



TinyML: Machine Learning meets the Internet of Things

Marco Zennaro, PhD

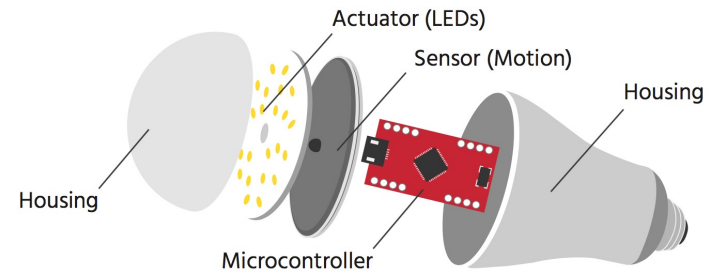
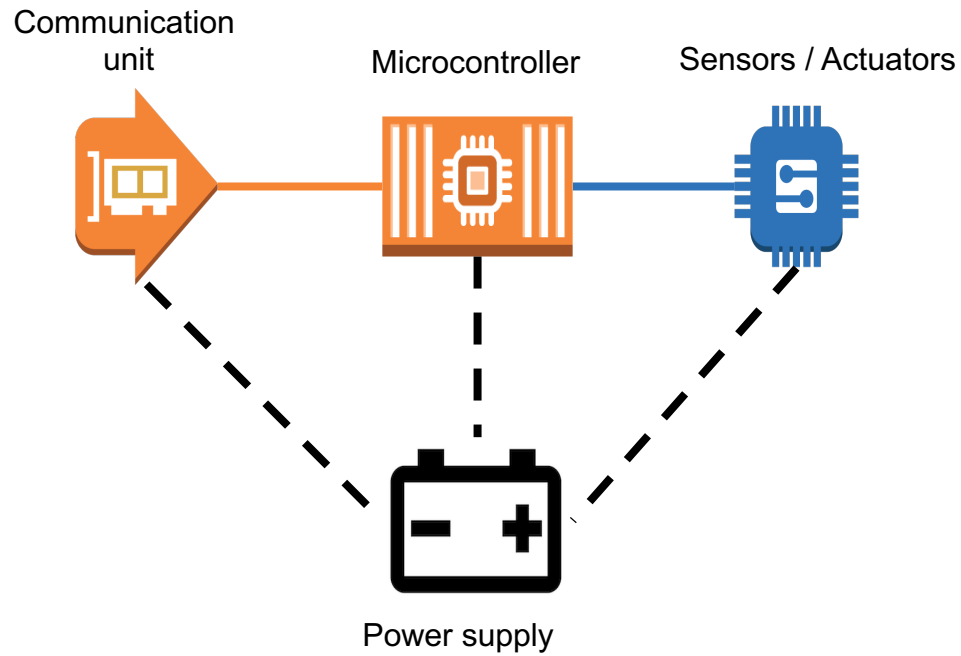
- The Abdus Salam International Centre for Theoretical Physics – Trieste - Italy
- mzennaro@ictp.it
- [@marcozennaro](https://twitter.com/marcozennaro)

The slides are based on Pietro Manzoni's course "IoT and edge computing: stories of a necessary relationship".

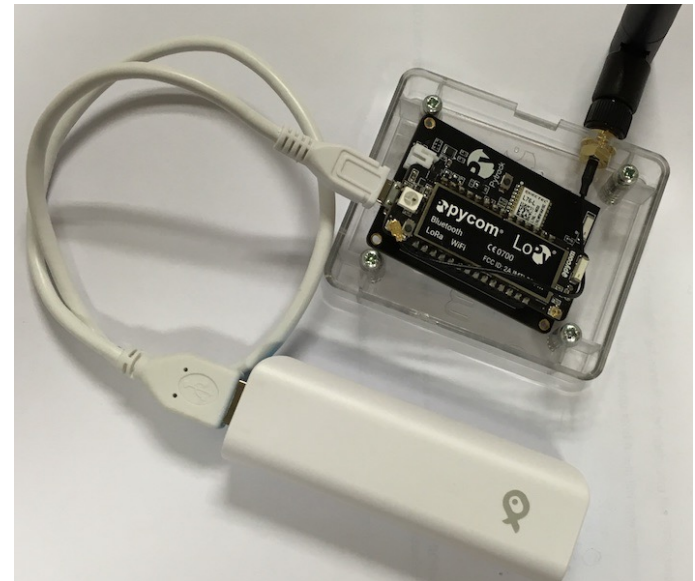
Pietro Manzoni is Professor at Universitat Politècnica de València (UPV).

<http://grc.webs.upv.es/>

- The **Internet of Things** (IoT) refers to a system of interrelated, Internet-connected objects ("*things*"), that can collect and transfer data.
- **Machine Learning** is typically associated with computationally heavy cloud-based solutions with relatively high latencies, high power consumption, and the need for high bandwidths links.
- **With TinyML, IoT and ML come together by shrinking deep learning networks to fit on tiny hardware, thus requiring low energy, low bandwidth links, and reduced latencies.**



A Reference Guide to the Internet of Things Copyright © 2017 Bridgera LLC, RIoT

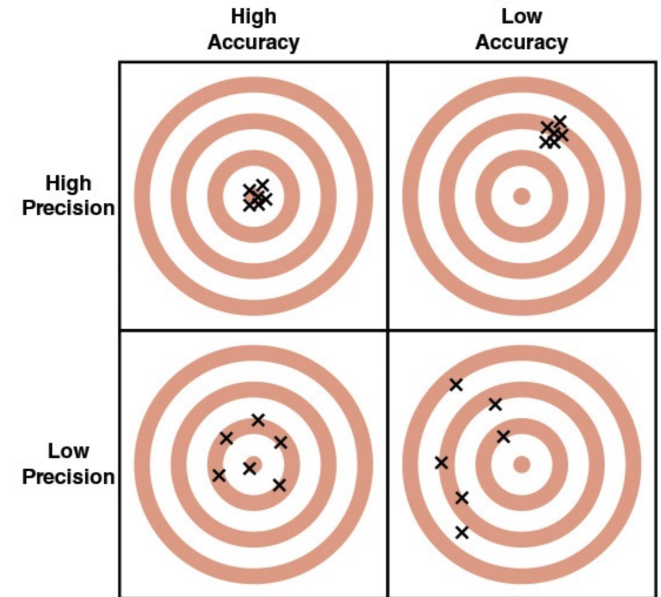
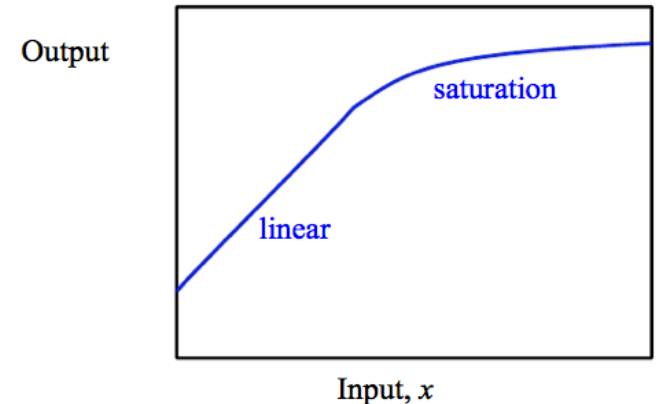


- They are mainly input components in IoT; are devices that receive a stimulus and responds with an electrical signal
- A **transfer function** for a sensor represents the input-output relation
 - Input: a physical measured parameter
 - Output: usually an electrical output signal.
- The simplest form of transfer function is a linear function which can be described as follows.

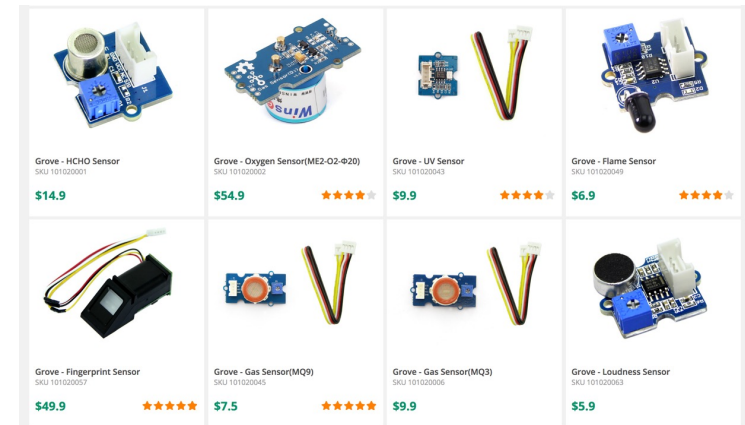
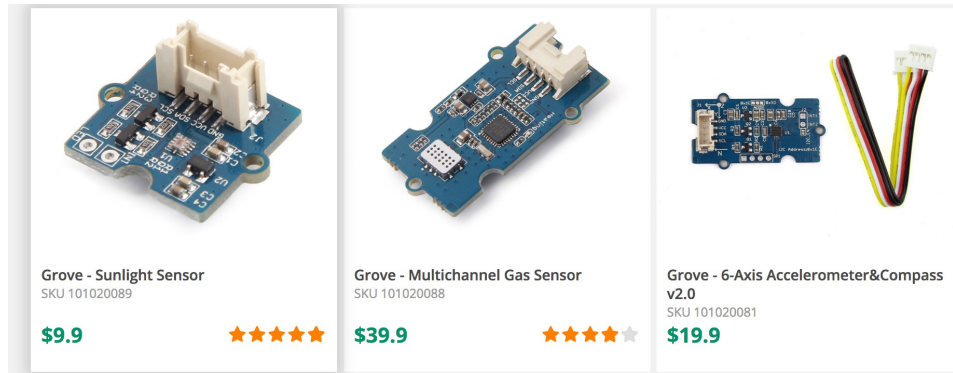
$$S = a + bx$$

- where **x** is the input, **b** is the slope (and sometimes called sensitivity), and **a** is the offset (or the output when the input is zero).

- **Saturation:**
 - The output becomes flat at some levels of input
- **Resolution:**
 - Describe the smallest increment of stimulus
 - Outputs change in small steps
- **Accuracy:**
 - Describes how close the measurement is to the true value
- **Precision or Repeatability:**
 - Describes the difference in results when measurements are taken repeatedly under the same conditions



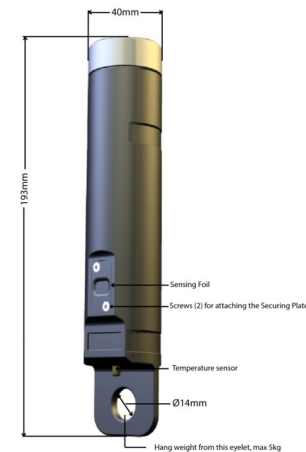
cheap...



expensive...



1.2 Oxygen Optode 4531 dimensions



Parameter	Output	Default range ²⁾	Calibrated range	Accuracy	Resolution
Oxygen Concentration	0 - 5V	0 to 800 μ M	0 to 500 μ M	<8 μ M or 5% whichever is greater	< 1 μ M
	4 - 20mA	0 to 800 μ M	0 to 500 μ M	<9 μ M or 5.2% whichever is greater	< 1 μ M
Oxygen Saturation	0 - 5V	0 - 200%	0 - 120%	<5 %	<0.4%
	4 - 20mA	0 - 200%	0 - 120%	<5.2 %	<0.4%
Temperature	0 - 5V	-5 to + 35°C	0 - 36°C	$\pm$ 0.1°C	$\pm$ 0.01°C
	4 - 20mA	-5 to + 35°C	0 - 36°C	$\pm$ 0.15°C	$\pm$ 0.02°C

- Today, the **most common power source is a battery**, but there are several other possibilities for power, such as **solar cells, piezoelectricity, radio-transmitted energy, and other forms of power scavenging**.
- Rechargeable batteries are not particularly well-suited to smart objects.
 - Instead of using rechargeable batteries, battery-equipped smart objects are typically designed so a single battery should last the entire lifetime of the smart object.



