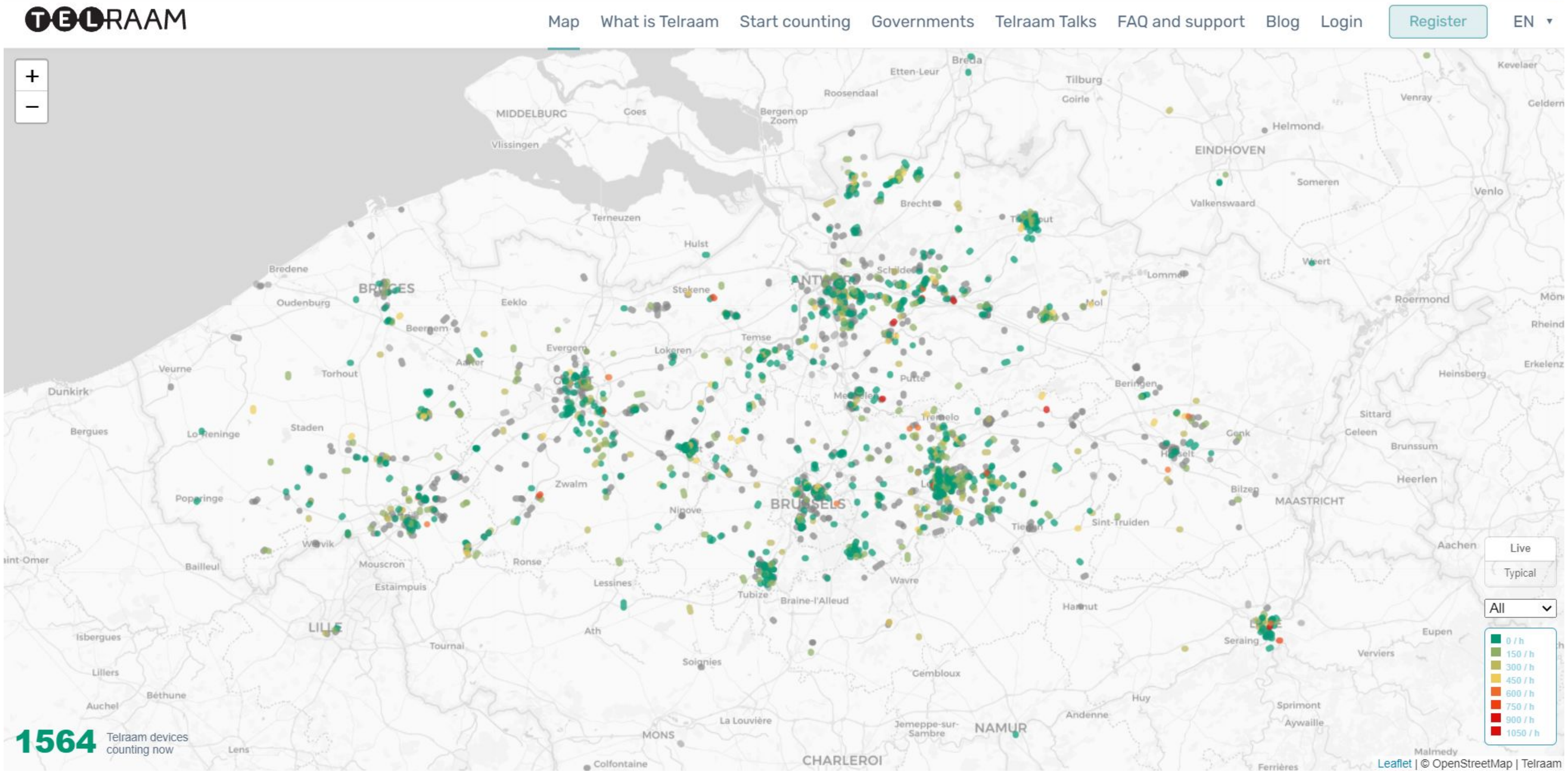
Loading custom resolvers

Totals reduced by 50 %
TF-Lite reduced by 85 %

```
Elf2Bin: induction-touch-ai-mbed
| Module                                  |        .text |     .data |      .bss |
|-----------------------------------------|--------------|-----------|-----------|
| [fill]                                  |       94(+4) |     5(-1) |    32(+0) |
| [lib]\c.a                               |    51904(+0) |  2474(+0) |    58(+0) |
| [lib]\gcc.a                             |   13072(+28) |     0(+0) |     0(+0) |
| [lib]\m.a                               | 1384(+1012)  |     1(+1) |     0(+0) |
| [lib]\misc                              |      200(+0) |     4(+0) |    28(+0) |
| [lib]\nosys.a                           |       32(+0) |     0(+0) |     0(+0) |
| [lib]\stdc++.a                          |       44(+0) |     0(+0) |    16(+0) |
| mbed-os\drivers                         |     3642(+0) |     0(+0) |     0(+0) |
| mbed-os\hal                             |     1402(+0) |     8(+0) |   114(+0) |
| mbed-os\platform                        |     4456(+0) |   260(+0) |  1252(+0) |
| mbed-os\targets                         |    11162(+0) |     8(+0) |   978(+0) |
| src\lib                                 |       94(+0) |     0(+0) |     0(+0) |
| src\main.o                              |    1624(+24) |     0(+0) |  5930(+0) |
| tensorflow-lite-micro-mbed\tensorflow   |  16806(+4380) |     0(+0) |     8(+0) |
