

Edge Computer Vision

With Arduino PRO (Portenta/Nicla)



(Material)

Prof. Marcelo José Rovai

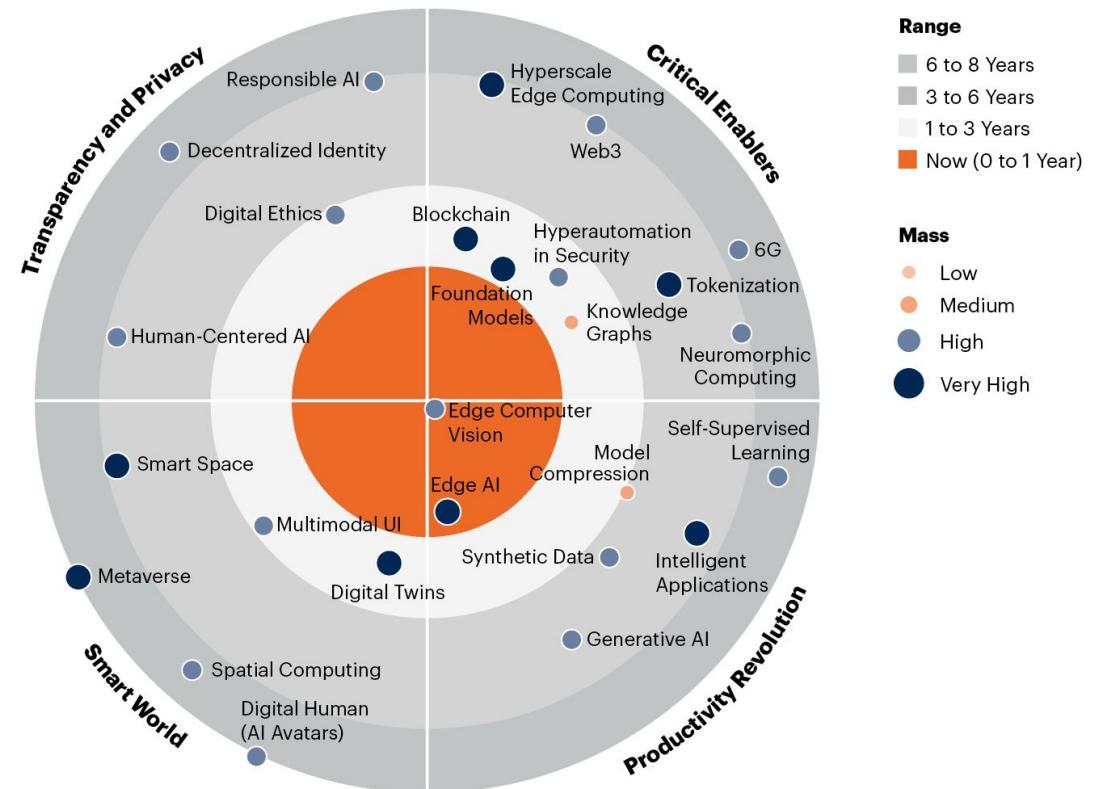
UNIFEI - Federal University of Itajubá, Brazil

TinyML4D Academic Network Co-Chair



UNIFEI

2023 Gartner Emerging Technologies and Trends Impact Radar



gartner.com

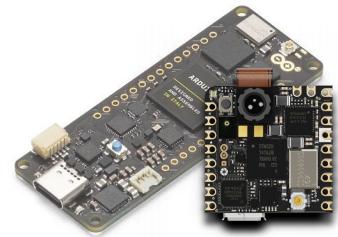
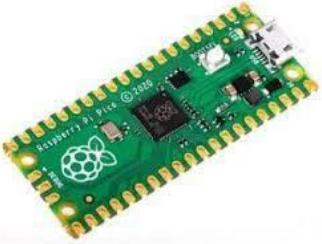
Note: Range measures number of years it will take the technology/trend to cross over from early adopter to early majority adoption. Mass indicates how substantial the impact of the technology or trend will be on existing products and markets.

Source: Gartner
© 2023 Gartner, Inc. All rights reserved. CM_GTS_2034284

Gartner®

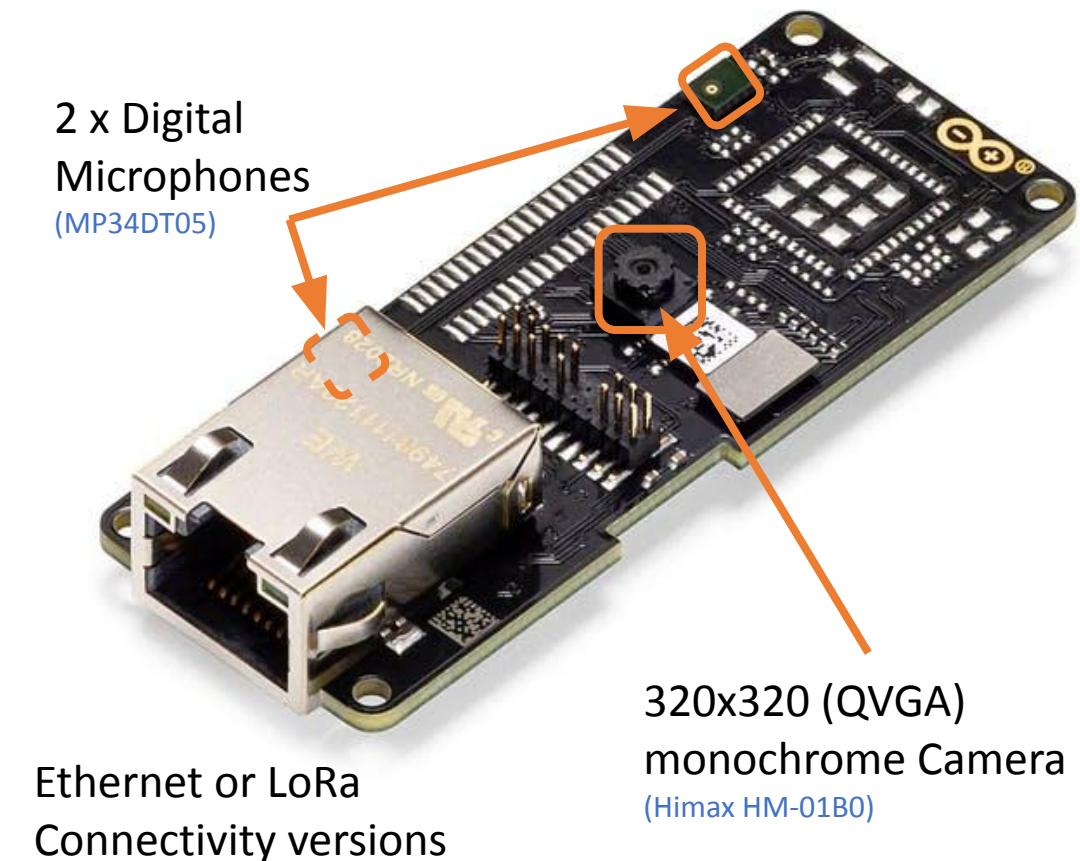
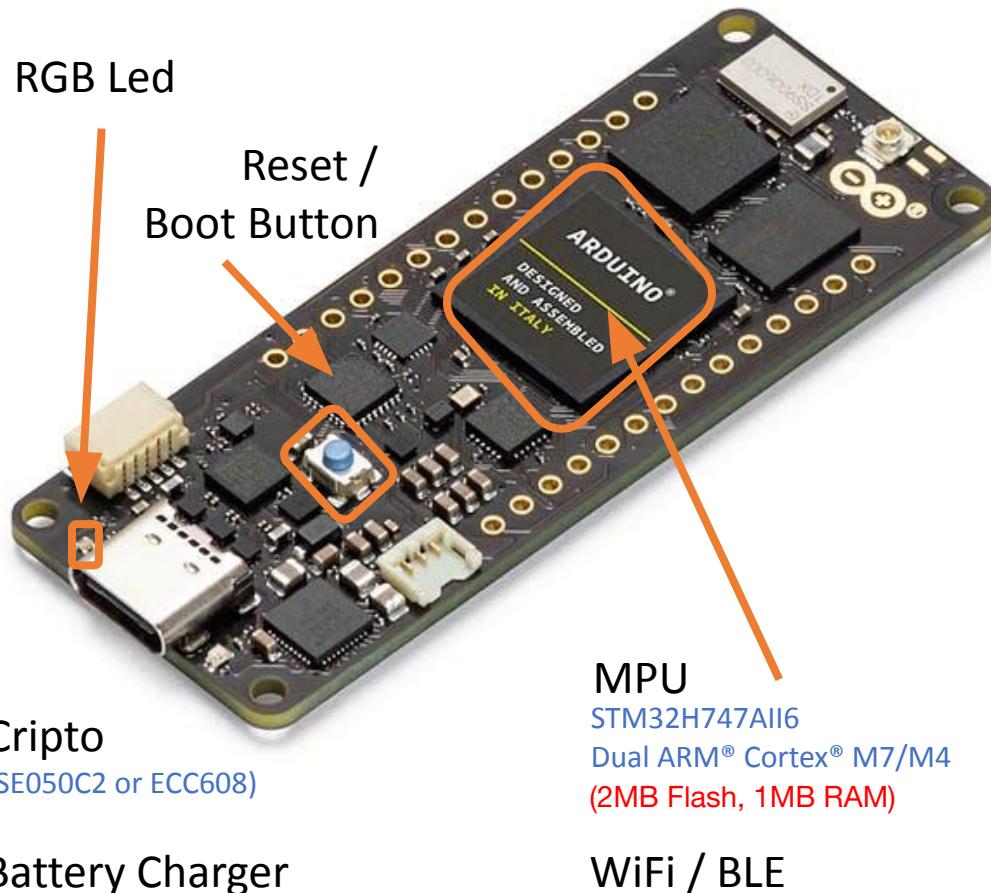
Edge Computer Vision has a very high impact potential, and it is for now!

Hardware



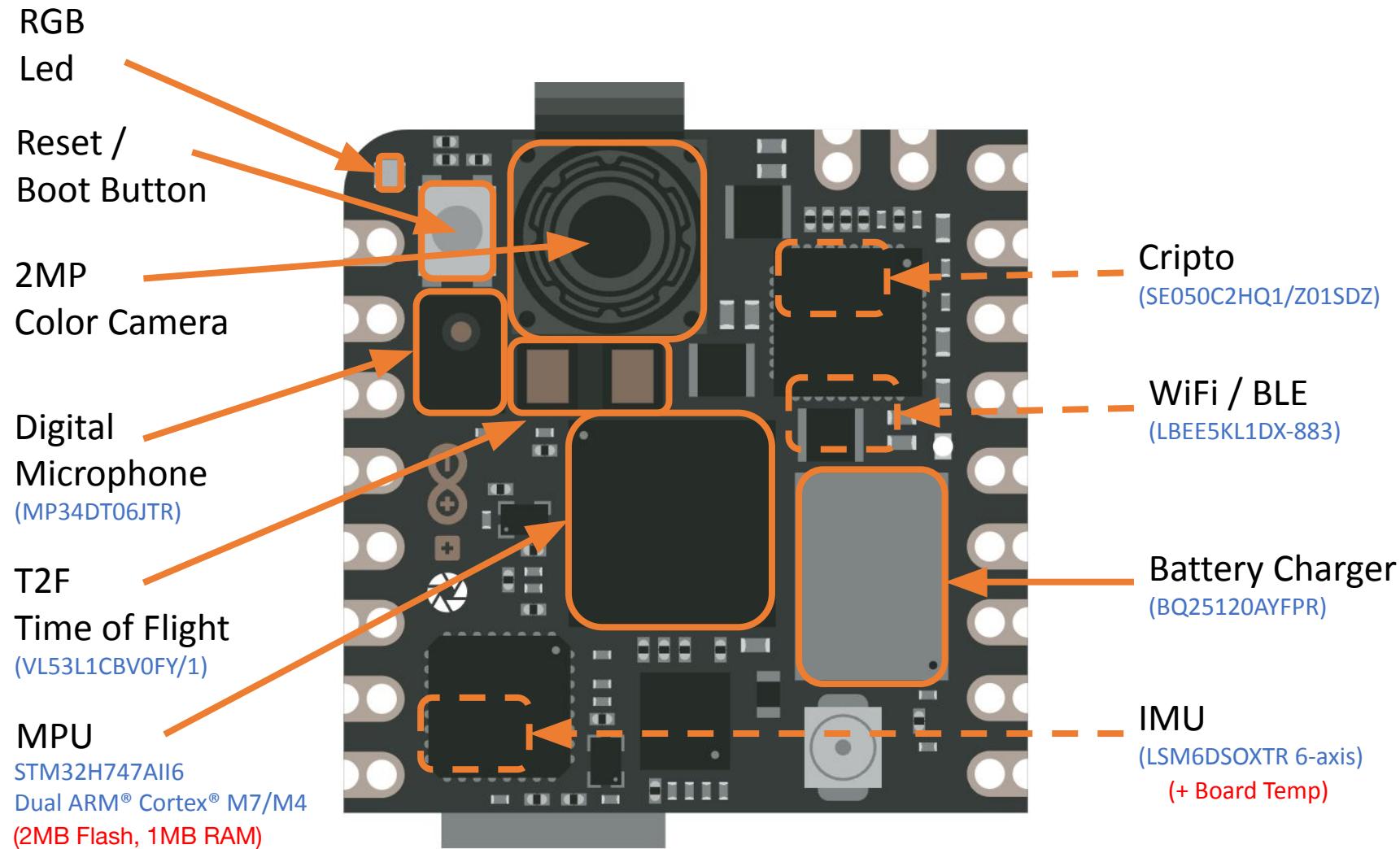
	Raspberry Pico (W)	Arduino Nano Sense	ESP 32	Seeed XIAO Sense / ESP32S3	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F (BLE) Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64 / 240MHz	480/240MHz
RAM	264KB	256KB	520KB (part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

Arduino Portenta H7 + Vision Shield



<https://docs.arduino.cc/hardware/portenta-h7>

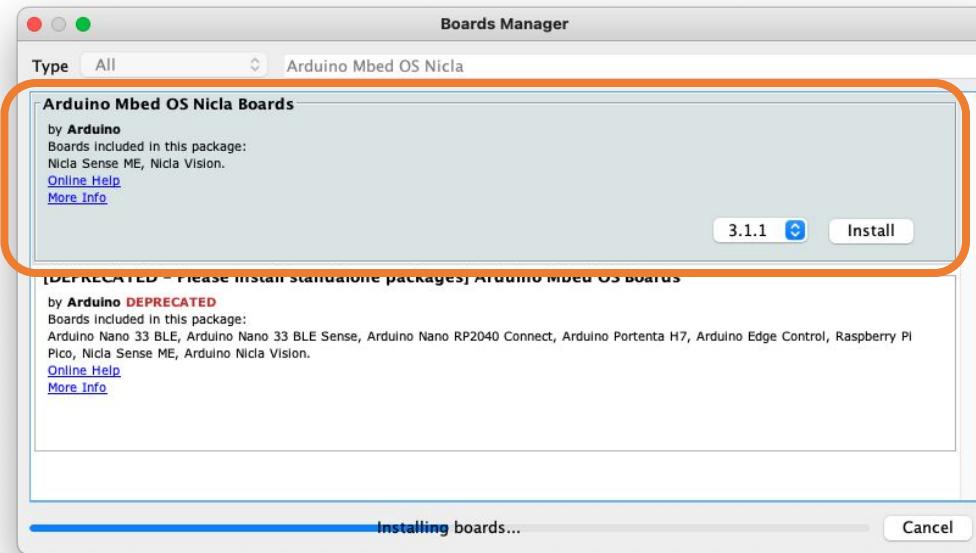
Arduino Nicla Vision



Install HW (Nicla-Vision)
& Sensors' test (optional)



Nicla Vision - Board Installation



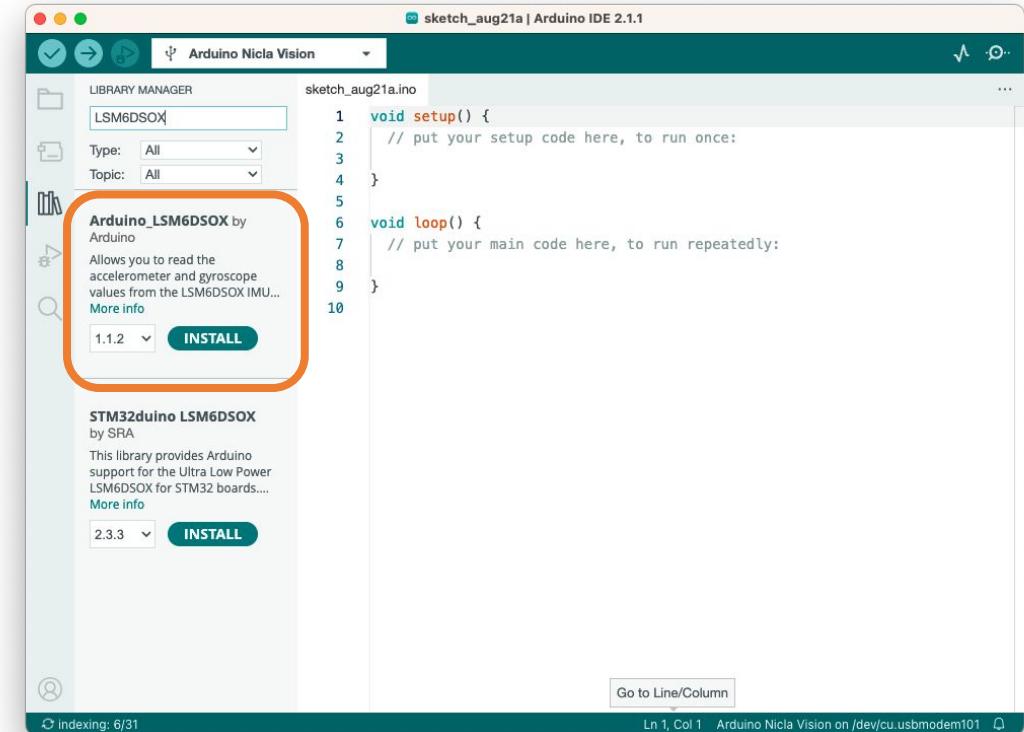
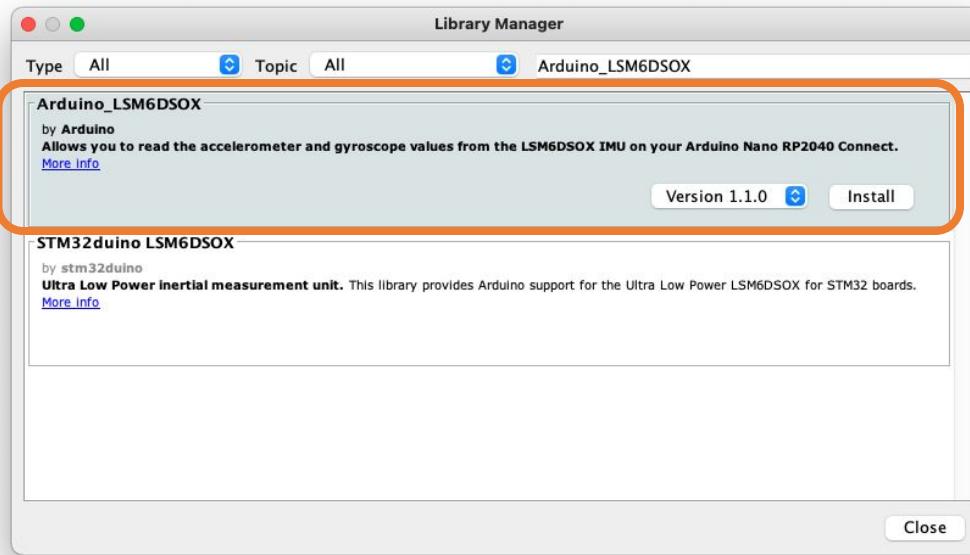
Arduino IDE 1.18



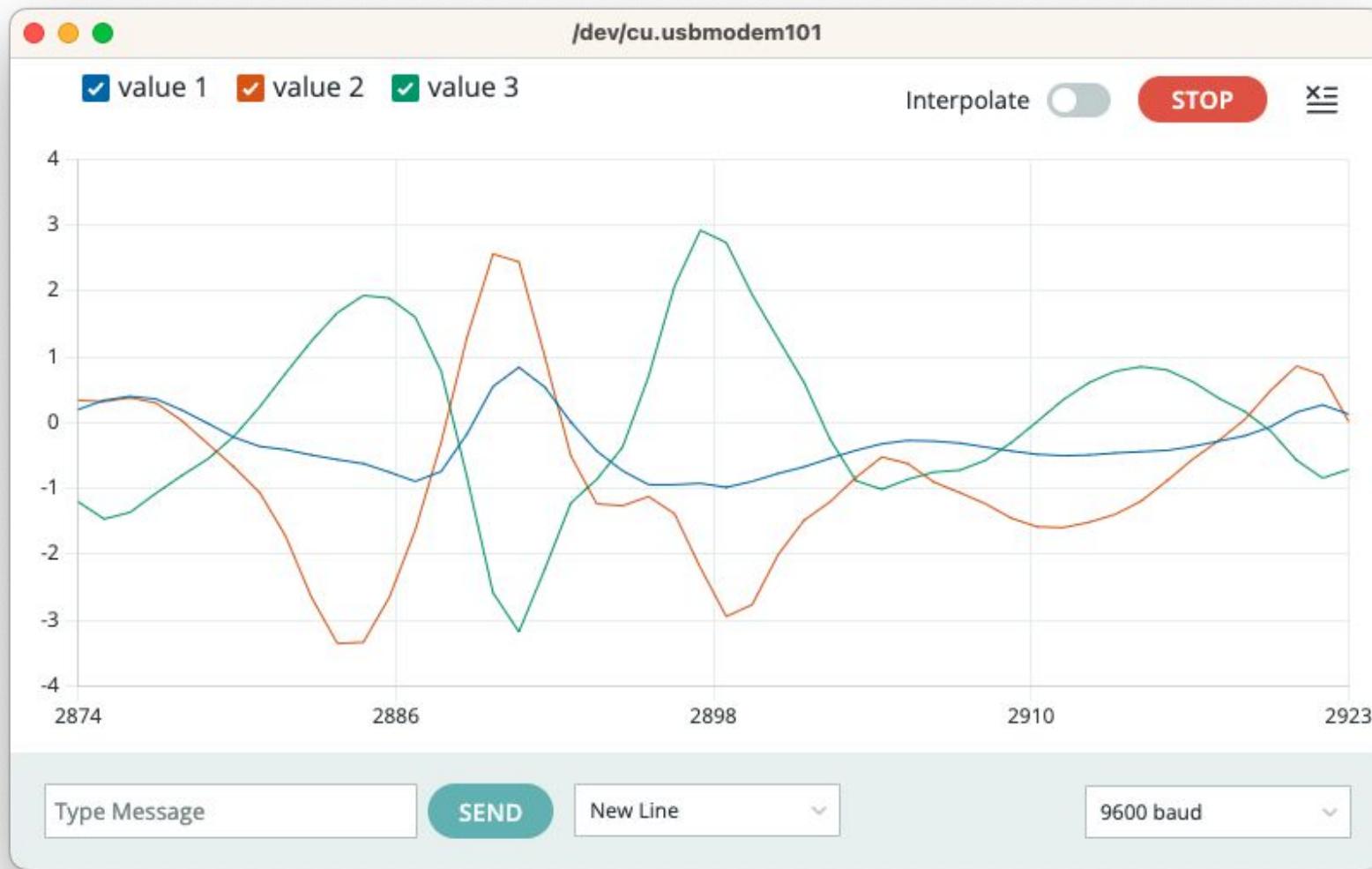
Arduino IDE 2.X

Test Hw ---> Run Blink (Green RGB Led)

IMU (LSM6DSOX) Library Installation



IMU (LSM6DSOX) Test



**Go to Examples >
Arduino_LSM6DSOX >
SimpleAccelerometer**

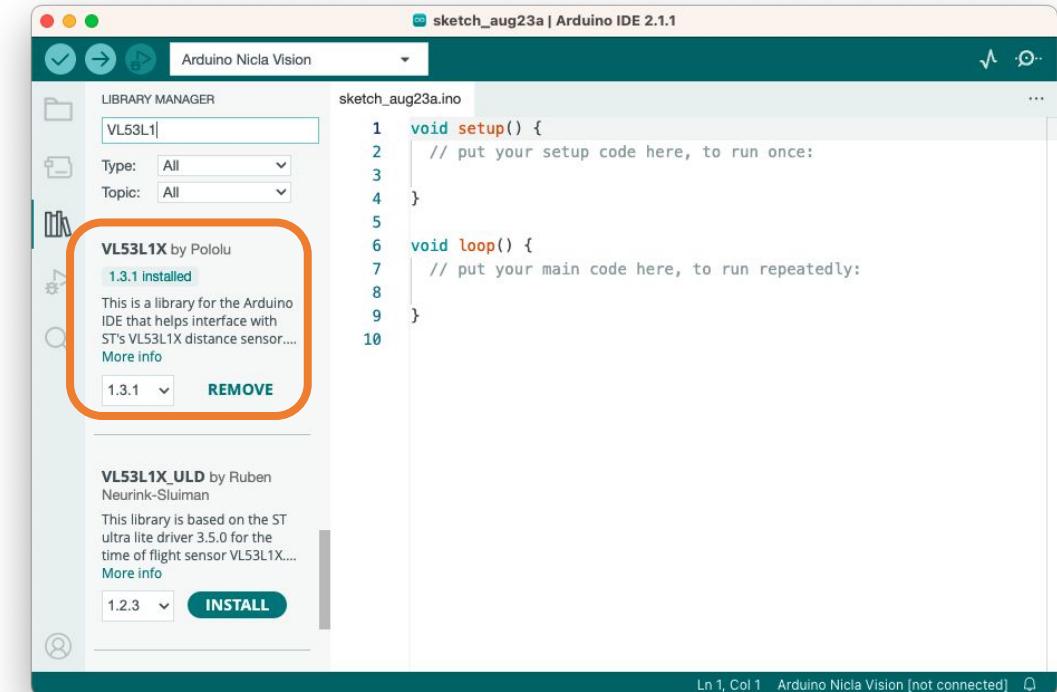
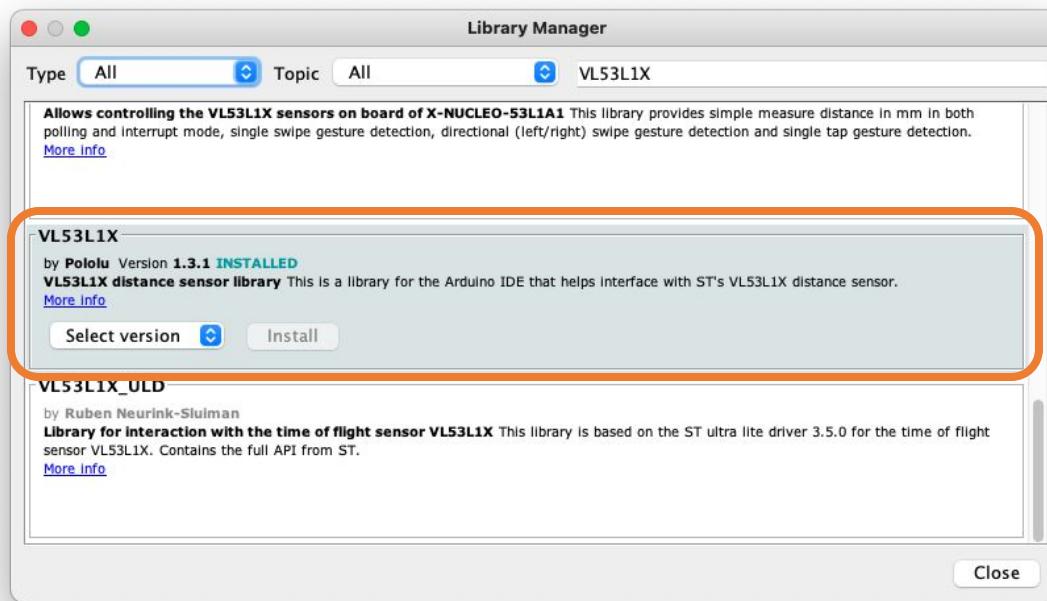
Run accelerometer Test (you
can also run Gyro and board
temp)

Microphone Test



Go to Examples > PDM >
PDMSerialPlotter

ToF (VL53L1X) Library Installation



ToF (VL53L1X) Test

The screenshot shows the Arduino IDE 2.1.1 interface. The top bar displays "proximity_detection | Arduino IDE 2.1.1". The left sidebar shows the file structure with "proximity_detection.ino" selected. The main code editor window contains the following C++ code:

```
1 #include "VL53L1X.h"
2 VL53L1X proximity;
3
4 bool blinkState = false;
5 int reading = 0;
6 int timeStart = 0;
7 int blinkTime = 2000;
8
9 void setup() {
10     Serial.begin(115200);
11     Wire1.begin();
12     Wire1.setClock(400000); // use 400 kHz I2C
13     proximity.setBus(&Wire1);
14
15     pinMode(LED_BUILTIN, OUTPUT);
16     digitalWrite(LED_BUILTIN, blinkState);
17
18     if (!proximity.init()) {
19         Serial.println("Failed to detect and initialize sensor!");
20         while (1);
21     }
22
23     proximity.setDistanceMode(VL53L1X::Long);
24     proximity.setMeasurementTimingBudget(10000);
25     proximity.startContinuous(10);
26 }
27
28 void loop() {
29     reading = proximity.read();
30     Serial.println(reading);
31
32     if (millis() - timeStart >= reading) {
33         digitalWrite(LED_BUILTIN, blinkState);
34         timeStart = millis();
35
36         blinkState = !blinkState;
37     }
38 }
```

The bottom part of the IDE shows the "Serial Monitor" window. A red box highlights the message input field and the message list area. The message list shows three entries:

- 14:56:15.691 -> 1436
- 14:56:15.726 -> 1269
- 14:56:15.726 -> 1150

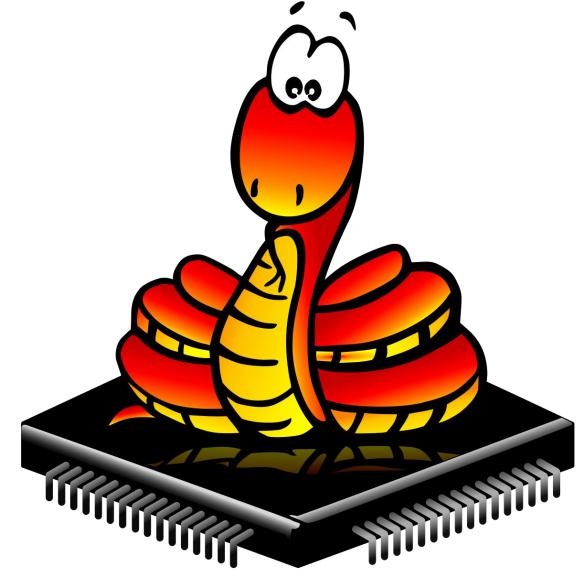
The status bar at the bottom indicates "indexing: 14/31".