LiPo



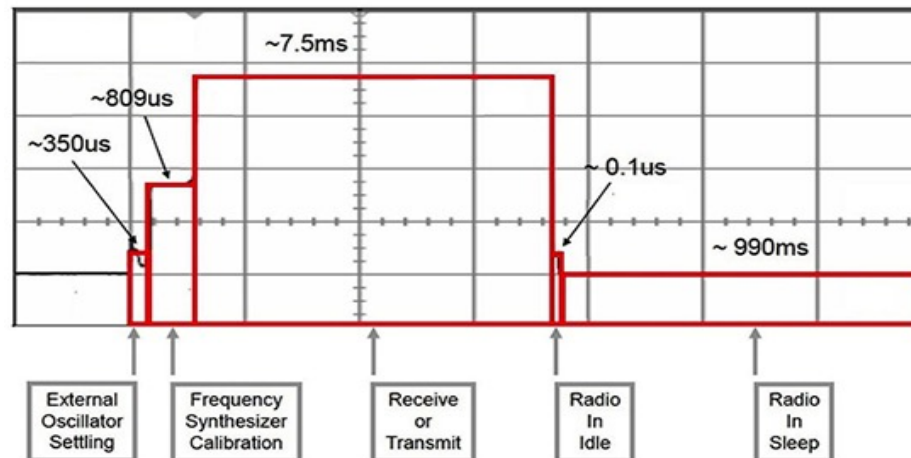
LiPo



Li-ion

Lithium Ion (Li-ion)
Lithium-Ion Polymer (Li-Po)

- Of the hardware components of a “thing”, the radio is usually the most power-consuming component.
 - Compared to the power consumption of the microcontroller or the sensors, the radio transceiver often uses ten times as much power.
- For low-power radios, only a small portion of the power consumption is used to send the radio signal into the air.
 - ➔ listening is as power consuming as sending.

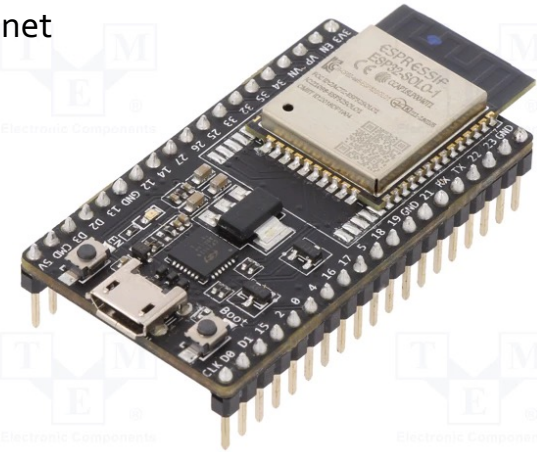


Microcontroller: ESP32

- The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations. ESP32 is created and developed by Espressif System
- CPU: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS
- Memory: 520 KiB SRAM
- Wireless connectivity:
 - Wi-Fi: 802.11 b/g/n
 - Bluetooth: v4.2 BR/EDR and BLE
- Peripheral interfaces:
 - 12-bit SAR ADC up to 18 channels
 - 2 × 8-bit DACs
 - 10 × touch sensors (capacitive sensing GPIOs)
 - ...

<https://www.espressif.com/en/products/socs/esp32>

<http://esp32.net>



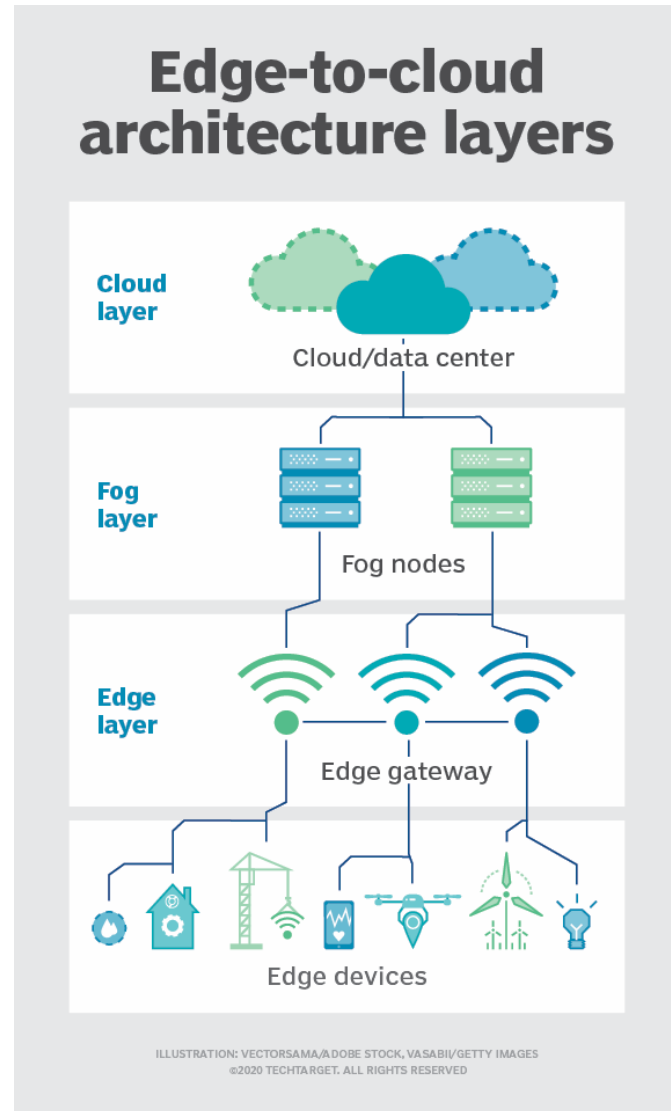
Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

Power mode	Description		Power consumption
Modem-sleep	The CPU is powered on.	240 MHz * Dual-core chip(s)	30 mA ~ 68 mA
		Single-core chip(s)	N/A
		160 MHz * Dual-core chip(s)	27 mA ~ 44 mA
		Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz Dual-core chip(s)	20 mA ~ 31 mA
		Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-		0.8 mA
Deep-sleep	The ULP coprocessor is powered on.		150 μ A
	ULP sensor-monitored pattern		100 μ A @1% duty
	RTC timer + RTC memory		10 μ A
Hibernation	RTC timer only		5 μ A
Power off	CHIP_PU is set to low level, the chip is powered off.		1 μ A

What does that all mean?



- A possible definition of **edge computing** is: “a part of a distributed computing topology in which information processing is located close to the edge — where things and people produce or consume that information.”



Advantages of Edge Computing... for IoT



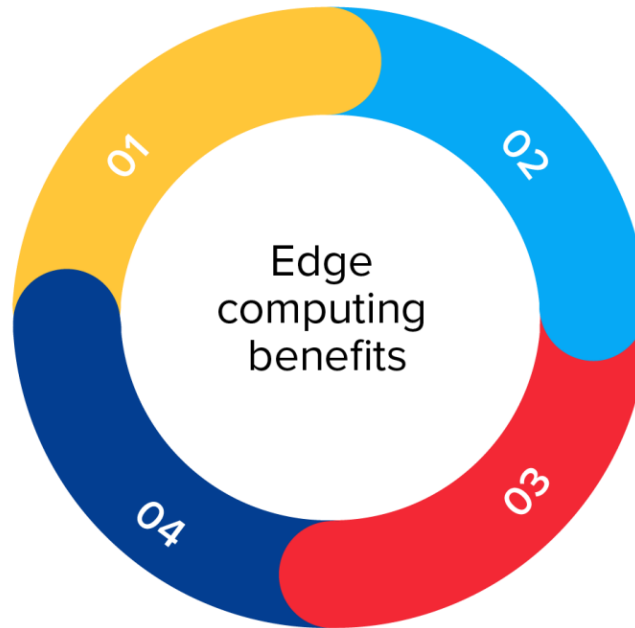
Latency

Reduction of latency by processing the data closer to the customer



Security → Privacy

Computing at the edge provides more security than computing at the cloud because it is less vulnerable to numerous variety of threats due to its scope



Bandwidth

Sending data from edge to the cloud takes up spectral resources; there's just not enough bandwidth for data transportation

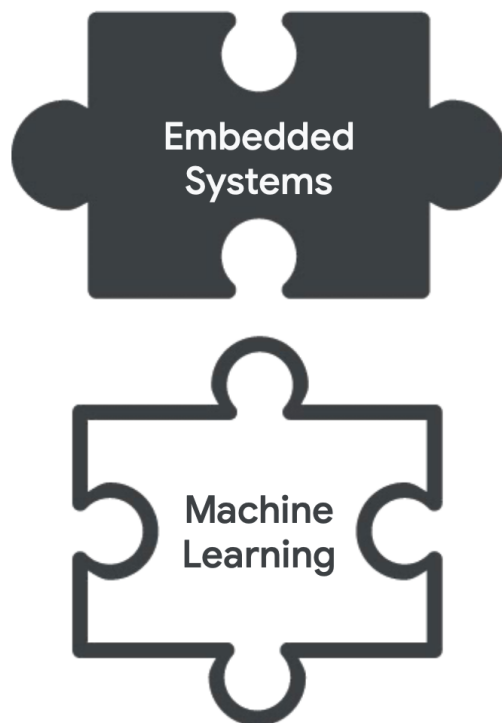


Reliability

By processing data at the edge, you eliminate network reliability problems

<https://www.wwt.com/article/show-me-the-money-drive-new-revenue-streams-with-edge-computing>

What is TinyML?



Original definition: "...a neural network model that runs at an energy cost of below 1 mW."

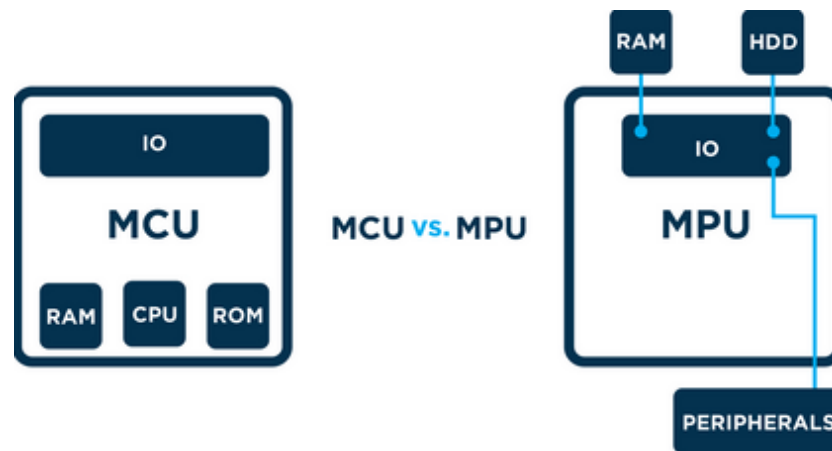
© "TinyML" by Pete Warden, Daniel Situnayake



TinyML

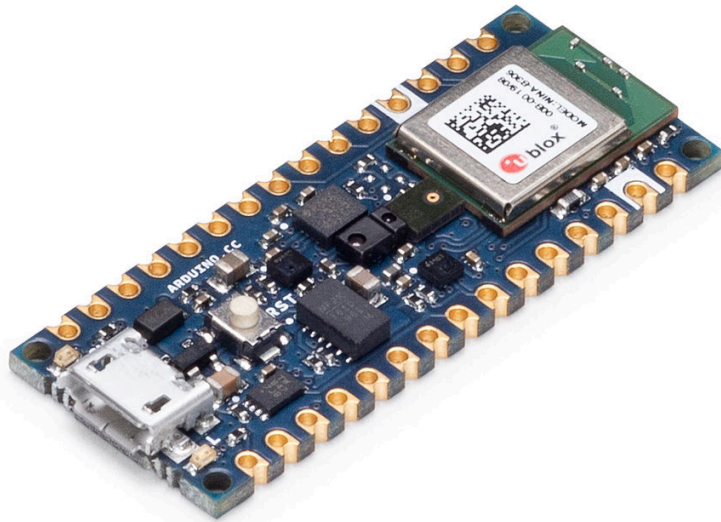
"Tiny machine learning (TinyML) is a fast-growing field of machine learning technologies and applications including algorithms, hardware, and software capable of performing on-device sensor data analytics at extremely low power consumption, **typically in the mW range and below**, enabling a variety of always-on ML use-cases **on battery-operated devices**."

