

ESP32 - KeyWord Spotting (KWS)



Prof. Marcelo José Rovai
UNIFEI - Federal University of Itajubá, Brazil
TinyML4D Academic Network Co-Chair

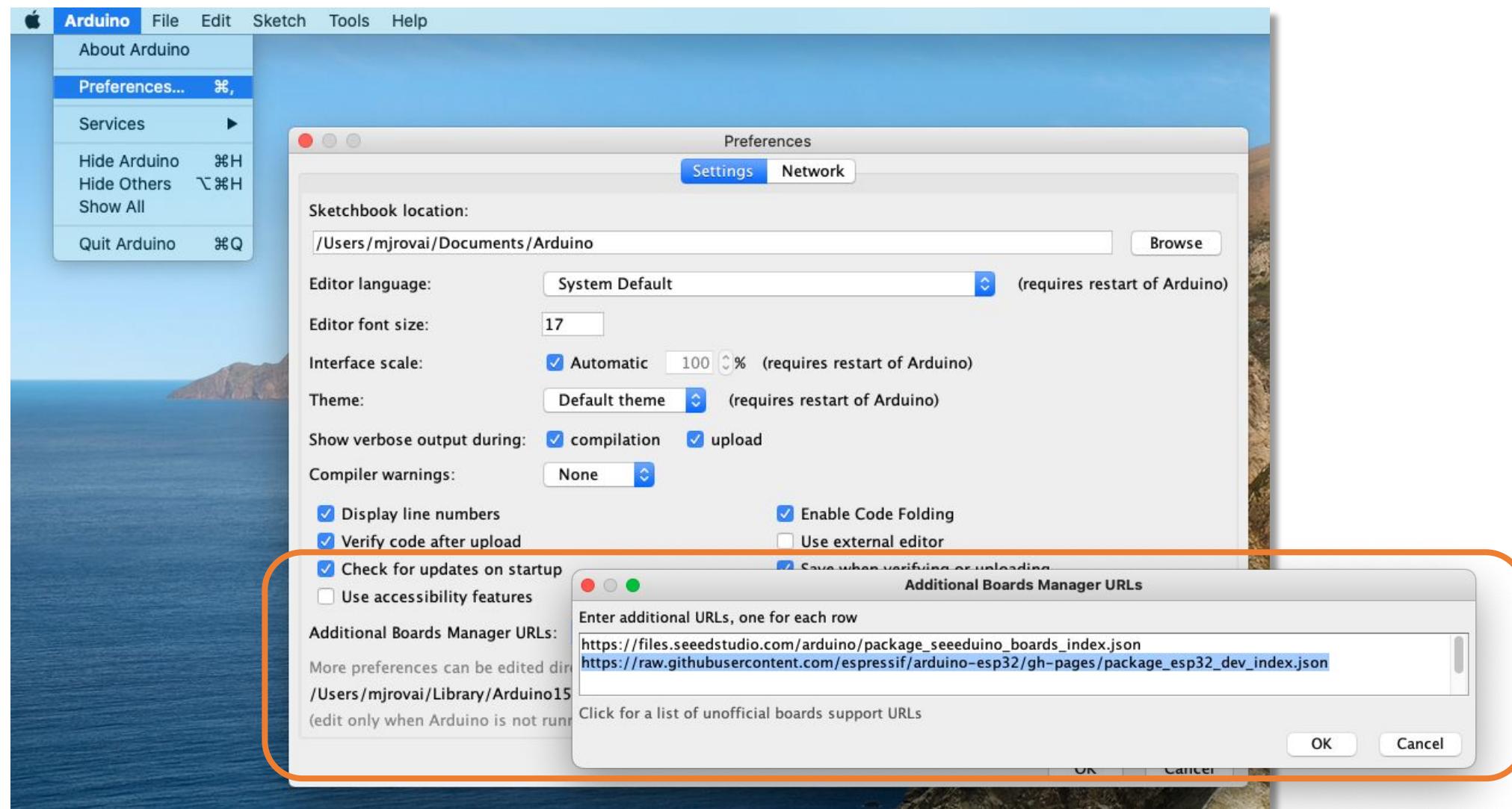


UNIFEI

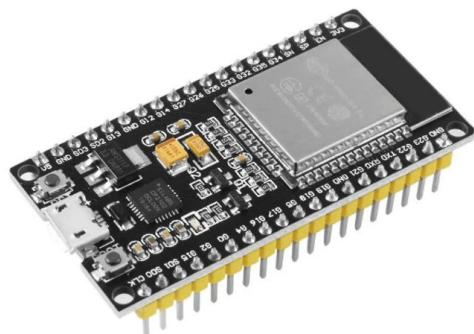
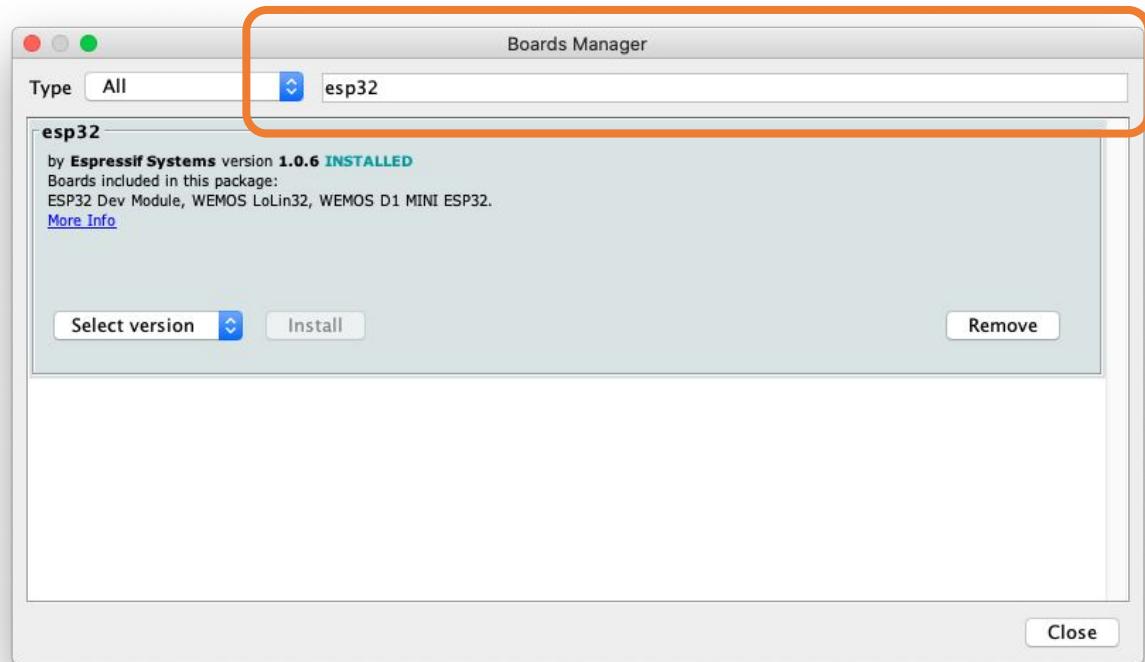
Code available at:

<https://github.com/Mjrovai/ESP32-TinyML/tree/main/ESP32-KWS>

On Arduino IDE, navigate to **File > Preferences**, and fill in "Additional Boards Manager URLs" with the URL below:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

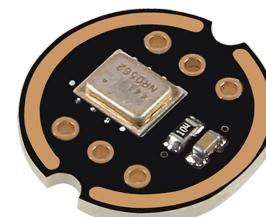


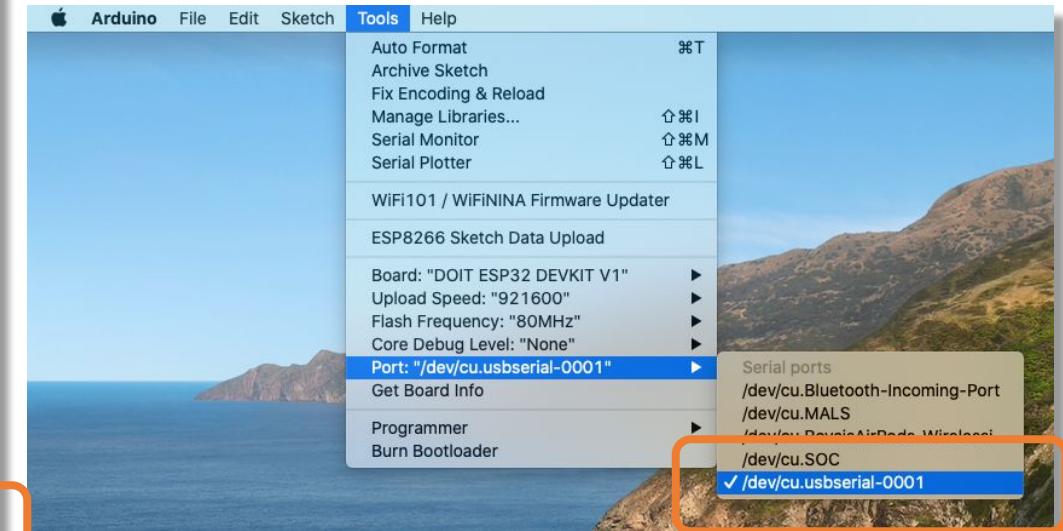
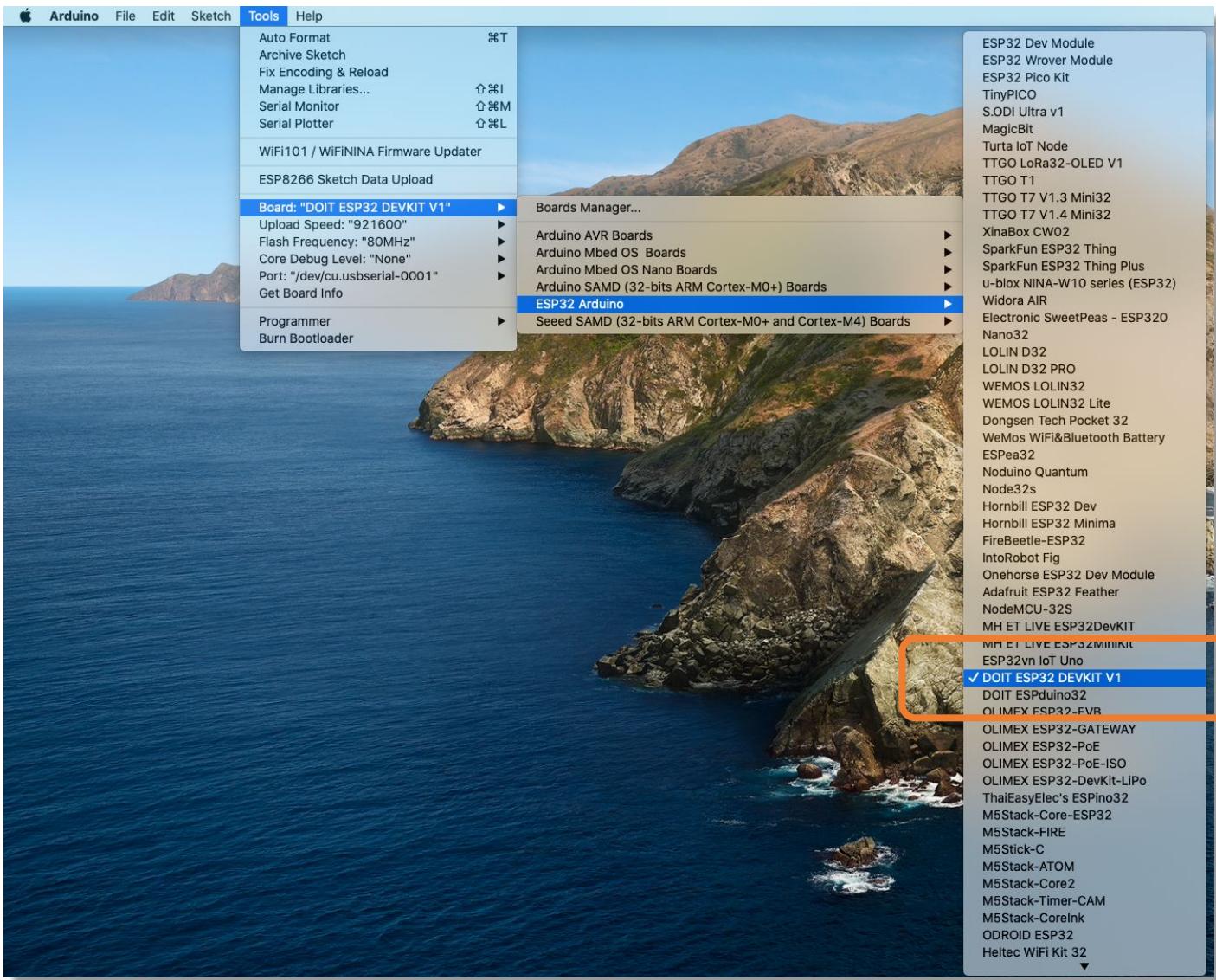
Install HW Libraries