Run the sketch:
proximity_detection.ino

<https://docs.arduino.cc/tutorials/nicla-vision/proximity>



openMV



IDE Installation & Sensors 'test

OpenMV IDE v4.0.1

OpenMV IDE is the premier integrated development environment for use with your OpenMV Cam. It features a powerful text editor, debug terminal, and frame buffer viewer w/ a histogram display. OpenMV IDE makes it easy to program your OpenMV Cam.

DOWNLOAD NOW
FOR WINDOWS 7, 8, 10, 11
OR LATER

DOWNLOAD NOW
FOR MACOS MONTEREY
OR LATER

DOWNLOAD NOW
FOR UBUNTU 20.04 LTS 64-BIT
OR LATER

DOWNLOAD NOW
FOR RASPBERRY PI OS BULLSEYE 64-BIT
OR LATER

OpenMV IDE v4.0.1 - Release Notes

Before connecting the Nicla to the OpenMV IDE, ensure you have the latest bootloader version.

To that, go to your Arduino ID, select the Nicla board and open the sketch on:

Examples >
STM_32H747_System >
STM_32H747_updateBootloader



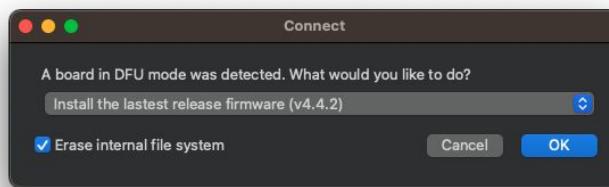
The screenshot shows the OpenMV IDE interface with several key components highlighted by orange boxes:

- Code Area**: The main workspace where Python code is written. It contains the "helloworld_1.py" script. A yellow arrow points from the text "Dataset Editor ->" to the icon for the Dataset Editor in the Tool Bar.
- Serial Terminal**: A terminal window at the bottom left for serial communication.
- Tool Bar**: A vertical bar on the left containing icons for various tools, with the Dataset Editor icon highlighted by a yellow box.
- Frame Buffer**: A preview window on the right labeled "No image". Below it is a "Histogram" and "Grayscale Color Space" viewer showing a grid pattern.

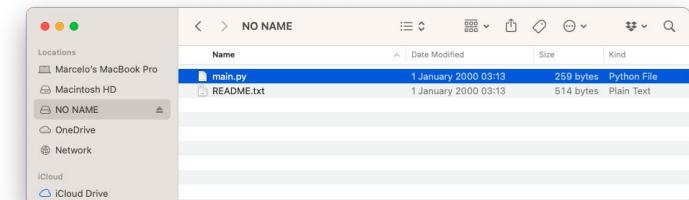
```
helloworld_1.py
1 # Hello World Example
2
3 # Welcome to the OpenMV IDE! Click on the green run arrow button below to run the script!
4
5 import sensor, image, time
6
7 sensor.reset()                      # Reset and initialize the sensor.
8 sensor.set_pixformat(sensor.RGB565)   # Set pixel format to RGB565 (or GRayscale)
9 sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
10 sensor.skip_frames(time = 2000)        # Wait for settings take effect.
11 clock = time.clock()                 # Create a clock object to track the FPS.
12
13 while(True):
14     clock.tick()                     # Update the FPS clock.
15     img = sensor.snapshot()          # Take a picture and return the image.
16     print(clock.fps())              # Note: OpenMV Cam runs about half as fast when connected
17                                     # to the IDE. The FPS should increase once disconnected.
```

Nicla connection to OpenMV IDE

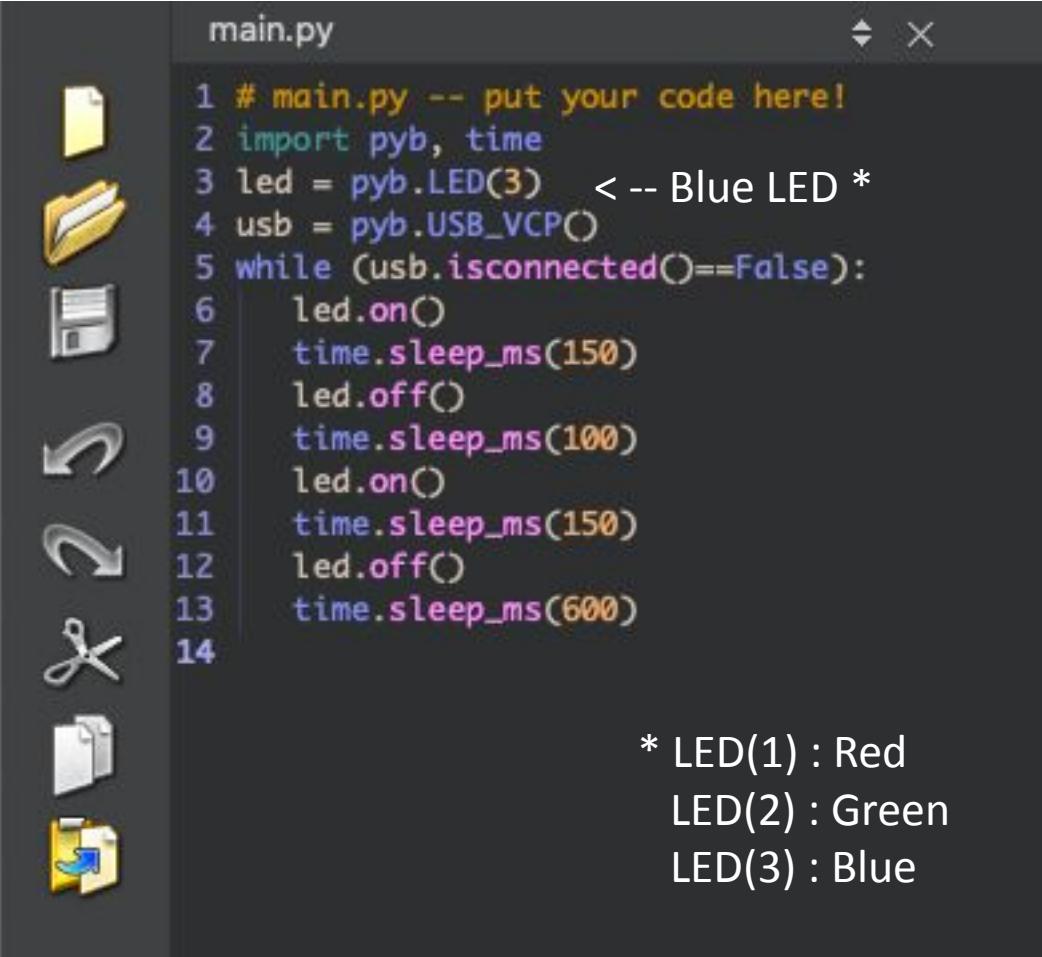
The first time that the Nicla is connected to the IDE, you need to install a FW



A drive named “NO NAME” will appear in your computer



The Blink MicroPython code



The screenshot shows a code editor window titled "main.py". The code is a MicroPython script for a board with a blue LED. It imports the pyb and time modules, initializes a blue LED at pin 3, and enters a loop that alternates the LED between on and off states every 150ms. The code is color-coded: comments are yellow, keywords are cyan, and variables are purple. Below the code, there is a note explaining the LED mapping: LED(1) is Red, LED(2) is Green, and LED(3) is Blue.

```
1 # main.py -- put your code here!
2 import pyb, time
3 led = pyb.LED(3)    <-- Blue LED *
4 usb = pyb.USB_VCP()
5 while (usb.isconnected() == False):
6     led.on()
7     time.sleep_ms(150)
8     led.off()
9     time.sleep_ms(100)
10    led.on()
11    time.sleep_ms(150)
12    led.off()
13    time.sleep_ms(600)
14

* LED(1) : Red
  LED(2) : Green
  LED(3) : Blue
```

class LED – LED object

The LED object controls an individual LED (Light Emitting Diode).

Constructors

`class pyb.LED(id)`

Create an LED object associated with the given LED:

- `id` is the LED number, 1-4.

Methods

`LED.off()`

Turn the LED off.

`LED.on()`

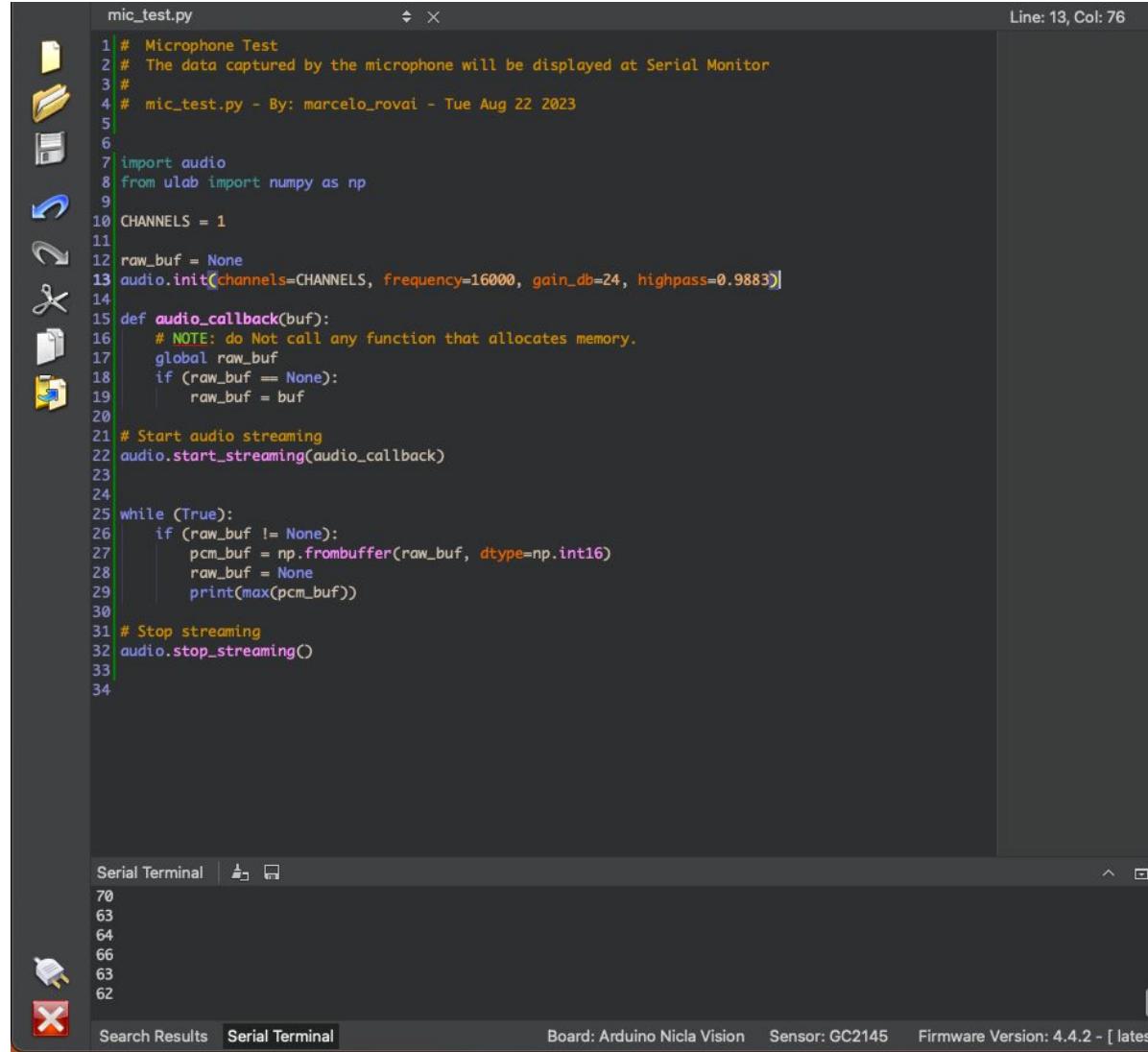
Turn the LED on, to maximum intensity.

`LED.toggle()`

Toggle the LED between on (maximum intensity) and off. If the LED is at non-zero intensity then it is considered “on” and toggle will turn it off.

On **RESET**, the board will always run the script **main.py**

Microphone test



The screenshot shows a code editor window titled "mic_test.py". The code is a Python script for a microphone test. It imports the "audio" module from "ulab" and initializes an audio stream with 1 channel at 16000 frequency, 24 gain_db, and 0.9883 highpass. It defines an audio callback function that converts raw buffer to PCM buffer and prints the maximum value. The script starts streaming, enters a loop to read raw buffers, converts them to PCM, and prints the max value, then stops streaming.

```
# Microphone Test
# The data captured by the microphone will be displayed at Serial Monitor
#
# mic_test.py - By: marcelo_movai - Tue Aug 22 2023

import audio
from ulab import numpy as np

CHANNELS = 1

raw_buf = None
audio.init(channels=CHANNELS, frequency=16000, gain_db=24, highpass=0.9883)

def audio_callback(buf):
    # NOTE: do Not call any function that allocates memory.
    global raw_buf
    if (raw_buf == None):
        raw_buf = buf

# Start audio streaming
audio.start_streaming(audio_callback)

while (True):
    if (raw_buf != None):
        pcm_buf = np.frombuffer(raw_buf, dtype=np.int16)
        raw_buf = None
        print(max(pcm_buf))

# Stop streaming
audio.stop_streaming()
```

The "Serial Terminal" tab at the bottom shows the output of the script, which consists of a series of numerical values: 70, 63, 64, 66, 63, 62.

At the very bottom, there are tabs for "Search Results" and "Serial Terminal", and status information: "Board: Arduino Nicla Vision", "Sensor: GC2145", and "Firmware Version: 4.4.2 ~ [latest]".

Third-party libraries

The following third-party libraries are built-in to your OpenMV Cam's firmware:

ulab – numpy-like array manipulation library

pid – Proportional/Integral/Derivative Control

bno055 – IMU Driver

Examples scripts are located in OpenMV IDE under the **IMU Shield** examples folder.

ssd1306 – OLED Driver

tb6612 – Stepper Motor Driver

Examples scripts are located in OpenMV IDE under the **Motor Shield** examples folder.

lsm6dsox – lsm6dsox Driver

modbus – modbus protocol library

Examples scripts are located in OpenMV IDE under the **Modbus** examples folder.

mqtt – mqtt protocol library

Examples scripts are located in OpenMV IDE under the **WiFi Shield** examples folder.

vl53l1x – ToF Distance Sensor Driver

Examples scripts are located in OpenMV IDE under the **Distance Shield** examples folder.

```
lsm6dsox_basic_1.py
1 # LSM6DSOX Gyro example.
2 import time
3 from lsm6dsox import LSM6DSOX
4 from machine import I2C, SPI, Pin
5
6 lsm = LSM6DSOX(SPI(5), cs.pin=Pin("PF6", Pin.OUT_PP, Pin.PULL_UP))
7
8 while (True):
9     print('Accelerometer: x:{:>8.3f} y:{:>8.3f} z:{:>8.3f}'.format(*lsm.read_accel()))
10    print('Gyroscope: x:{:>8.3f} y:{:>8.3f} z:{:>8.3f}'.format(*lsm.read_gyro()))
11    print('')
12    time.sleep_ms(100)
13 |
```

Serial Terminal | Gyroscope: x: -0.061 y: -0.244 z: -0.061
Accelerometer: x: 0.158 y: -0.641 z: -0.766
Gyroscope: x: -0.122 y: -0.244 z: -0.061
Accelerometer: x: 0.158 y: -0.641 z: -0.764
Gyroscope: x: -0.183 y: -0.244 z: -0.122
Accelerometer: x: 0.158 y: -0.640 z: -0.765
Gyroscope: x: -0.244 y: -0.305 z: -0.061

Search Results Serial Terminal Board: Arduino Nicla Vision Sensor: CC2145 Firmware Version: 4.4.2 - [latest]

```
distance_image_meter.py
1 # Hello World Example + Flip/Mirror camera
2 # ToF distance meter
3 # By: marcelo_rovai - Mon Aug 21 202
4
5 import sensor, image, time
6 from machine import I2C
7 from vl53l1x import VL53L1X
8
9 i2c = I2C(2)
10 distance = VL53L1X(i2c)
11
12 sensor.reset() # Reset and initialize the sensor.
13 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRayscale)
14 sensor.set_framesize(sensor.QVGA) # Set frame size to QVGA (320x240)
15 sensor.skip_frames(time = 2000) # Wait for settings take effect.
16 clock = time.clock() # Create a clock object to track the FPS.
17 #sensor.set_vflip(True) # Flips the image vertically
18 #sensor.set_hmirror(True) # Mirrors the image horizontally
19
20 while(True):
    range: mm 251
    range: mm 252
    range: mm 252
    range: mm 257
    range: mm 257
    range: mm 256
    range: mm 256
    range: mm 257
    range: mm 257
    range: mm 256
    range: mm 256
    range: mm 255
```

Search Results Serial Terminal Board: Arduino Nicla Vision Sensor: GC2145 Firmware Version: 4.4.2 -

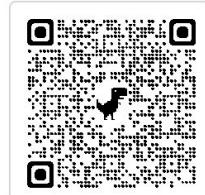
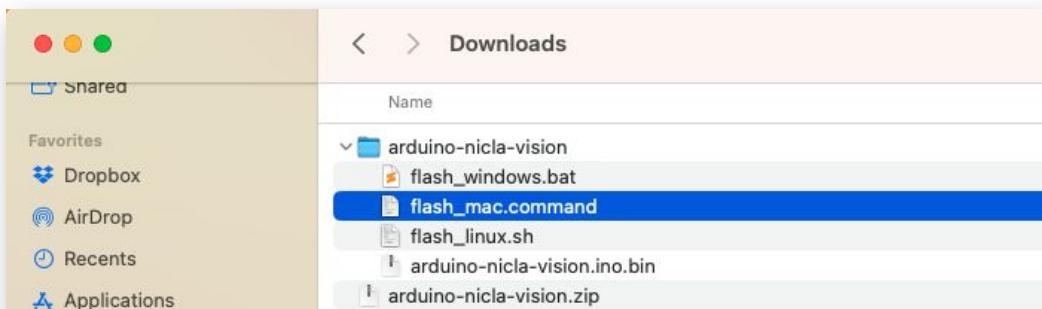
Nicla connection & Sensors' capture data



**EDGE
IMPULSE**

Connecting with EI

1. Put the Nicla-Vision on Boot mode (press the button twice)
2. Download the FW from the Edge Impulse Site
3. Open the zip file on your computer and select the uploader related to your OS



```
marcelo_rovai -- flash_mac.command -- 129x52

last login: Mon Aug 21 11:24:00 on ttys000
/Users/marcelo_rovai/Downloads/arduino-nicla-vision/flash_mac.command ; exit;
(base) marcelo_rovai@Marcelos-MacBook-Pro ~ % /Users/marcelo_rovai/Downloads/arduino-nicla-vision/flash_mac.command ; exit;
You're using an untested version of Arduino CLI, this might cause issues (found: 0.31.0, expected: 0.18.x)
Finding Arduino Mbed core...
Finding Arduino Mbed OK
Finding Arduino Nicla Vision...
Finding Arduino Nicla Vision OK
Flashing board...
dfu-util 0.10-dev

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2021 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Warning: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release
Opening DFU capable USB device...
Device ID 2341:035f
Device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Interface #0 ...
Determining device status...
DFU state(2) = dfuIDLE, status(0) = No error condition is present
DFU mode device DFU version 011a
Device returned transfer size 4096
DfuSe interface name: "Internal Flash"
Downloading element to address = 0x08040000, size = 557336
Erase [=====] 100% 557336 bytes
Erase done.
Download [=====] 100% 557336 bytes
Download done.
File downloaded successfully
Transitioning to dfuMANIFEST state

A new release of Arduino CLI is available: 0.31.0 → 0.34.0
https://arduino.github.io/arduino-cli/latest/installation/#latest-packages

Flashed your Arduino Nicla Vision development board.
To set up your development with Edge Impulse, run 'edge-impulse-daemon'
To run your impulse on your development board, run 'edge-impulse-run-impulse'

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Process completed]
```

studio.edgeimpulse.com wants to connect to a serial port

MJRoBot (Marcelo Rovai) / NICLA-Vision_Classification

EDGE II

Dashboard Devices Data acquisition Impulse design EON Tuner Retrain model Live classification Model testing Versioning Deployment

GETTING STARTED Documentation Forums

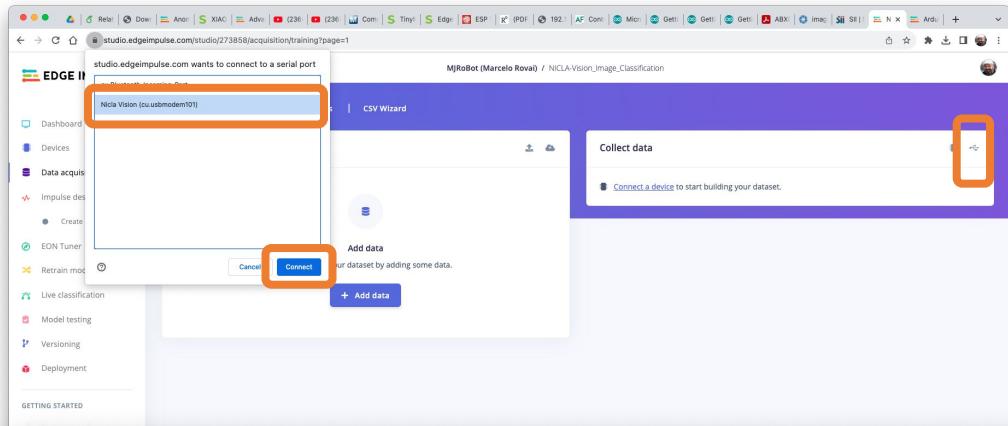
Nica Vision (cu.usbmodem101)

Connect

Add data + Add data

Collect data

Connect a device to start building your dataset.



studio.edgeimpulse.com/studio/273858/acquisition/training?page=1

NICLA-Vision_Classification

EDGE IMPULSE

Dataset Data explorer Data sources CSV Wizard

DATA COLLECTED 20s TRAIN / TEST SPLIT 100% / 0% ⚠

Dataset

Training (2) Test (0)

SAMPLE NAME	LABEL	ADDED	LENGTH
mic.48bnfgtp	mic	Today, 12:56:53	10s
imu.48bn95k	imu	Today, 12:53:28	10s

Collect data

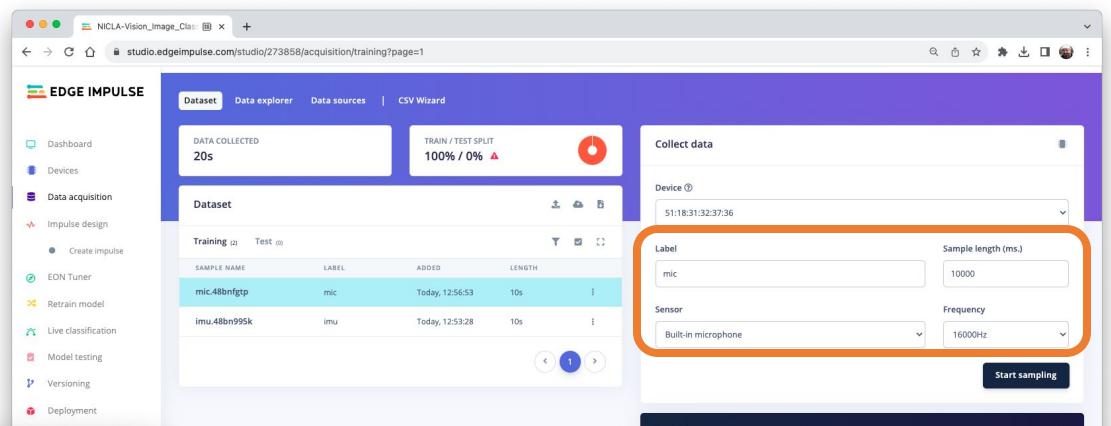
Device (2) 51:18:31:32:37:36

Label mic Sample length (ms.) 10000

Sensor Built-in microphone Frequency 16000Hz

Start sampling

RAW DATA mic.48bnfgtp



studio.edgeimpulse.com/studio/273858/acquisition/training?page=1

MJRoBot (Marcelo Rovai) / NICLA-Vision_Classification

EDGE IMPULSE

Dataset Data explorer Data sources CSV Wizard

DATA COLLECTED 20s TRAIN / TEST SPLIT 100% / 0% ⚠

Dataset

Training (2) Test (0)

SAMPLE NAME	LABEL	ADDED	LENGTH
img.48bnidib	img	Today, 12:58:27	-
mic.48bnfgtp	mic	Today, 12:56:53	10s
imu.48bn95k	imu	Today, 12:53:28	10s

