IESTI01 - TinyML

Part 2 - TinyML Applications

Prof. Marcelo Rovai

June 9th, 2021

Tiny Machine Learning (TinyML)

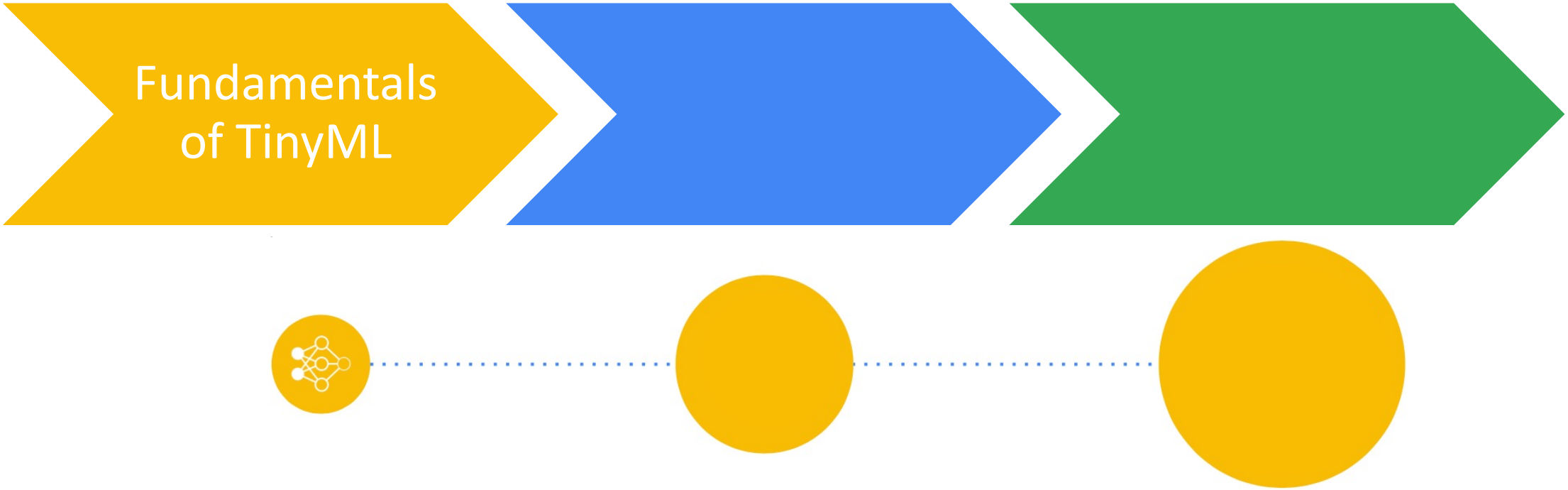
What we learned so far

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

- Fast-growing field of **machine learning**
- Algorithms, **hardware, and software**
- **On-device** sensor data analytics
- Extreme **low power** consumption
- **Always-on ML** use-cases
- **Battery**-operated devices

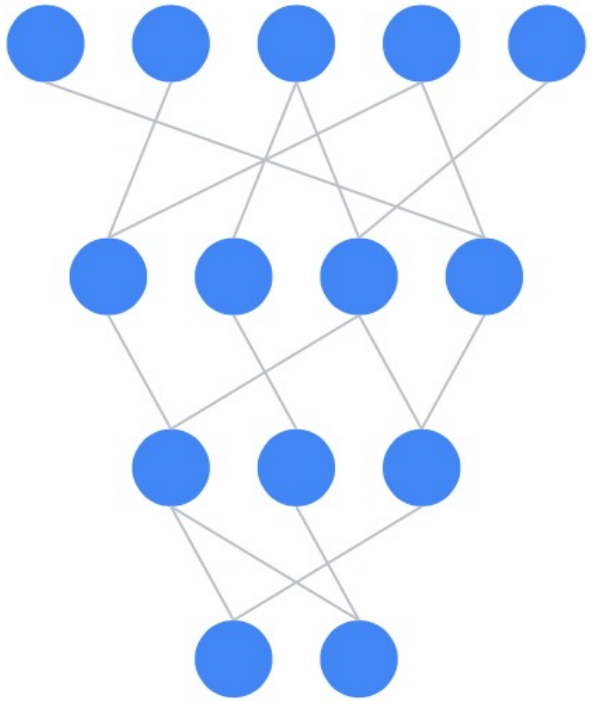
What we already learned?

April/May



In April and May (Part 1) we introduced ML with TensorFlow. Was all about talking about what is the language of machine learning.

