AI@EDGE

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

VLAIO TETRA HBC.2019.2641

User group meeting 4

29-04-2022

ai-edge.be

iot-incubator.be

www.eavise.be

hogeschool vives

KU LEUVEN

Agenda

1. Introduction
2. Academic use cases
3. Industrial use cases
4. Future planning
5. Questionnaire
6. Discussion

Introduction

User Group - Updates - Workplan

User Group Members

ed&a
custom-made electronics

Melexis
INSPIRED ENGINEERING

Sensotec

PICANOL
Let's grow together

DP
TECHNICS

ScioTeq

Edgise

TRANSPORT & MOBILITY LEUVEN

6WOLVES

Q MINERAL
analysis & consulting

Duo LarbitsSisters
Larbitslab

hogeschool vives

KU LEUVEN

AI@EDGE

Project updates

- Change of project coordinator: Jonas Lannoo
- New user group member: LarbitsSisters
- Two hands-on workshops:
  - Introduction to DL + Edge Impulse
  - Vision & quantisation
- Project finalisation: end May 2022

Workplan - Progress

WP1: Exploratie (3 mm)

- WP 1.1: studie frameworks voor low-cost embedded systemen
- WP 1.2: studie optimalisatietechnieken voor Deep Learning op embedded systemen
- WP 1.3: bevraging leden begeleidingsgroep

WP2: Proof of concept (6 mm)

- WP 2.1: selectie hardware en frameworks
- WP 2.2: verzamelen en annoteren van data
- WP 2.3: implementatie
- WP 2.4: testen en validatie

WP3: Industriële gevalstudies (18 mm)

- WP 3.1: opstellen functionele en niet-functionele vereisten
- WP 3.2: selectie en operationaliseren van hardware en framework
- WP 3.3: implementatie
- WP 3.4: optimaliseren
- WP 3.5: testen en validatie

WP4: Valorisatie (9 mm)

- WP 4.1: overzicht van hardware en frameworks op website
- WP 4.2: handleiding met best-practices
- WP 4.3: hands-on workshops
- WP 4.4: wetenschappelijke publicaties
- WP 4.5: slotsymposium

User group interaction

Question/remarks so far?

Academic use-cases

AB Writing

Seat Detection

Car Detection

AB-Writing

Goal: Detection of handwritten letters/symbols/numbers

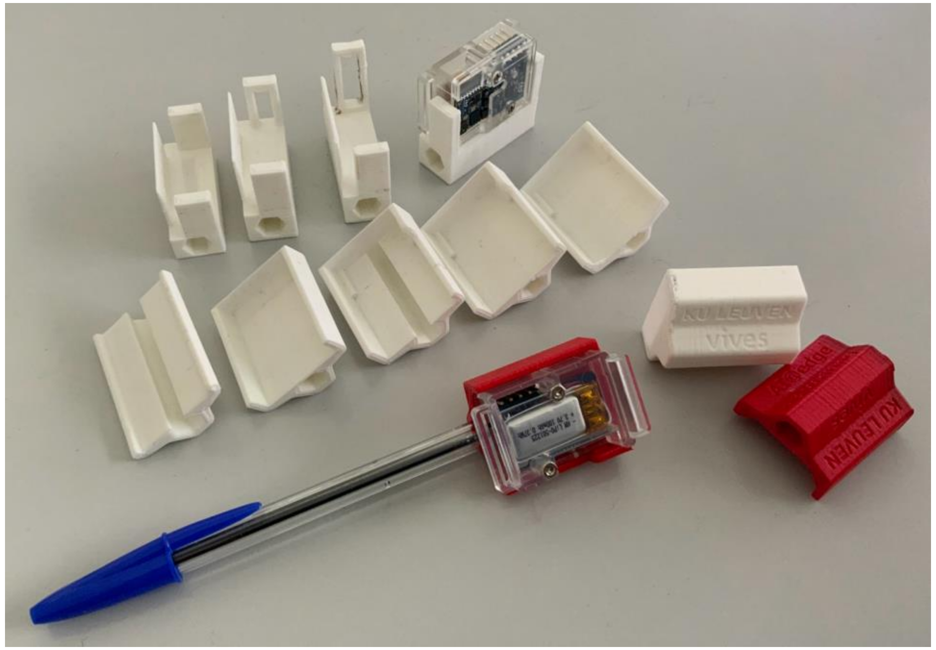
Challenges:

- Using accelerometer
- Small microcontroller

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

Approach:

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



AB-Writing

- 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

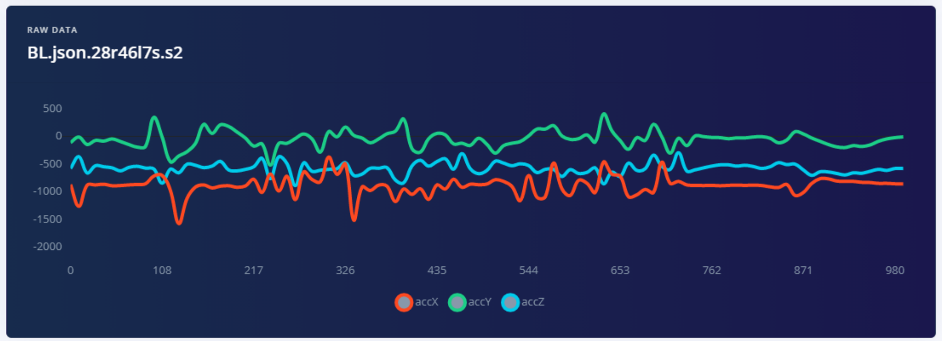
AB-Writing

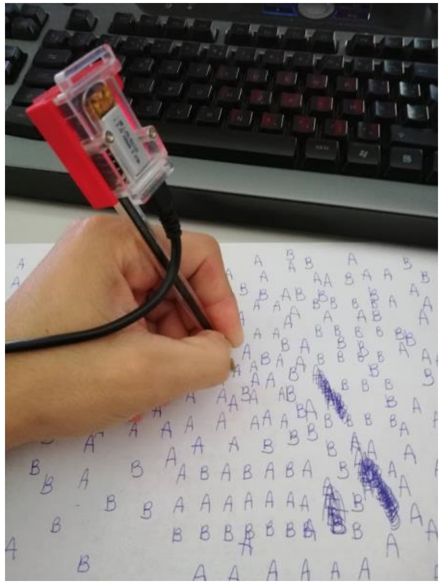
- Step 2: Data acquisition & annotation

Data from colleagues, students, workshop attendees

Dataset classes:

Sym

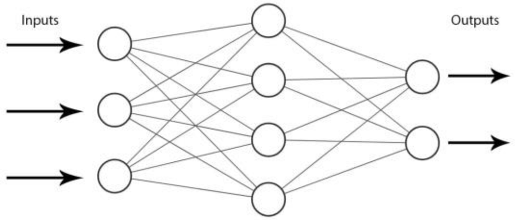




AB-Writing

EDGE IMPULSE

- Step 3: Implementation



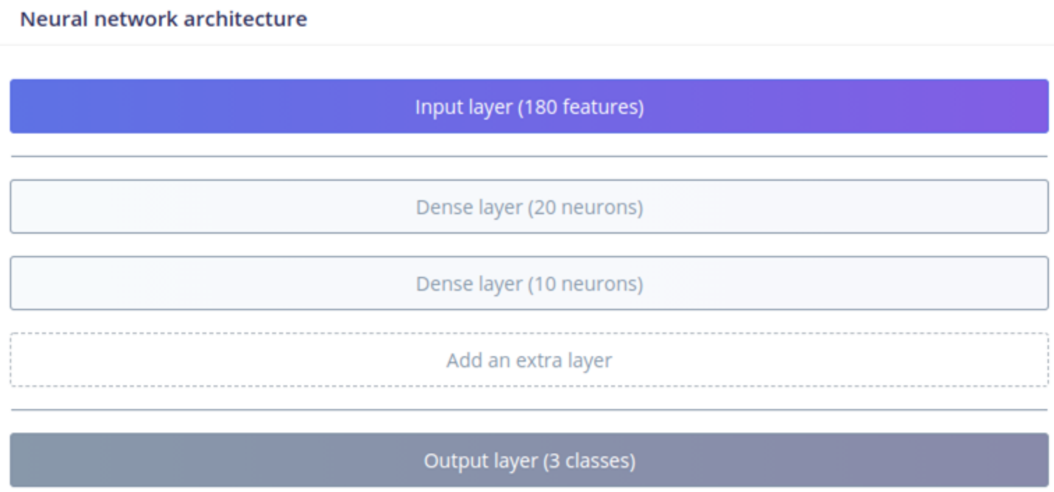
Preprocessing: none/raw (DL approach)

Analysis of network size & layers

FFNN + x hidden

Time domain inp

Online training



AB-Writing

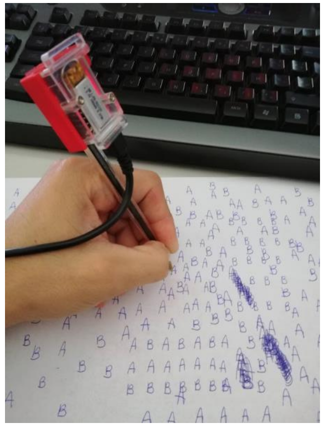
EDGE IMPULSE

- Step 4: Testing & validation

Validate model with test c

Deployment on µC

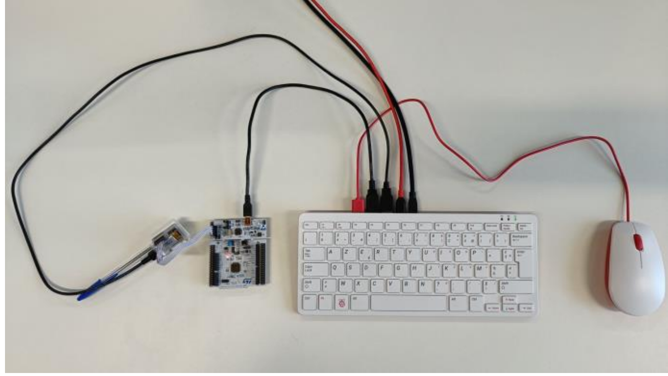
	AL	AR	BL	BR	IDLE
AL	61.1%	16.7%	16.7%	0%	5.6%
AR	0%	63.2%	5.3%	31.6%	0%
BL	9.5%	0%	81.0%	9.5%	0%
BR	5.9%	29.4%	5.9%	58.8%	0%
IDLE	0%	0%	0%	0%	100%
F1 SCORE	0.69	0.62	0.79	0.57	0.99



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 our first hands-on workshop (9 Dec)

Full workflow:

Data collection → inference

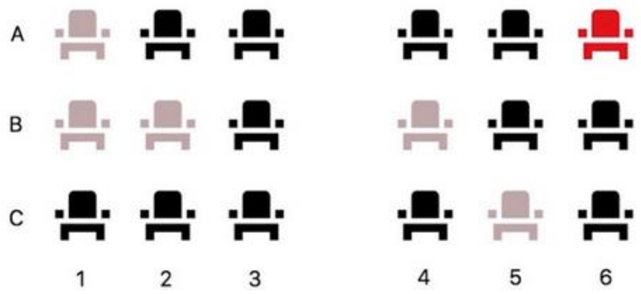
Hardware: Raspberry Pi 400 + µC

Planned:

KU Leuven students → 18/05

For industry → TBA

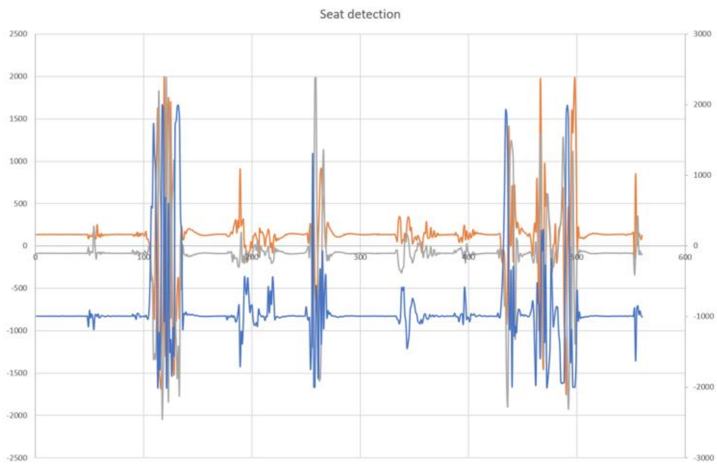
Seat Detection



Goal: Count number of people in a room

Challenges:

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



Approach:

- Small microcontroller: STM Sensortile
- Acclerometer

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

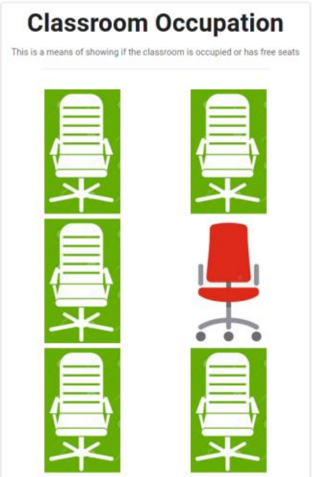
Seat Detection

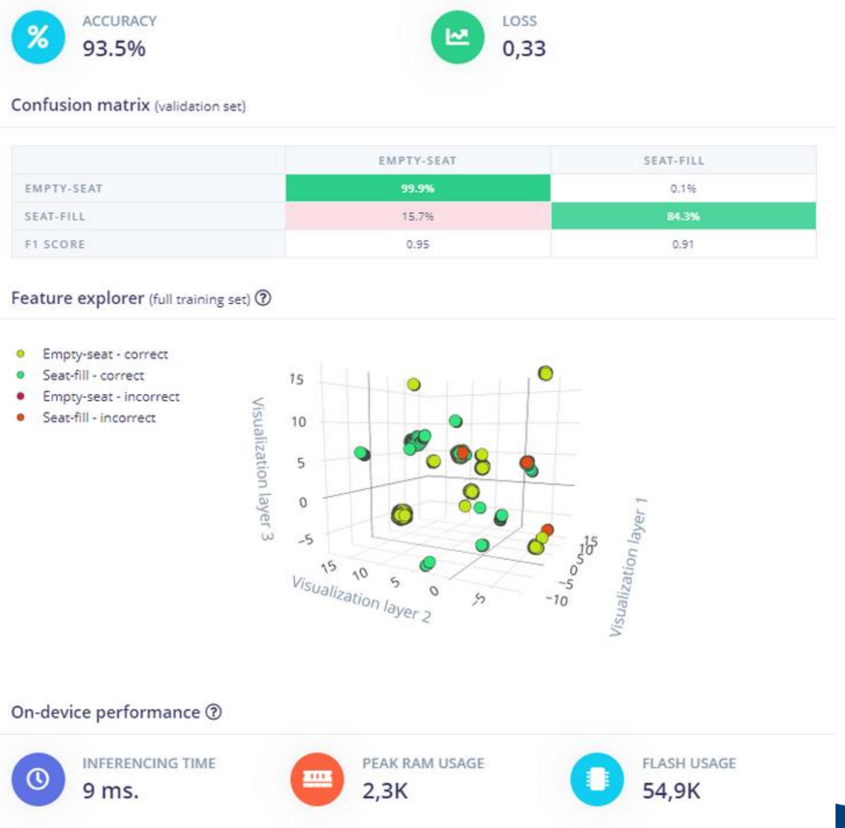
1. Static seat detection

Static => seat moving or not

Slight vibration when seated

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





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

hogeschool vives KU LEUVEN

AI@EDGE

Seat Detection

2. Dynamic seat detection

Dynamic = movement detection of seat

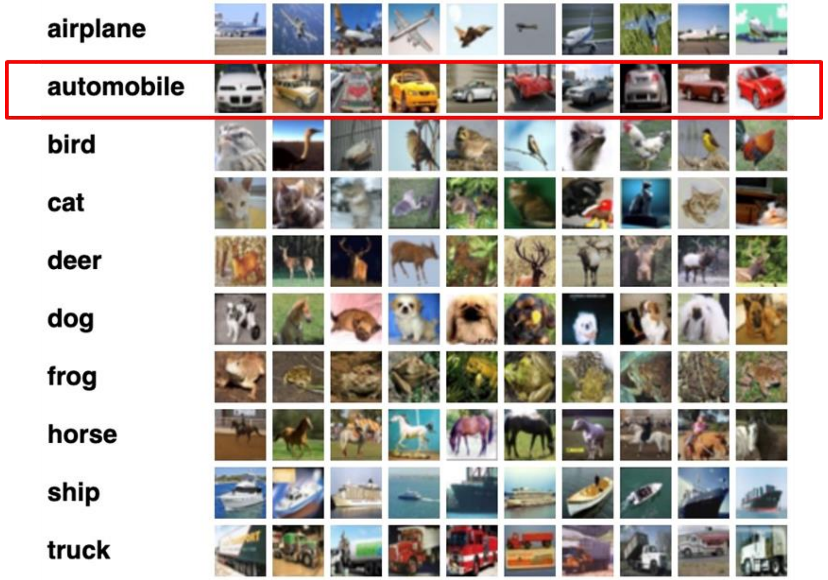
Forward & backwards sliding

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

Car Detection

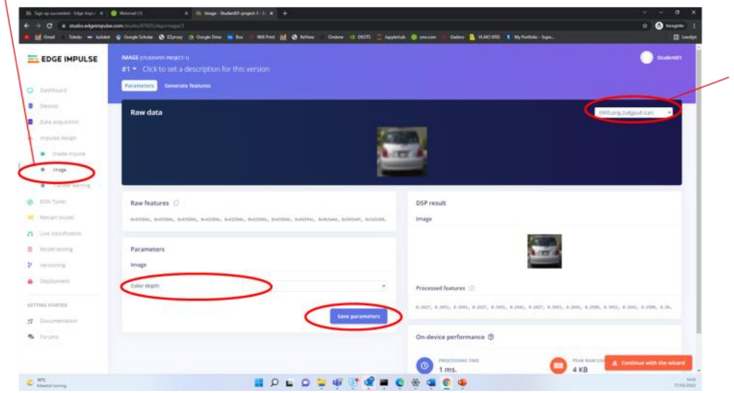
Workshop STEM (secondary school)

- 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

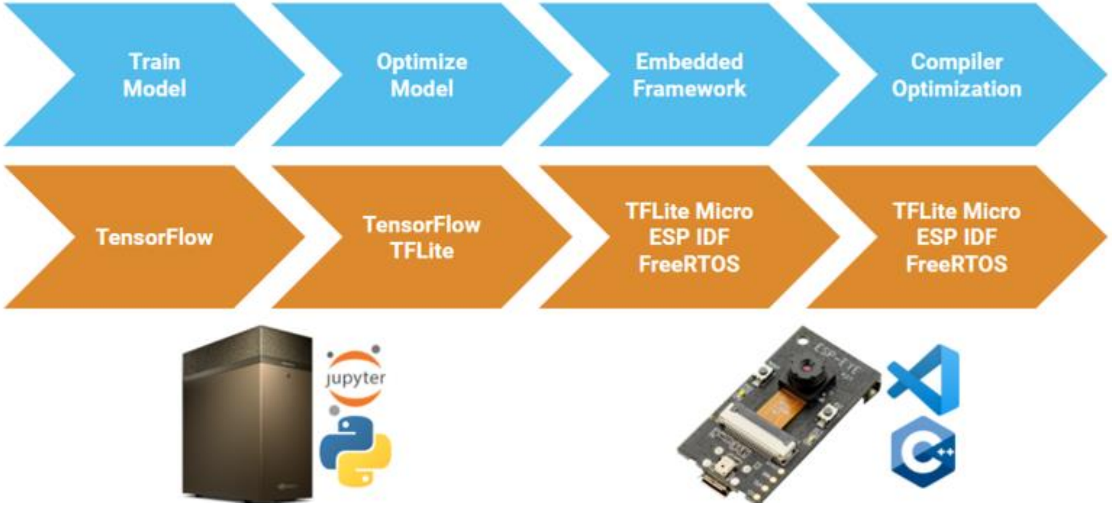
Workshop STEM (secondary school)

Workshop booked:

- 29/03/2022 AM
- 02/04/2022 AM
- 02/04/2022 PM
- 03/05/2022 AM
- …

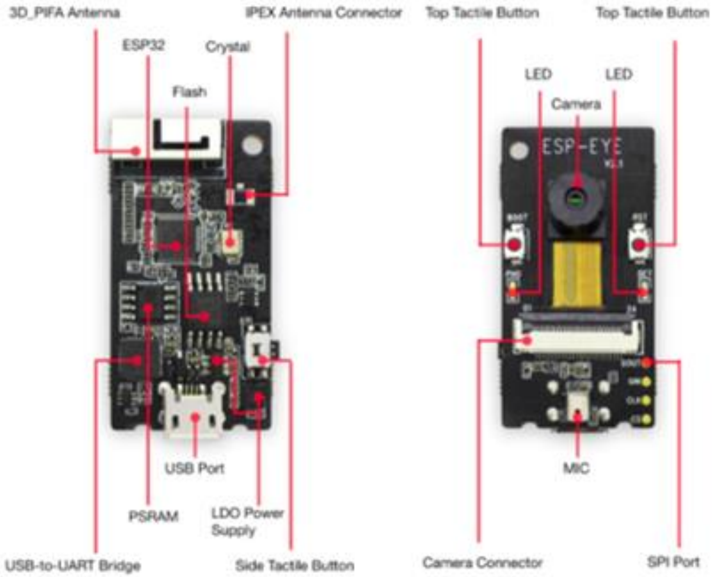
hogeschool vives KU LEUVEN AI@EDGE

Workshop Embedded AI Optimization



hands-on workshop on 22/04/2022

-> repeat?



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

User group interaction

Question/remarks so far?