| Subtotals                               | 105916(+5448) |  2760(+0) |  8416(+0) |
Total Static RAM memory (data + bss): 11176(+0) bytes
Total Flash memory (text + data): 108676(+5448) bytes
```

E.D.&A. - Demo

TML: Introduction



TML: Introduction

Telraam project

- Easy traffic counting
- Using Raspberry Pi with camera
- Insights about traffic density with user supplied data

How?

- Classic ML & CV:
  - Background subtraction: slow
  - Classifier for object blobs: difficult and inaccurate

TML: Use Case

Goals:

- Traffic counting at home
- Using Raspberry Pi with camera
- 2 labeled data sets available
- Detecting 5 different classes: pedestrian, bike, car, truck and other
- Frame rate of +/-5 fps

TML: Methodology

Use object detector to detect object class and location

➡ Slow (~ seconds/frame) in normal DL framework

TF Lite is perfect for low power devices!

Combine with Object Detection API

TML: Detection

Pre-trained (MS COCO) SSD+MobileNetV2 in TF

Train further on mix of data sets

160x160 resolution

Dynamic range post training quantization of weights (TF Lite default settings)

Export to TF Lite model

TML: Tracking

Passersby should only be counted **once**

→ track them!

Using motpy library

Detect when in certain “zone”

TML: Setup

Raspberry Pi 4 4GB RAM with Raspberry Pi OS

Python code, includes:

- TF Lite interpreter
- motpy tracker
- OpenCV

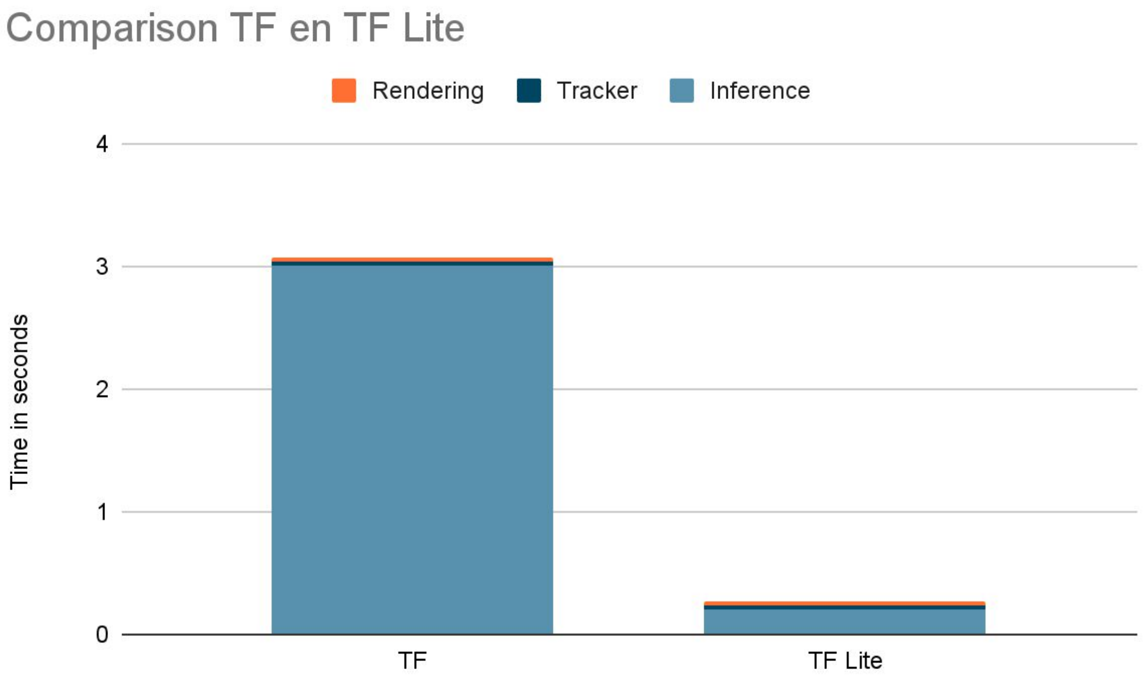
Valid .tflite model
.tflite compatible labelmap file