Total Recall from **Part 1**



“Language” for **Part 1**

Training Data

Neural Network

Training

Validation Data

Inference

Gradient Descent

Test Data

Features

Loss Function

Classification

Filters

Overfitting

Kernels

Regression

Data augmentation

CNNs

DNNs

Responsible AI

Preprocessing

“Language” for **Part 1**

Training Data

Neural Network

Training

Validation Data

Inference

Gradient Descent

Test Data

Features

Loss Function

Classification

Filters

Overfitting

Kernels

Regression

Data augmentation

CNNs

DNNs

Responsible AI

Preprocessing

“Language” for **Part 1**

Training Data

Neural Network

Training

Validation Data

Inference

Gradient Descent

Test Data

Features

Classification

Loss Function

Filters

Overfitting

Kernels

Regression

Data augmentation

CNNs

DNNs

Responsible AI

Preprocessing

“Language” for **Part 1**

Training Data

Neural Network

Training

Validation Data

Inference

Gradient Descent

Test Data

Features

Classification

Loss Function

Filters

Overfitting

Kernels

Regression

Data augmentation

CNNs

DNNs

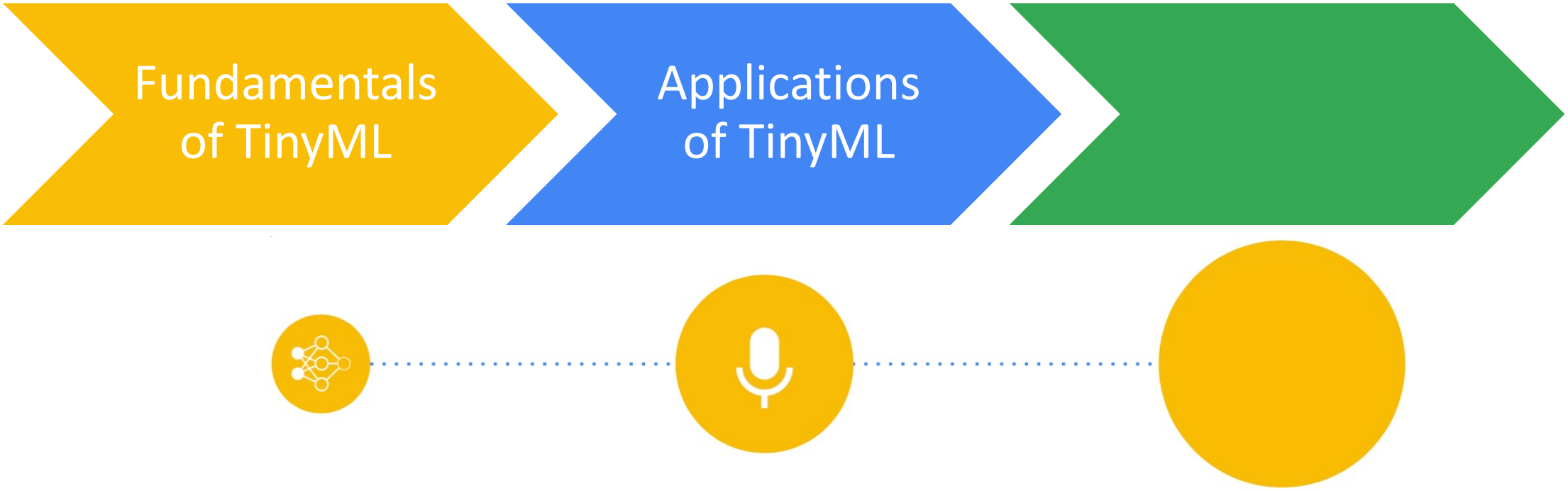
Responsible AI

Preprocessing

What we will learn?

April/May

June



In June (Part 2), We will learn the importance of dataset engineering, get a sneak peek into different TinyML applications, as keyword spotting (“Alexa”), gesture recognition, and image detection.

What we will learn?

April/May

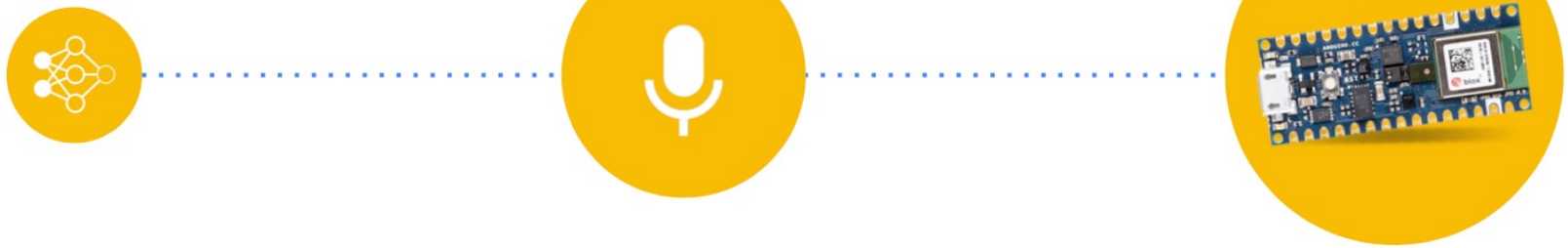
Fundamentals of TinyML

June

Applications of TinyML

July/August

Deploying TinyML



In July and August (Part 3), We will learn how to deploy models on a real microcontroller. Along the way we will explore the challenges unique to and amplified by TinyML (e.g., preprocessing, post-processing, dealing with resource constraints).