Industrial use-cases

Melexis - E.D.&A. - TML - Yogalife - LarbitsSisters - Gemeente Sint-Katelijne-Waver

Melexis

See separate slide deck Maarten (pdf slides)

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

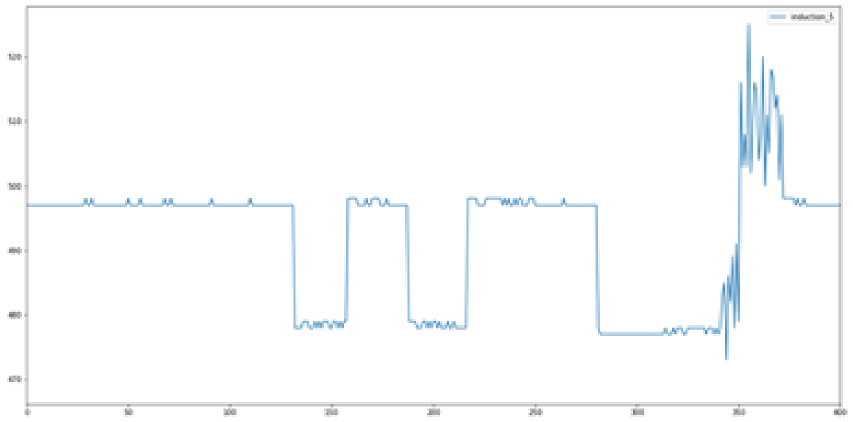
- Can an AI algorithm detect button presses?
- Can AI make the sense algorithm more robust?

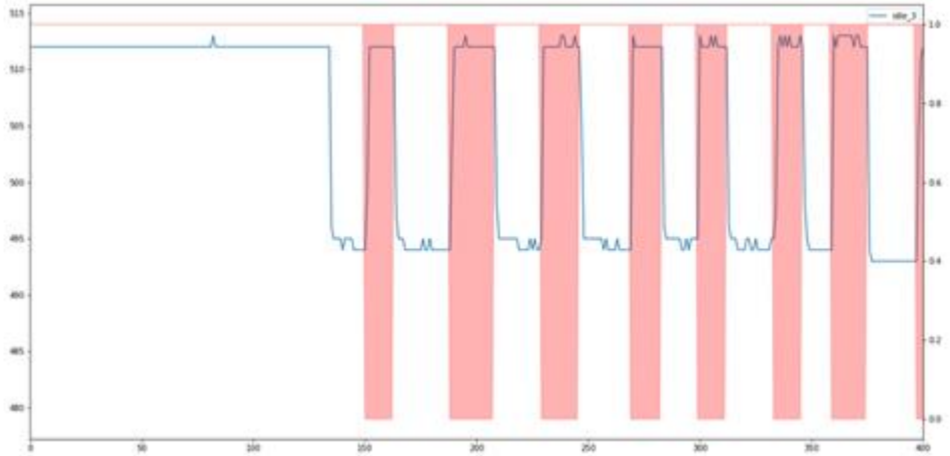
E.D.&A.

```
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

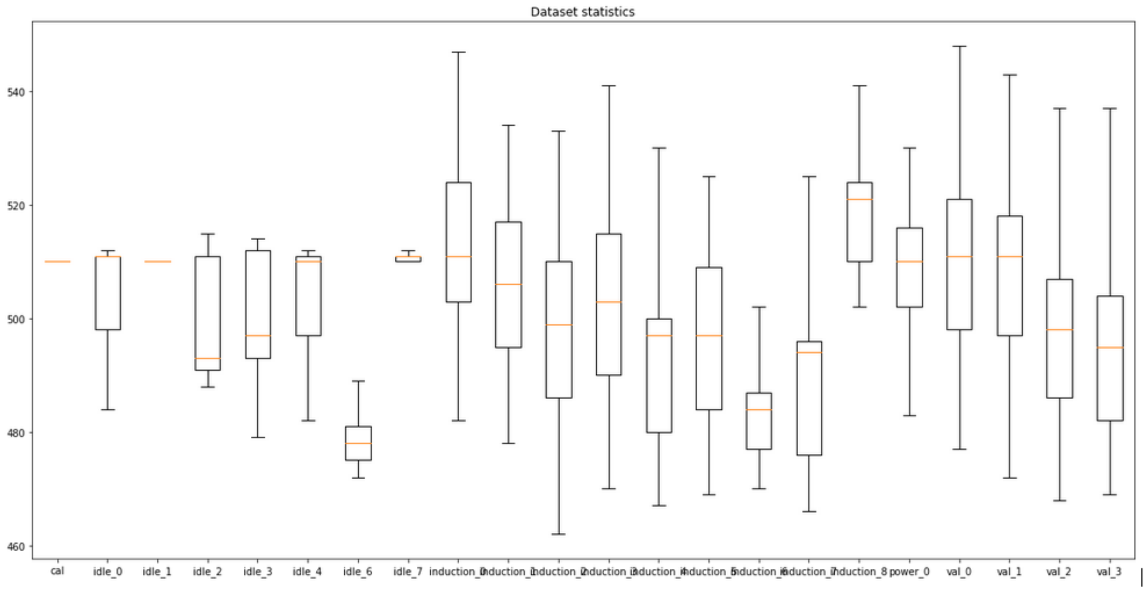




E.D.&A.

- 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



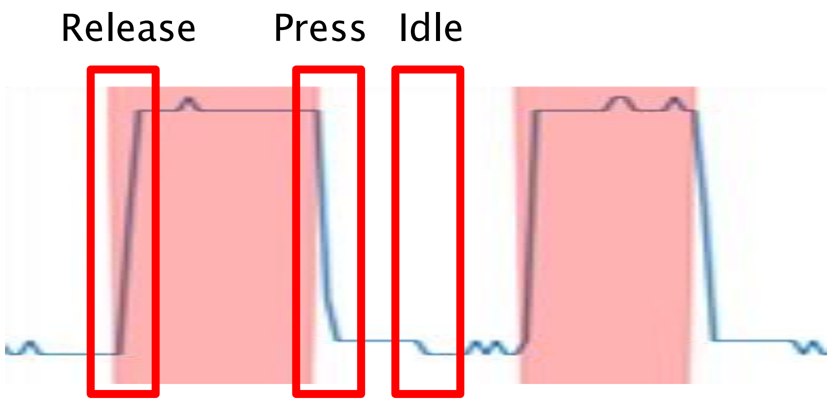

```
E,990,425,514
E,031,425,513
E,073,425,514
E,115,425,514
G,155,425,513
G,195,425,514
G,236,425,514
G,275,425,514
G,315,425,514
G,355,425,512
G,395,426,514
G,435,425,513
G,475,425,513
```

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

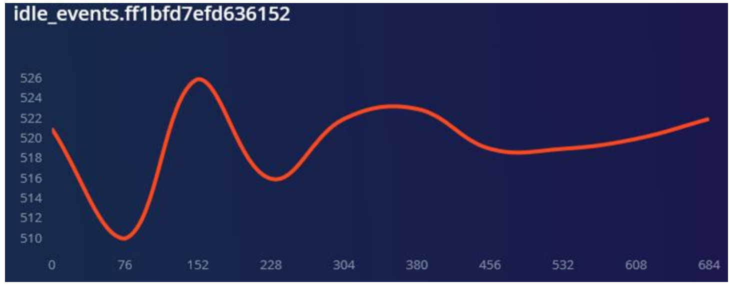
```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

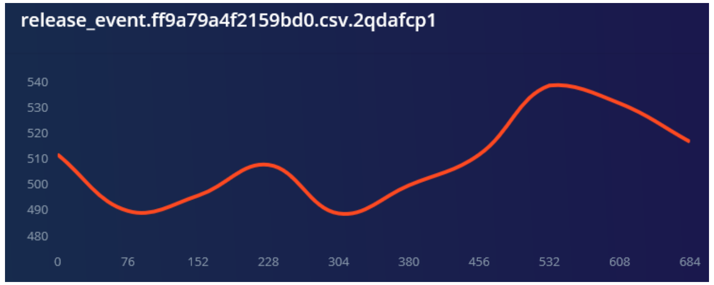
mechanical finger activation

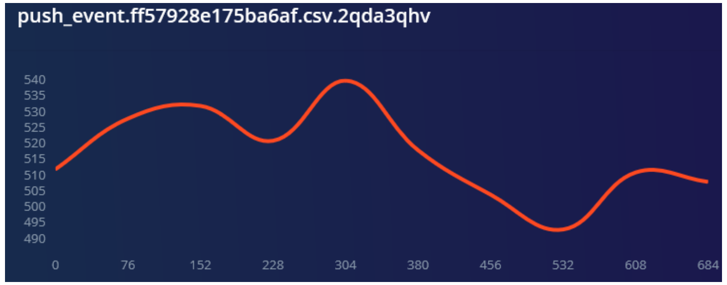
E.D.&A.