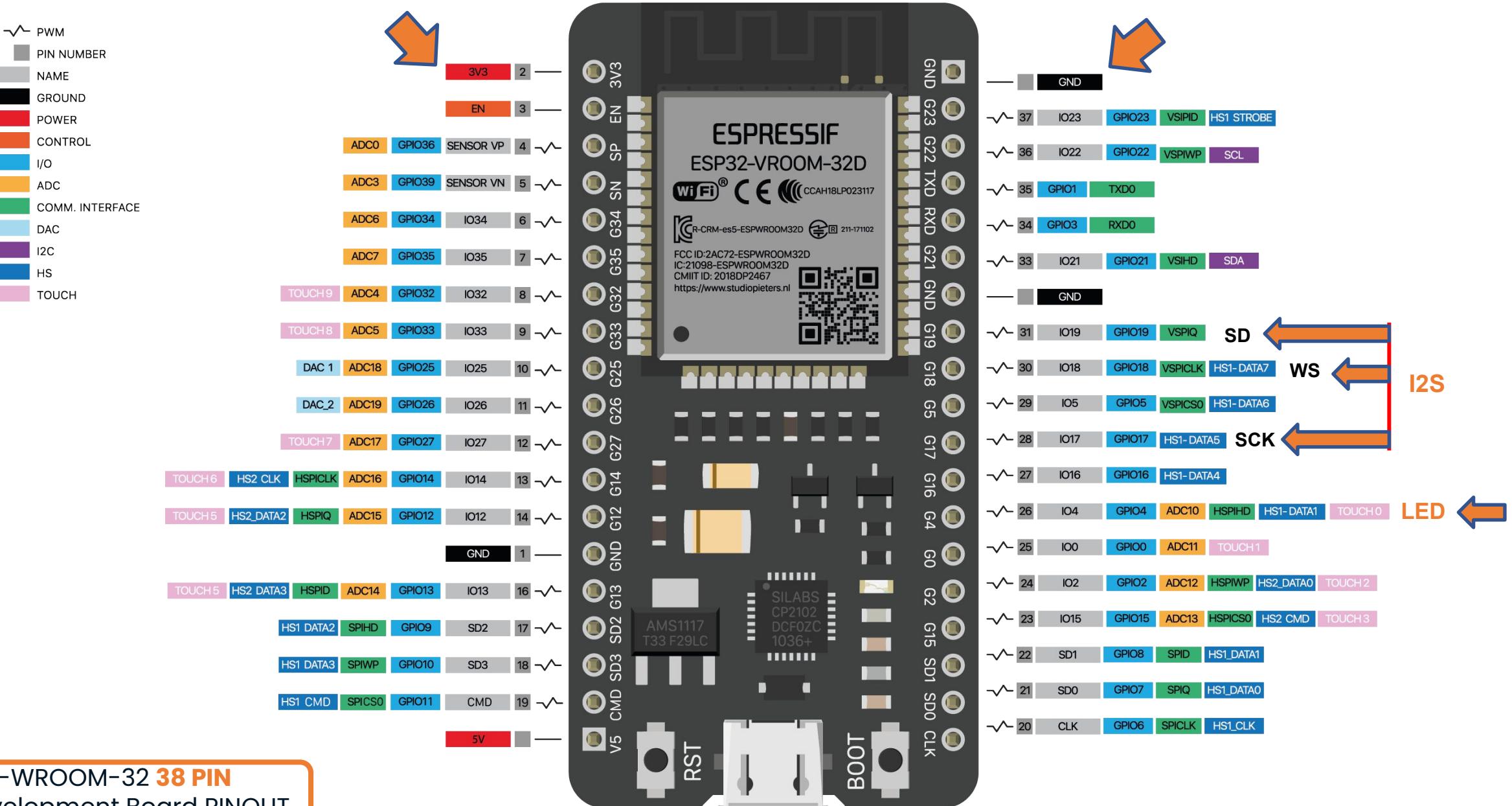
INMP441 Omnidirectional Microphone Module MEMS
High Precision Low Power I2S Interface

Library ("driver/i2s.h"), is part of ESP32 basic code





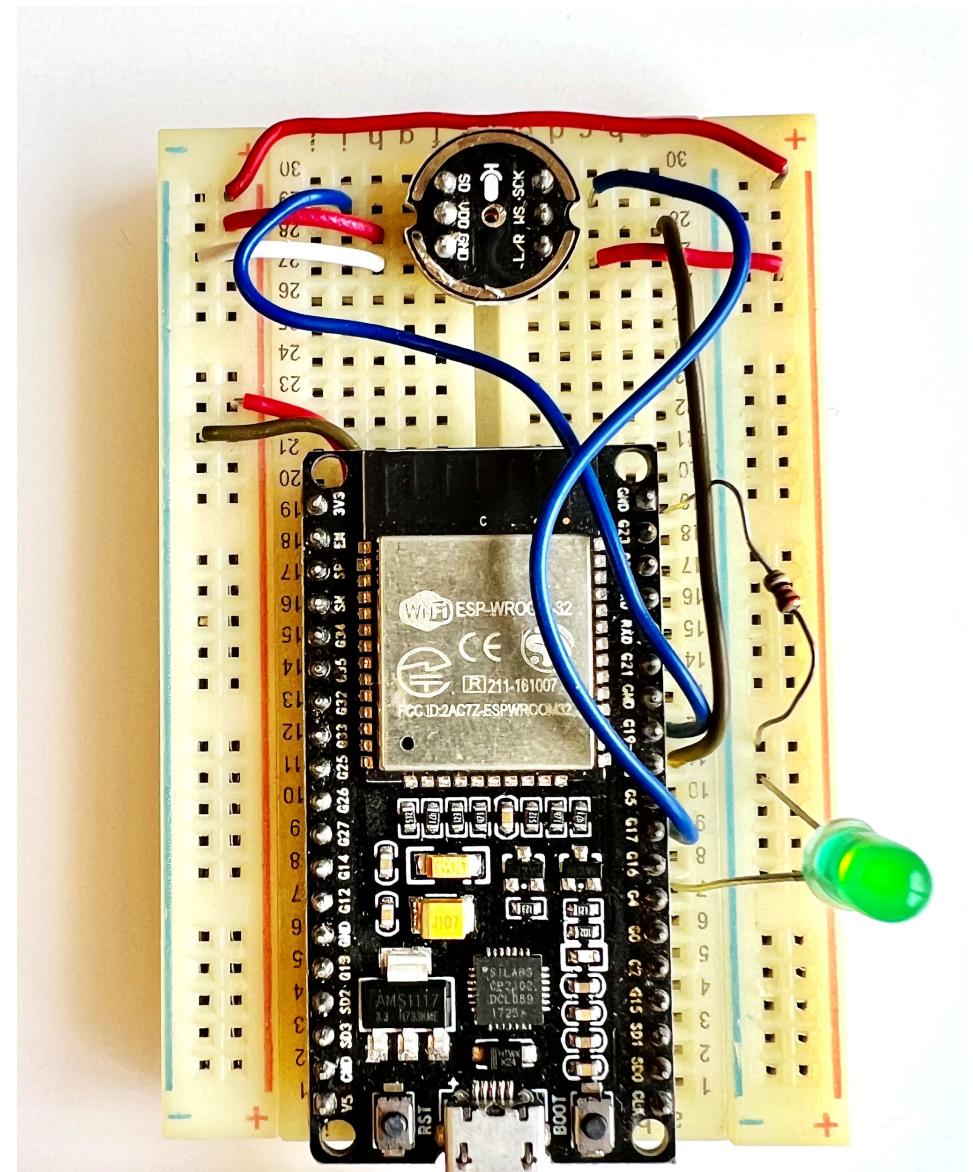
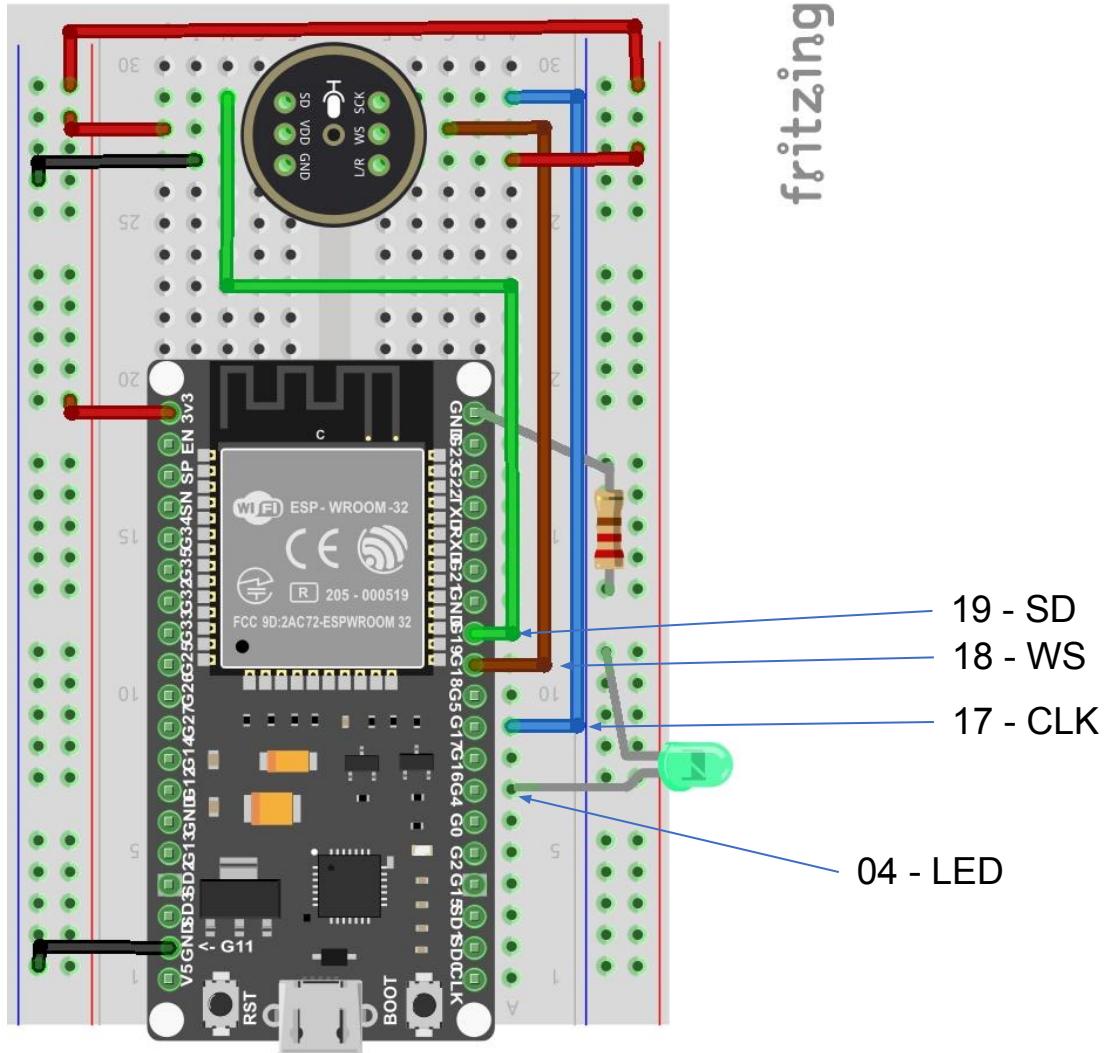
Select Board and Port