Collect data

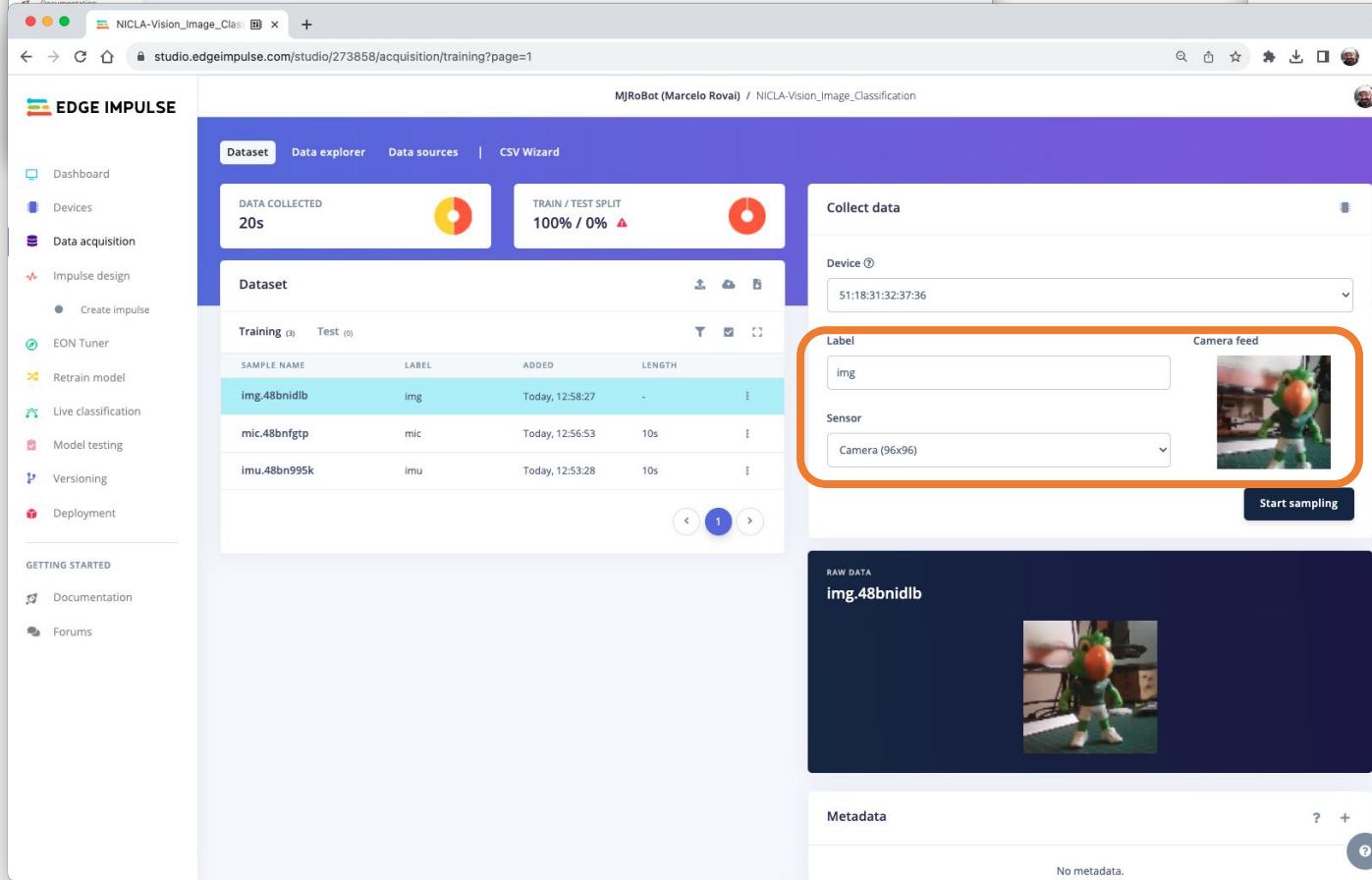
Device (2) 51:18:31:32:37:36

Label img Camera feed

Sensor Camera (96x96)

Start sampling

RAW DATA img.48bnidib



studio.edgeimpulse.com/studio/273858/acquisition/training?page=1

NICLA-Vision_Classification

EDGE IMPULSE

Dataset Data explorer Data sources CSV Wizard

DATA COLLECTED 30s TRAIN / TEST SPLIT 100% / 0% ⚠

Dataset

Training (2) Test (0)

SAMPLE NAME	LABEL	ADDED	LENGTH
imu.48bn95k	imu	Today, 12:53:28	10s
mic.48bn6c6s	mic	Today, 12:51:53	10s
mic.48bn4c9h	mic	Today, 12:50:47	10s

Collect data

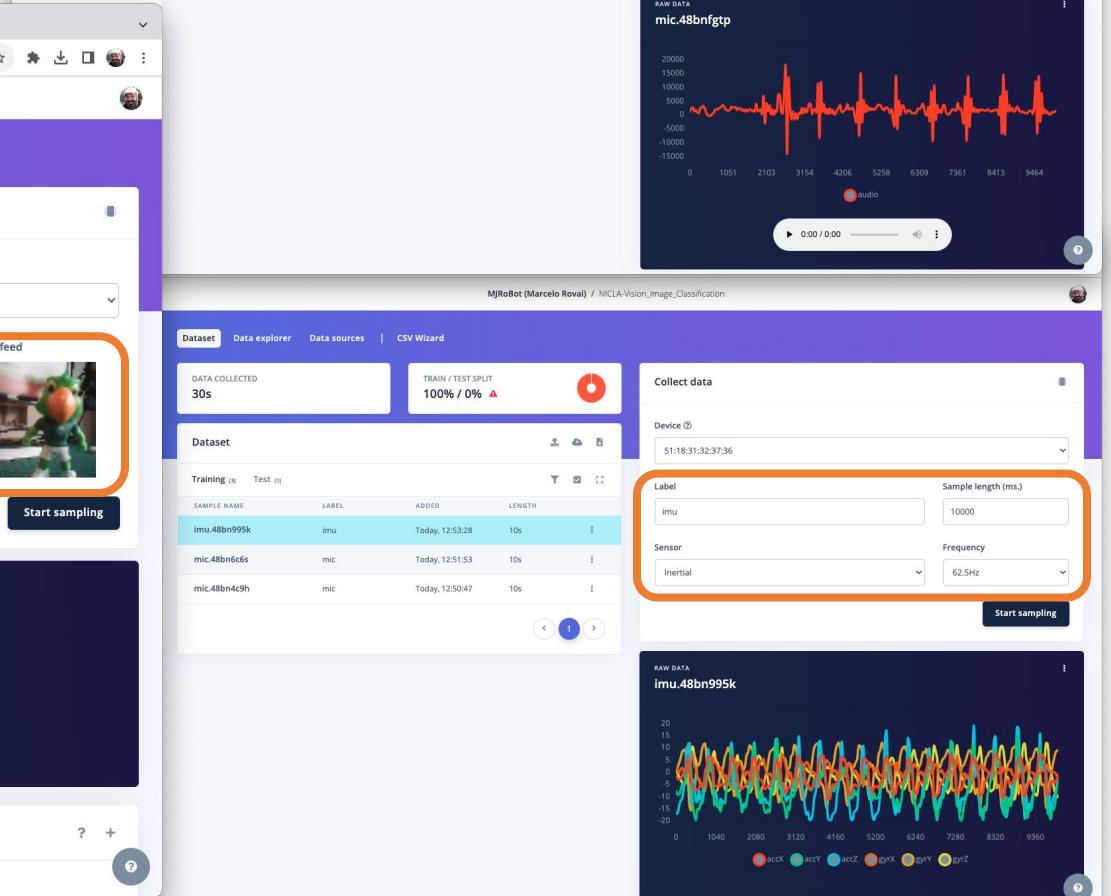
Device (2) 51:18:31:32:37:36

Label imu Sample length (ms.) 10000

Sensor Inertial Frequency 62.5Hz

Start sampling

RAW DATA imu.48bn95k



TinyML in Computer Vision, Introduction

Computer Vision Main Types

Image Classification (Multi-Class Classification)

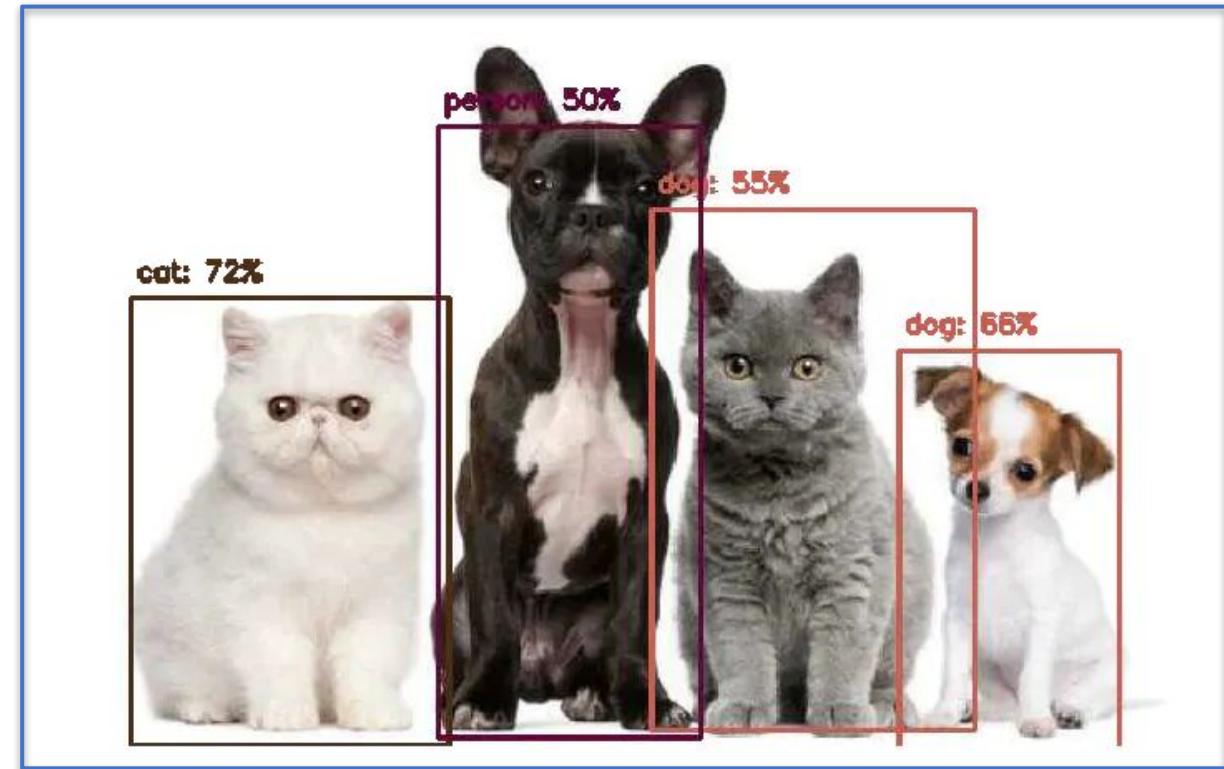


Cat: 70%



Dog: 80%

Object Detection Multi-Label Classification + Object Localization



Computer Vision Main Types

Image Classification (Multi-Class Classification)

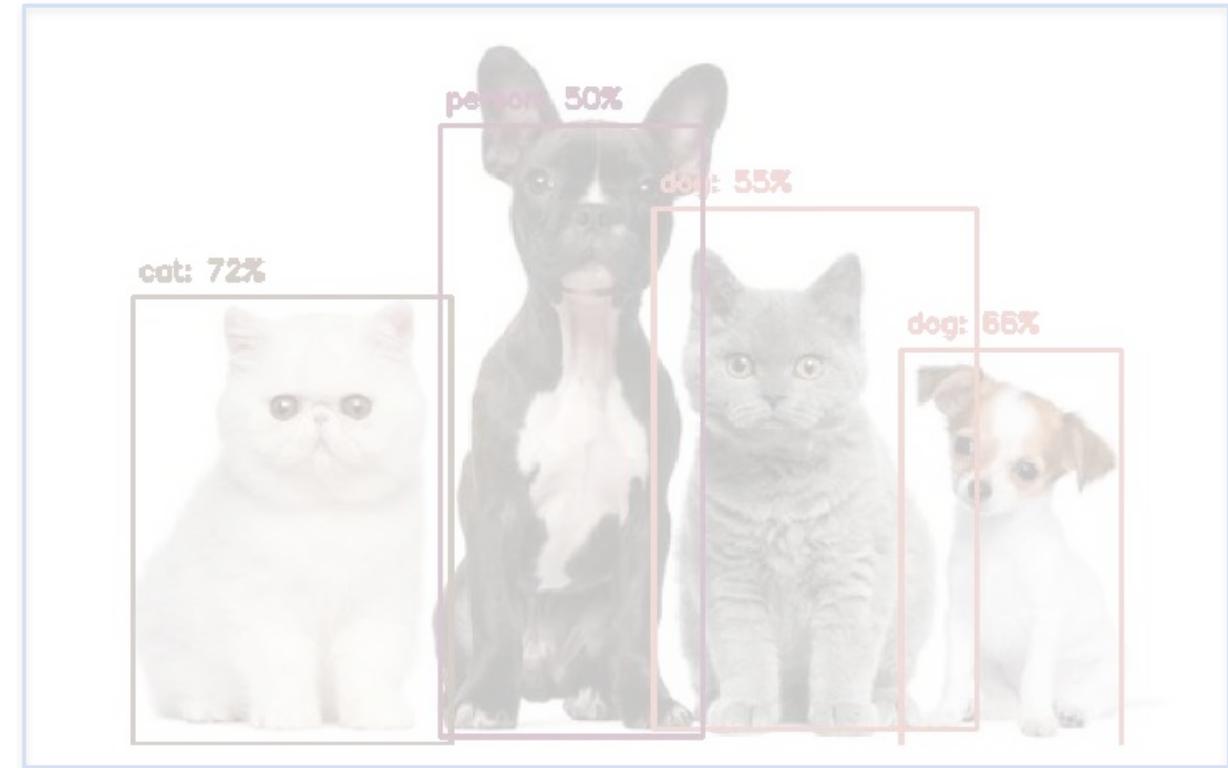


Cat: 70%



Dog: 80%

Object Detection Multi-Label Classification + Object Localization

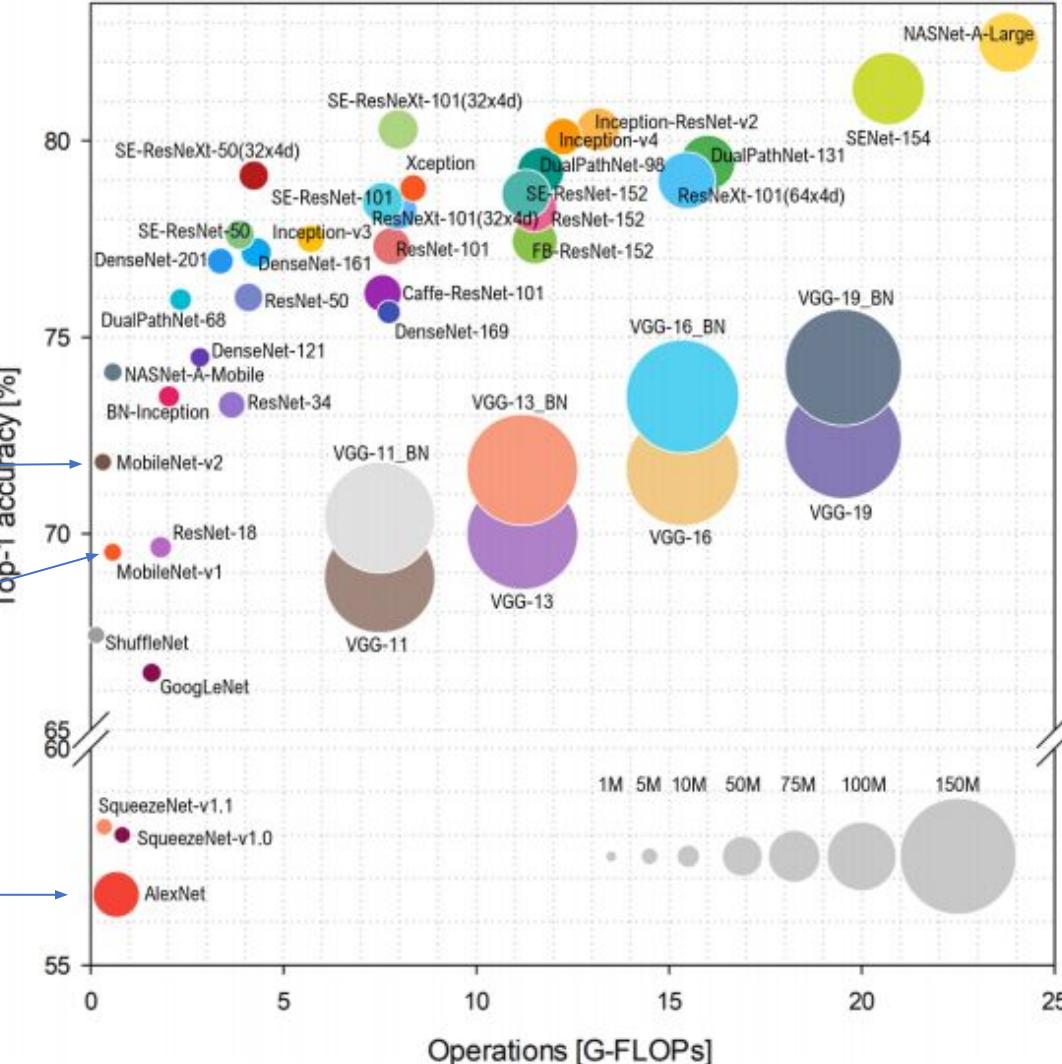


Computer Vision Models evolution

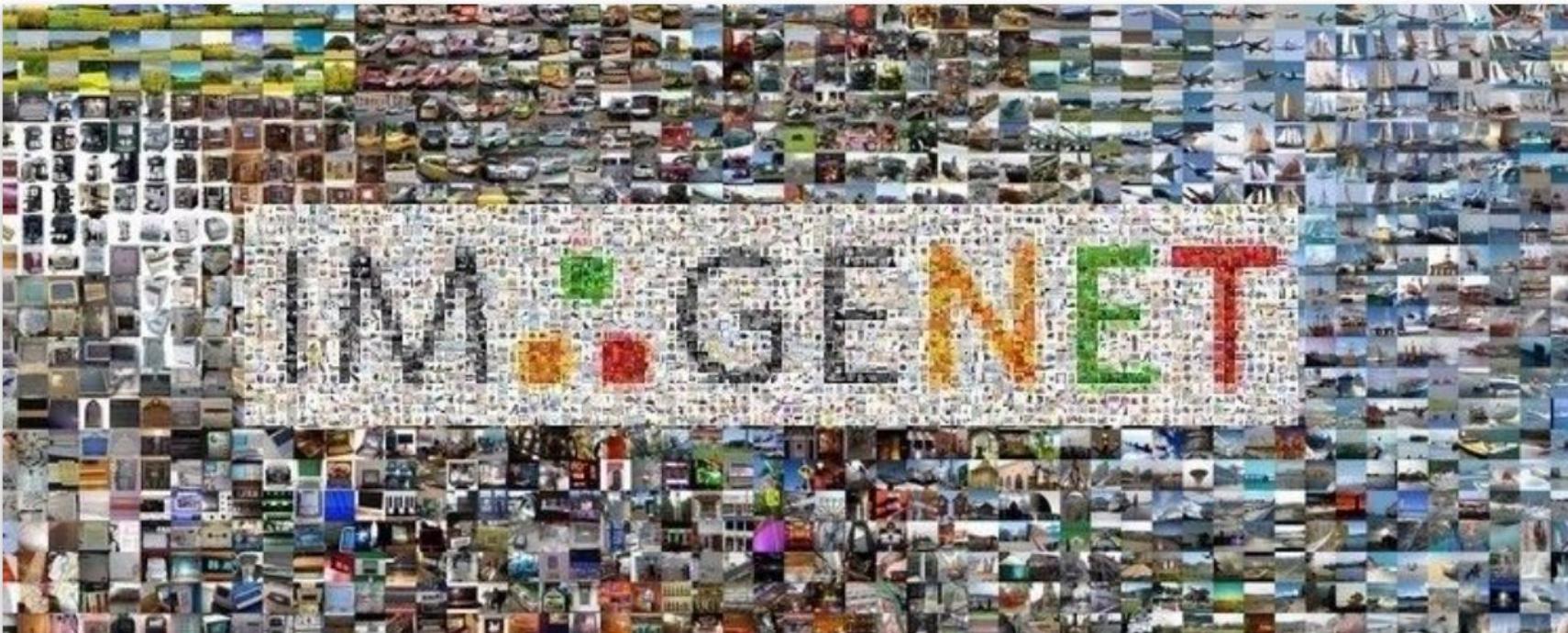
(2018)

(2017)

(2012)



Training Pipeline: Need Lots of Data

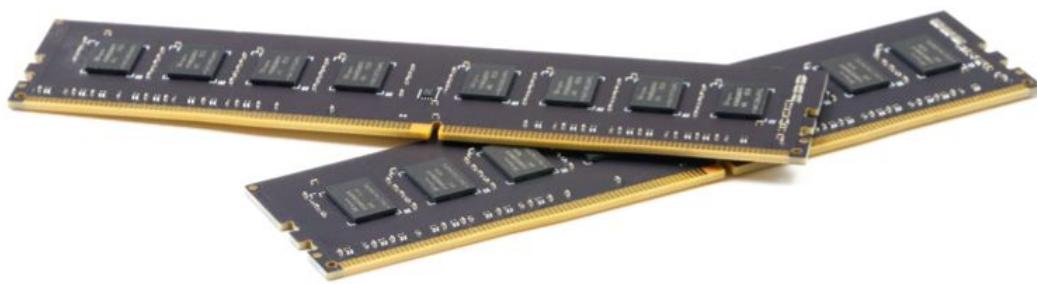


1000 Classes

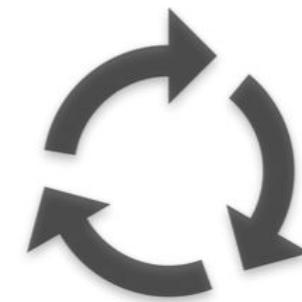
1000 Images / Class

*Total: 1,281,167 training images, 50,000 validation images, and 100,000 test images