- 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

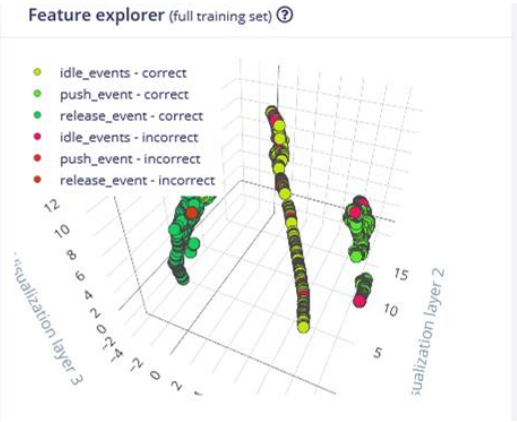






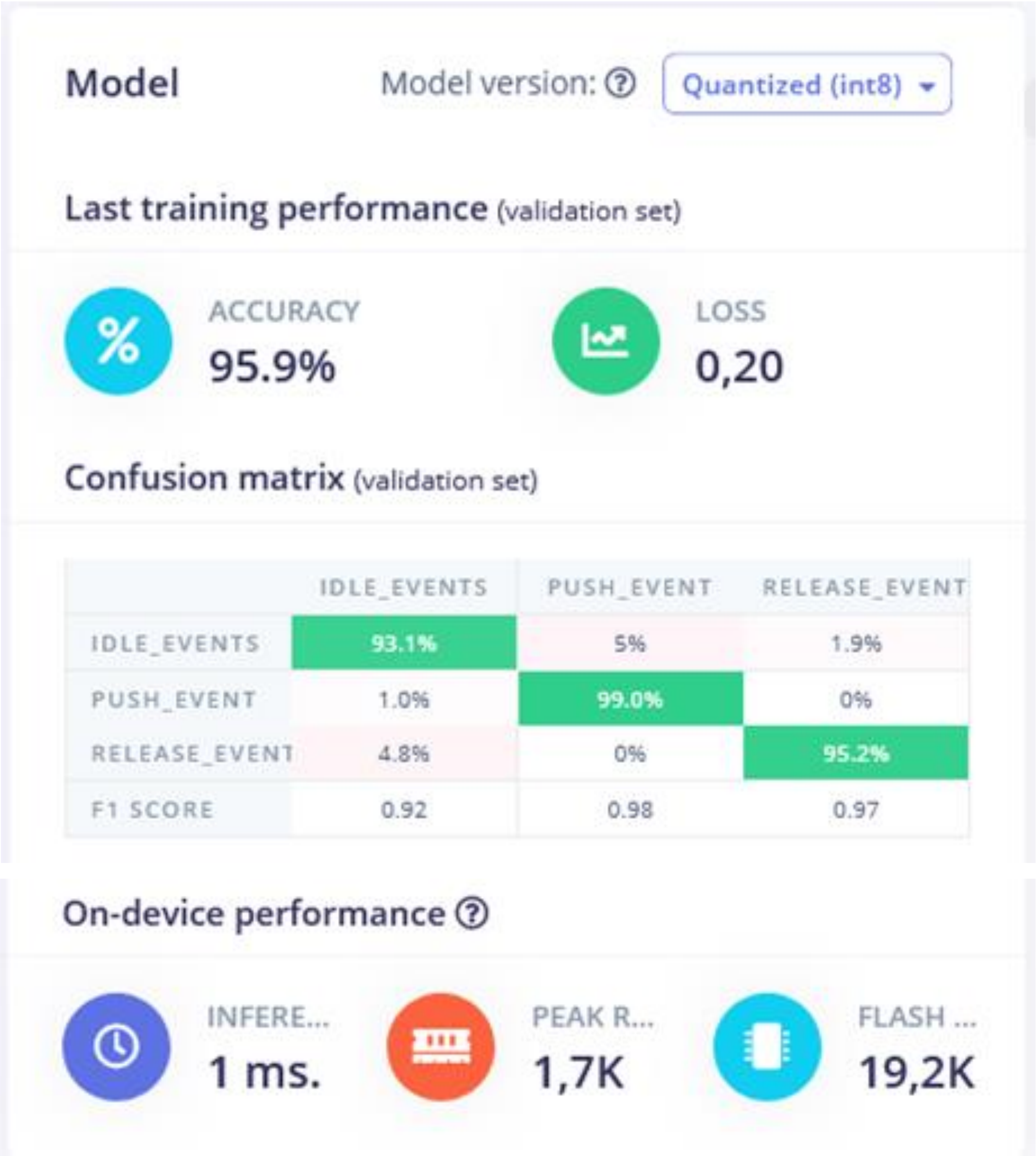
hogeschool vives KU LEUVEN AI@EDGE

E.D.&A.



Edge Impulse

- Neural network that inputs raw sensor values
  - Input layer with 9 input features
  - Dense layer with 27 neurons
  - Dense layer with 18 neurons
  - Output layer with 3 classes



E.D.&A.

UART

Test setup

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

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.