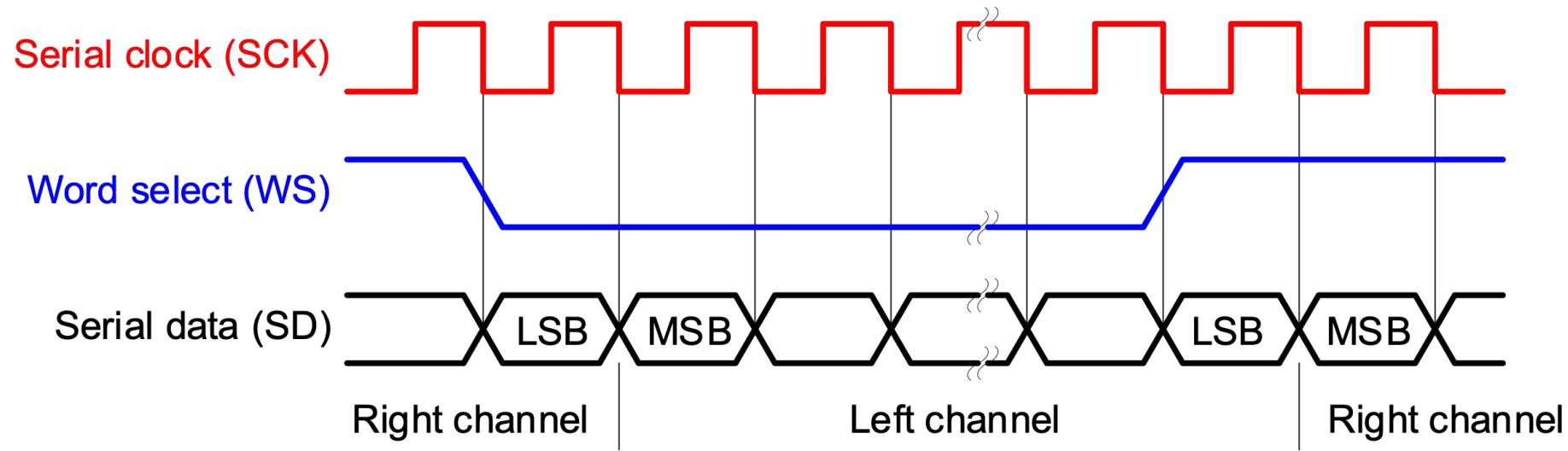
ESP-WROOM-32 **38 PIN** Development Board PINOUT

<https://www.studiopieters.nl/esp32-pinout/>

Connect HW



I2S Signals



Test ESP32 + Mic

ES32_sound_i2s_32bits_test | Arduino 1.8.19

```
3 * https://github.com/atomic14/esp32-i2s-mic-test/tree/main
4 */
5
6#include <driver/i2s.h>
7
8// you shouldn't need to change these settings
9#define SAMPLE_BUFFER_SIZE 2048
10#define SAMPLE_RATE 16000
11
12// most microphones will probably default to left channel but you may need to tie the L/R pin low
13#define I2S_MIC_CHANNEL I2S_CHANNEL_FMT_ONLY_LEFT
14
15// either wire your microphone to the same pins or change these to match your wiring
16#define I2S_MIC_SERIAL_CLOCK GPIO_NUM_17
17#define I2S_MIC_LEFT_RIGHT_CLOCK GPIO_NUM_18
18#define I2S_MIC_SERIAL_DATA GPIO_NUM_19
19
20// don't mess around with this
21i2s_config_t i2s_config = {
22    .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
23    .sample_rate = SAMPLE_RATE,
24    .bits_per_sample = (i2s_bits_per_sample_t)32,
25    .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
26    .communication_format = I2S_COMM_FORMAT_I2S,
27    .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
28    .dma_buf_count = 4,
29    .dma_buf_len = 1024,
30    .use_apll = false,
31    .tx_desc_auto_clear = false,
32    .fixed_mclk = 0};
33
34// and don't mess around with this
35i2s_pin_config_t i2s_mic_pins = {
36    .bck_io_num = I2S_MIC_SERIAL_CLOCK,
37    .ws_io_num = I2S_MIC_LEFT_RIGHT_CLOCK,
38    .data_out_num = I2S_PIN_NO_CHANGE,
39    .data_in_num = I2S_MIC_SERIAL_DATA};

Done uploading.

Leaving...
Hard resetting via RTS pin...
```

24 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on /dev/cu.usbserial-0001

ES32_sound_i2s_32bits_test | Arduino 1.8.19

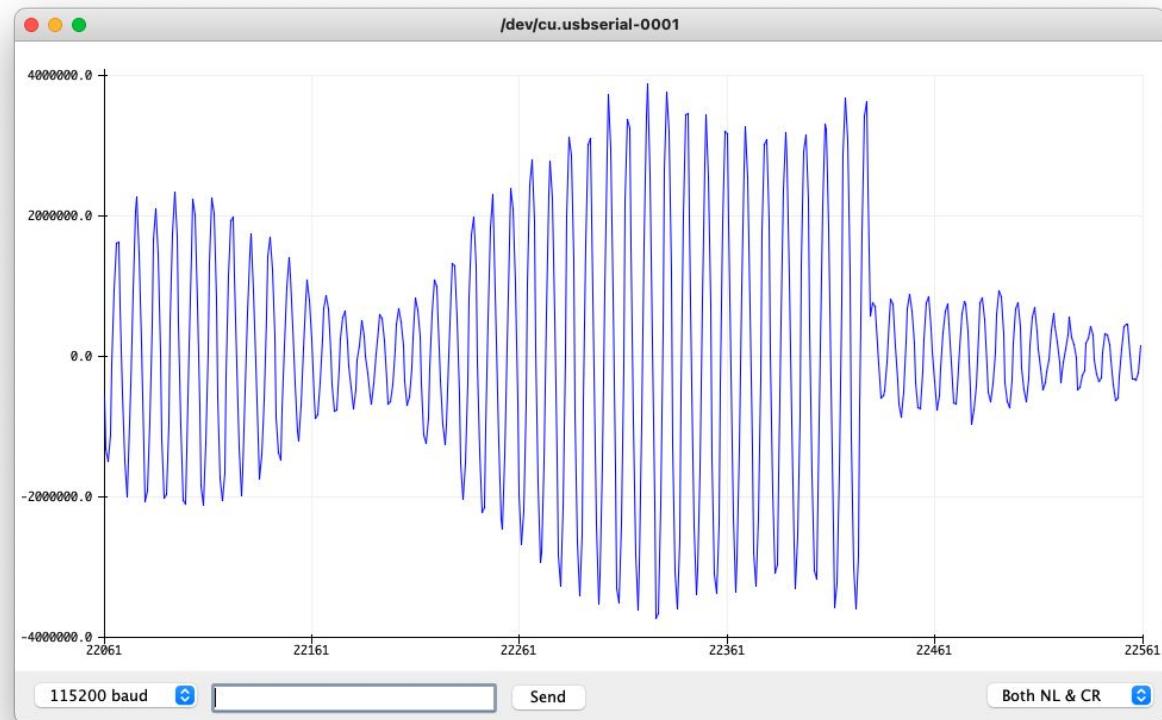
```
40
41 void setup()
42 {
43     // we need serial output for the plotter
44     Serial.begin(115200);
45     // start up the I2S peripheral
46     i2s_driver_install(I2S_NUM_0, &i2s_config, 0, NULL);
47     i2s_set_pin(I2S_NUM_0, &i2s_mic_pins);
48 }
49
50 int32_t raw_samples[SAMPLE_BUFFER_SIZE]; //sampleBuffer
51 void loop()
52 {
53     // read from the I2S device
54     size_t bytes_read = 0;
55     i2s_read(I2S_NUM_0, raw_samples, sizeof(int32_t) * SAMPLE_BUFFER_SIZE, &bytes_read, portMAX_DELAY);
56     int samples_read = bytes_read / sizeof(int32_t);
57     // dump the samples out to the serial channel.
58     for (int i = 0; i < samples_read; i++)
59     {
60         Serial.printf("%ld\n", raw_samples[i]);
61     }
62 }

Done uploading.

Leaving...
Hard resetting via RTS pin...

58 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on /dev/cu.usbserial-0001
```

Test ESP32 + Mic



Graph showing raw audio data from /dev/cu.usbserial-0001. The Y-axis ranges from -4000000.0 to 4000000.0. The X-axis shows time points at 22061, 22161, 22261, 22361, 22461, and 22561. The signal is highly oscillatory, representing raw microphone input.

12:44:41.147 -> -51200
12:44:41.147 -> -62466
12:44:41.147 -> -147456
12:44:41.147 -> -91648
12:44:41.147 -> -77314
12:44:41.147 -> -87042
12:44:41.147 -> -67586
12:44:41.147 -> -133120
12:44:41.147 -> -52226
12:44:41.147 -> -79360
12:44:41.147 -> -121344
12:44:41.147 -> -22016
12:44:41.147 -> -60930
12:44:41.147 -> -1536
12:44:41.147 -> -105474
12:44:41.147 -> -72706
12:44:41.147 -> 23040
12:44:41.147 -> -106496
12:44:41.147 -> -16898
12:44:41.147 -> -49152
12:44:41.147 -> 9726
12:44:41.147 -> -

Autoscroll Show timestamp Both NL & CR 115200 baud

studio.edgeimpulse.com/studio/profile/projects

EDGE IMPULSE

Projects Custom ML blocks

Projects

+ Create new project

Create a new project

Enter the name for your new project:

ESP32-NMP441-KWS

Choose your project type:

Developer
20 min job limit, 4GB or 4 hours of data, limited collaboration.

Enterprise
No job or data size limits, higher performance, custom blocks.

Create under organization: Edge Impulse Experts

Create new project

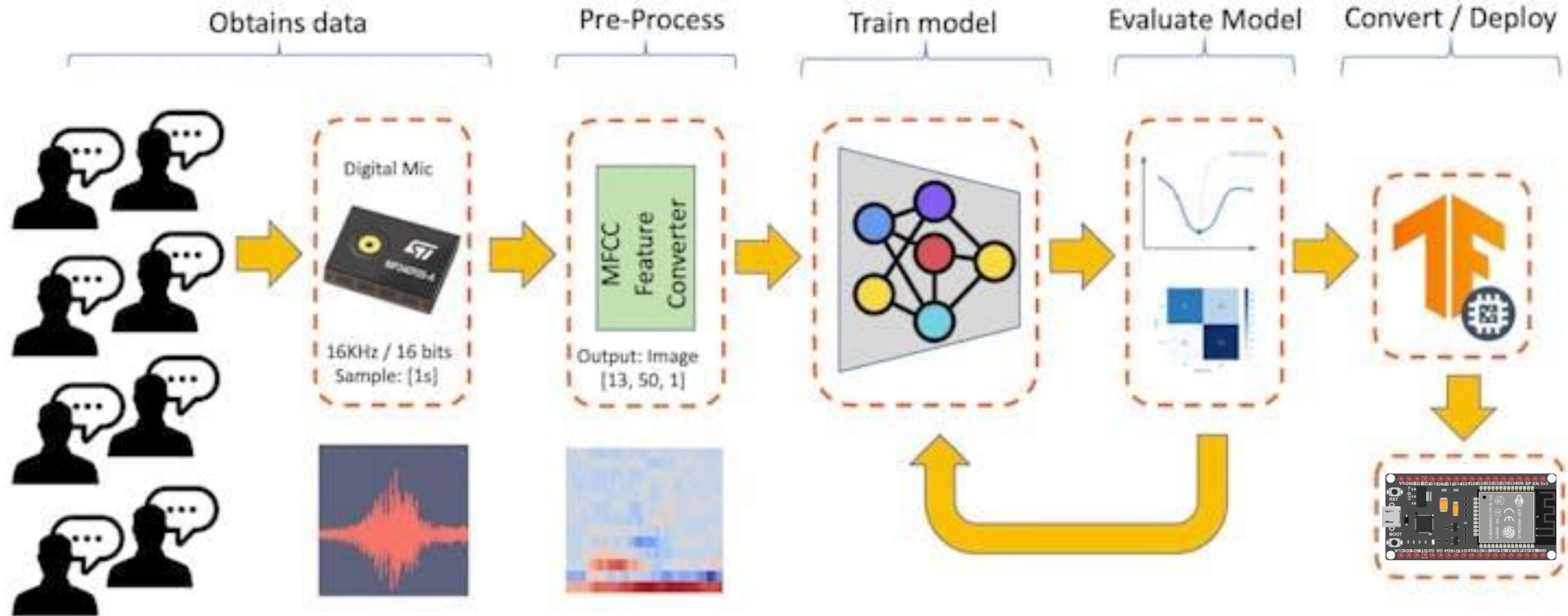
MJRoBot (Marcelo Rovai) / Accelerometer-Nano-Ble-IoT

