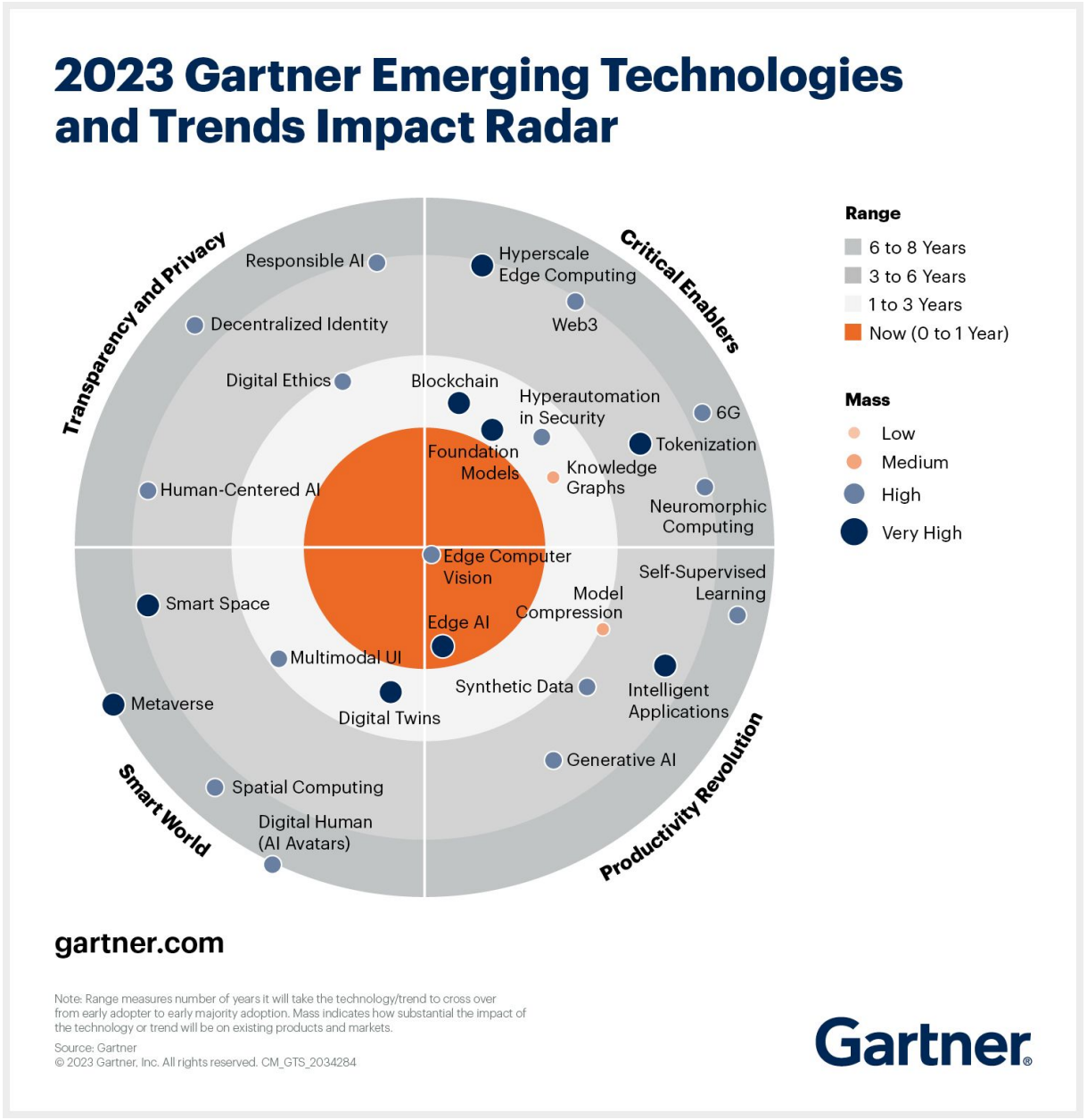
Edge Computer Vision

With Arduino PRO (Portenta/Nicla)

(Material)

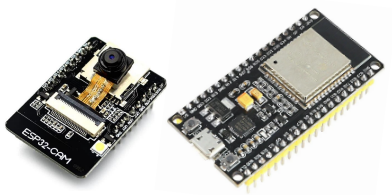
Prof. Marcelo José Rovai
UNIFEI - Federal University of Itajubá, Brazil
TinyML4D Academic Network Co-Chair

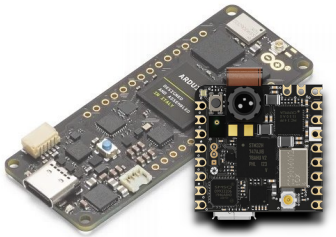
UNIFEI



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

Hardware

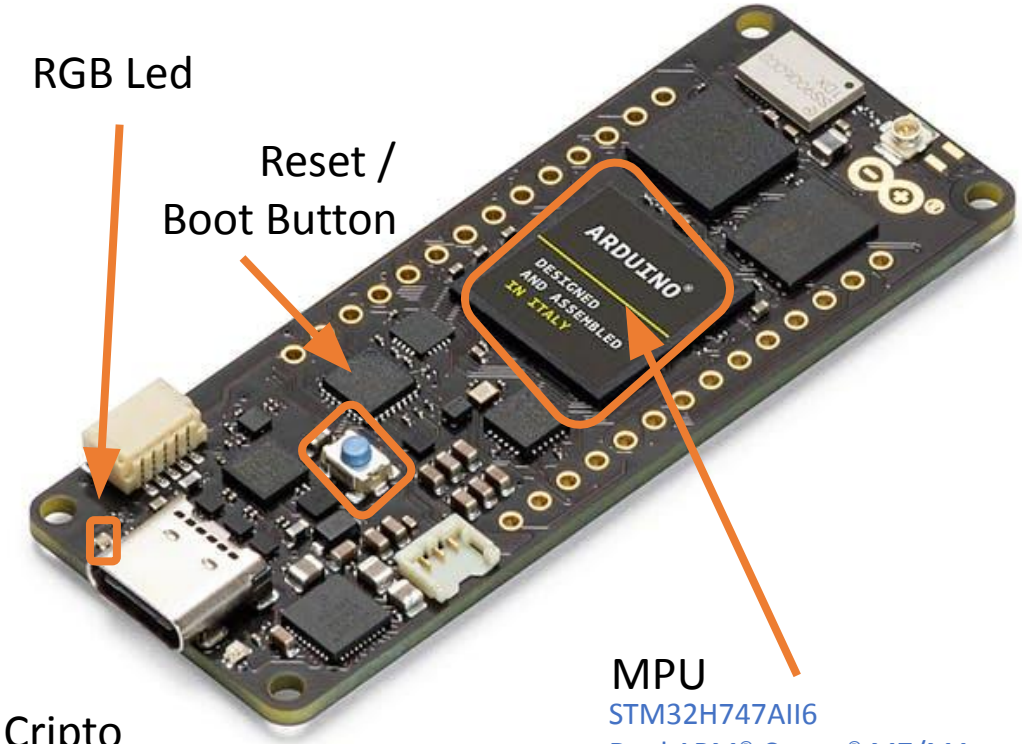


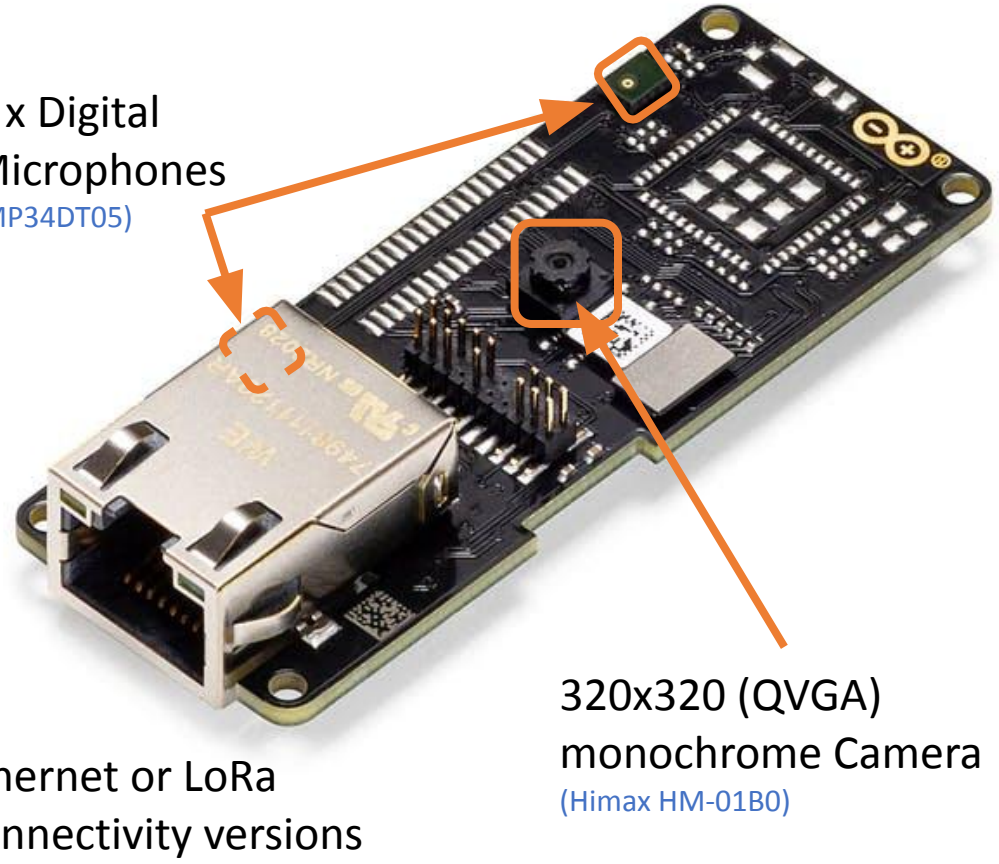


	Raspberry Pico (W)	Arduino Nano Sense	ESP 32	Seeed XIAO Sense / ESP32S3	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F (BLE) Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64 / 240MHz	480/240MHz
RAM	264KB	256KB	520KB (part 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	$	$$$	$	$$	$$$$$

https://media.digikey.com/Resources/Maker/the-original-guide-to-boards-2022.pdf

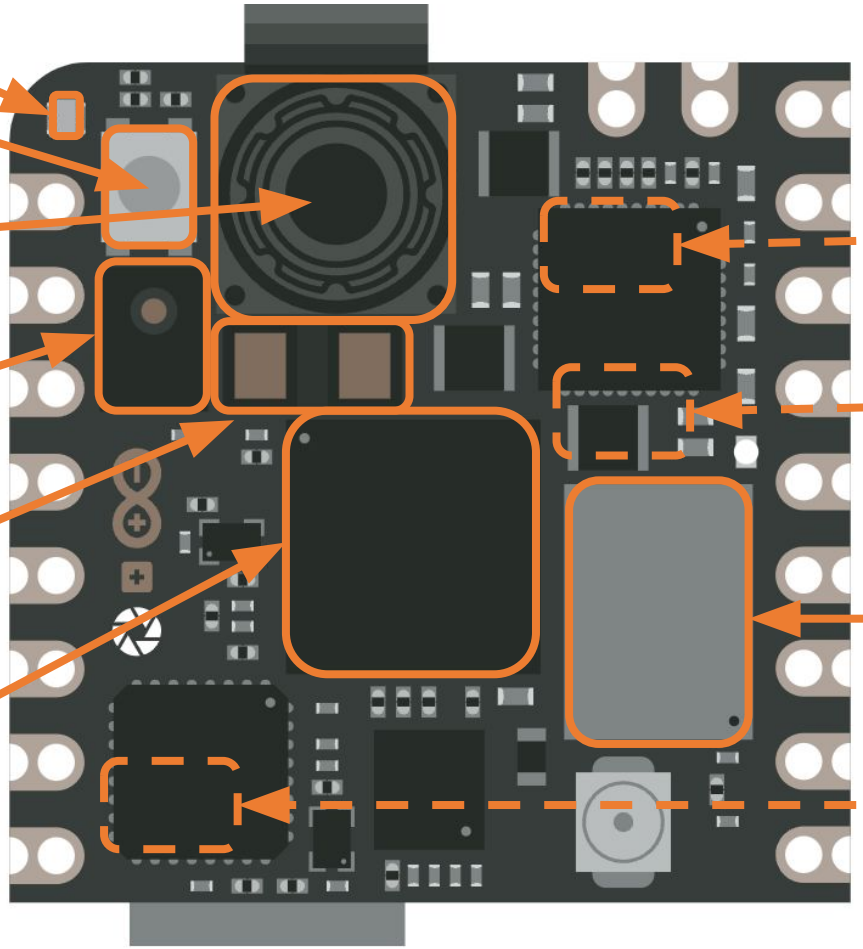
Arduino Portenta H7 + Vision Shield





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

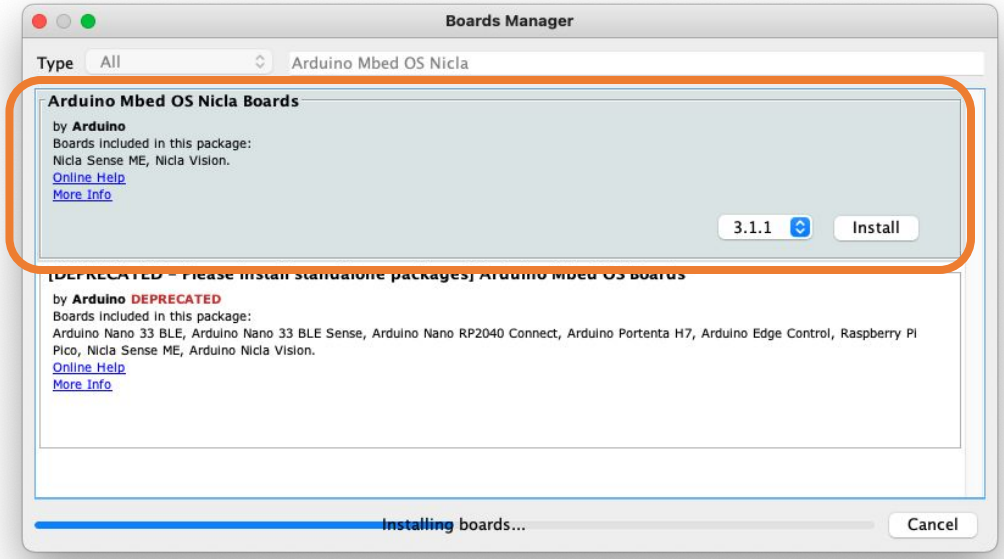
Arduino Nicla Vision



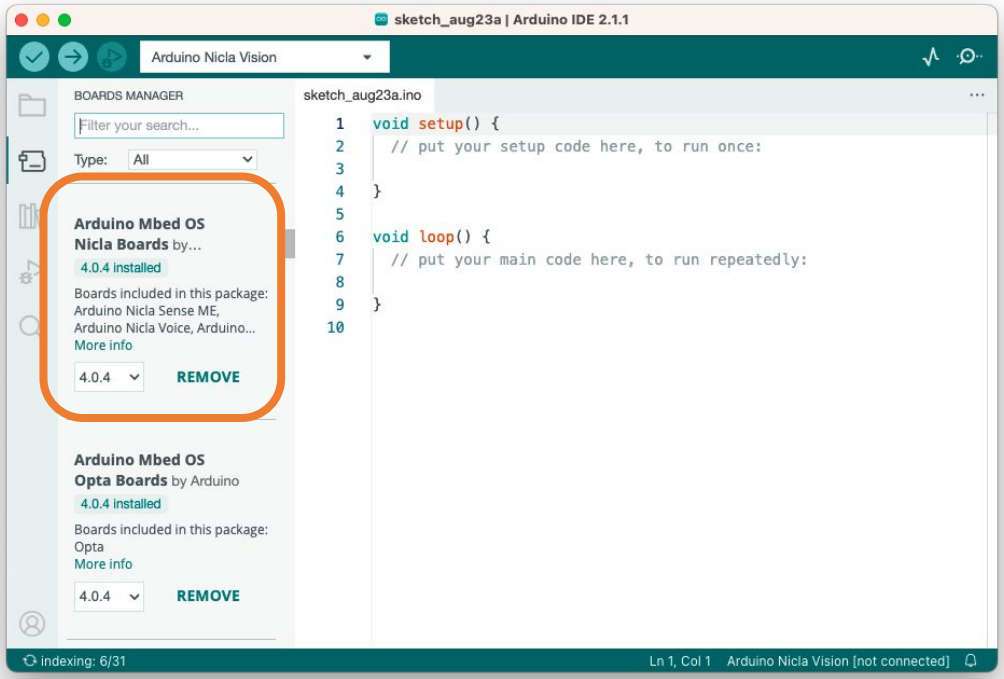
https://docs.arduino.cc/hardware/nicla-vision

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

Nicla Vision - Board Installation



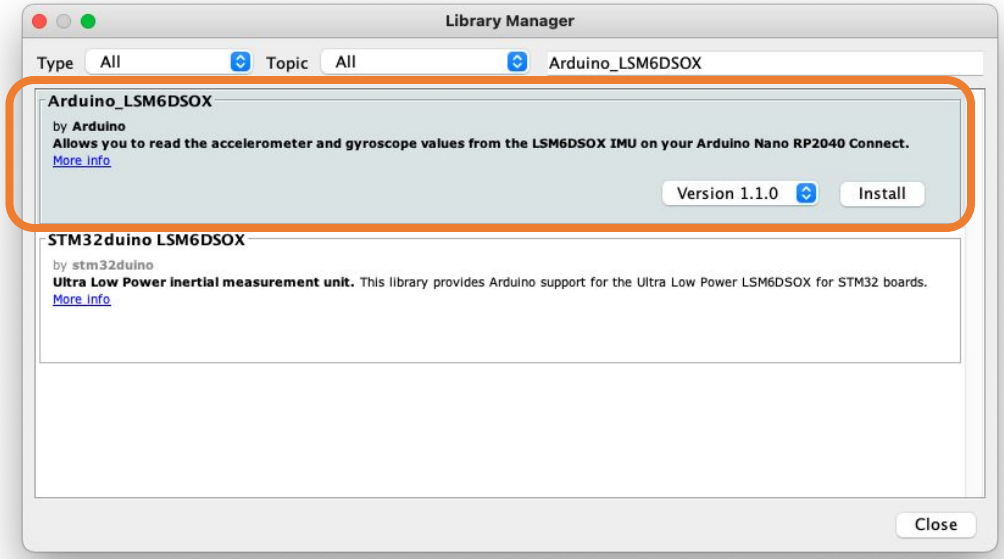
Arduino IDE 1.18

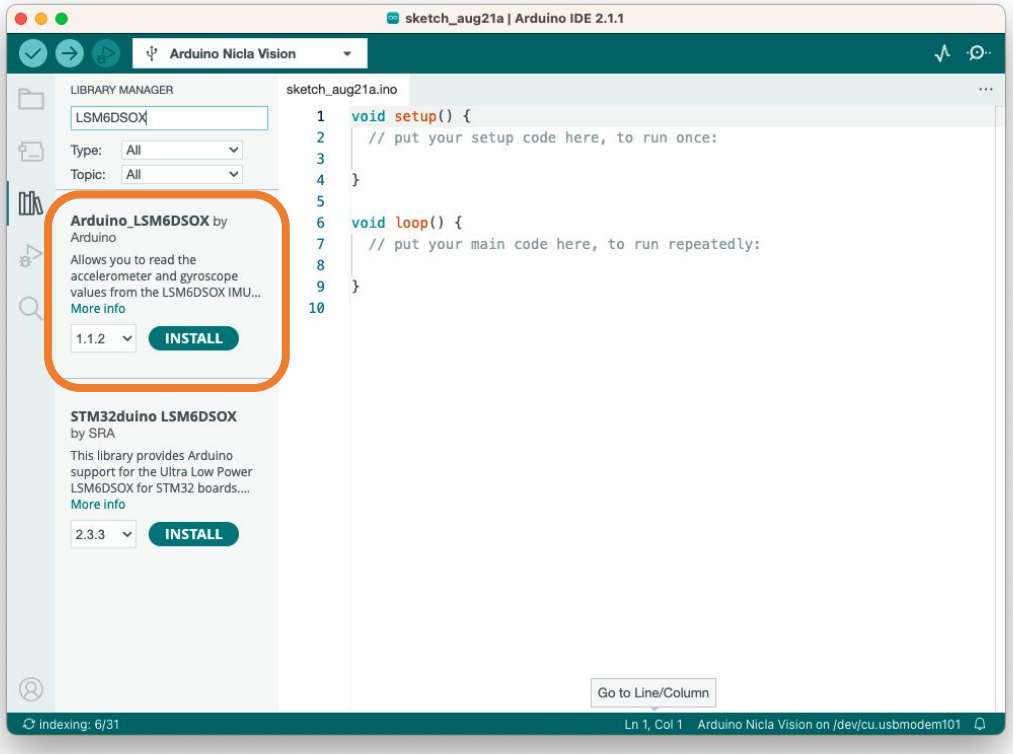


Arduino IDE 2.X

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

IMU (LSM6DSOX) Library Installation



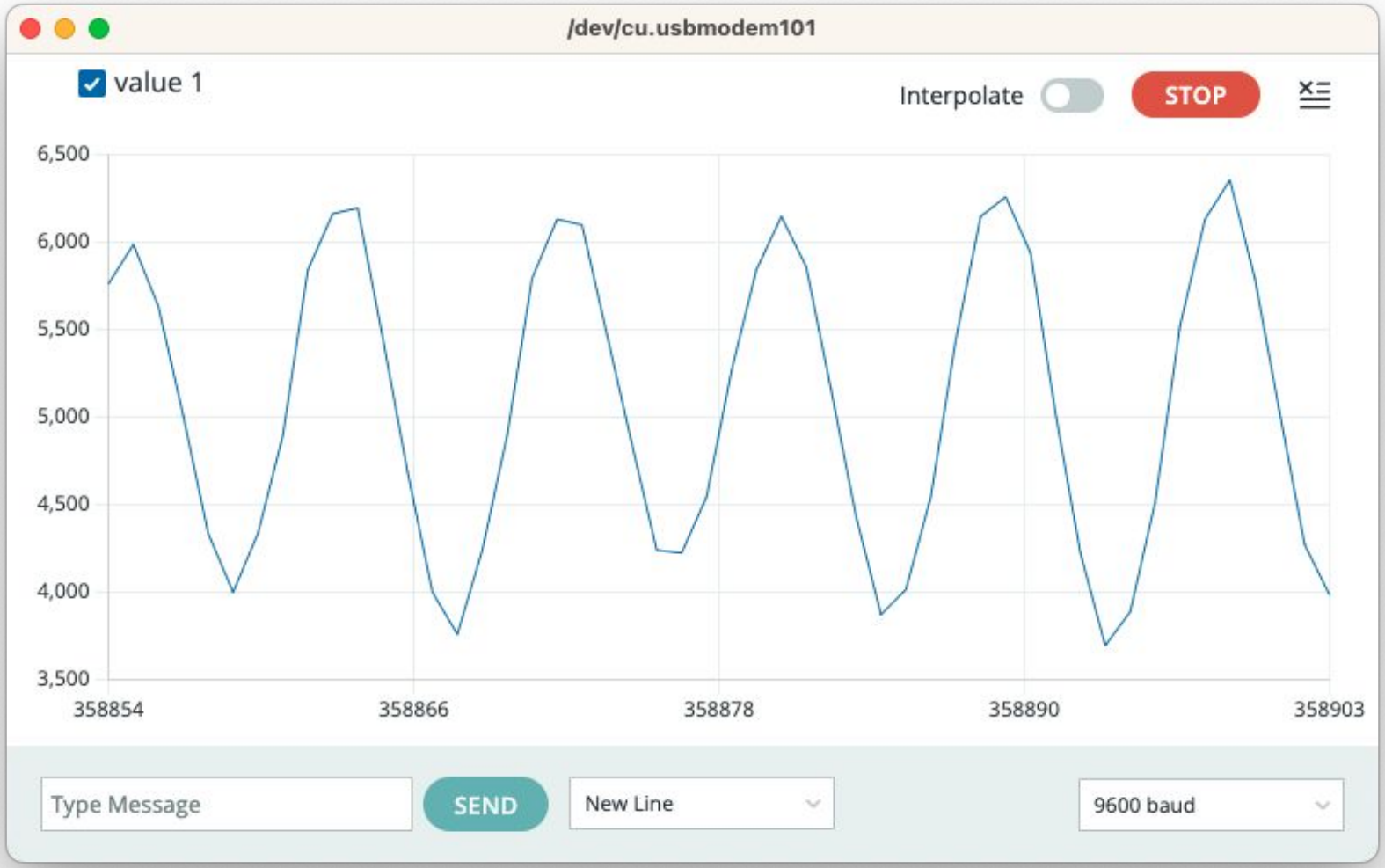


IMU (LSM6DSOX) Test

Go to Examples > Arduino_LSM6DSOX > SimpleAccelerometer

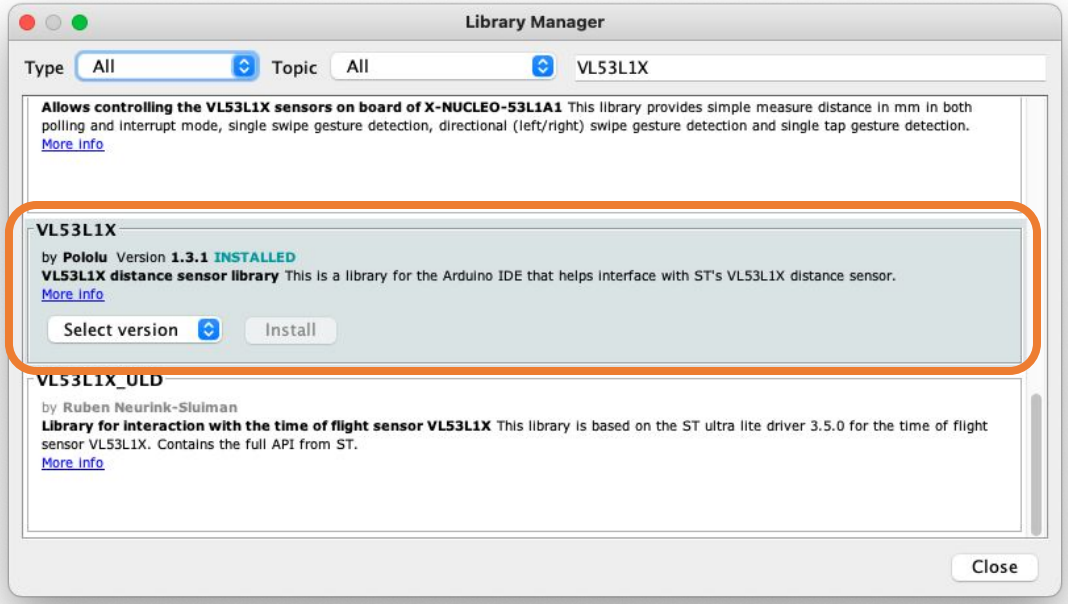
Run accelerometer Test (you can also run Gyro and board temp)

