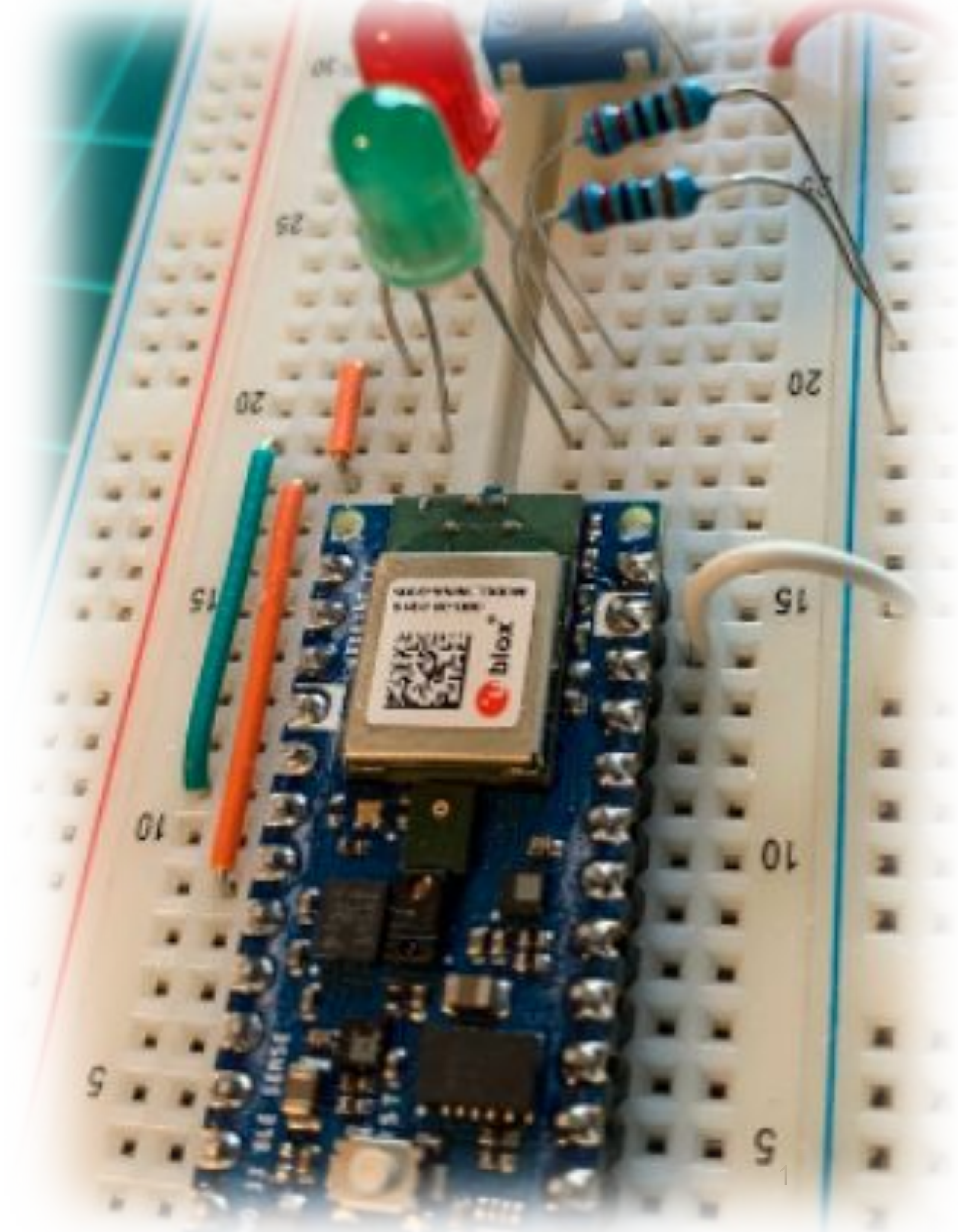
IESTI01 – TinyML

Embedded Machine Learning

28. Responsible AI & Curse Wrap-up



Prof. Marcelo Rovai
UNIFEI

Embedded ML

Curse Wrap-up

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

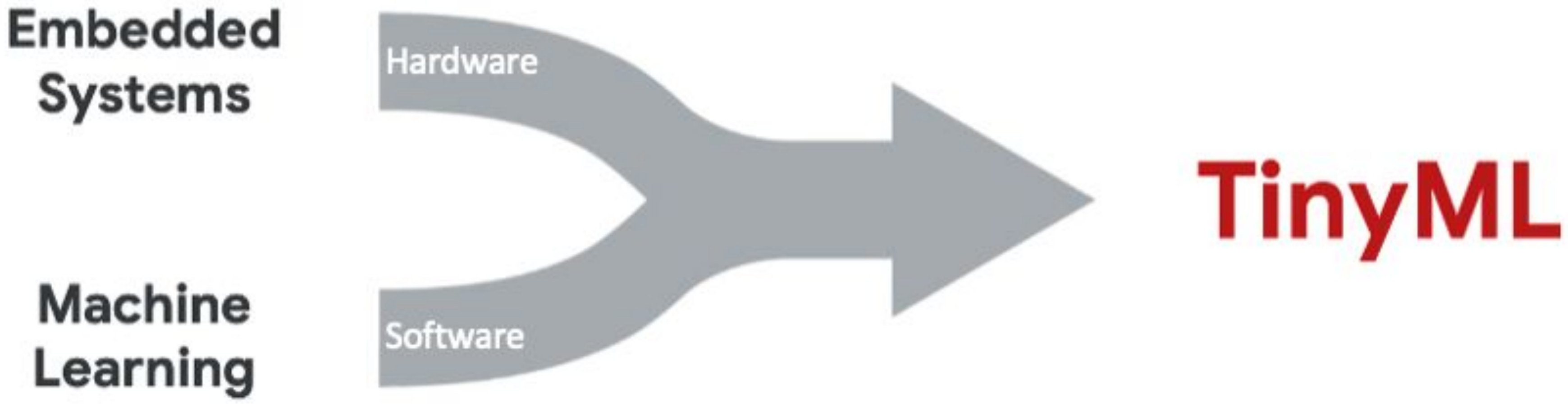
On-device sensor analytics

Low power consumption

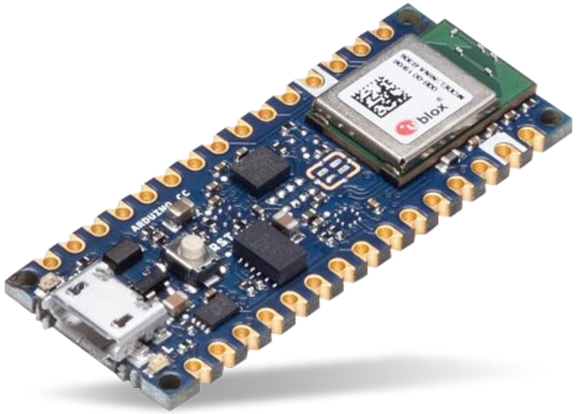
Always-on ML

Battery-operated

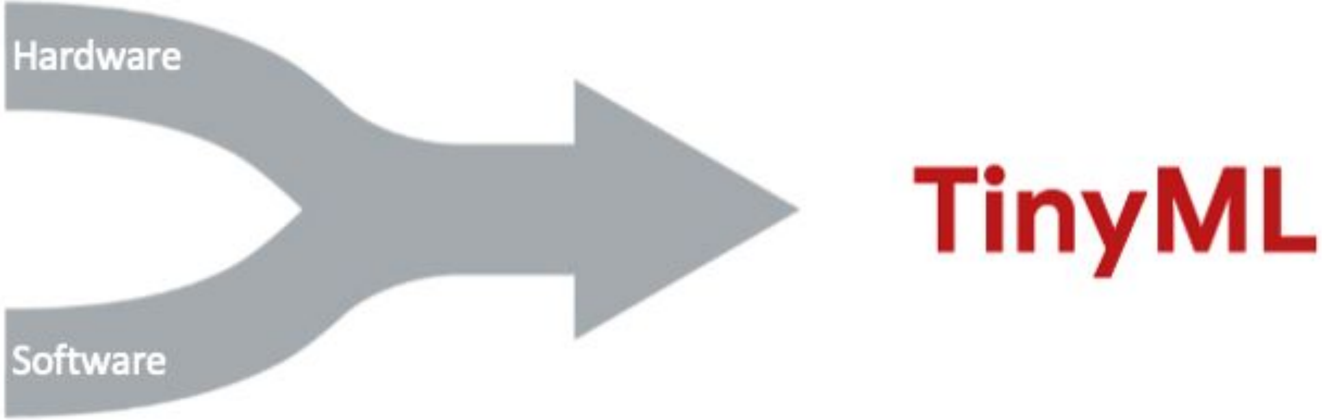
What Makes **TinyML**?



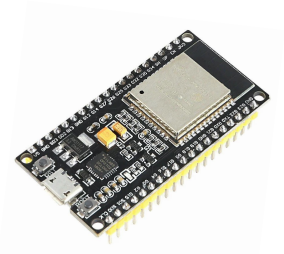
What Makes TinyML?

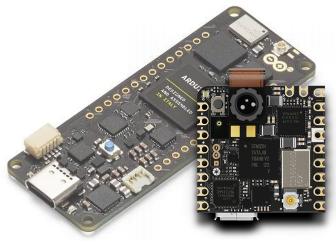


TensorFlow Lite

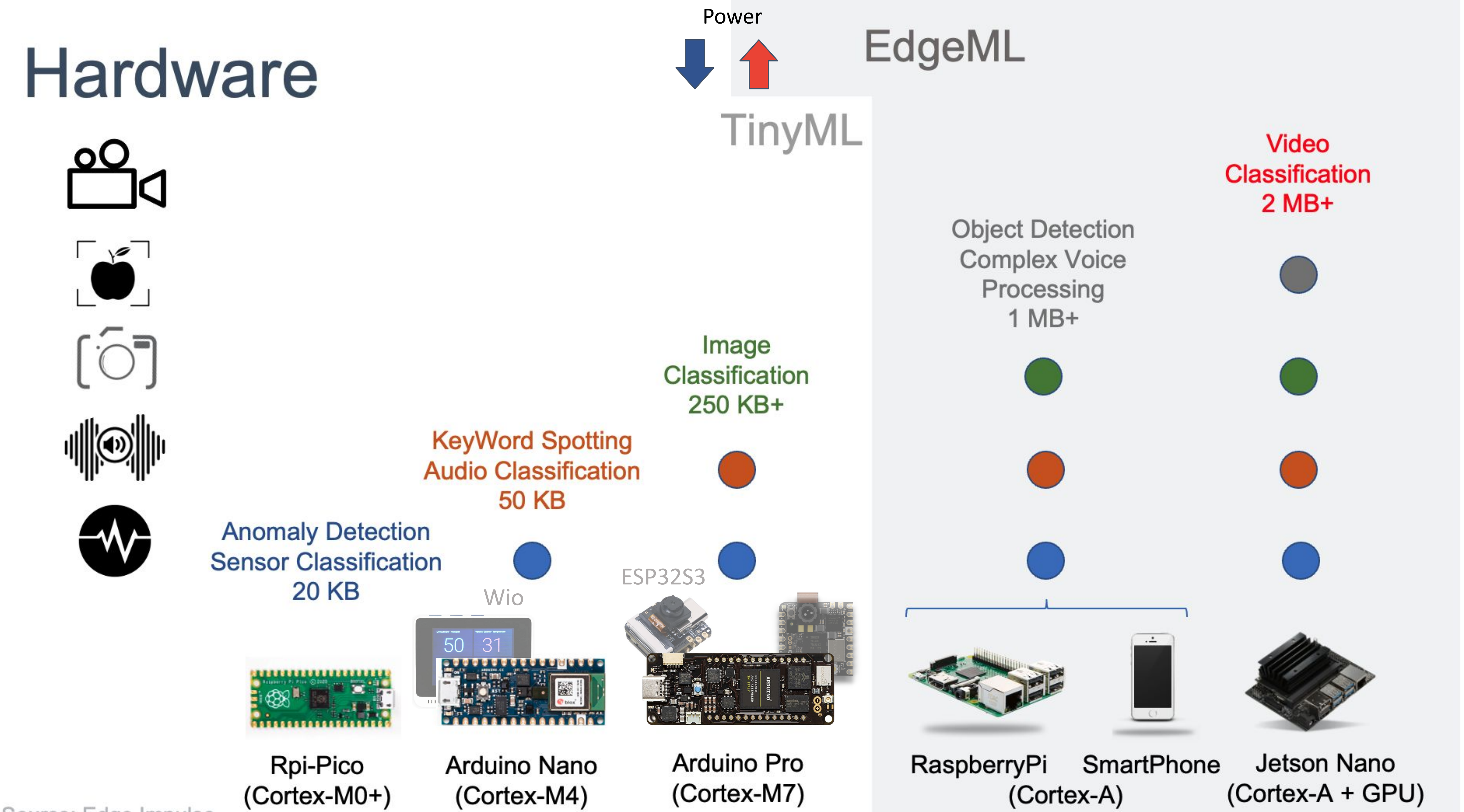


Hardware (Dev. Boards)