TML: Results

59% COCO mAP

85% PASCAL VOC mAP

Average detection: 0.2 seconds → +/- 5 fps



TML: Improvements

- More data: better generalization!
  Retrain model for better results

- In-depth optimization using TF Lite:
  various quantization strategies + more to come!

6Wolves/Yogalife

Goal: Replace the judge or physiotherapist to see if a fitness exercise has been performed in a correct way

Challenge: Using IMU's on body

Sensors provided by 6Wolves

Approach:

- Training & validation dataset using camera
- Train IMU data using visual dataset

6Wolves/Yogalife

Exercise to validate: a “normal” squat

Step 0: Annotate IMU dataset → can we automate this?

Step 1: Annotate a dataset with body keypoints

- Detection of body position using movenet
    - Other possibilities?
        - (posenet/openpose/others?)

6Wolves/Yogalife

Step 2: Convert keypoints to good/bad position

- Keypoints from movenet trained in Edge Impulse
- OR Keypoints reduced to 2D distance model

Step 3: Auto-annotate IMU data with AI model

Step 4: Inference on IMU's

- Bluetooth Low Energy
- Challenge: multiple devices

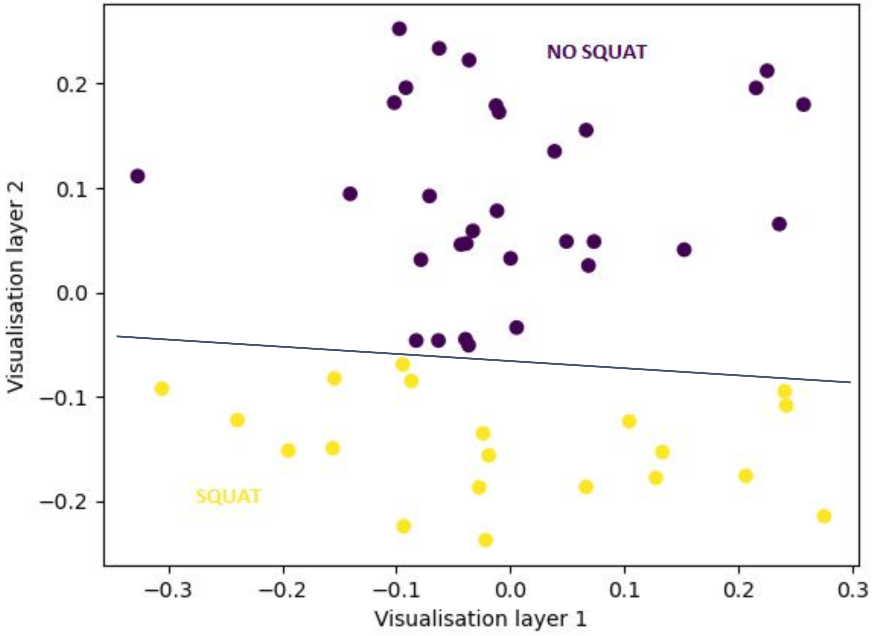
6Wolves/Yogalife

Create data trainer:

- Image to 34 keypoints translation with movenet
- Keypoints to 2D decomposition

Result: SQUAT / NO SQUAT classification

Goal: interpolation between both positions



6Wolves/Yogalife

We need more data!