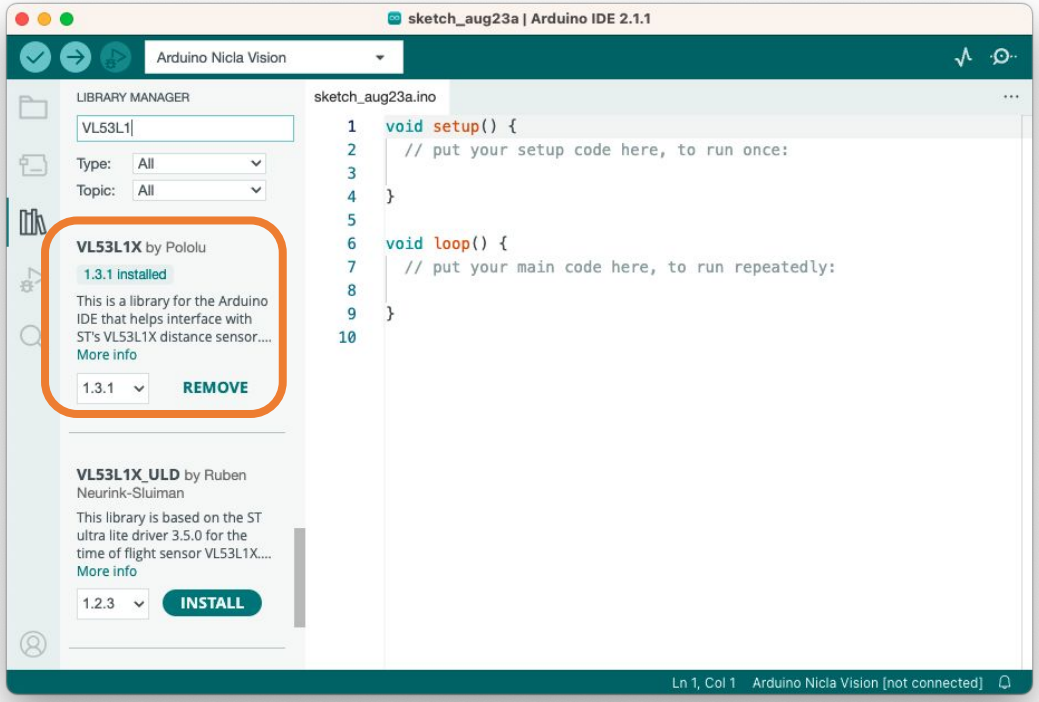
Microphone Test



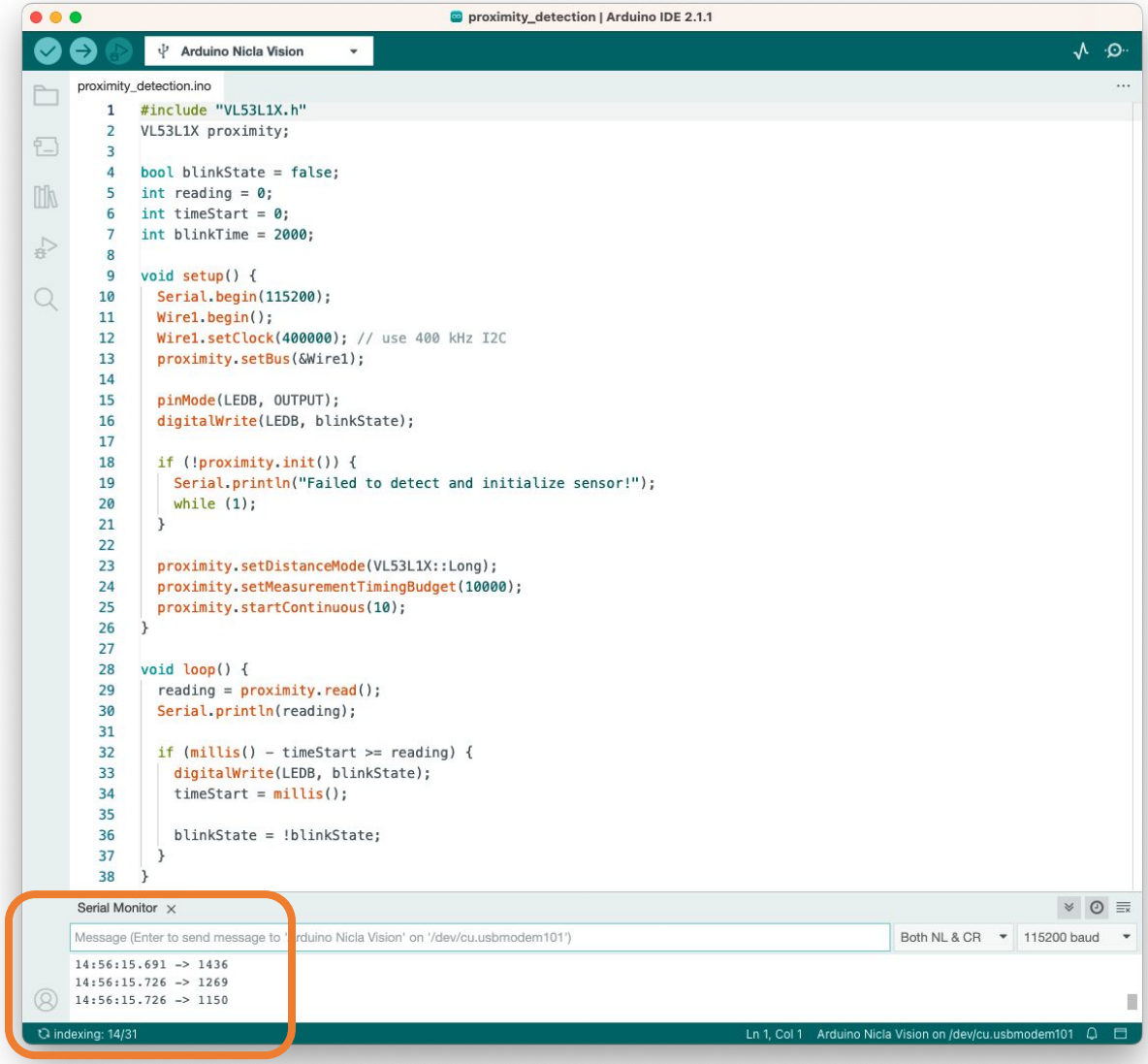
Go to Examples > PDM > PDMSerialPlotter

ToF (VL53L1X) Library Installation





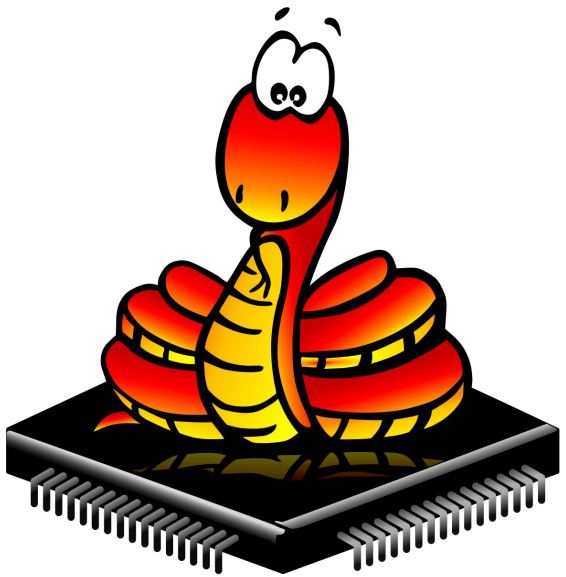
ToF (VL53L1X) Test



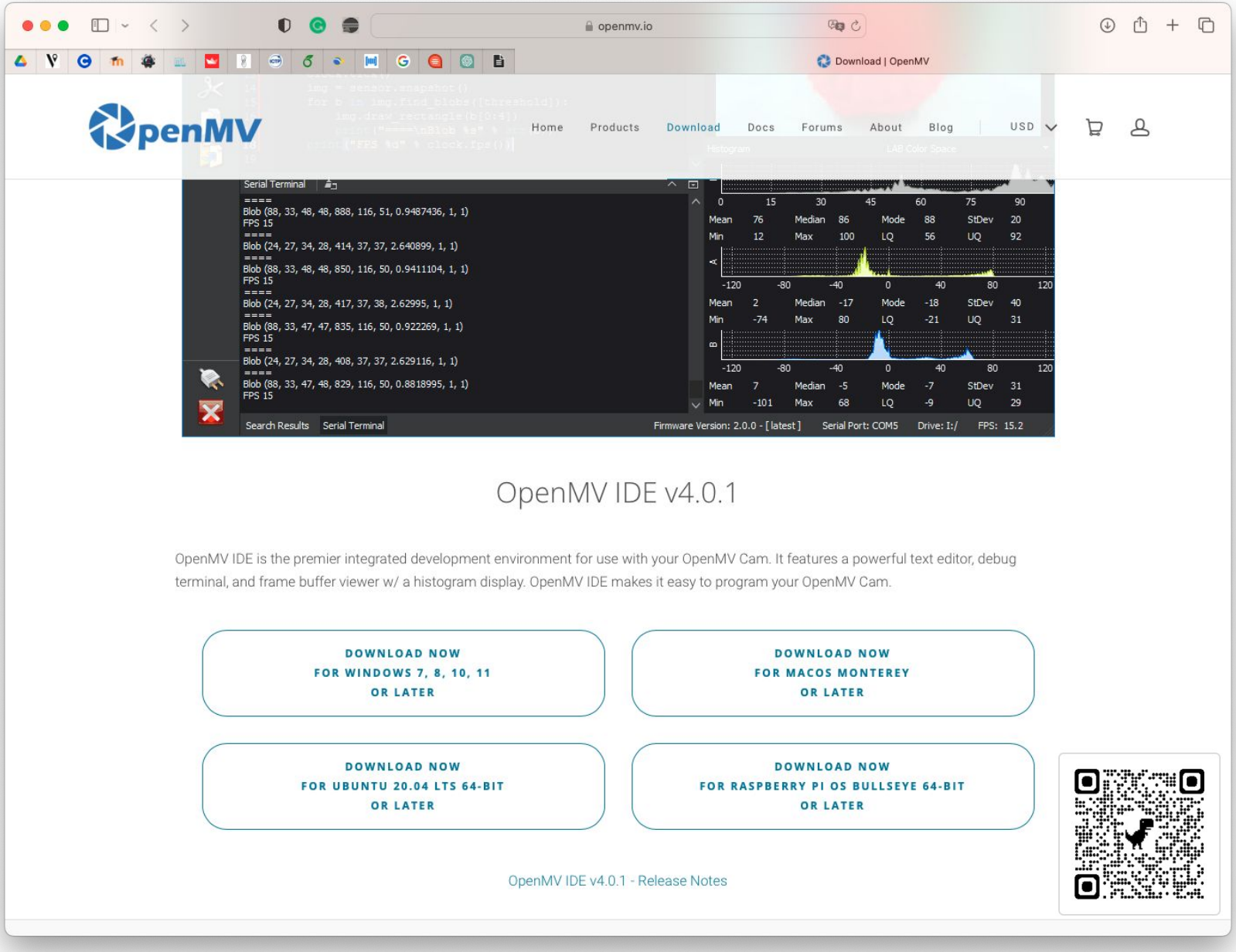
Run the sketch: proximity_detection.ino

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

OpenMV



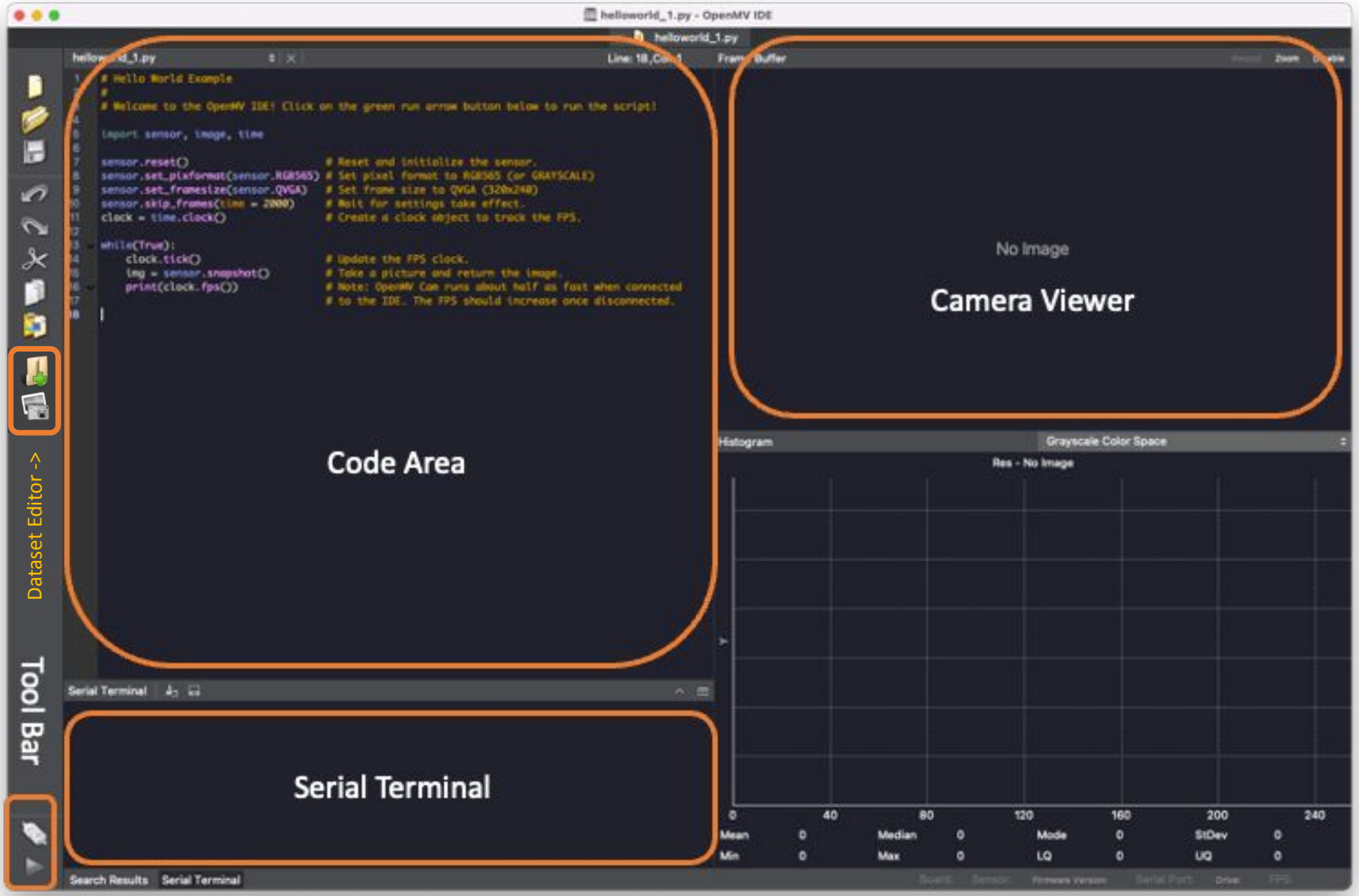
IDE Installation & Sensors 'test



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

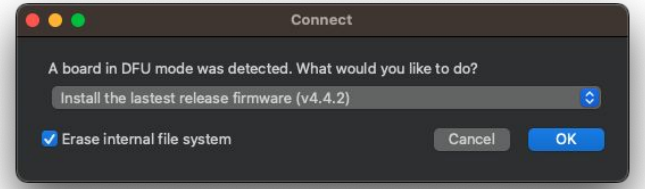
To that, go to your Arduino ID, select the Nicla board and open the sketch on:

Examples > STM_32H747_System > STM_32H747_updateBootloader

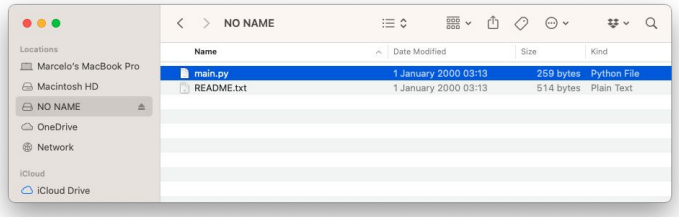

helloworld_1.py - OpenMV IDE
helloworld_1.py
Hello World Example
#
# Welcome to the OpenMV IDE! Click on the green run arrow button below to run the script!
import sensor, image, time
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 2000)
clock = time.clock()
while(True):
clock.tick()
img = sensor.snapshot()
print(clock.fps())
Code Area
No Image
Camera Viewer
Histogram
Grayscale Color Space
Res - No Image
Mean
Median
Mode
StDev
Min
Max
LQ
UQ
Serial Terminal
Serial Terminal
Search Results
Serial Terminal
Dataset Editor ->
Tool Bar

Nicla connection to OpenMV IDE

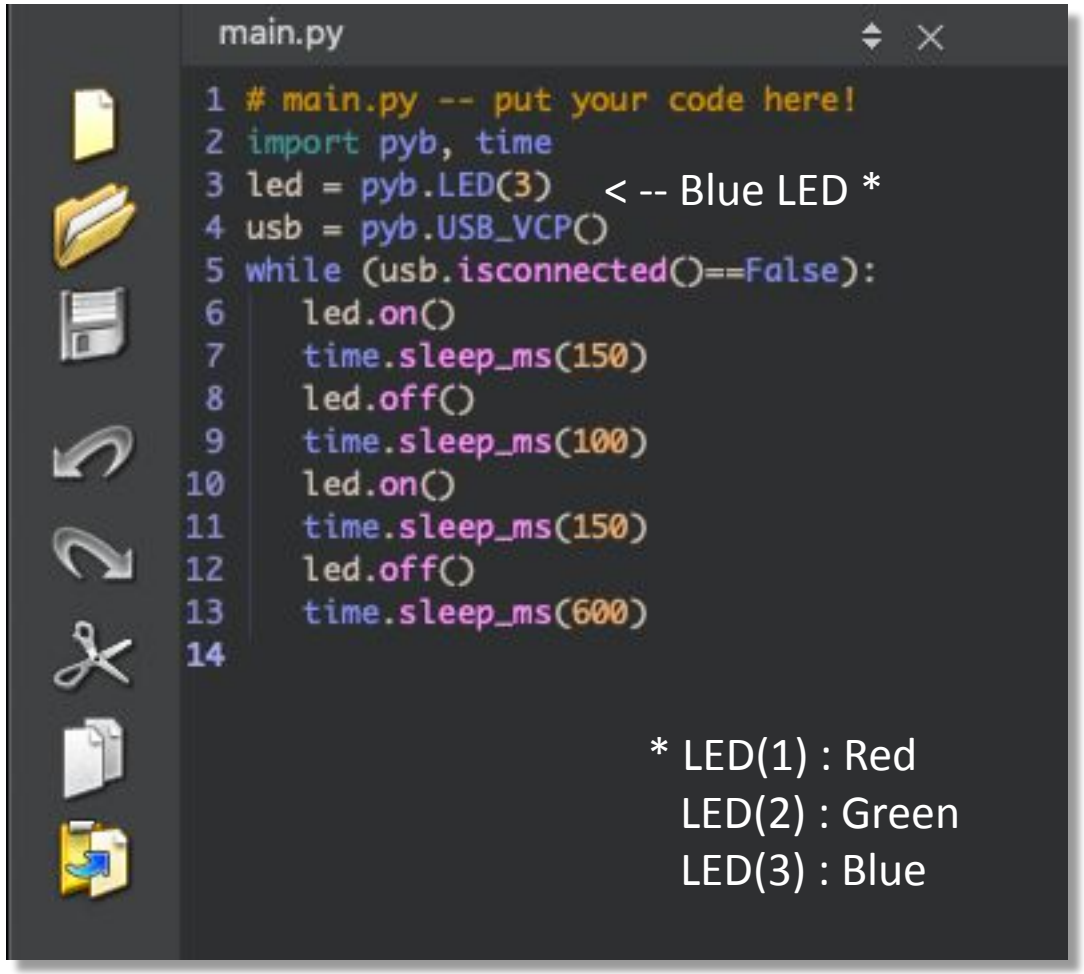
The first time that the Nicla is connected to the IDE, you need to install a FW



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



The Blink MicroPython code



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()

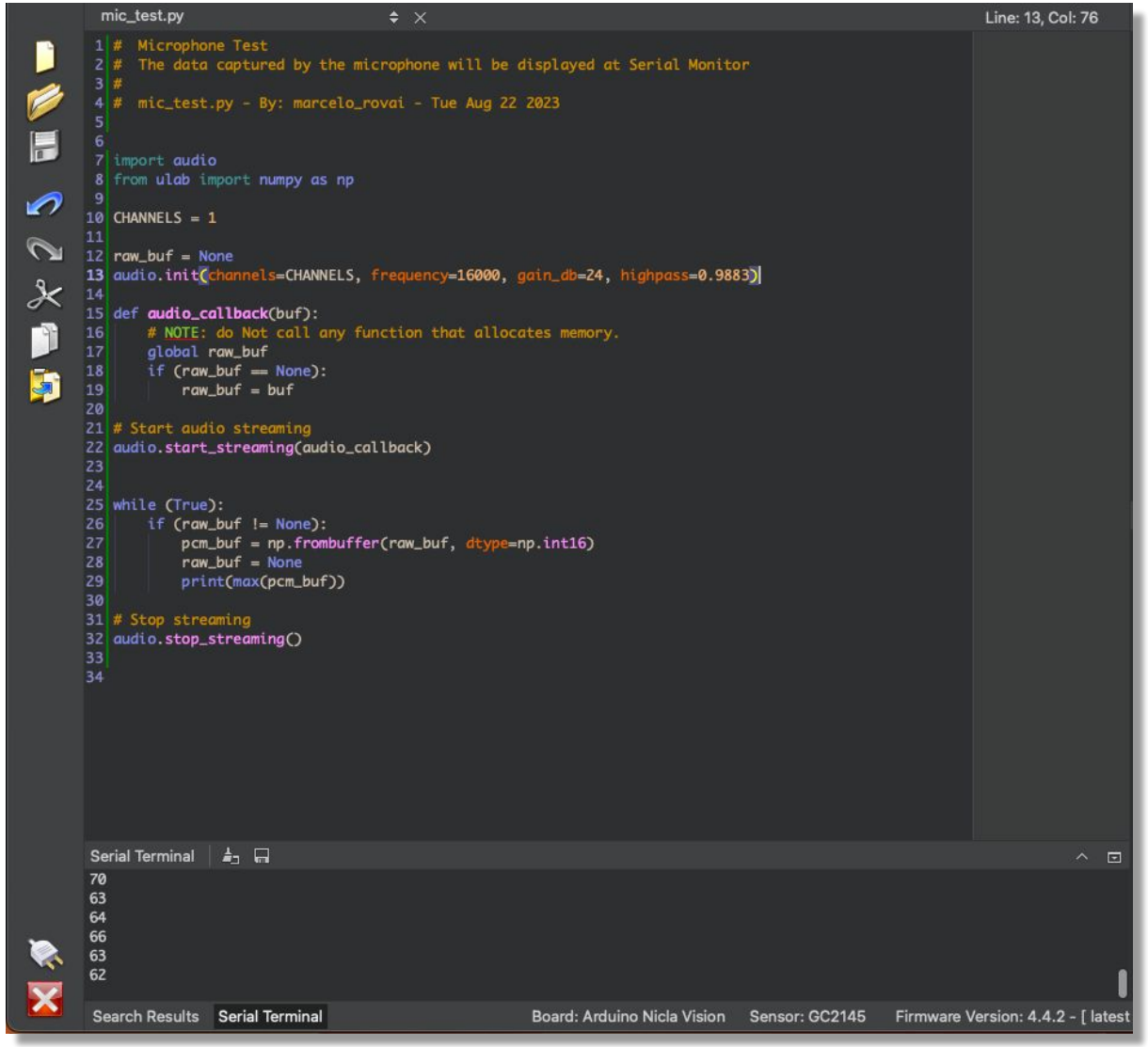
Turn the LED on, to maximum intensity.