TensorFlow

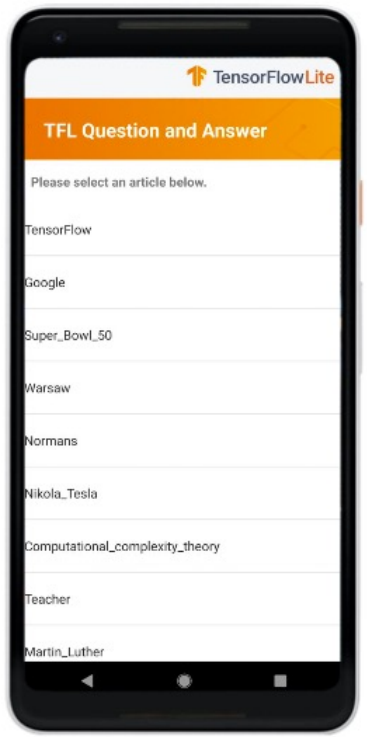
TensorFlow Lite

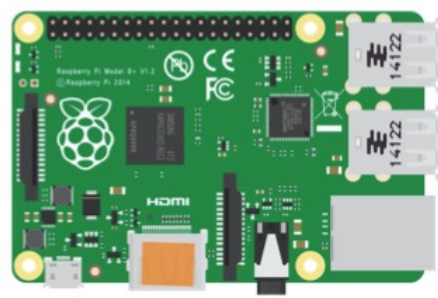
Train a model → Convert model → Optimize model → Deploy model at Edge → Make inferences at Edge

TensorFlow

TensorFlow Lite

Train a model → Convert model → Optimize model → Deploy model at Edge → Make inferences at Edge





Tiny Machine Learning (TinyML)

Applications

TinyML Application Areas

Home

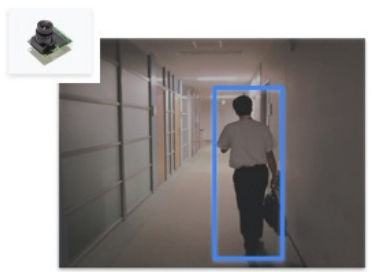
Office

Industry

TinyML Application Areas

Home

Office



Industry

Questions

- How do we **capture** the data to feed into the neural network?
- How do you **design** the neural network to take in the speech signal?
- What **dataset** does the neural network need to be trained?
- How do we **pre-process** the data for neural network inference?
- How do you **post-process** the neural network output?
- How do you make sure there is no **bias** in the dataset?
- How do you **deploy** this on the microcontroller?

Endpoints Have **Sensors**, Tons of Sensors

Motion Sensors
Gyroscope, Radar, Accelerometer

Acoustic Sensors
Ultrasonic, Microphones, Geophones, Vibrometers

Environmental Sensors
Temperature, Humidity, Pressure, IR, etc.

Touchscreen Sensors
Capacitive, IR

Image Sensors
Thermal, Image

Biometric Sensors
Fingerprint, Heart rate, etc.