- A microcontroller is a microprocessor with built-in memory, timers, and hardware for connecting external devices such as sensors, actuators, and radio transceivers.
- Typically, a smart object microcontroller has a few kilobytes of on-chip memory and is run at a clock speed of a few megahertz.

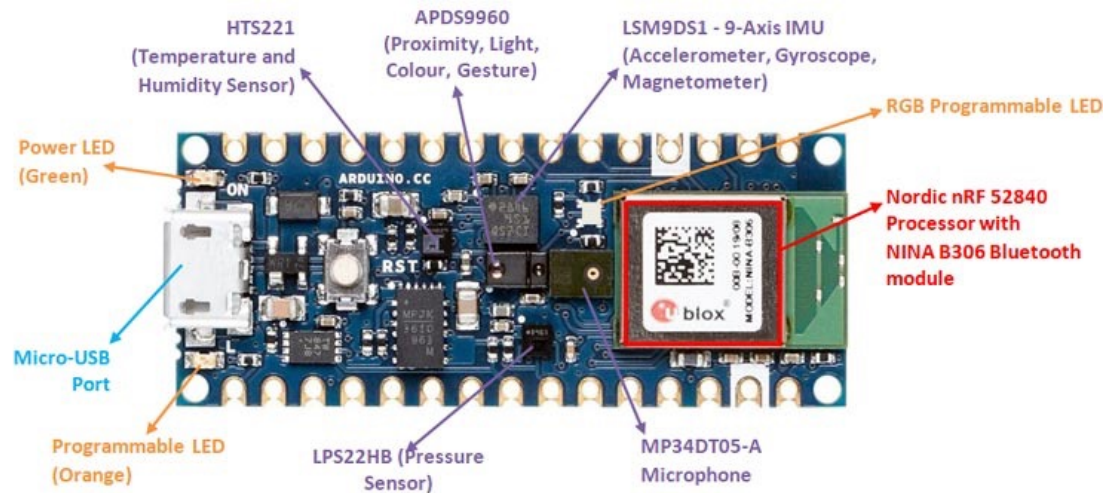


	Microprocessor	>	Microcontroller
Platform			
Compute	1GHz–4GHz	~10X	1MHz–400MHz
Memory	512MB–64GB	~10000X	2KB–512KB
Storage	64GB–4TB	~100000X	32KB–2MB
Power	30W–100W	~1000X	150μW–23.5mW

Most widely used HW for TinyML (currently)



Arduino Nano 33 BLE Sense
64 MHz Arm® Cortex-M4F (with FPU)
1 MB Flash + 256 KB RAM
Bluetooth® 5 multiprotocol radio





The future of health wrapped around your finger — monitoring your sleep, heart rate, activity, and temperature with personalized insights.

Applications of TinyML: the beginning

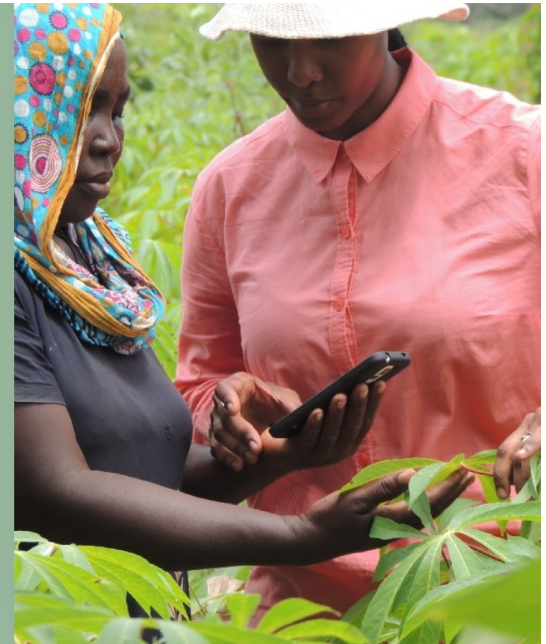


But the range of applications is wide...