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



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**

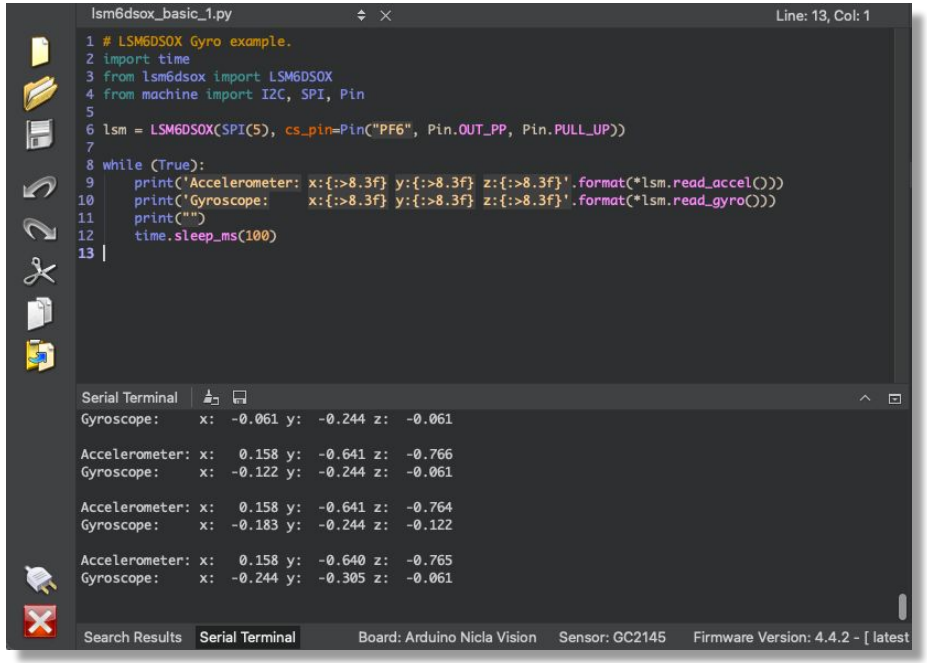
Examples scripts are located in OpenMV IDE under the `Modbus` examples folder.

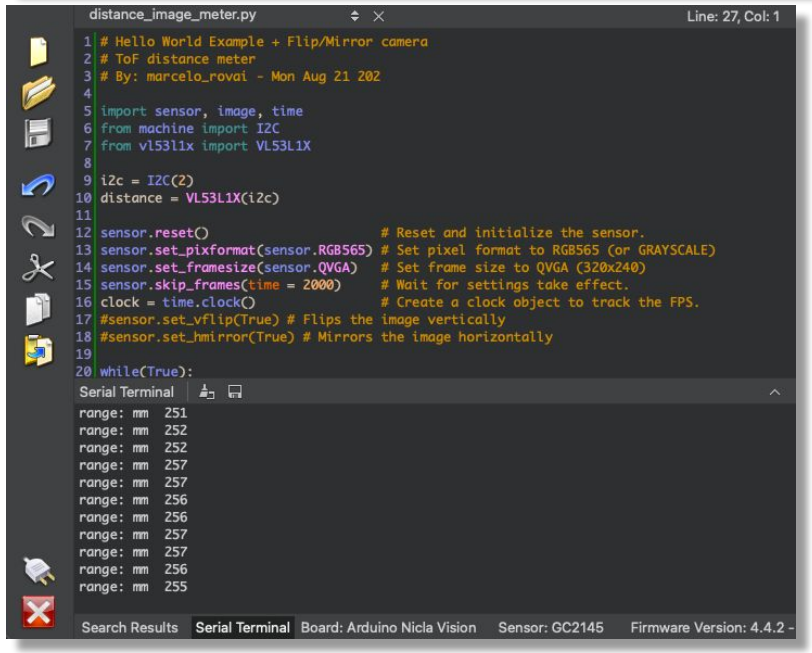
`mqtt` — **mqtt protocol library**

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.



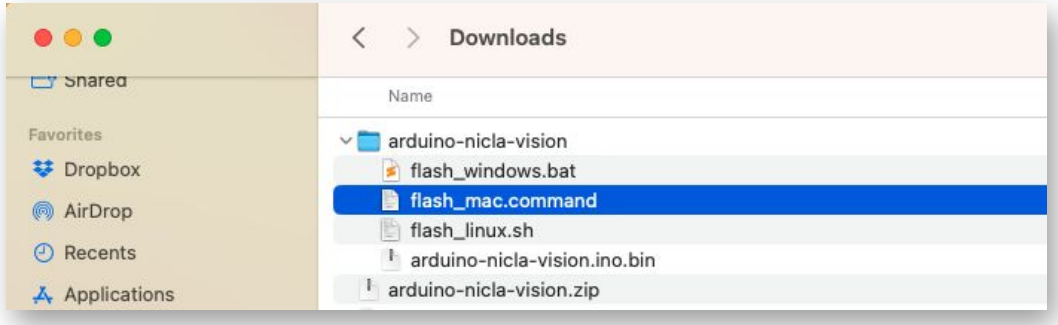


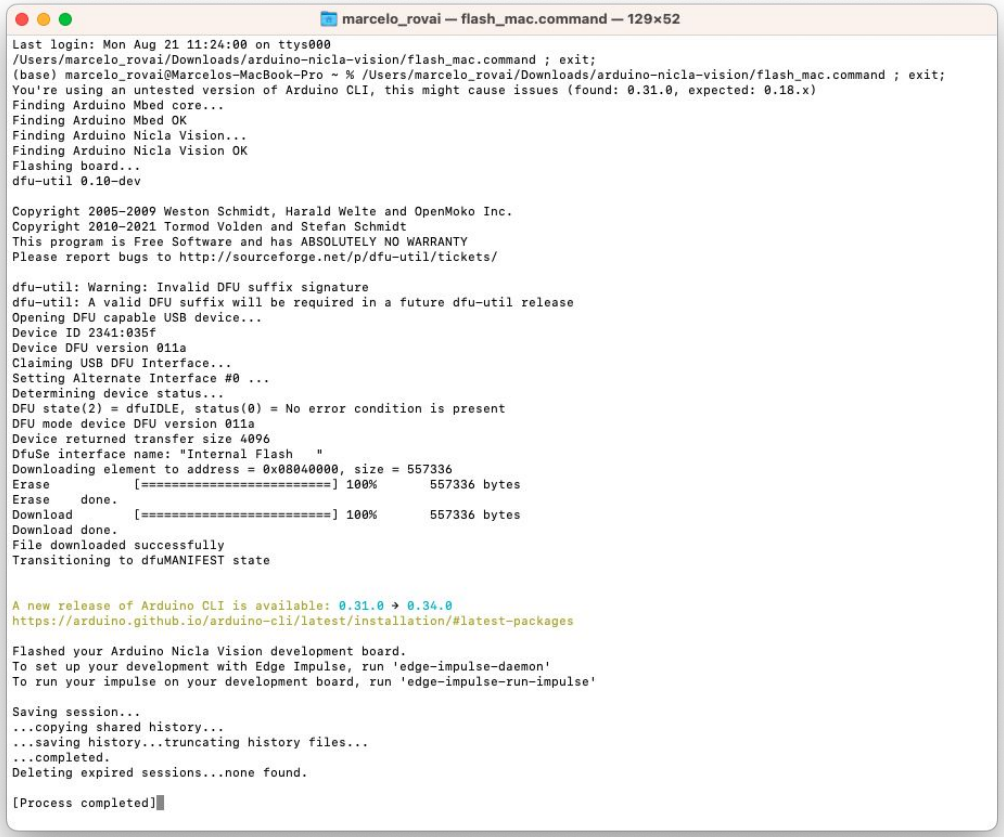
Nicla connection & Sensors' capture data

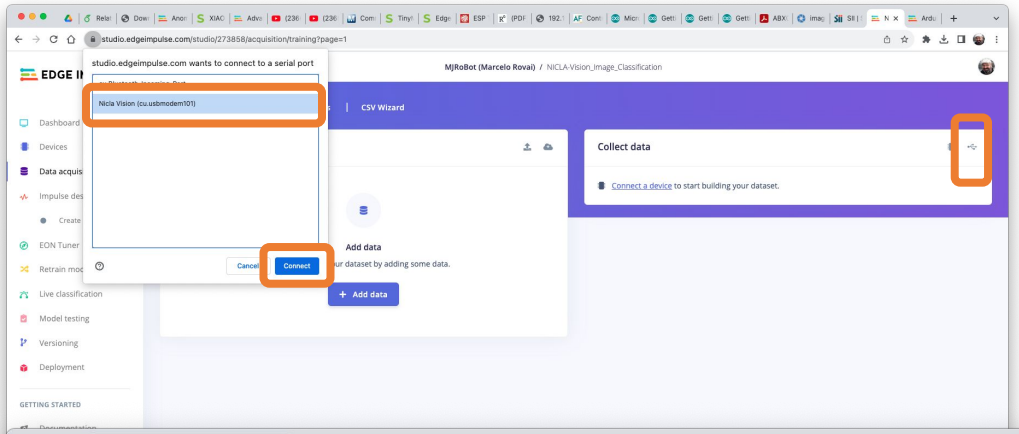
EDGE IMPULSE

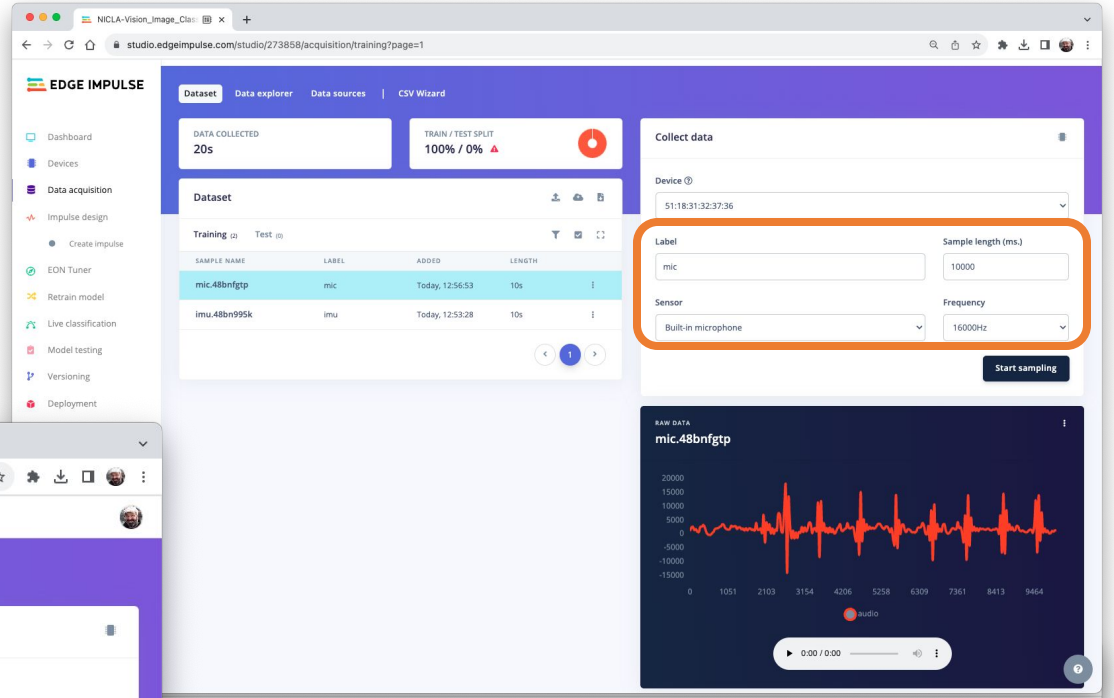
Connecting with EI

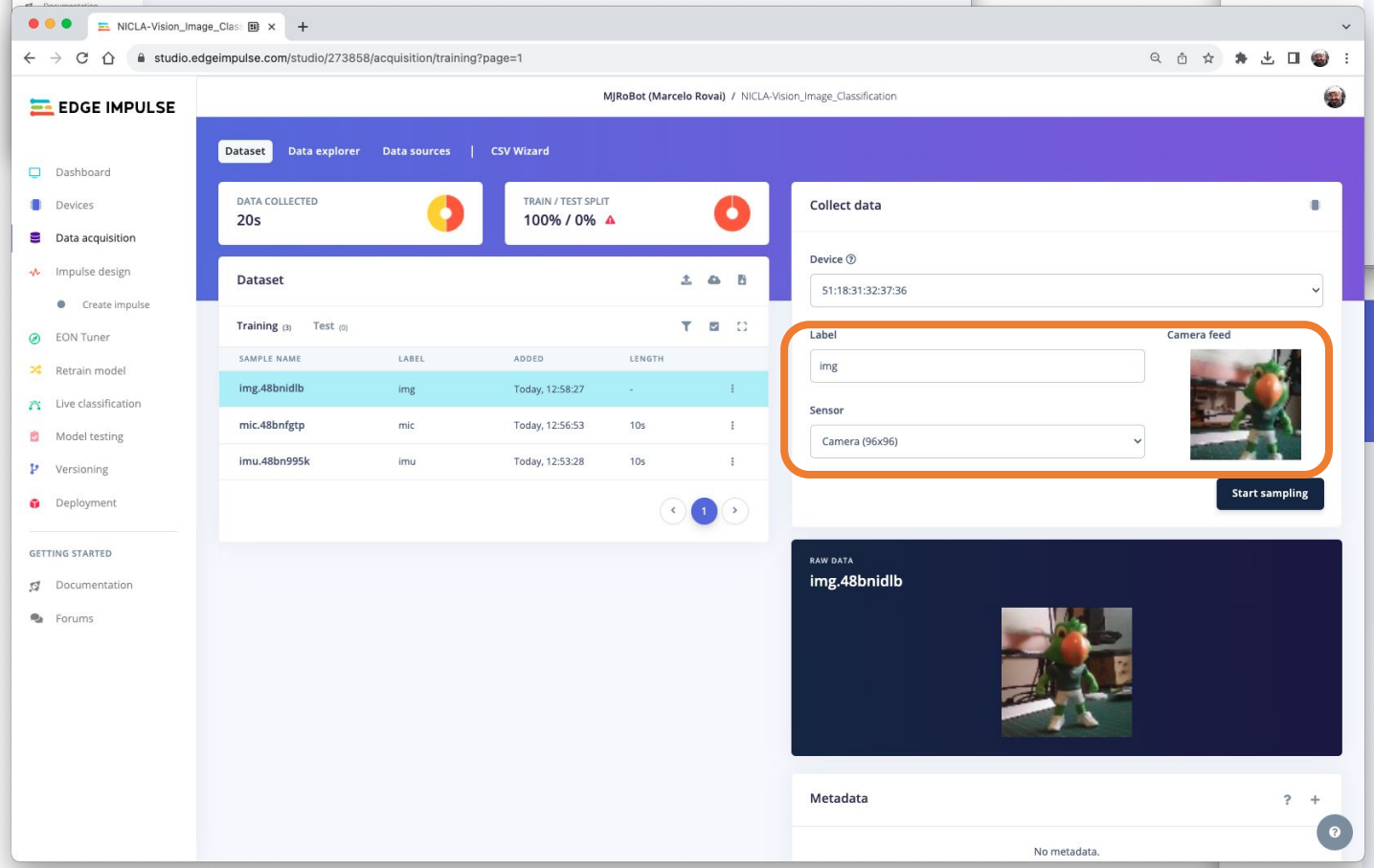
1. Put the Nicla-Vision on Boot mode (press the button twice)
2. Download the FW from the Edge Impulse Site
3. Open the zip file on your computer and select the uploader related to your OS

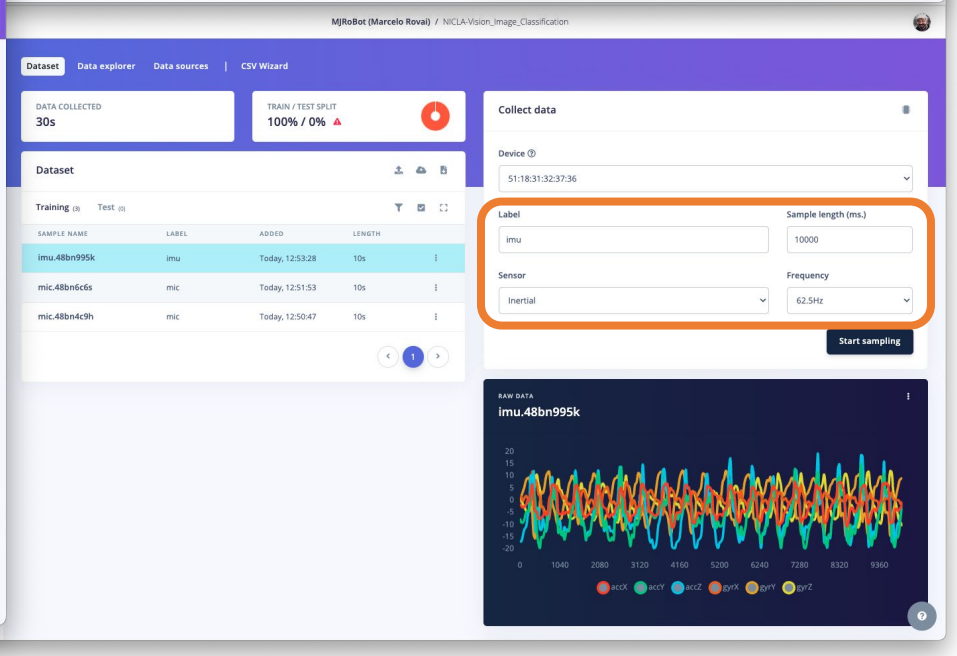





studio.edgeimpulse.com wants to connect to a serial port
Nicla Vision (cu.usbmodem101)
Cancel
Connect
MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification
CSV Wizard
Collect data
Connect a device to start building your dataset.
Add data
+ Add data


studio.edgeimpulse.com/studio/273858/acquisition/training?page=1
EDGE IMPULSE
Dataset
Data explorer
Data sources
CSV Wizard
DATA COLLECTED
20s
TRAIN / TEST SPLIT
100% / 0%
Collect data
Device
51:18:31:32:37:36
Label
mic
Sample length (ms.)
10000
Sensor
Built-in microphone
Frequency
16000Hz
Start sampling
Dataset
Training (2)
Test (0)
SAMPLE NAME
LABEL
ADDED
LENGTH
mic.48bnfgtp
mic
Today, 12:56:53
10s
imu.48bn995k
imu
Today, 12:53:28
10s
RAW DATA
mic.48bnfgtp
audio
0:00 / 0:00


studio.edgeimpulse.com/studio/273858/acquisition/training?page=1
EDGE IMPULSE
MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification
Dataset
Data explorer
Data sources
CSV Wizard
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
DATA COLLECTED
20s
TRAIN / TEST SPLIT
100% / 0%
Dataset
Training (3)
Test (0)
SAMPLE NAME
LABEL
ADDED
LENGTH
img.48bnidlb
img
Today, 12:58:27
mic.48bnfgtp
mic
Today, 12:56:53
10s
imu.48bn995k
imu
Today, 12:53:28
10s
Collect data
Device
51:18:31:32:37:36
Label
img
Camera feed
Sensor
Camera (96x96)
Start sampling
RAW DATA
img.48bnidlb
Metadata
No metadata.


MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification
Dataset
Data explorer
Data sources
CSV Wizard
DATA COLLECTED
30s
TRAIN / TEST SPLIT
100% / 0%
Collect data
Device
51:18:31:32:37:36
Label
imu
Sample length (ms.)
10000
Sensor
Inertial
Frequency
62.5Hz
Start sampling
Dataset
Training (3)
Test (0)
SAMPLE NAME
LABEL
ADDED
LENGTH
imu.48bn995k
imu
Today, 12:53:28
10s
mic.48bn6c6s
mic
Today, 12:51:53
10s
mic.48bn4c9h
mic
Today, 12:50:47
10s
RAW DATA
imu.48bn995k
accX
accY
accZ
gyrX
gyrY
gyrZ

TinyML in Computer Vision, Introduction

Computer Vision Main Types

Image Classification

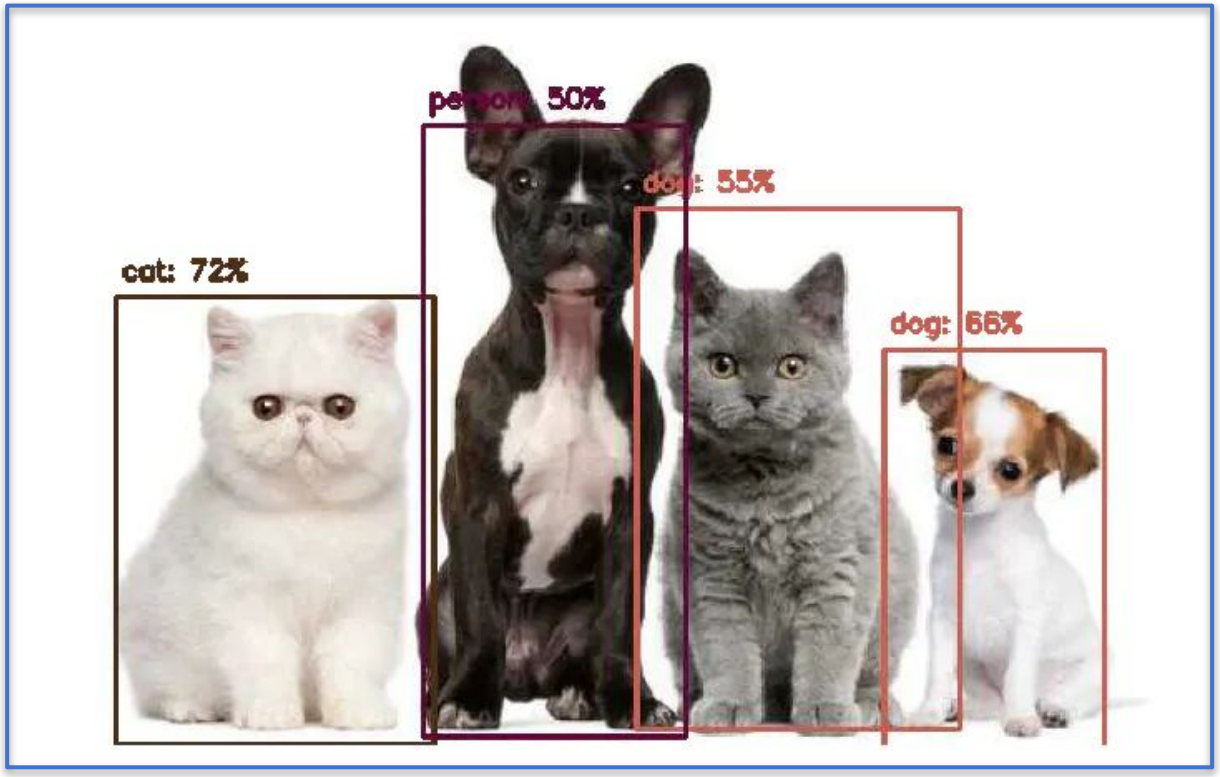
(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

Object Detection

Multi-Label Classification + Object **Localization**



Computer Vision Main Types

Image Classification

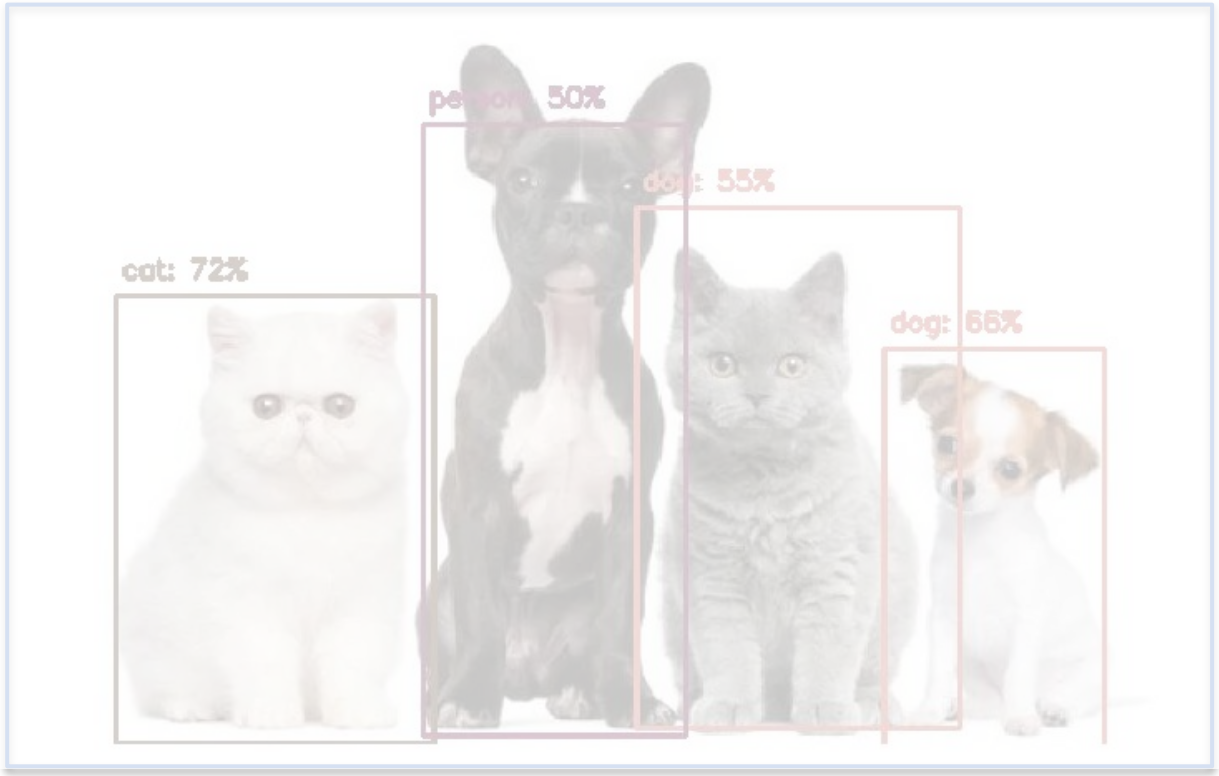
(**Multi-Class** Classification)