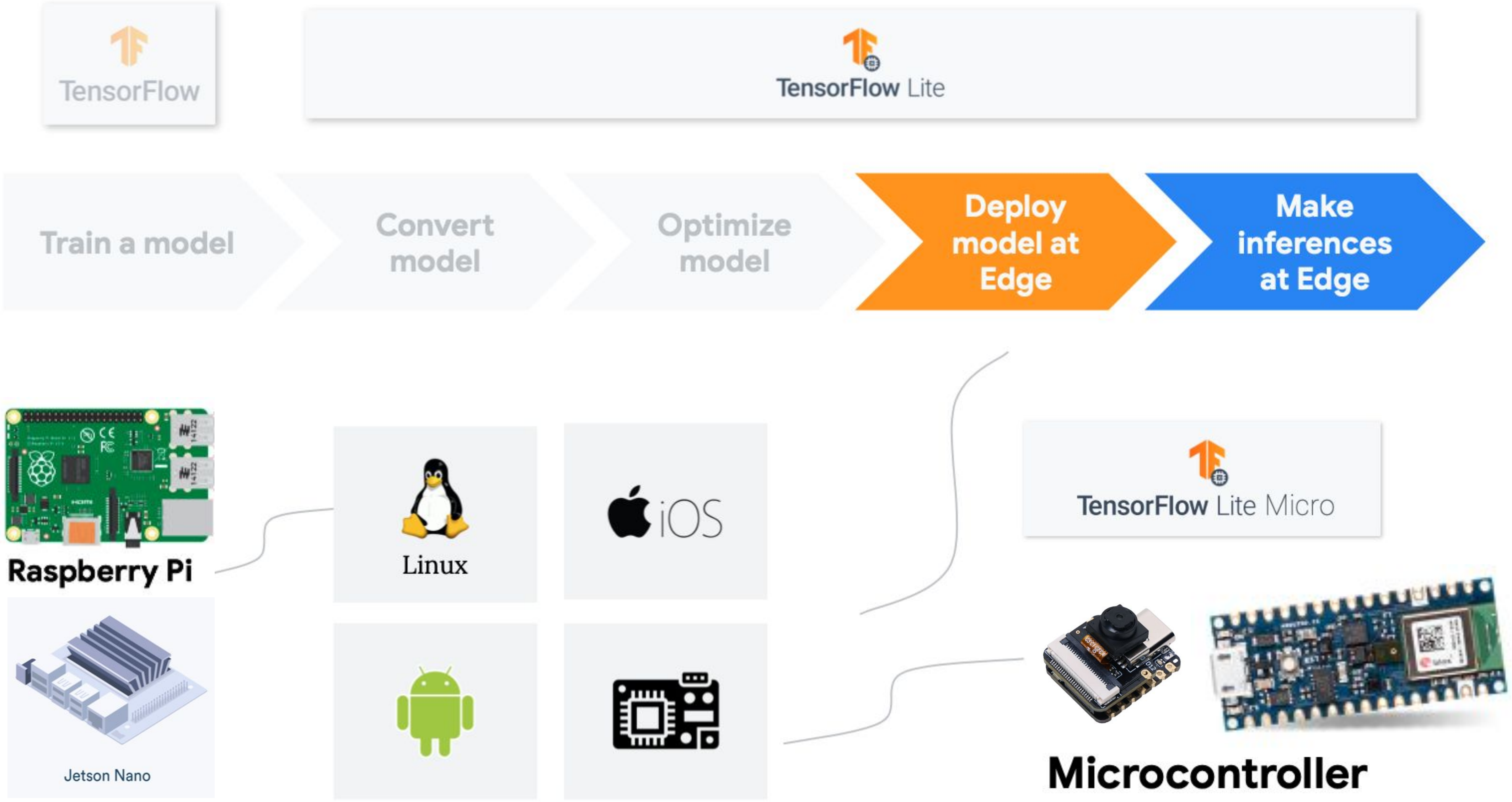


	Raspberry Pico (W)	Arduino Nano Sense	Espressif ESP 32	Seeed XIAO ESP32S3 Sense	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	240MHz	480/240MHz
RAM	264KB	256KB	520KB (part available)	8MB (PSRAM)	1MB
ROM	2MB	1MB	2MB	8MB	2MB
Radio	(Yes for W)	BLE	BLE / WiFi	BLE / WiFi	BLE / WiFi
Sensors	No	Yes	No	Yes	Yes (Nicla)
Bat. Power Manag.	No	No	No	Yes	Yes
Price	$	$$$	$	$$	$$$$$



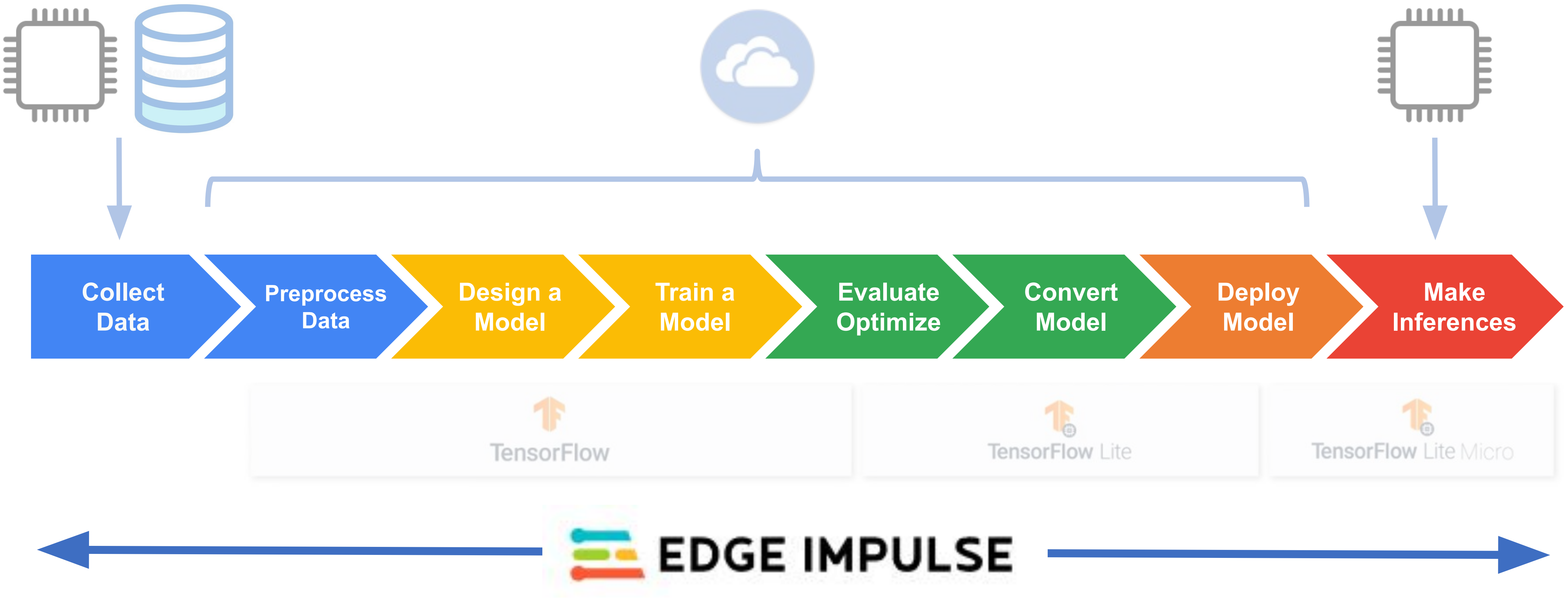
Source: Edge Impulse

Software

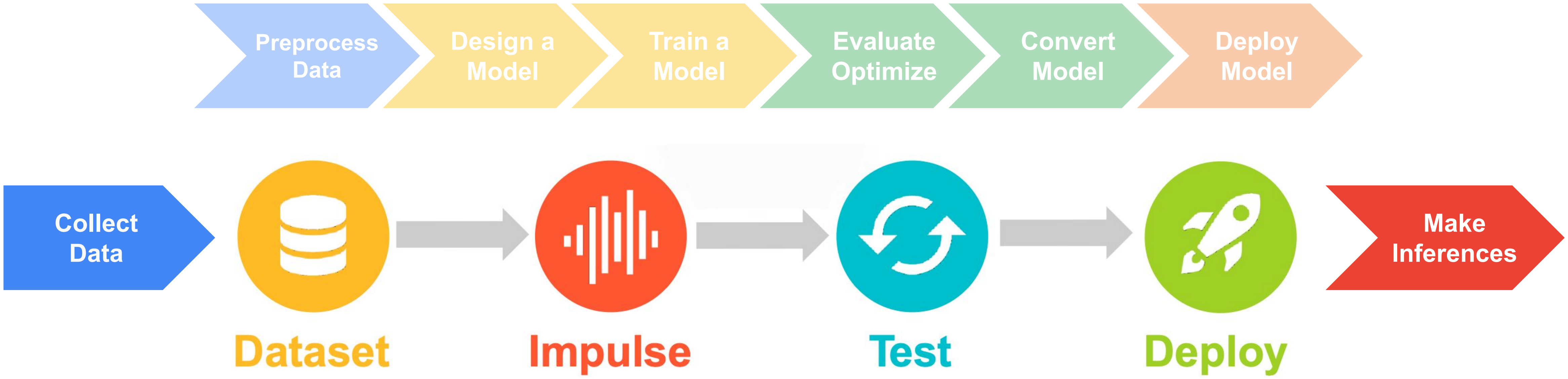

TensorFlow
TensorFlow Lite
Train a model
Convert model
Optimize model
Deploy model at Edge
Make inferences at Edge
Raspberry Pi
Linux
iOS
Jetson Nano
TensorFlow Lite Micro
Microcontroller

How to Train a ML Model?