**We help African
smallholder farmers
adapt to climate change**

Help us reach more farmers

Donate ❤️ & support farmers →



Plant Village: <https://plantvillage.psu.edu>

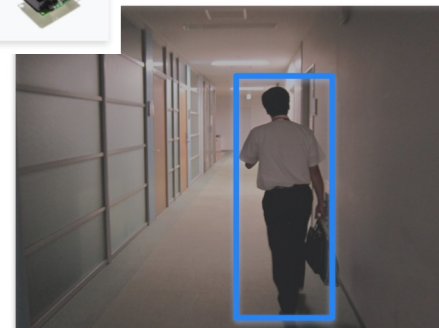
Sound



Vibration



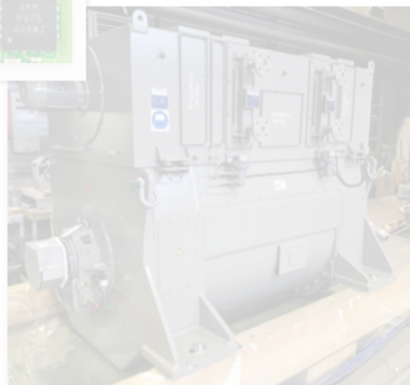
Vision



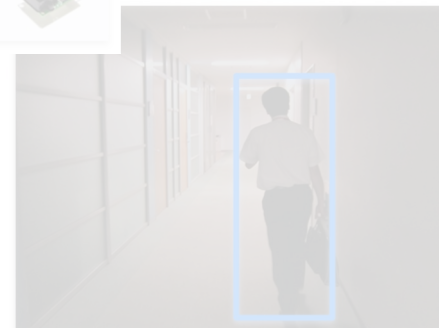
Sound



Vibration



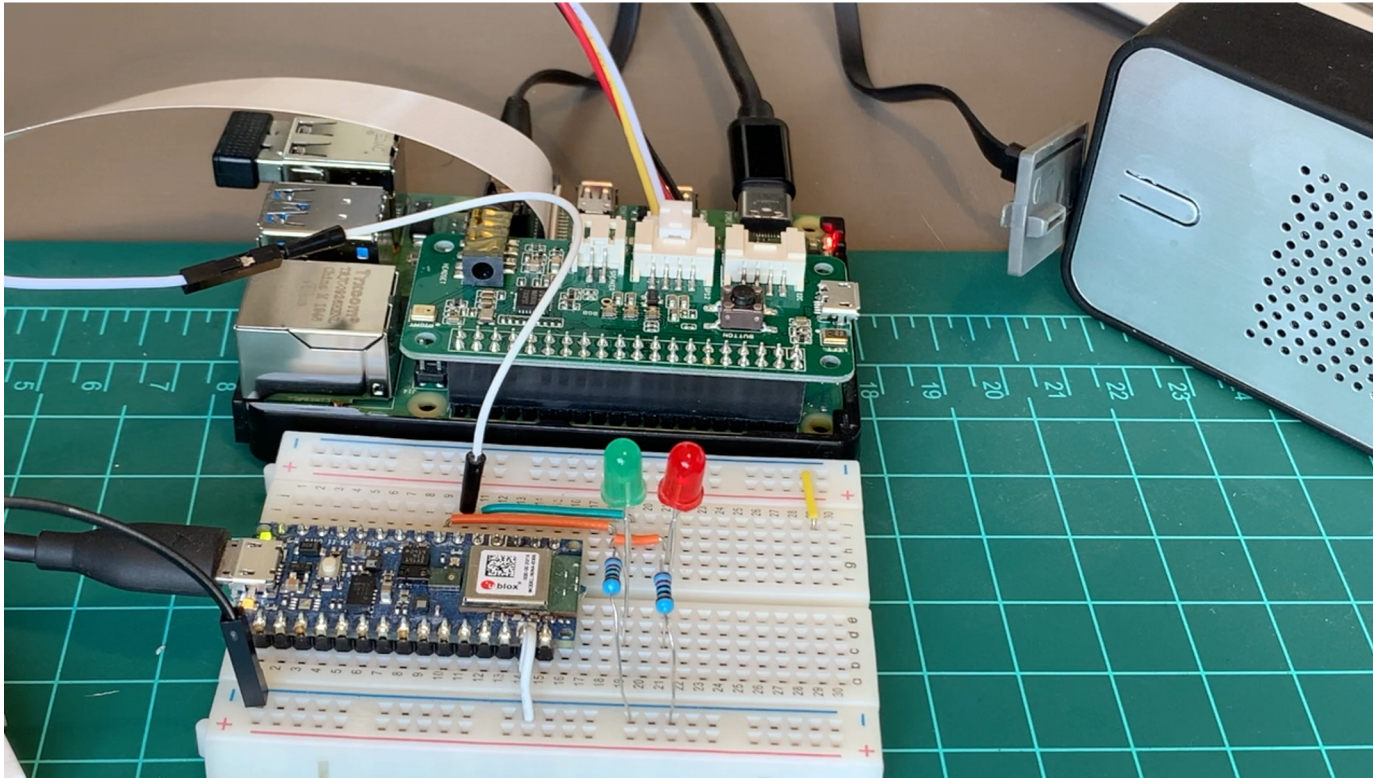
Vision



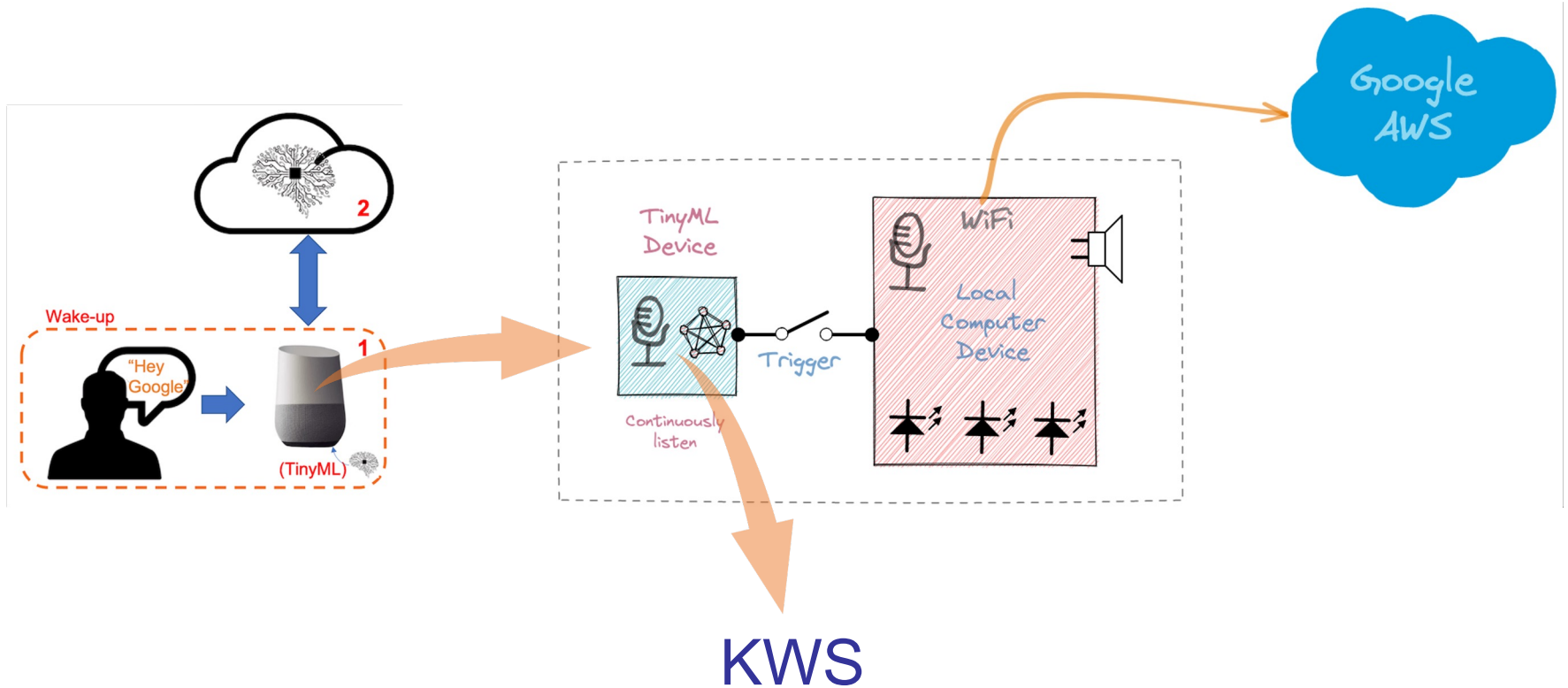
Personal Assistant



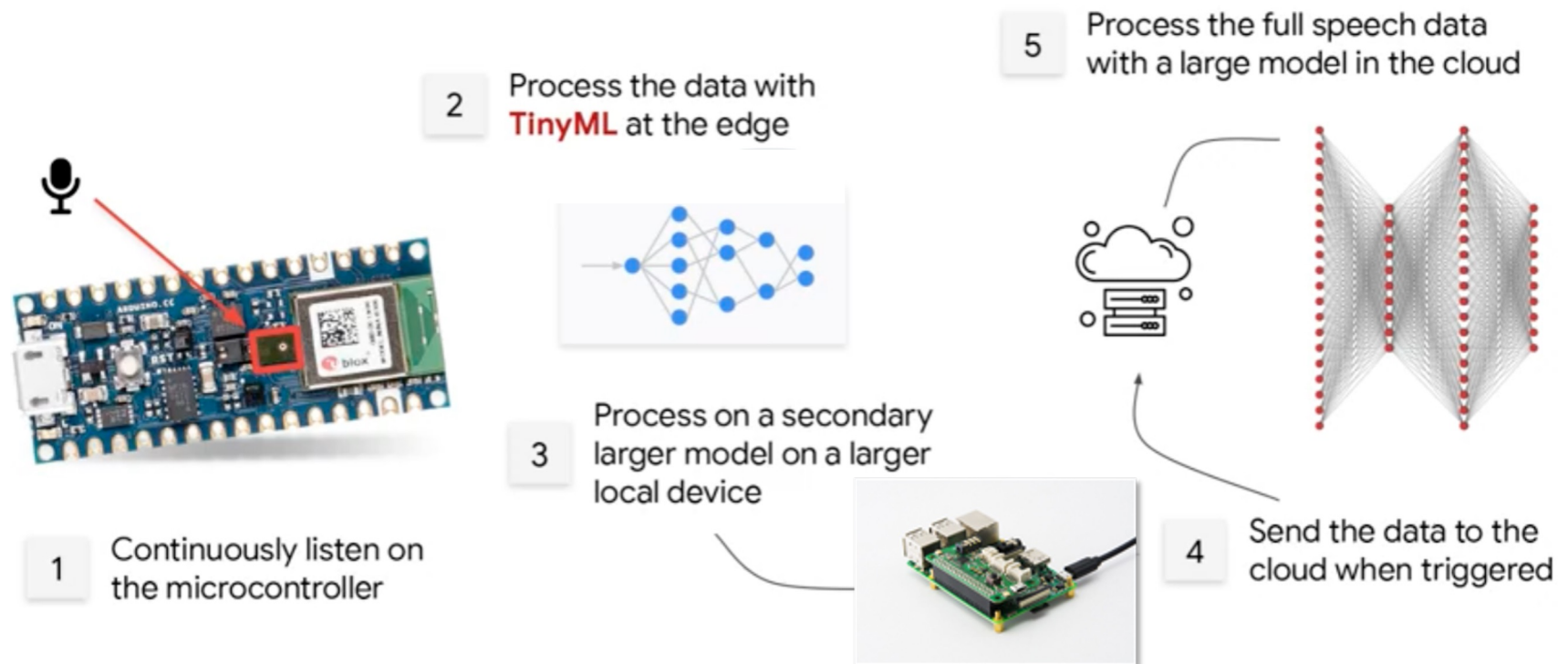
Personal Assistant



Personal Assistant



“Cascade” Detection: multi-stage model





Classifying mosquito wingbeat sound using TinyML

Moez Altayeb
University of Khartoum, Sudan
ICTP, Trieste, Italy
mohedahmed@hotmail.com

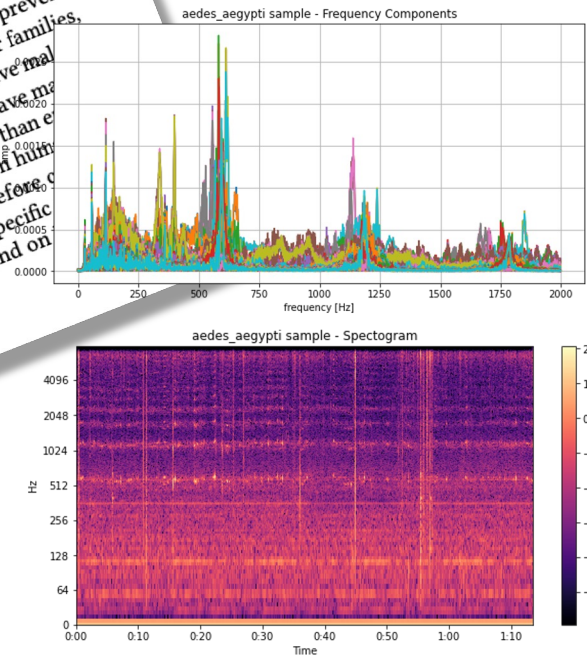
Marcelo Rovai
Universidade Federal de Itajubá
Itajubá, Brazil
rovai@unifei.edu.br

Marco Zennaro
ICTP
Trieste, Italy
mzennaro@ictp.it

ABSTRACT

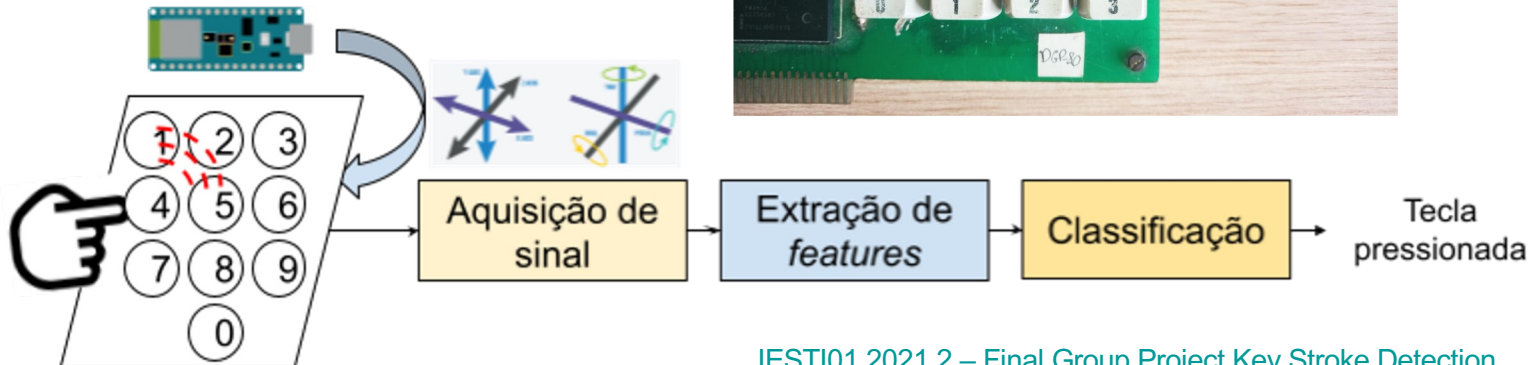
Every year more than one billion people are infected and more than one million people die from vector-borne diseases including malaria, dengue, zika and chikungunya. Mosquitoes are the best known disease vector and are geographically spread worldwide. It is important to raise awareness of mosquito proliferation by monitoring their incidence, especially in poor regions. Acoustic detection of mosquitoes has been studied for long and ML can be used to automatically identify mosquito species by their wingbeat. We present a prototype solution based on an openly available dataset, on the Edge Impulse platform and on three commercially-available TinyML devices. The proposed solution is low-power, low-cost and can run without human intervention in resource-constrained areas. This insect monitoring system can reach a global scale.

affected. People from poor communities with little access to health care and clean water sources are also at risk. Although anti-malarial drugs exist, there's currently no malaria vaccine. Vector-borne diseases also exacerbate poverty. Illness prevents people from working and supporting themselves and their families, impeding economic development. Countries with intensive malaria have much lower income levels than those that don't have malaria. Countries affected by malaria turn to control rather than prevention. Vector control means decreasing contact between human disease carriers on an area-by-area basis. It is therefore possible to be able to detect the presence of mosquitoes in a specific paper presents an approach based on TinyML and on embedded devices.

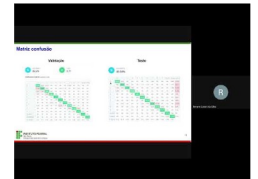


<https://www.edgeimpulse.com/blog/the-world-is-all-abuzz-about-tinyml>

Keystroke **Sound** Detection



[IESTI01 2021.2 – Final Group Project Key Stroke Detection](#)



Renam Castro
Professor IFESP

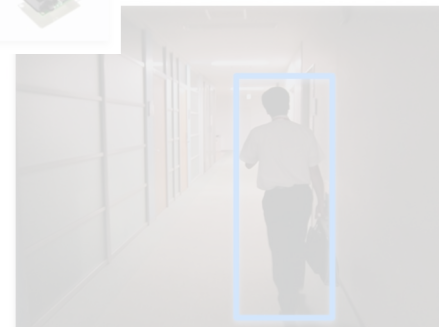
Sound



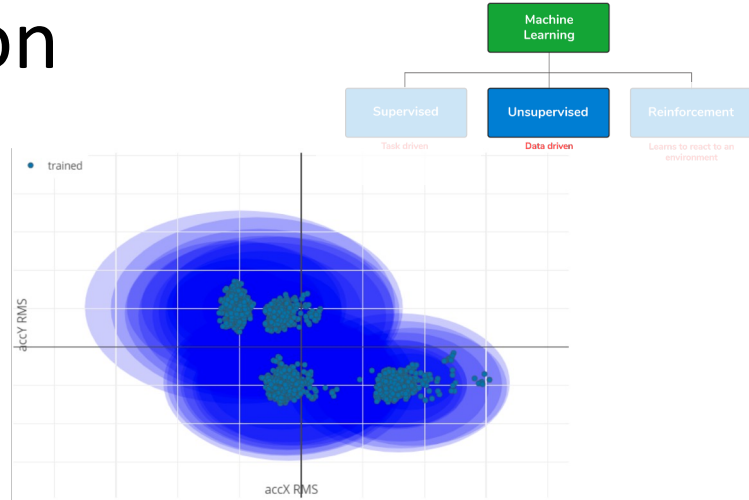
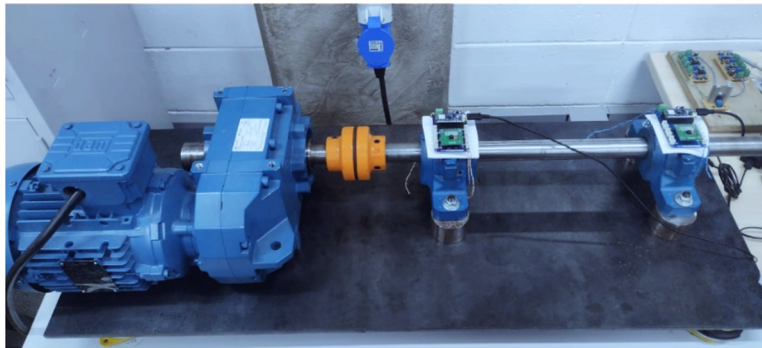
Vibration