Next steps:

- Capture IMU data + label with trainer

- Train Deep Learning network with dataset

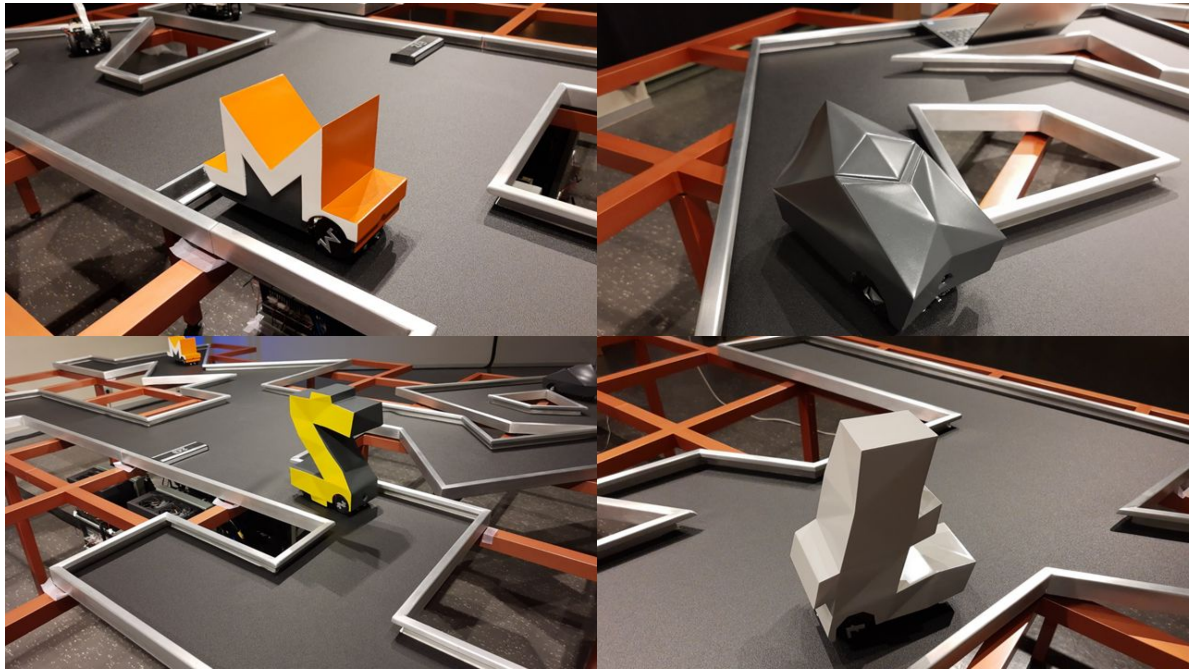
Artists Duo - LarbitsSisters

Artists Duo - LarbitsSisters

- Art exhibition project: NTAA ‘22 (Ghent)
  - New Technological Art Award
  - 836 candidates from 72 countries ⇒ 20 selected

- CMC: Crypto miner car - concept
  - Mine 4 cryptocurrencies (Ethereum, Zcash, Monero, Lite Coin)
  - Recover GPU heat ⇒ generate electricity
  - Charge 4 robots which autonomously drive a track in the form of the cryptocurrency logo

Artists Duo - LarbitsSisters



Central in the installation is a crypto mining rig with GPU units hacked to recover waste heat and fuel little electric cars, whilst crypto-currency is being mined.

The experimental work explores the shifting nature of the digital economy in the light of the ecological and social crisis. It presents a prototype for wealth redistribution that confronts todays technological and environmental challenges with disruptive thoughts on an alternative vision for the use of energy.