Force Sensors
Pressure, Strain

Rotation Sensors
Encoders

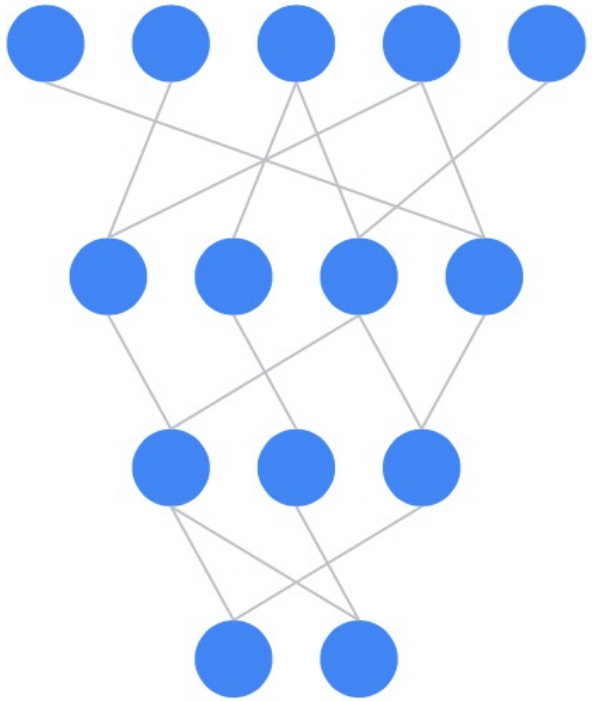
Sensors

Acoustic Sensors
Ultrasonic, **Microphones**, Geophones, Vibrometers

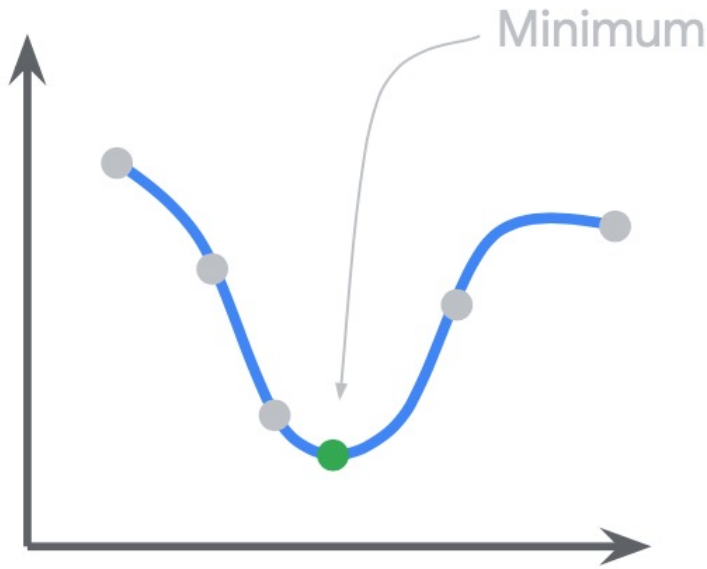
Image Sensors
Thermal, **Image**

Motion Sensors
Gyroscope, Radar, **Accelerometer**

Models



Metrics



End-to-end **TinyML** application design

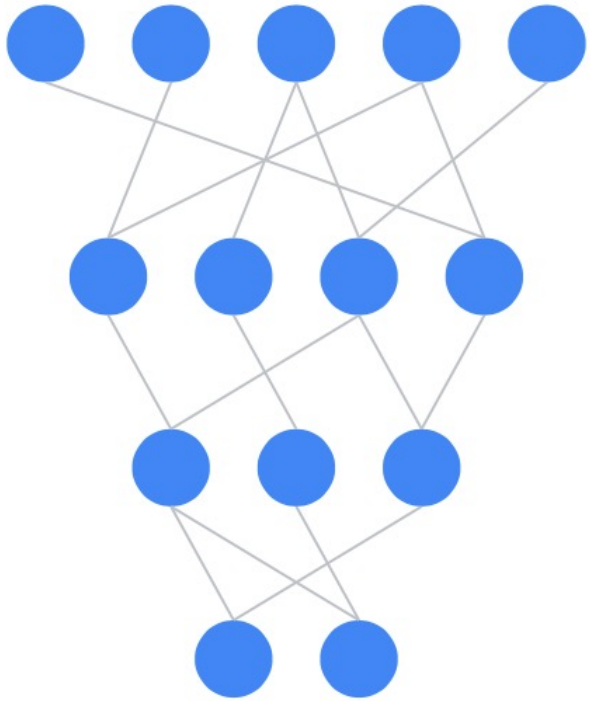
Datasets Preprocessing

Sound

Vision