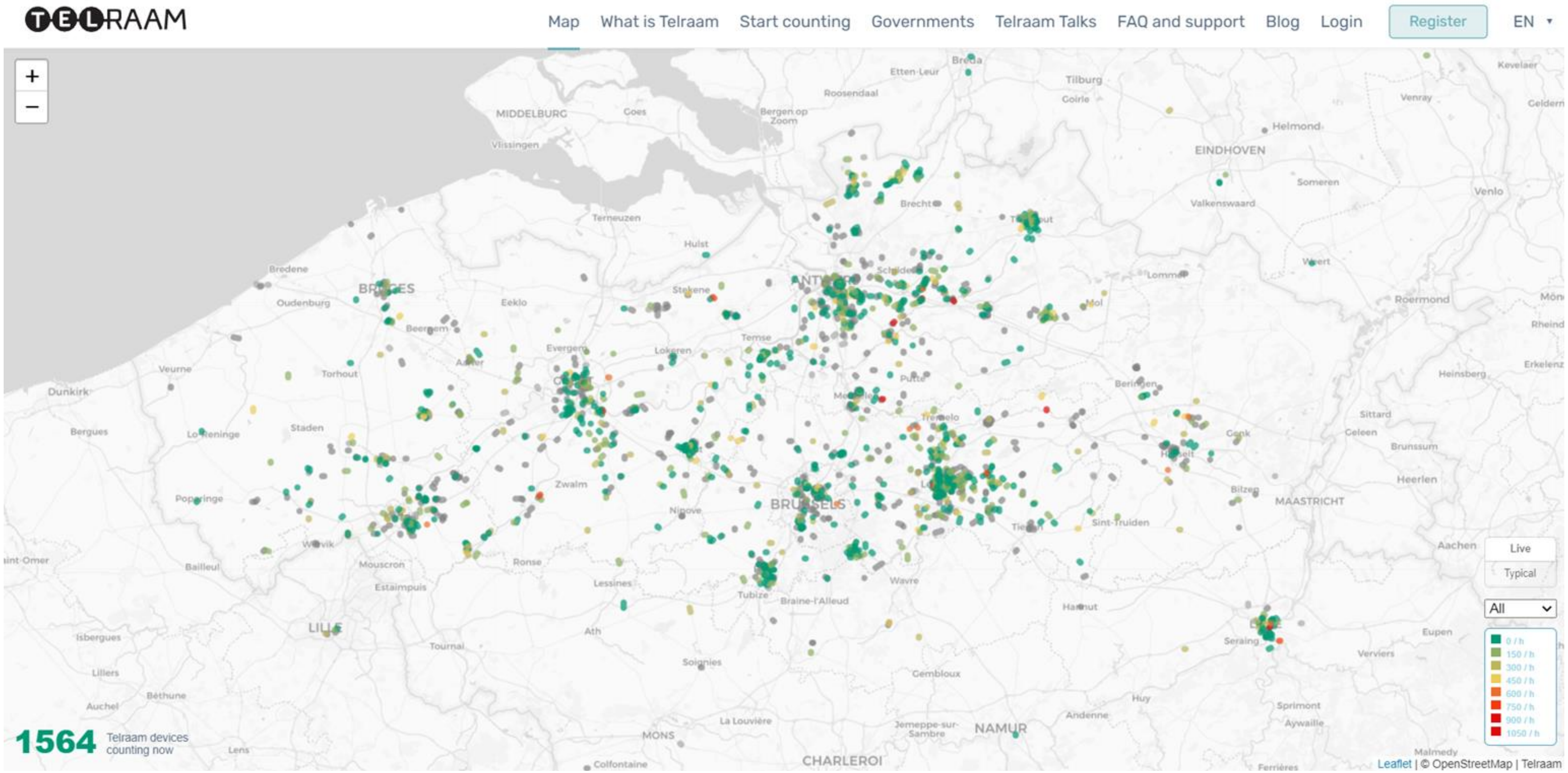
Demo

E.D.&A.

Future work

- Optimize NN, a smaller network might work equally reliable
- Implement on smaller µC with Cortex M0

TML: Introduction



TML: Introduction

Telraam project

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

How?

- Classical 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

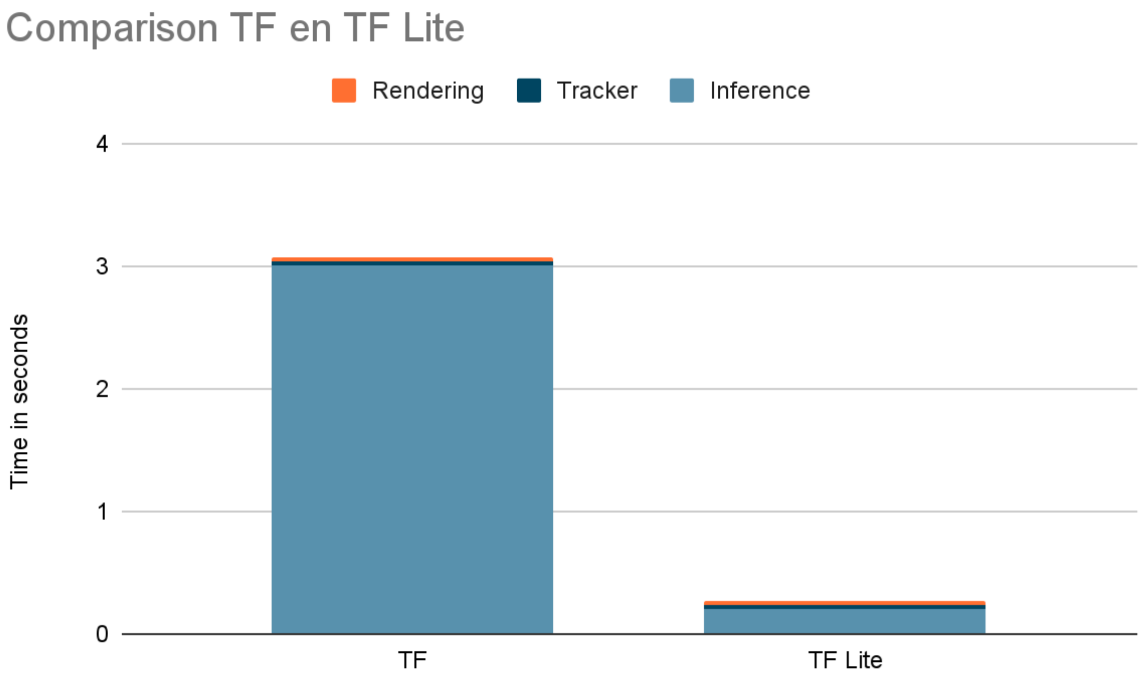
Valid .tflite model
.tflite compatible labelmap file

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:

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

6Wolves/Yogalife

Exercise to validate: a squat

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

Step 1: Annotate a dataset with body keypoints

Detection of body position using movenet

→ Further investigation required

(movenet/openpose/other

6Wolves/Yogalife

Step 2: Convert keypoints to good/bad position

Keypoints from movenet imported in Edge Impulse

→ Further investigation required for AI network

Step 3: Auto-annotate IMU data with AI model

Step 4: Inference on IMU's

Bluetooth Low Energy

Challenge: multiple devices

6Wolves/Yogalife

Other research questions:

- Position of IMU's
- Number of IMU's

Progress:

- Student now working on step 2
- Slow progression
- → We will continue

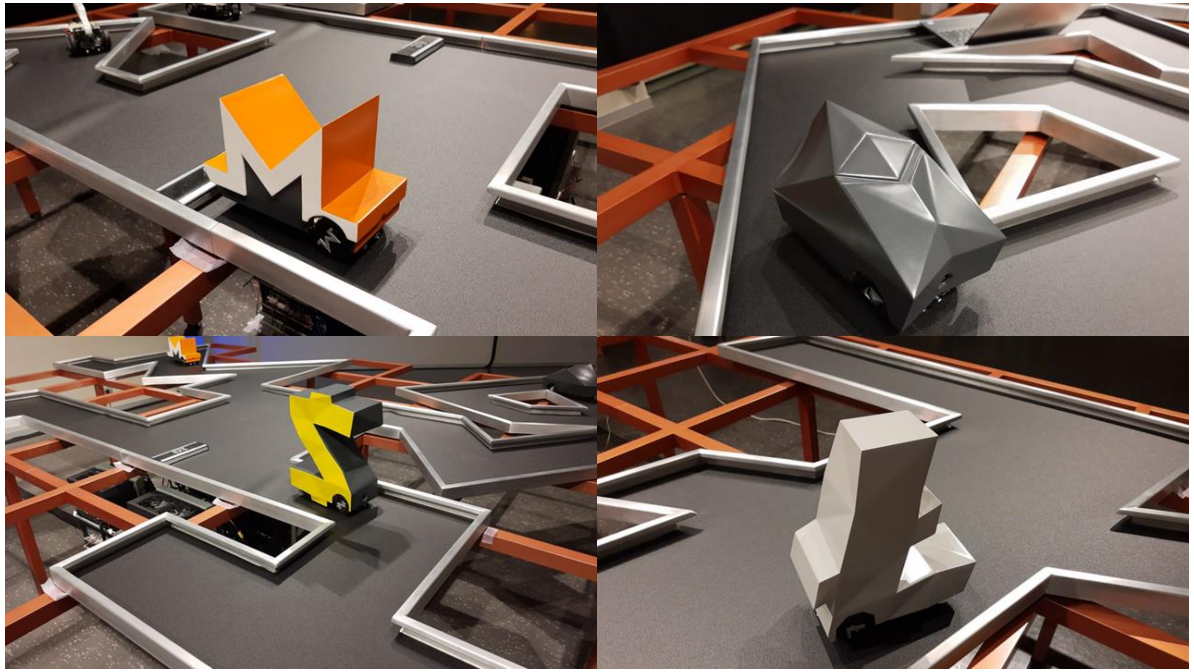
Artists Duo - LarbitsSisters

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 (Etherium, 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.