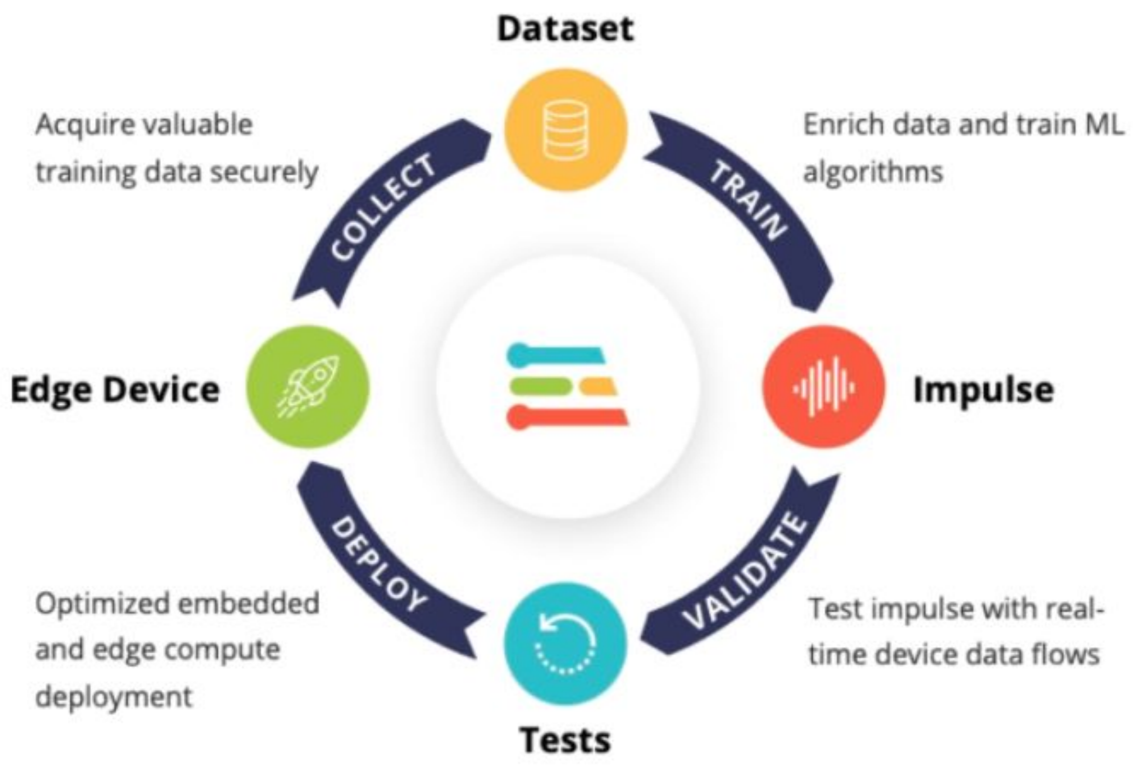
Machine Learning Workflow

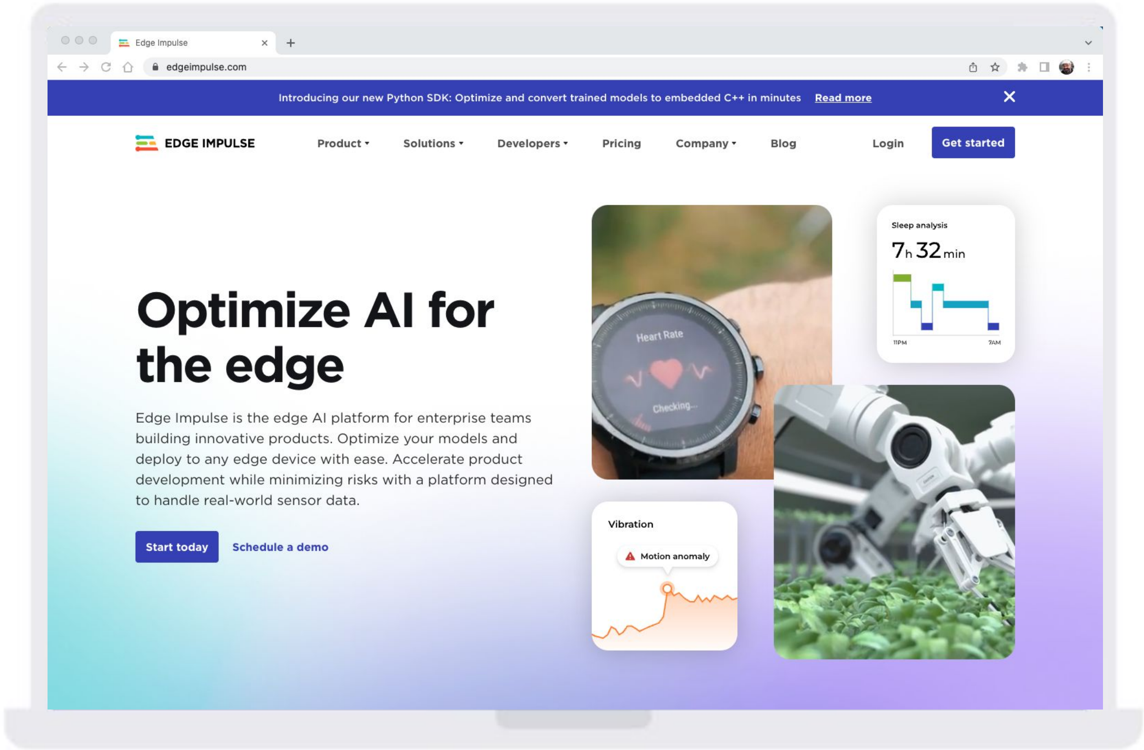


Machine Learning Workflow



EI Studio - Embedded ML platform



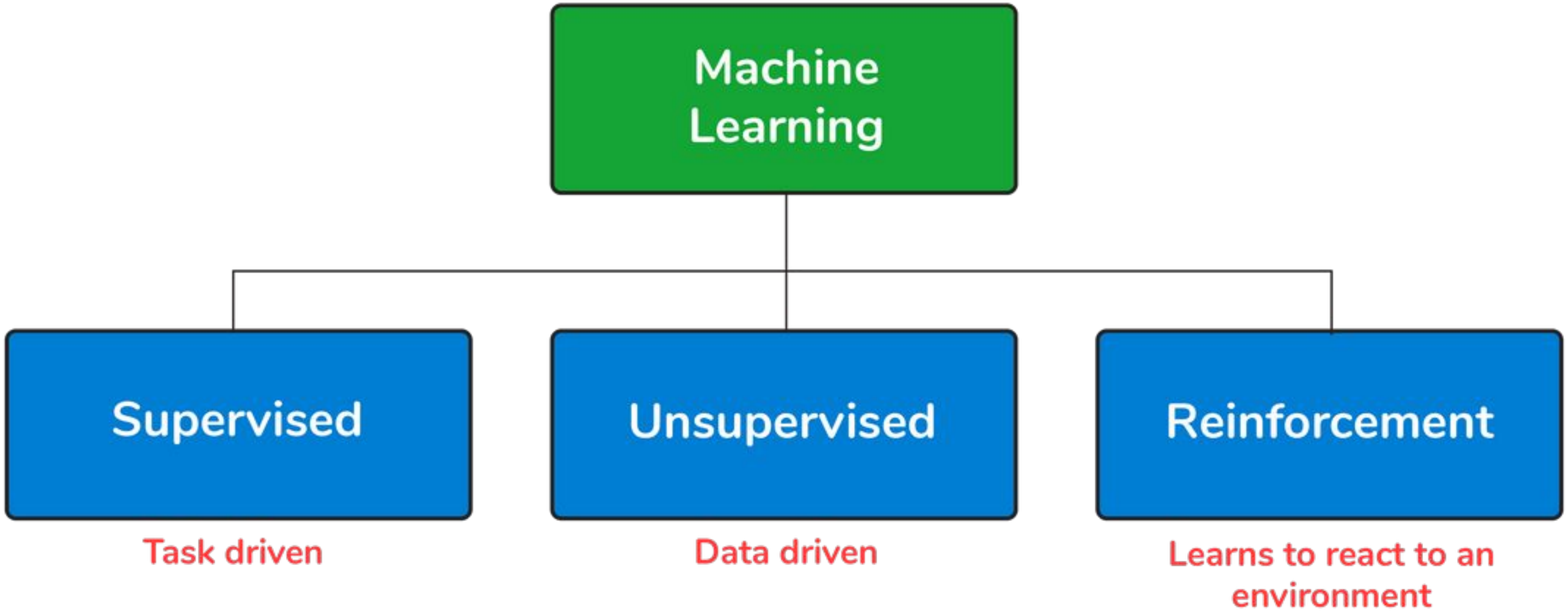


Learn more at http://edgeimpulse.com

[2023 SciTinyML Workshop - Edge Impulse Overview by Shawn Hymel]()