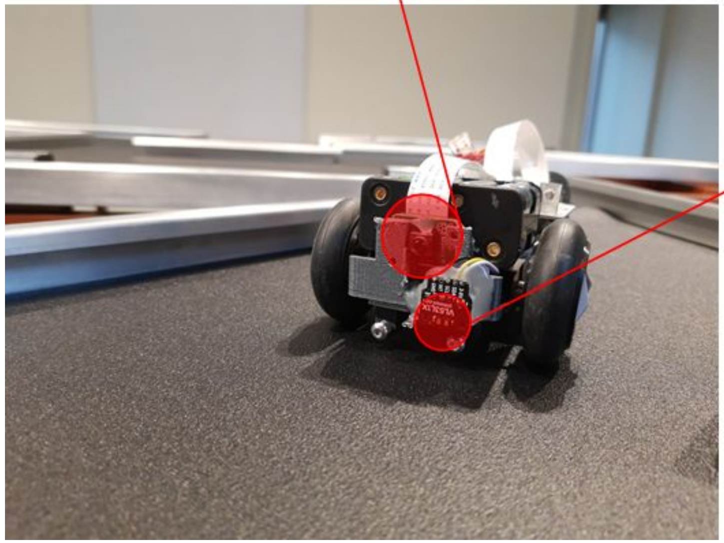
The car, once status symbol of modernity, acts here as a visionary trigger probing possible visions of the future between reality and fiction. The CMC brings a car that moves towards a new and disruptive form of mobility. Within the critical discourse on climate change, CO_2 emissions and global warming, it explores how the computational process and massive computing power involved in the mining process of crypto-currencies can be deployed in the urban and social fabric.

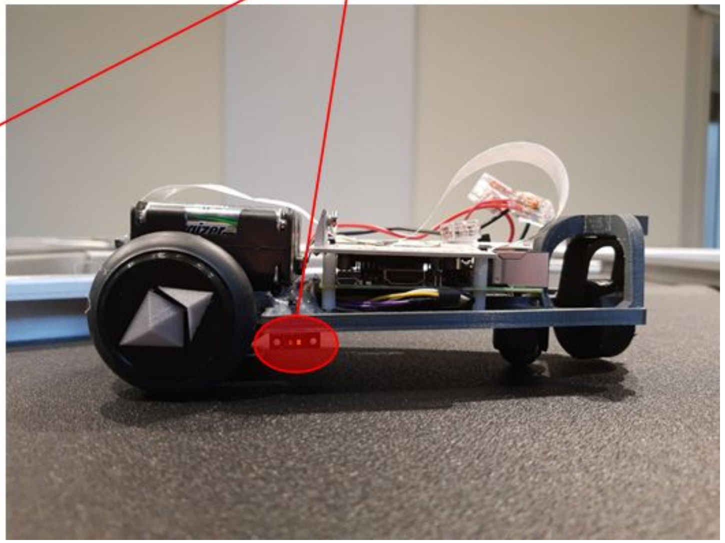
Artists Duo - LarbitsSisters

- Cars should drive (and charge) autonomously, with an AI learning-based component, computation on RPi3

Raspberry Pi Camera (640 x 480 resolution)

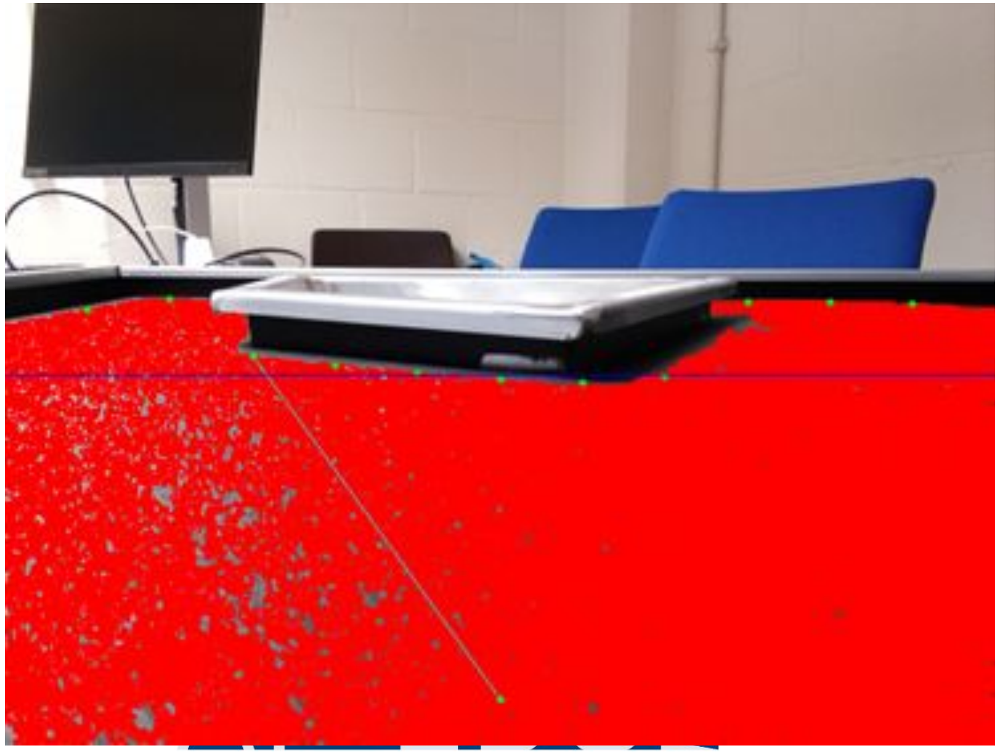
Three distance sensors (TOF, VL53L1X): Front, left, right