Vibration

Quantization Pruning

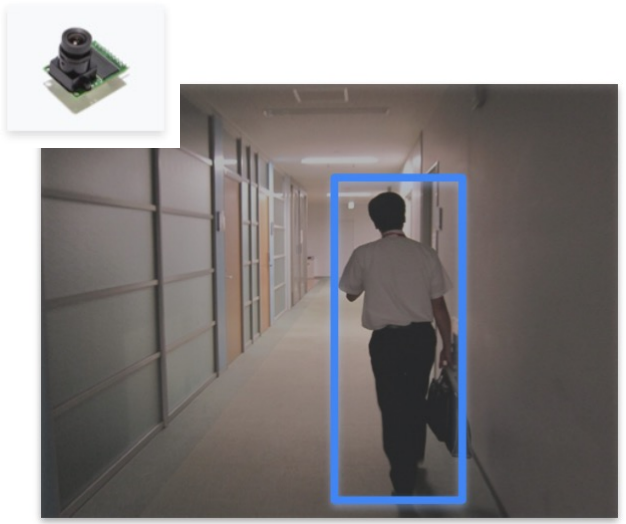


Resource constraints

End-to-end **TinyML** application design

Sound

Vision



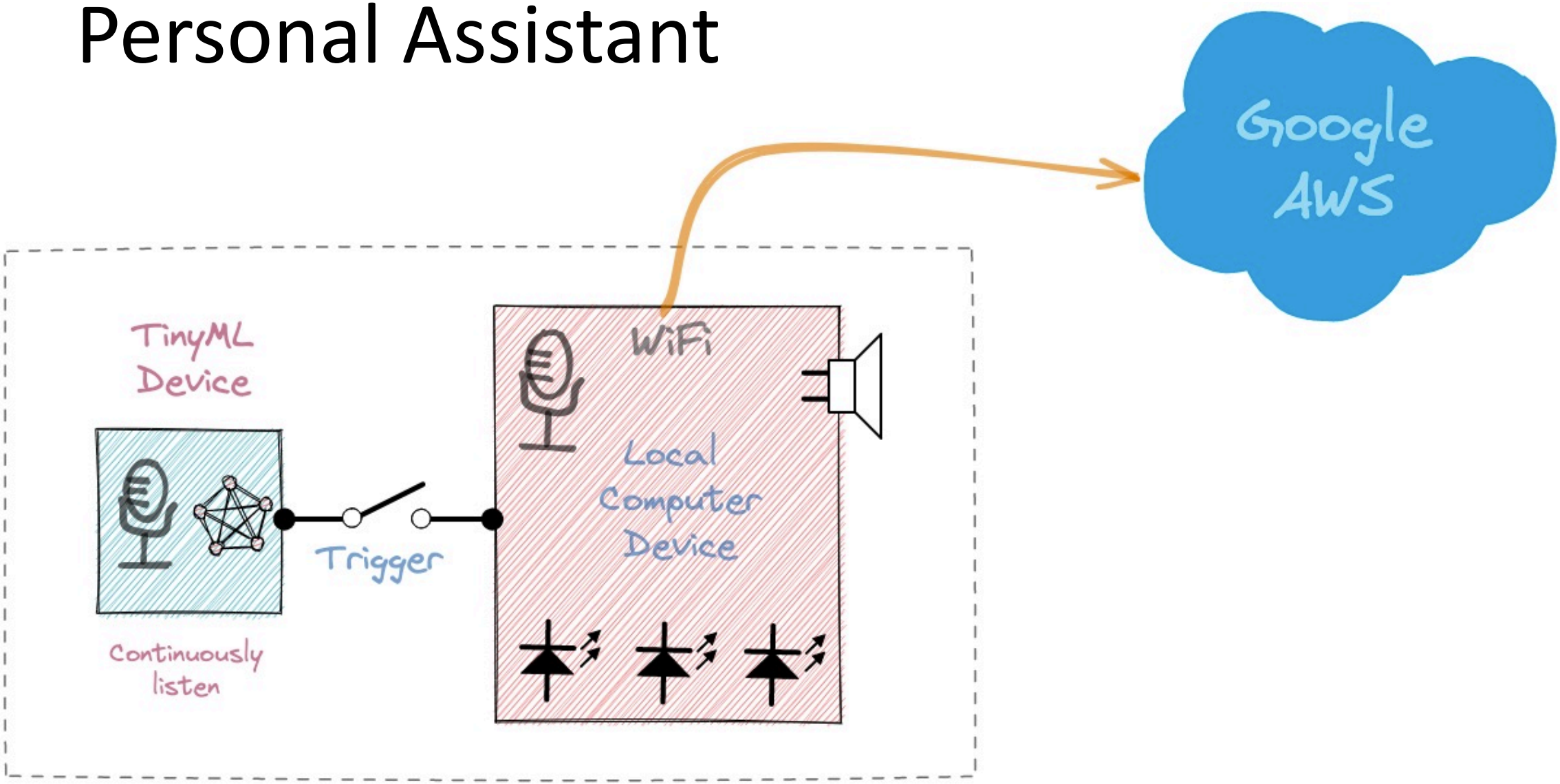
Vibration

TinyML Application

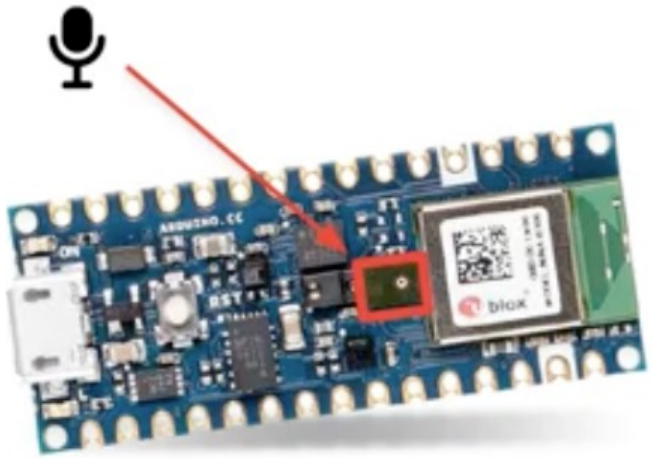
Example

Personal Assistant

Personal Assistant



“Cascade” Detection: multi-stage model



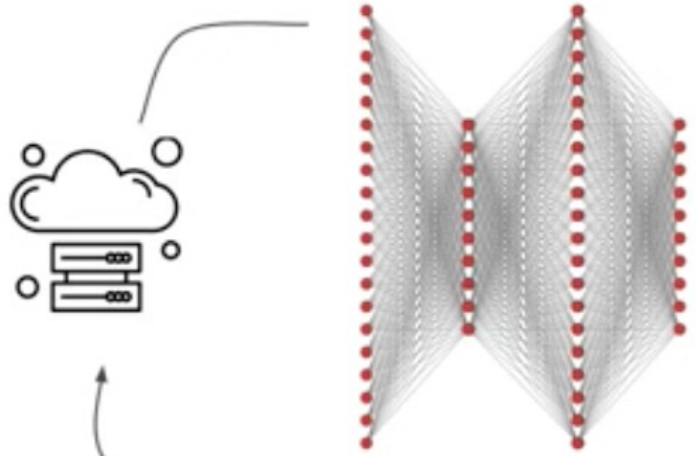
1 Continuously listen on the microcontroller