TinyML Application

Examples



- Regression
- Classification
- Object Detection

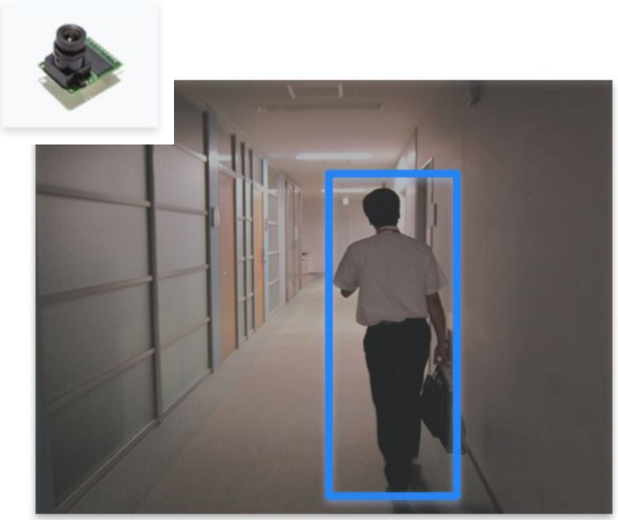
- Anomaly Detection

- Autonomous Navigation

Sound

Vibration

Vision

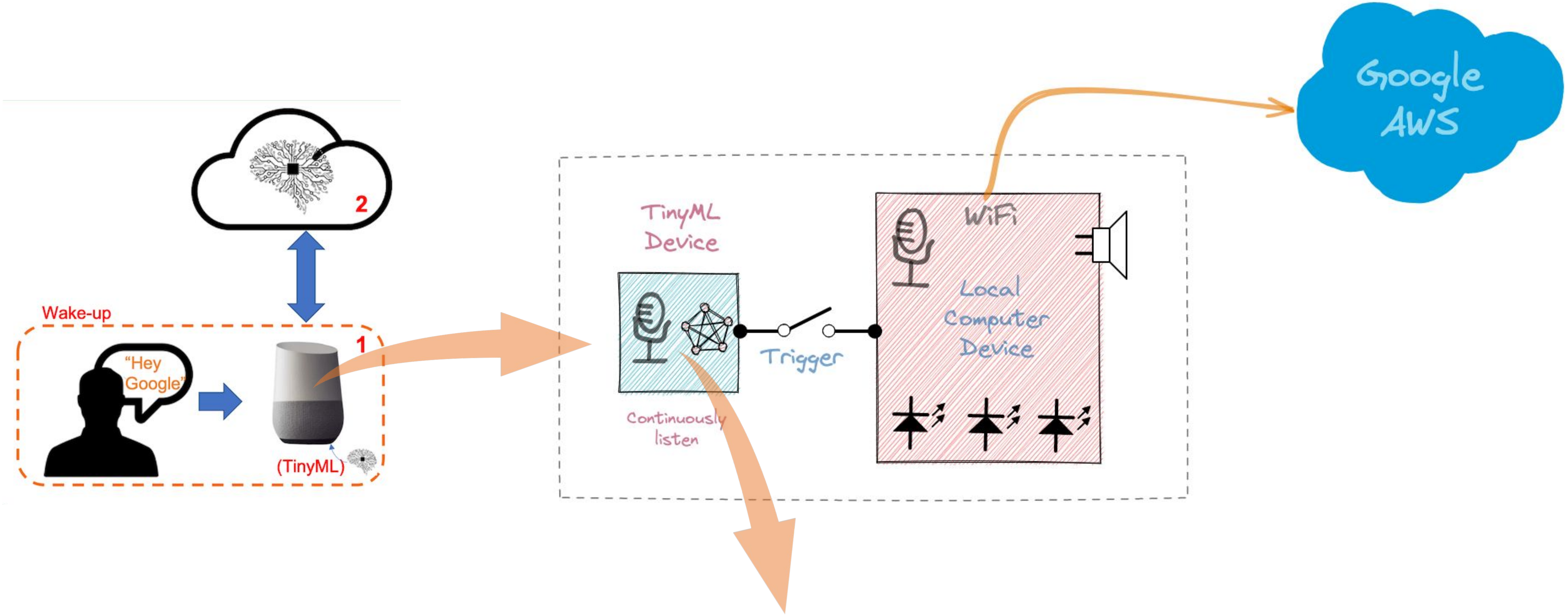


Sound

Vibration

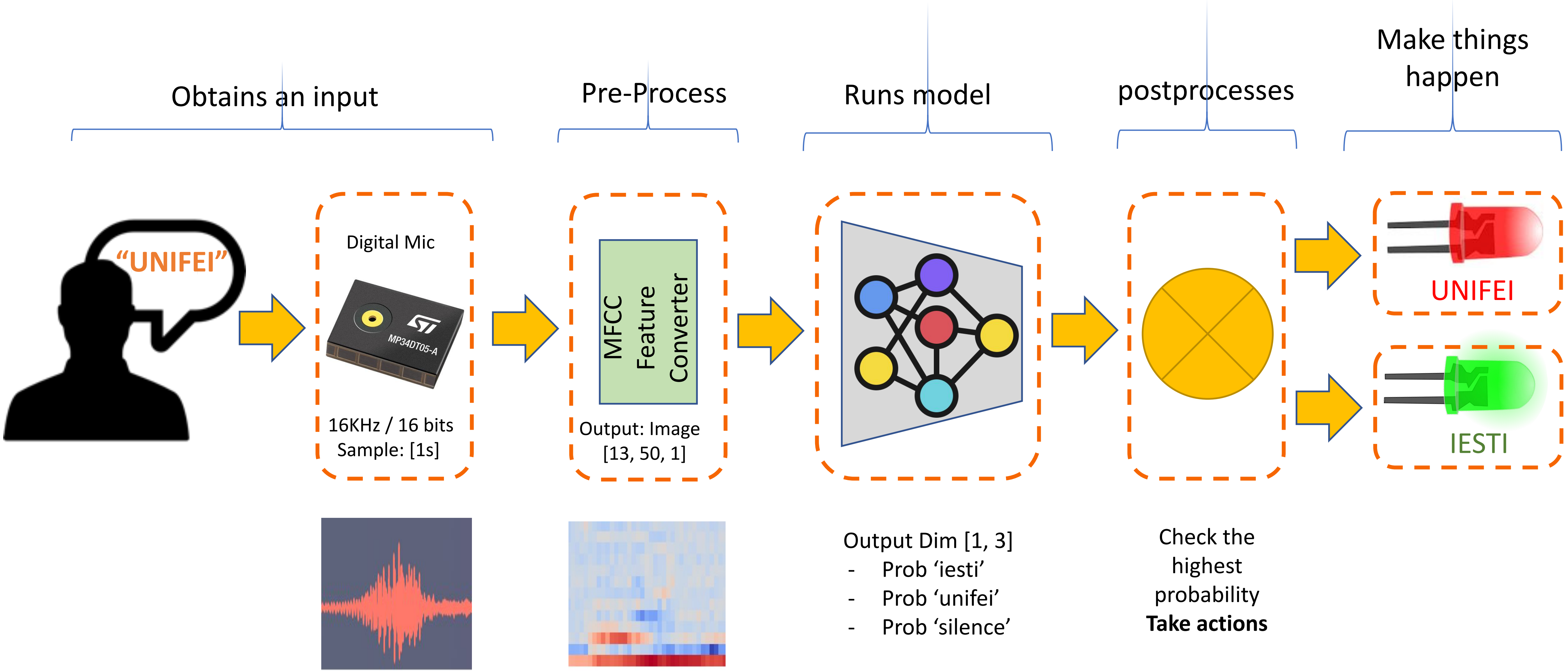
Vision

Personal Assistant



KWS

KeyWord Spotting (KWS) - Inference



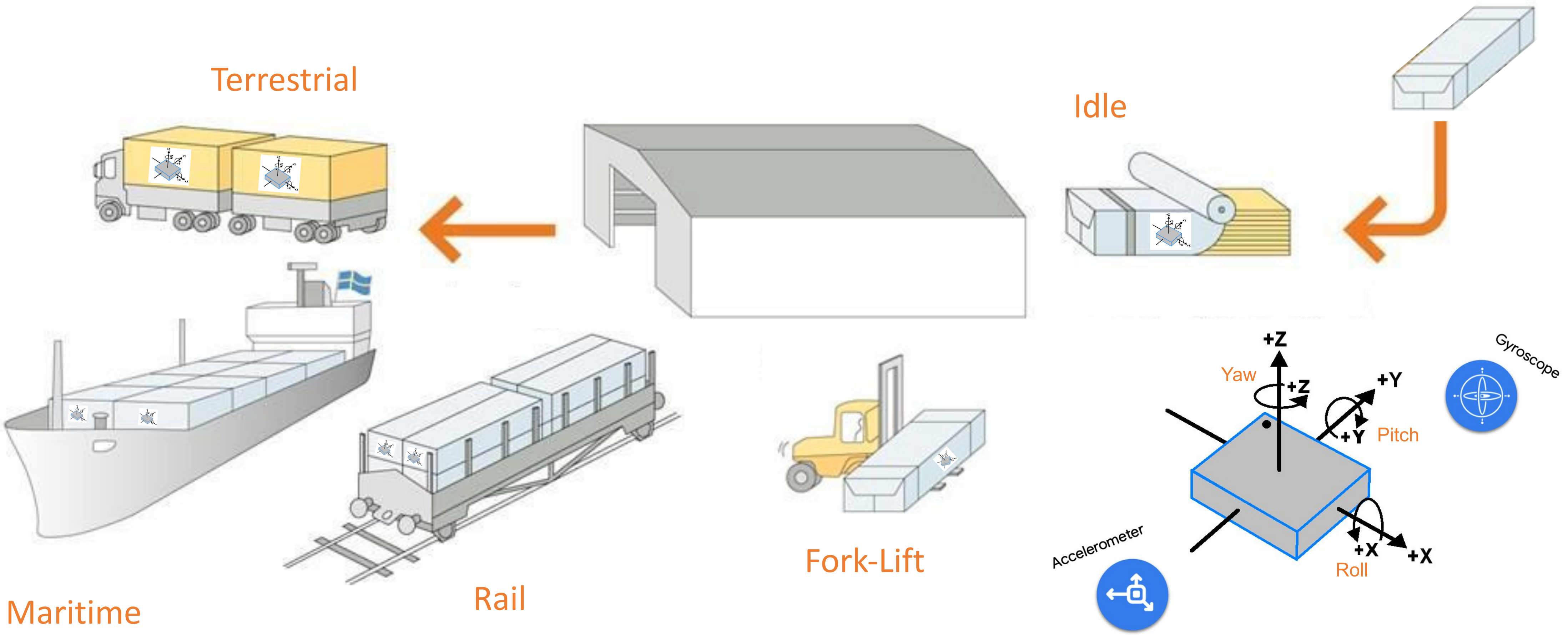
https://youtu.be/XnFYz-RSNe8

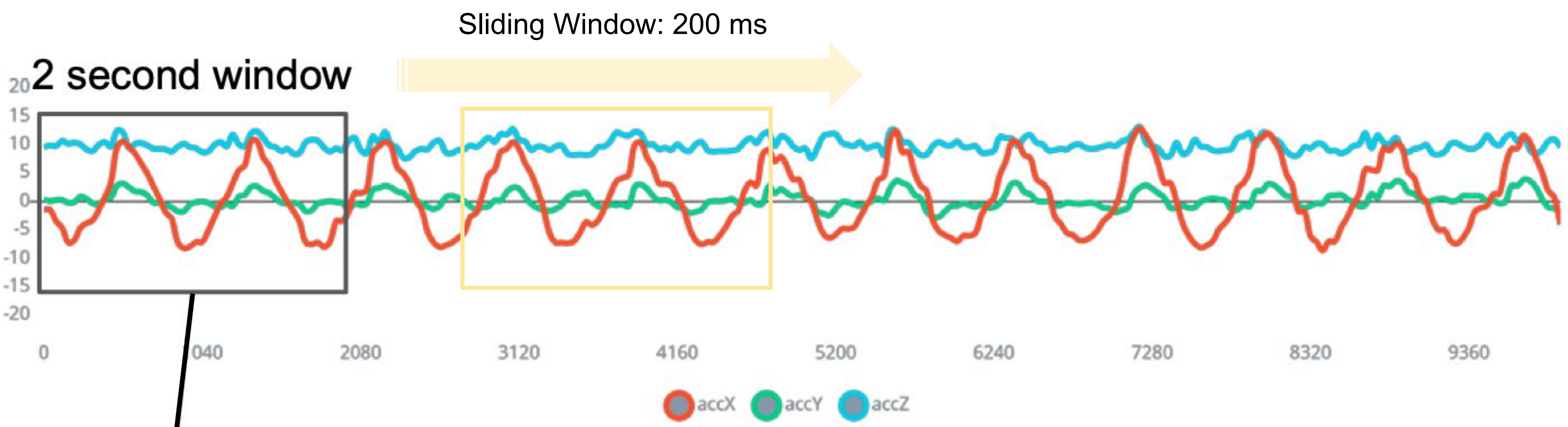
Sound

Vibration

Vision

Mechanical Stresses in Transport





300 Raw Features

Manual Feature Extraction

Raw Data from sensor → **Spectral Analysis** → Features → **NN Classifier** → Classes

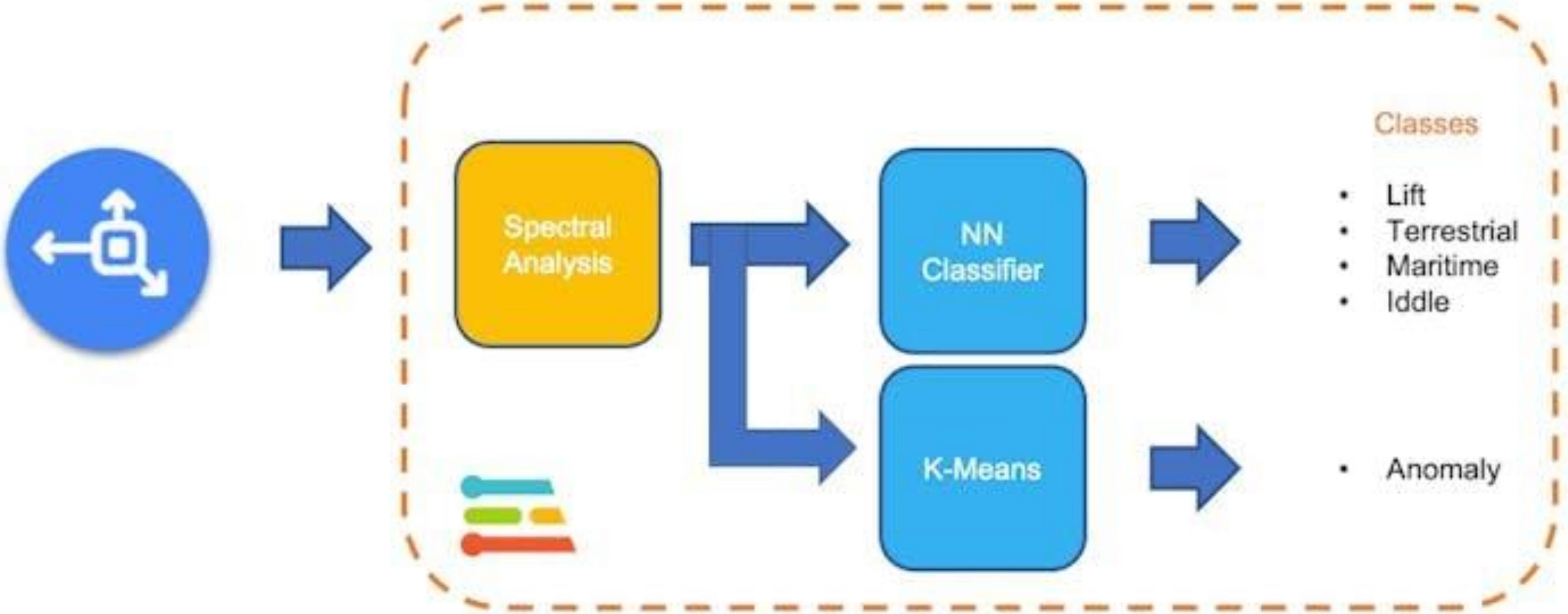
Features

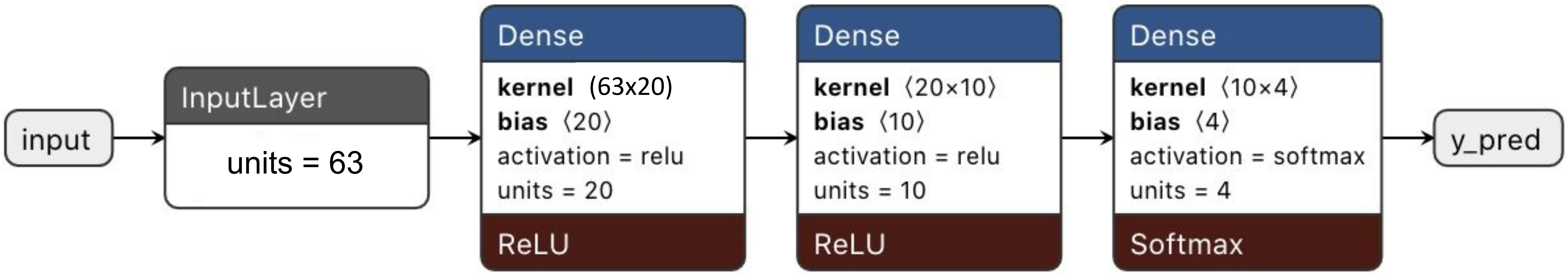
- RMS
- SKEW
- KURT
- FFT
- PSD

Classes

- Lift
- Terrestrial
- Maritime
- Idle

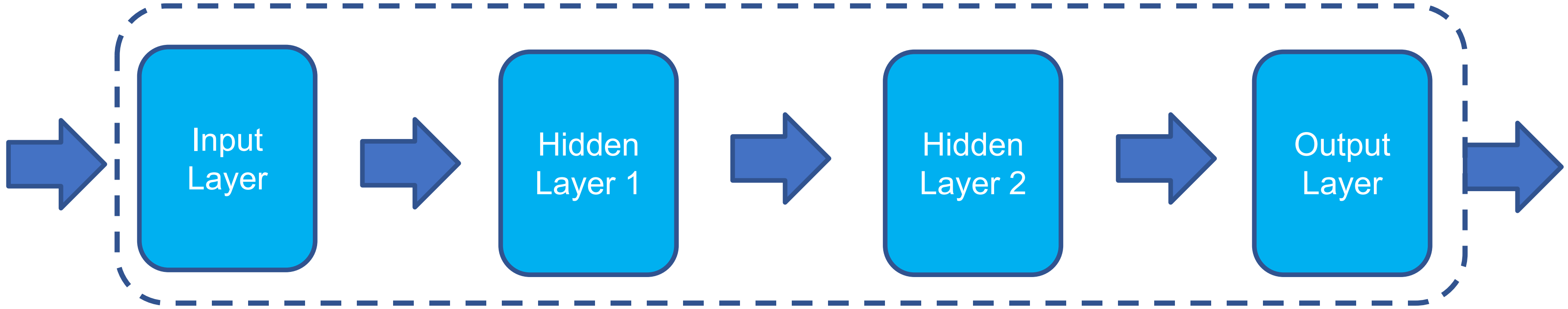
TinyML under the hood: Spectral Analysis


Spectral Analysis
NN Classifier
K-Means
Classes
Lift
Terrestrial
Maritime
Iddle
Anomaly