Training Pipeline: Need Compute Resources



Memory

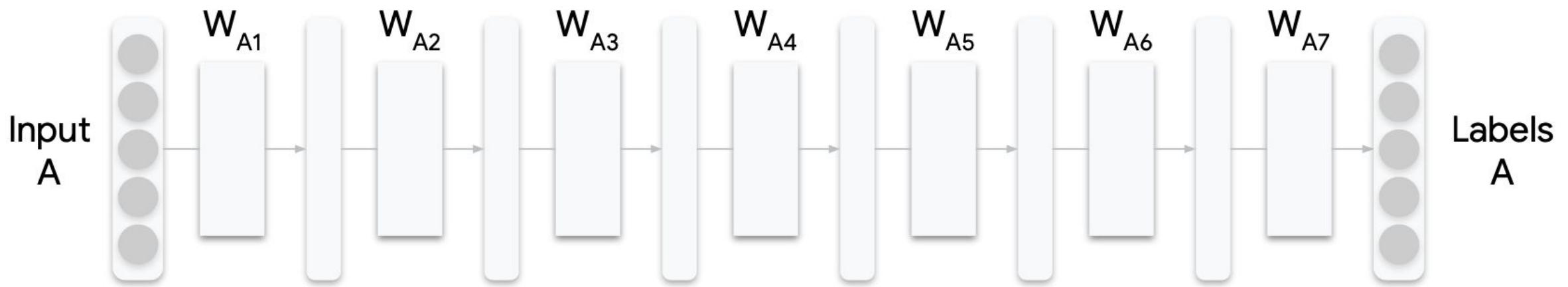


Compute

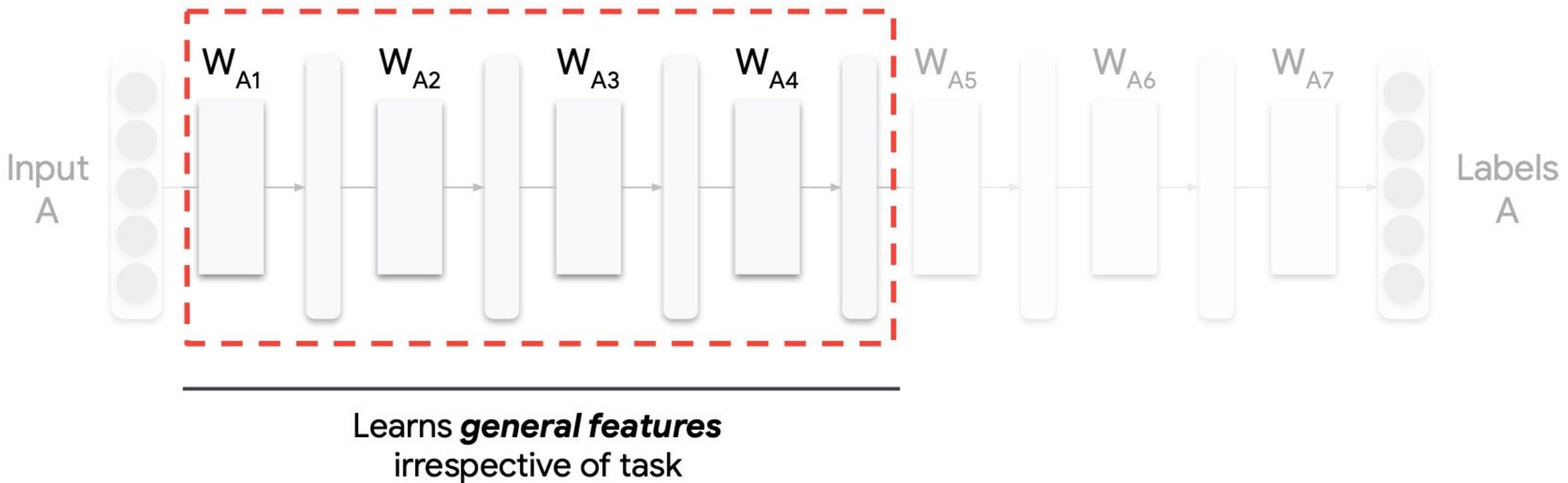


GPU and
Accelerators

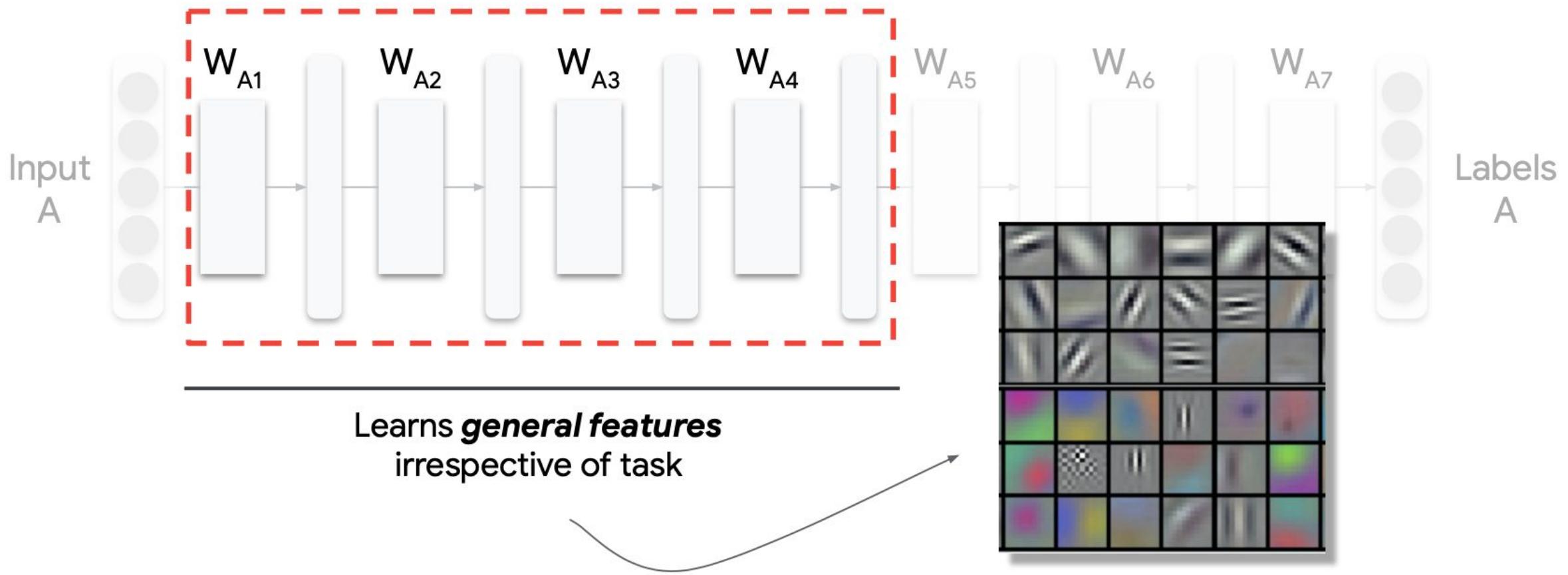
End Result of Training



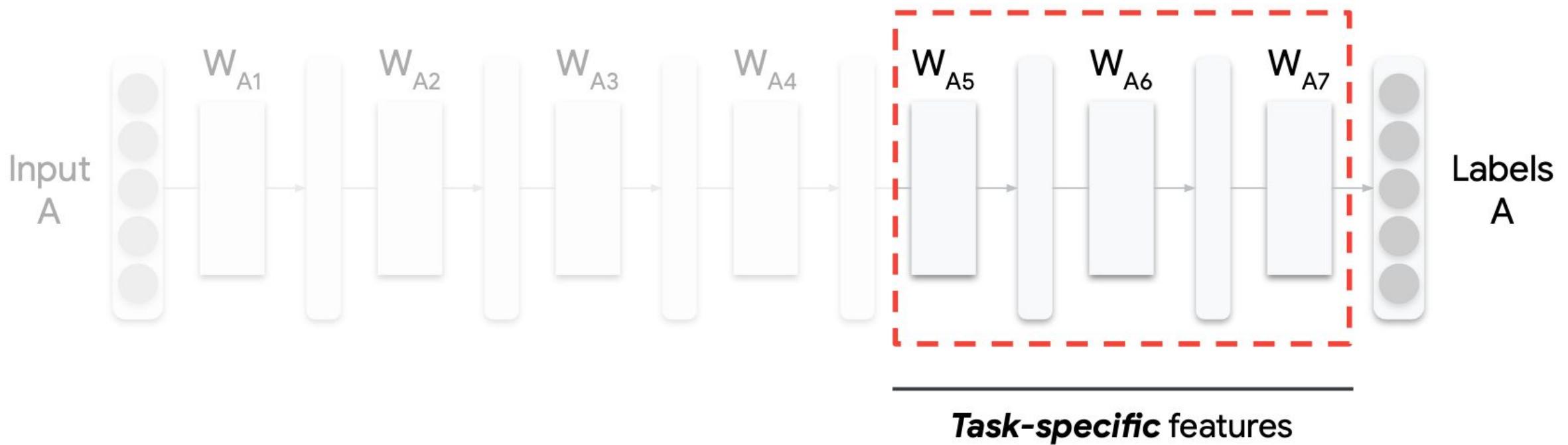
End Result of Training



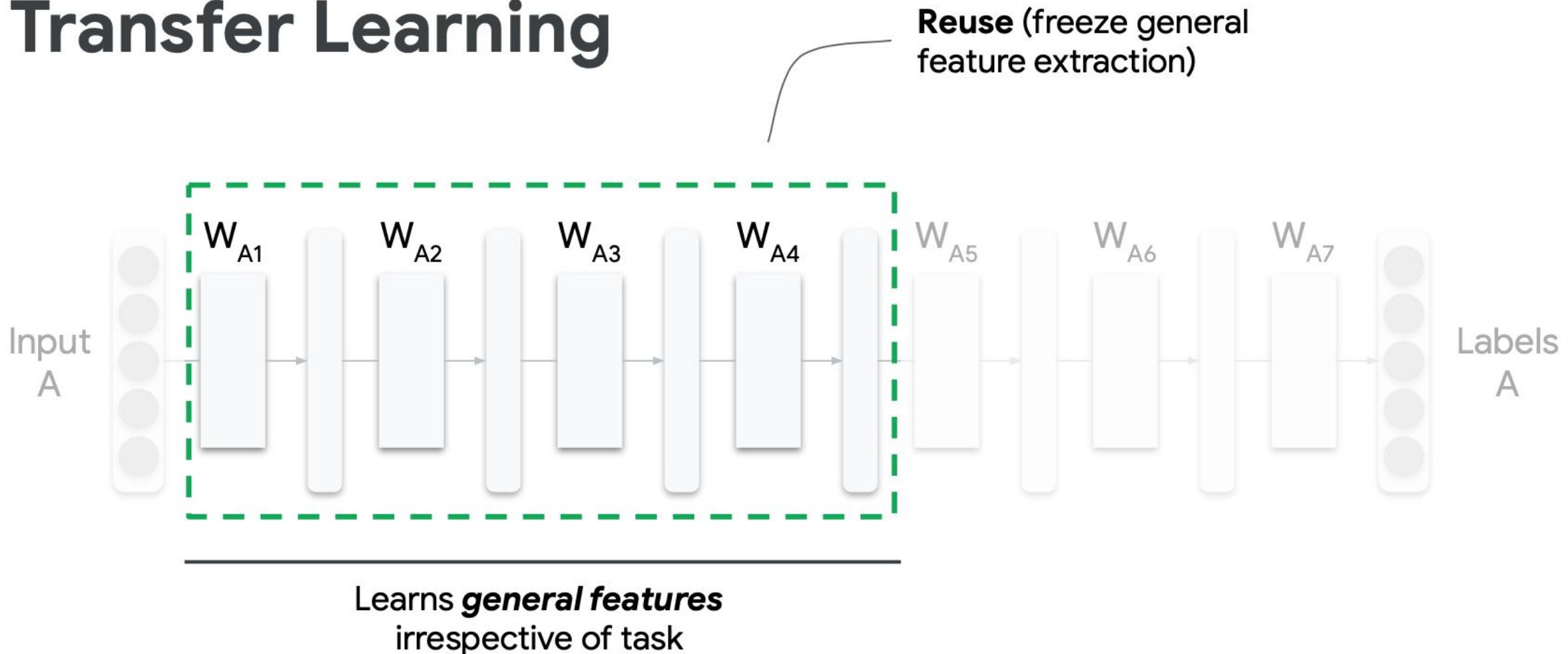
End Result of Training



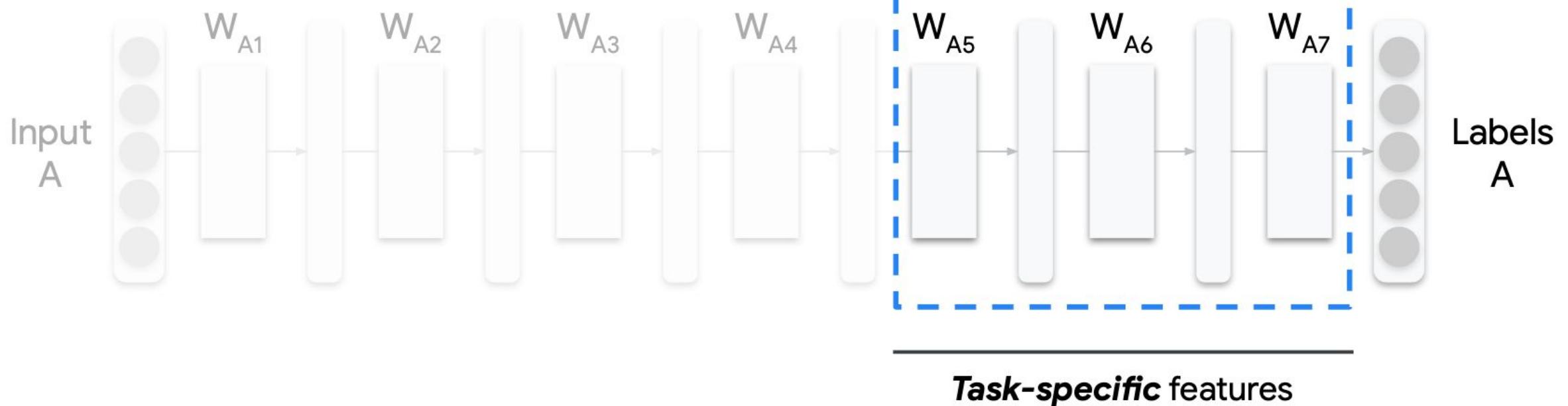
End Result of Training



Transfer Learning



Transfer Learning



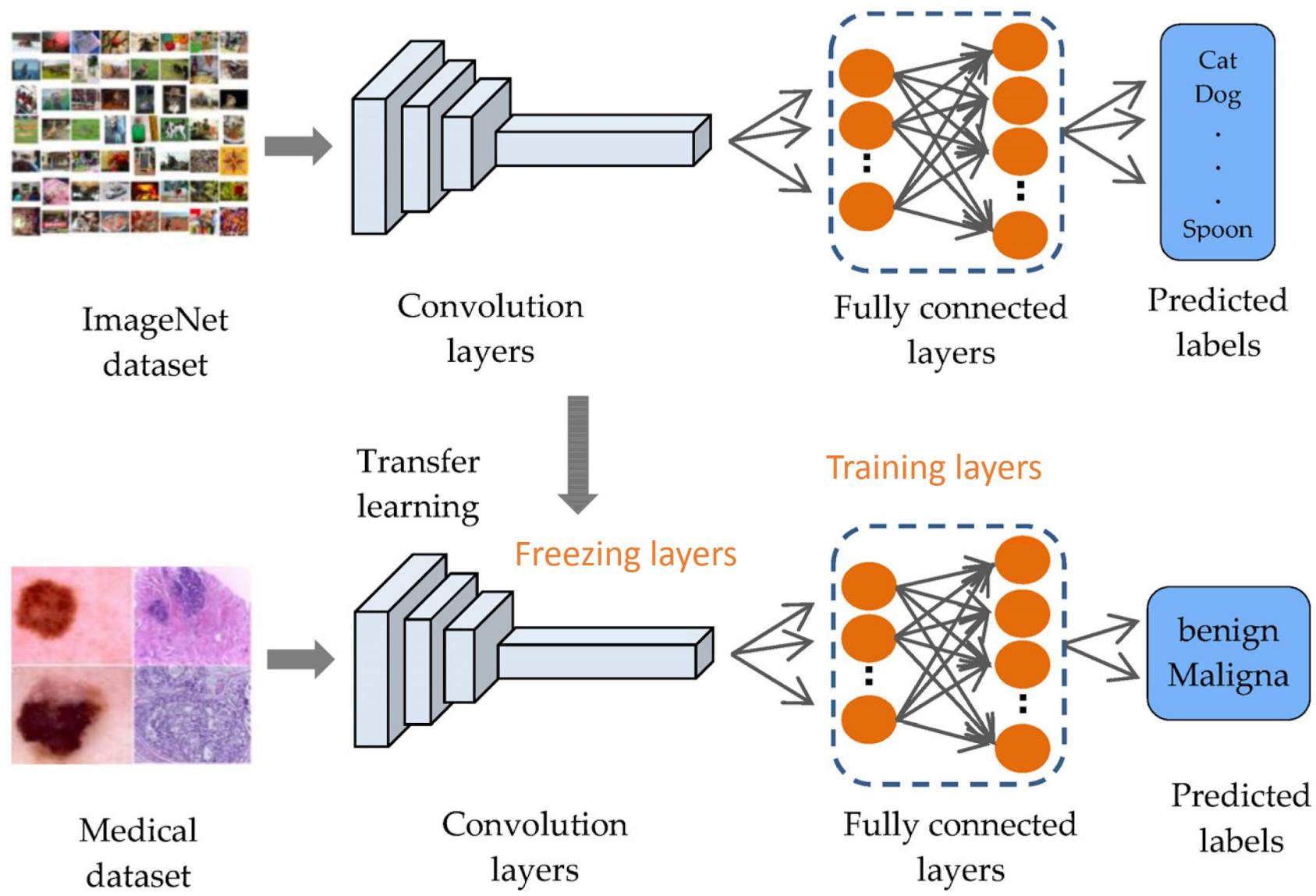


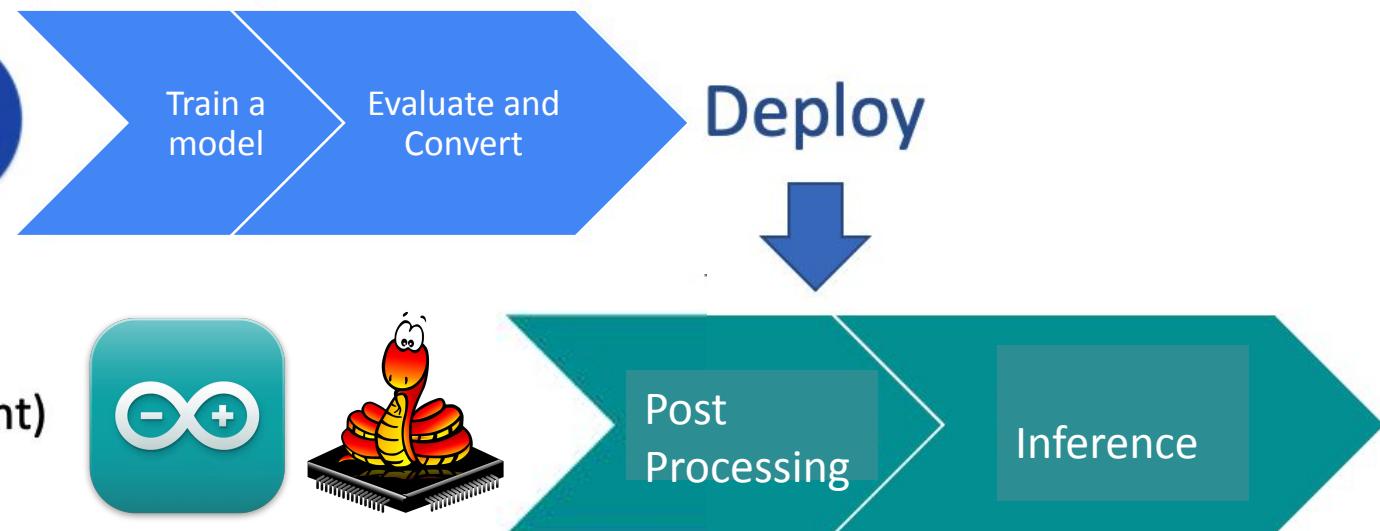
Image Classification Application: Design, Train, Test and Deploy





* Feature Extraction

On-Premises / Cloud (Server)



Package Inspection



Deep Learning at the Edge Simplifies Package Inspection

Image Classification Project 1

Decide a Goal

- Possible Images:
 - medicine
 - background

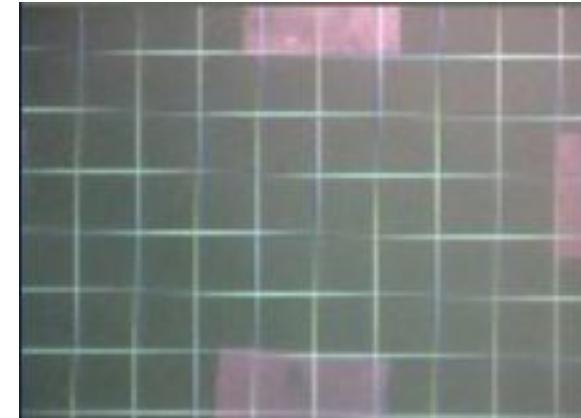


Image Classification Project 2

Decide a Goal

- Classes:
 - background
 - periquito
 - robot



Profile - Projects - Edge Impulse | NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/profile/projects

EDGE IMPULSE

Projects Custom ML blocks

MJRoBot (Marcelo Rovai)

Organizations EIE

Projects

+ Create new project

Create a new project

Enter the name for your new project: NICLA Vision - Image Classification

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

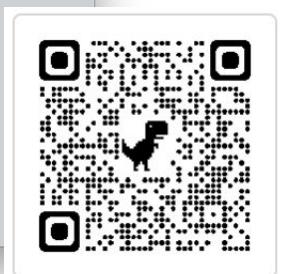
MJRoBot (Marcelo Rovai) / Motion-Project PUBLIC

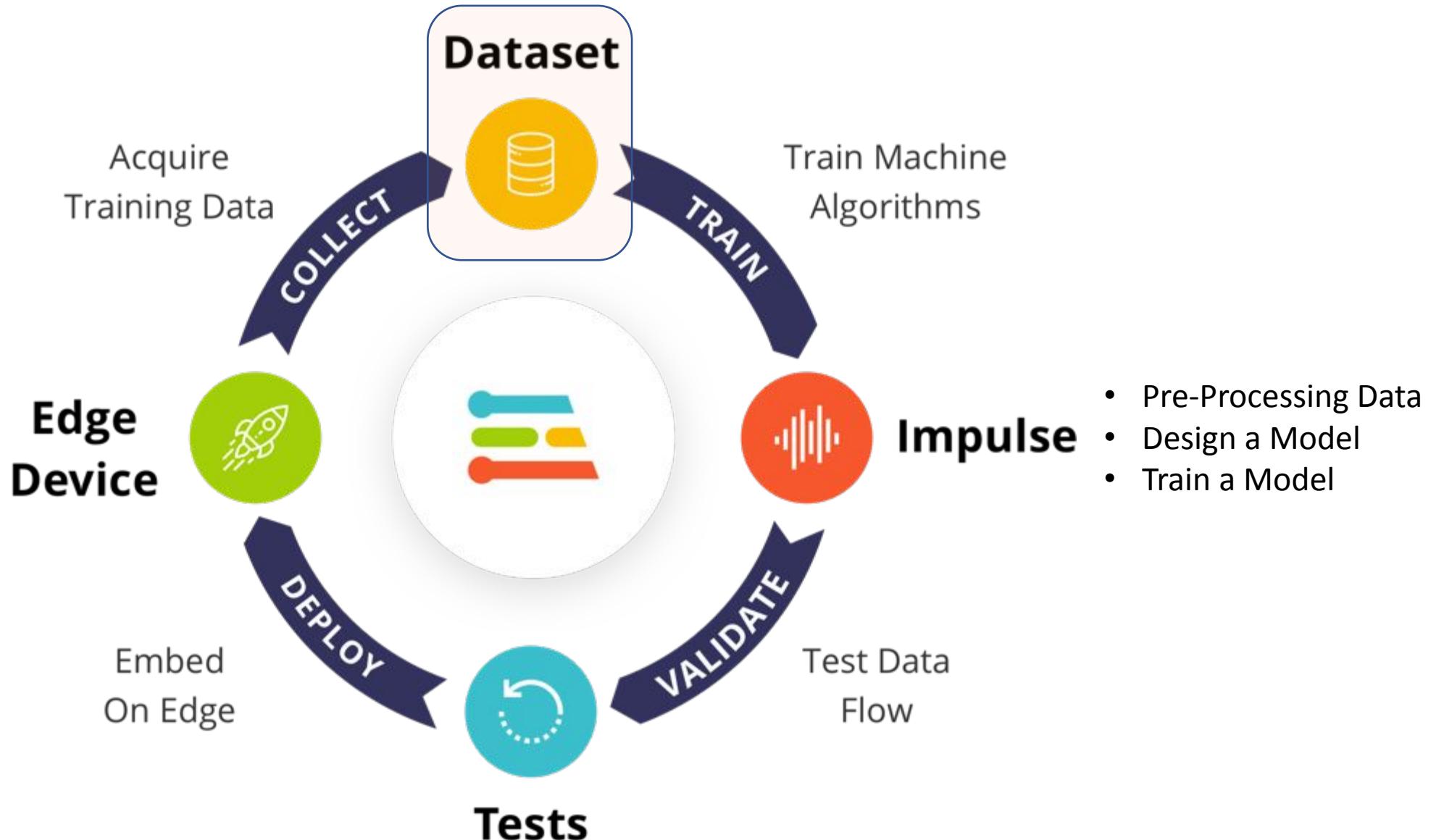
MJRoBot (Marcelo Rovai) / video_tinyml_raw

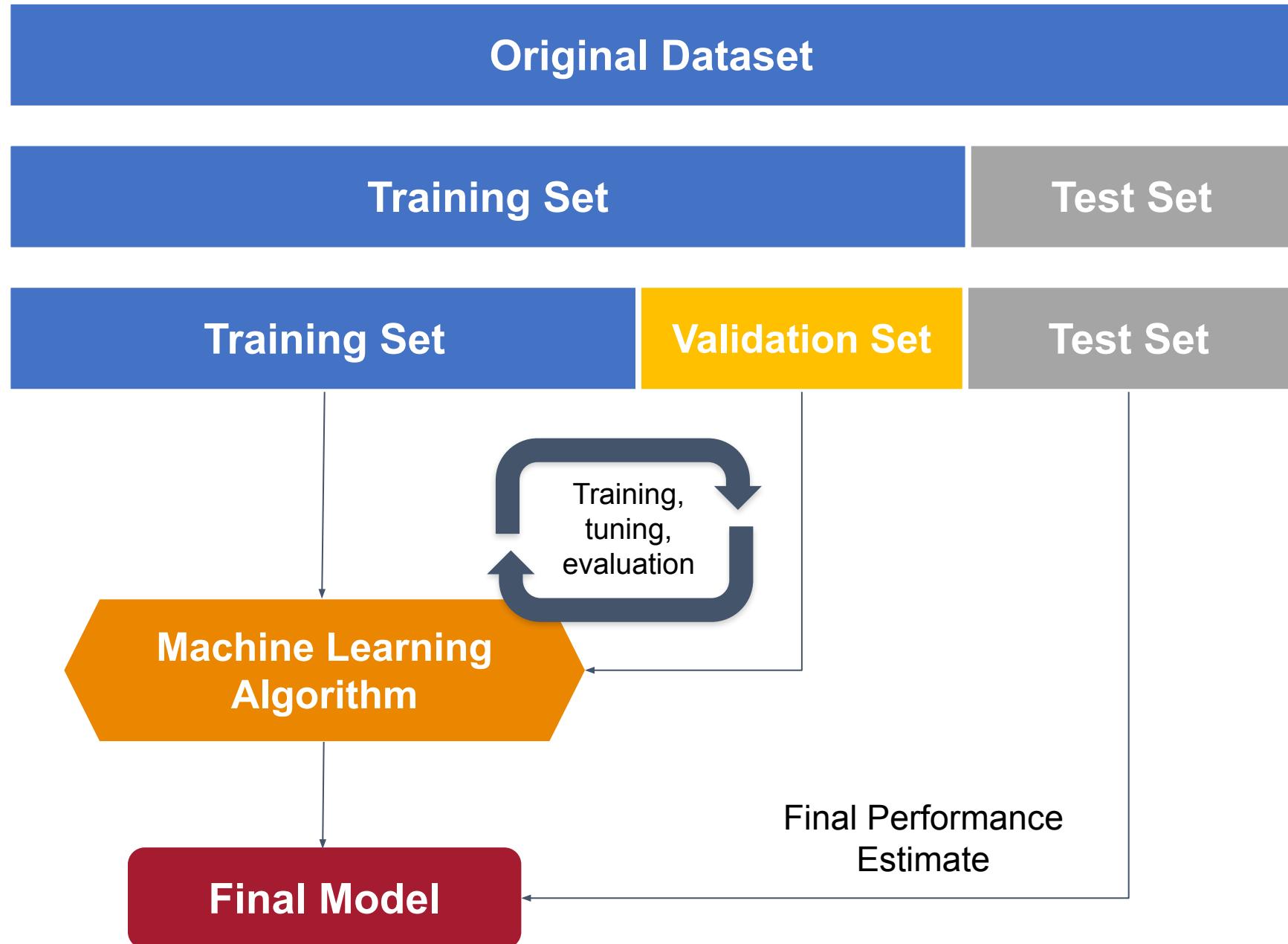
MJRoBot (Marcelo Rovai) / Pico_Motion_Detection PUBLIC

MJRoBot (Marcelo Rovai) / oi_rovis_kws_meetup

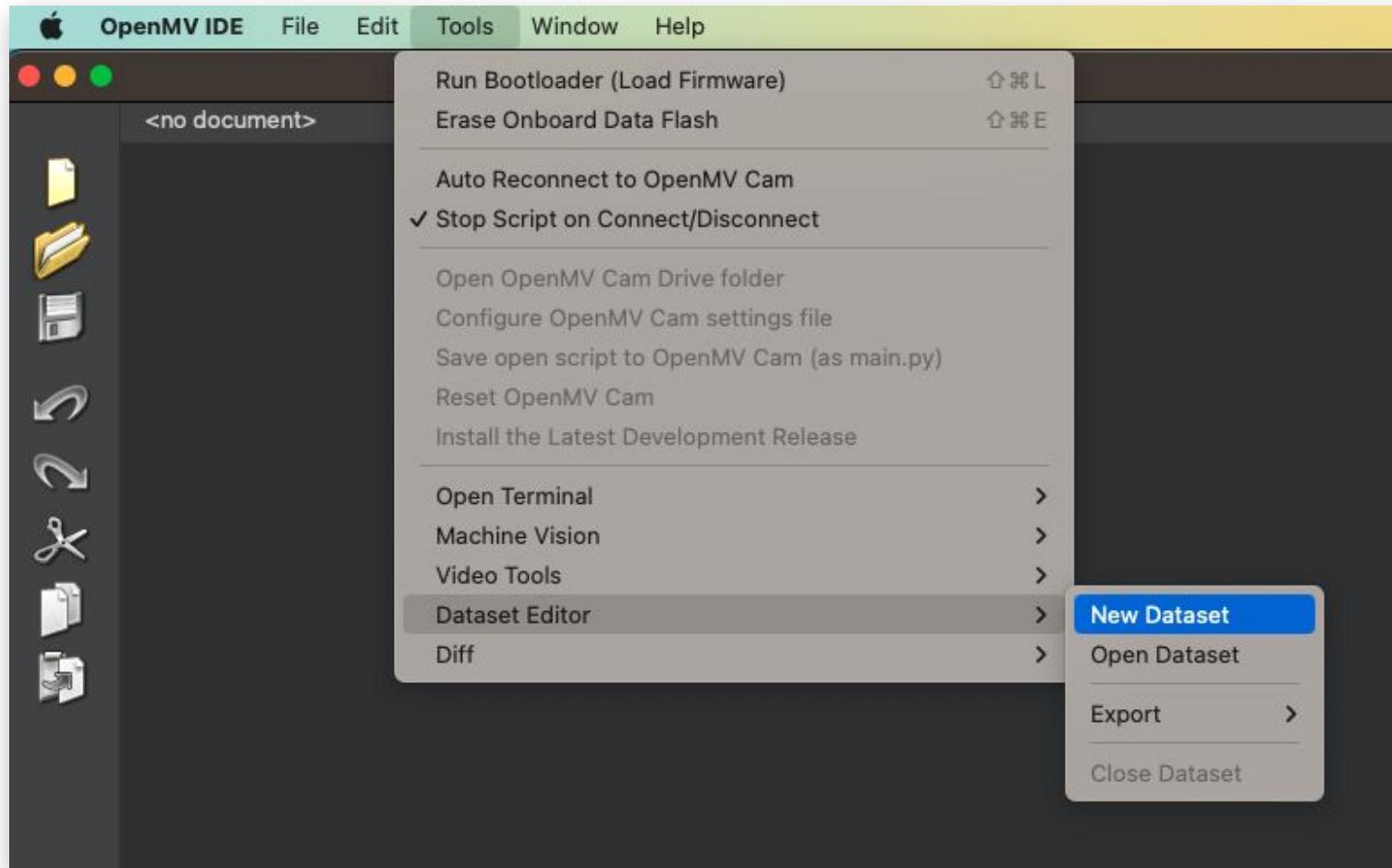
MJRoBot (Marcelo Rovai) / ECG_Unifei PUBLIC







Collecting Dataset with OpenMV IDE

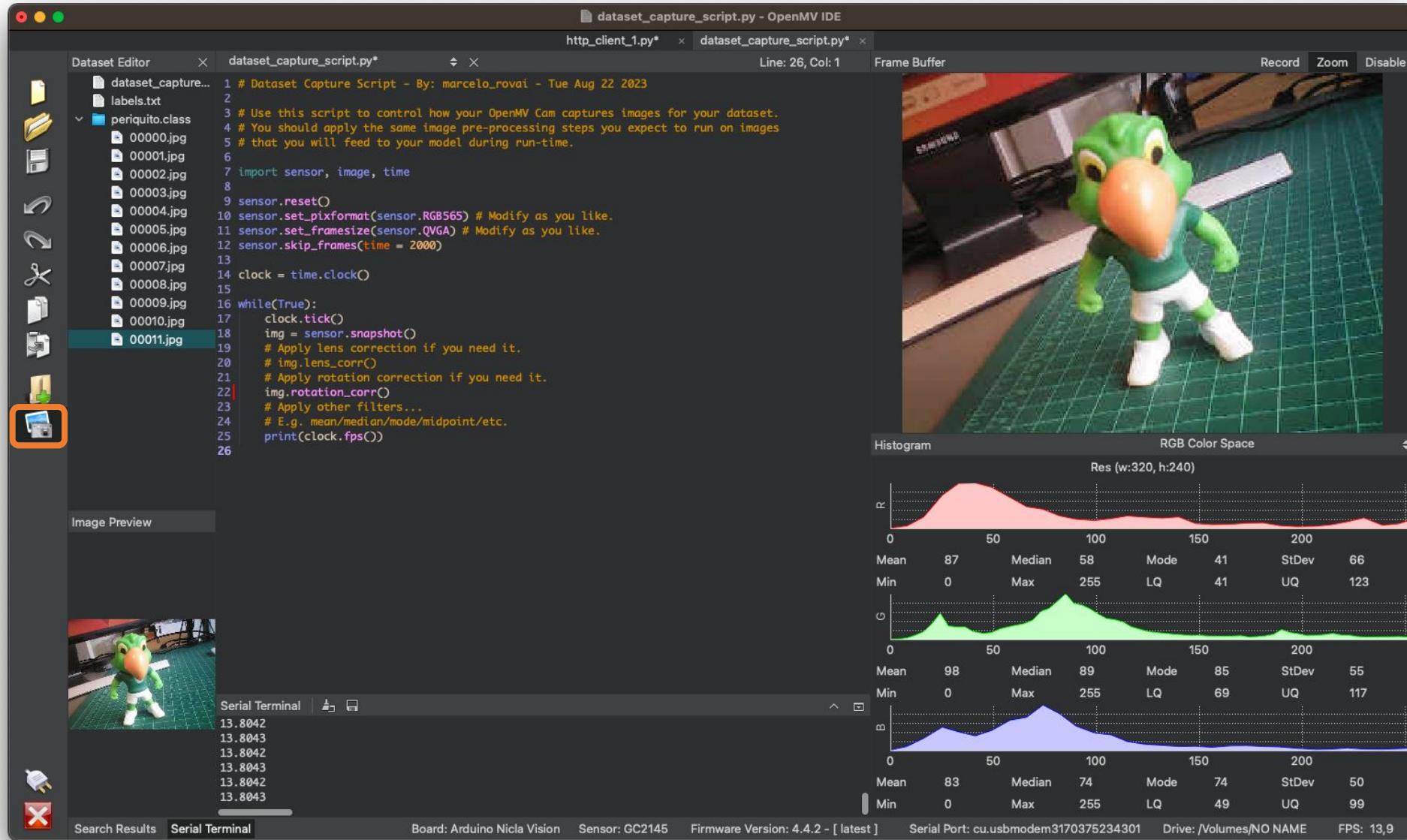


Create a folder, for example, **data**, and create a **class name**

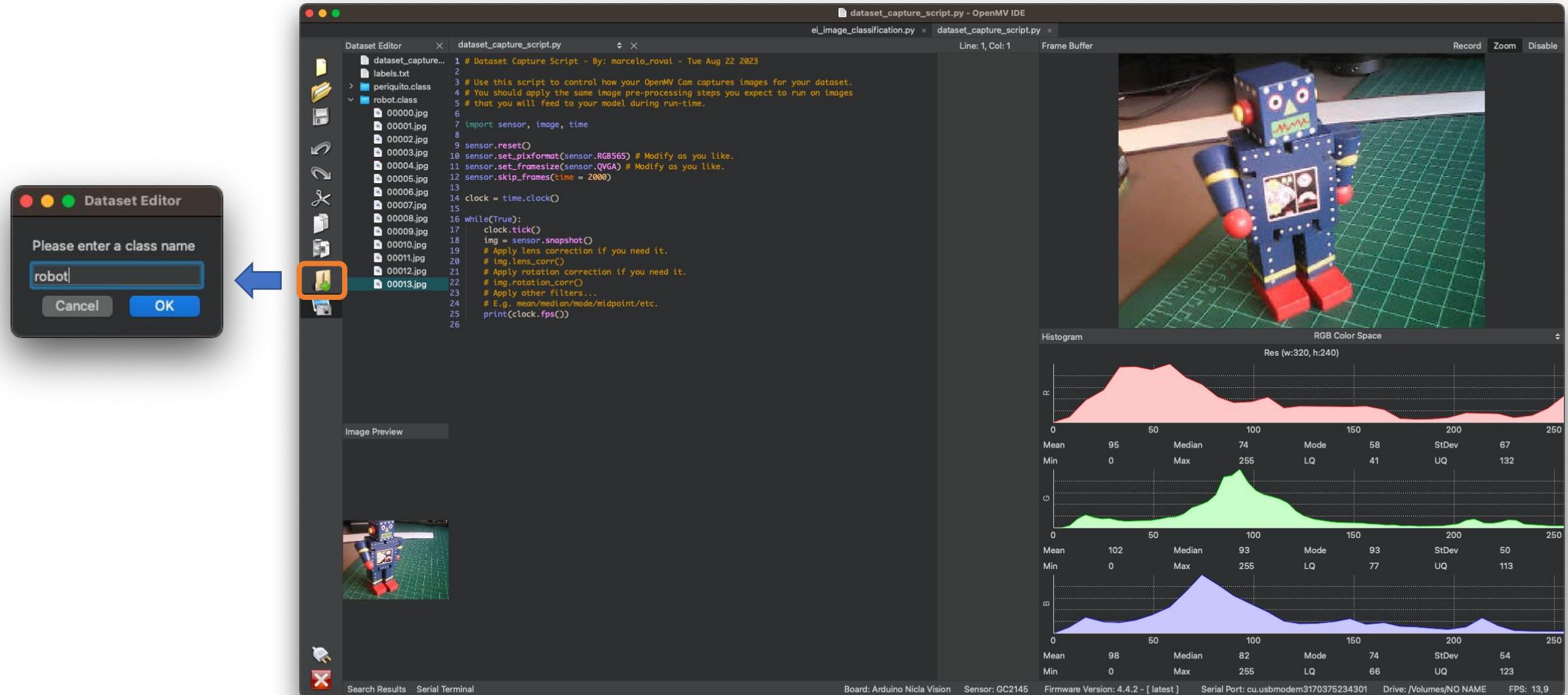


Start Collecting images ...