Cat: 70%

Dog: 80%

Object Detection

Multi-Label Classification + Object **Localization**

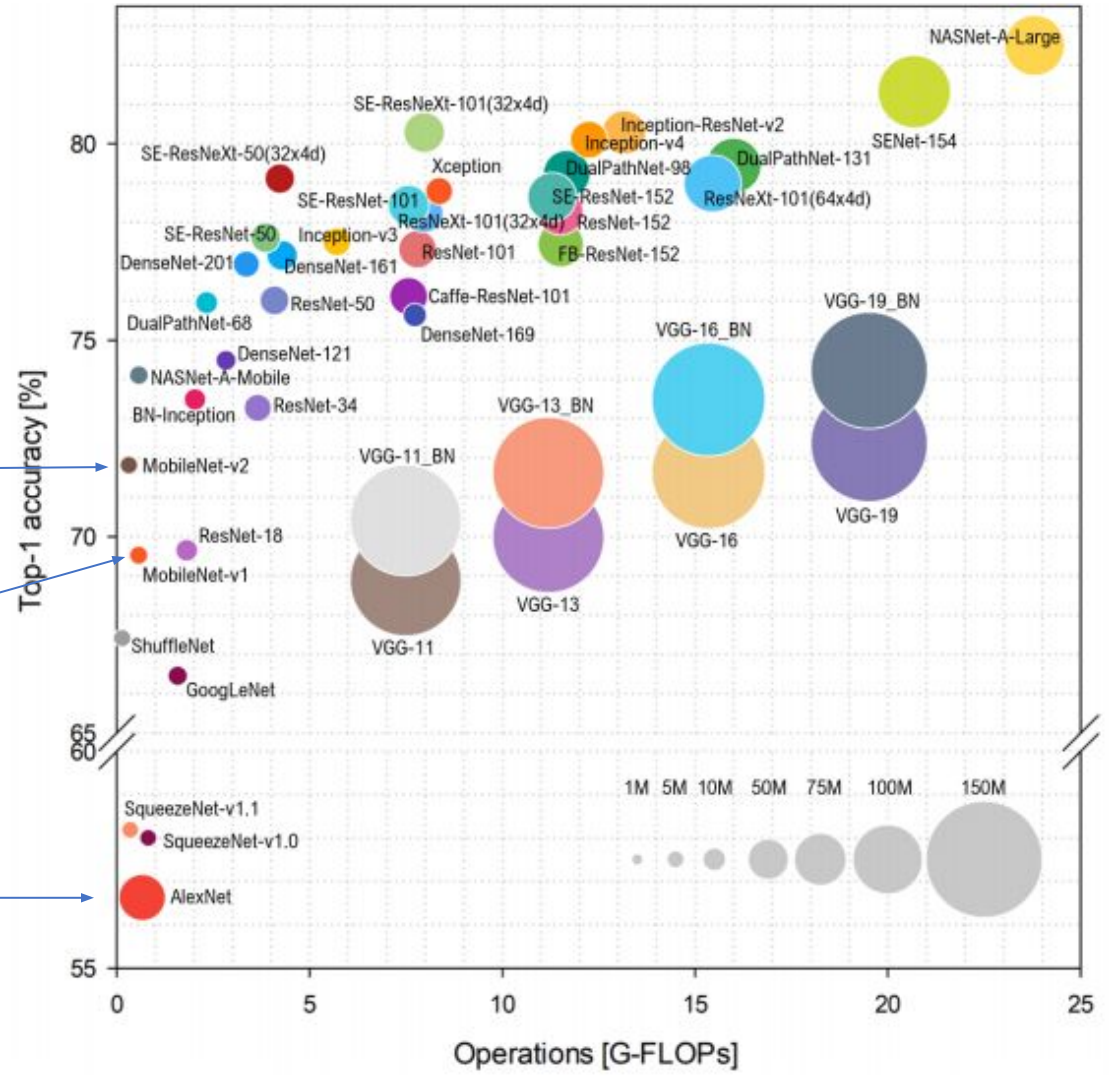


Computer Vision Models evolution

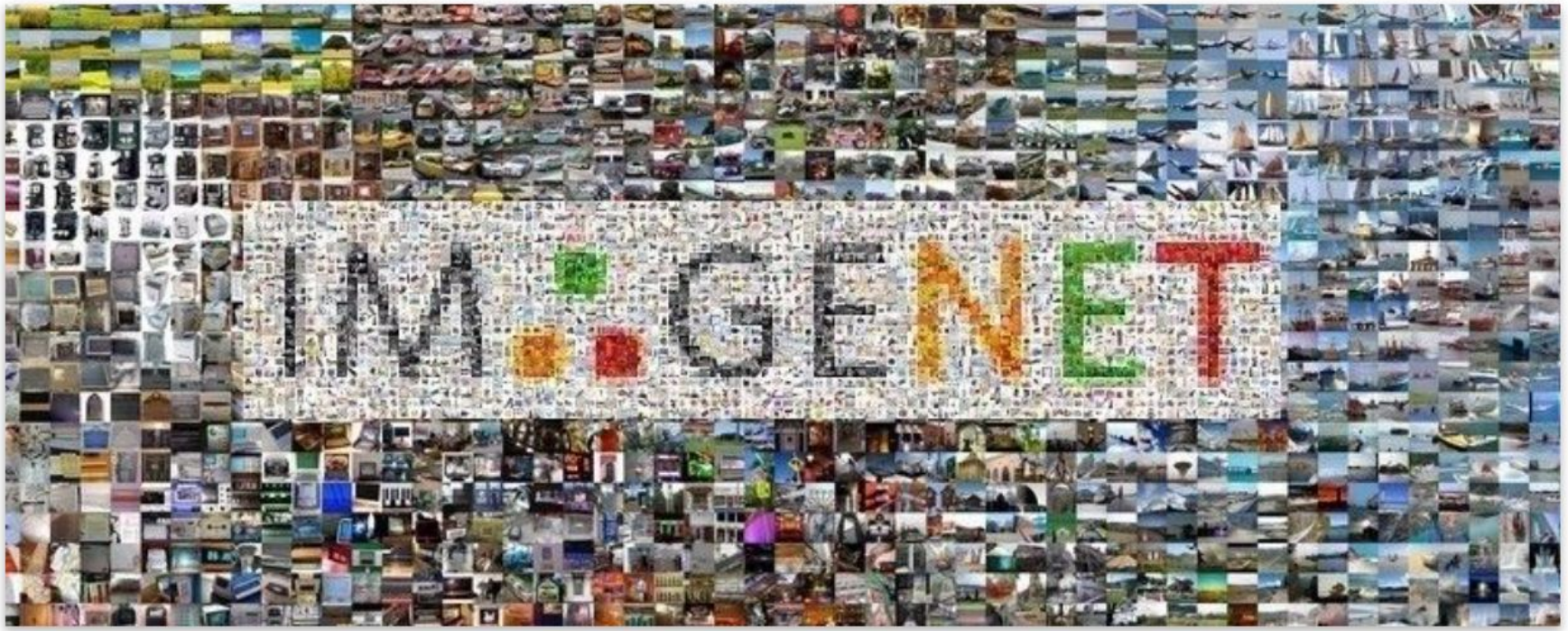
(2018)

(2017)

(2012)



Training Pipeline: **Need Lots of Data**

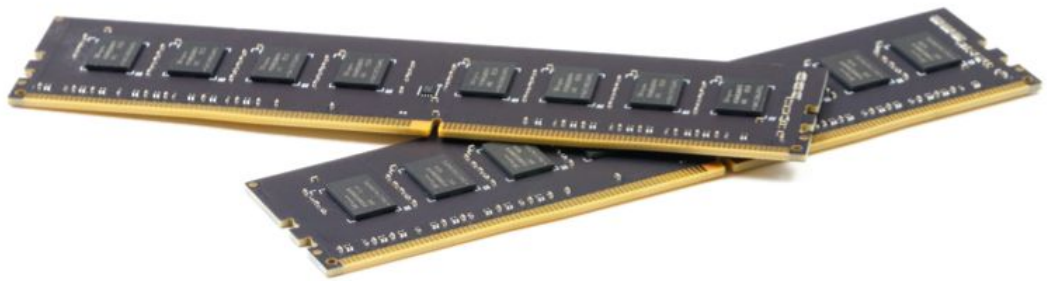


1000 Classes

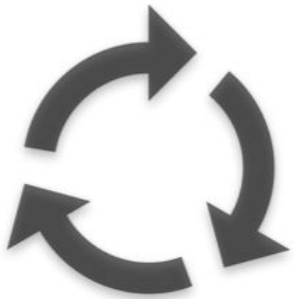
1000 Images / Class

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

Training Pipeline: **Need Compute Resources**



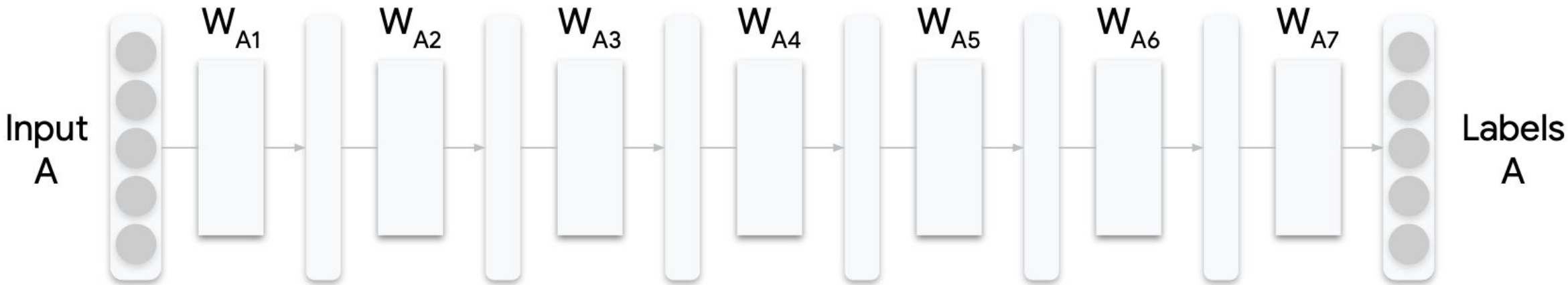
Memory



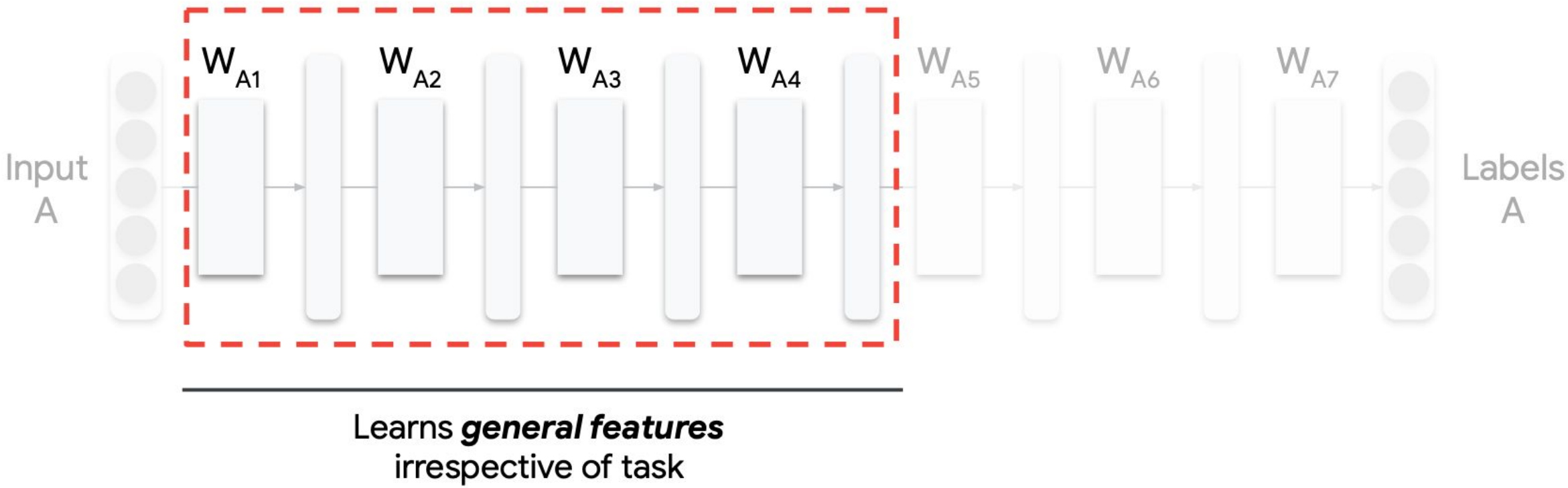
Compute

GPU and
Accelerators

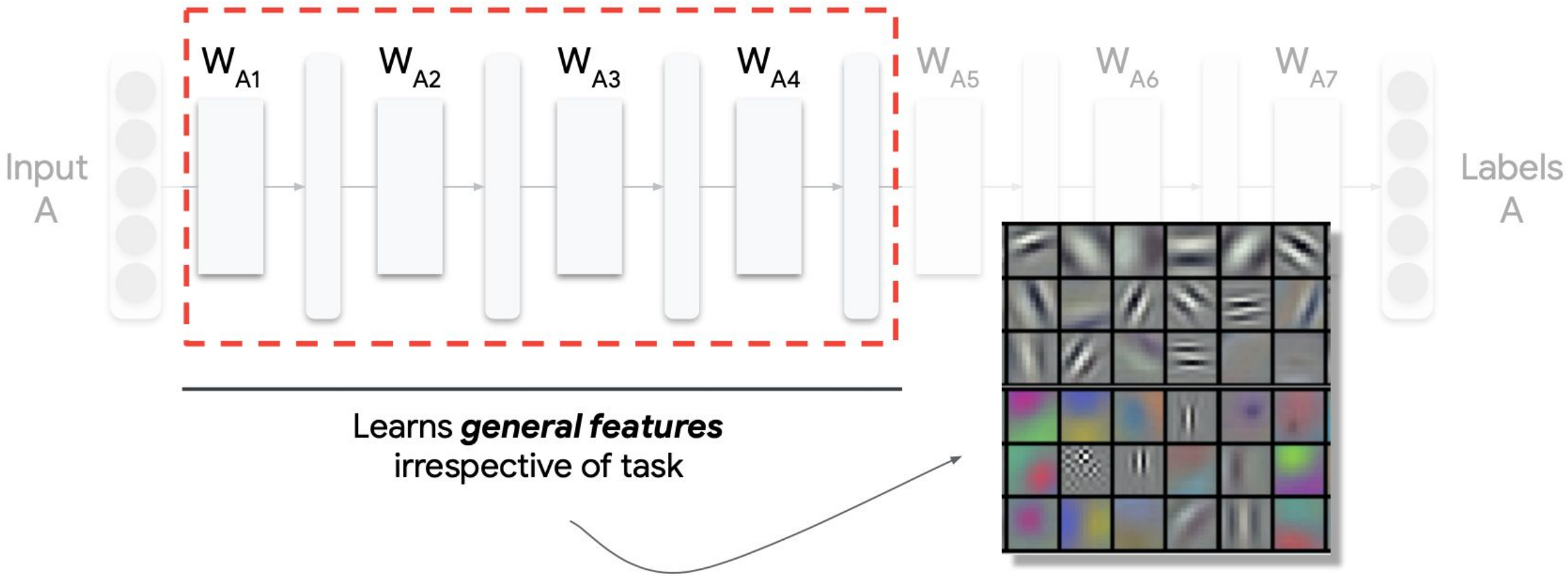
End Result of Training



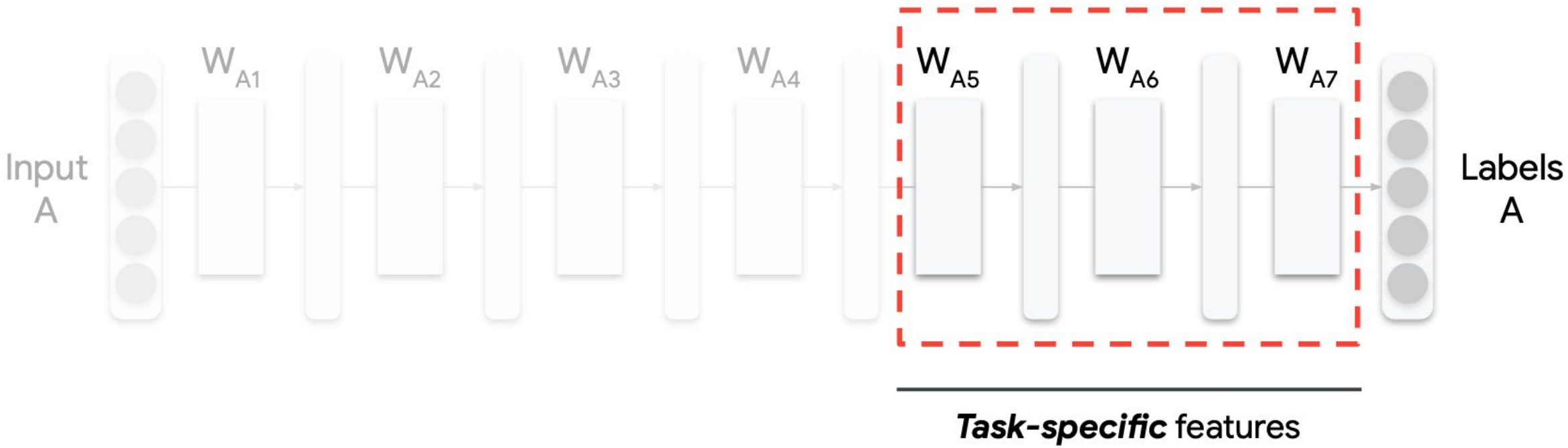
End Result of Training



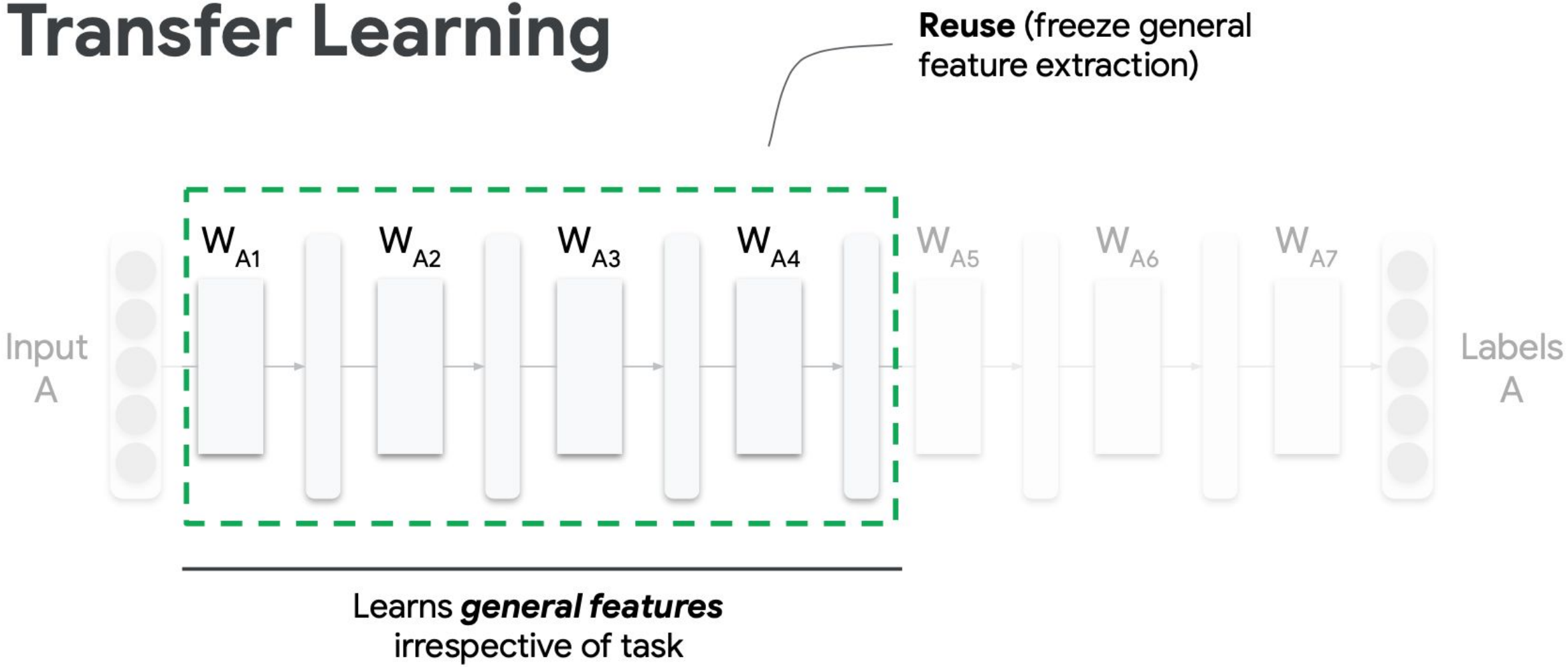
End Result of Training



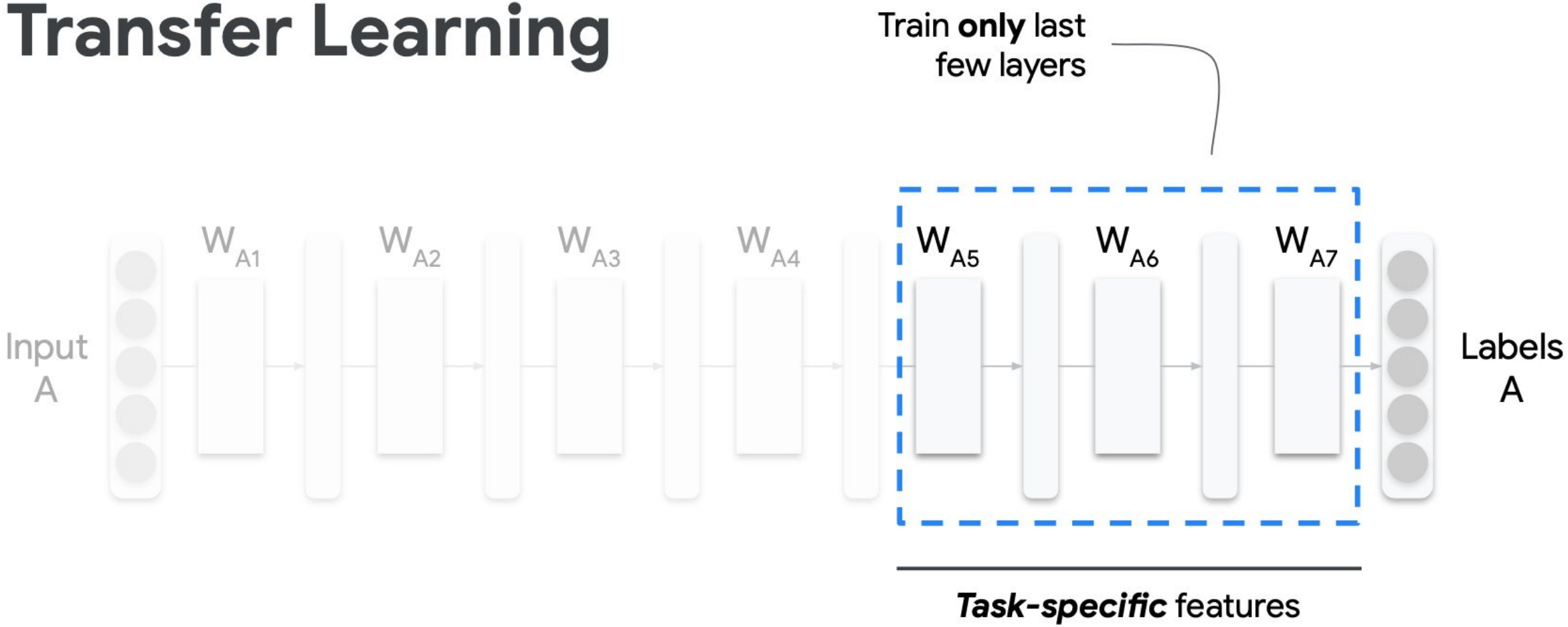
End Result of Training



Transfer Learning



Transfer Learning



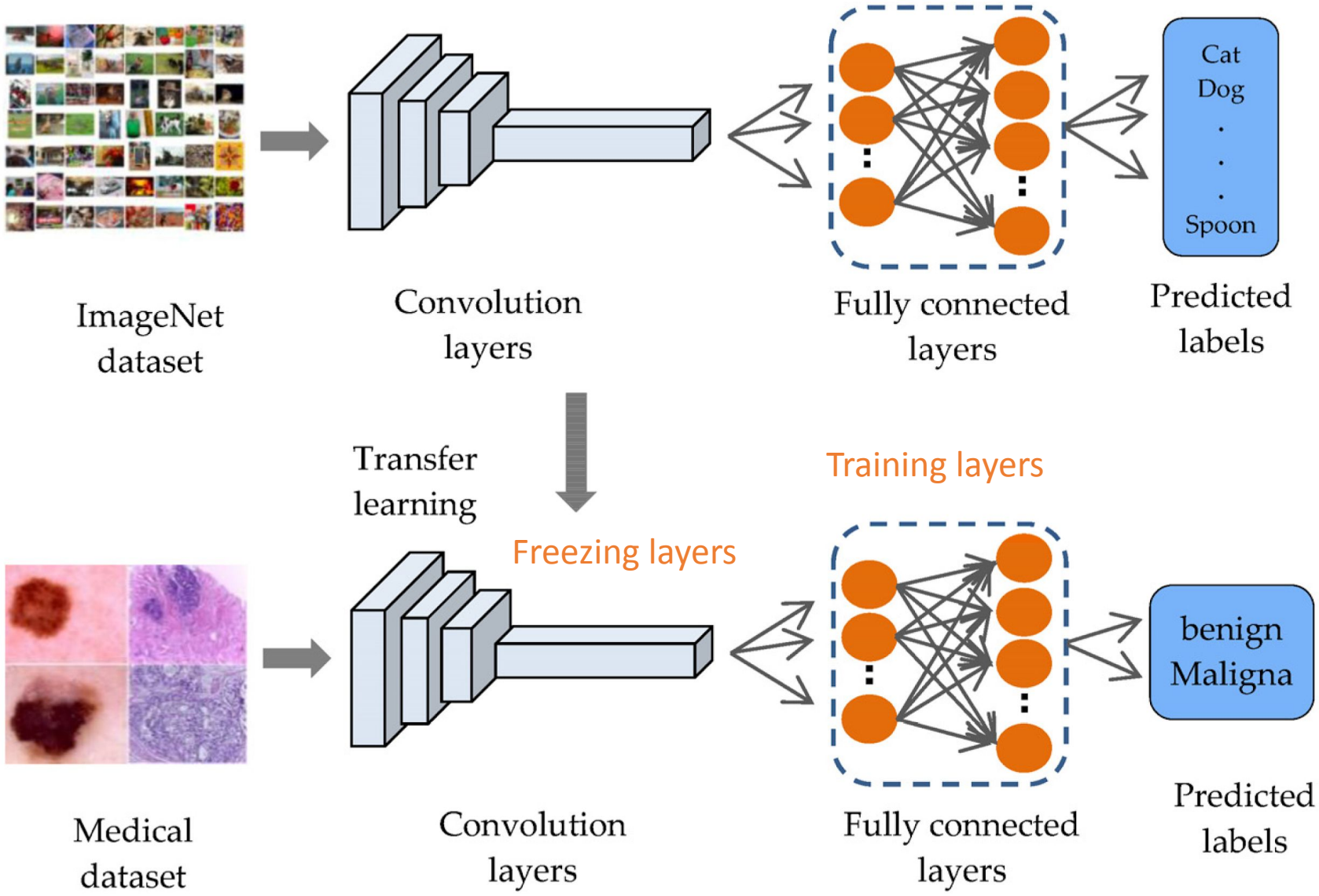
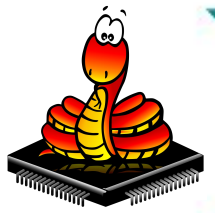