Artists Duo - LarbitsSisters

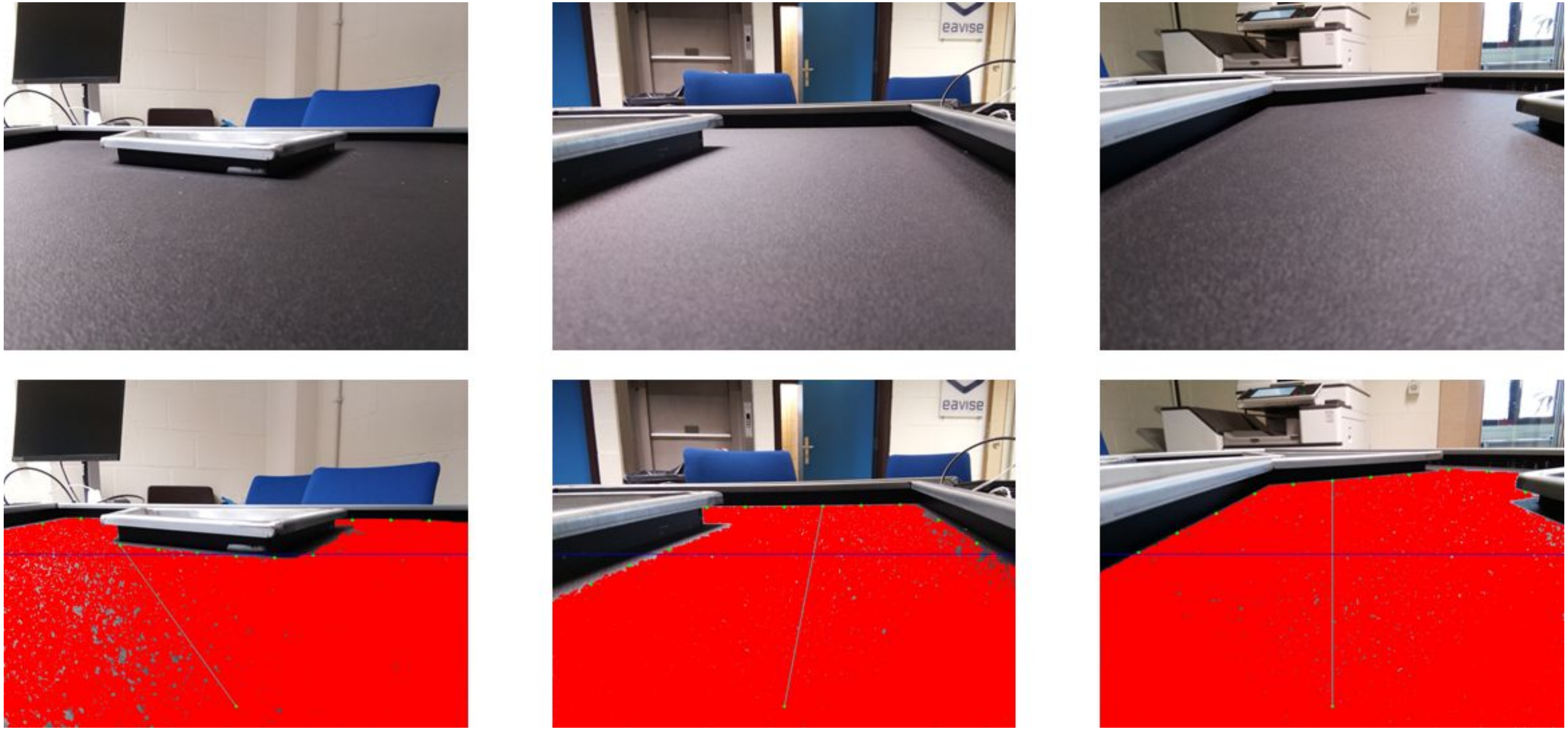
- Vision algorithm:
  - Segment track based on Floodfill algorithm
  - Divide image in 11 equidistant segments
  - For each segment, find furthest point in segmented track (green dots)
  - Threshold the segments (blue line)
  - Determine largest group of points
  - Find middle of largest group as best direction