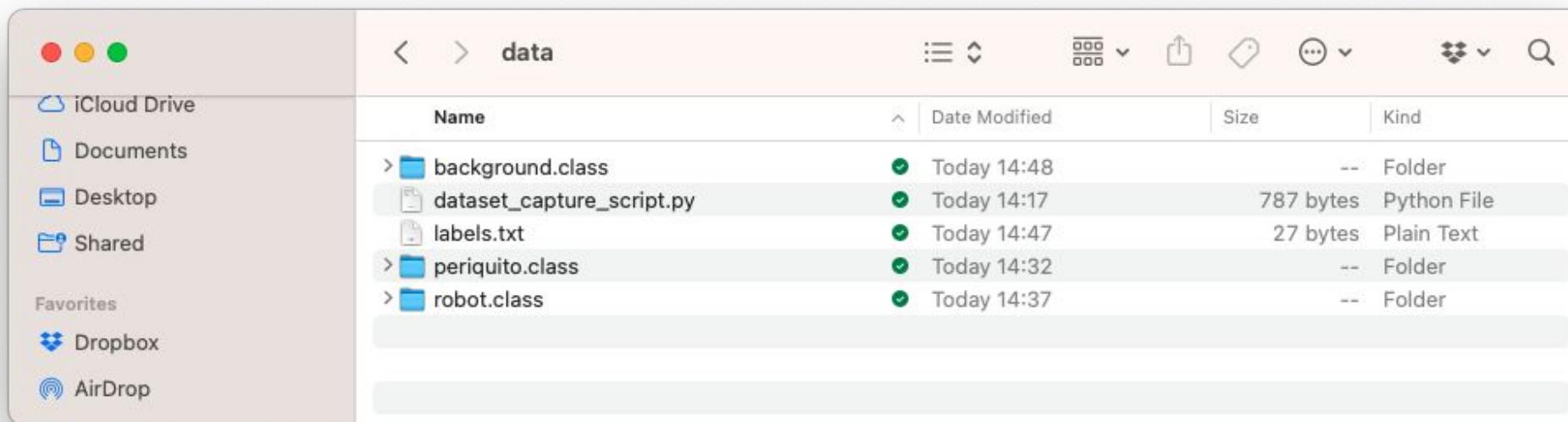
Click here
to
capture
an image



Repeat with other classes ...



- After capturing your dataset, close the Dataset Editor Tool.
- On your computer, you will end with a dataset that contains three classes: periquito, robot, and background.



- Go to Edge Impulse Studio and upload the dataset in your project

NICLA-Vision_Image_Classific | nome do mascote periquito do | Collecting image data with the | +

studio.edgeimpulse.com/studio/273858/acquisition/training?page=1#upload

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

Dataset

DATA

2 ite

Training

SAMPLE

periquito

periquito

Upload data

You can upload existing data to your project in the Data Acquisition Format (CBOR, JSON, CSV), or as WAV, JPG, PNG, AVI or MP4 files. We also support uploading image datasets with labels in a number of different formats. Include labels when uploading and we'll try to convert your dataset into a format recognized in Studio on upload. Learn more here.

Upload mode

Locations

- Marcelo's Mac...
- Macintosh HD
- NO NAME
- OneDrive
- Network
- iCloud
- iCloud Drive
- Documents
- Desktop
- Shared
- Favorites
- Dropbox
- Recents
- Applications
- Downloads
- marcelo_rovai

periquito.class

	Today	Size	Kind	Date Added
00050.jpg	15 KB	JPEG image	Today 14:32	
00049.jpg	13 KB	JPEG image	Today 14:32	
00048.jpg	14 KB	JPEG image	Today 14:32	
00047.jpg	15 KB	JPEG image	Today 14:31	
00046.jpg	14 KB	JPEG image	Today 14:31	
00045.jpg	13 KB	JPEG image	Today 14:31	
00044.jpg	15 KB	JPEG image	Today 14:31	
00043.jpg	15 KB	JPEG image	Today 14:31	
00042.jpg	13 KB	JPEG image	Today 14:31	
00041.jpg	18 KB	JPEG image	Today 14:31	
00040.jpg	15 KB	JPEG image	Today 14:31	
00039.jpg	15 KB	JPEG image	Today 14:31	
00038.jpg	17 KB	JPEG image	Today 14:31	
00037.jpg	13 KB	JPEG image	Today 14:31	
00036.jpg	16 KB	JPEG image	Today 14:31	
00035.jpg	15 KB	JPEG image	Today 14:31	
00034.jpg	15 KB	JPEG image	Today 14:31	

Search

Cancel Open

Back

Upload data

Metadata

No metadata.

Sample length (ms.)

Frequency

Start sampling

?

+

?

NICLA-Vision_Image_Classification | nome do mascote periquito do | Collecting image data with the | +

studio.edgeimpulse.com/studio/273858/acquisition/training?page=1#upload

EDGE IMPULSE

Dashboard Devices Data acquisition Impulse design Create impulse EON Tuner Retrain model Live classification Model testing Versioning Deployment

GETTING STARTED Documentation Forums

Upload data

You can upload existing data to your project in the [Data Acquisition Format](#) (CBOR, JSON, CSV), or as WAV, JPG, PNG, AVI or MP4 files. We also support uploading image datasets with labels in a number of different formats. Include labels when uploading and we'll try to convert your dataset into a format recognized in Studio on upload. [Learn more here.](#)

Upload mode

Select individual files [?](#)
 Select a folder [?](#)

Select files

[Choose Files](#) 51 files

Upload into category

Automatically split between training and testing [?](#)
 Training
 Testing

Label

Infer from filename [?](#)
 Leave data unlabeled [?](#)
 Enter label:

[Back](#) [Upload data](#)

MIDobot / Marcelo Bouzi / NICLA-Vision Image Classification

2 items

SAMPLES

periquito

periquito

Sample length (ms.)

Frequency

Start sampling

Metadata

No metadata.

NICLA-Vision_Image_Classification

nome do mascote periquito do

Collecting image data with the

studio.edgeimpulse.com/studio/273858/acquisition/training?page=4

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

EDGE IMPULSE

Dataset Data explorer Data sources | CSV Wizard

DATA COLLECTED 158 items

TRAIN / TEST SPLIT 78% / 22%

Dataset

Training (124) Test (34)

SAMPLE NAME	LABEL	ADDED
00051	background	Today, 14:58:43
00043	background	Today, 14:58:43
00036	background	Today, 14:58:43
00038	background	Today, 14:58:43
00034	background	Today, 14:58:43
00046	background	Today, 14:58:43
00001	robot	Today, 14:58:12
00003	robot	Today, 14:58:12
00002	robot	Today, 14:58:12
00004	robot	Today, 14:58:12
00005	robot	Today, 14:58:12
00006	robot	Today, 14:58:12

Collect data

Device ⓘ No devices connected

Label periquito

Sample length (ms.)

Sensor

Frequency

Start sampling

RAW DATA

00001

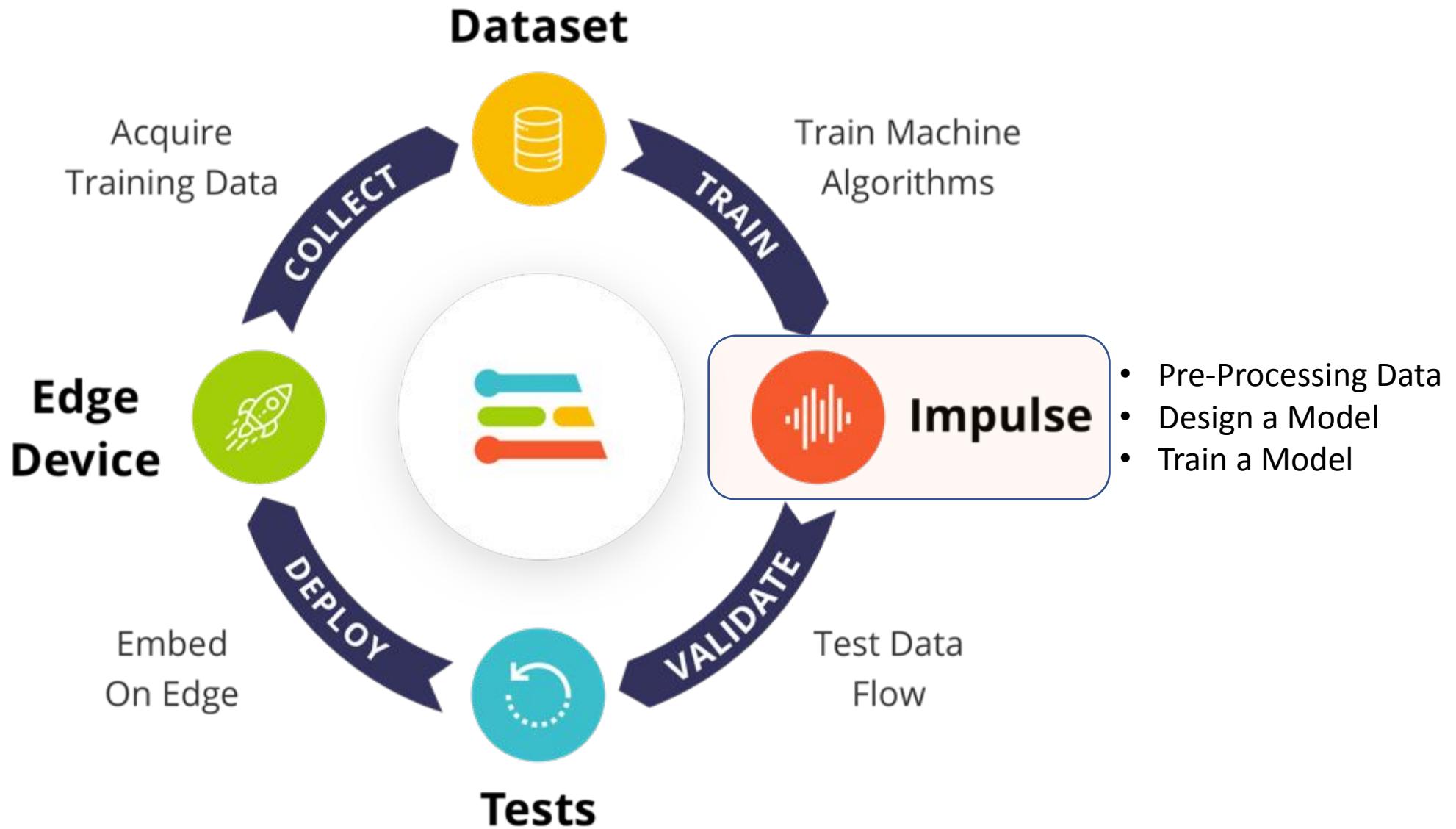
Metadata

No metadata.

?

+

< 1 2 3 4 5 6 ... 11 >



NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/create-impulse

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

EDGE IMPULSE

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

Image data

Input axes
image

Image width 96 **Image height** 96

Resize mode Squash

For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.

Image

Name Image

Input axes (1)
 image

Transfer Learning (Images)

Name Transfer learning

Input features
 Image

Output features
3 (background, periquito, robot)

Output features

3 (background, periquito, robot)

Save Impulse

Add a processing block

Add a learning block

© 2023 Edgimpulse Inc. All rights reserved

NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/dsp/image/3

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

EDGE IMPULSE

#1 ▾ Click to set a description for this version

Parameters Generate features

Raw data

Show: All labels 00022 (periquito)



Raw features

0x50e0b, 0x70d0b, 0x90c0b, 0xd0c0d, 0x100e0f, 0x121011, 0x131313, 0x141716, 0x131b18, 0x171d1b,...

Parameters

Image

Color depth: RGB

Save parameters

DSP result

Image



Processed features

0.0196, 0.0549, 0.0431, 0.0275, 0.0510, 0.0431, 0.0353, 0.0471, 0.0431, 0.0510, 0.0471, 0.0510,...

On-device performance

PROCESSING TIME: 1 ms.

PEAK RAM USAGE: 4 KB

© 2023 EdgeImpulse Inc. All rights reserved

NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/dsp/image/3/generate-features

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

EDGE IMPULSE

#1 ▾ Click to set a description for this version

Parameters Generate features

Training set

Data in training set 124 items

Classes 3 (background, periquito, robot)

Generate features

Feature explorer

background (blue), periquito (orange), robot (green)

On-device performance ②

PROCESSING TIME 1 ms.

PEAK RAM USAGE 4 KB

© 2023 EdgeImpulse Inc. All rights reserved

NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/data-explorer

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

EDGE IMPULSE

Dataset Data explorer Data sources | CSV Wizard

Data explorer

The data explorer shows a complete view of all data in your project. Use it to quickly label your data, or spot outliers. [Learn more](#).

How should we generate the data explorer?

- Using a pretrained visual model
Great for most image classification projects.
- Using your trained impulse
Works great if you have collected some labeled data already and have a trained model.
- Using the preprocessing blocks in your impulse
Use this if you don't have any labels for your data yet, and thus can't train a full model.

Dimensionality reduction technique

- t-SNE
Recommended for your dataset. Separates best, but takes a significant amount of time on large datasets.
- PCA
Separates less well, but works on any dataset size.

[Generate data explorer](#)

© 2023 Edgimpulse Inc. All rights reserved

NICLA-Vision_Image_Classifier X +

studio.edgeimpulse.com/studio/273858/data-explorer

The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items. [Learn more.](#)

EDGE IMPULSE

Drag mode: Box select items

0 items selected [Set labels](#) [Delete items](#) Save changes (0 pending)

background periquito robot

Dashboard Devices Data acquisition Impulse design Create impulse Image Transfer learning EON Tuner Retrain model Live classification Model testing Versioning Deployment

GETTING STARTED Documentation Forums

00017 Label: periquito [View sample](#)

Set label Delete item

Background data points (blue dots) are scattered across the grid. Periquito data points (orange dots) are clustered in the lower-left quadrant. Robot data points (green dots) are clustered in the upper-right quadrant. A detailed image of a green and yellow parrot toy is shown in the bottom right corner.

Model Design

MobileNetV1 96x96 0.1

Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Model

MobileNetV2 96x96 0.35

Uses around 296.8K RAM and 575.2K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Image Size

MobileNetV2 96x96 0.1

Uses around 270.2K RAM and 212.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

Alpha

MobileNetV2 96x96 0.05

Uses around 265.3K RAM and 162.4K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.

NICLA-Vision_Image_Classifier

studio.edgeimpulse.com/studio/273858/learning/keras-transfer-image/5

Choose a different model

Did you know? You can customize your model through the Expert view (click on to switch), or can even bring your own model (in PyTorch, Keras or scikit-learn).

MODEL	AUTHOR
MobileNetV1 96x96 0.25 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV1 96x96 0.2 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV1 96x96 0.1 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV2 96x96 0.35 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV2 96x96 0.1 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV2 96x96 0.05 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV2 160x160 1.0 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse
MobileNetV2 160x160 0.75 <small>OFFICIALLY SUPPORTED</small>	Edge Impulse

Target: Arduino Portenta H7 (Cortex-M7 480MHz)

(0)

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- Transfer learning

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

© 2023 EdgeImpulse Inc. All rights reserved.

?

Neural Network settings

Training settings

Number of training cycles [?](#)

Learning rate [?](#)

Data augmentation [?](#)

Advanced training settings

Validation set size [?](#) %

Split train/validation set on metadata key [?](#)

Auto-balance dataset [?](#)

Profile int8 model [?](#)

Neural network architecture

Input layer (27,648 features)



MobileNetV2 96x96 0.1 (final layer: 12 neurons, 0.15 dropout)

Choose a different model

Output layer (3 classes)

Start training

Model

Model version: [?](#) Quantized (int8) [▼](#)

Last training performance (validation set)


ACCURACY
92.0%


LOSS
0.13

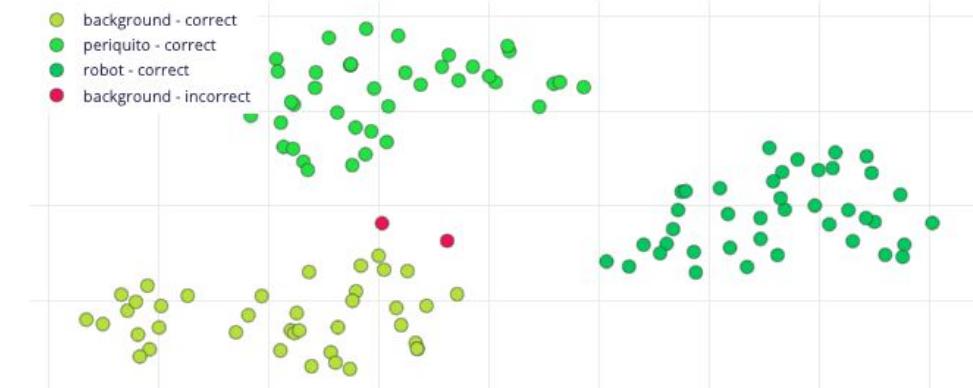
Confusion matrix (validation set)

	BACKGROUND	PERIQUITO	ROBOT
BACKGROUND	77.8%	11.1%	11.1%
PERIQUITO	0%	100%	0%
ROBOT	0%	0%	100%
F1 SCORE	0.88	0.94	0.94

Data explorer (full training set) [?](#)

Legend:

- background - correct (light green)
- periquito - correct (medium green)
- robot - correct (dark green)
- background - incorrect (red)

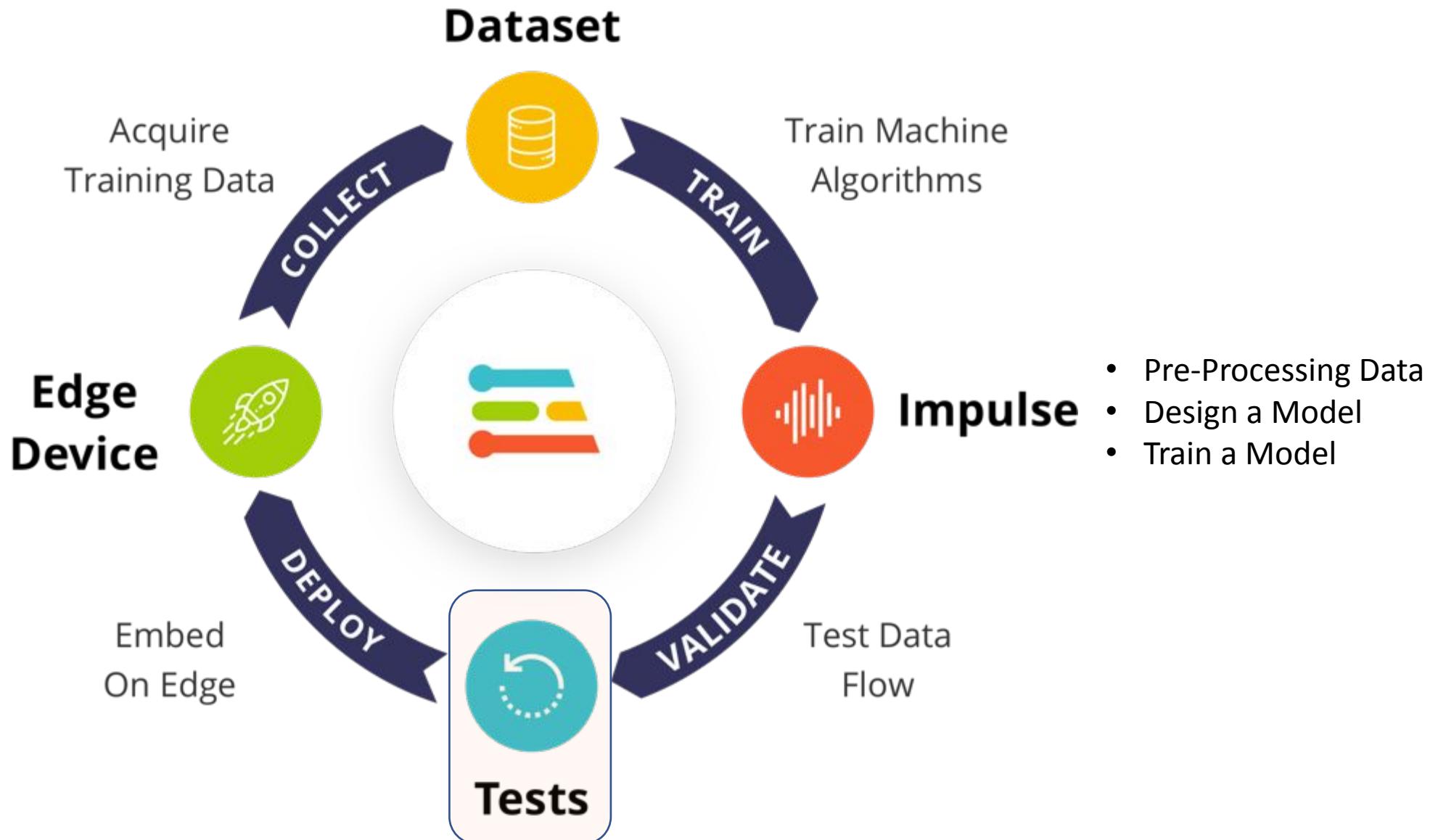


On-device performance [?](#)


INFERRING TIME
77 ms.


PEAK RAM USAGE
280.0K


FLASH USAGE
211.8K



NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/validation

Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT
00037	background	-	100%	1 background
00000	robot	-	100%	1 robot
00008	robot	-	100%	1 robot
00007	robot	-	100%	1 robot
00016	robot	-	100%	1 robot
00014	robot	-	100%	1 robot
00021	robot	-	100%	1 robot
00026	robot	-	100%	1 robot
00041	robot	-	100%	1 robot
00044	robot	-	100%	1 robot
			100%	1 robot
			100%	1 robot
			100%	1 periquito

Model testing output

```
Copying features from DSP block...
Copying features from DSP block OK
Copying features from processing blocks OK

Classifying data for float32 model...
Scheduling job in cluster...
Container image pulled!
Job started
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Classifying data for Transfer learning OK

Job completed
```

Model testing results

ACCURACY 97.06%

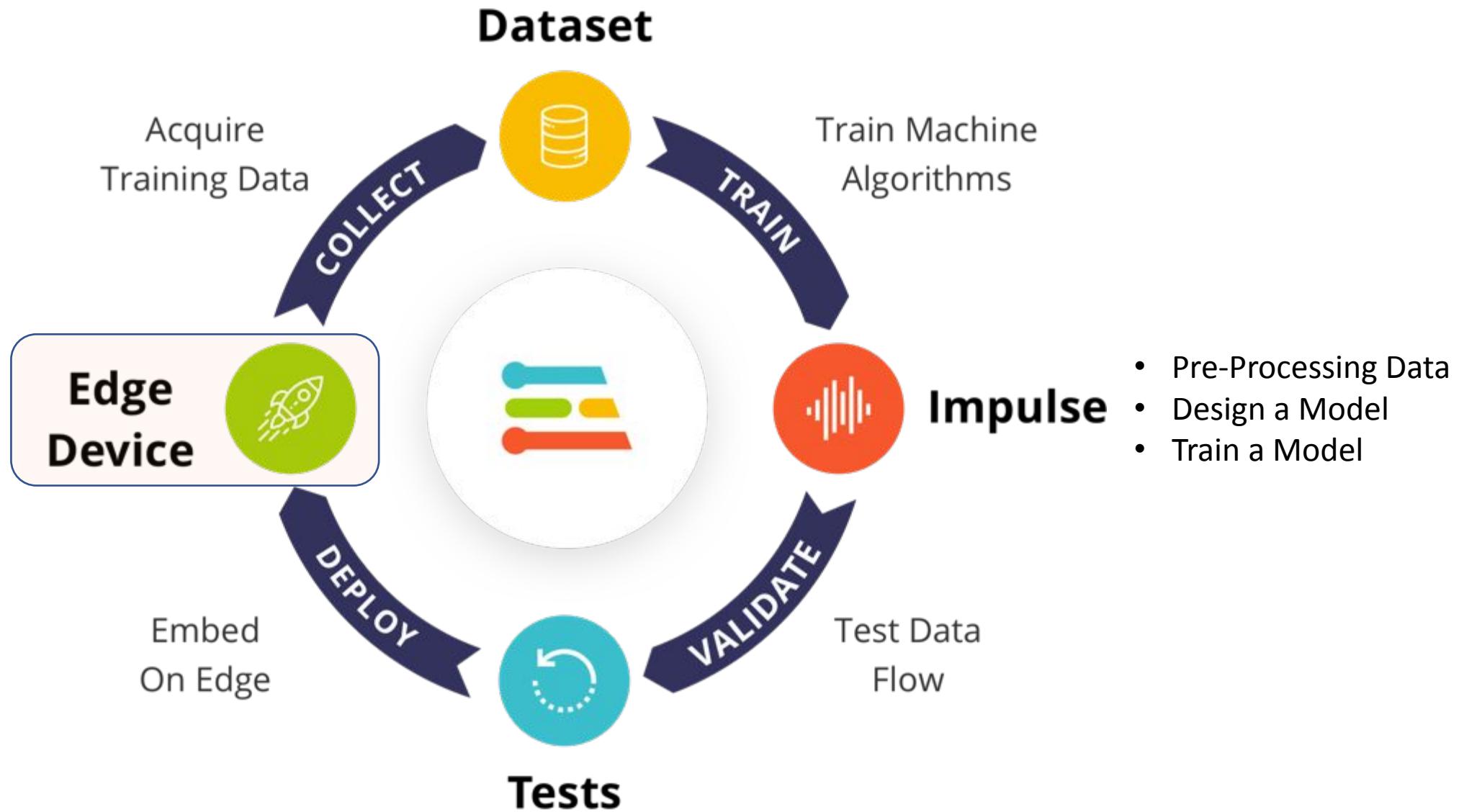
	BACKGROUND	PERIQUITO	ROBOT	UNCERTAIN
BACKGROUND	100%	0%	0%	0%
PERIQUITO	0%	90.9%	0%	9.1%
ROBOT	0%	0%	100%	0%
F1 SCORE	1.00	0.95	1.00	

Feature explorer

Legend:

- background - correct (light green)
- periquito - correct (dark green)
- robot - correct (dark green)
- periquito - incorrect (red)

Final Performance Estimate



NICLA-Vision_Image_Classification

studio.edgeimpulse.com/studio/273858/deployment-view

EDGE IMPULSE

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- Transfer learning

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

- Documentation
- Forums

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.](#)

OpenMV Firmware

SELECTED DEPLOYMENT

OpenMV Firmware

Firmware binary that includes your model and runs on OpenMV cameras.