ImageNet dataset
Convolution layers
Fully connected layers
Cat
Dog
Spoon
Predicted labels
Transfer learning
Training layers
Freezing layers
Medical dataset
Convolution layers
Fully connected layers
benign
Maligna
Predicted labels

Image Classification Application:
Design, Train, Test and Deploy

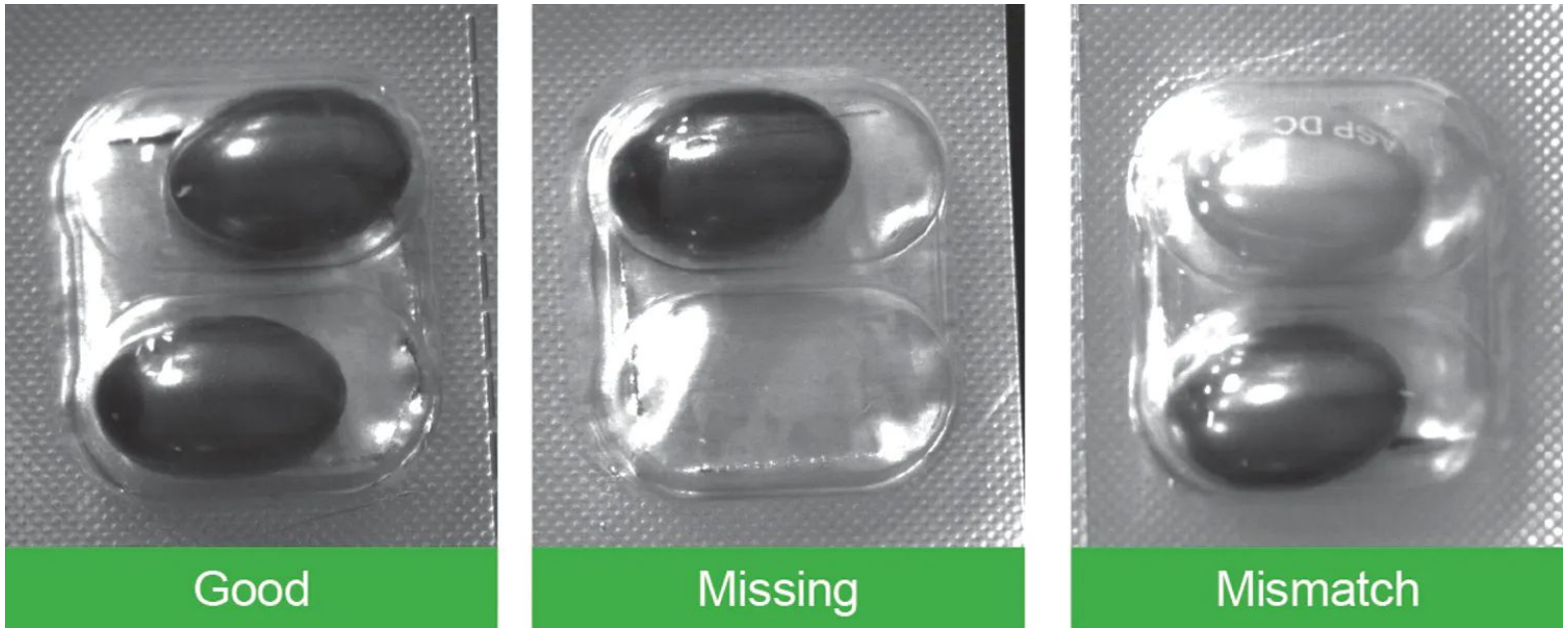
* Feature Extraction

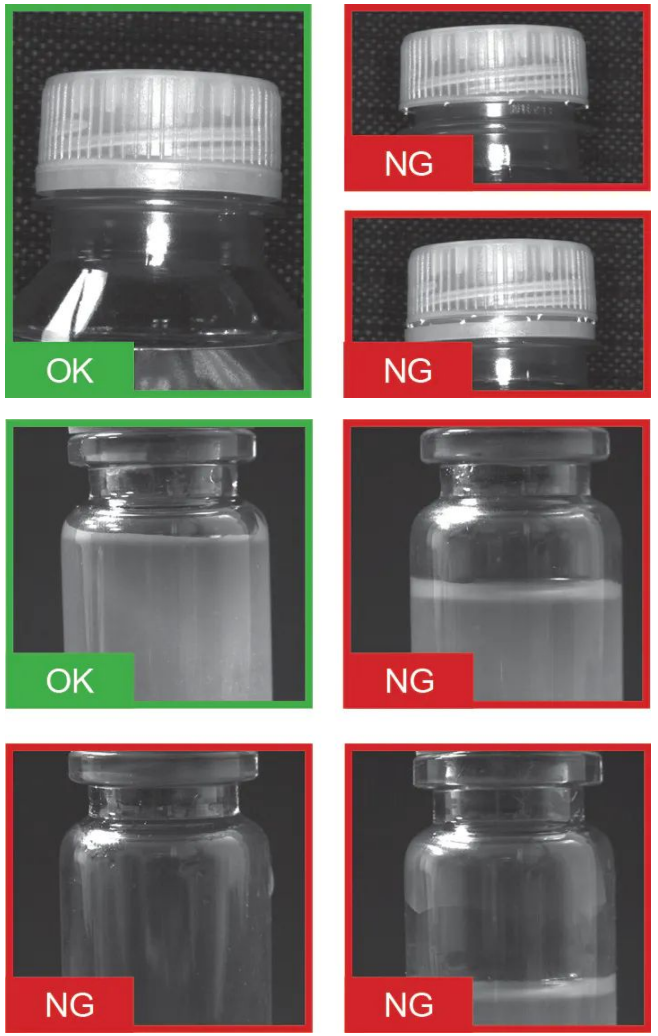
On-Premises / Cloud (Server)

Edge Device (Client)



Package Inspection





[Deep Learning at the Edge Simplifies Package Inspection]

Image Classification Project 1

Decide a Goal

- Possible Images:
  - medicine
  - background

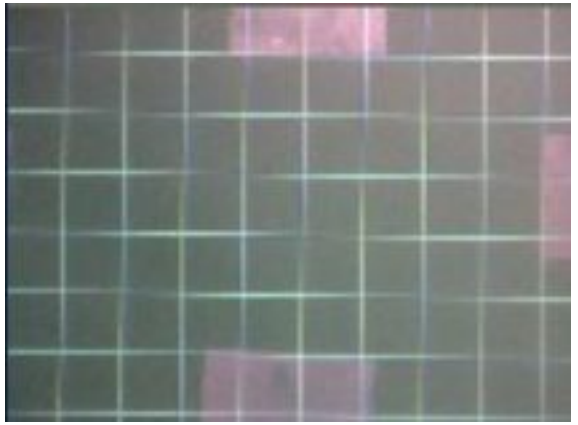
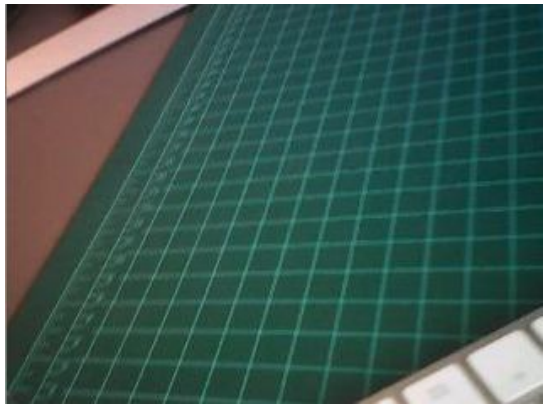
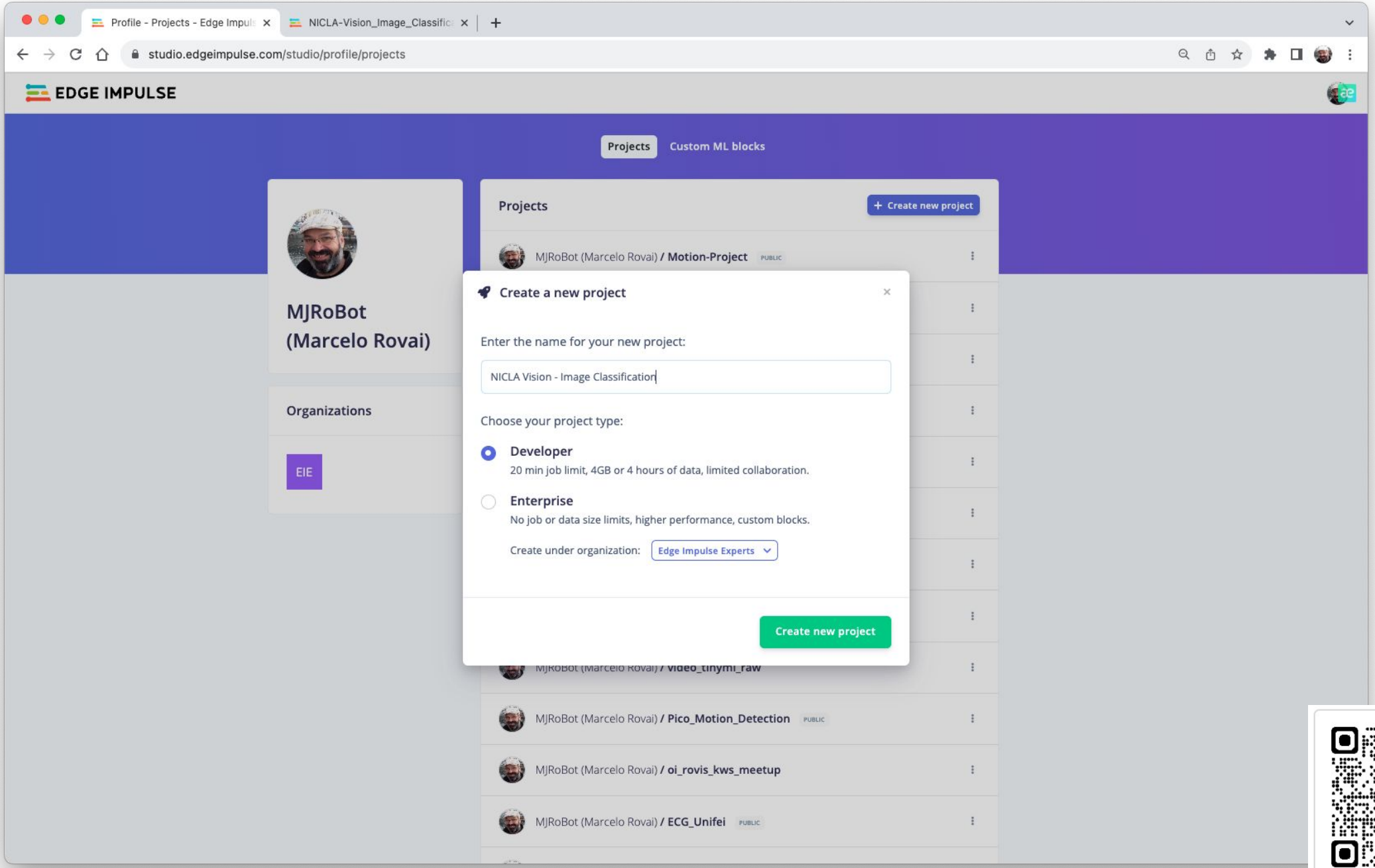


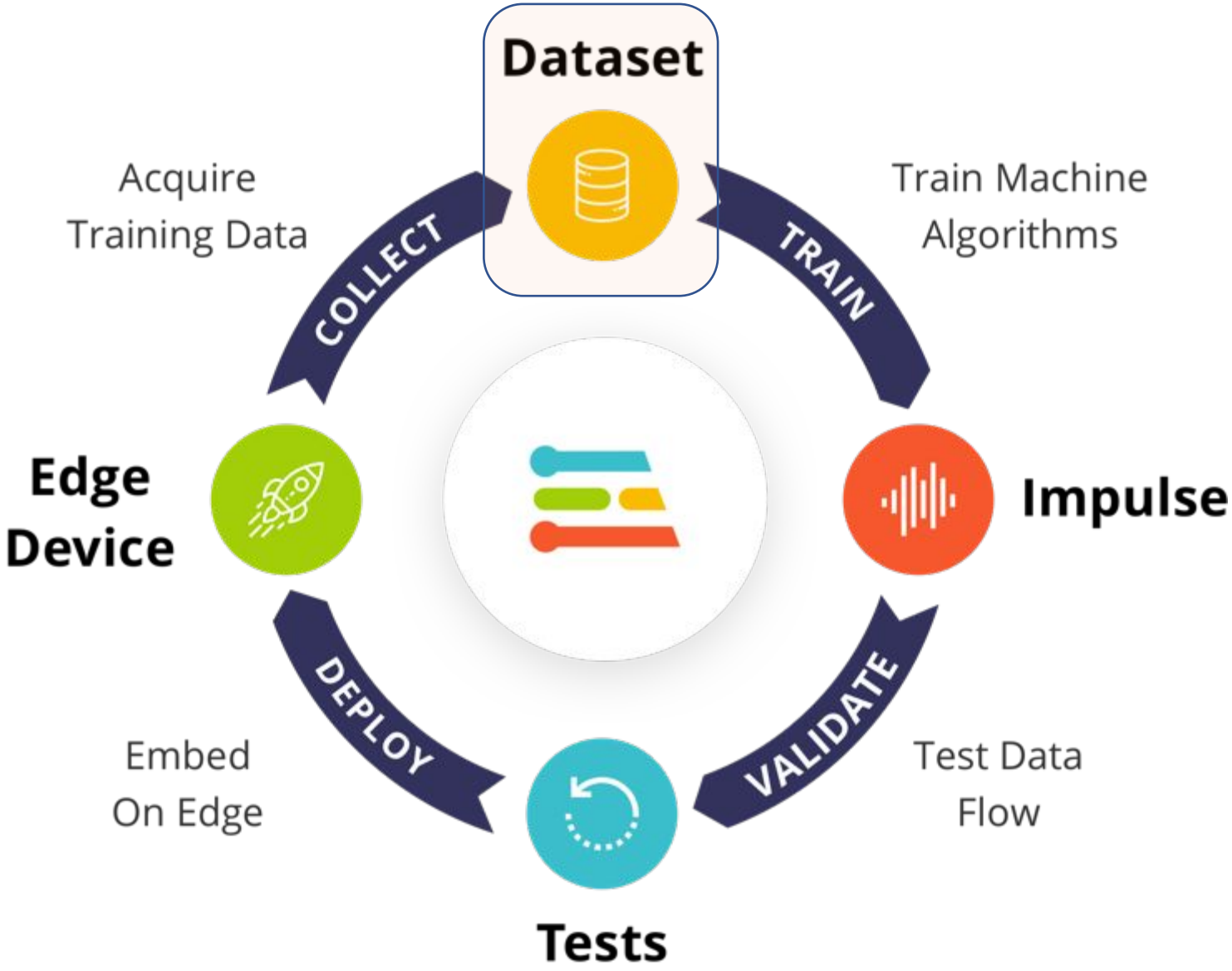
Image Classification Project 2

Decide a Goal

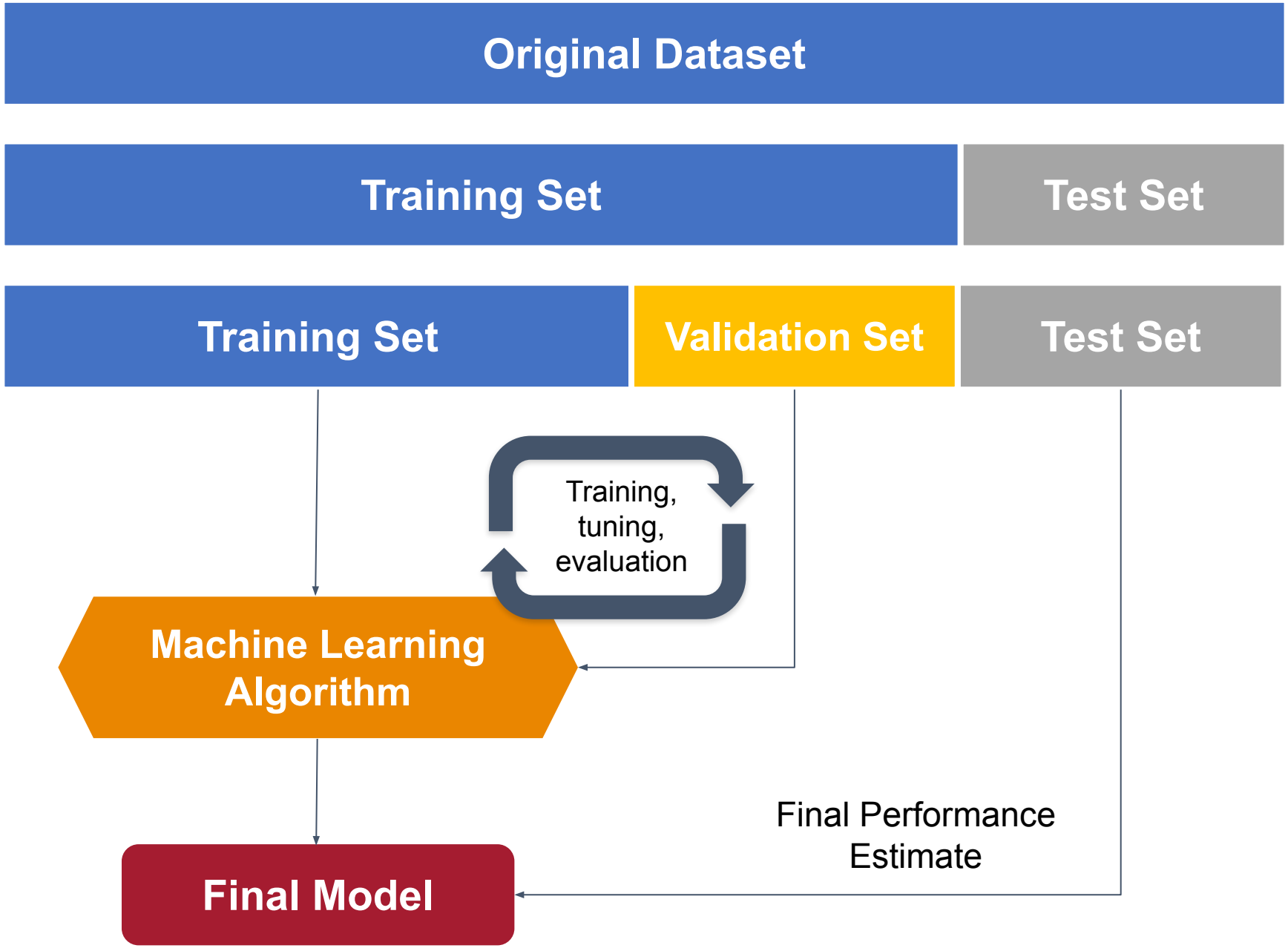
- Classes:
  - background
  - periquito
  - robot



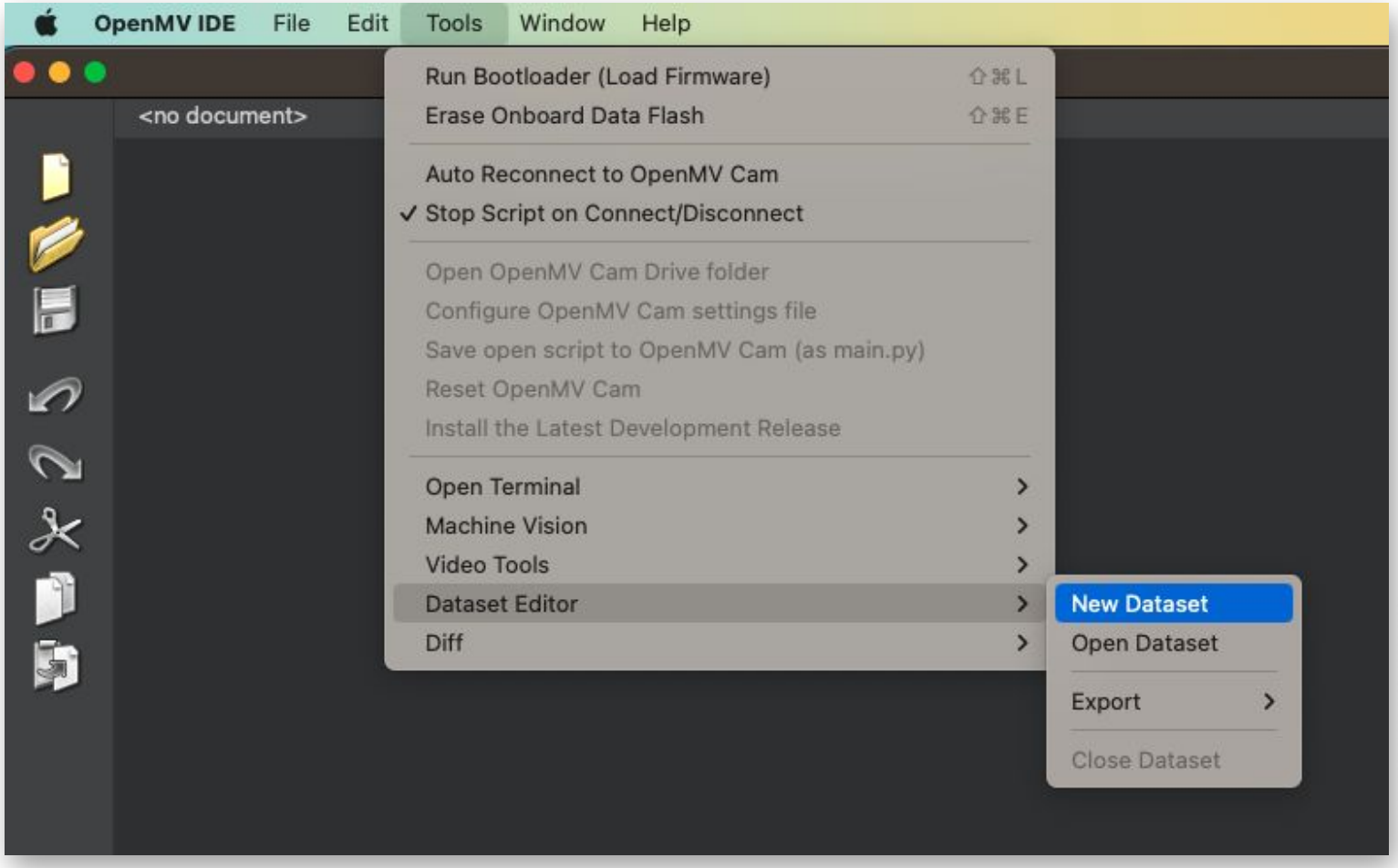

Profile - Projects - Edge Impuls
NICLA-Vision_Image_Classific
studio.edgeimpulse.com/studio/profile/projects
EDGE IMPULSE
Projects
Custom ML blocks
MJRoBot (Marcelo Rovai)
Organizations
EIE
Projects
+ Create new project
MJRoBot (Marcelo Rovai) / Motion-Project PUBLIC
Create a new project
Enter the name 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) / Pico_Motion_Detection PUBLIC
MJRoBot (Marcelo Rovai) / oi_rovis_kws_meetup
MJRoBot (Marcelo Rovai) / ECG_Unifei PUBLIC

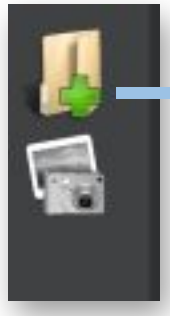


- Pre-Processing Data
- Design a Model
- Train a Model


Original Dataset
Training Set
Test Set
Training Set
Validation Set
Test Set
Training,
tuning,
evaluation
Machine Learning
Algorithm
Final Performance
Estimate
Final Model