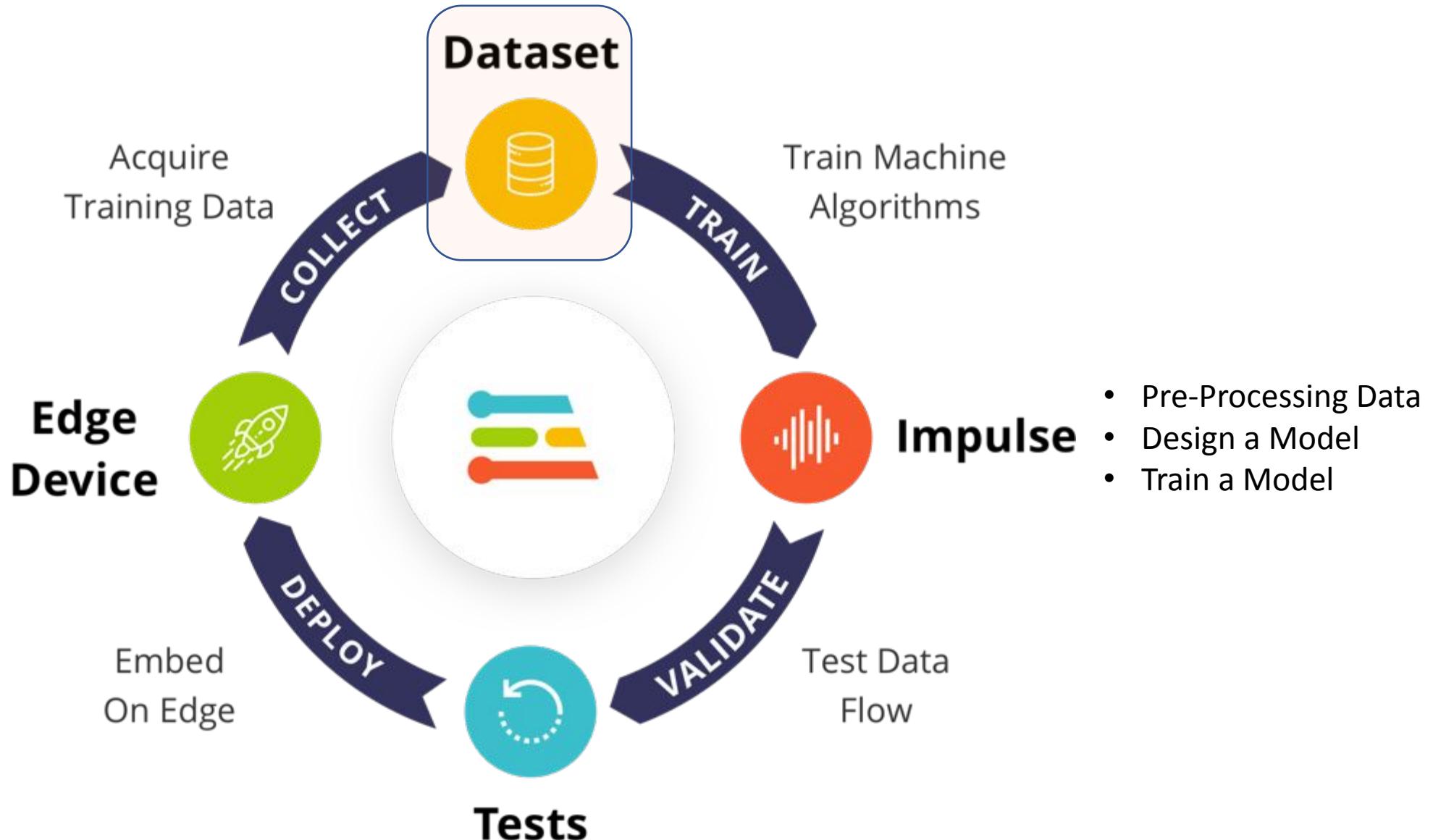
MJRoBot (Marcelo Rovai) / video_tinyml_raw

MJRoBot (Marcelo Rovai) / Pico_Motion_Detection PUBLIC

INMP441.fpz INMP441.fpz I2S_Timing.svg I2S_Timing.svg 720x1280-yes-n.jpg 720x1280-yes-n.jpg ei-esp32-nmp441-kws.zip ei-esp32-nmp441-kws.zip ei-iesti01-2023-1....zip ei-iesti01-2023-1....zip Show All

KeyWord Spotting (KWS) - Inference





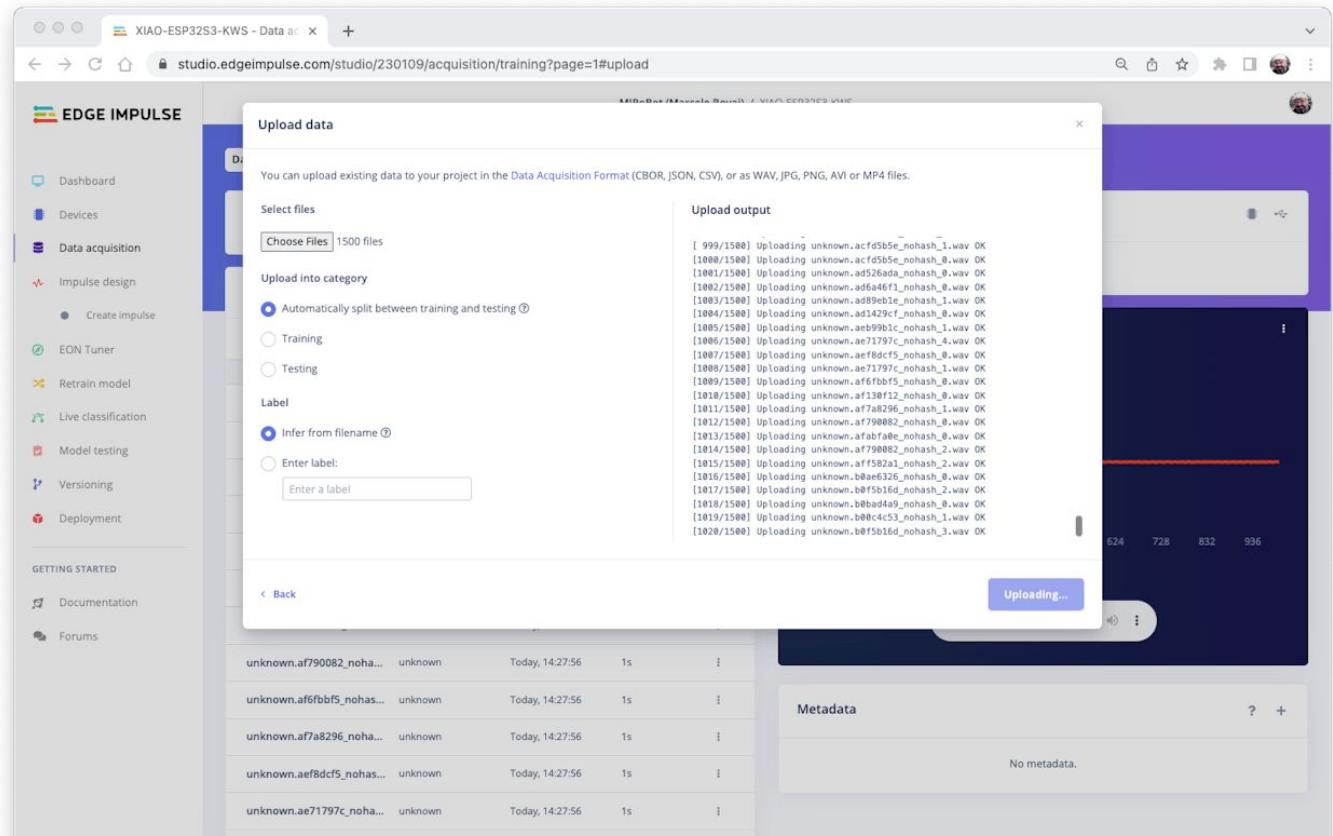
Dataset – Speech Commands

Pete Warden's "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition".

- 35 keywords (with +1, 000 samples each), such as yes, no, stop, and go.
- In some words, we can get 1,500 samples, such as yes and no.

Download a small portion of the dataset from Edge Studio (Keyword spotting pre-built dataset), which includes samples from the four classes we will use in this project: **yes, no, noise, and background**. For this, follow the steps below:

- Download the [keywords dataset](#).
- Unzip the file in a location of your choice.
- Upload to the Studio



XIAO-ESP32S3-KWS - Data acquisition

studio.edgeimpulse.com/studio/230109/acquisition/training?page=1#

EDGE IMPULSE

Dataset Data explorer Data sources | CSV Wizard

DATA COLLECTED
1h 40m 13s

TRAIN / TEST SPLIT
80% / 20%

Collect data

Connect a device to start building your dataset.

Dataset

Training (4,830) Test (1,183)

Apply filters

Clear filters

By label

no (1204)

noise (1229)

unknown (1202)

yes (1195)

By name

Enter a sample name

By signature validity

Valid & invalid signatures

Enabled & disabled samples

Enabled & disabled samples

Delete (0) Edit labels (0) Move to test set (0) Enable (0) Disable (0)

SAMPLE NAME LABEL ADDED LENGTH

unknown.ff21fb59_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	
unknown.fe1916ba_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	
unknown.ff4ed4f3_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	
unknown.feb1d305_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	
unknown.ffb86d3c_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	
unknown.fe5c4a7a_nohas... <input type="checkbox"/>	unknown	Today, 14:28:33	1s	