63 Features

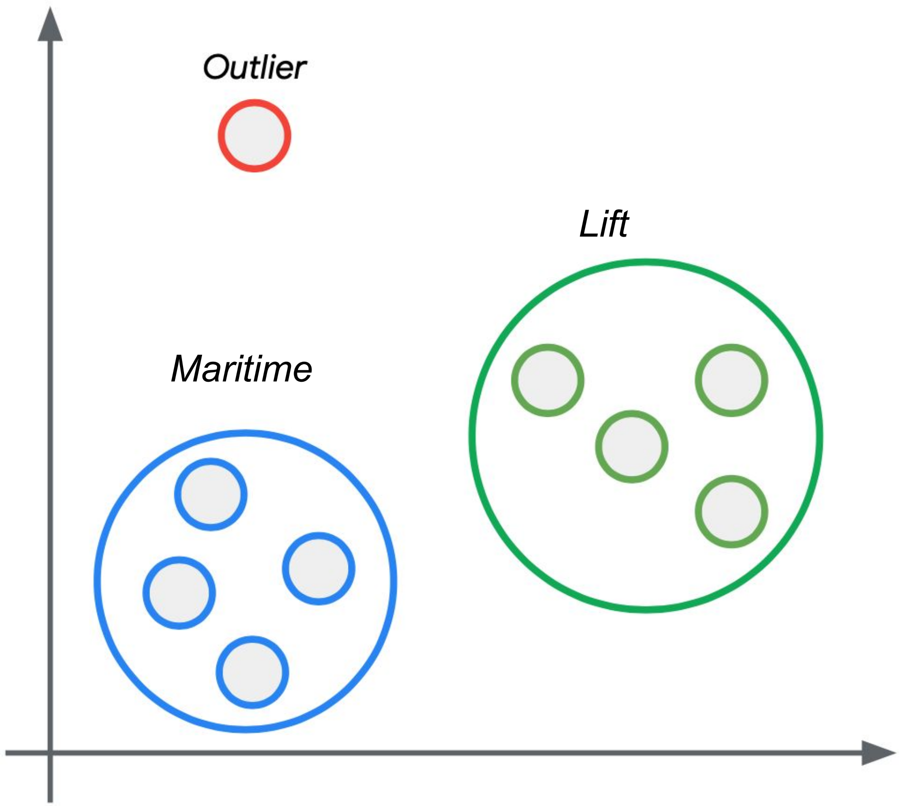
- RMS
- SKEW
- KURT
- FFT
- PSD



Classes

- Lift
- Terrestrial
- Maritime
- Idle

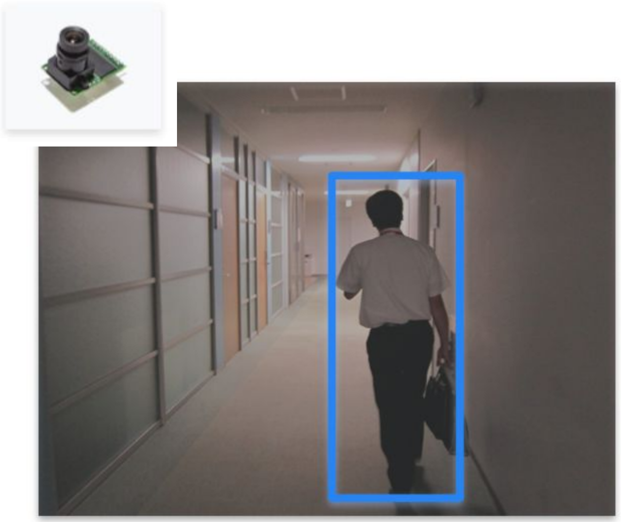
Anomaly Detection – K-Means



Sound

Vibration

Vision



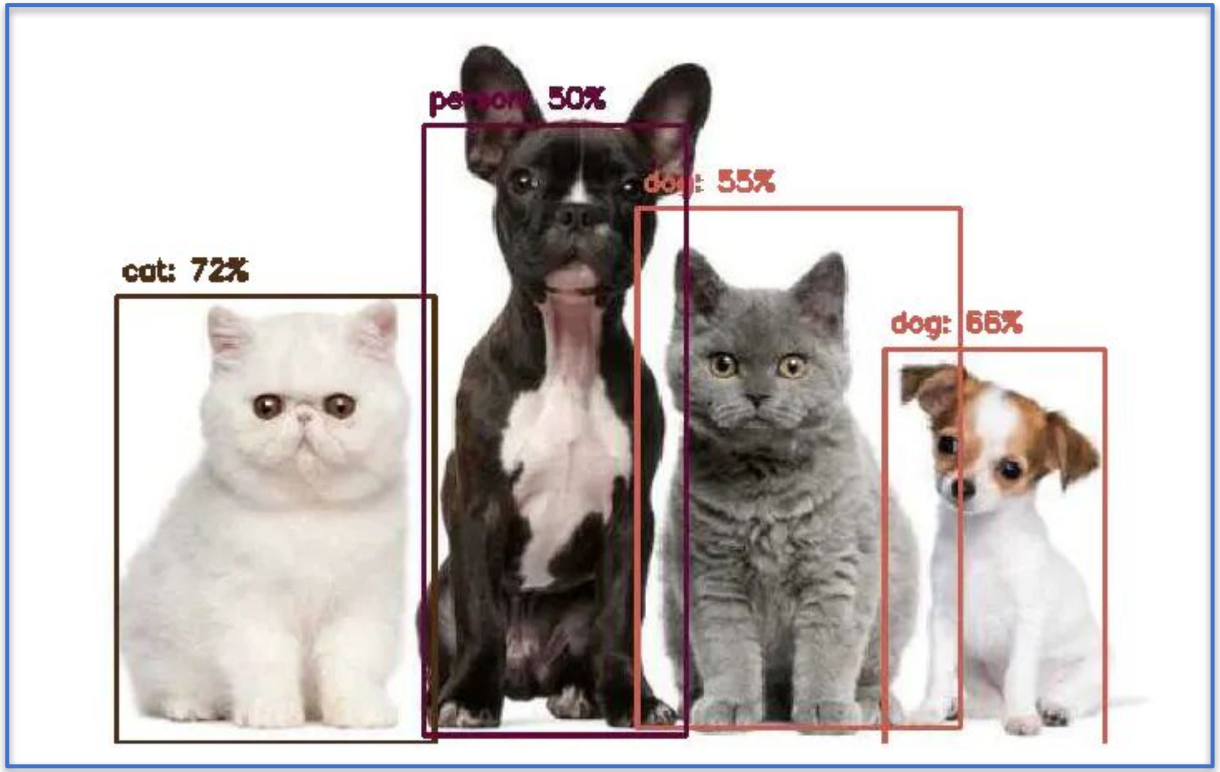
Computer Vision Main Types

Image Classification
(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

Object Detection
Multi-Label Classification + Object **Localization**



Computer Vision Main Types

Image Classification

(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

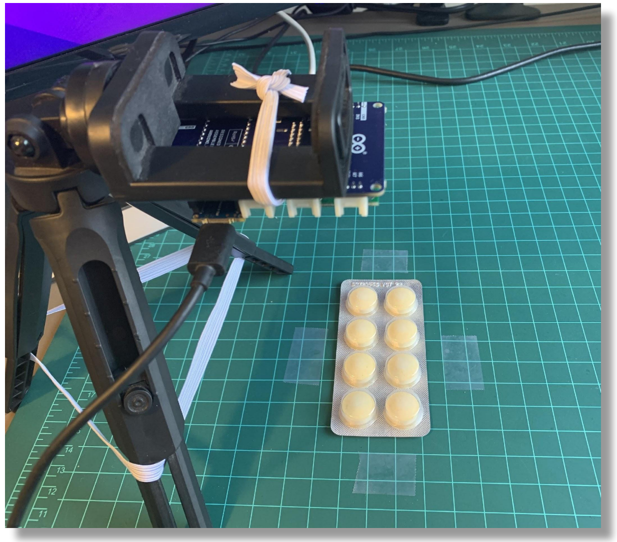
Object Detection

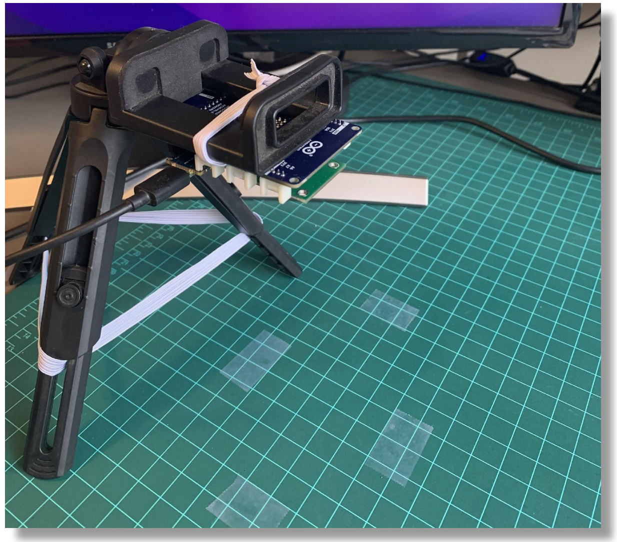
Multi-Label Classification + Object **Localization**

Image Classification Project

Decide a Goal

- Possible Images:
  - Medicine
  - background



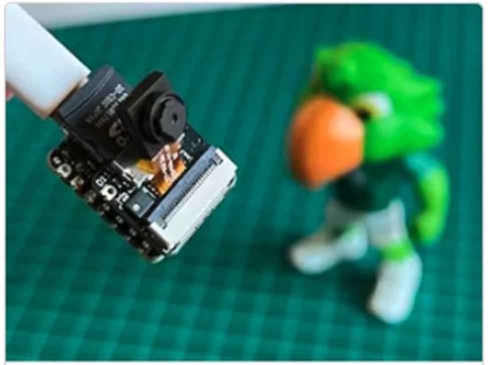


/dev/cu.usbmodem1101

Send

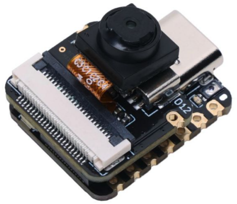
```
10:44:47.849 ->     banana: 0.01953
10:44:47.849 ->     potato: 0.12891
10:44:48.103 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.103 ->     apple: 0.86328
10:44:48.103 ->     banana: 0.03906
10:44:48.103 ->     potato: 0.10156
10:44:48.356 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.356 ->     apple: 0.90234
10:44:48.356 ->     banana: 0.02344
10:44:48.356 ->     potato: 0.07422
10:44:48.612 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.612 ->     apple: 0.91797
10:44:48.612 ->     banana: 0.02344
10:44:48.612 ->     potato: 0.05859
10:44:48.861 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
10:44:48.861 ->     apple: 0.88281
10:44:48.861 ->     banana: 0.03516
10:44:48.861 ->     potato: 0.08203
10:44:49.114 -> Predictions (DSP: 3 ms., Classification: 135 ms., Anomaly: 0 ms.):
```

✓ Autoscroll ✓ Show timestamp | Both NL & CR | 115200 baud | Clear output



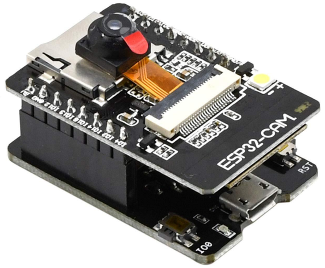
TinyML Made Easy: Image Classification

MJRoBot (Marcelo Rovai)



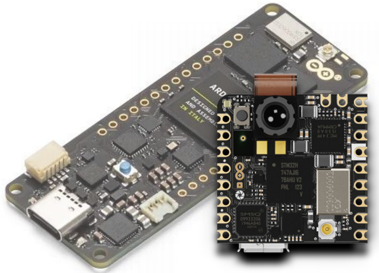
135 ms

XIAO ESP32S3
Xtensa LX7
240 MHz



171 ms

ESP - CAM
Xtensa LX6
240 MHz



45 ms

ARDUINO Pro
ARM H7
480 MHz

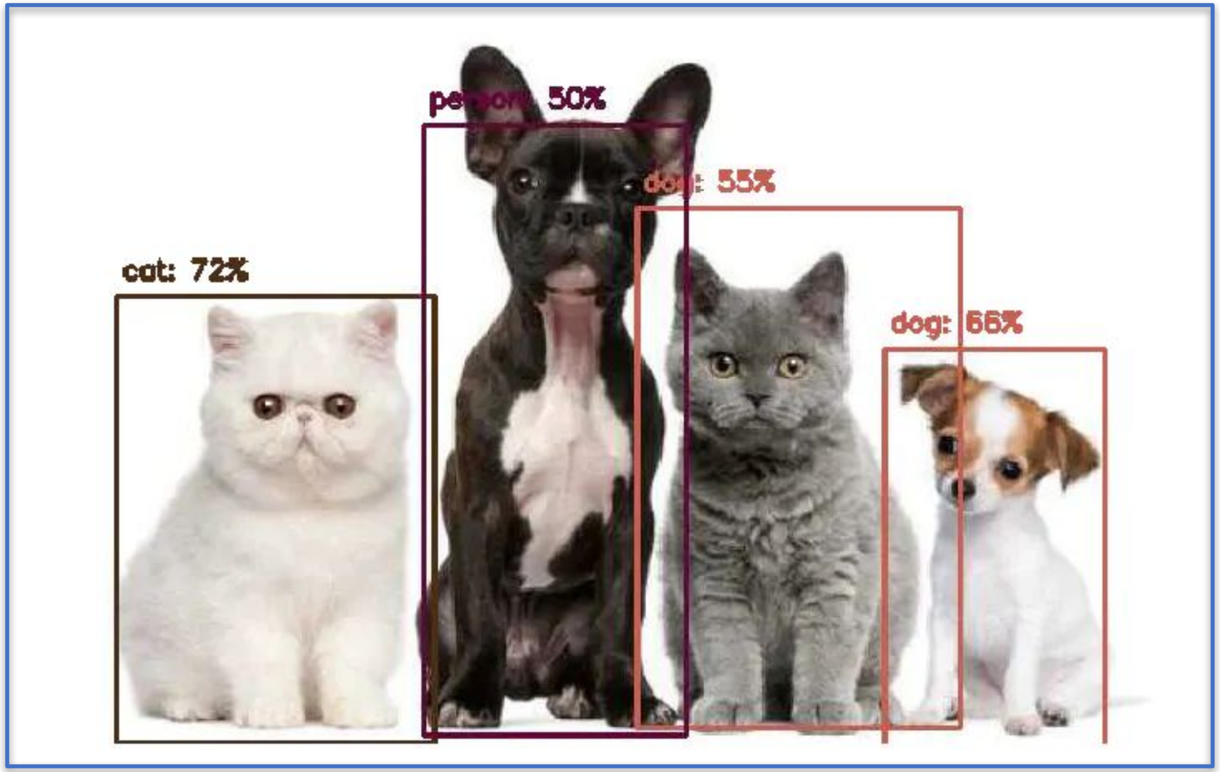
Computer Vision Main Types

Image Classification
(**Multi-Class** Classification)