Vision

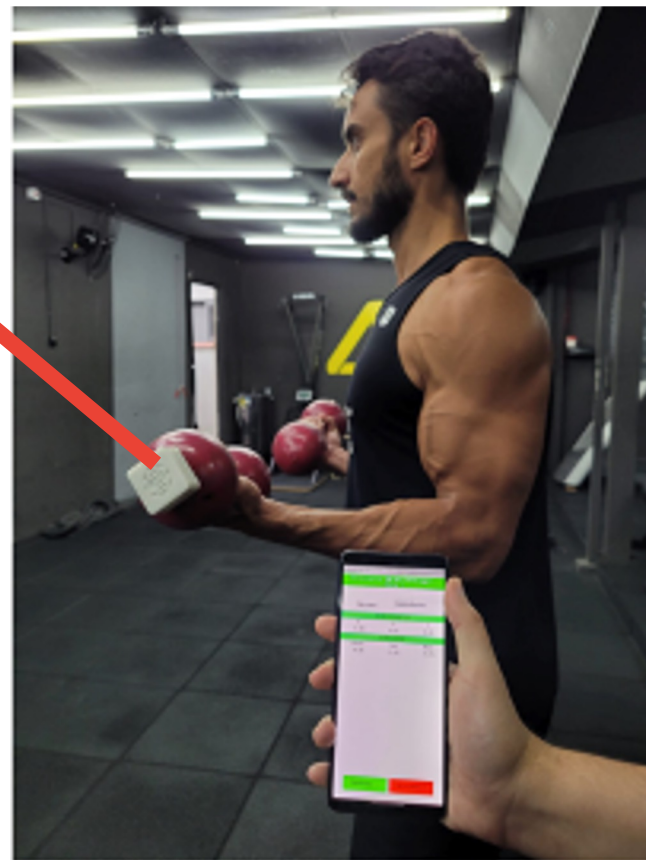
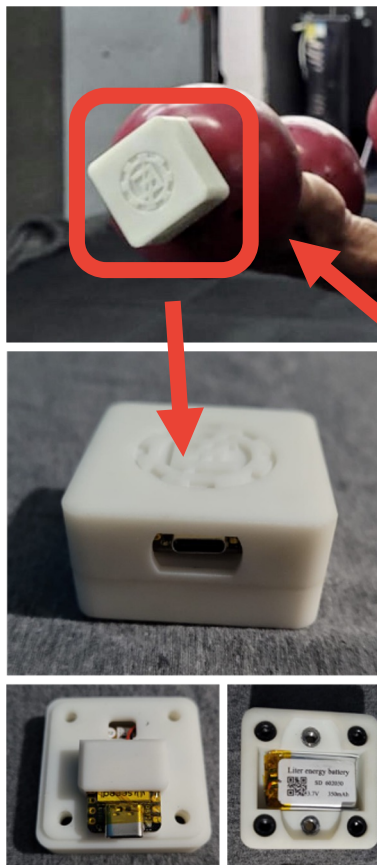


Industrial – Anomaly Detection



[IESTI01 2021.2 - Final Group Project: Bearing Failure Detection](#)

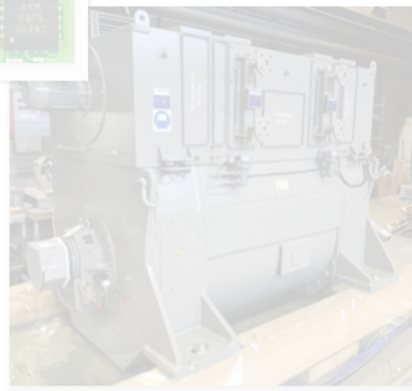
Movement Classification



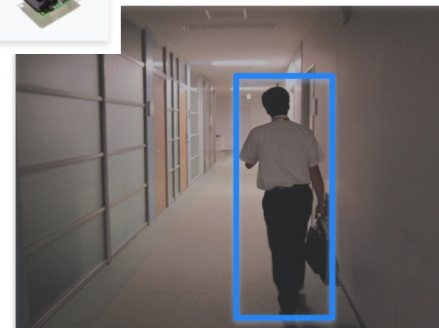
Sound



Vibration



Vision



Computer Vision Main Types

Image Classification (Multi-Class Classification)

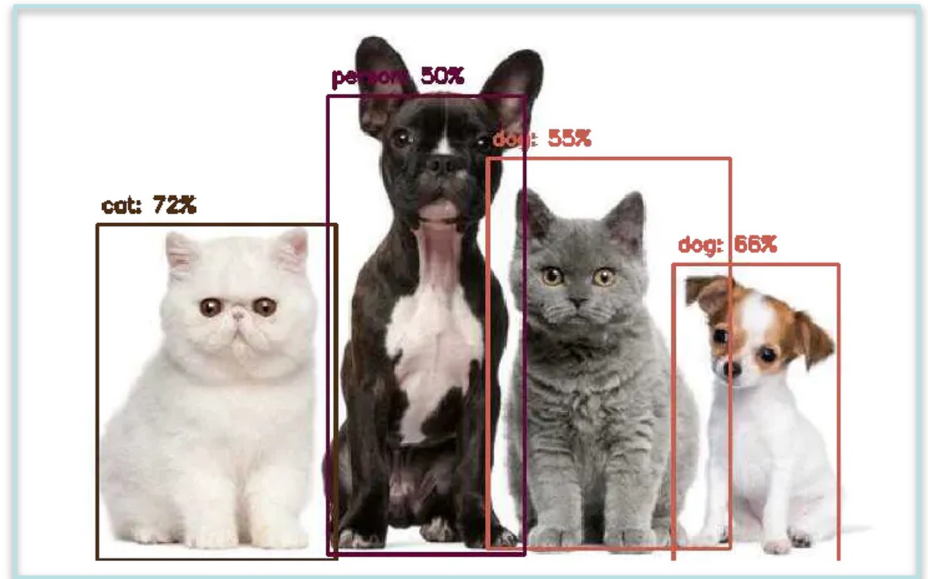


Cat: 70%



Dog: 80%

Object Detection Multi-Label Classification + Object Localization



Computer Vision Main Types

Image Classification (Multi-Class Classification)

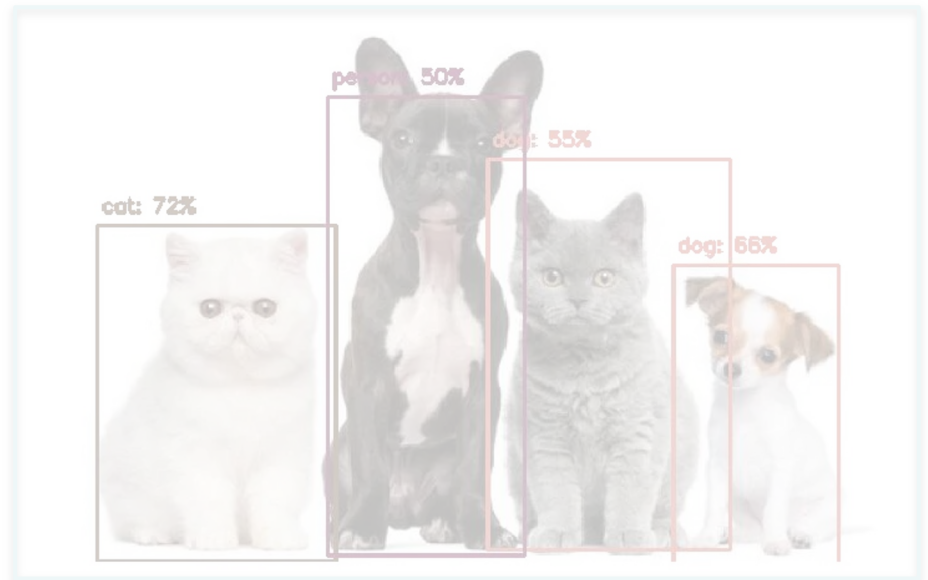


Cat: 70%



Dog: 80%

Object Detection Multi-Label Classification + Object Localization



Forest Fire Detection



[TinyML Aerial Forest Fire Detection](#)



[IESTI01 - Forest Fire Detection – Proof of Concept](#)

Coffee Disease Classification



<https://www.hackster.io/Yukio/coffee-disease-classification-with-ml-b0a3fc>



João Vitor Yukio Bordin Yamashita
Graduando em Engenharia Eletrônica pela UNIFEI

Computer Vision Main Types

Image Classification (Multi-Class Classification)

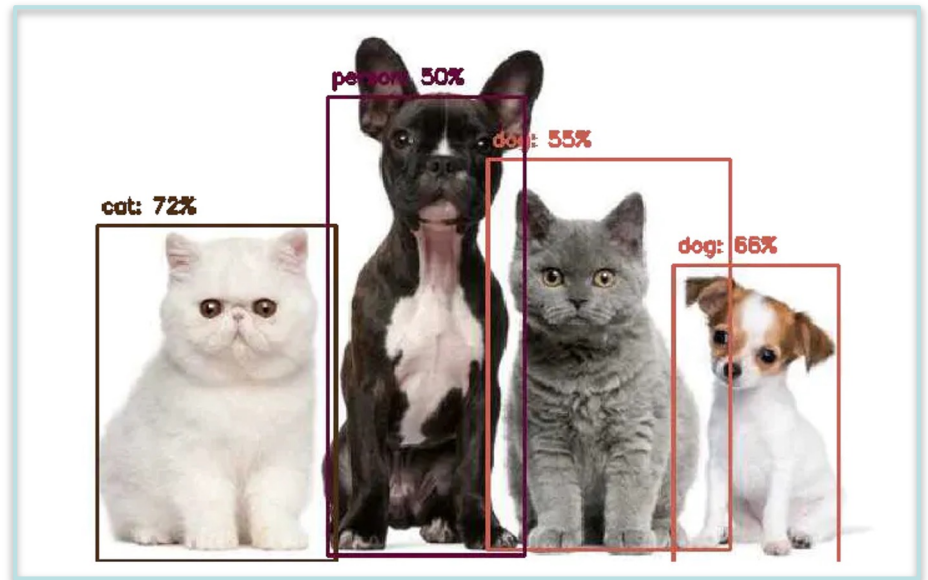


Cat: 70%

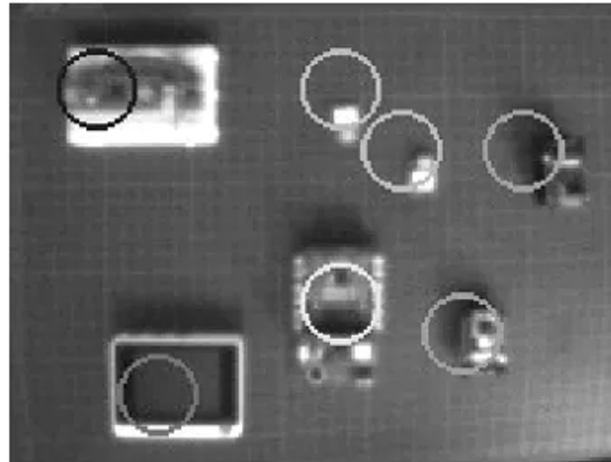
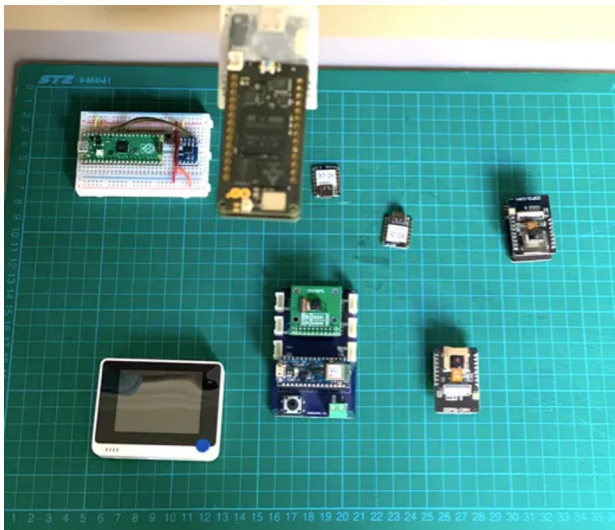