RAW DATA

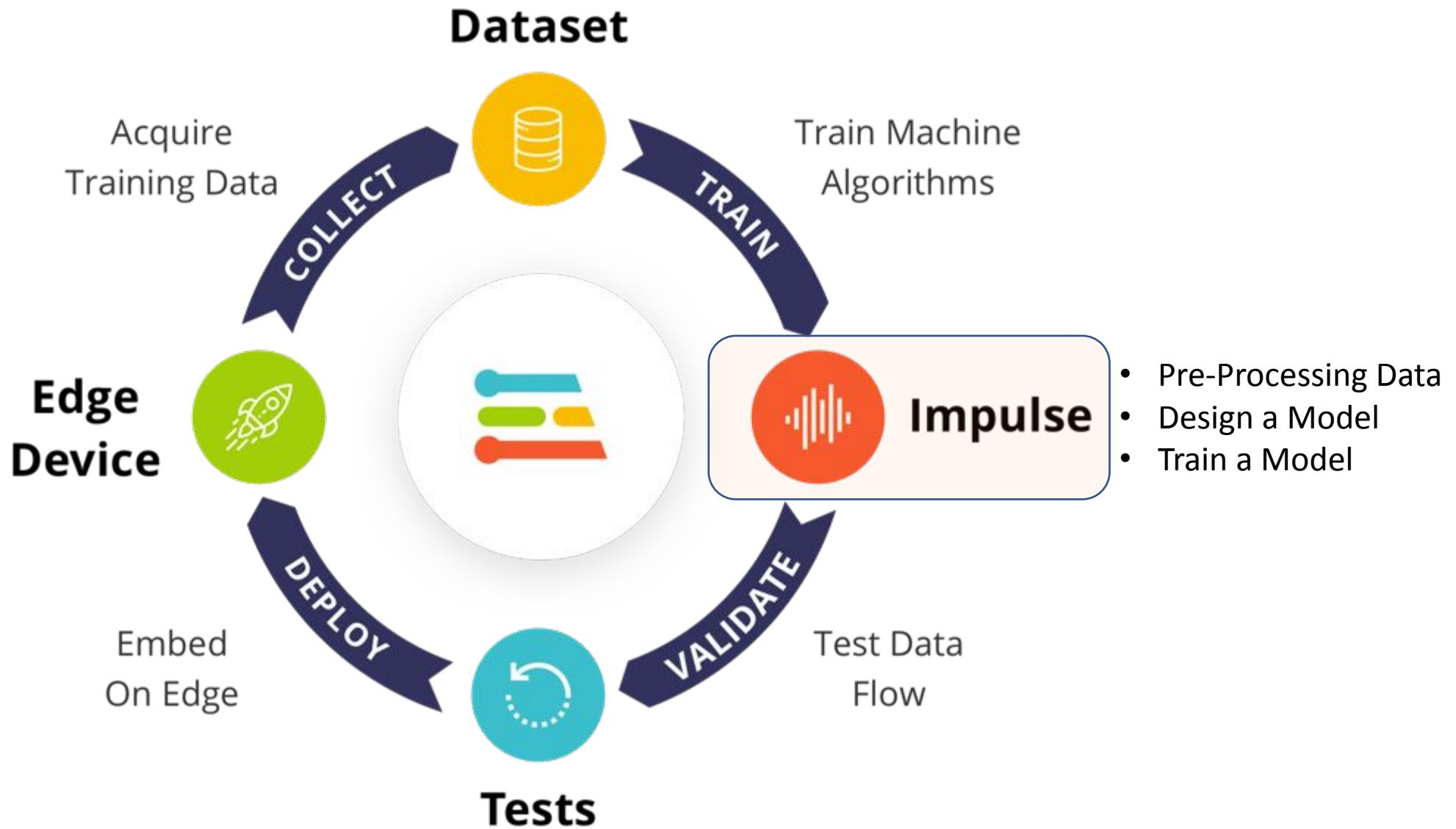
unknown.ff21fb59_nohash_0

audio

0:00 / 0:00

Metadata

No metadata.



XIAO-ESP32S3-KWS - Create

studio.edgeimpulse.com/studio/230109/create-impulse

EDGE IMPULSE

MJRoBot (Marcelo Rovai) / XIAO-ESP32S3-KWS

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

Time series data

Input axes: audio

Window size: 1000 ms.

Window increase: 500 ms.

Frequency (Hz): 16000

Zero-pad data:

Audio (MFCC)

Name: MFCC

Input axes (1): audio

Classification

Name: Classifier

Input features: MFCC

Output features: 4 (no, noise, unknown, yes)

Output features: 4 (no, noise, unknown, yes)

Save Impulse

Add a processing block

Add a learning block

© 2023 EdgeImpulse Inc. All rights reserved

XIAO-ESP32S3-KWS - MFCC

studio.edgeimpulse.com/studio/230109/dsp/mfcc/3

Raw data

Show: yes
yes.ffb86d3c_nohash_0 (yes)

Raw features

144, 201, 157, 168, 123, 113, 26, -31, -19, -19, -29, -5, 17, -35, -17, 31, 65, 61, ...

Parameters

Autotune parameters

Mel Frequency Cepstral Coefficients

Number of coefficients: 13

Frame length: 0.025

Frame stride: 0.02

Filter number: 32

FFT length: 512

Normalization window size: 151

Low frequency: 80

High frequency: Click to set

Pre-emphasis

Coefficient: 0.98

Save parameters

DSP result

Cepstral Coefficients

Processed features

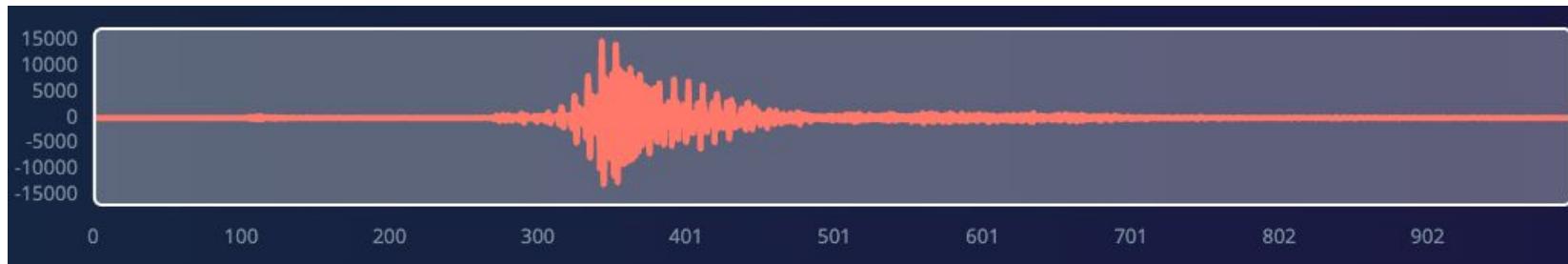
-1.3118, 0.6242, -1.1163, -0.3401, 0.3516, -0.5983, -0.0561, -1.6334, -2.4966, -1.03...

On-device performance

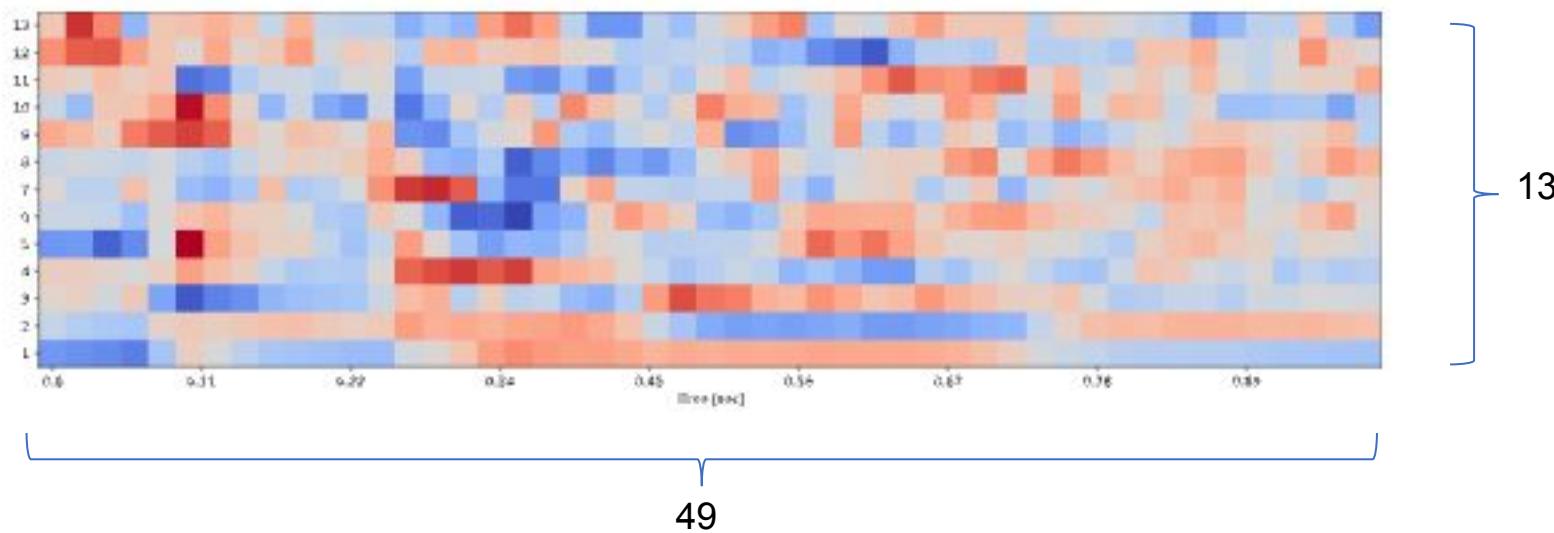
PROCESSING TIME: 675 ms.

PEAK RAM USAGE: 16 KB

Raw data -> 16,000 features



Processed features -> 637 features (13 x 49)



XIAO-ESP32S3-KWS - MFCC

studio.edgeimpulse.com/studio/230109/dsp/mfcc/3/generate-features

MJRoBot (Marcelo Roval) / XIAO-ESP32S3-KWS

EDGE IMPULSE

#1 Click to set a description for this version

Parameters Generate features

Training set

Data in training set	1h 20m 30s
Classes	4 (no, noise, unknown, yes)
Training windows	4,830

Generate features

Feature generation output

```
completed 100 / 500 epochs  
Still running...  
completed 150 / 500 epochs  
completed 200 / 500 epochs  
completed 250 / 500 epochs  
Still running...  
completed 300 / 500 epochs  
completed 350 / 500 epochs  
completed 400 / 500 epochs  
Still running...  
completed 450 / 500 epochs  
Mon May 22 18:41:49 2023 Finished embedding  
Reducing dimensions for visualizations OK  
Job completed
```

Feature explorer

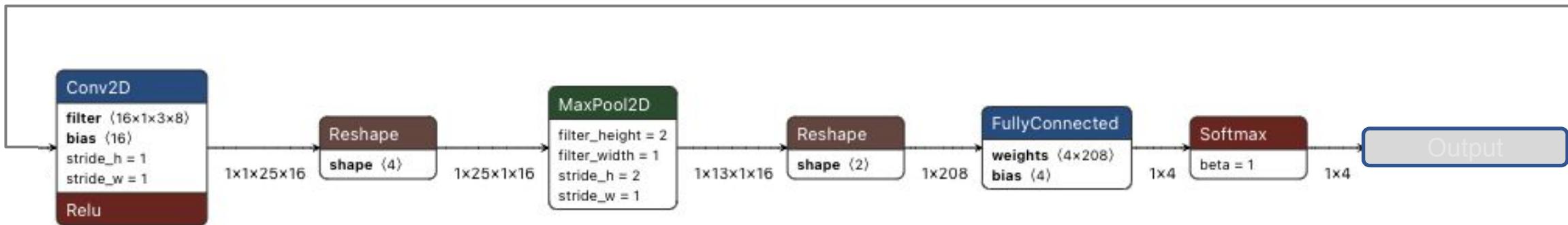
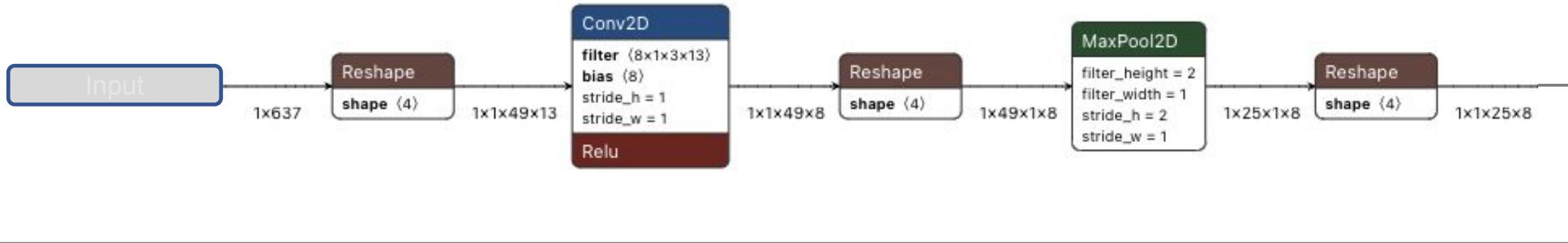
On-device performance