MODEL OPTIMIZATIONS

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

Quantized (int8)

Selected ✓

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	1 ms.	77 ms.	78 ms.
RAM	4.0K	320.2K	320.2K
FLASH	-	284.6K	-
ACCURACY			97.06%

Unoptimized (float32)

Select

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	1 ms.	156 ms.	157 ms.
RAM	4.0K	1.1M	1.1M
FLASH	-	471.8K	-
ACCURACY			97.06%

Estimate for Arduino Nicla Vision (Cortex-M7 480MHz) - [Change target](#)

Build

Latest build

v8 (OpenMV Firmware)
Today, 09:38:51

Run this model

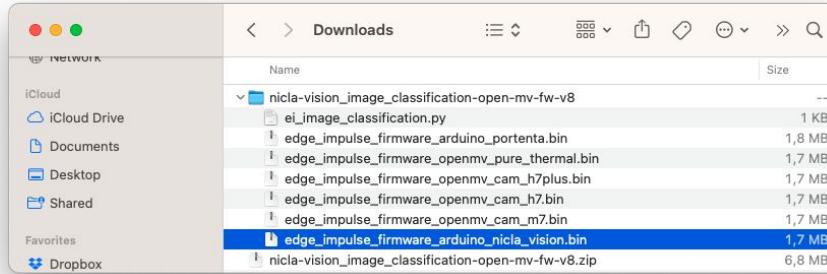
Scan QR code or launch in browser to test your prototype

QR code

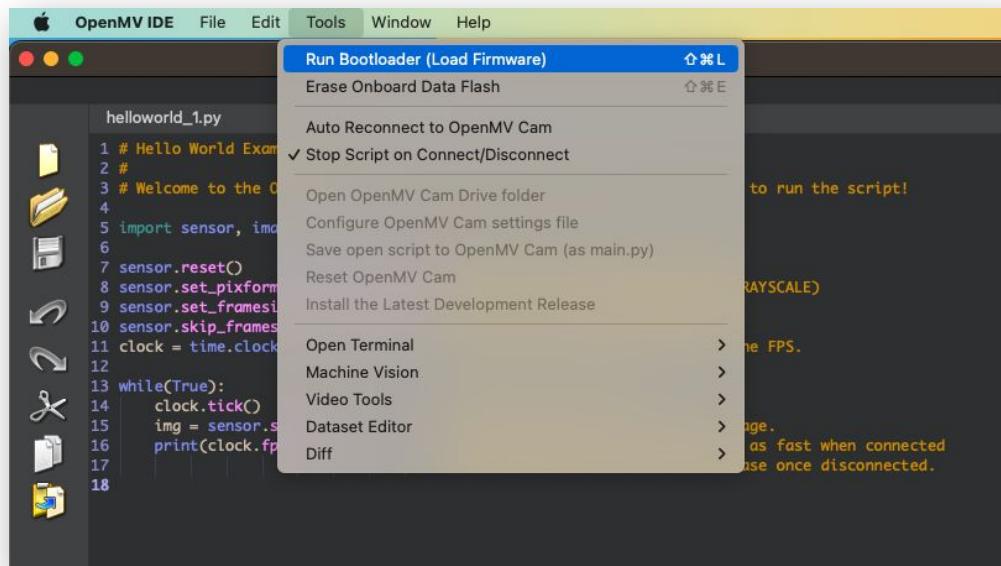
Launch in browser

Deploying model on OpenMV IDE

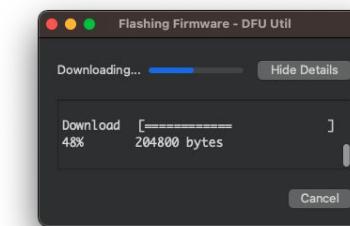
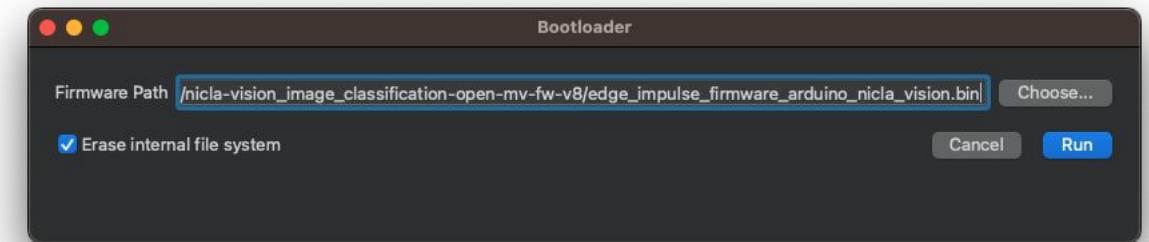
1. On your computer, you will find a ZIP file. Open it:



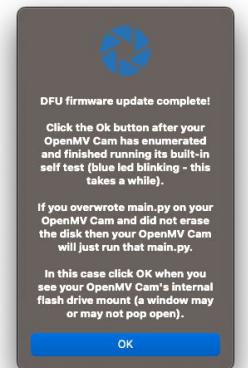
2. Use the Bootloader tool on OpenMV IDE to load it on your board:



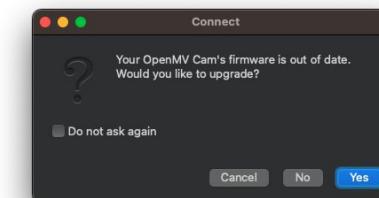
3. Select the appropriate file (.bin for Nicla-Vision) and press [Run]:



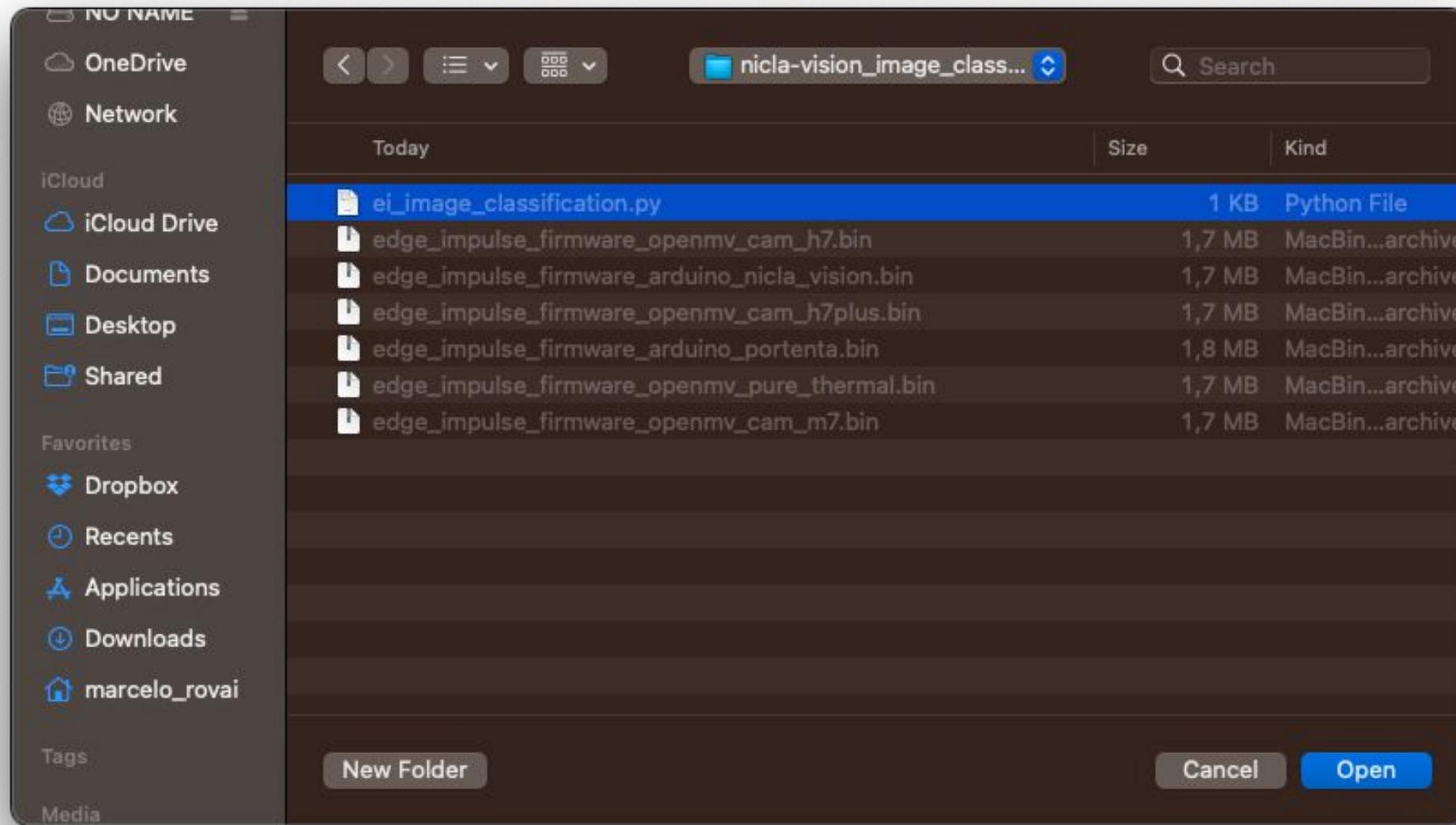
4. After the download is finished, press OK:



5. If a message saying that the FW is outdated, DO NOT UPGRADE



Open the script: **ei_image_classification.py** that was downloaded from the Studio, and run it:



ei_image_classification.py - OpenMV IDE

helloworld_1.py periquitoVsrobot.py ei_image_classification.py

Line: 38, Col: 1 Frame Buffer

Record Zoom Disable

ei_image_classification.py

```

1 # Edge Impulse - OpenMV Image Classification Example
2
3 import sensor, image, time, os, tf, uos, gc
4
5 sensor.reset()          # Reset and initialize the sensor.
6 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRayscale)
7 sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
8 sensor.set_windowing((240, 240))    # Set 240x240 window.
9 sensor.skip_frames(time=2000)       # Let the camera adjust.
10
11 net = None
12 labels = None
13
14 try:
15     # Load built in model
16     labels, net = tf.load_builtin_model('trained')
17 except Exception as e:
18     raise Exception(e)
19
20
21 clock = time.clock()
22 while(True):
23     clock.tick()
24
25     img = sensor.snapshot()
26
27     # default settings just do one detection... change them to search the image...
28     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
29         print("*****\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
30         img.draw_rectangle(obj.rect())
31     # This combines the labels and confidence values into a list of tuples
32     predictions_list = list(zip(labels, obj.output()))
33
34     for i in range(len(predictions_list)):
35         print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
36
37     print(clock.fps(), "fps")
38 
```

Serial Terminal

```

Predictions at [x=0,y=0,w=240,h=240]
background = 0.000000
periquito = 0.996094
robot = 0.000000
7.34533 fps
*****
Predictions at [x=0,y=0,w=240,h=240]
background = 0.000000
periquito = 0.996094
robot = 0.003906
7.34535 fps

```

Search Results Serial Terminal

Board: Arduino Nicla Vision Sensor: GC2145 Firmware Version: 4.3.1 - [out of date - click here to upgrade] Serial Port: cu.usbmodem3170375234301 Drive: /Volumes/NO NAME FPS: 7,4

Frame Buffer

Histogram

RGB Color Space

Res (w:240, h:240)

	R	G	B
Mean	96	106	92
Min	0	0	0
Max	255	255	255
Median	74	97	82
Mode	49	85	90
StDev	65	57	56
LQ	49	69	49
UQ	123	134	115

```

ei_image_classification.py      ◆ ×

1 # Edge Impulse - OpenMV Image Classification Example
2
3 import sensor, image, time, os, tf, uos, gc
4
5 sensor.reset()          # Reset and initialize the sensor.
6 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRayscale)
7 sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
8 sensor.set_windowing((240, 240))   # Set 240x240 window.
9 sensor.skip_frames(time=2000)       # Let the camera adjust.
10
11 net = None
12 labels = None
13
14 try:
15     # Load built in model
16     labels, net = tf.load_builtin_model('trained')
17 except Exception as e:
18     raise Exception(e)
19
20
21 clock = time.clock()
22 while(True):
23     clock.tick()
24
25     img = sensor.snapshot()
26
27     # default settings just do one detection... change them to search the image...
28     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
29         print("*****\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
30         img.draw_rectangle(obj.rect())
31     # This combines the labels and confidence values into a list of tuples
32     predictions_list = list(zip(labels, obj.output()))
33
34     for i in range(len(predictions_list)):
35         print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
36
37 print(clock.fps(), "fps")
38

```

Serial Terminal |

Predictions at [x=0,y=0,w=240,h=240]

background = 0.000000

periquito = 0.007813

robot = 0.992188

7.33936 fps

Predictions at [x=0,y=0,w=240,h=240]

background = 0.000000

periquito = 0.015625

robot = 0.984375

7.33938 fps



```

ei_image_classification.py
1 # Edge Impulse - OpenMV Image Classification Example
2
3 import sensor, image, time, os, tf, uos, gc
4
5 sensor.reset()          # Reset and initialize the sensor.
6 sensor.set_pixformat(sensor.RGB565) # Set pixel format to RGB565 (or GRayscale)
7 sensor.set_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)
8 sensor.set_windowing((240, 240))   # Set 240x240 window.
9 sensor.skip_frames(time=2000)      # Let the camera adjust.
10
11 net = None
12 labels = None
13
14 try:
15     # Load built in model
16     labels, net = tf.load_builtin_model('trained')
17 except Exception as e:
18     raise Exception(e)
19
20
21 clock = time.clock()
22 while(True):
23     clock.tick()
24
25     img = sensor.snapshot()
26
27     # default settings just do one detection... change them to search the image...
28     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
29         print("*****\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
30         img.draw_rectangle(obj.rect())
31     # This combines the labels and confidence values into a list of tuples
32     predictions_list = list(zip(labels, obj.output()))
33
34     for i in range(len(predictions_list)):
35         print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
36
37 print(clock.fps(), "fps")
38

```

Serial Terminal |

Predictions at [x=0,y=0,w=240,h=240]

background = 0.957031

periquito = 0.039063

robot = 0.003906

7.33808 fps

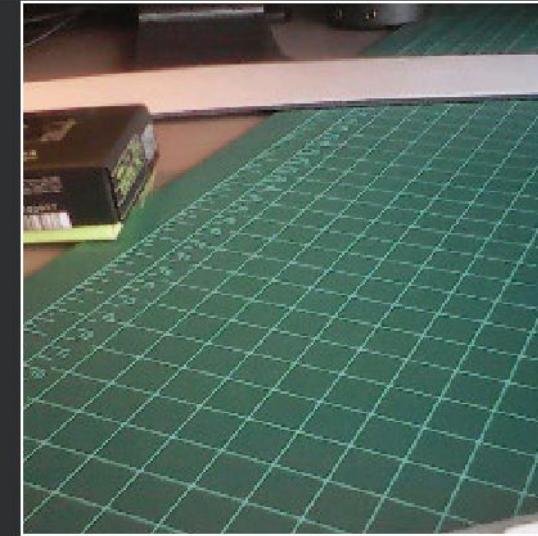
Predictions at [x=0,y=0,w=240,h=240]

background = 0.933594

periquito = 0.062500

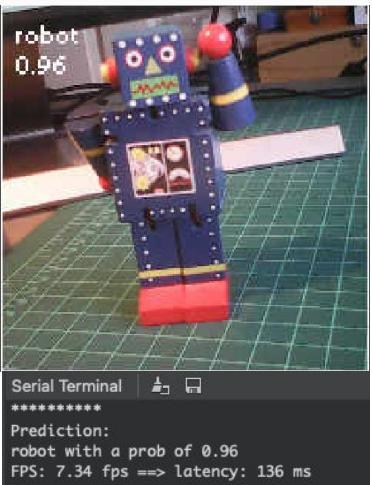
robot = 0.003906

7.33809 fps



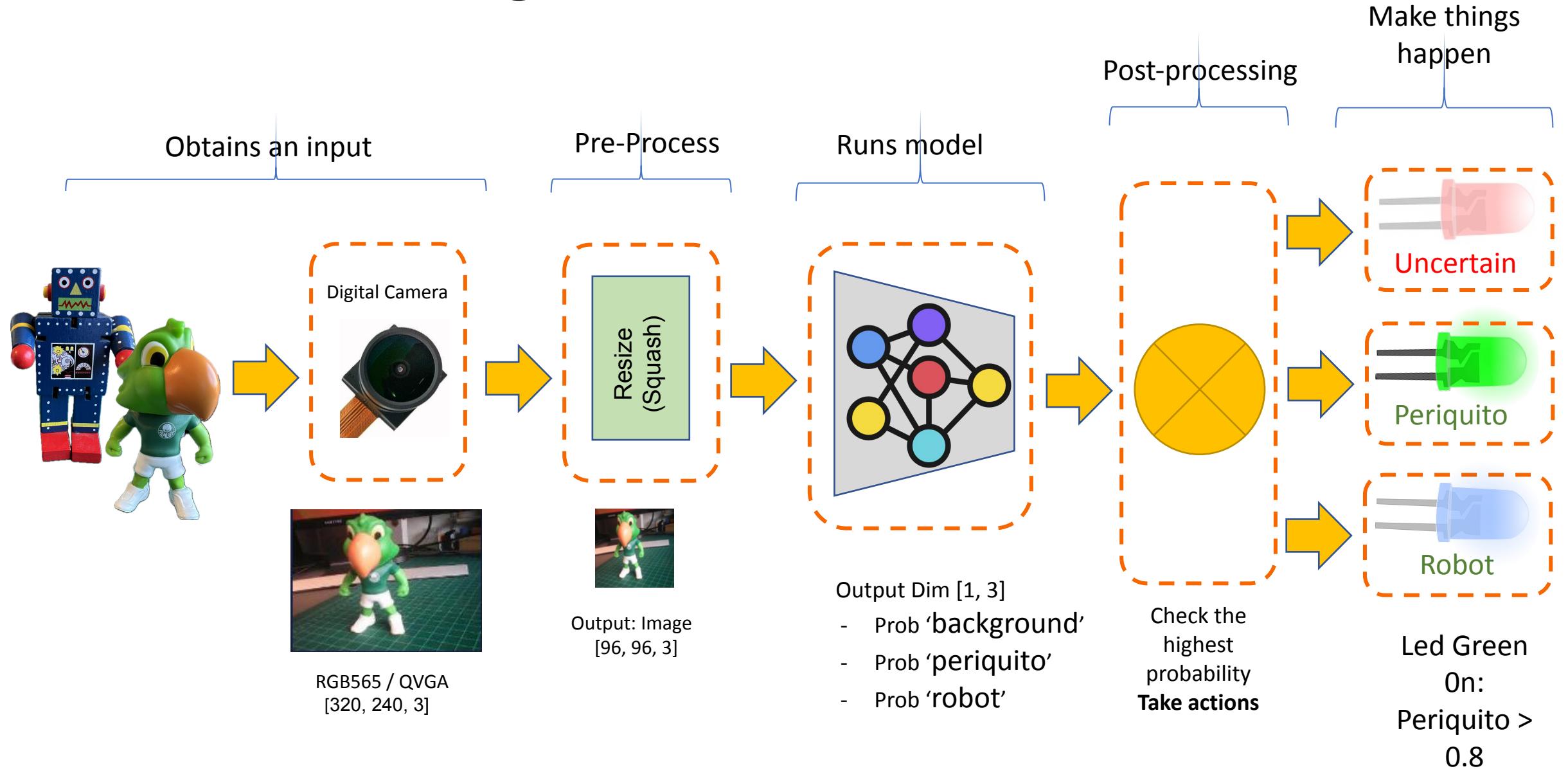
Changing Code to add labels:

```
1 # Marcelo Rovai - NICLA Vision - Image Classification
2 # Adapted from Edge Impulse - OpenMV Image Classification Example
3 # @24Aug23
4
5 import sensor, image, time, os, tf, uos, gc
6
7 sensor.reset()                      # Reset and initialize the sensor.
8 sensor.set_pixformat(sensor.RGB565)   # Set pixel format to RGB565 (or GRayscale)
9 sensor.set_framesize(sensor.QVGA)     # Set frame size to QVGA (320x240)
10 sensor.set_windowing((240, 240))    # Set 240x240 window.
11 sensor.skip_frames(time=2000)        # Let the camera adjust.
12
13 net = None
14 labels = None
15
16 try:
17     # Load built in model
18     labels, net = tf.load_builtin_model('trained')
19 except Exception as e:
20     raise Exception(e)
21
22 clock = time.clock()
23
24
```



```
25 while(True):
26     clock.tick() # Starts tracking elapsed time.
27
28     img = sensor.snapshot()
29
30     # default settings just do one detection... change them to search the image...
31     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
32         fps = clock.fps()
33         lat = clock.avg()
34
35         print("*****\nPrediction:")
36         img.draw_rectangle(obj.rect())
37         # This combines the labels and confidence values into a list of tuples
38         predictions_list = list(zip(labels, obj.output()))
39
40         max_val = predictions_list[0][1]
41         max_lbl = 'background'
42         for i in range(len(predictions_list)):
43             val = predictions_list[i][1]
44             lbl = predictions_list[i][0]
45             #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
46
47             if val > max_val:
48                 max_val = val
49                 max_lbl = lbl
50
51         # Print label with the highest probability
52         if max_val < 0.5:
53             max_lbl = 'uncertain'
54
55         #print("%s with a prob of %f" % (max_lbl, max_val))
56         print("{} with a prob of {:.2f}.".format(max_lbl, max_val))
57         # print(clock.fps(), "fps")
58         print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))
59
60         # Draw label with highest probability to image viewer
61         img.draw_string(
62             10, 10,
63             max_lbl + "\n{:.2f}".format(max_val),
64             mono_space = False,
65             scale=2
66         )
67
```

Post-Processing with LEDs:



Post-Processing with LEDs:

```
nicla_image_classification_LED.py  ◊ ×

1 # Marcelo Rovai - NICLA Vision - Image Classification with LEDs
2 # Adapted from Edge Impulse - OpenMV Image Classification Example
3 # @24Aug23
4
5 import sensor, time, os, tf, uos, gc, pyb
6
7 ledRed = pyb.LED(1)
8 ledGre = pyb.LED(2)
9 ledBlu = pyb.LED(3)
10
11 sensor.reset()          # Reset and initialize the sensor.
12 sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRayscale)
13 sensor.set_framesize(sensor.QVGA)        # Set frame size to QVGA (320x240)
14 sensor.set_windowing((240, 240))        # Set 240x240 window.
15 sensor.skip_frames(time=2000)            # Let the camera adjust.
16
17 net = None
18 labels = None
19
20 ledRed.off()
21 ledGre.off()
22 ledBlu.off()
23
24 try:
25     # Load built in model
26     labels, net = tf.load_builtin_model('trained')
27 except Exception as e:
28     raise Exception(e)
29
30 clock = time.clock()
31
32 def setLEDs(max_lbl):
33
34     if max_lbl == 'uncertain':
35         ledRed.on()
36         ledGre.off()
37         ledBlu.off()
38
39     if max_lbl == 'periquito':
40         ledRed.off()
41         ledGre.on()
42         ledBlu.off()
43
44     if max_lbl == 'robot':
45         ledRed.off()
46         ledGre.off()
47         ledBlu.on()
48
49     if max_lbl == 'background':
50         ledRed.off()
51         ledGre.off()
52         ledBlu.off()
53
```

```
54
55
56 while(True):
57     clock.tick() # Starts tracking elapsed time.
58
59     img = sensor.snapshot()
60
61     # default settings just do one detection... change them to search the image...
62     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
63         fps = clock.fps()
64         lat = clock.avg()
65
66         print("*****\nPrediction:")
67         img.draw_rectangle(obj.rect())
68         # This combines the labels and confidence values into a list of tuples
69         predictions_list = list(zip(labels, obj.output()))
70
71         max_val = predictions_list[0][1]
72         max_lbl = 'background'
73         for i in range(len(predictions_list)):
74             val = predictions_list[i][1]
75             lbl = predictions_list[i][0]
76             #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
77
78             if val > max_val:
79                 max_val = val
80                 max_lbl = lbl
81
82         # Print label and turn on LED with the highest probability
83         if max_val < 0.8:
84             max_lbl = 'uncertain'
85
86         setLEDs(max_lbl)
87
88
89         #print("%s with a prob of %f" % (max_lbl, max_val))
90         print("{} with a prob of {:.2f}.".format(max_lbl, max_val))
91         # print(clock.fps(), "fps")
92         print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))
93
94         # Draw label with highest probability to image viewer
95         img.draw_string(
96             10, 10,
97             max_lbl + "\n{:.2f}".format(max_val),
98             mono_space = False,
99             scale=2
100            )
101
102
```



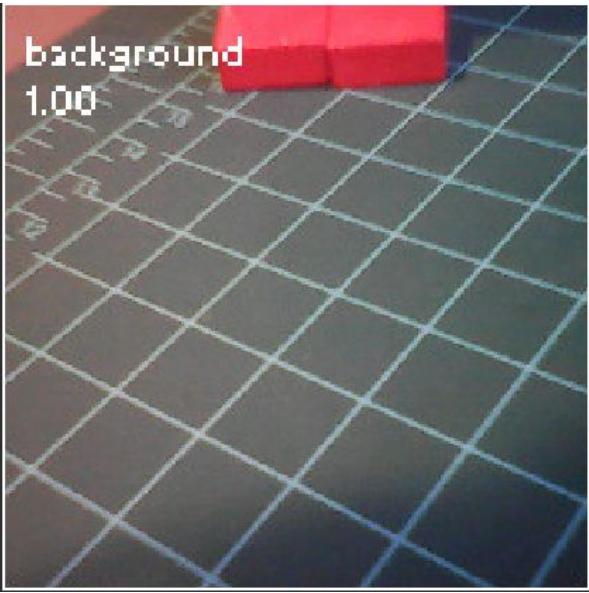
Serial Terminal | ↻ ⟲

```
*****  
Prediction:  
periquito with a prob of 0.99  
FPS: 7.37 fps ==> latency: 136 ms
```



Serial Terminal | ↻ ⟲

```
*****  
Prediction:  
robot with a prob of 0.96  
FPS: 7.34 fps ==> latency: 136 ms
```



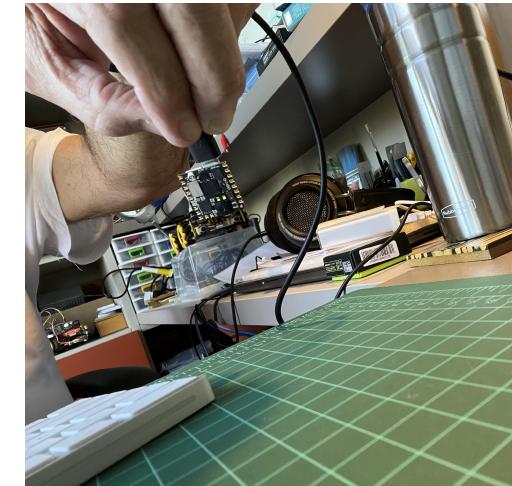
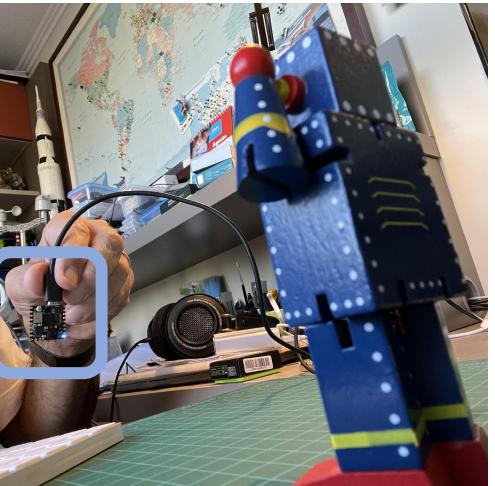
Serial Terminal | ↻ ⟲

```
*****  
Prediction:  
background with a prob of 1.00  
FPS: 7.34 fps ==> latency: 136 ms
```



Serial Terminal | ↻ ⟲

```
*****  
Prediction:  
uncertain with a prob of 0.76  
FPS: 7.42 fps ==> latency: 135 ms
```



The clock started just before inference.