2 Process the data with **TinyML** at the edge

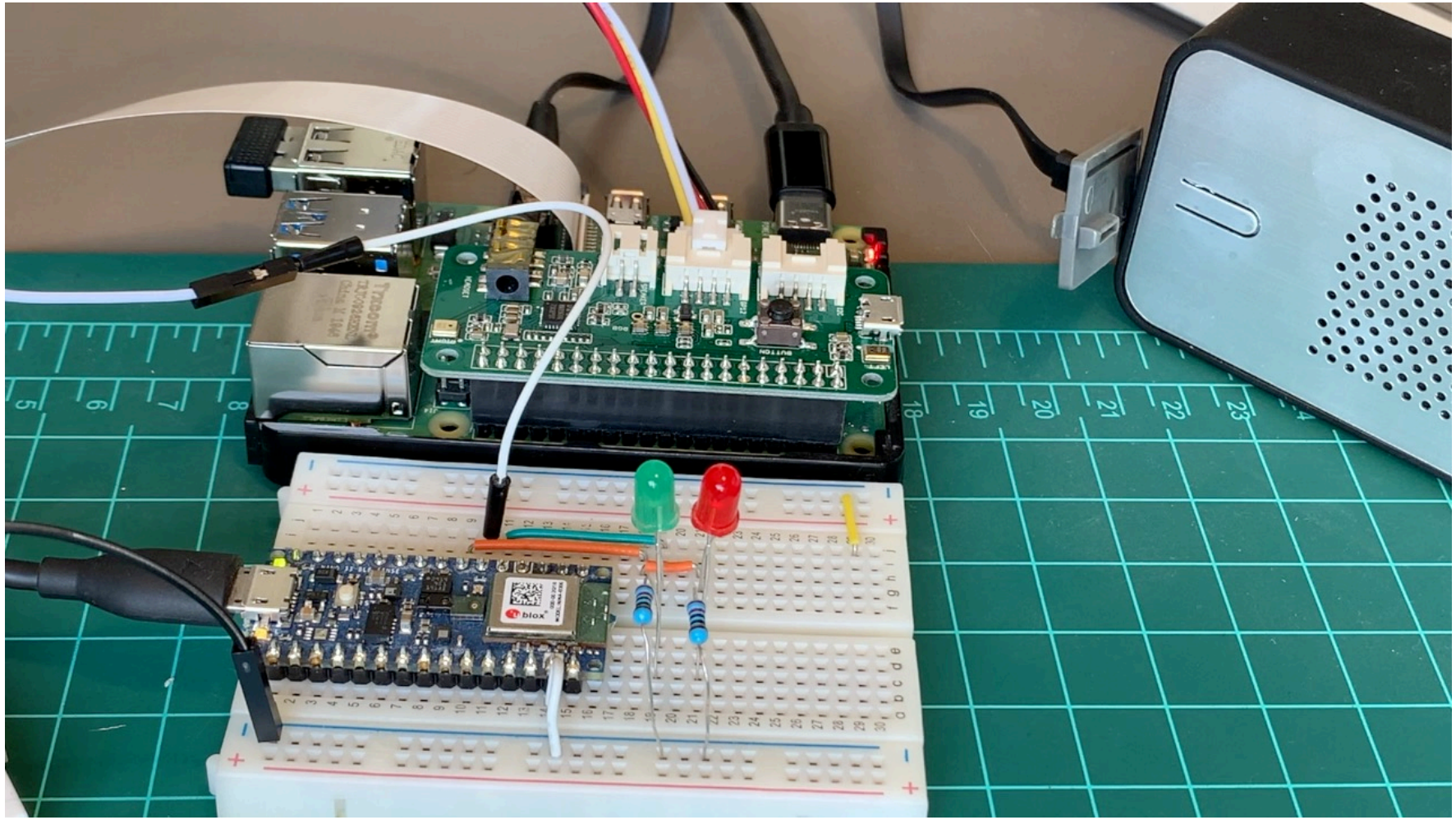
3 Process on a secondary larger model on a larger local device