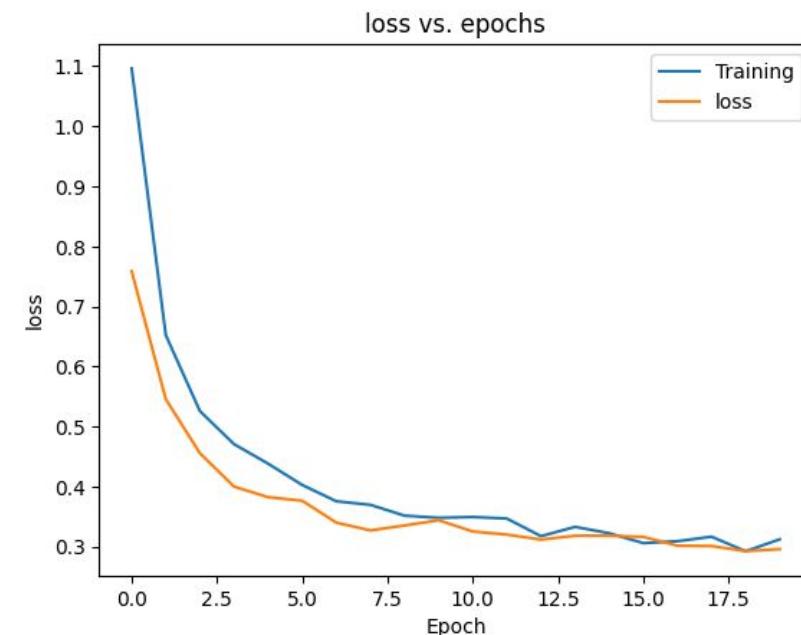
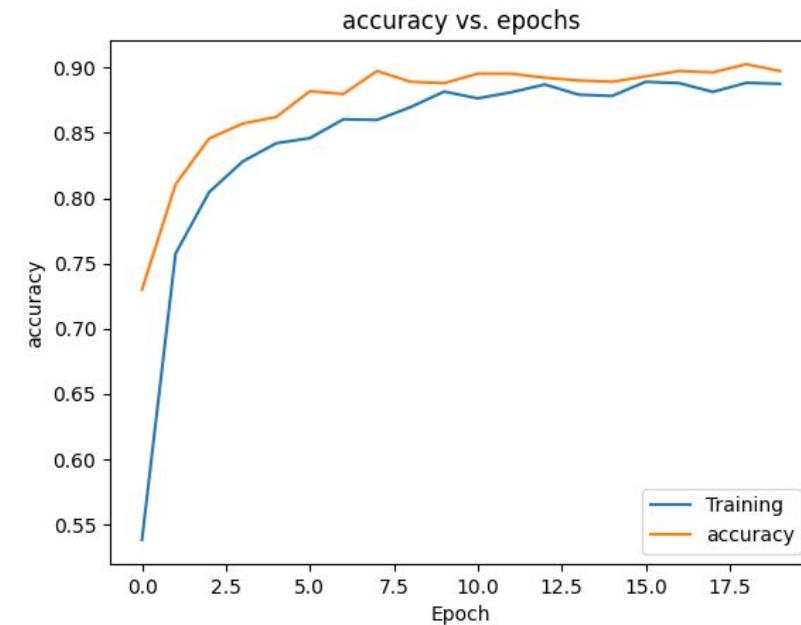
PROCESSING TIME	675 ms.
PEAK RAM USAGE	16 KB