Dog: 80%

Object Detection Multi-Label Classification + Object Localization



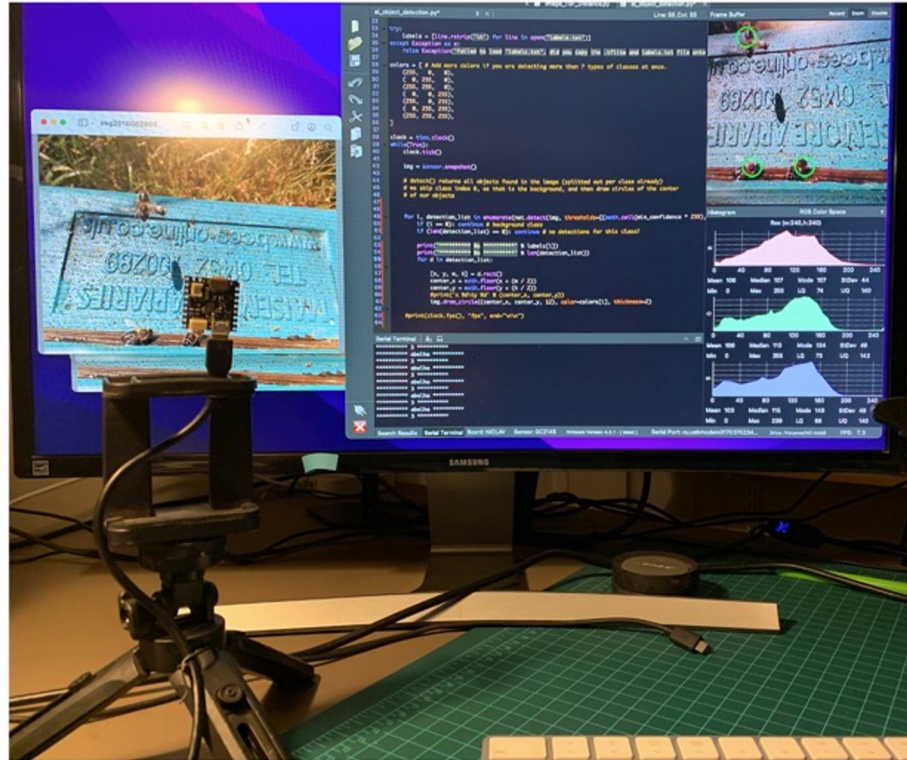
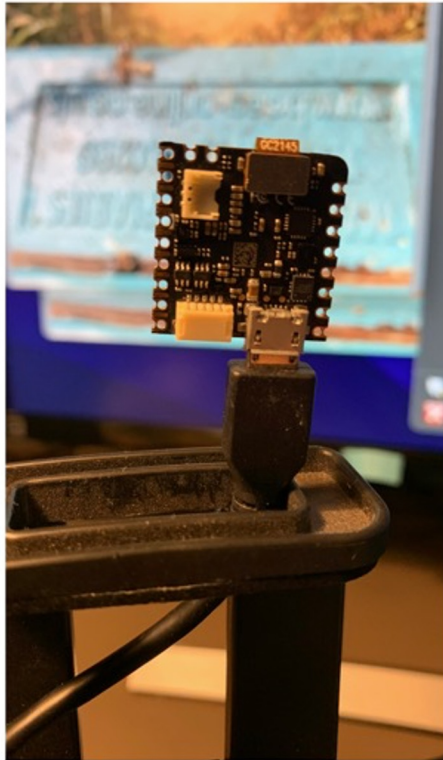
Detecting Objects using TinyML (FOMO)



```
***** 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](#)

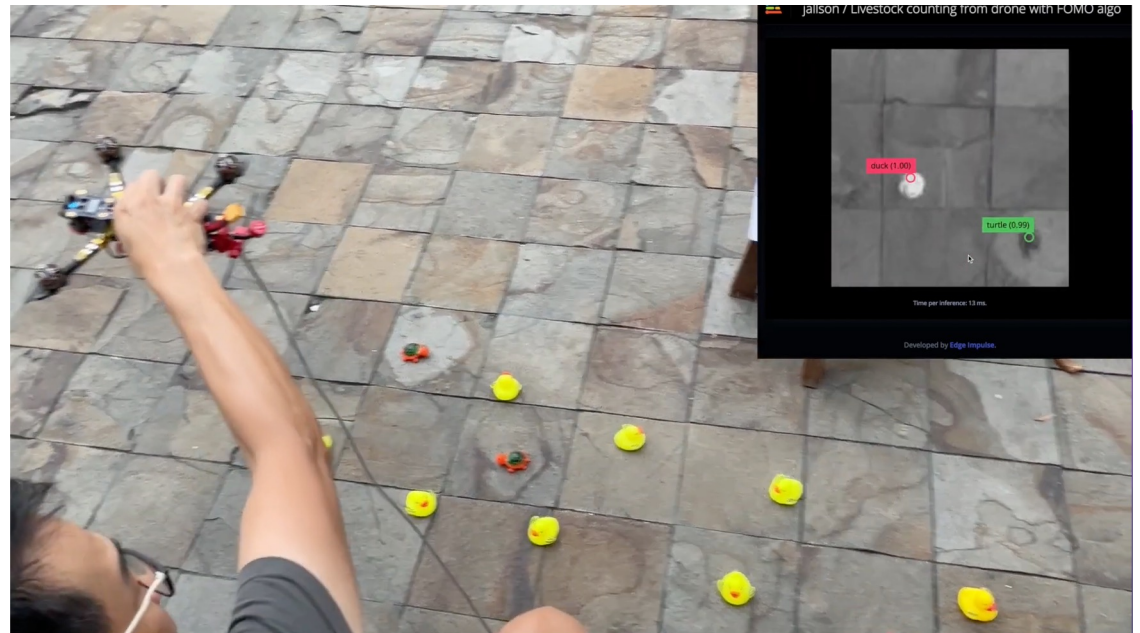
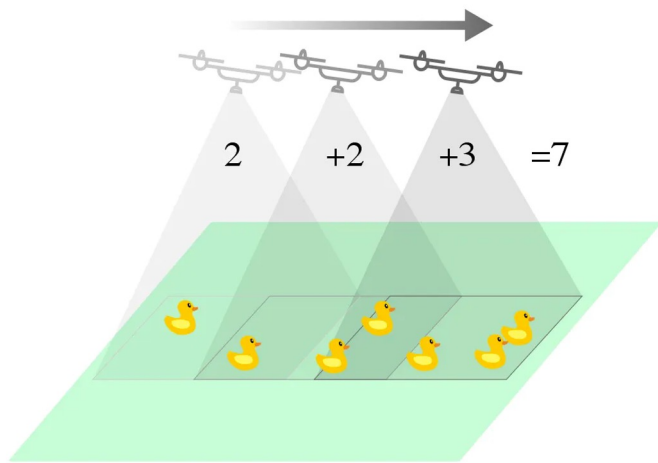
Detecting Objects using TinyML (FOMO)



MicroPython



Livestock / Wildlife Counting from Drone with FOMO



<https://www.hackster.io/jallsonsuryo/livestock-wildlife-counting-from-drone-with-fomo-algorithm-a2f734>

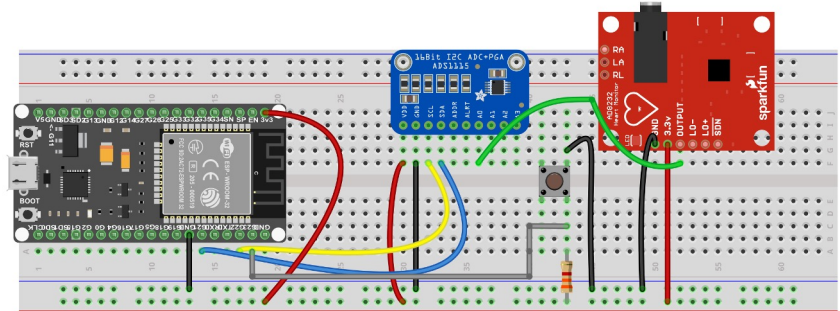
Other Sensors / MCUs / Models

Examples

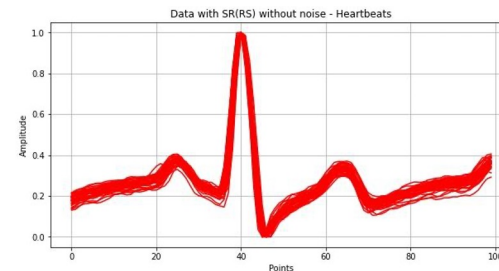
AD8232 - Single Lead Heart Rate Monitor



[Atrial Fibrillation Detection on ECG using TinyML](#)
[Silva et al. UNIFEI 2021](#)

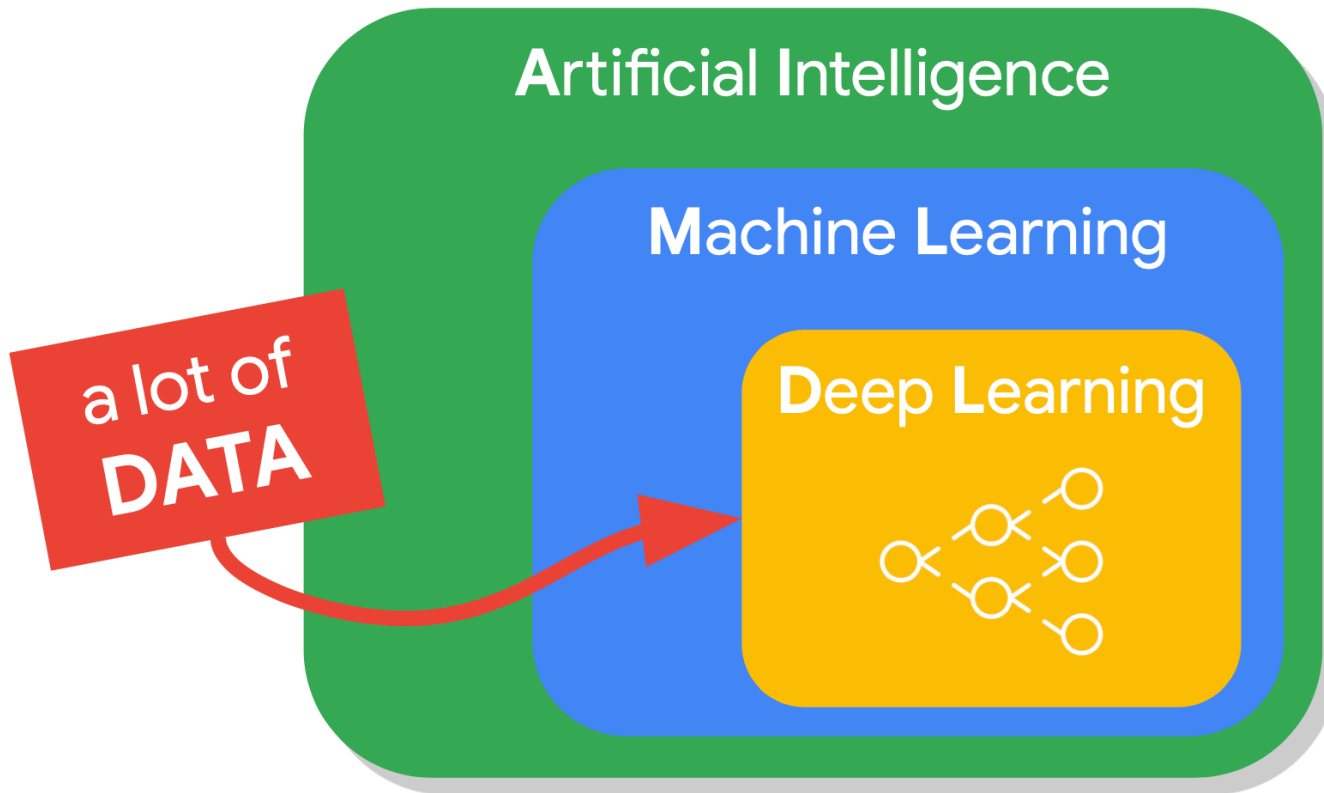


fritzing



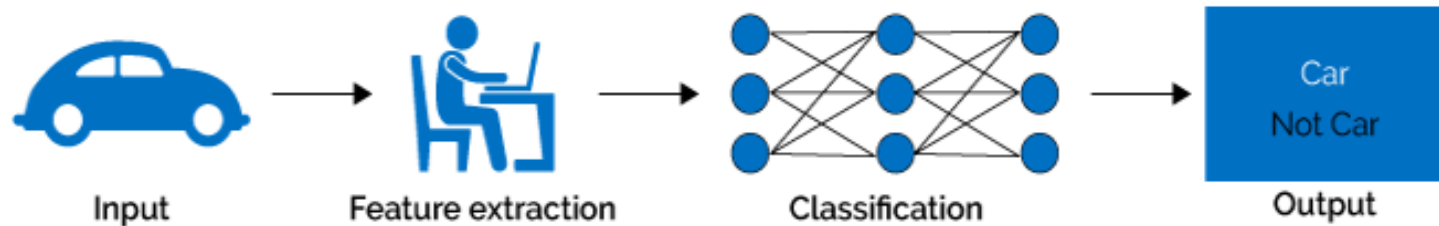
Guilherme Silva
Engenheiro - UNIFEI

Machine Learning: Some initial definitions

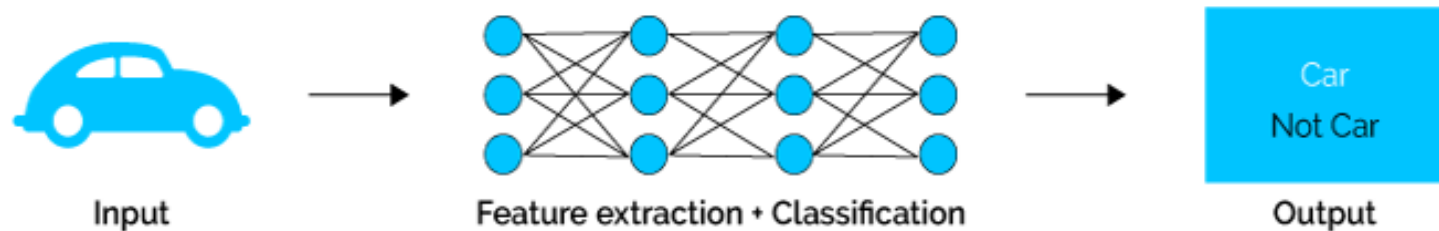


Machine Learning vs Deep Learning

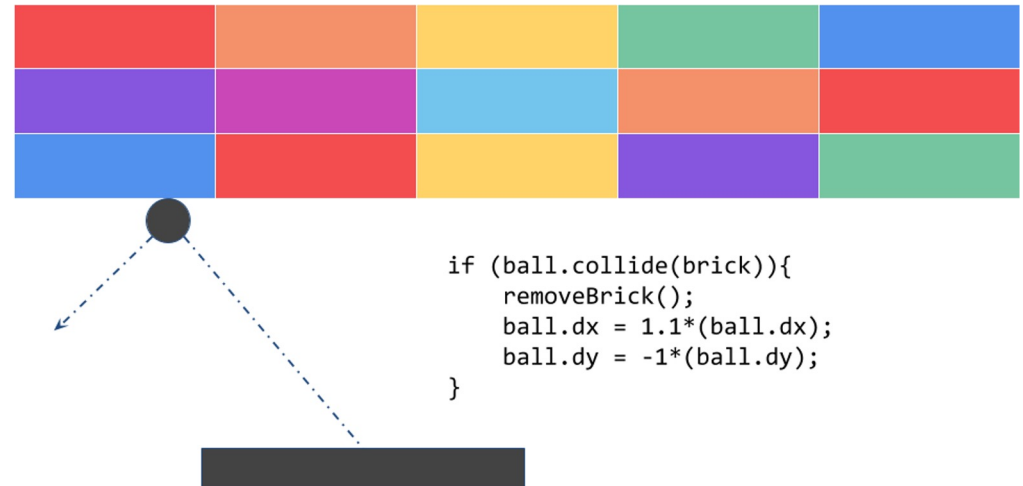
Machine Learning



Deep Learning



- **Defining rules** that determine behavior of a program
- Everything is **pre-calculated and pre-determined** by the programmer
- **Scenarios are limited** by program complexity



The Traditional Programming Paradigm



Consider Activity Detection



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