4 Send the data to the cloud when triggered

5 Process the full speech data with a large model in the cloud

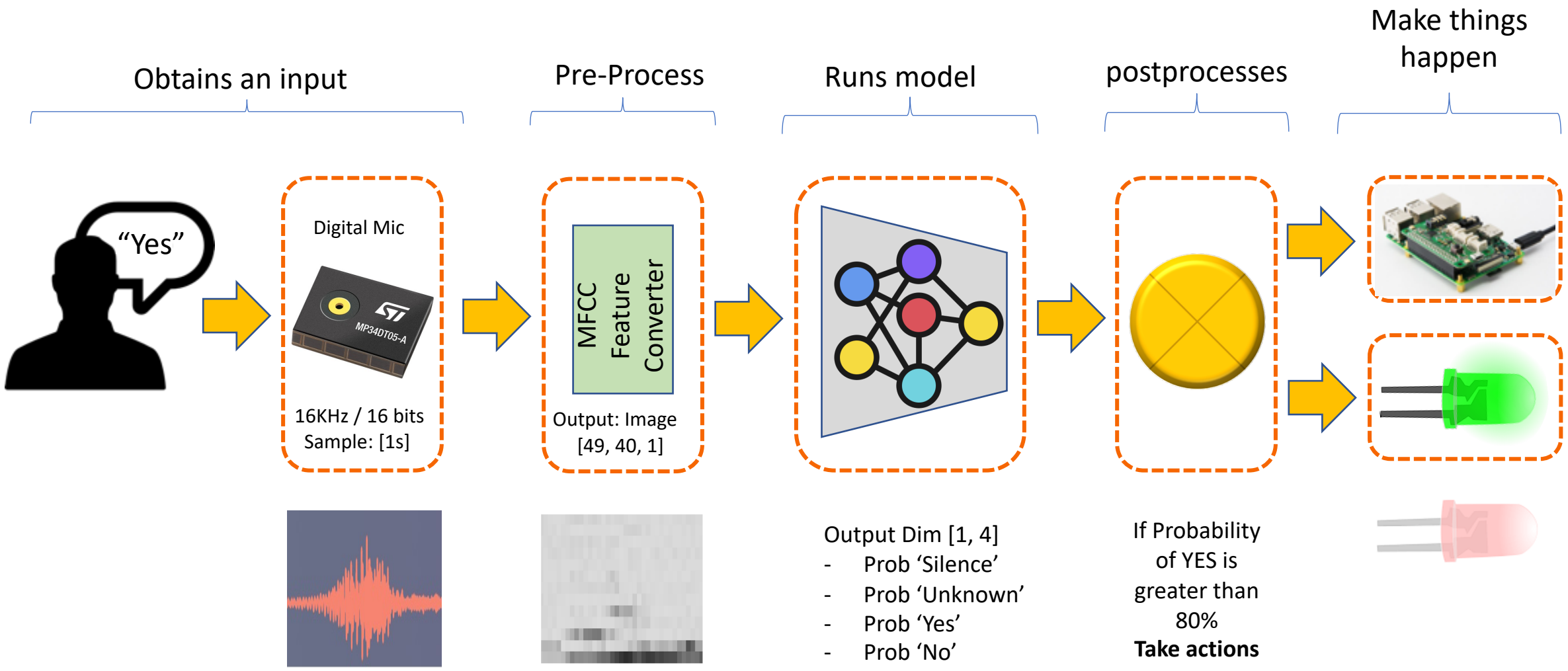


KeyWord Spotting (KWS)