Collecting Dataset with OpenMV IDE

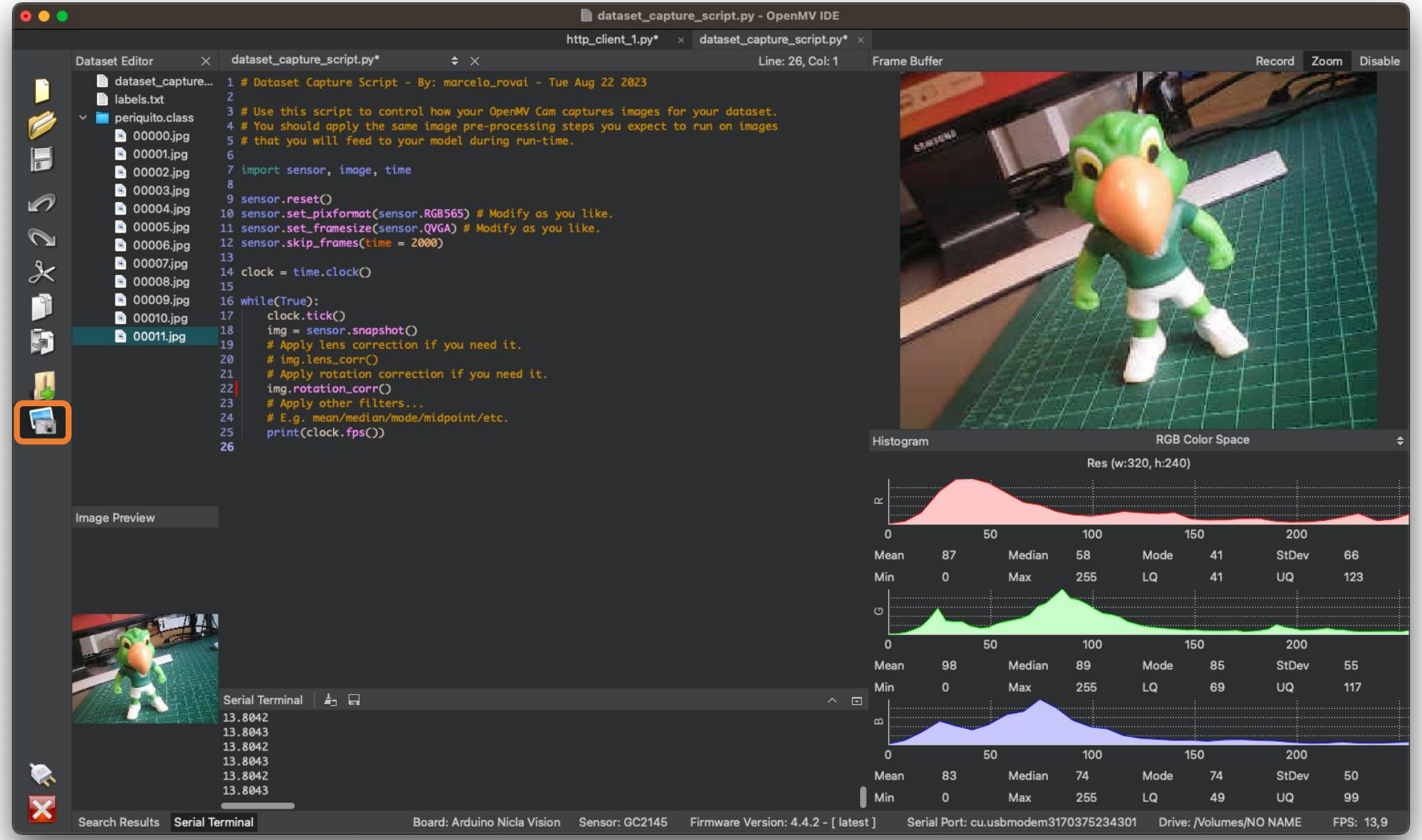




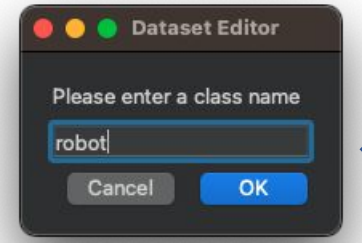
Create a folder, for example, ***data***, and create a **class name**

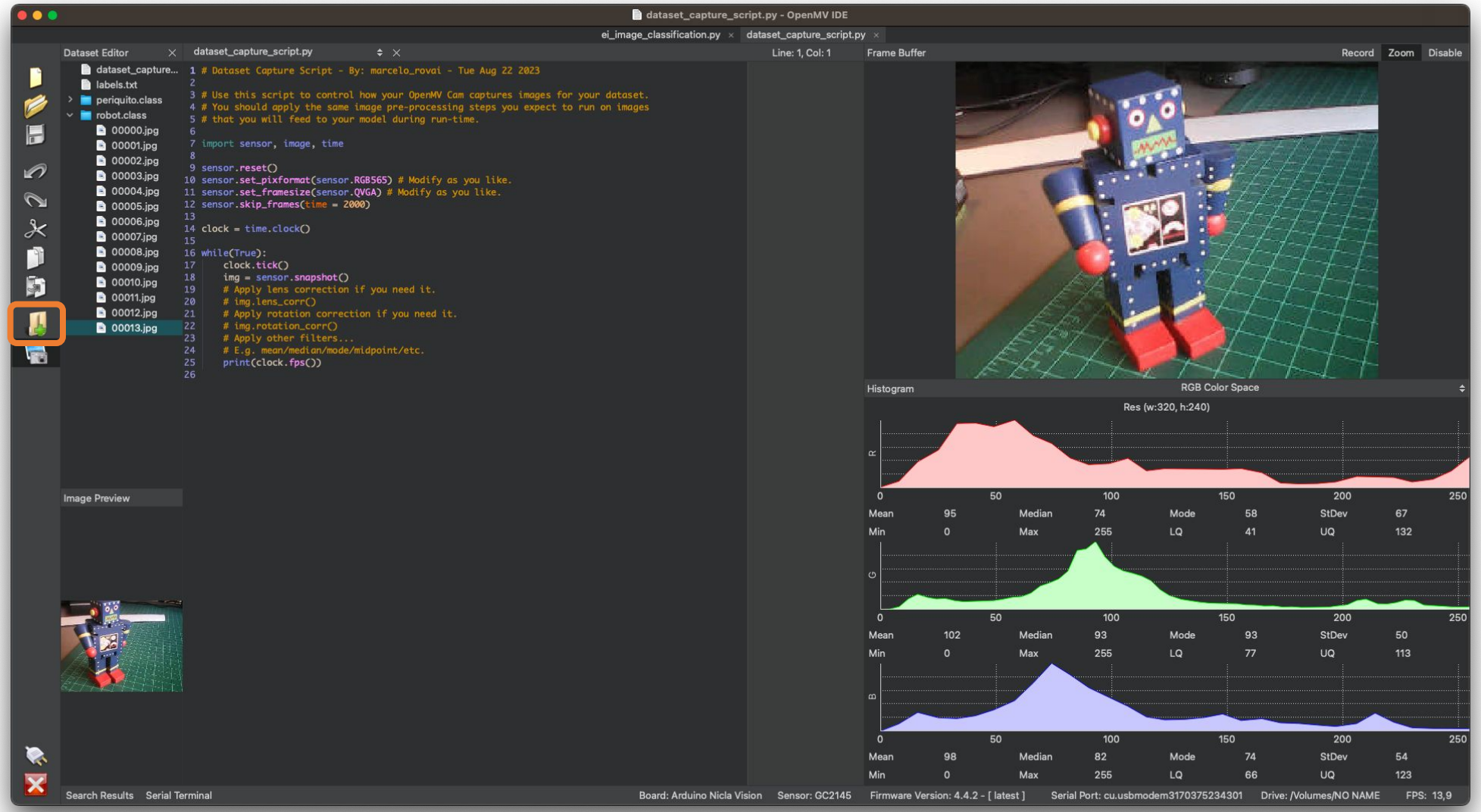
Start Collecting images …

Click here to capture an image

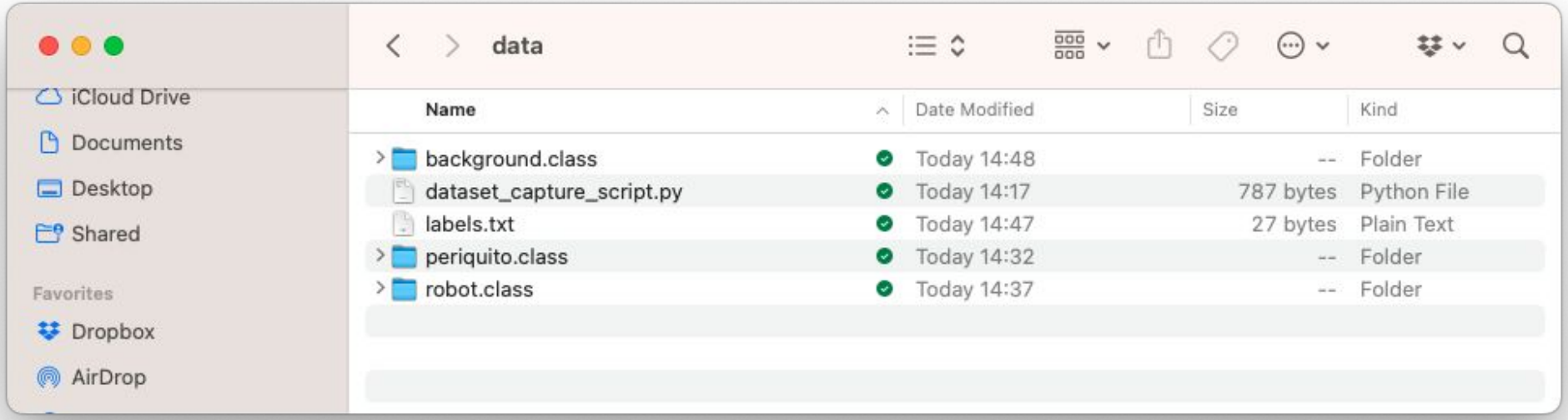


Repeat with other classes …

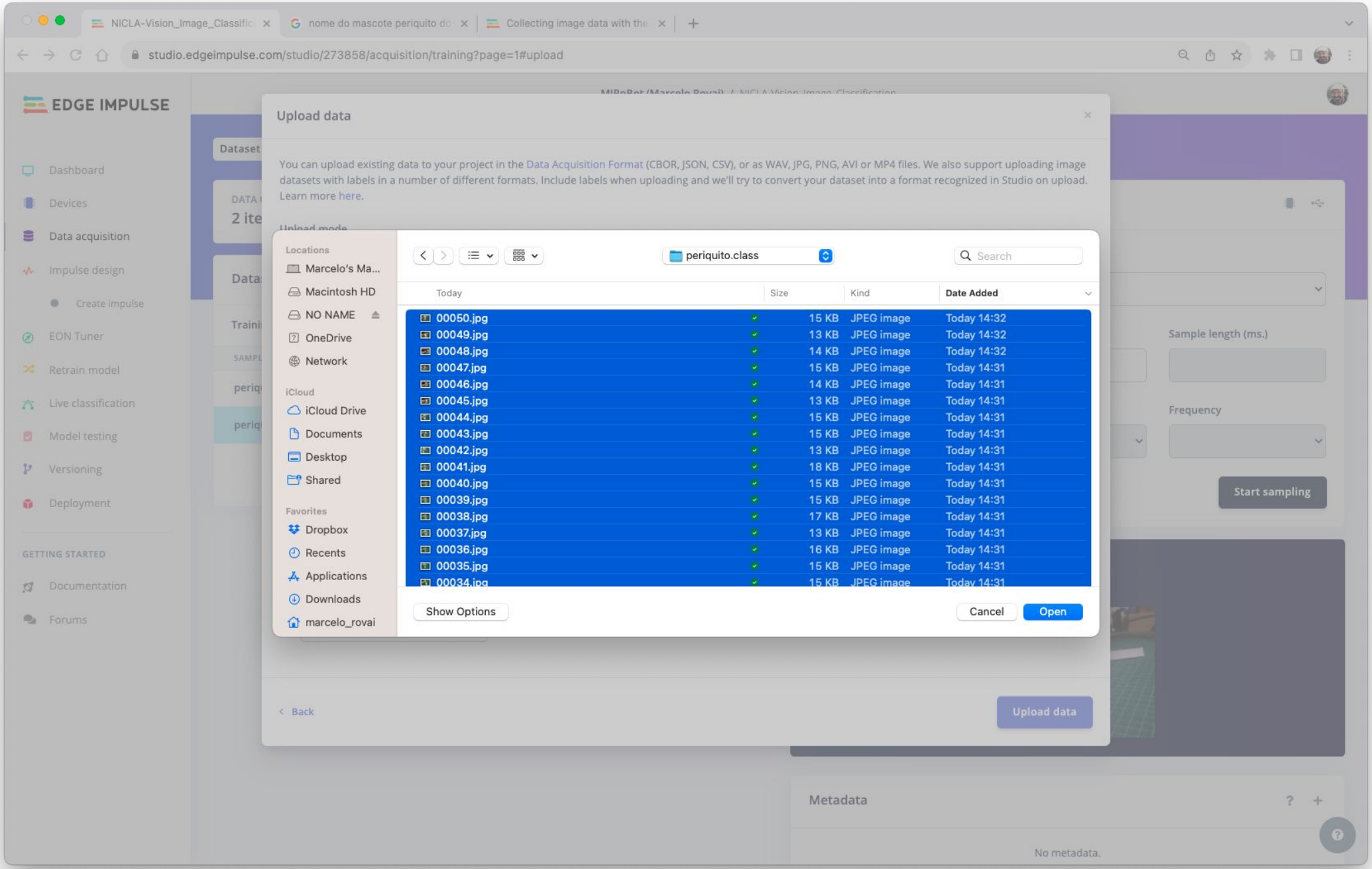


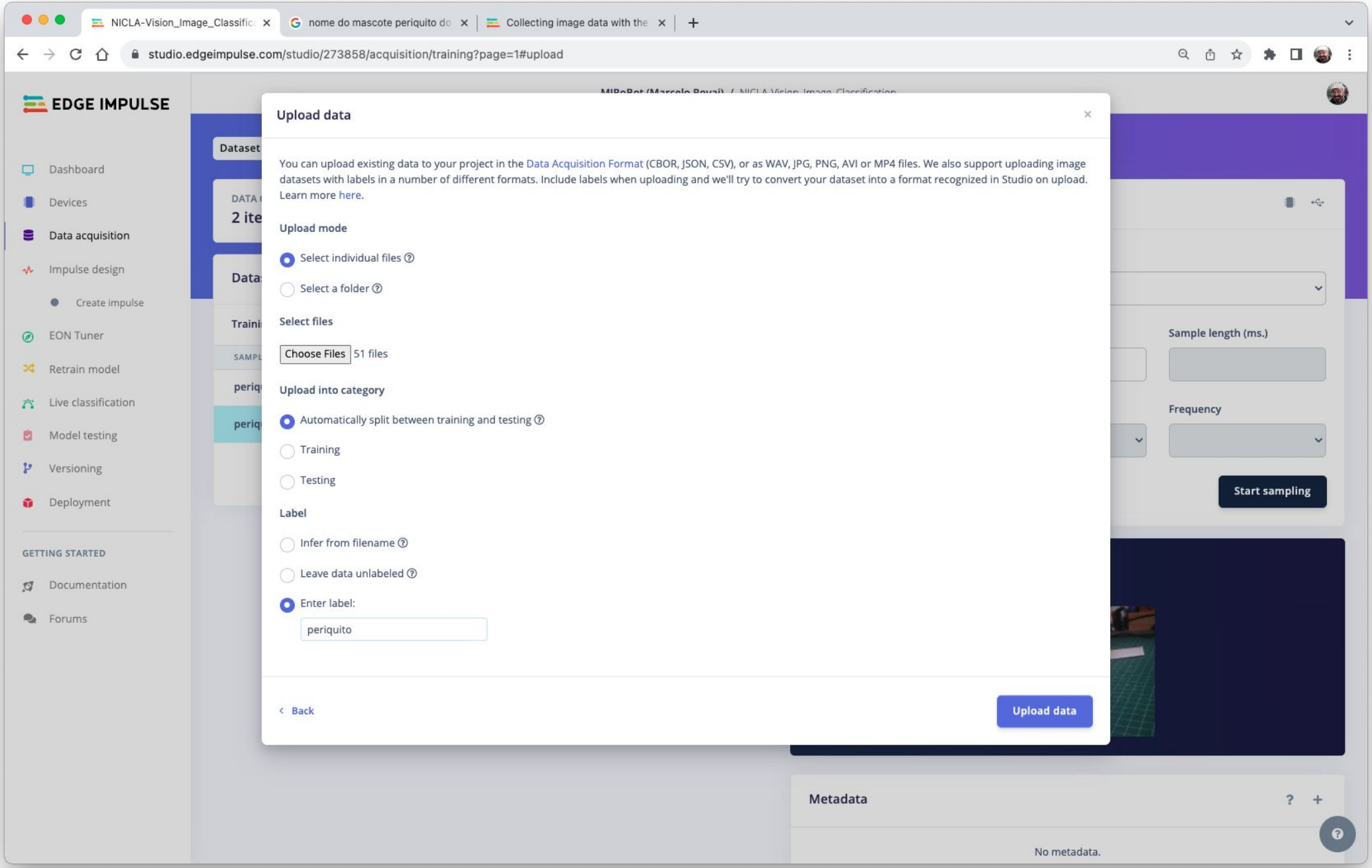


- After capturing your dataset, close the Dataset Editor Tool.
- On your computer, you will end with a dataset that contains three classes: periquito, robot, and background.



- 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
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.
Locations
Marcelo's Ma...
Macintosh HD
NO NAME
OneDrive
Network
iCloud
iCloud Drive
Documents
Desktop
Shared
Favorites
Dropbox
Recents
Applications
Downloads
marcelo_rovai
periquito.class
Search
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
Show Options
Cancel
Open
Back
Upload data
Sample length (ms.)
Frequency
Start sampling
Metadata
No metadata.


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
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:
periquito
Back
Upload data
Sample length (ms.)
Frequency
Start sampling
Metadata
No metadata.

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

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

EDGE IMPULSE

MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification

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

GETTING STARTED

- Documentation
- Forums

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

‹ 1 2 3 4 5 6 ... 11 ›

Collect data

Device
No devices connected

Label
periquito

Sample length (ms.)

Sensor

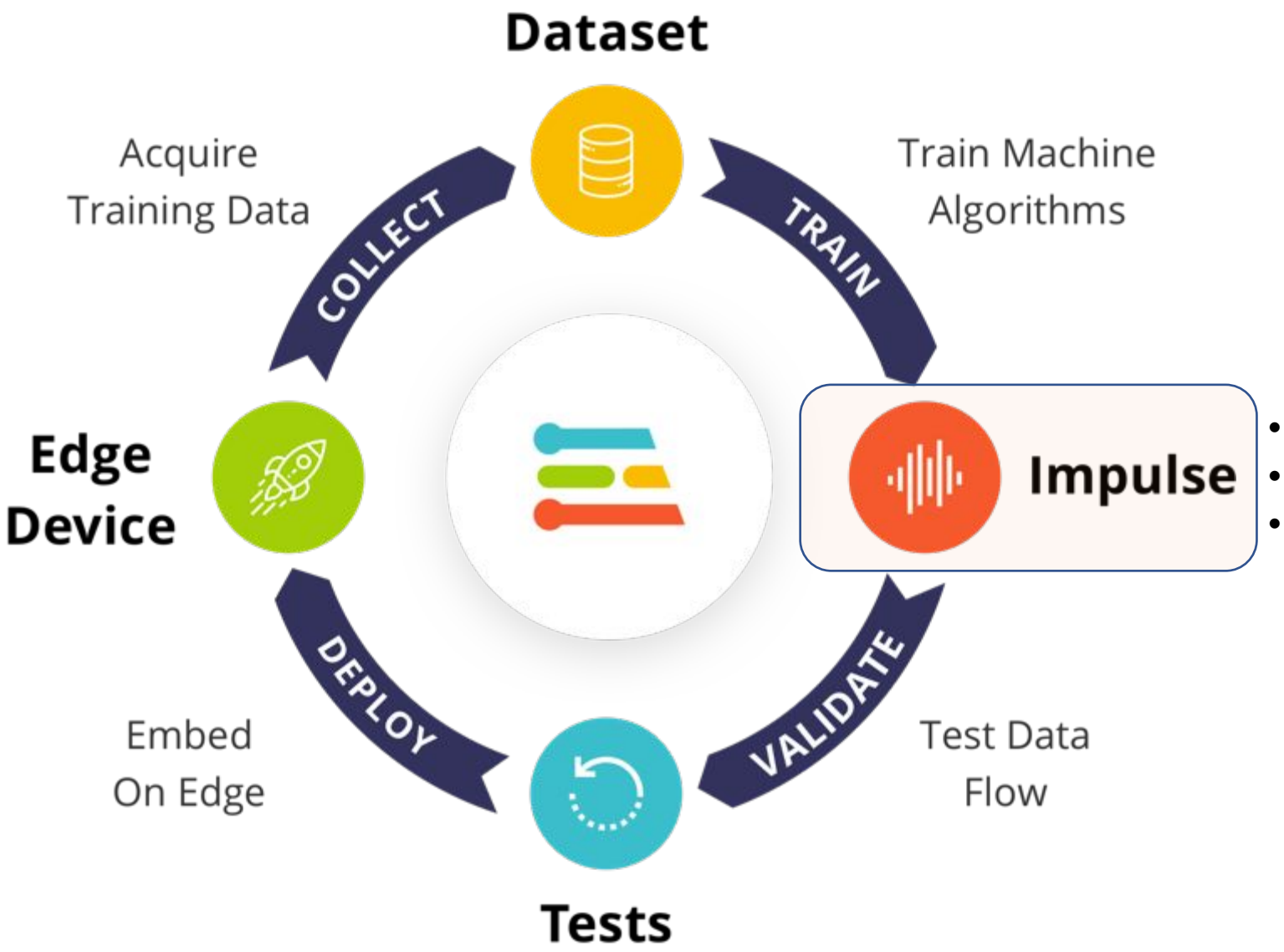
Frequency

Start sampling

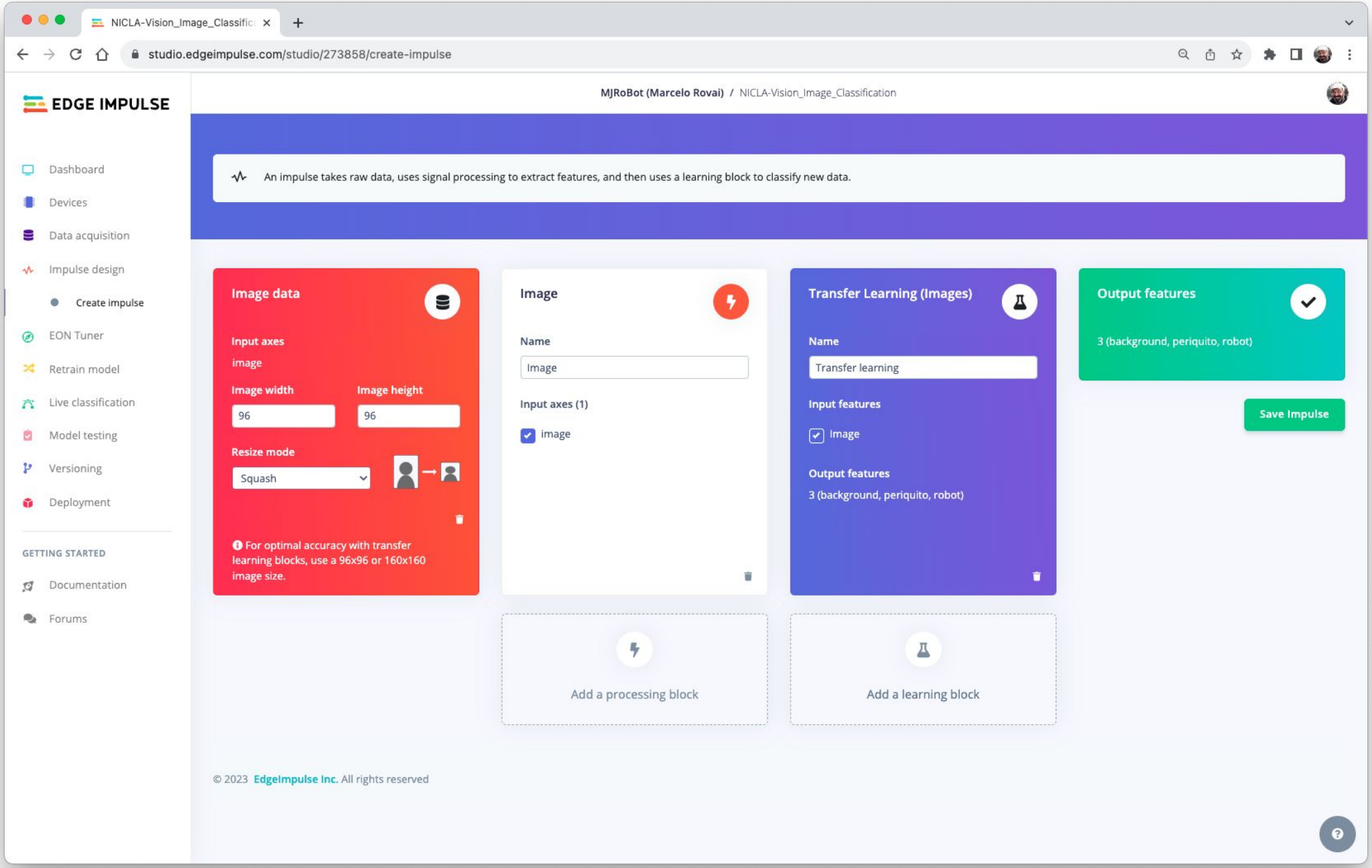
RAW DATA
00001

Metadata

No metadata.