```
// ???
```

Way too
complex
to code!

The Machine Learning Paradigm



Activity Detection with Machine Learning



```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING



```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING



```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

Label = BIKING



```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING

Activity Detection with Machine Learning



```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING



```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING



```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

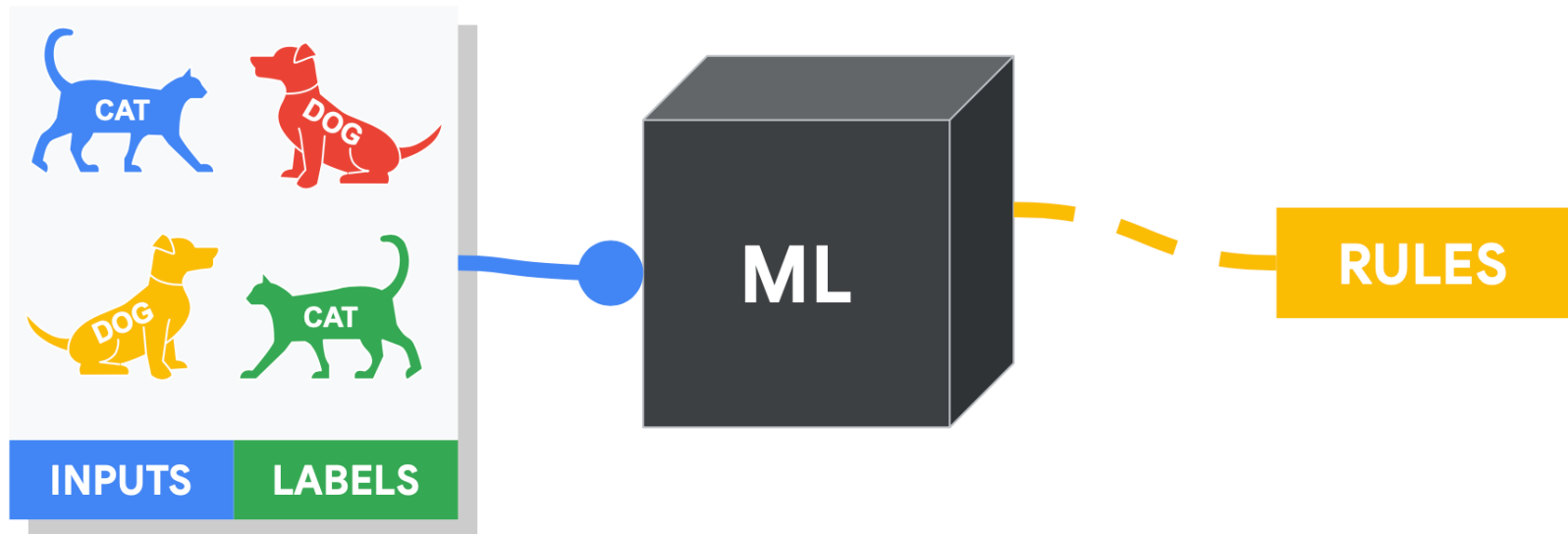
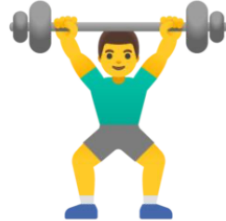
Label = BIKING