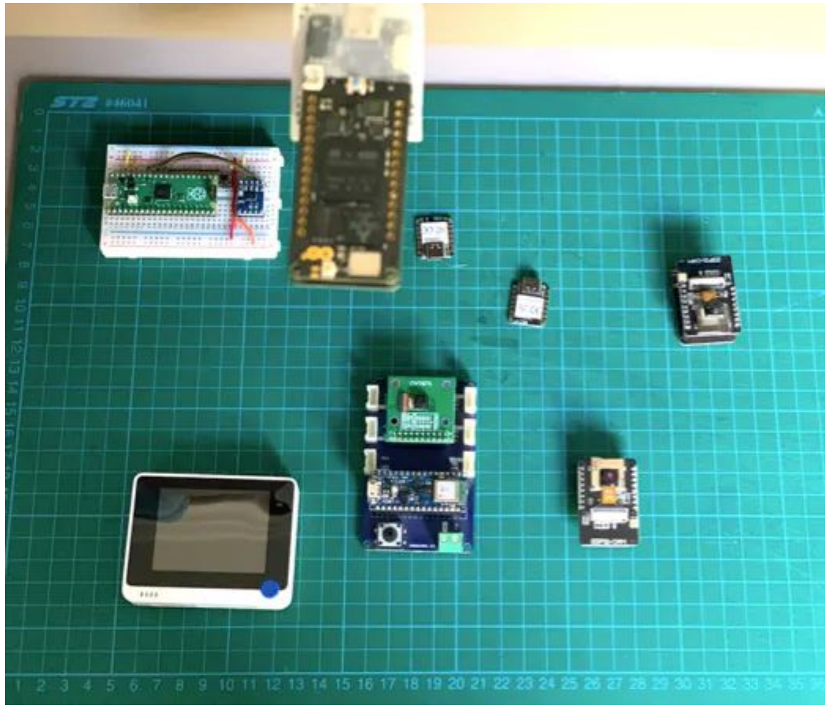
Cat: 70%

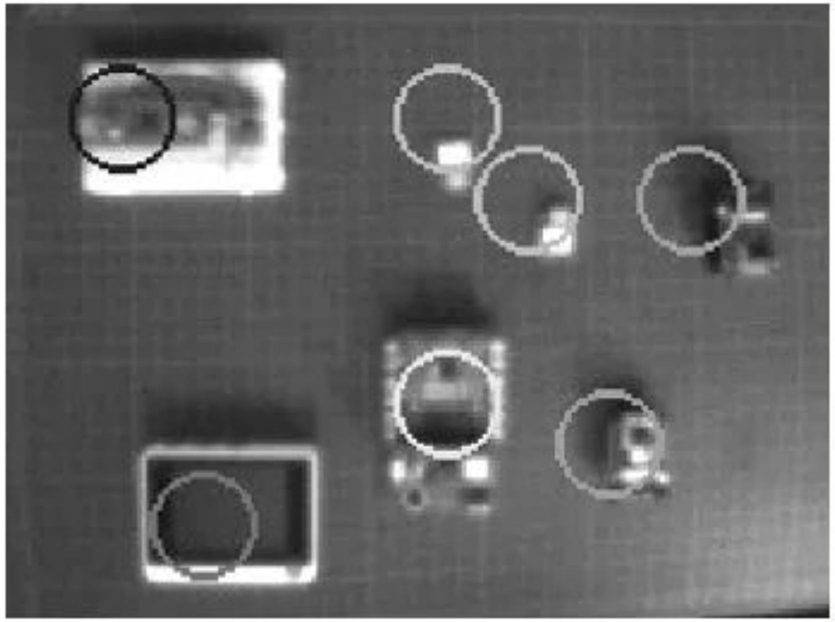
Dog: 80%

Object Detection
Multi-Label Classification + Object **Localization**



Detecting Objects using TinyML (FOMO)

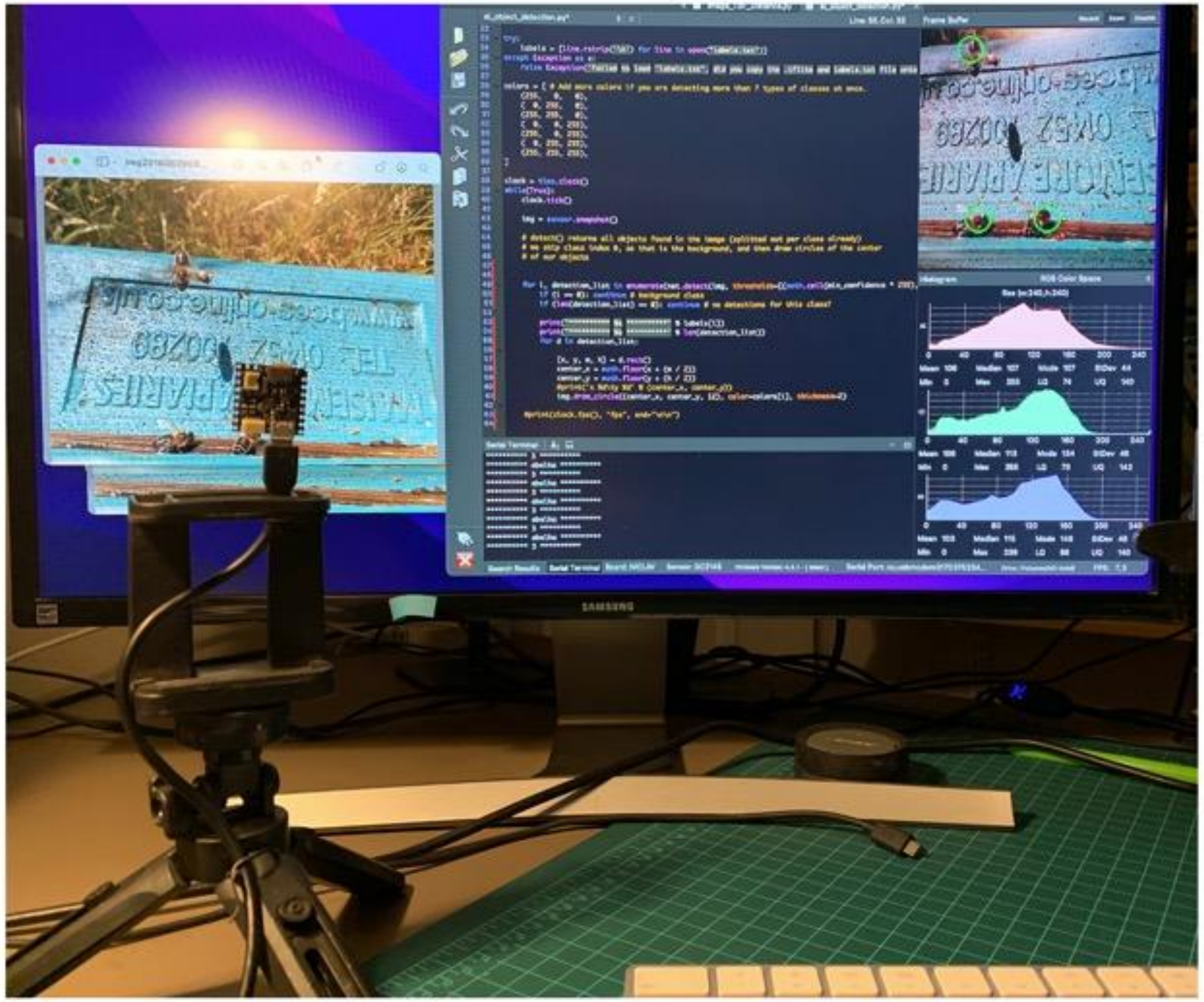


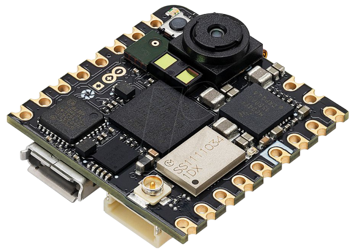


```
********** espcam **********
x 70    y 150
x 130   y 170
********** nano **********
x 70    y 110
********** pico **********
x 150   y 30
********** wio **********
x 50    y 50
********** xiao **********
x 150   y 110
x 130   y 130
6.97512 fps
```

[EdgeAI made simple - Exploring Image Processing (Object Detection) on microcontrollers with Arduino Portenta, Edge Impulse FOMO, and OpenMV]()

Detecting Objects using TinyML (FOMO)