- Pre-Processing Data
- Design a Model
- Train a Model


NICLA-Vision_Image_Classific
studio.edgeimpulse.com/studio/273858/create-impulse
EDGE IMPULSE
MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
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 EdgeImpulse Inc. All rights reserved

NICLA-Vision_Image_Classific

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

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

#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_Classific

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

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

#1 Click to set a description for this version

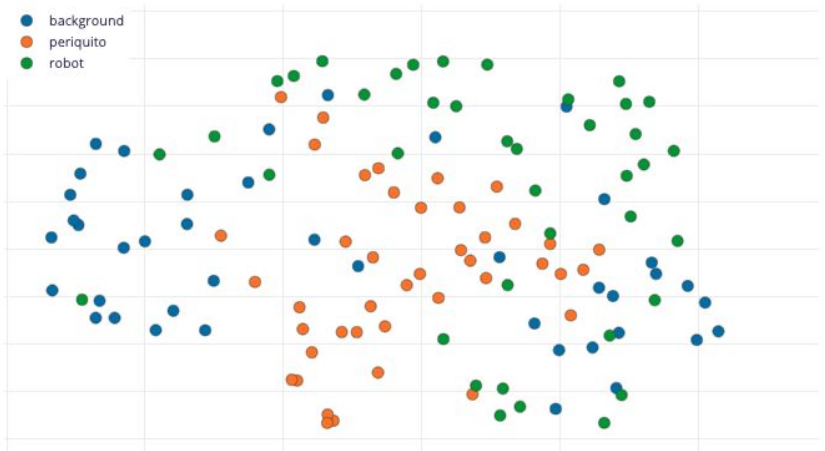
Parameters | Generate features

Training set

Data in training set	124 items
Classes	3 (background, periquito, robot)

Generate features

Feature explorer

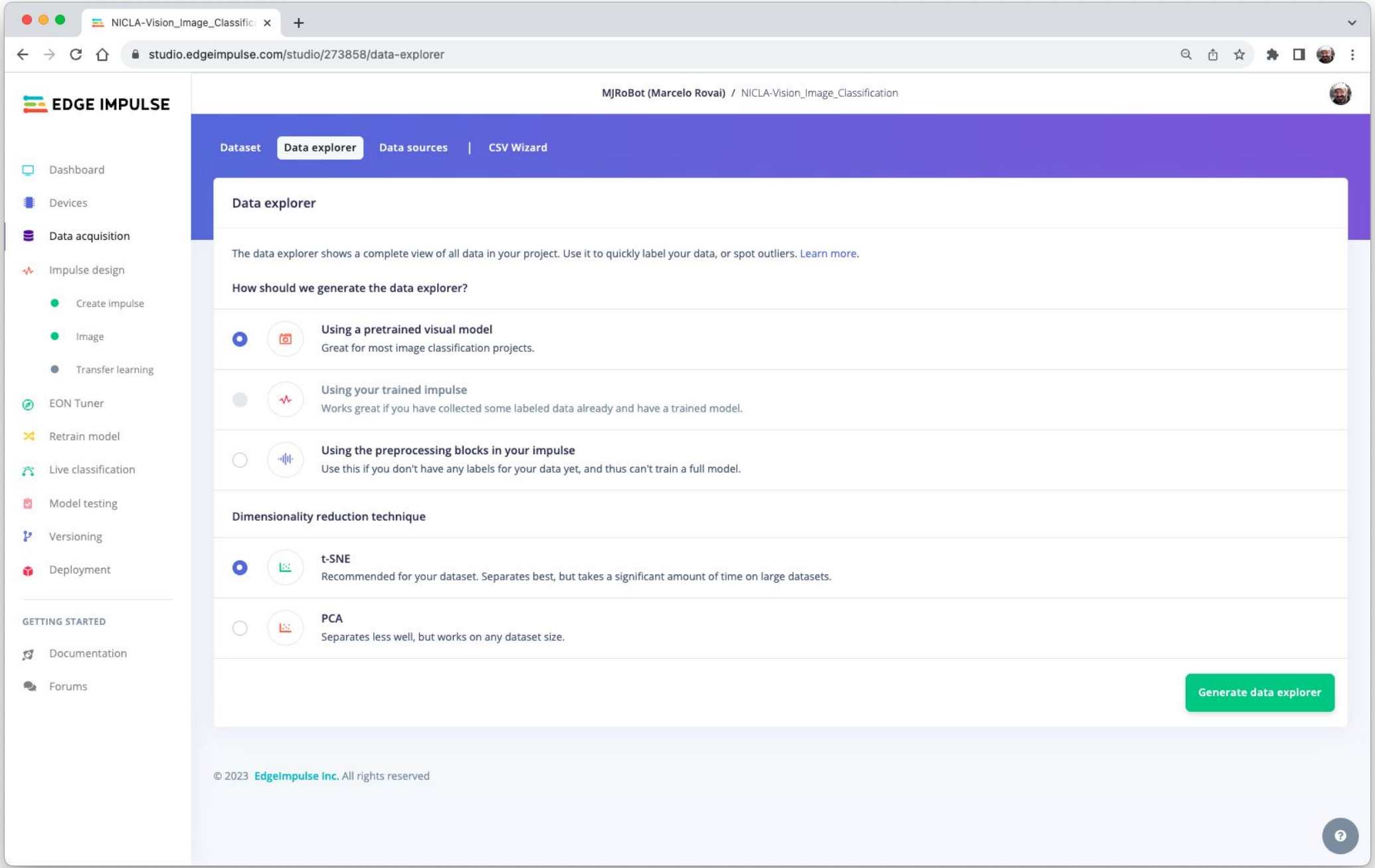


On-device performance

PROCESSING TIME
1 ms.

PEAK RAM USAGE
4 KB

© 2023 EdgeImpulse Inc. All rights reserved


NICLA-Vision_Image_Classific
studio.edgeimpulse.com/studio/273858/data-explorer
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
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 EdgeImpulse Inc. All rights reserved

NICLA-Vision_Image_Classific

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

EDGE IMPULSE

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

GETTING STARTED

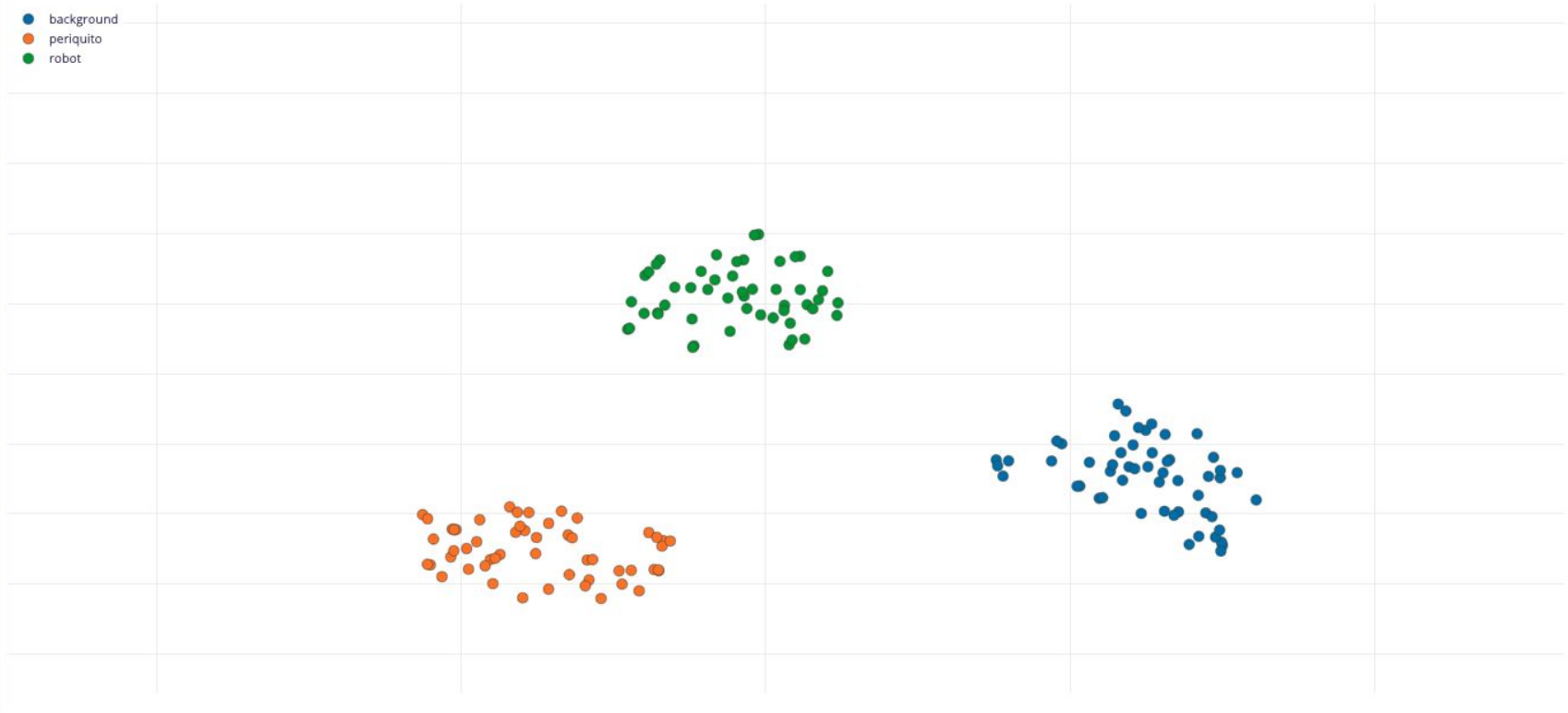
- Documentation
- Forums

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.

Drag mode: Box select items

Save changes (0 pending)

0 items selected Set labels Delete items



00017

Label: periquito

View sample

Set label Delete item

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

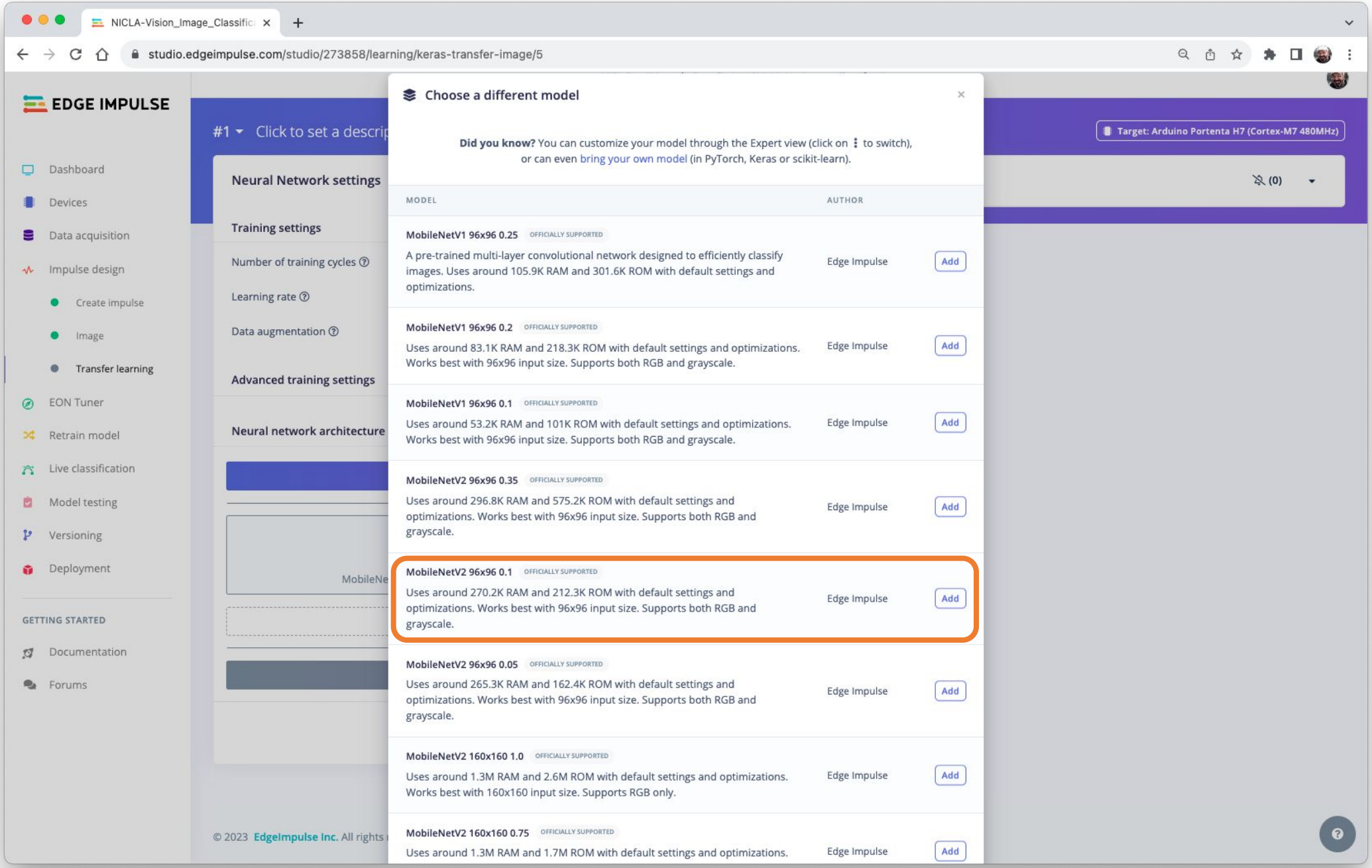
MobileNetV2 96x96 0.1

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_Classific
studio.edgeimpulse.com/studio/273858/learning/keras-transfer-image/5
EDGE IMPULSE
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
#1
Click to set a descrip
Target: Arduino Portenta H7 (Cortex-M7 480MHz)
(0)
Neural Network settings
Training settings
Number of training cycles
Learning rate
Data augmentation
Advanced training settings
Neural network architecture
MobileNe
© 2023 EdgeImpulse Inc. All rights
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 OFFICIALLY SUPPORTED
A pre-trained multi-layer convolutional network designed to efficiently classify images. Uses around 105.9K RAM and 301.6K ROM with default settings and optimizations.
Edge Impulse
Add
MobileNetV1 96x96 0.2 OFFICIALLY SUPPORTED
Uses around 83.1K RAM and 218.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.
Edge Impulse
Add
MobileNetV1 96x96 0.1 OFFICIALLY SUPPORTED
Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale.
Edge Impulse
Add
MobileNetV2 96x96 0.35 OFFICIALLY SUPPORTED
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.
Edge Impulse
Add
MobileNetV2 96x96 0.1 OFFICIALLY SUPPORTED
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.
Edge Impulse
Add
MobileNetV2 96x96 0.05 OFFICIALLY SUPPORTED
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.
Edge Impulse
Add
MobileNetV2 160x160 1.0 OFFICIALLY SUPPORTED
Uses around 1.3M RAM and 2.6M ROM with default settings and optimizations. Works best with 160x160 input size. Supports RGB only.
Edge Impulse
Add
MobileNetV2 160x160 0.75 OFFICIALLY SUPPORTED
Uses around 1.3M RAM and 1.7M ROM with default settings and optimizations.
Edge Impulse
Add

Neural Network settings

Training settings

Number of training cycles: 20

Learning rate: 0.0005

Data augmentation: ☑

Advanced training settings

Validation set size: 20 %

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)

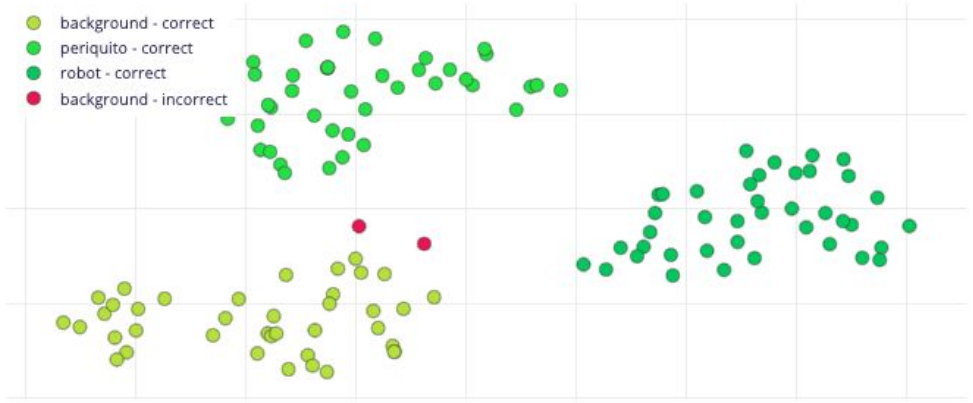
ACCURACY
92.0%

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)

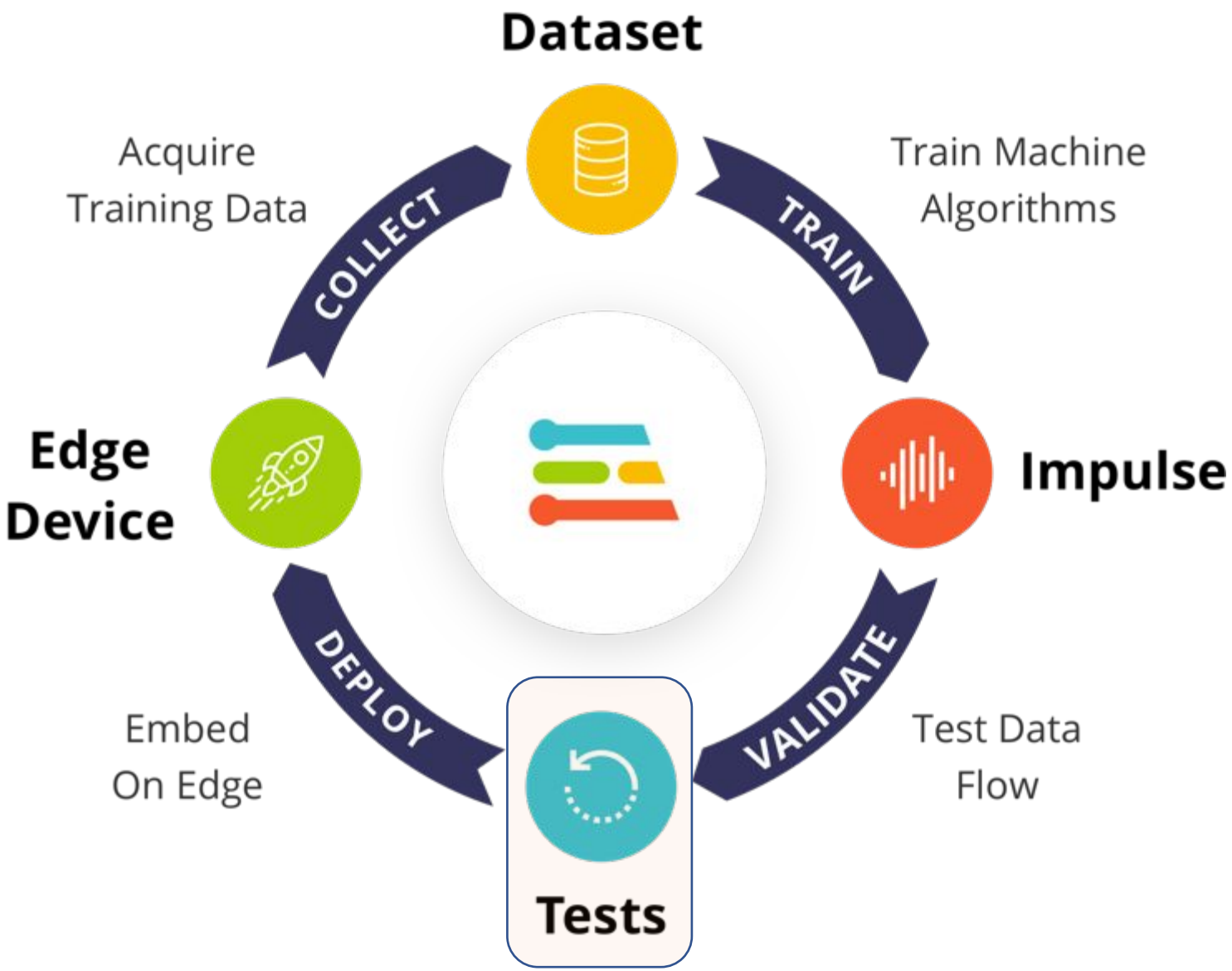


On-device performance

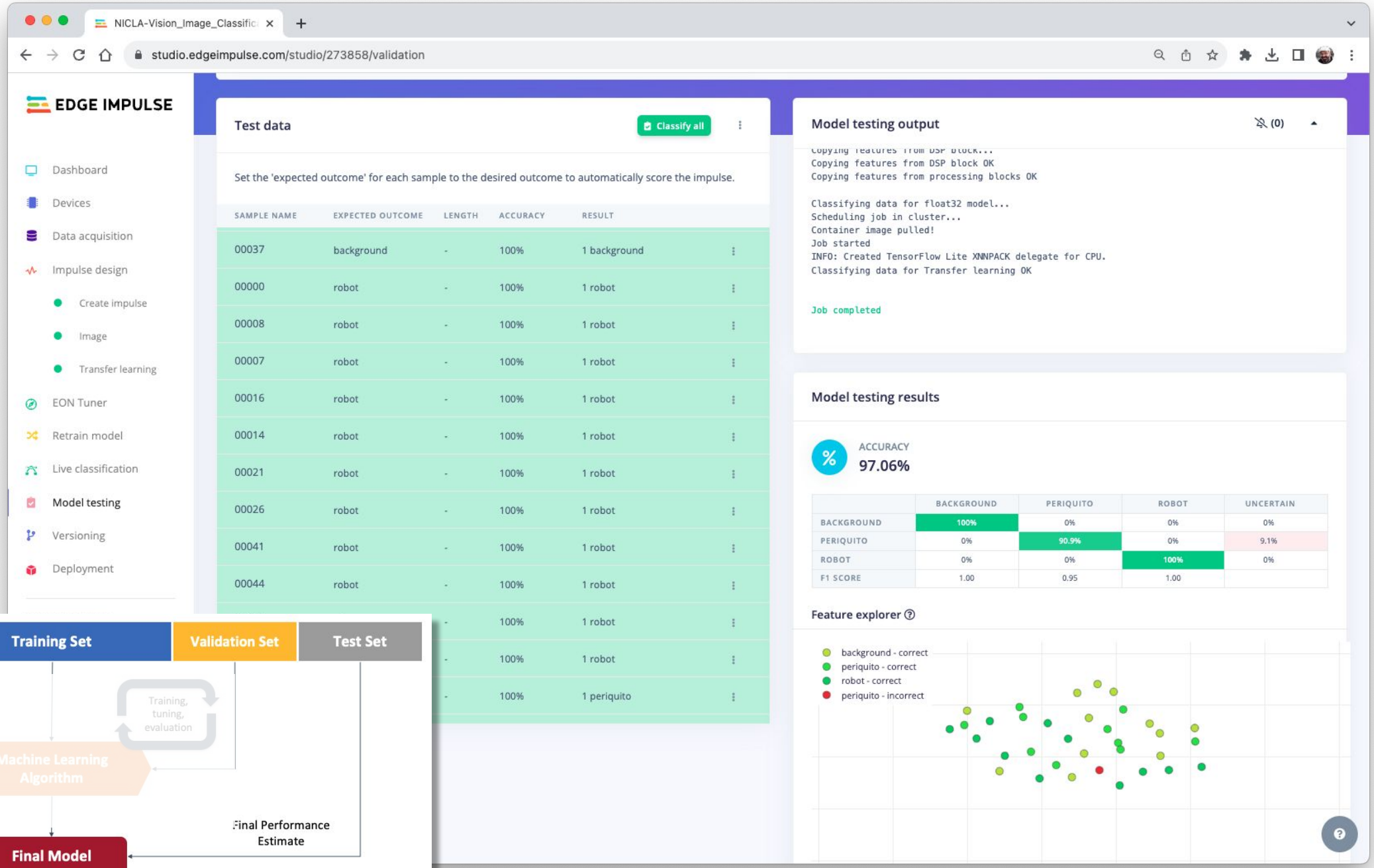
INFERENCING TIME
77 ms.