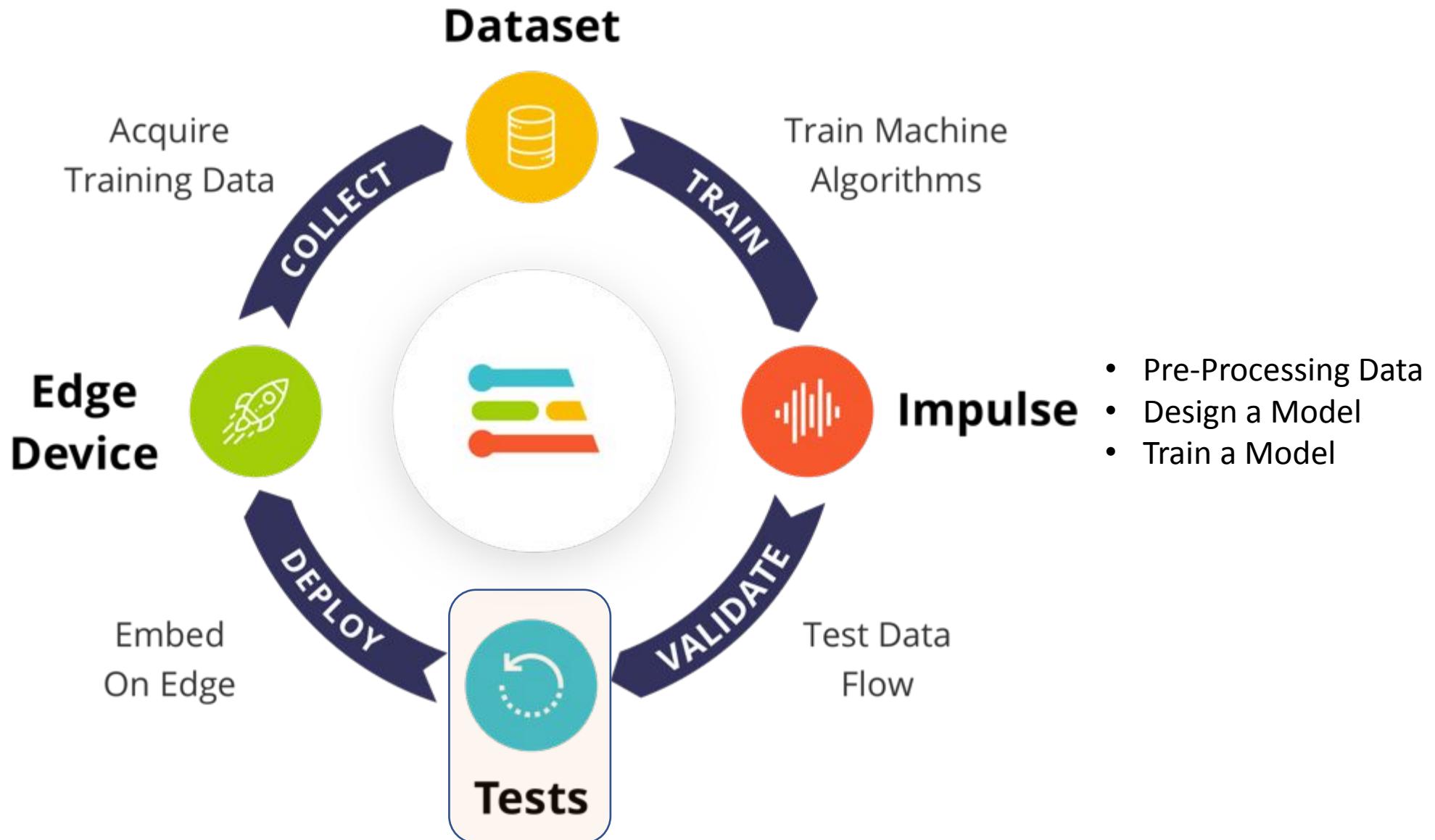
GETTING STARTED

Documentation

Forums







Model testing results

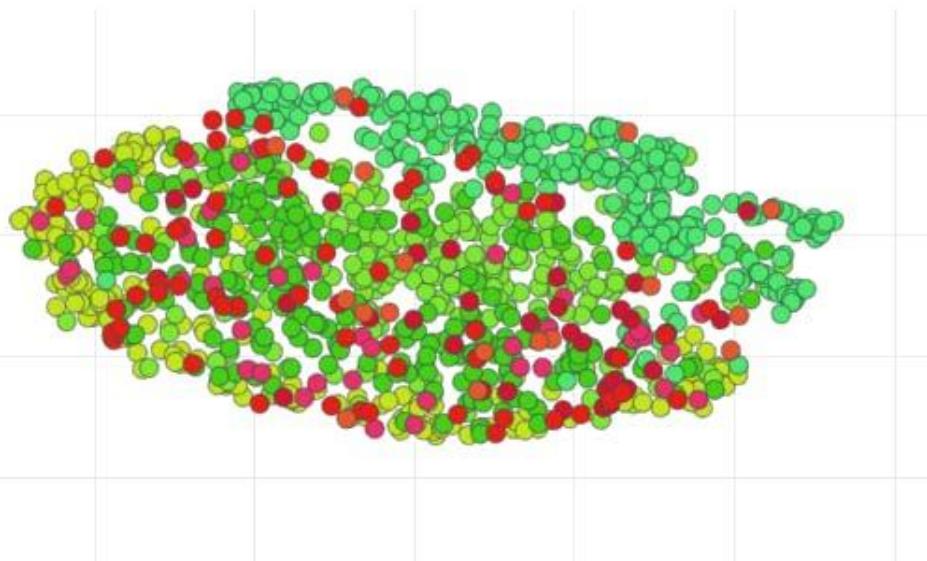


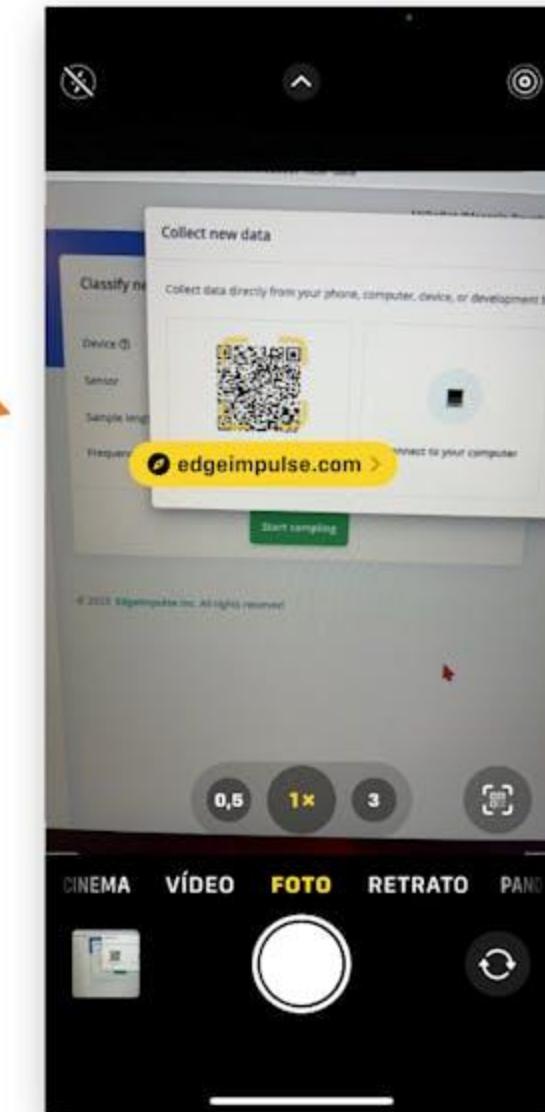
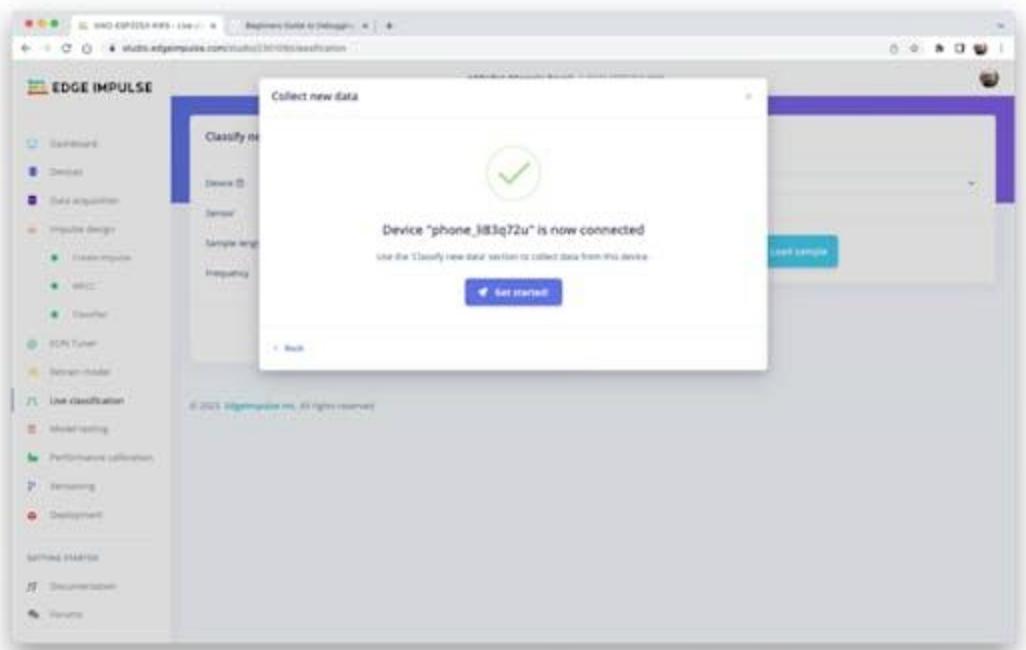
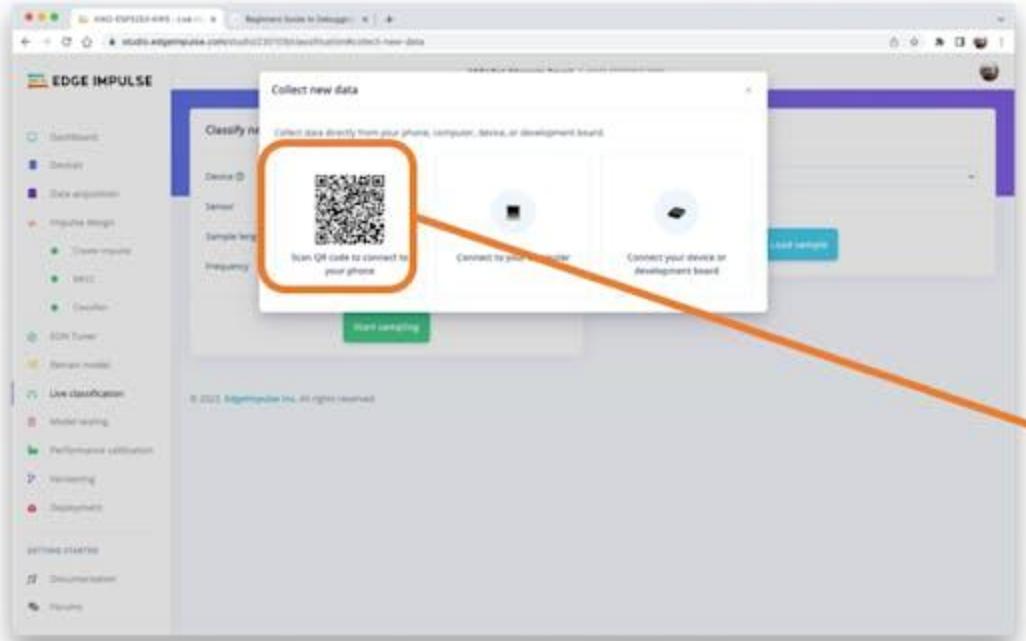
ACCURACY
86.73%

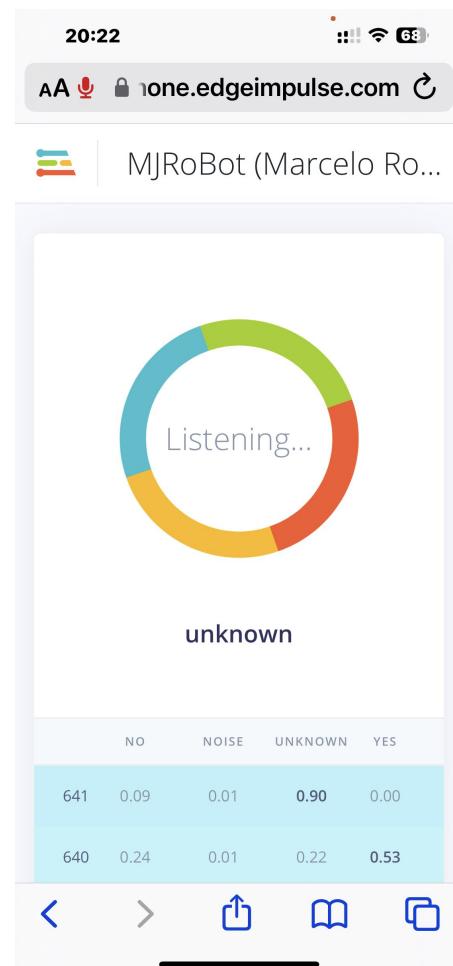
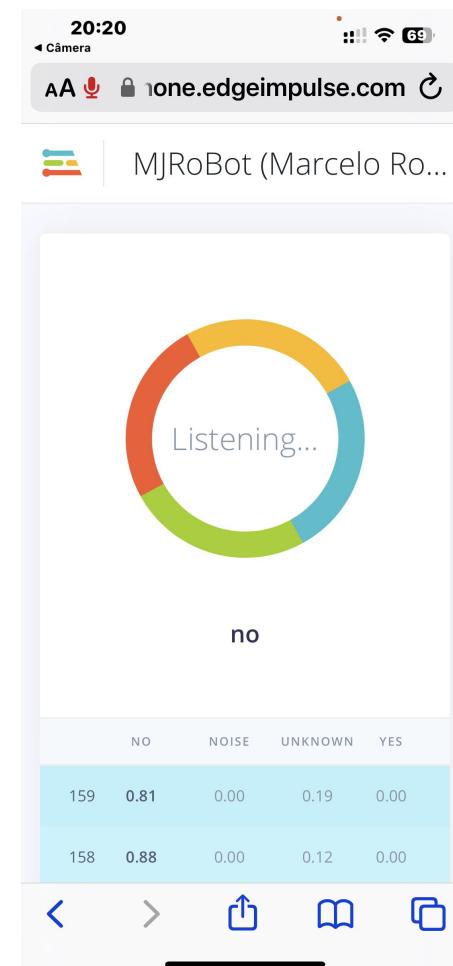
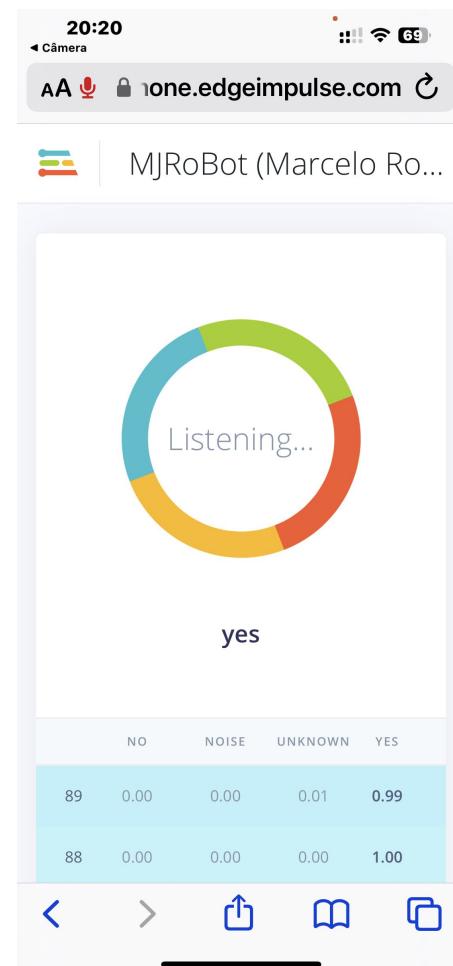
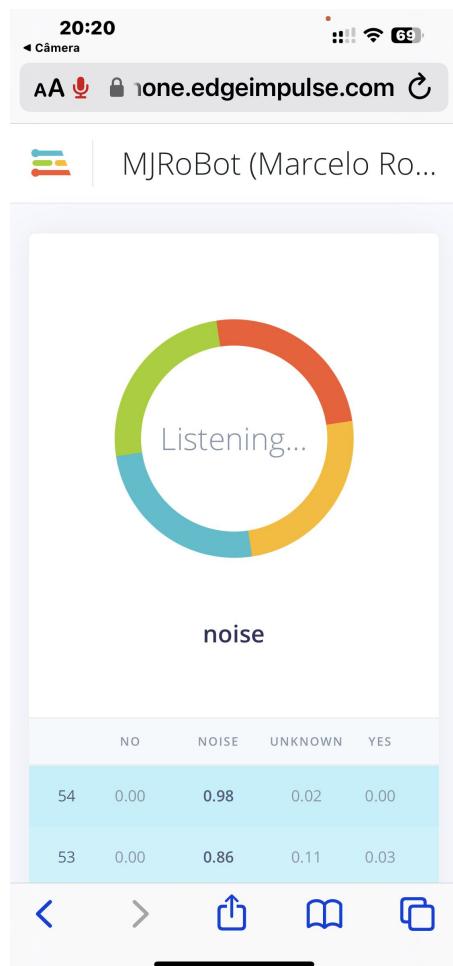
	NO	NOISE	UNKNOWN	YES	UNCERTAIN
NO	86.3%	0.7%	3.9%	1.4%	7.7%
NOISE	0%	88.6%	3.3%	0.7%	7.5%
UNKNOWN	4.4%	2.7%	78.1%	1.7%	13.1%
YES	0.3%	0%	0.7%	93.9%	5.1%
F1 SCORE	0.90	0.92	0.84	0.95	