- Output vector example:

  [0 0 1 0 0 0 0 0 0 0 0]

Artists Duo - LarbitsSisters



Artists Duo - LarbitsSisters

- AI component: Q-learning
  - Output vector is used as input for a Q Learning reinforcement algorithm
  - Model free
  - Q Learning determines best action: rotate left, rotate right or move forward
  - Trained in simulation for 1000 actions

$$Q^{new}(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t,a_t)}_{\text{old value}} \bigg)}^{\text{temporal difference}}$$

Initialized

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	327	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	0	0	0	0	0	0

Training

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

- The vision output determines the driving direction

Images: https://en.wikipedia.org/wiki/Q-learning

Artists Duo - LarbitsSisters

- Three distance sensors complement vision
  - Viewing-angle of RPi-cam too small
  - Type: Time-Of-Flight (TOF) – VLX53L1X

- Implementation:
  - L & R distance sensors used to slightly correct forward maneuver to stay in the middle of the track (5% speed correction)
  - When too close to left or right border, perform maneuver to re-center
  - Forward driving is priority; if opening left or right is seen and the front distance is small, a turn is made (random direction if possible)
  - If no visual path is found, move forward if possible
  - When too close to wall with front sensor, drive backwards

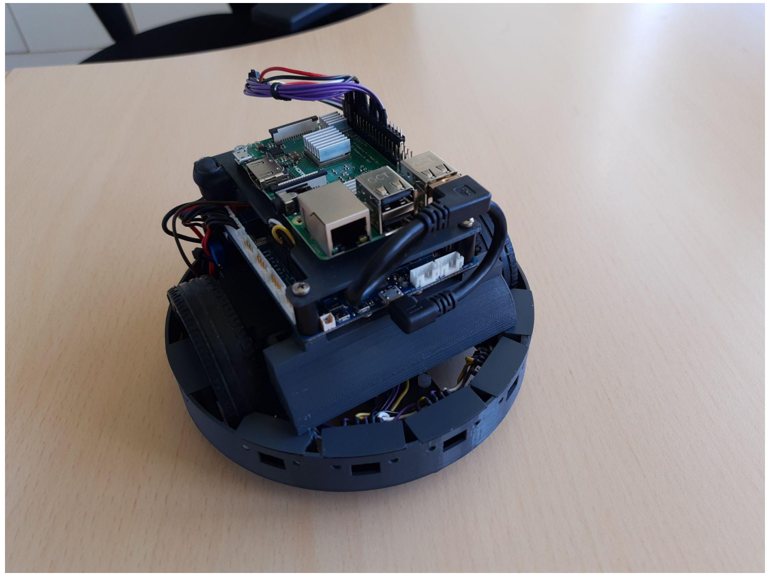
Artists Duo - LarbitsSisters

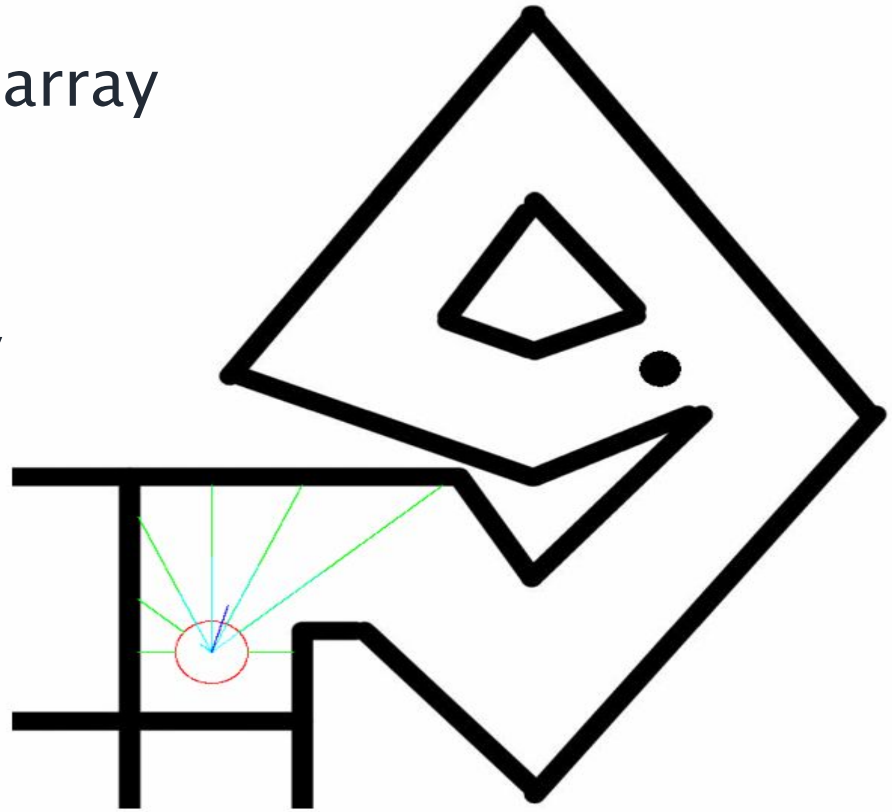
Artists Duo - LarbitsSisters

- Issues with existing robots:
  - Too large for track
  - Unable to turn 180 degrees
  - Mechanically too weak
  - Issues with illumination