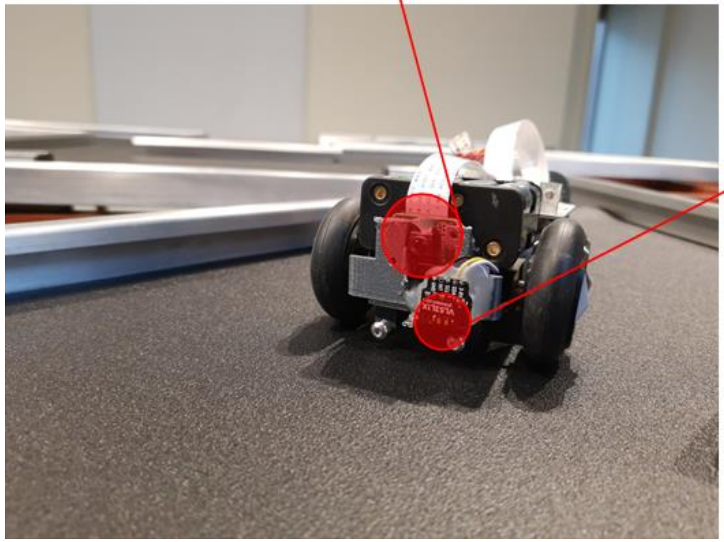
The car, once status symbol of modernity, acts here as a visionary trigger probing possible visions of the future between reality and fiction. The CMC brings a car that moves towards a new and disruptive form of mobility. Within the critical discourse on climate change, CO_2 emissions and global warming, 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.

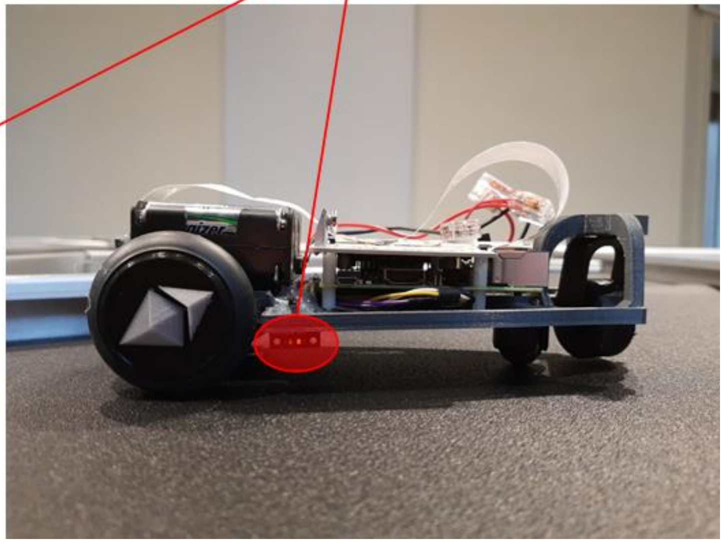
Artists Duo - LarbitsSisters

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

Raspberry Pi Camera (640 x 480 resolution)

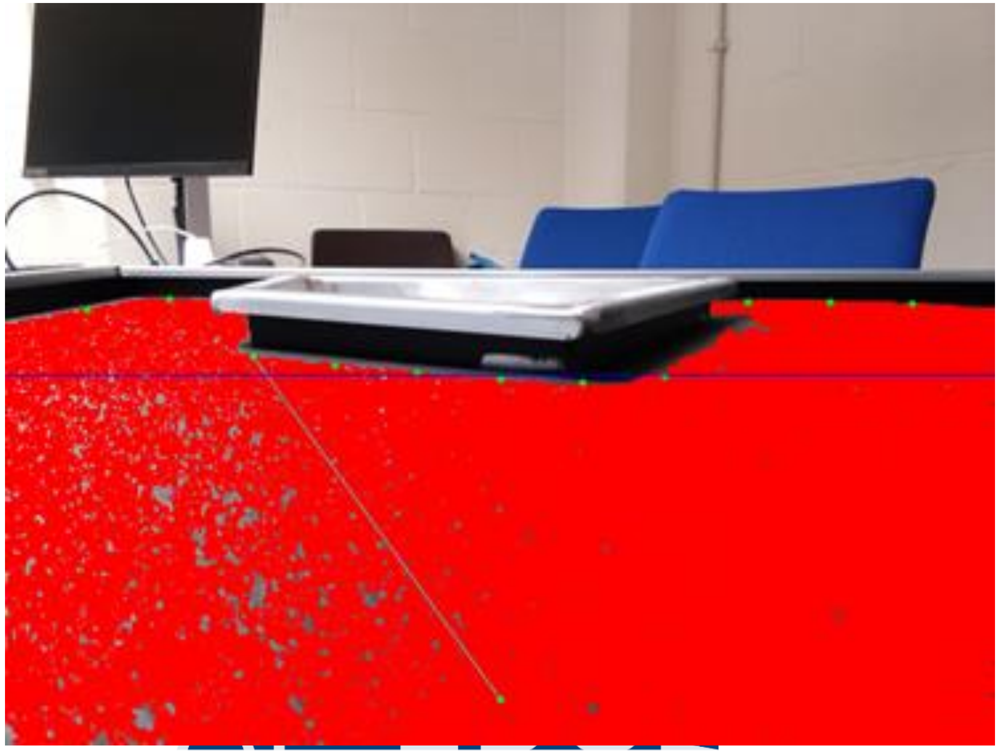
Three distance sensors (TOF, VL53L1X): Front, left, right