Here, the latency is only related to the time used for inference

```

49      if max_lbl == 'background':
50          ledRed.off()
51          ledGre.off()
52          ledBlu.off()
53
54
55 while(True):
56
57     img = sensor.snapshot()
58
59     clock.tick() # Starts tracking elapsed time.
60
61     # default settings just do one detection... change them to search the image...
62     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
63         fps = clock.fps()
64         lat = clock.avg()
65
66         print("*****\nPrediction:")
67         img.draw_rectangle(obj.rect())
68         # This combines the labels and confidence values into a list of tuples
69         predictions_list = list(zip(labels, obj.output()))
70
71         max_val = predictions_list[0][1]
72         max_lbl = 'background'
73         for i in range(len(predictions_list)):
74             val = predictions_list[i][1]
75             lbl = predictions_list[i][0]
76             #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
77
78             if val > max_val:
79                 max_val = val
80                 max_lbl = lbl
81
82
83         # Print label and turn on LED with the highest probability
84         if max_val < 0.5:
85             max_lbl = 'uncertain'
86
87         setLEDs(max_lbl)
88
89
90         #print("%s with a prob of %f" % (max_lbl, max_val))
91         print("{} with a prob of {:.2f}.".format(max_lbl, max_val))
92         # print(clock.fps(), "fps")
93         print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))
94
95         # Draw label with highest probability to image viewer
96         img.draw_string(
97             10, 10,
98             max_lbl + "\n{:.2f}.".format(max_val),
99             mono_space = False,
100            scale=2
101        )
102
103

```

Serial Terminal

```

*****
Prediction:
robot with a prob of 0.93
FPS: 14.03 fps ==> latency: 71 ms

```

Frame Buffer

Histogram

RGB Color Space

Res (w:240, h:240)

	0	50	100	150	200	250
A	Mean	108	Median	90	Mode	74
A	Min	0	Max	255	LQ	66
A	StDev	61	UQ	132		

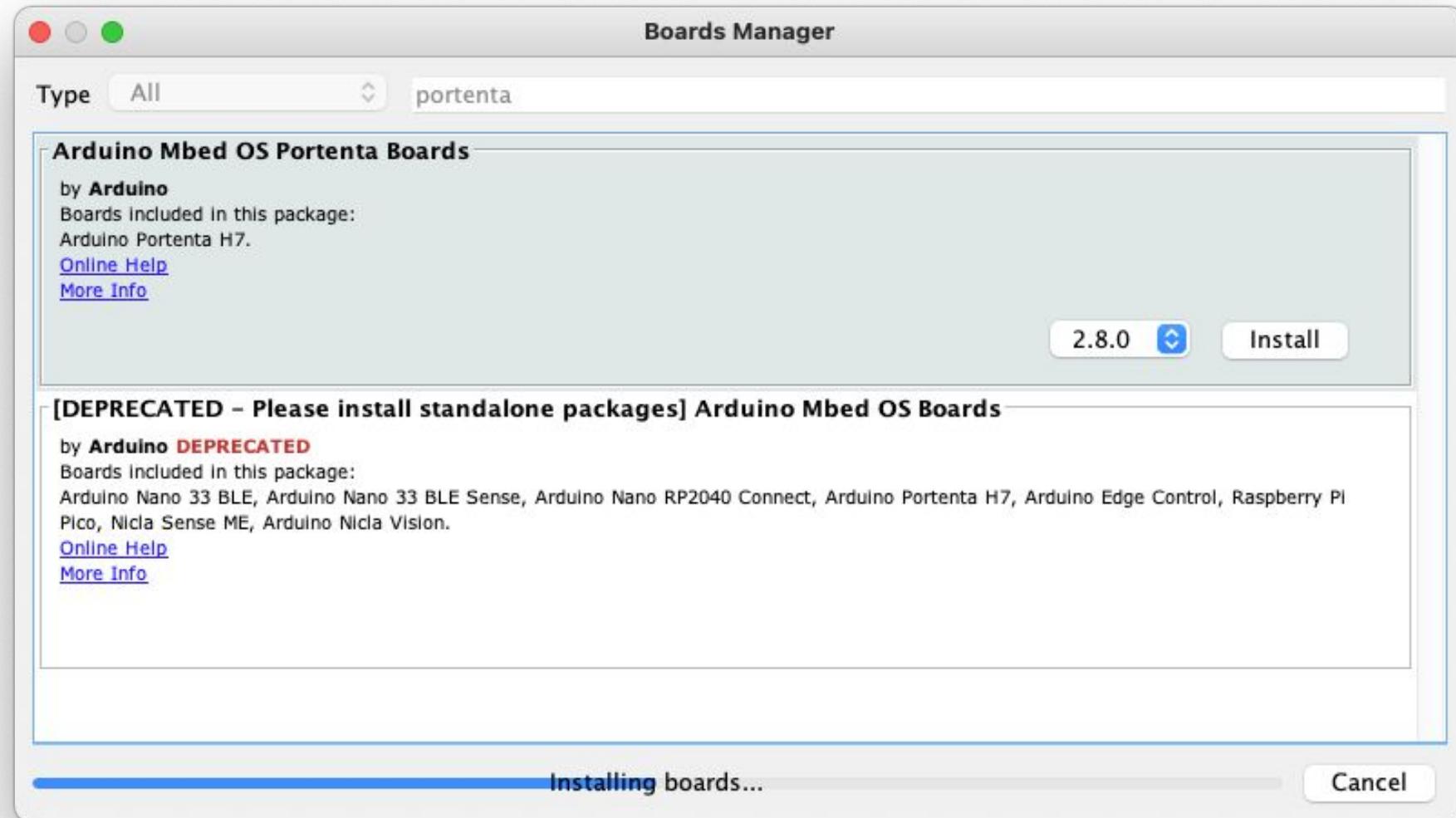
	0	50	100	150	200	250
G	Mean	95	Median	85	Mode	97
G	Min	0	Max	255	LQ	53
G	StDev	55	UQ	121		

	0	50	100	150	200	250
B	Mean	95	Median	82	Mode	90
B	Min	0	Max	255	LQ	58
B	StDev	53	UQ	115		

Board: Arduino Nicla Vision Sensor: GC2145 Firmware Version: 4.3.1 - [out of date - click here to upgrade] Serial Port: cu.usbmodem3170375234301 Drive: /Volumes/NO NAME FPS: 74

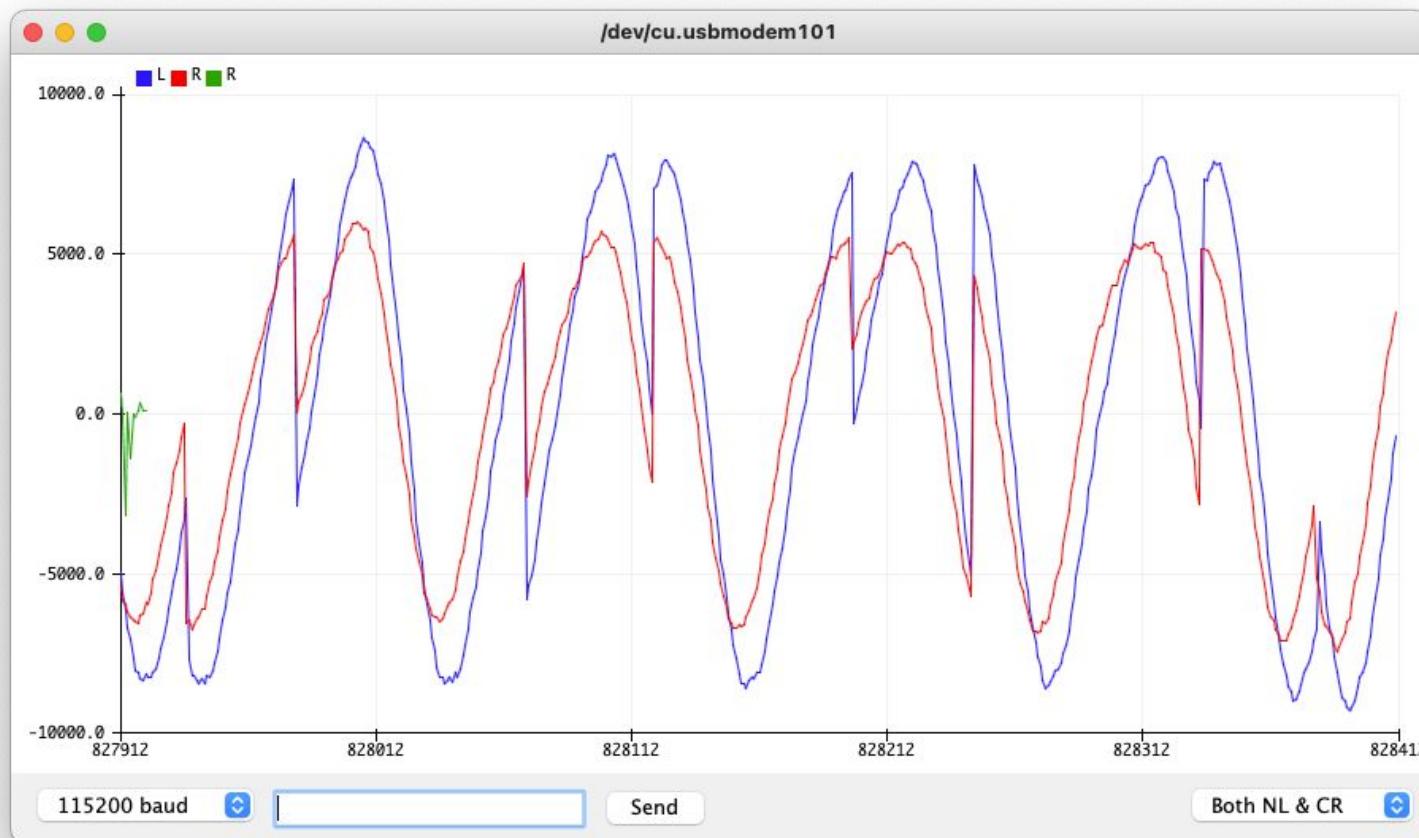
Install HW (Portenta H7)
& Sensors' test (optional)





Test Hw ---> Run Blink (Green RGB Led)

Microphones Test



**Go to Examples > PDM >
PDMSerialPlotter**

**and test the 2 channels (left
and right) and sample freq.
of 32KHz**

```
// default number of output channels
static const char channels = 2;

// default PCM output frequency
static const int frequency = 32000;

// Buffer to read samples into, each sample is 16-bits
short sampleBuffer[512];
```

Profile - Projects - Edge Impulse X +

studio.edgeimpulse.com/studio/profile/projects

EDGE IMPULSE

Projects Custom ML blocks

MJRoBot (Marcelo Rovai)

Organizations EIE

Projects

+ Create new project

Create a new project

Enter the name for your new project: Portenta-Image Classification

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) / Motion-Project PUBLIC

MJRoBot (Marcelo Rovai) / video_tinyml_raw

MJRoBot (Marcelo Rovai) / Pico_Motion_Detection PUBLIC

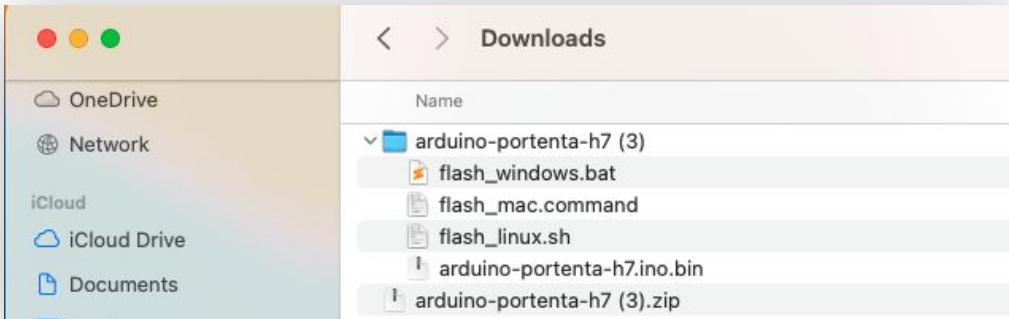
MJRoBot (Marcelo Rovai) / oi_rovis_kws_meetup

MJRoBot (Marcelo Rovai) / ECG_Unifei PUBLIC

QR code

Connecting with EI

1. Put the Portenta H7n on Boot mode (press the button twice)
2. Download the FW from the Edge Impulse Site
3. Open the zip file on your computer and select the uploader related to your OS



```
marcelo_rovai — flash_mac.command — 108x53

Last login: Fri Aug 25 13:02:27 on ttys000
/Users/marcelo_rovai/Downloads/arduino-portenta-h7\ \(3\)/flash_mac.command ; exit;
(base) marcelo_rovai@Marcelos-MacBook-Pro ~ % /Users/marcelo_rovai/Downloads/arduino-portenta-h7\ \(3\)/flash_mac.command ; exit;
You're using an untested version of Arduino CLI, this might cause issues (found: 0.31.0, expected: 0.18.x)
Finding Arduino Mbed core...
Finding Arduino Mbed OK
Finding Arduino Portenta H7...
Finding Arduino Portenta H7 OK
Flashing board...
dfu-util 0.10-dev

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2021 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Warning: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release
Opening DFU capable USB device...
Device ID 2341:035b
Device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Interface #0 ...
Determining device status...
DFU state(2) = dfuIDLE, status(0) = No error condition is present
DFU mode device DFU version 011a
Device returned transfer size 4096
DfuSe interface name: "Internal Flash"
Downloading element to address = 0x08040000, size = 564896
Erase [=====] 100% 564896 bytes
Erase done.
Download [=====] 100% 564896 bytes
Download done.
File downloaded successfully
Transitioning to dfuMANIFEST state

A new release of Arduino CLI is available: 0.31.0 → 0.34.0
https://arduino.github.io/arduino-cli/latest/installation/#latest-packages

Flashed your Arduino Portenta H7 development board.
To set up your development with Edge Impulse, run 'edge-impulse-daemon'
To run your impulse on your development board, run 'edge-impulse-run-impulse'

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Process completed]
```

Portenta-Vision_Image_Classif X +

studio.edgeimpulse.com/studio/275395/dsp/image/3

EDGE IMPULSE MJRoBot (Marcelo Rovai) / Portenta-Vision_Image_Classification

#1 ▾ Click to set a description for this version

Parameters Generate features

Raw data Show: All labels 00015 (periquito)

Raw features 0x75361b, 0x763818, 0x773c17, 0x784014, 0x784214, 0x7e441e, 0x854623, 0x864623, 0x834c27, 0x825...

Parameters

Image Color depth Grayscale Save parameters

DSP result Image

Processed features 0.2736, 0.2780, 0.2879, 0.2970, 0.3016, 0.3177, 0.3327, 0.3339, 0.3460, 0.3522, 0.3583, 0.3628...

On-device performance Processing time 1 ms. Peak RAM usage 4 KB

© 2023 EdgeImpulse Inc. All rights reserved

Portenta-Vision_Image_Classif X +

studio.edgeimpulse.com/studio/275395/deployment-view

EDGE IMPULSE MJRoBot (Marcelo Rovai) / Portenta-Vision_Image_Classification

Dashboard Devices Data acquisition Impulse design Create impulse Image Transfer learning EON Tuner Retrain model Live classification Model testing Versioning Deployment

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.](#)

OpenMV Firmware

SELECTED DEPLOYMENT OpenMV Firmware

Firmware binary that includes your model and runs on OpenMV cameras.

MODEL OPTIMIZATIONS

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

Quantized (int8)

Selected ✓

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	1 ms.	52 ms.	53 ms.
RAM	4.0K	320.2K	320.2K
FLASH	-	284.4K	-
ACCURACY			-

Unoptimized (float32)

Select

	IMAGE	TRANSFER LEARNING	TOTAL
LATENCY	1 ms.	212 ms.	213 ms.
RAM	4.0K	1.1M	1.1M
FLASH	-	471.3K	-
ACCURACY			94.12%

To compare model accuracy, run model testing for all available optimizations. [Run model testing](#)

Estimate for Arduino Portenta H7 (Cortex-M7 480MHz) - [Change target](#)

Build

Latest build

v12 (C++ library)
Today, 17:04:40 [View docs](#)

Run this model

Scan QR code or launch in browser to test your prototype

QR code

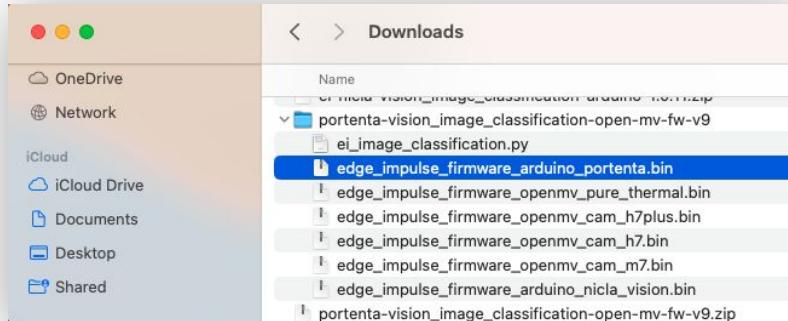
Launch in browser

?

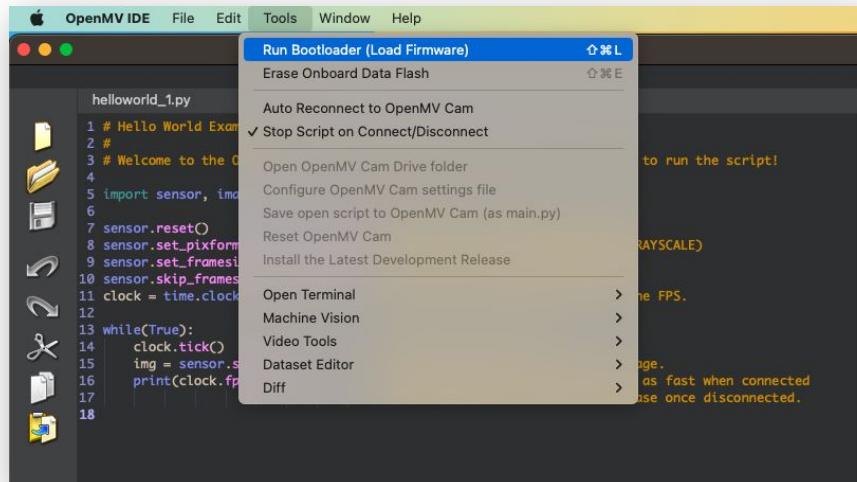
Deploying model on OpenMV IDE

1. On your computer, you will find a ZIP file.

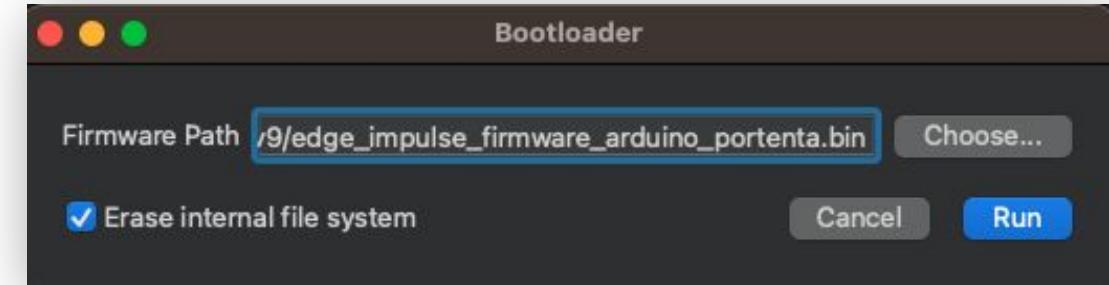
Open it:



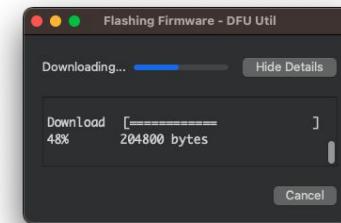
2. Use the Bootloader tool on OpenMV IDE to load it on your board:



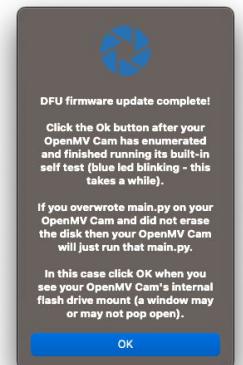
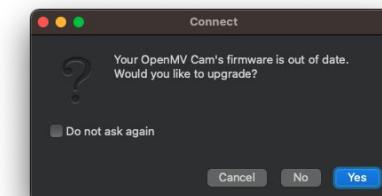
3. Select the appropriate file (.bin for Portenta) and press [Run]:



4. After the download is finished, press OK:



5. If a message saying that the FW is outdated, DO NOT UPGRADE



```
nicla_image_classification_LED.py*  ⇧ ×
1 # Marcelo Roval - NICLA Vision - Image Classification with LEDs
2 # Adapted from Edge Impulse - OpenMV Image Classification Example
3 # @24Aug23
4
5 import sensor, image, time, os, tf, uos, gc, pyb
6
7 ledRed = pyb.LED(1)
8 ledGre = pyb.LED(2)
9 ledBlu = pyb.LED(3)
10
11 sensor.reset()          # Reset and initialize the sensor.
12 sensor.set_pixformat(sensor.GRAYSCALE) # Set pixel format to RGB565 (or GRayscale)
13 sensor.set_framesize(sensor.QVGA)       # Set frame size to QVGA (320x240)
14 sensor.set_windowing((240, 240))       # Set 240x240 window.
15 sensor.skip_frames(time=2000)           # Let the camera adjust.
16
17 net = None
18 labels = None
19
20 ledRed.off()
21 ledGre.off()
22 ledBlu.off()
23
24 try:
25     # Load built in model
26     labels, net = tf.load_builtin_model('trained')
27 except Exception as e:
28     raise Exception(e)
29
30 clock = time.clock()
31
32
33 def setLEDs(max_lbl):
34
35     if max_lbl == 'uncertain':
36         ledRed.on()
37         ledGre.off()
38         ledBlu.off()
39
40     if max_lbl == 'periquito':
41         ledRed.off()
42         ledGre.on()
43         ledBlu.off()
44
45     if max_lbl == 'robot':
46         ledRed.off()
47         ledGre.off()
48         ledBlu.on()
49
50     if max_lbl == 'background':
51         ledRed.off()
52         ledGre.off()
53         ledBlu.off()
54
55
56 while(True):
```

You can run the same code developed for the Nicla, only Set pixel format from RGB565 to GRAYSCALE.

sensor.set_pixformat(sensor.GRAYSCALE)

nicla_image_classification_LED.py* x

```

48     ledBlu.on()
49
50     if max_lbl == 'background':
51         ledRed.off()
52         ledGre.off()
53         ledBlu.off()
54
55
56 while(True):
57     clock.tick() # Starts tracking elapsed time.
58
59     img = sensor.snapshot()
60
61     # default settings just do one detection... change them to search the image...
62     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
63         fps = clock.fps()
64         lat = clock.avg()
65
66         print("*****\nPrediction:")
67         img.draw_rectangle(obj.rect())
68         # This combines the labels and confidence values into a list of tuples
69         predictions.list = list(zip(labels, obj.output()))
70
71         max_val = predictions.list[0][1]
72         max_lbl = 'background'
73         for i in range(len(predictions.list)):
74             val = predictions.list[i][1]
75             lbl = predictions.list[i][0]
76             print("%s = %f" % (predictions.list[i][0], predictions.list[i][1]))
77
78             if val > max_val:
79                 max_val = val
80                 max_lbl = lbl
81
82         # Print label and turn on LED with the highest probability
83         if max_val < 0.5:
84             max_lbl = 'uncertain'
85
86         setLEDs(max_lbl)
87
88
89         #print("%s with a prob of %f" % (max_lbl, max_val))
90         print("{} with a prob of {:.2f}.".format(max_lbl, max_val))
91         # print(clock.fps(), "fps")
92         print("FPS: {:.2f} fps => latency: {:.0f} ms".format(fps, lat))
93
94         # Draw label with highest probability to image viewer
95         img.draw_string(
96             10, 10,
97             max_lbl + "\n{:.2f}".format(max_val),
98             mono_space = False,
99             scale=2
100 )
101
102

```

Serial Terminal |

```

*****
Prediction:
periquito with a prob of 0.91
FPS: 14.31 fps => latency: 70 ms

```

Search Results Serial Terminal

nicla_image_classification_LED.py* x

helloworld_LEDpy x nicla_image_classification_LED.py* x

Line: 86, Col: 21 Frame Buffer Record Zoom Disable

periquito
0.91

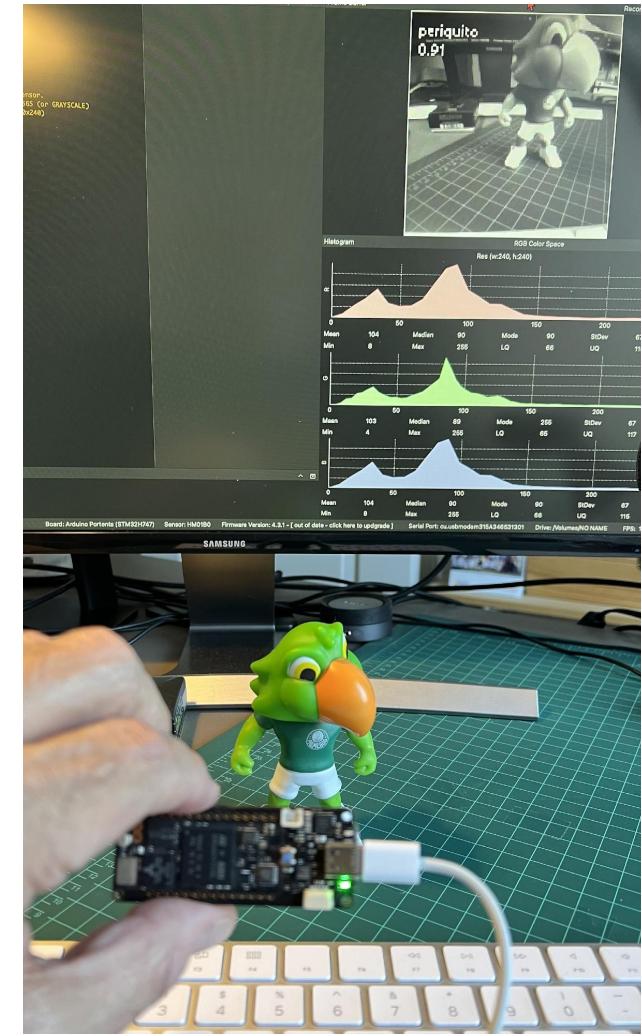
Histogram RGB Color Space Res (w:240, h:240)