  - Slow movement with pauses (intended)

Artists Duo - LarbitsSisters

- Second iteration finished
  - Uses 7 distance sensors in 180 degree array
    - Based on force-field algorithm
  - Circular chassis, more reliable motors
  - Much faster, more agile
  - Able to dock and charge autonomously





https://youtu.be/2Xhqavry95c

Artists Duo - LarbitsSisters

Short break

Take a look at the demonstrations:

Cryptominercar - AB Writing - IR people detection

Induction cooker - Automatic Garage Door

Telraam Traffic Counting - Squat detection

Industry talk

Melexis - Luc Buydens

hogeschool vives KU LEUVEN

Conclusion

Output

Demonstrations of the use-cases

Manual & best-practices

- Guidelines from workshops
- Explanation of use-cases
- Frameworks used
- Tutorials

Documentation release: September 2022

Embedded AI for Industry

Post Universitary Centrum KU Leuven
3-day Summer School (14 to 16 September)
Theory alternated with hands-on workshops

Focus on different topics, industry-oriented

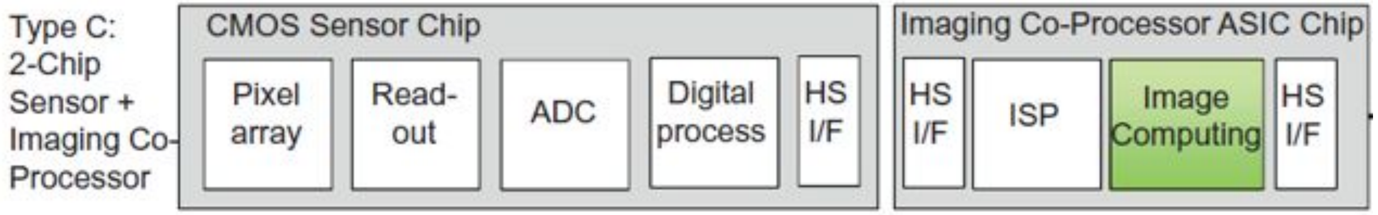
- Introduction to machine-/deep-learning
- Edge Impulse
- Model reduction, CMSIS-CNN
- Vision & quantisation
- Distributed AI
- Embedded AI for crypto-cybersecurity

New TETRA project: AI to the Source

Goal: apply AI computer vision techniques directly on raw sensor data (hyperspectral, IR, depth,...)

Several **advantages**: ultra-low latency, high bit resolutions, no data loss, inherent privacy, low data bandwidth, ...

Interested? Contact us - http://eavise.be/
Start: 01/10/'22



hogeschool vives
KU LEUVEN

Thank you!

Take a look at the demonstrations & enjoy the reception.
Any questions? Ask the AI@EDGE Team!

Toon

Kristof

Maarten

Jonas

Sille