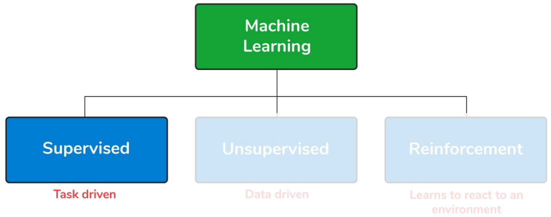
OpenMV

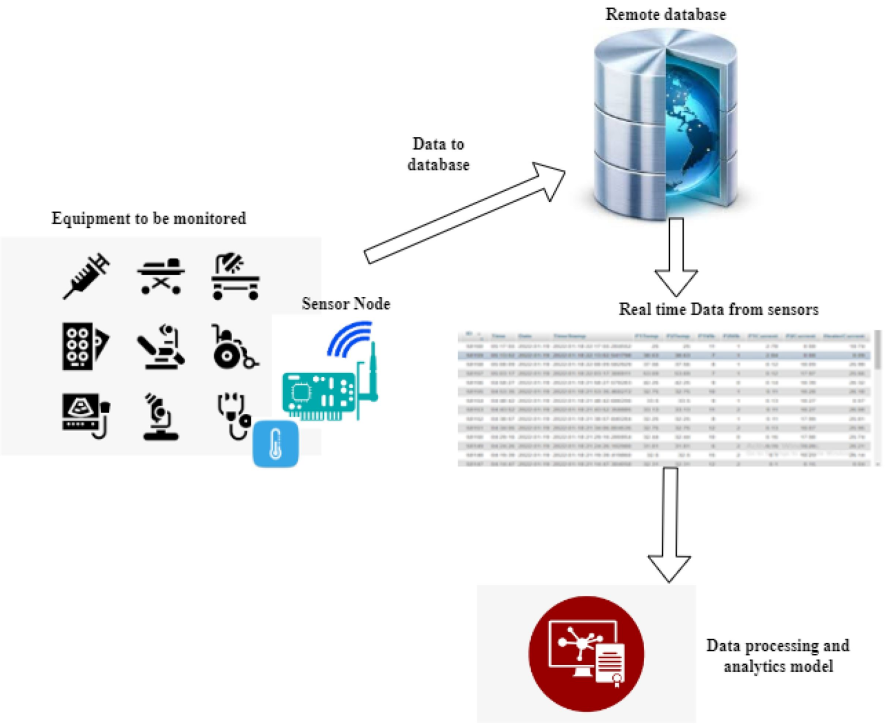
MicroPython

EDGE IMPULSE

https://youtu.be/jOZKxO3KnIA

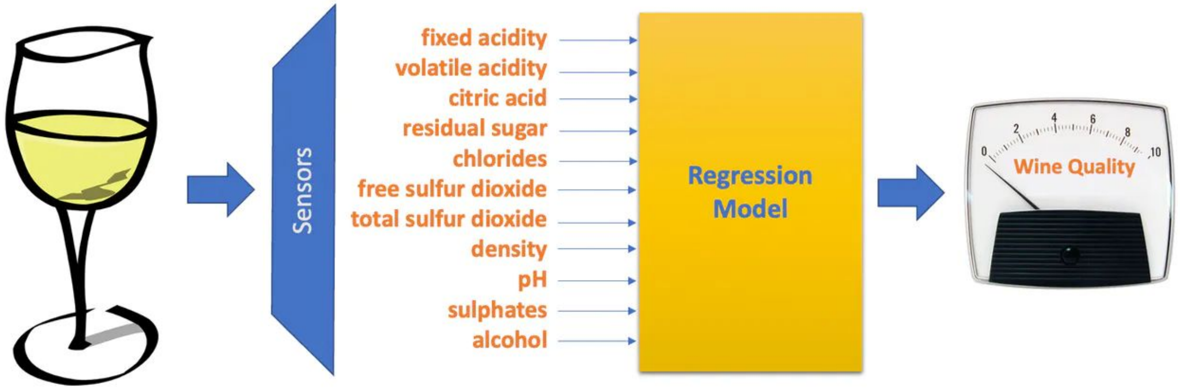
Regression on TinyML





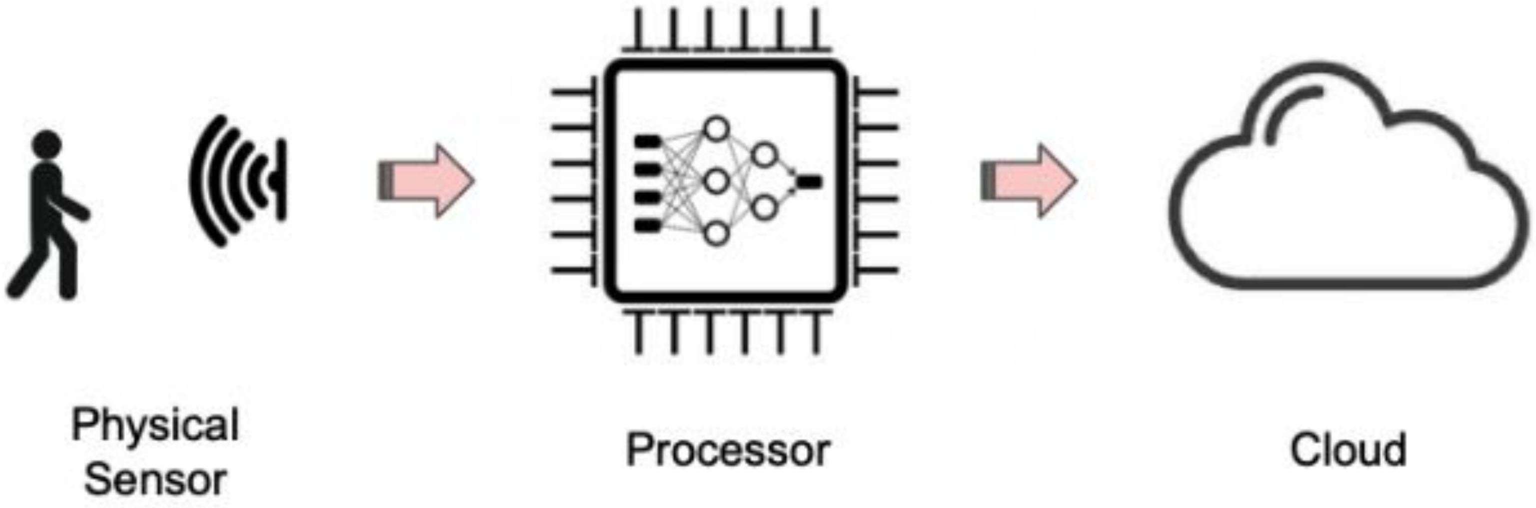
[On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse]()

Sensor fusion

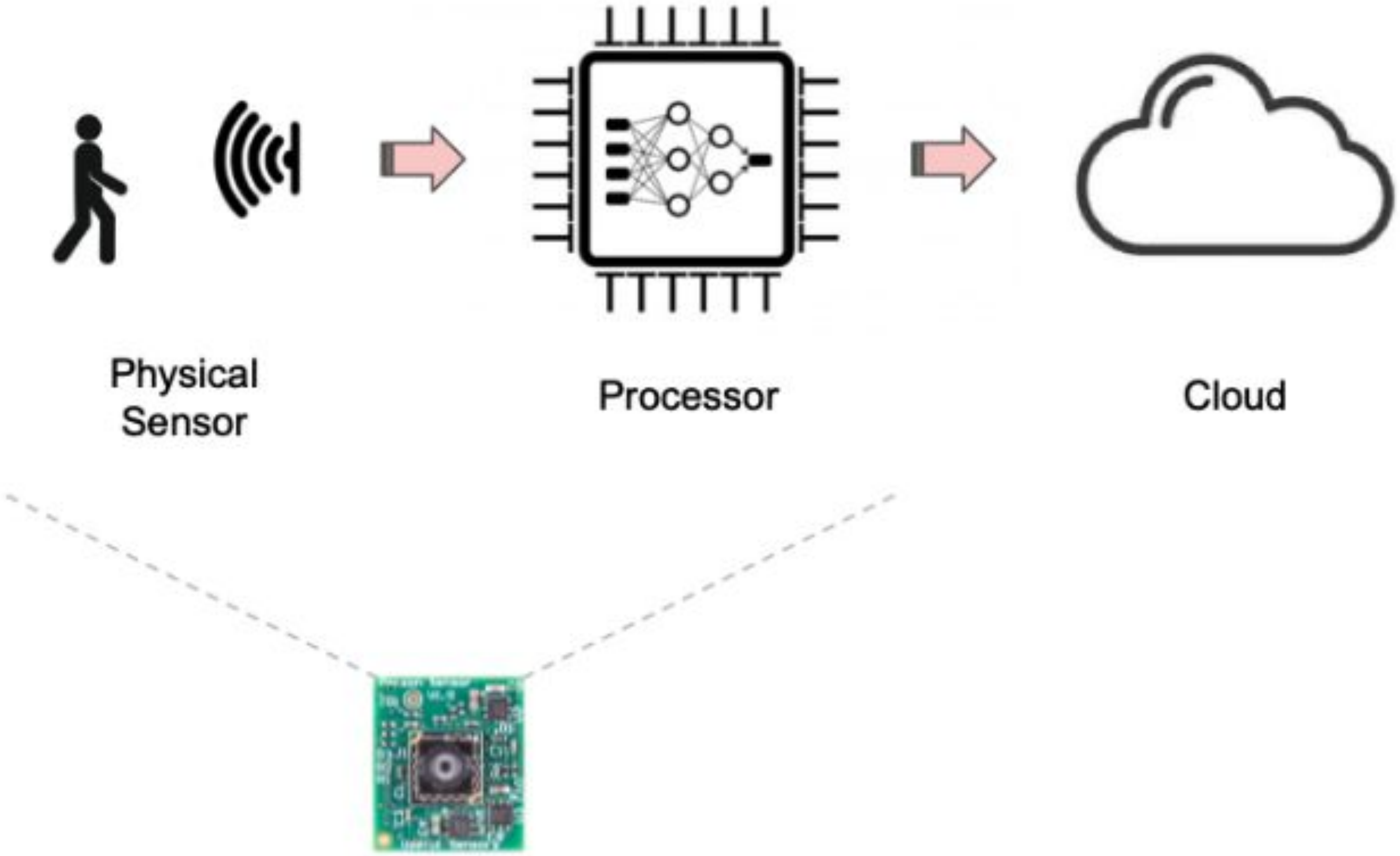


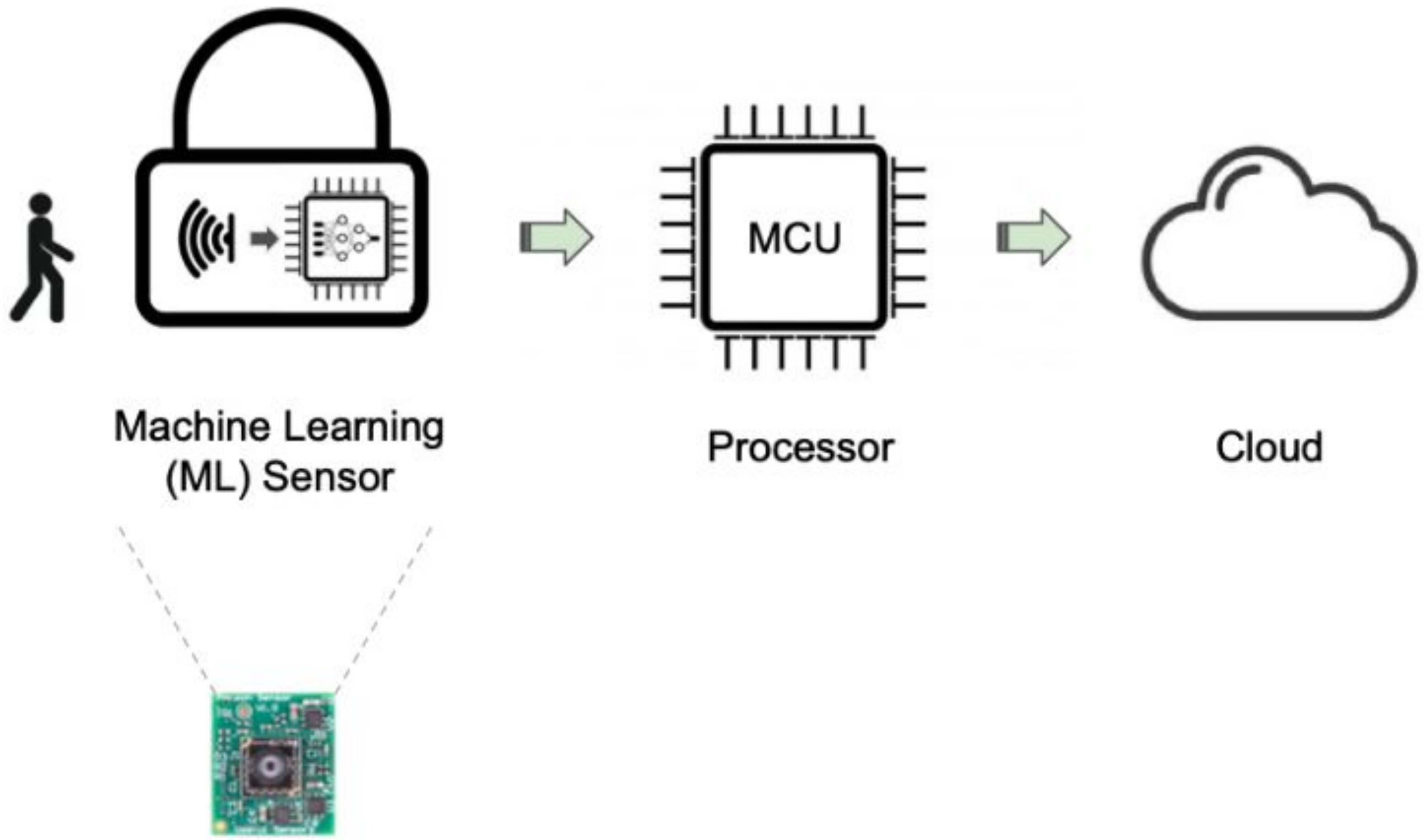
[TinyML Made Easy: Exploring Regression - White Wine Quality]()

The Future of the EdgeAI/ML



Sensor 1.0


Physical
Sensor
Processor
Cloud



Sensor 2.0

useful sensors

The Person Sensor has built in facial recognition and determines how many people there are, as well as their relative position.