Artists Duo - LarbitsSisters

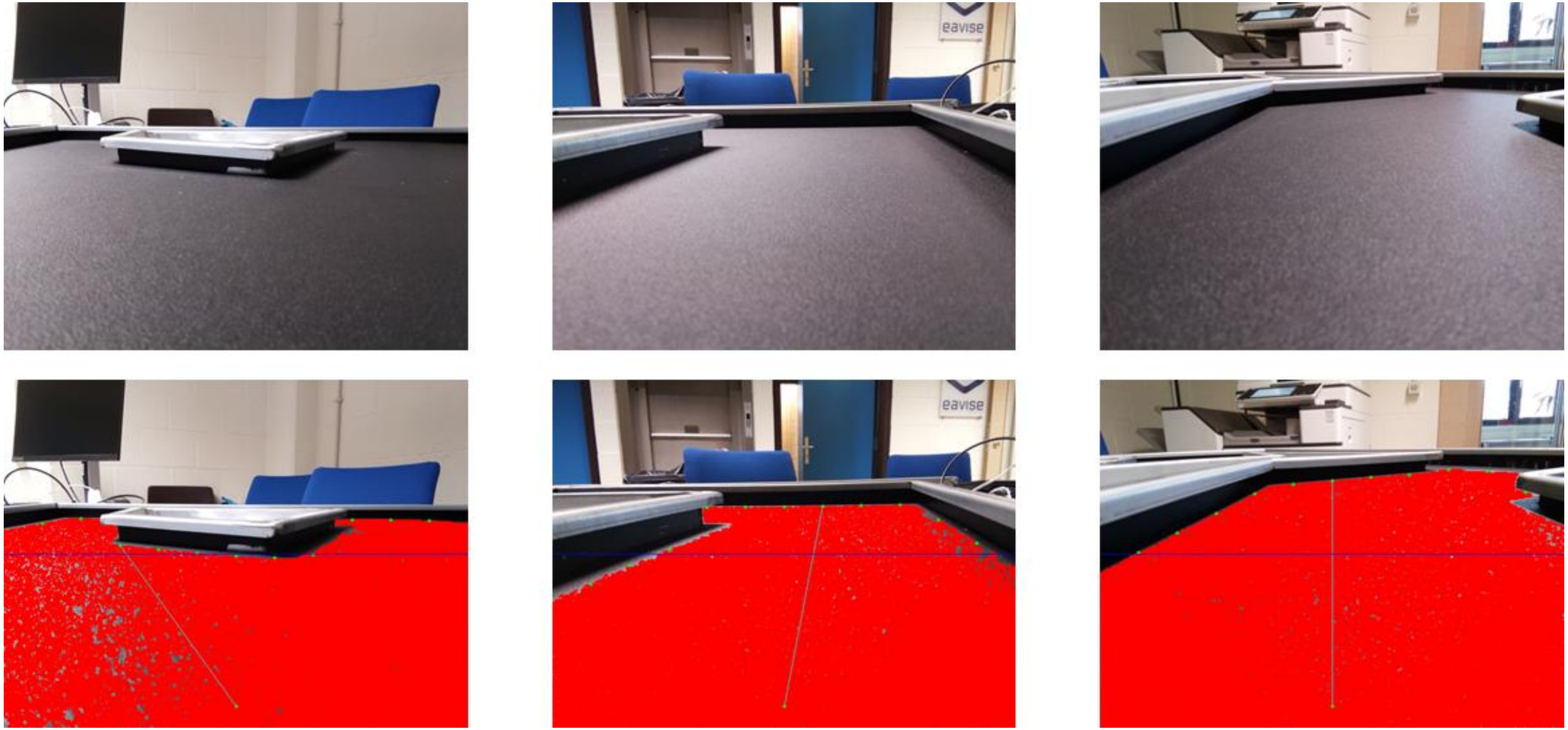
- Vision algorithm:
    - Segment track based on Floodfill algorithm
    - Divide image in 11 equidistant segments
    - For each segment, find furthest point in segmented track (green dots)
    - Threshold the segments (blue line)
    - Determine largest group of points
    - Find middle of largest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

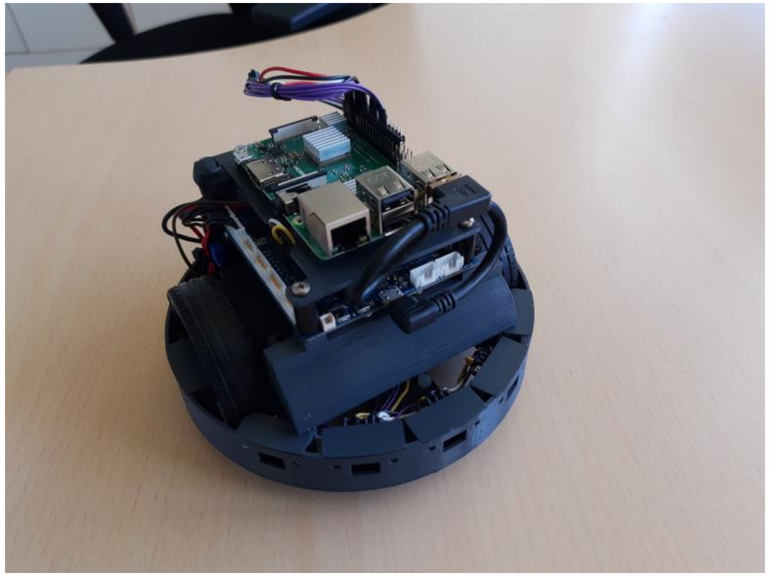
Artists Duo - LarbitsSisters

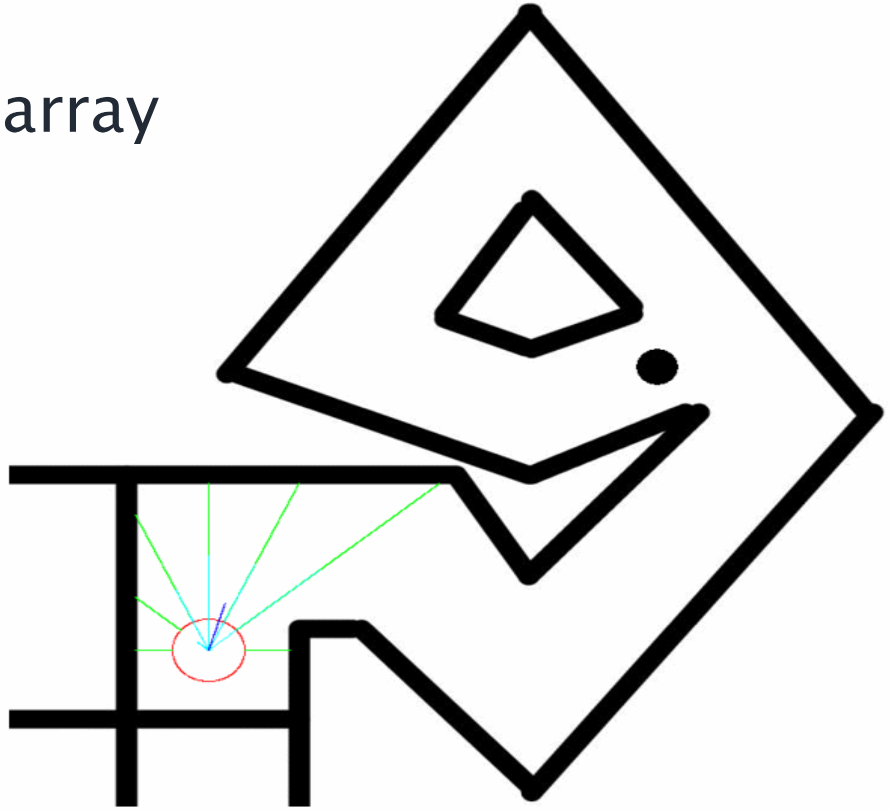
Artists Duo - LarbitsSisters

- 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
  - Uses 7 distance sensors in 180 degree array
    - Based on force-field algorithm
  - Circular chassis, more reliable motors
  - Much faster, more agile





hogeschool vives KU LEUVEN

AI@EDGE

https://youtu.be/2Xhqavry95c

Artists Duo - LarbitsSisters

Gemeente Sint-Katelijne-Waver

- Catching stray cats via trap cage
- Goal: automatic signal when a cat is caught
  - camera
  - cat detection signal
  - wireless link
- Embedded solar-powered off-grid solution

User group interaction

Question/remarks so far?

Future Plans

Manual - Best practices

First priority: Finalising running use-cases

Next: Manual & best-practices

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

Expected finalisation: July-August 2022

Final Symposium

Planned:

Beginning of June, date to be determined

Location:

Hogeschool VIVES, Xaverianenstraat 10, 8200 Brugge

For who:

Open for broad public & industry

Content:

Project overview, industry talks, use-case exhibition

Postuniversitair Centrum

3 day summerschool (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

User group interaction

Question/remarks so far?

Questionnaire

Questionnaire

Administration: Project evaluation/user poll

Link to Google Forms:
https://forms.office.com/r/UZJADqcM24

hogeschool vives KU LEUVEN

AI@EDGE

Discussion

Questions, remarks, suggestions?

Thank you!

We hope to see you on the final symposium!

The AI@EDGE Team