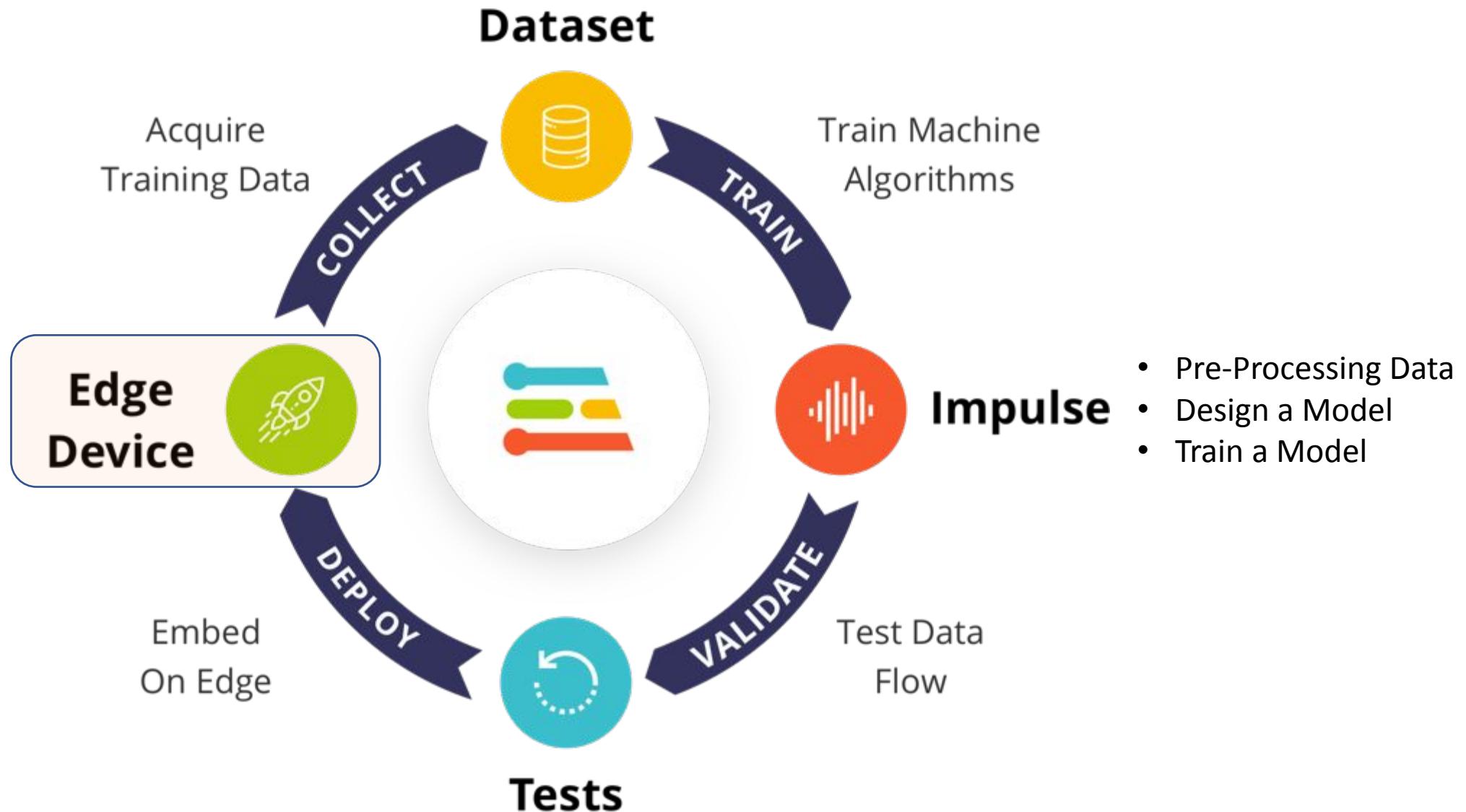
Feature explorer

- no - correct
- noise - correct
- unknown - correct
- yes - correct
- no - incorrect
- noise - incorrect
- unknown - incorrect
- yes - incorrect









Configure your deployment

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

Arduino library x

SELECTED DEPLOYMENT


Arduino library
 An Arduino library with examples that runs on most Arm-based Arduino development boards.

MODEL OPTIMIZATIONS
 Model optimizations can increase on-device performance but may reduce accuracy.

Enable EON™ Compiler *Same accuracy, up to 50% less memory. [Learn more](#)*

Quantized (int8)

	MFCC	CLASSIFIER	TOTAL
LATENCY	675 ms.	6 ms.	681 ms.
RAM	15.6K	6.0K	15.6K
FLASH	-	49.9K	-
ACCURACY			-

Unoptimized (float32)

	MFCC	CLASSIFIER	TOTAL
LATENCY	675 ms.	31 ms.	706 ms.
RAM	15.6K	10.5K	15.6K
FLASH	-	53.2K	-
ACCURACY			-

To compare model accuracy, run model testing. Run model testing

Estimate for Espressif ESP-EYE (ESP32 240MHz) - Change target

Build

esp32_microphone | Arduino 1.8.19

```

250 i2s_config_t i2s_config = {
251   .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX | I2S_MODE_TX),
252   .sample_rate = sampling_rate,
253   .bits_per_sample = (i2s_bits_per_sample_t)16,
254   .channel_format = I2S_CHANNEL_FMT_ONLY_RIGHT,
255   .communication_format = I2S_COMM_FORMAT_I2S,
256   .intr_alloc_flags = 0,
257   .dma_buf_count = 8,
258   .dma_buf_len = 512,
259   .use_apll = false,
260   .tx_desc_auto_clear = false,
261   .fixed_mclk = -1,
262 };
263 i2s_pin_config_t pin_config = {
264   .bck_io_num = 17, // IIS_SCLK
265   .ws_io_num = 18, // IIS_LCLK
266   .data_out_num = -1, // IIS_DSIN
267   .data_in_num = 19, // IIS_DOUT
268 };
269 esp_err_t ret = 0;
270
271 ret = i2s_driver_install((i2s_port_t)1, &i2s_config, 0, NULL);
272 if (ret != ESP_OK) {
273   ei_printf("Error in i2s_driver_install");
274 }
275
276 ret = i2s_set_pin((i2s_port_t)1, &pin_config);
277 if (ret != ESP_OK) {
278   ei_printf("Error in i2s_set_pin");
279 }

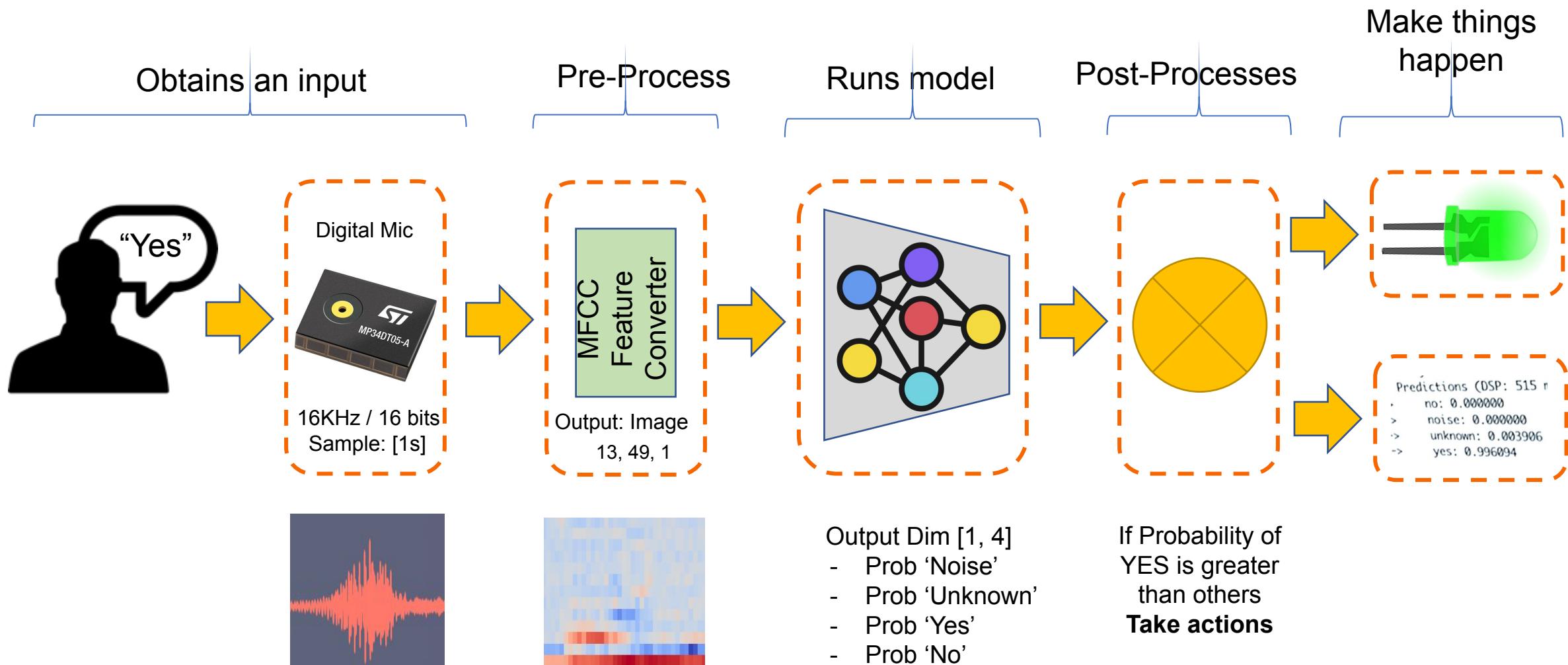
Done uploading.

Leaving...
Hard resetting via RTS pin...

```

47 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None, Disabled on /dev/cu.usbserial-0001

KeyWord Spotting (KWS) - Inference



/dev/cu.usbserial-0001

Send

```
13:22:34.389 -> yes: 0.007813
13:22:35.354 -> Predictions (DSP: 542 ms., Classification: 4 ms., Anomaly: 0 ms.):
13:22:35.354 -> no: 0.585937
13:22:35.354 -> noise: 0.000000
13:22:35.354 -> unknown: 0.011719
13:22:35.354 -> yes: 0.402344
13:22:36.375 -> Predictions (DSP: 542 ms., Classification: 4 ms., Anomaly: 0 ms.):
13:22:36.409 -> no: 0.015625
13:22:36.409 -> noise: 0.921875
13:22:36.409 -> unknown: 0.054687
13:22:36.409 -> yes: 0.007813
13:22:37.331 -> Predictions (DSP: 542 ms., Classification: 4 ms., Anomaly: 0 ms.):
13:22:37.364 -> no: 0.000000
13:22:37.364 -> noise: 0.996094
13:22:37.364 -> unknown: 0.003906
13:22:37.364 -> yes: 0.000000
13:22:38.359 -> Predictions (DSP: 542 ms., Classification: 4 ms., Anomaly: 0 ms.):
13:22:38.359 -> no: 0.003906
13:22:38.359 -> noise: 0.000000
13:22:38.359 -> unknown: 0.003906
13:22:38.359 -> yes: 0.988281
```

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

```

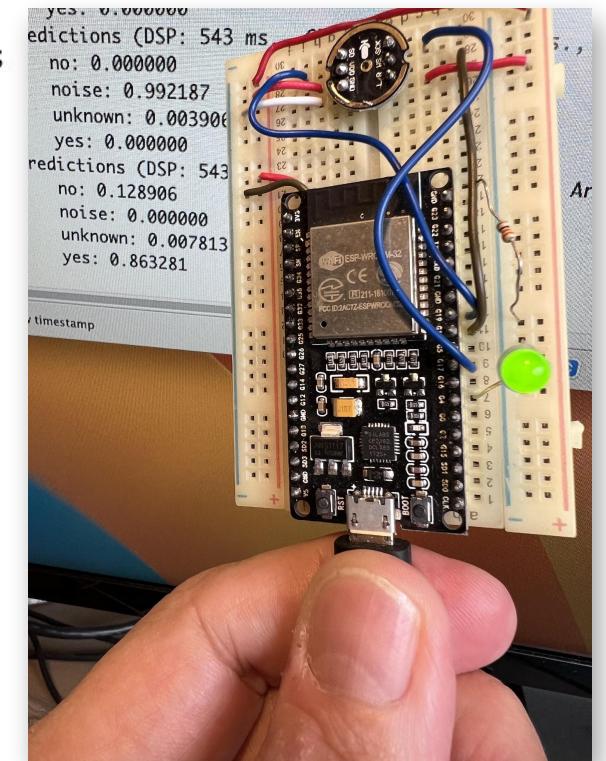
41 /* Includes -----
42 #include <ESP32-NMP441-KWS_inferencing.h>
43
44 #include "freertos/FreeRTOS.h"
45 #include "freertos/task.h"
46
47 #include "driver/i2s.h"
48
49 #define LED_BUILT_IN 4
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 void setup()
69 {
70     // put your setup code here, to run once:
71     Serial.begin(115200);
72     // comment out the below line to cancel the wait
73     while (!Serial);
74     Serial.println("Edge Impulse Inferencing Demo");
75
76     pinMode(LED_BUILT_IN, OUTPUT);

```

```

120     int pred_index = 0;      // Initialize pred_index
121     float pred_value = 0;    // Initialize pred_value
122
123     // print the predictions
124     ei_printf("Predictions ");
125     ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",
126               result.timing.dsp, result.timing.classification, result.timing.anomaly);
127     ei_printf(": \n");
128     for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
129         ei_printf("    %s: ", result.classification[ix].label);
130         ei_printf_float(result.classification[ix].value);
131         ei_printf("\n");
132
133         if (result.classification[ix].value > pred_value){
134             pred_index = ix;
135             pred_value = result.classification[ix].value;
136         }
137     }
138
139     // show the inference result on LED
140     if (pred_index == 3){
141         digitalWrite(LED_BUILT_IN, HIGH); //Turn on
142     }
143     else{
144         digitalWrite(LED_BUILT_IN, LOW); //Turn off
145     }

```



Thanks



UNIFEI