[USD 10 -> https://www.sparkfun.com/products/21231](https://www.sparkfun.com/products/21231)

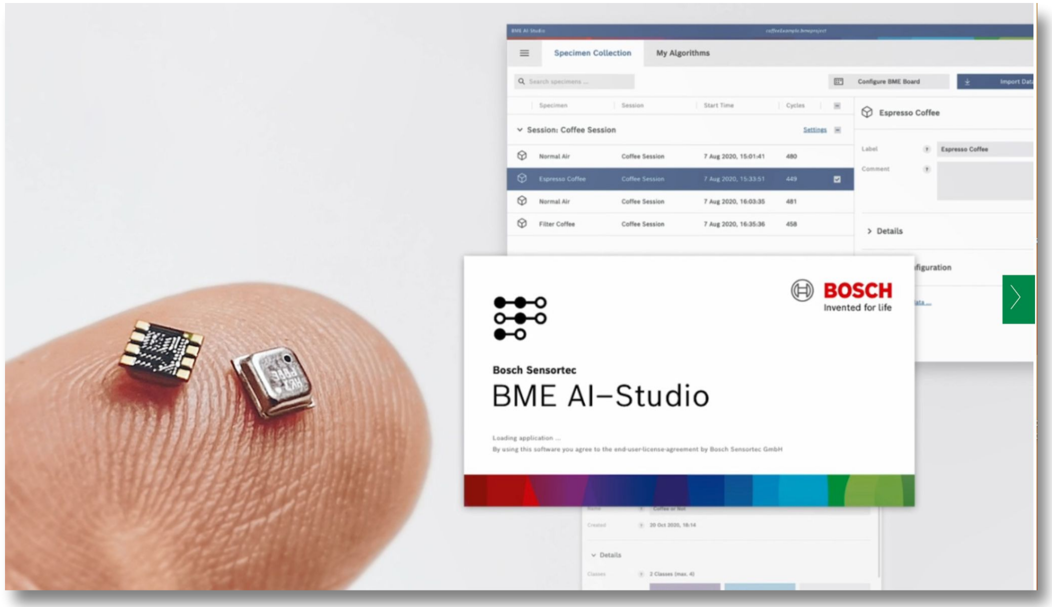
Bosch BME688 - Environmental sensing with AI

Relative humidity barometric pressure

Excellent temperature stability

Humidity

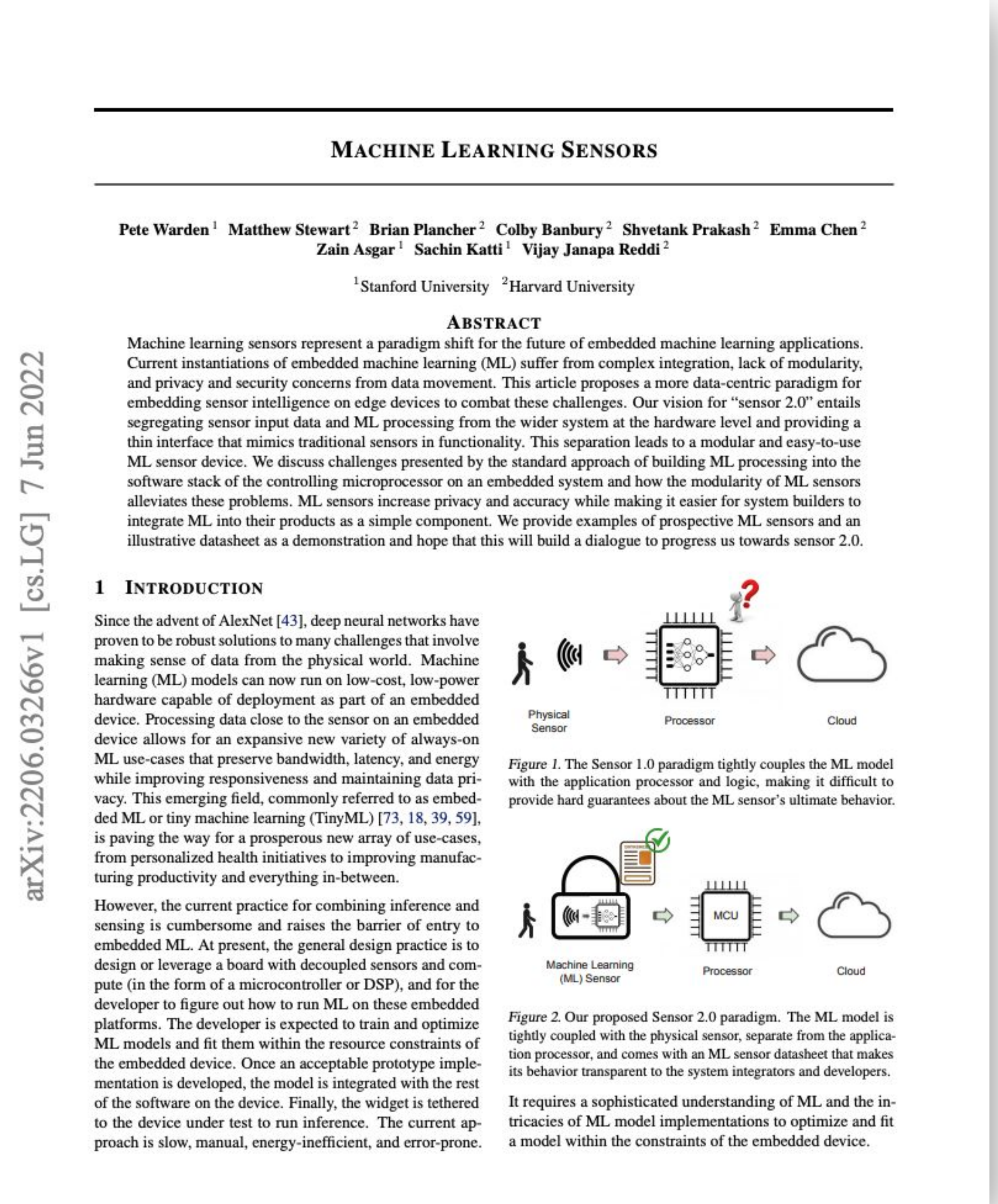
Gas sensing



https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme688/

mlsensors.org

https://github.com/harvard-edge/ML-Sensors



MACHINE LEARNING SENSORS

Pete Warden [1] Matthew Stewart [2] Brian Plancher [2] Colby Banbury [2] Shvetank Prakash [2] Emma Chen [2]
Zain Asgar [1] Sachin Katti [1] Vijay Janapa Reddi [2]

[1]Stanford University [2]Harvard University

arXiv:2206.03266v1 [cs.LG] 7 Jun 2022

ABSTRACT

Machine learning sensors represent a paradigm shift for the future of embedded machine learning applications. Current instantiations of embedded machine learning (ML) suffer from complex integration, lack of modularity, and privacy and security concerns from data movement. This article proposes a more data-centric paradigm for embedding sensor intelligence on edge devices to combat these challenges. Our vision for "sensor 2.0" entails segregating sensor input data and ML processing from the wider system at the hardware level and providing a thin interface that mimics traditional sensors in functionality. This separation leads to a modular and easy-to-use ML sensor device. We discuss challenges presented by the standard approach of building ML processing into the software stack of the controlling microprocessor on an embedded system and how the modularity of ML sensors alleviates these problems. ML sensors increase privacy and accuracy while making it easier for system builders to integrate ML into their products as a simple component. We provide examples of prospective ML sensors and an illustrative datasheet as a demonstration and hope that this will build a dialogue to progress us towards sensor 2.0.

1 INTRODUCTION

Since the advent of AlexNet [43], deep neural networks have proven to be robust solutions to many challenges that involve making sense of data from the physical world. Machine learning (ML) models can now run on low-cost, low-power hardware capable of deployment as part of an embedded device. Processing data close to the sensor on an embedded device allows for an expansive new variety of always-on ML use-cases that preserve bandwidth, latency, and energy while improving responsiveness and maintaining data privacy. This emerging field, commonly referred to as embedded ML or tiny machine learning (TinyML) [73, 18, 39, 59], is paving the way for a prosperous new array of use-cases, from personalized health initiatives to improving manufacturing productivity and everything in-between.

However, the current practice for combining inference and sensing is cumbersome and raises the barrier of entry to embedded ML. At present, the general design practice is to design or leverage a board with decoupled sensors and compute (in the form of a microcontroller or DSP), and for the developer to figure out how to run ML on these embedded platforms. The developer is expected to train and optimize ML models and fit them within the resource constraints of the embedded device. Once an acceptable prototype implementation is developed, the model is integrated with the rest of the software on the device. Finally, the widget is tethered to the device under test to run inference. The current approach is slow, manual, energy-inefficient, and error-prone.

Physical Sensor
Processor
Cloud

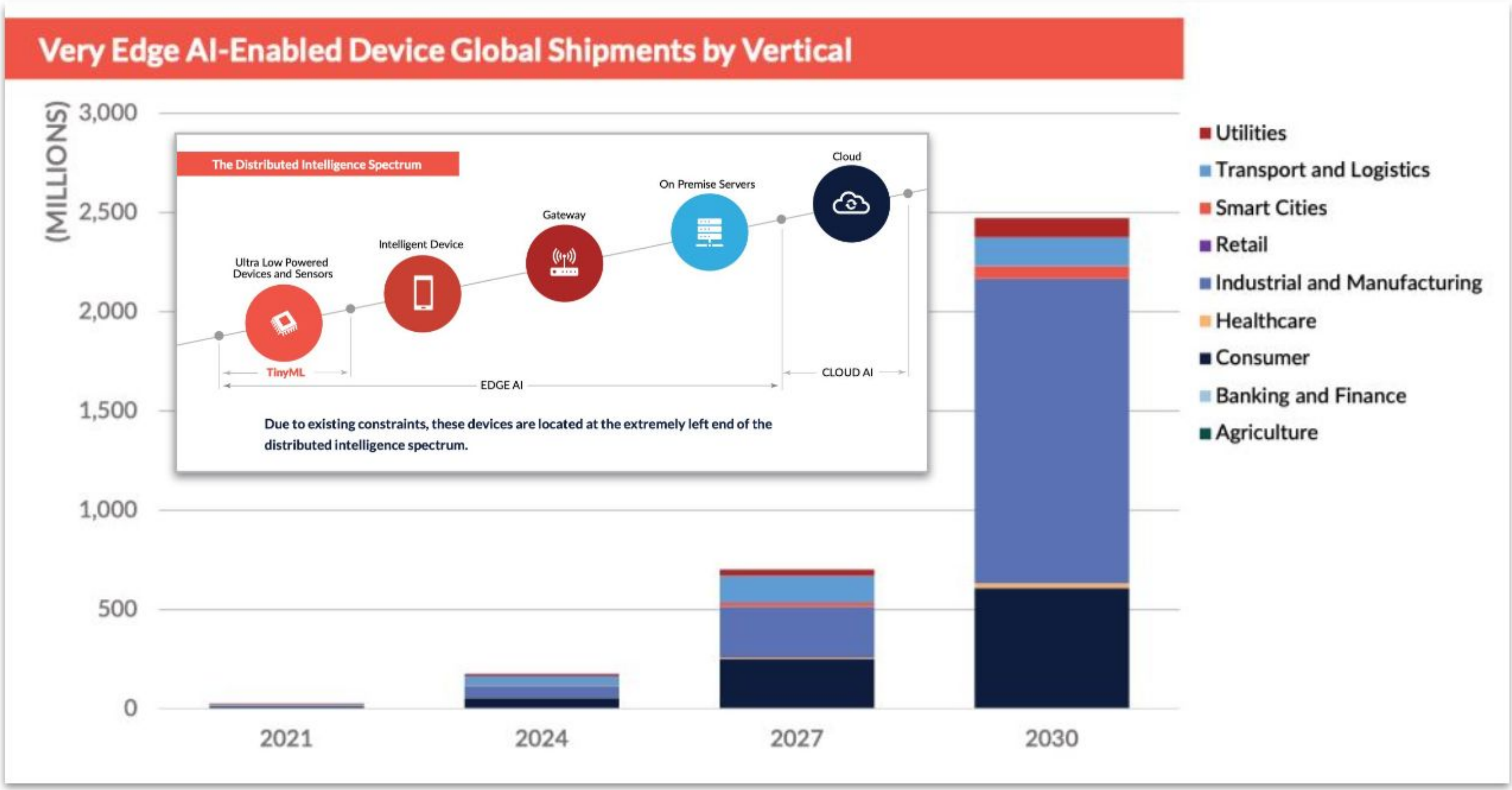
Figure 1. The Sensor 1.0 paradigm tightly couples the ML model with the application processor and logic, making it difficult to provide hard guarantees about the ML sensor's ultimate behavior.

MCU
Machine Learning (ML) Sensor
Processor
Cloud

Figure 2. Our proposed Sensor 2.0 paradigm. The ML model is tightly coupled with the physical sensor, separate from the application processor, and comes with an ML sensor datasheet that makes its behavior transparent to the system integrators and developers.

It requires a sophisticated understanding of ML and the intricacies of ML model implementations to optimize and fit a model within the constraints of the embedded device.

Massive Potential for Impact



Source: ABI Research: TinyML

Edge AI has a very high impact potential, and it is for now!

Conclusion

The Future of ML is Tiny and Bright

***Vijay Janapa Reddi**, Ph. D. | Associate Professor |*
John A. Paulson School of Engineering and Applied Sciences | Harvard University |

Responsible AI

Suzan Kennedy, Ph.D.

Santa Clara University

[SciTinyML Seminar - Slides]

[SciTinyML Seminar - Video]

SciTinyML
Scientific Use of Machine Learning on Low-Power Devices
October 18-22 2021

To learn more …

- IESTI01 TinyML - Machine Learning for Embedding Devices (Videos: Pt)
- WALC 22 – Applied AI - TinyML (Videos in Spanish)
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
- "Deep Learning with Python" book by François Chollet
- "TinyML" book by Pete Warden, Daniel Situnayake
- "TinyML Cookbook" by Gian Marco Iodice
- "AI at the Edge" book by Daniel Situnayake, Jenny Plunkett

On the TinyML4D website, You can find lots of educational materials on TinyML. They are all free and open-source for educational uses – we ask that if you use the material, please cite them! TinyML4D is an initiative to make TinyML education available to everyone globally.

Thanks

UNIFEI