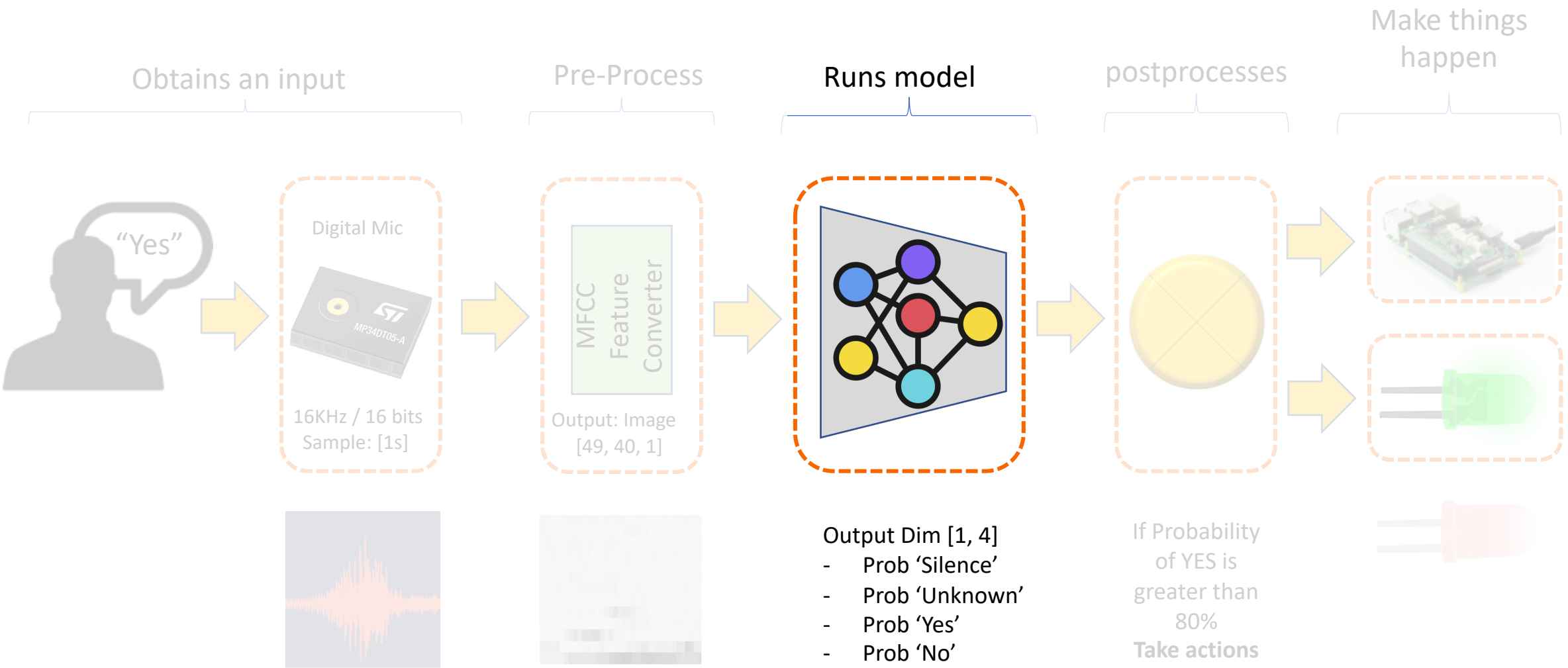


https://mjrobot.org/2021/01/27/building-an-intelligent-voice-assistant-from-scratch/

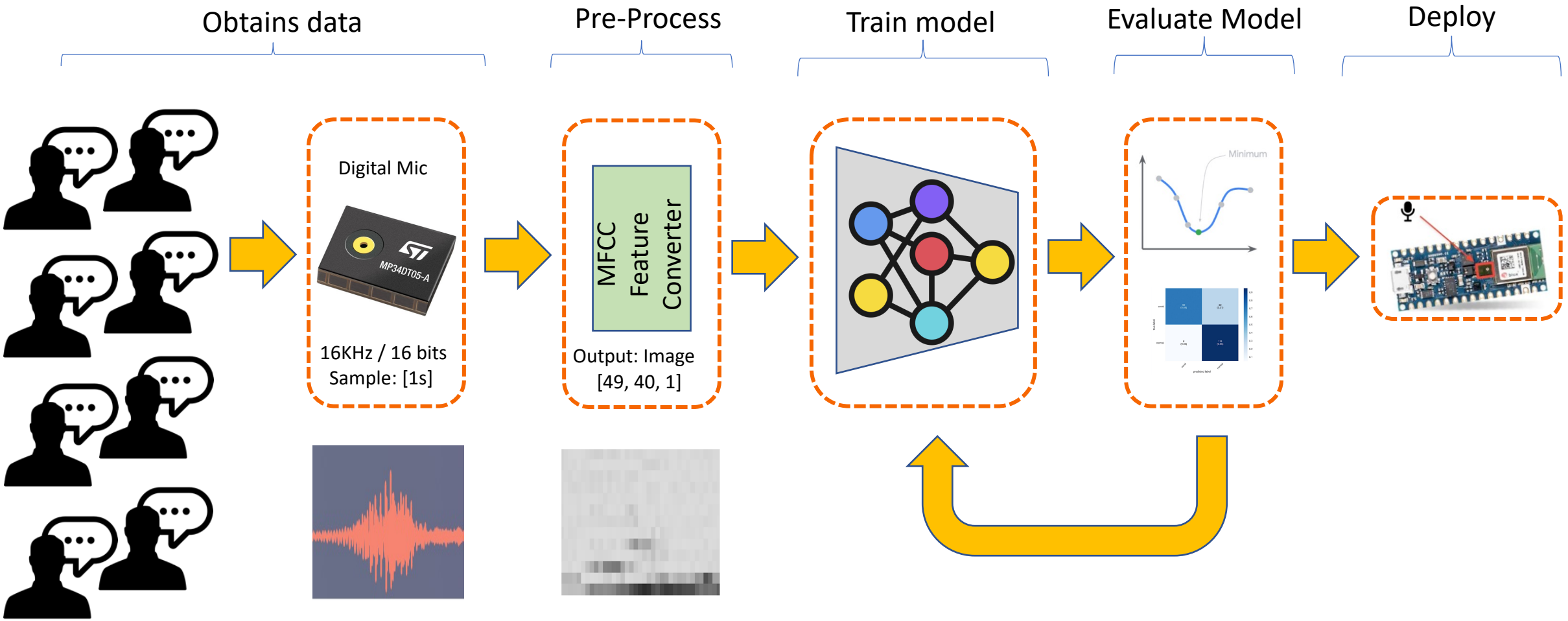
KeyWord Spotting (KWS) - **Inference**



KeyWord Spotting (KWS) - **Model**



KeyWord Spotting (KWS) – Create Model (Training)



Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning (Coursera)
- Text Book: "TinyML" by Pete Warden, Daniel Situnayake

I want to thank Laurence Moroney from Google, Harvard professor Vijay Janapa Reddi, Ph.D. student Brian Plancher and their staff for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the TinyML4D, an initiative to make TinyML education available to everyone globally.

Thanks

And stay safe!