Mean: 115 Median: 82 Mode: 255 StDev: 75
Min: 0 Max: 255 LQ: 58 UQ: 173

Mean: 115 Median: 85 Mode: 255 StDev: 75
Min: 0 Max: 255 LQ: 61 UQ: 170

Mean: 115 Median: 82 Mode: 255 StDev: 75
Min: 0 Max: 255 LQ: 58 UQ: 173

Board: Arduino Portenta (STM32H747) Sensor: HM01B0 Firmware Version: 4.3.1 - [out of date - click here to upgrade] Serial Port: cu.usbmodem315A346531301 Drive: /Volumes/NO NAME FPS: 14,3



nicla_image_classification_LED.py* x

helloworld_LEDpy x nicla_image_classification_LED.py* x

Line: 86, Col: 21 Frame Buffer Record Zoom Disable

```

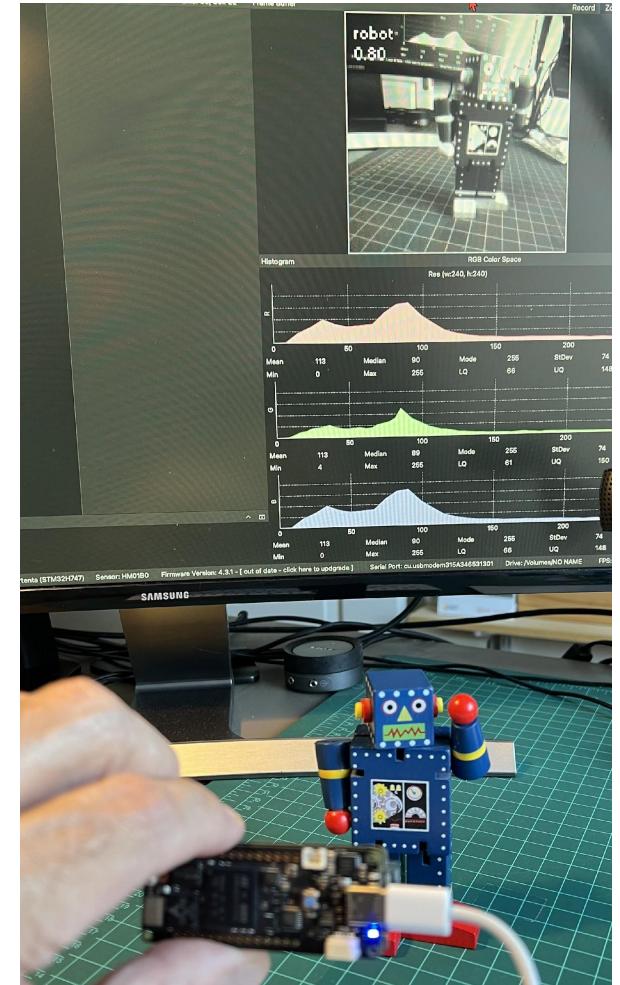
48     ledBlu.on()
49
50     if max_lbl == 'background':
51         ledRed.off()
52         ledGre.off()
53         ledBlu.off()
54
55     while(True):
56         clock.tick() # Starts tracking elapsed time.
57
58         img = sensor.snapshot()
59
60         # default settings just do one detection... change them to search the image...
61         for obj in net.classify(img, min_scale=1.0, scale_mui=0.8, x_overlap=0.5, y_overlap=0.5):
62             fps = clock.fps()
63             lat = clock.avg()
64
65             print("*****\nPrediction:")
66             img.draw_rectangle(obj.rect())
67             # This combines the labels and confidence values into a list of tuples
68             predictions_list = list(zip(labels, obj.output()))
69
70             max_val = predictions_list[0][1]
71             max_lbl = 'background'
72             for i in range(len(predictions_list)):
73                 val = predictions_list[i][1]
74                 lbl = predictions_list[i][0]
75                 #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
76
77                 if val > max_val:
78                     max_val = val
79                     max_lbl = lbl
80
81             # Print label and turn on LED with the highest probability
82             if max_val < 0.5:
83                 max_lbl = 'uncertain'
84
85             setLEDs(max_lbl)
86
87
88             #print("%s with a prob of %f" % (max_lbl, max_val))
89             print("I with a prob of {:.2f}.".format(max_lbl, max_val))
90             # print(clock.fps(), "fps")
91             print("FPS: {:.2f} fps => latency: {:.0f} ms".format(fps, lat))
92
93             # Draw label with highest probability to image viewer
94             img.draw_string(
95                 10, 10,
96                 max_lbl + "\n{:.2f}.".format(max_val),
97                 mono_space = False,
98                 scale=2
99             )
100
101
102
Serial Terminal | 1/2
*****
Prediction:
robot with a prob of 1.00
FPS: 14.31 fps => latency: 70 ms

```

Search Results Serial Terminal

Board: Arduino Portenta (STM32H747) Sensor: HM01B0 Firmware Version: 4.3.1 - [out of date - click here to upgrade] Serial Port: cu.usbmodem315A346531301 Drive:/Volumes/NO NAME FPS: 14,3

The screenshot shows a software development environment with a code editor on the left containing Python code for image classification. The main area displays a camera feed of a small robot, with a bounding box and text overlay indicating the predicted class 'robot' with a confidence of 1.00. Below the video feed are three histograms for the RGB color space, showing the distribution of pixel values for each channel. The histograms include statistical data such as Mean, Median, Mode, and Standard Deviation.



helloworld_LEDpy x ncia_image_classification_LED.py*

Line: 86, Col: 21 Frame Buffer Record Zoom Disable

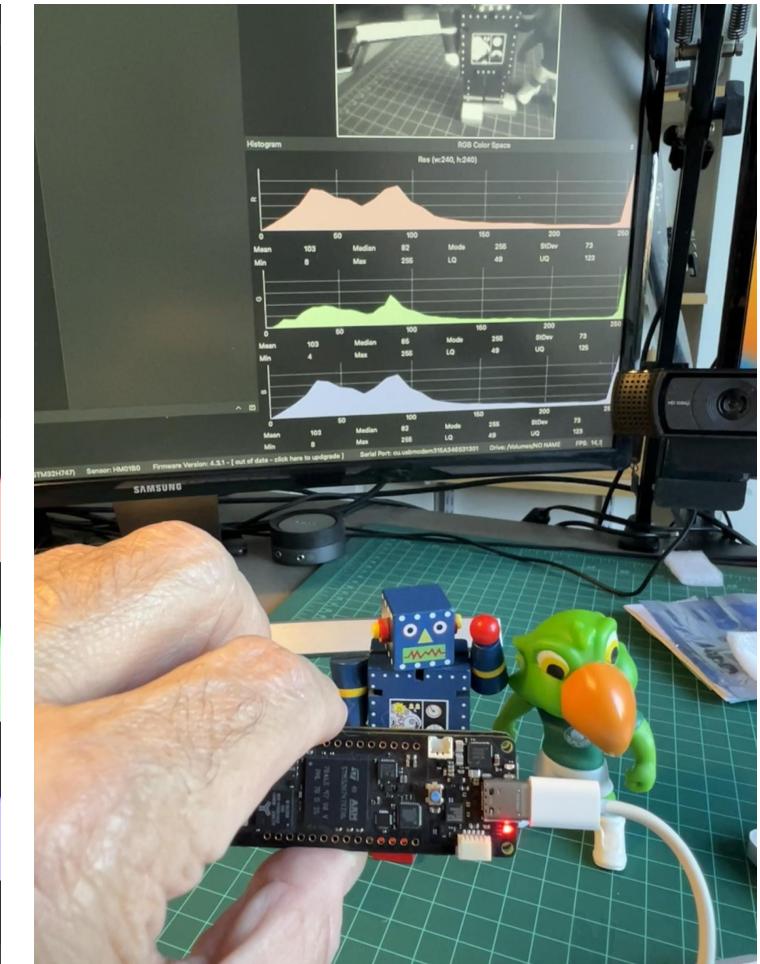
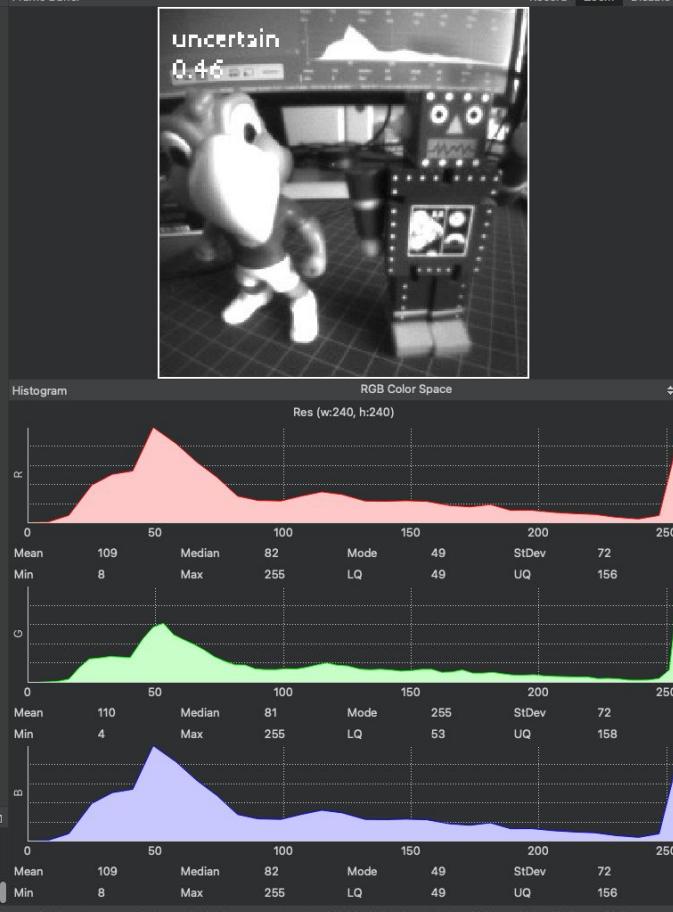
```

48     ledBlu.on()
49
50     if max_lbl == 'background':
51         ledRed.off()
52         ledGre.off()
53         ledBlu.off()
54
55
56     while(True):
57         clock.tick() # Starts tracking elapsed time.
58
59     img = sensor.snapshot()
60
61     # default settings just do one detection... change them to search the image...
62     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
63         fps = clock.fps()
64         lat = clock.avg()
65
66         print("*****\nPrediction:")
67         img.draw_rectangle(obj.rect())
68         # This combines the labels and confidence values into a list of tuples
69         predictions_list = list(zip(labels, obj.output()))
70
71         max_val = predictions_list[0][1]
72         max_lbl = 'background'
73         for i in range(len(predictions_list)):
74             val = predictions_list[i][1]
75             lbl = predictions_list[i][0]
76             #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
77
78             if val > max_val:
79                 max_val = val
80                 max_lbl = lbl
81
82         # Print label and turn on LED with the highest probability
83         if max_val < 0.5:
84             max_lbl = 'uncertain'
85
86         setLEDs(max_lbl)
87
88
89         #print("%s with a prob of %f" % (max_lbl, max_val))
90         print("{} with a prob of {:.2f}.".format(max_lbl, max_val))
91         # print(clock.fps(), "fps")
92         print("FPS: {:.2f} fps => latency: {:.0f} ms".format(fps, lat))
93
94         # Draw label with highest probability to image viewer
95         img.drawString(
96             10, 10,
97             max_lbl + "\n{:.2f}.".format(max_val),
98             mono_space = False,
99             scale=2
100        )
101
102
Serial Terminal
*****
Prediction:
uncertain with a prob of 0.46
FPS: 14.31 fps => latency: 70 ms

```

Search Results Serial Terminal

Board: Arduino Portenta (STM32H747) Sensor: HM01B0 Firmware Version: 4.3.1 - [out of date - click here to upgrade] Serial Port: cu.usbmodem315A346531301 Drive: /Volumes/NO NAME FPS: 14,1



```

48     ledBlu.on()
49
50     if max_lbl == 'background':
51         ledRed.off()
52         ledGre.off()
53         ledBlu.off()
54
55
56 while(True):
57     clock.tick() # Starts tracking elapsed time.
58
59     img = sensor.snapshot()
60
61     # default settings just do one detection... change them to search the image...
62     for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
63         fps = clock.fps()
64         lat = clock.avg()
65
66         print("*****\nPrediction:")
67         img.draw_rectangle(obj.rect())
68         # This combines the labels and confidence values into a list of tuples
69         predictions_list = list(zip(labels, obj.output()))
70
71         max_val = predictions_list[0][1]
72         max_lbl = 'background'
73         for i in range(len(predictions_list)):
74             val = predictions_list[i][1]
75             lbl = predictions_list[i][0]
76             #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
77
78             if val > max_val:
79                 max_val = val
80                 max_lbl = lbl
81
82     # Print label and turn on LED with the highest probability
83     if max_val < 0.5:
84         max_lbl = 'uncertain'
85
86     setLEDs(max_lbl)
87
88
89     #print("%s with a prob of %f" % (max_lbl, max_val))
90     print("{} with a prob of {:.2f}".format(max_lbl, max_val))
91     # print(clock.fps(), "fps")
92     print("FPS: {:.2f} fps => latency: {:.0f} ms".format(fps, lat))
93
94     # Draw label with highest probability to image viewer
95     img.draw_string(
96         10, 10,
97         max_lbl + "\n{:.2f}".format(max_val),
98         mono_space = False,
99         scale=2
100    )

```

Serial Terminal

Prediction:
background with a prob of 0.99
FPS: 14.31 fps => latency: 70 ms



Search Results



Serial Terminal

Board: Arduino Portenta (STM32H747)

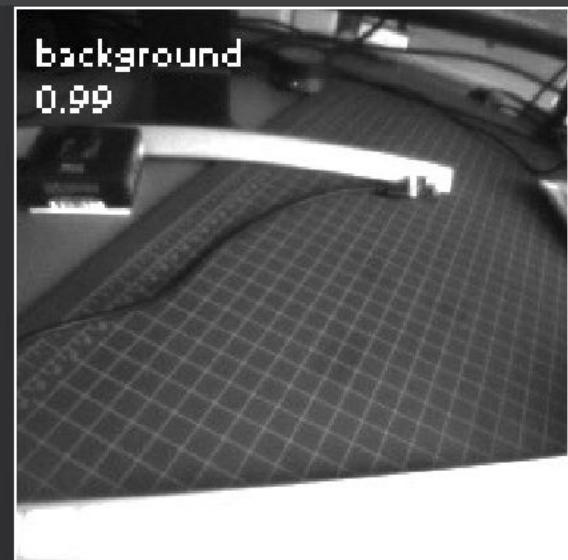
Sensor: HM01B0

Firmware Version: 4.3.1 - [out of date - click here to upgrade]

Serial Port: cu.usbmodem315A346531301

Drive: /Volumes/NO NAME

FPS: 14,3



Mug, or not Mug, that is the question!

EdgeAI made simple - Exploring Image Processing (Image Classification) on microcontrollers with Arduino Portenta, Edge Impulse, and OpenMV

 Intermediate  Full instructions provided  8 hours  4,229

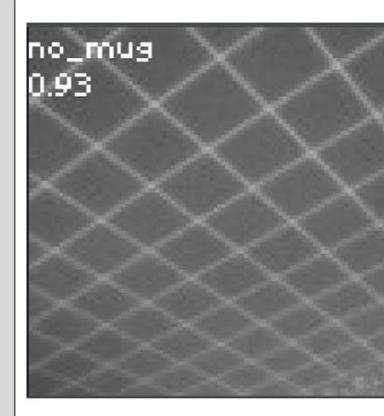


With Arduino Portenta

```
Serial Terminal | ↻ 
mug: 0.988281
no_mug: 0.0117188
FPS: 6.97367
-----
mug: 0.992188
no_mug: 0.0078125
FPS: 6.97379
-----
mug: 0.980469
no_mug: 0.0195313
FPS: 6.9739
```



```
Serial Terminal | ↻ 
mug: 0.078125
no_mug: 0.921875
FPS: 6.97485
-----
mug: 0.0976563
no_mug: 0.902344
FPS: 6.97472
-----
mug: 0.0664063
no_mug: 0.933594
FPS: 6.97458
```

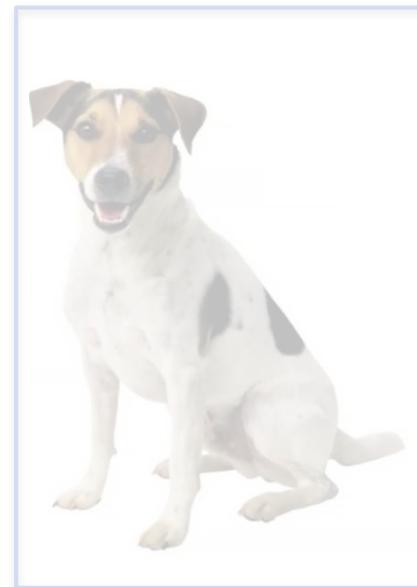


Computer Vision Main Types

Image Classification
(Multi-Class Classification)

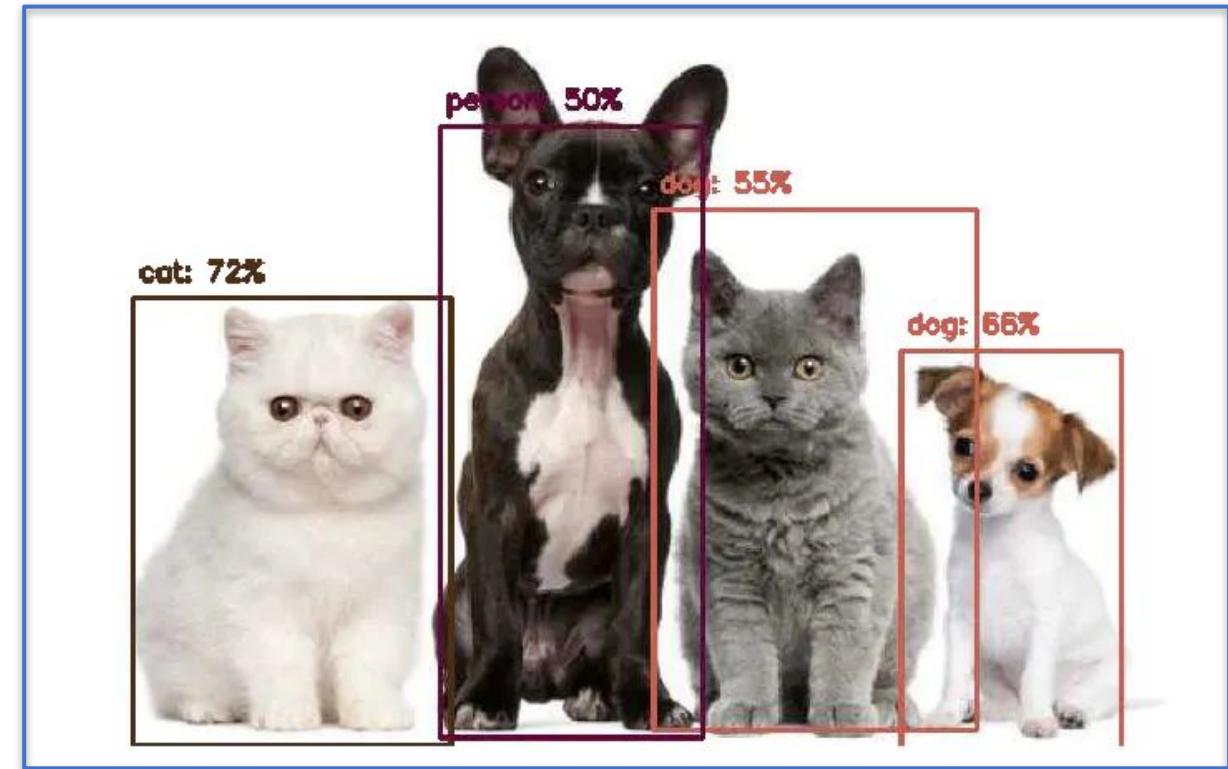


Cat: 70%

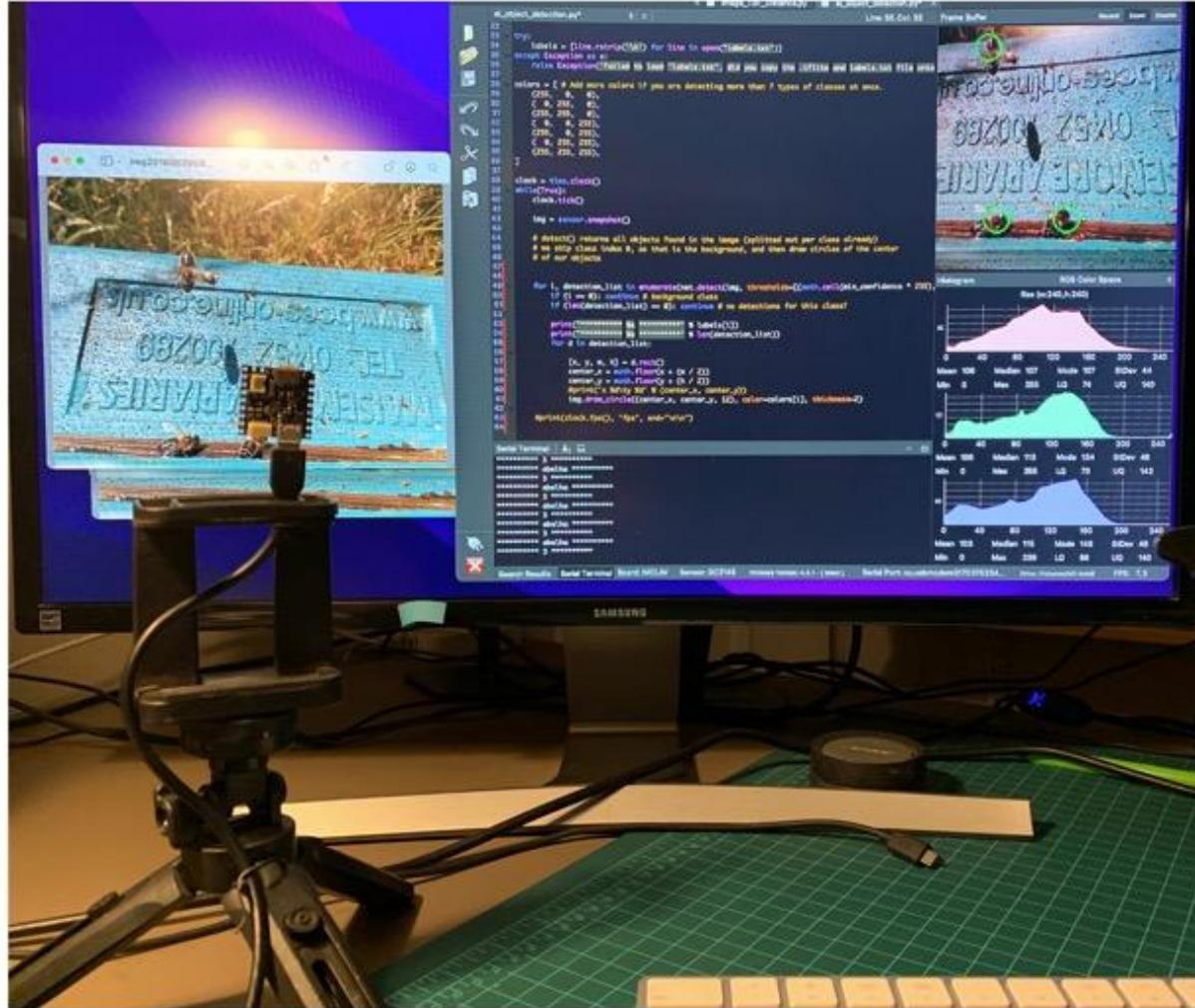


Dog: 80%

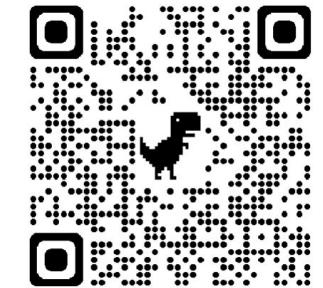
Object Detection
Multi-Label Classification + Object Localization



Detecting Objects using TinyML (FOMO)



MicroPython

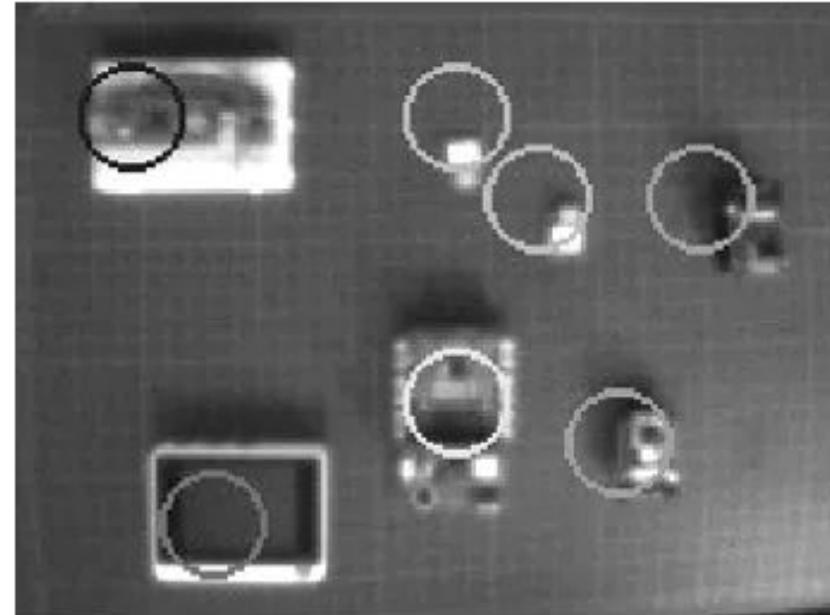
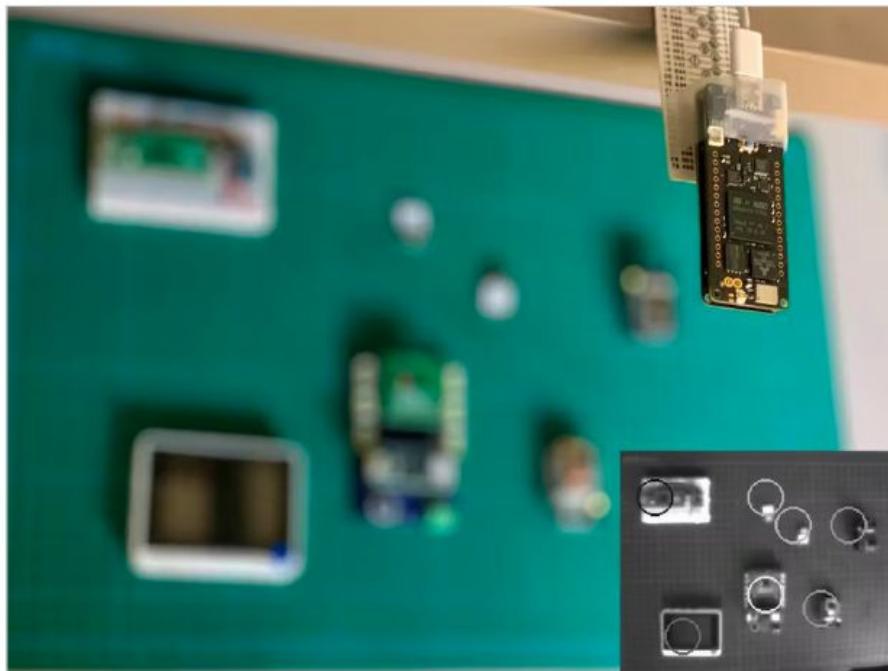


Detecting Objects using TinyML (FOMO)

Where are my tinyML devices?

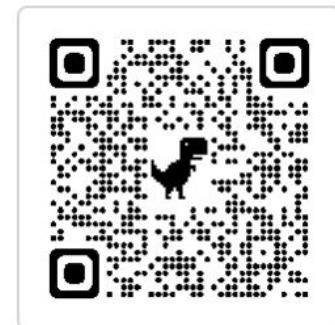
EdgeAI made simple - Exploring Image Processing (Object Detection) on microcontrollers with Arduino Portenta, Edge Impulse FOMO, and OpenMV

Intermediate Full instructions provided 8 hours 3,153



```
***** 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
```

With Arduino Portenta



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.

TinyML4D Show&Tell Presentations

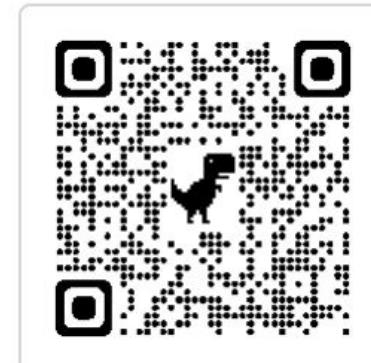
Date	Thread	Video
August 31 st , 2023	TBD	Video here when ready
May 25th, 2023	Thread here	Video here when ready
April 20 th , 2023	Thread here	https://youtu.be/u0M_ljXjDFY
March 30th, 2023	thread here	https://youtu.be/UQ0I-SwBwUY
February 23rd, 2023	thread here	https://youtu.be/BAEdil7X68Y
January 26th, 2023	thread here 17	https://youtu.be/-0xRZ-5UYUc 9
December 1st, 2022	thread here 2	https://youtu.be/e49pkjnIMIQ 8
October 27th, 2022	thread here 2	https://youtu.be/s8_hKpOWUwY 1

[TinymML4D Academic Network Show and Tell Main Index.](#)

The TinyML4D Academic Network Students should use this form to sign up for the latest presentations.

<https://forms.gle/ic52HZMqVv4pBrkP7> 2

The Show and Tell are typically held via Zoom or Meet at 2 pm UTC on the last Thursday of each month.



Q & A

Thanks



UNIFEI