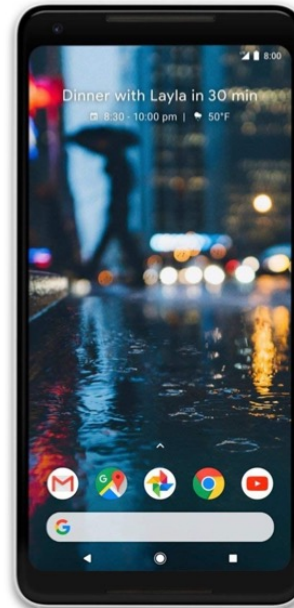
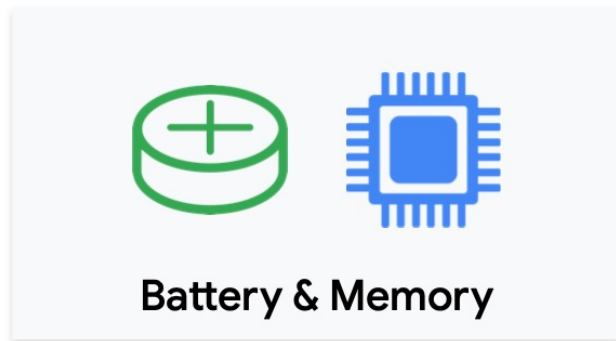


```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING

Training the machine





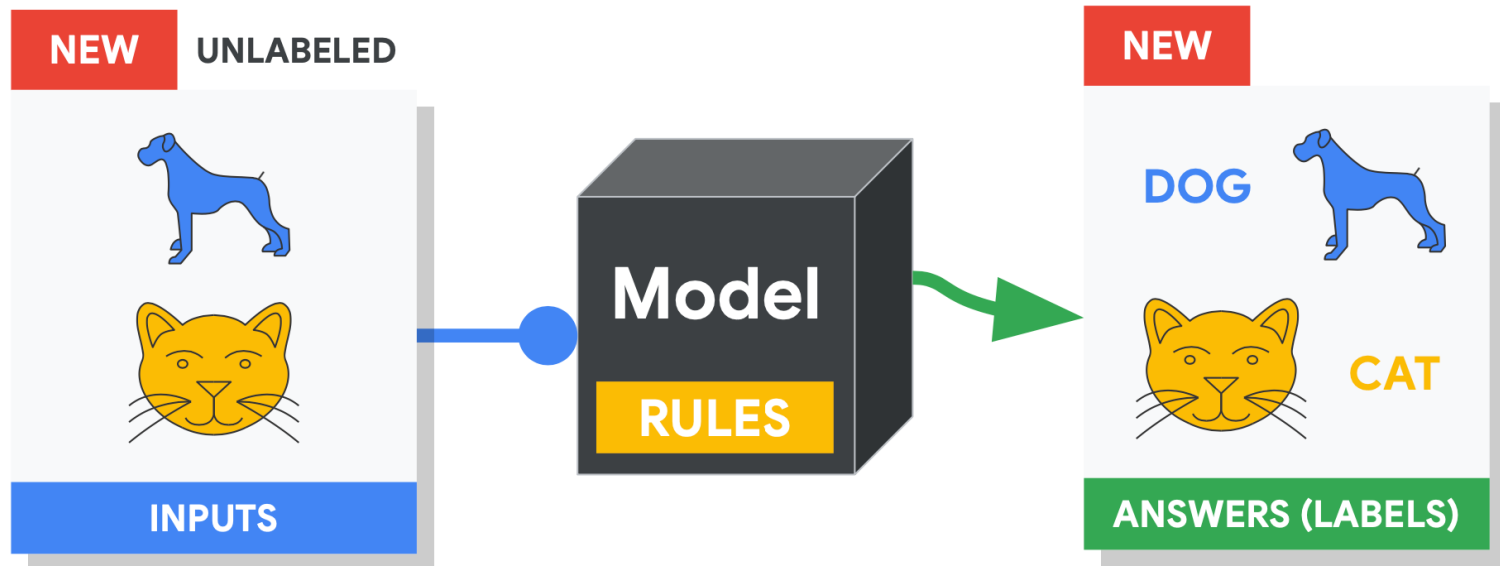
Less memory

Less compute power

Only focused on *inference*



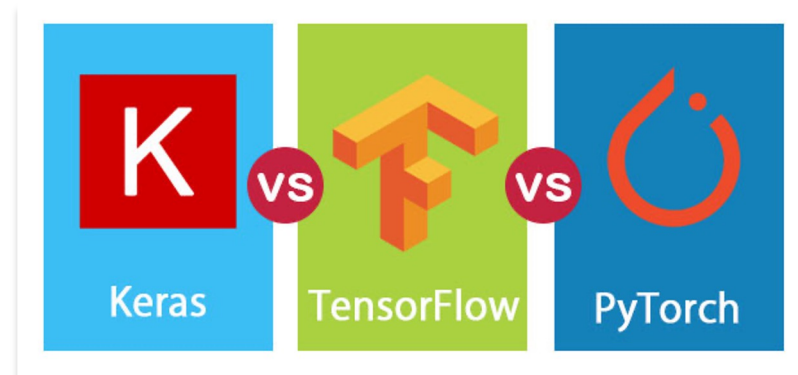
Making predictions:



- **Machine Learning** occurs in two stages – learning and inferencing. **At present, TinyML only handles inferencing**
- Training refers to the process of creating a machine learning algorithm. Training involves using a machine learning framework and a training dataset.
 - IoT data provides a source of training data that data scientists and engineers can use to train machine learning models.
- Inference refers to the process of using a trained machine-learning algorithm to make a prediction.
 - IoT data can be used as the input to a trained machine learning model, enabling predictions that can guide decision logic on the device, at the edge.

About the available software frameworks

- The three main frameworks which are available as an open-source library are opted by data scientist in deep learning are TensorFlow, Keras, and PyTorch.
 - **TensorFlow** is used to perform multiple tasks in data flow programming and machine learning applications.
 - **Keras** is a neural network library scripted in Python and executes on the top layer of TensorFlow.
 - **PyTorch** is a machine learning library that is mainly used in natural language processing.

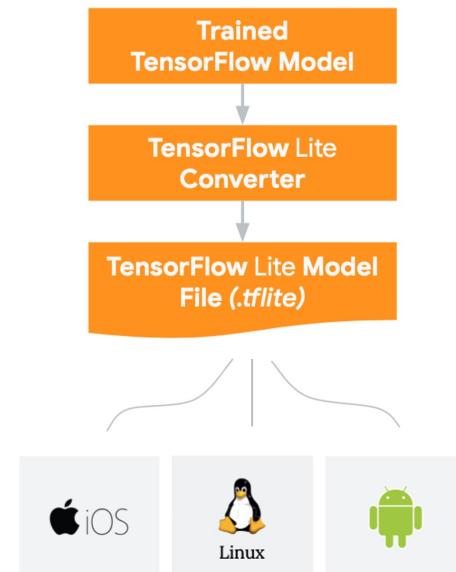
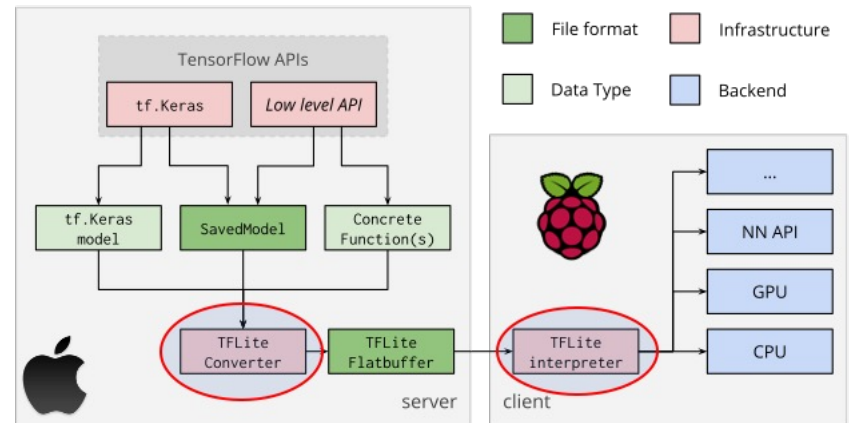




- <https://www.tensorflow.org/lite/microcontrollers>
- TensorFlow is Google's open-source machine learning framework which helps in developing machine learning models quickly.
- For TinyML, there is TensorFlow Lite Micro, which is a specialised version of TensorFlow for microcontrollers.
 - TFLite Micro is written in C++ 11 and requires a 32-bit platform and is mostly compatible with **Arm Cortex-M** Series processors.
 - [An ESP32 port is available as well](#)
- TFLite Micro allows you to easily compress regular TensorFlow models into just a few kilobytes and comes with numerous example models.

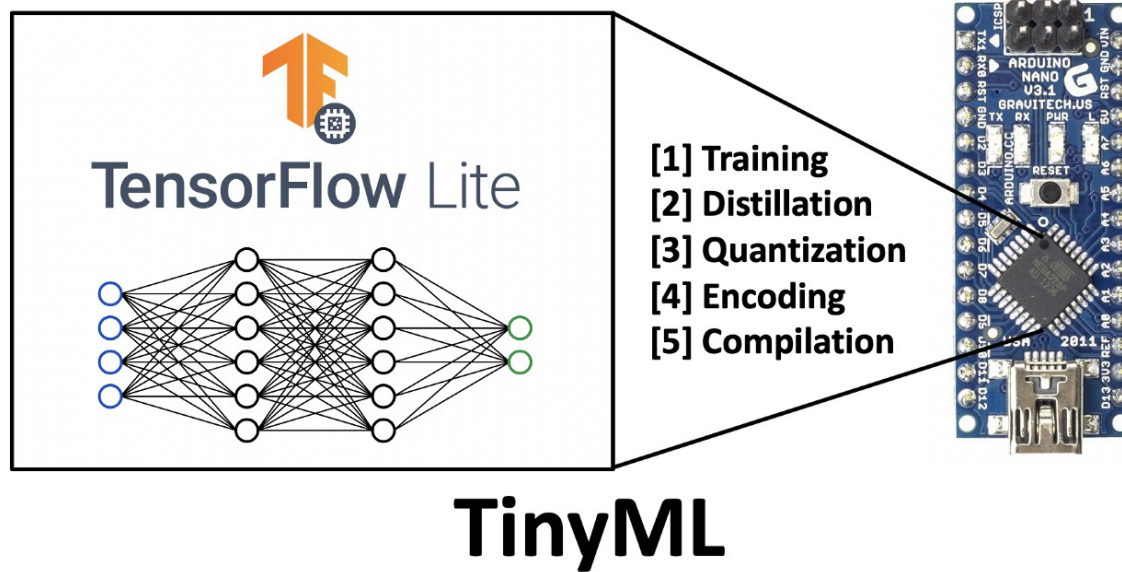
TensorFlow Lite (TFLite)

- Consists of two main components:
 - The **TFLite converter**, which converts TensorFlow models into an efficient form for use by the interpreter and can introduce optimizations to improve binary size and performance.
 - The **TFLite interpreter** runs with specially optimized models on many different hardware types, including mobile phones, embedded Linux devices, and microcontrollers.

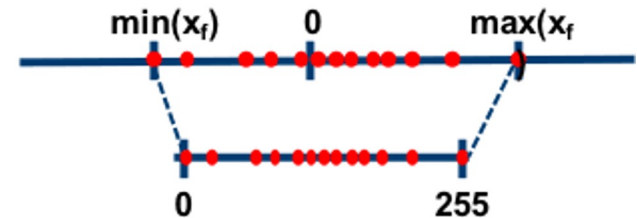
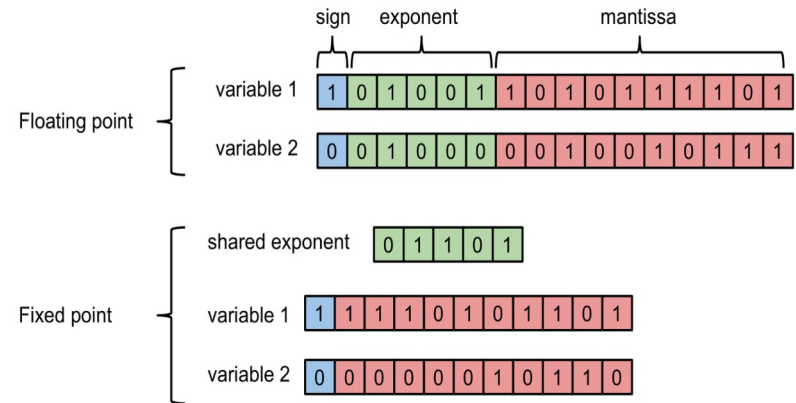


TensorFlow Lite optimizations

- Some optimizations:
 - Quantization
 - Pruning
- ..and
 - Separable Convolutions
 - Knowledge Distillation



- Neural Networks generally use 32 bit floating point weights
- These require more memory to save and custom logic to execute efficiently (which many micro-controllers do not have)
- By converting weights to INT8 (8 bits), we can save memory and compute faster
- However, it leads to a loss in accuracy since we lose precision



[source: heartbeat.fritz.ai](https://heartbeat.fritz.ai)

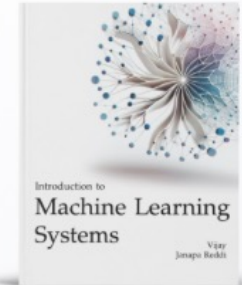
TinyML will become the largest driver of the microcontroller market in the next 10 years, according to Remi El-Ouazzane, the president of STMicroelectronics' microcontrollers and digital ICs group.

“I really believe this is the beginning of a tsunami wave,” he told EE Times in an exclusive interview. “We’re going to see a tsunami of products coming with ML functionality: It’s only going to increase, and it’s going to attract a lot of attention.”

STMicro has roughly a quarter of the microcontroller (MCU) market today, shipping between five and 10 million STM32 MCUs every day. According to El-Ouazzane, over the next five years, 500 million of those MCUs will be running some form of tinyML or AI workloads.

TinyML, which refers to running AI or machine learning inference on otherwise generic MCUs, “will become the largest endpoint market in the world,” he said.

- Main refs
 - <https://www.mlsysbook.ai>
- TinyML Academic Community
 - <http://tinymledu.org>
- HarvardX Profession Certificate in Tiny Machine Learning (TinyML)
 - <https://github.com/tinyMLx/courseware>
- Tensorflow lite:
 - <https://www.tensorflow.org/lite/>
 - <https://experiments.withgoogle.com/collection/tfliteformicrocontroller>
[s](#)
- [Seedstudio: Everything About TinyML – Basics, Courses, Projects & More!](#)



Early Access Preview
Publisher: MIT Press (2026)