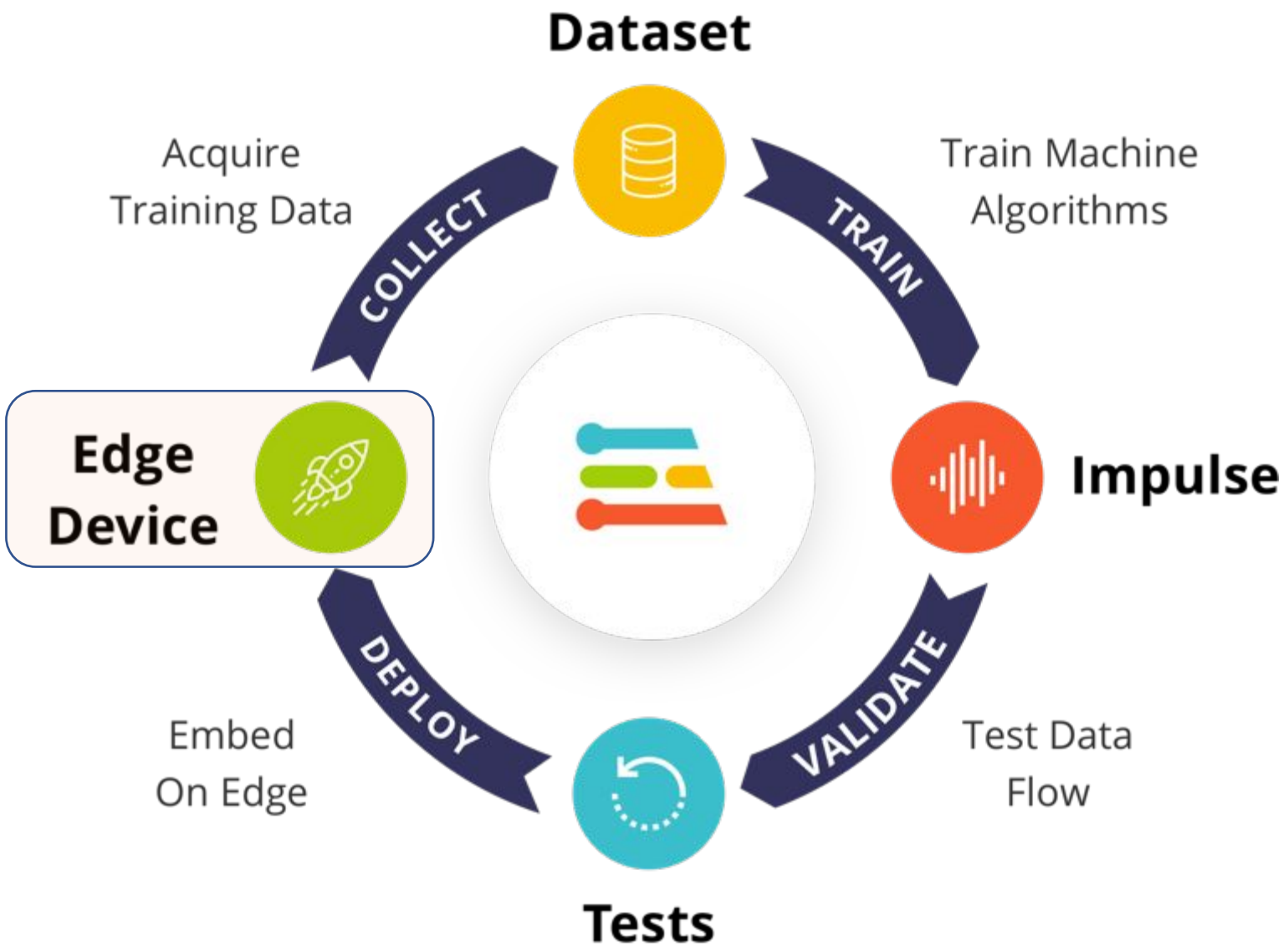
PEAK RAM USAGE
280.0K

FLASH USAGE
211.8K

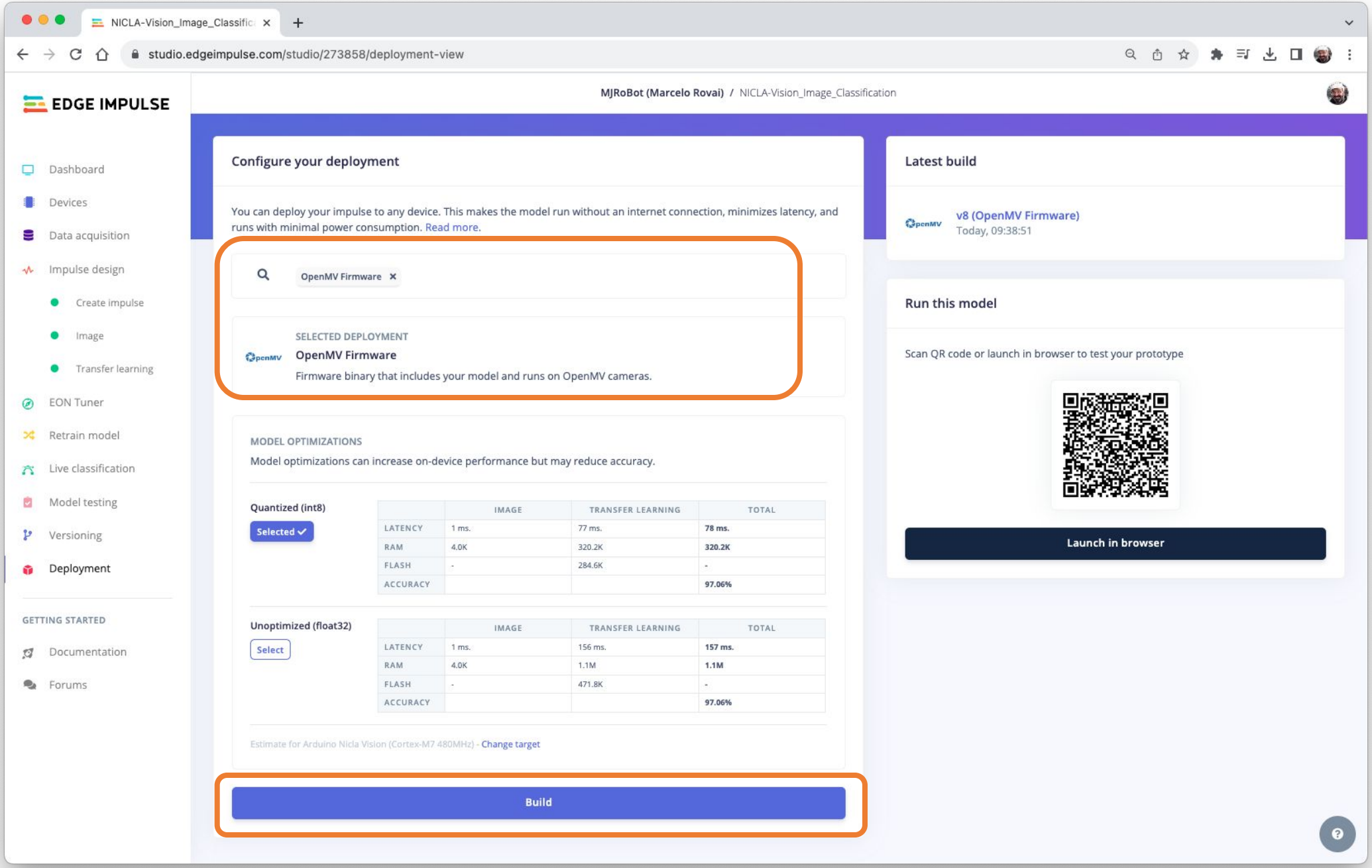


- Pre-Processing Data
- Design a Model
- Train a Model


NICLA-Vision_Image_Classific
studio.edgeimpulse.com/studio/273858/validation
EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
Test data
Classify all
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
(0)
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
background - correct
periquito - correct
robot - correct
periquito - incorrect
Training Set
Validation Set
Test Set
Training, tuning, evaluation
Machine Learning Algorithm
Final Performance Estimate
Final Model

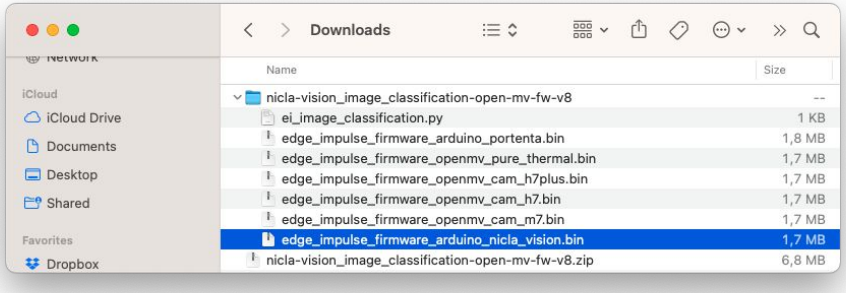


- Pre-Processing Data
- Design a Model
- Train a Model

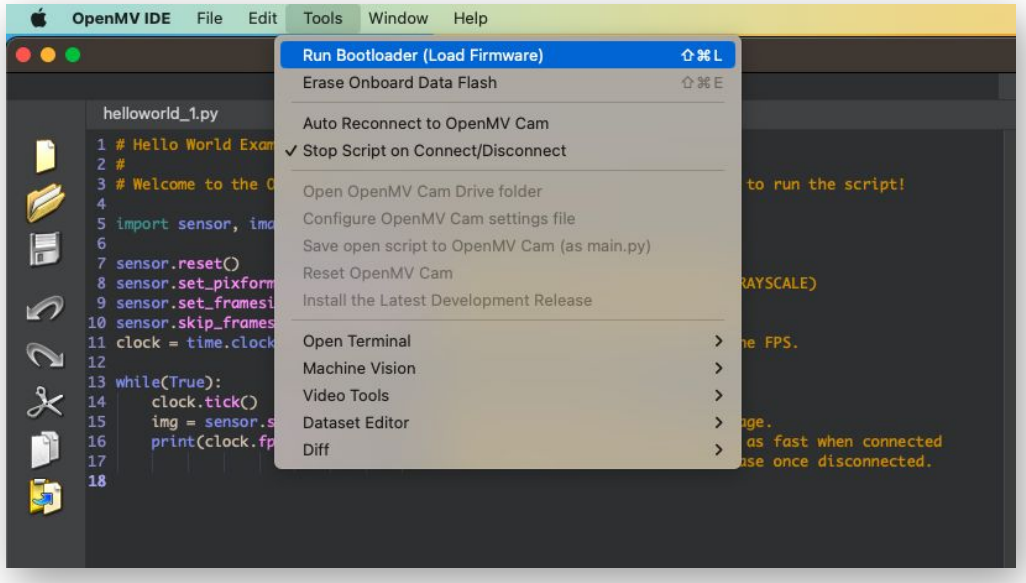

NICLA-Vision_Image_Classific
studio.edgeimpulse.com/studio/273858/deployment-view
EDGE IMPULSE
MJRoBot (Marcelo Rovai) / NICLA-Vision_Image_Classification
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
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
Launch in browser

Deploying model on OpenMV IDE

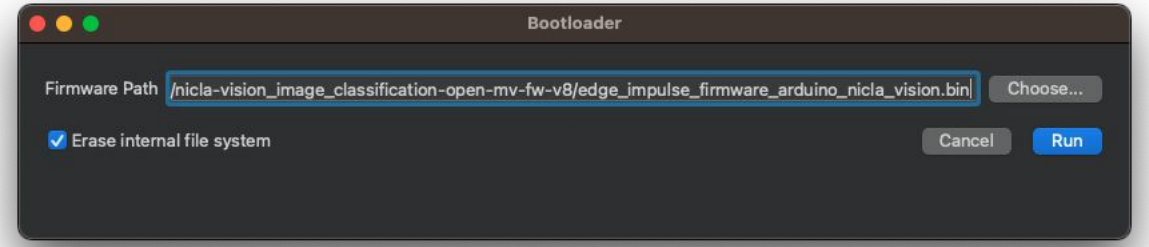
1. On your computer, you will find a ZIP file. Open it:

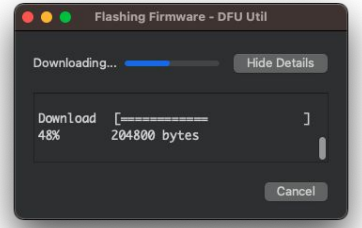


2. Use the Bootloader tool on OpenMV IDE to load it on your board:



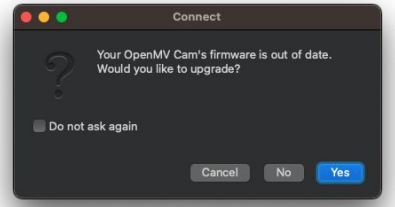
3. Select the appropriate file (.bin for Nicla-Vision) and press [Run]:



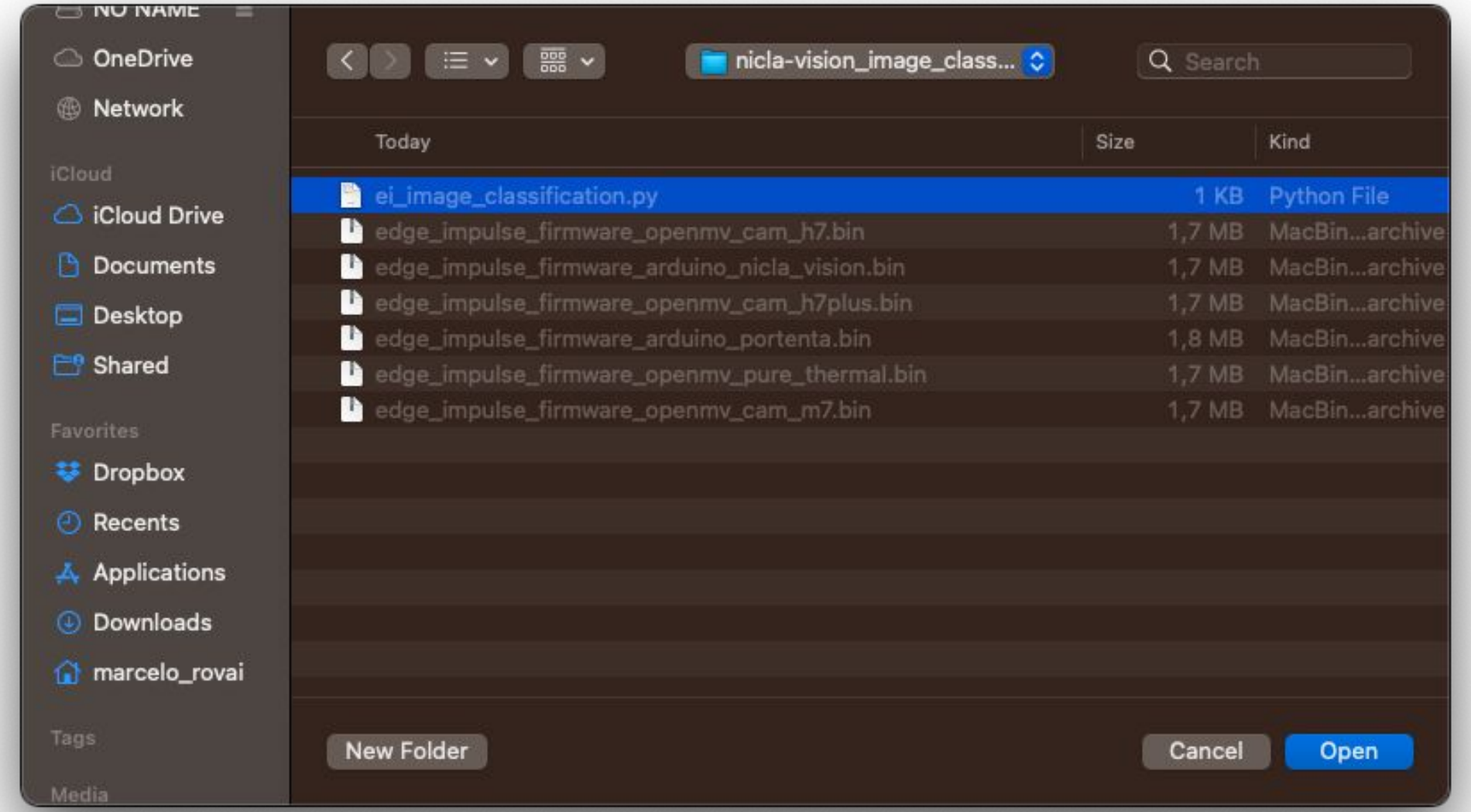


4. After the download is finished, press OK:

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

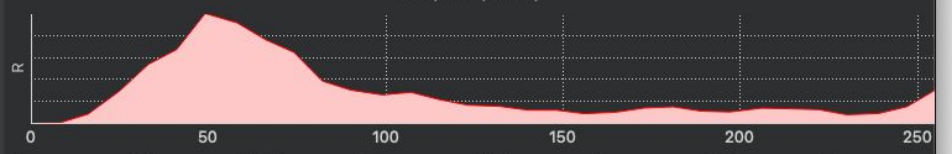
ei_image_classification.py — Line: 38, Col: 1

```python
# Edge Impulse - OpenMV Image Classification Example

import sensor, image, time, os, tf, uos, gc

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)


clock = time.clock()
while(True):
    clock.tick()

    img = sensor.snapshot()

    # default settings just do one detection... change them to search the image...
    for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        print("**********\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
        img.draw_rectangle(obj.rect())
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        for i in range(len(predictions_list)):
            print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))

    print(clock.fps(), "fps")
```

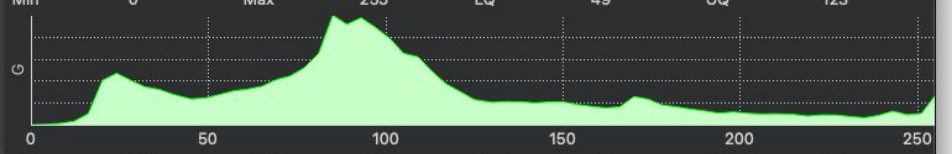
Frame Buffer — Record | Zoom | Disable

Histogram — RGB Color Space

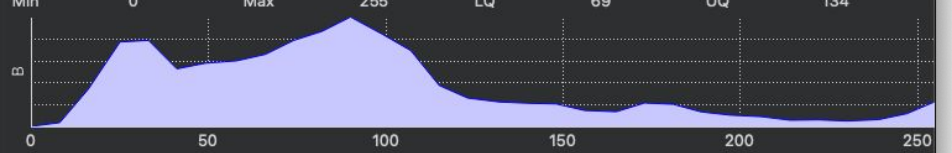
Res (w:240, h:240)



Mean	96	Median	74	Mode	49	StDev	65
Min	0	Max	255	LQ	49	UQ	123



Mean	106	Median	97	Mode	85	StDev	57
Min	0	Max	255	LQ	69	UQ	134



Mean	92	Median	82	Mode	90	StDev	56
Min	0	Max	255	LQ	49	UQ	115

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 updgrade] | Serial Port: cu.usbmodem3170375234301 | Drive: /Volumes/NO NAME | FPS: 7,4

ei_image_classification.py - OpenMV IDE

helloworld_1.py | periquitoVsrobot.py | ei_image_classification.py

ei_image_classification.py — Line: 38, Col: 1

```python
# Edge Impulse - OpenMV Image Classification Example

import sensor, image, time, os, tf, uos, gc

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)


clock = time.clock()
while(True):
    clock.tick()

    img = sensor.snapshot()

    # default settings just do one detection... change them to search the image...
    for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        print("**********\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
        img.draw_rectangle(obj.rect())
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        for i in range(len(predictions_list)):
            print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))

    print(clock.fps(), "fps")

```

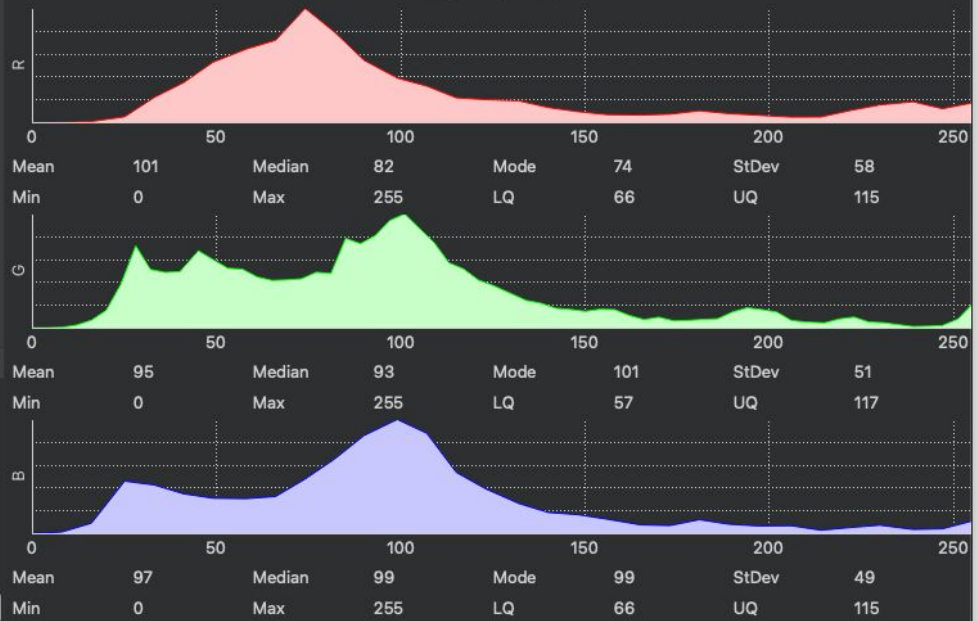
Serial Terminal

```
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
```

Frame Buffer — Record | Zoom | Disable

Histogram — RGB Color Space

Res (w:240, h:240)



Mean	101	Median	82	Mode	74	StDev	58
Min	0	Max	255	LQ	66	UQ	115
Mean	95	Median	93	Mode	101	StDev	51
Min	0	Max	255	LQ	57	UQ	117
Mean	97	Median	99	Mode	99	StDev	49
Min	0	Max	255	LQ	66	UQ	115

Search Results | Serial Terminal | Board: Arduino Nicla Vision | Sensor: GC2145 | Firmware Version: 4.3.1 - [out of date - click here to updgrade] | Serial Port: cu.usbmodem3170375234301 | Drive: /Volumes/NO NAME | FPS: 7,3

ei_image_classification.py - OpenMV IDE

helloworld_1.py | periquitoVsrobot.py | ei_image_classification.py

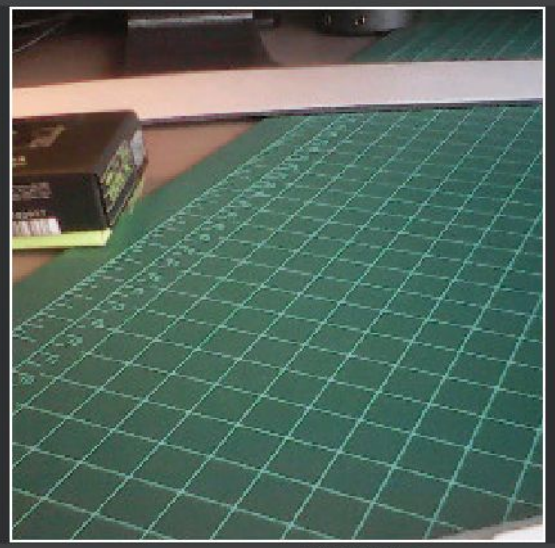
ei_image_classification.py — Line: 38, Col: 1

```python
# Edge Impulse - OpenMV Image Classification Example

import sensor, image, time, os, tf, uos, gc

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)


clock = time.clock()
while(True):
    clock.tick()

    img = sensor.snapshot()

    # default settings just do one detection... change them to search the image...
    for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        print("**********\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % obj.rect())
        img.draw_rectangle(obj.rect())
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        for i in range(len(predictions_list)):
            print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))

    print(clock.fps(), "fps")

```

Serial Terminal

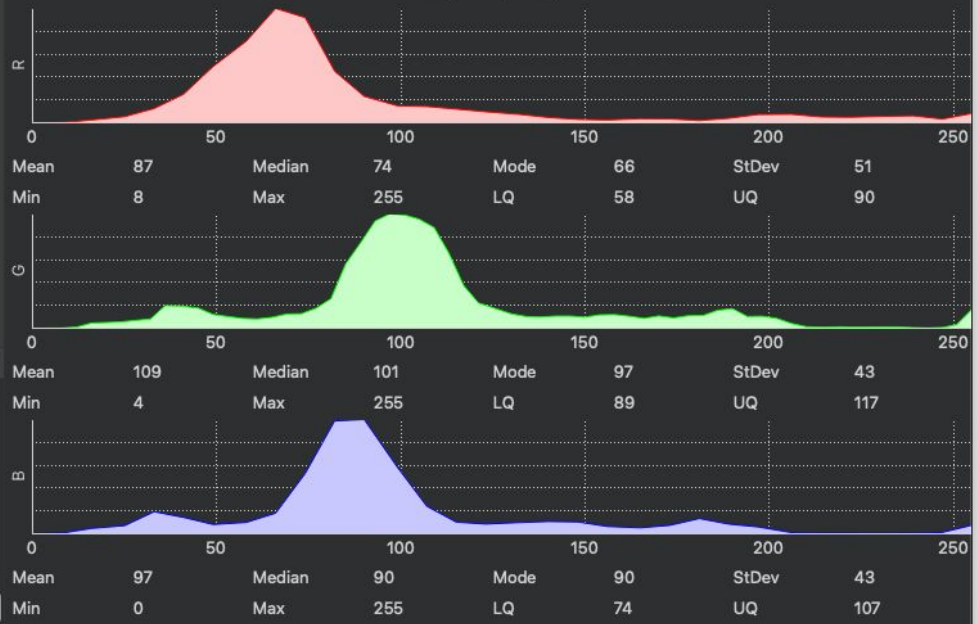
```
Predictions at [x=0,y=0,w=240,h=240]
background = 0.957031
periquito = 0.039063
robot = 0.003906
7.33808 fps
**********
Predictions at [x=0,y=0,w=240,h=240]
background = 0.933594
periquito = 0.062500
robot = 0.003906
7.33809 fps
```

Frame Buffer — Record | Zoom | Disable



Histogram — RGB Color Space

Res (w:240, h:240)



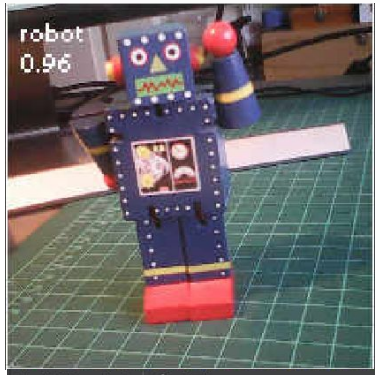
Search Results | Serial Terminal | Board: Arduino Nicla Vision | Sensor: GC2145 | Firmware Version: 4.3.1 - [out of date - click here to updgrade] | Serial Port: cu.usbmodem3170375234301 | Drive: /Volumes/NO NAME | FPS: 7,4

Changing Code to add labels:

```python
# Marcelo Rovai - NICLA Vision - Image Classification
# Adapted from Edge Impulse - OpenMV Image Classification Example
# @24Aug23

import sensor, image, time, os, tf, uos, gc

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)

clock = time.clock()


```

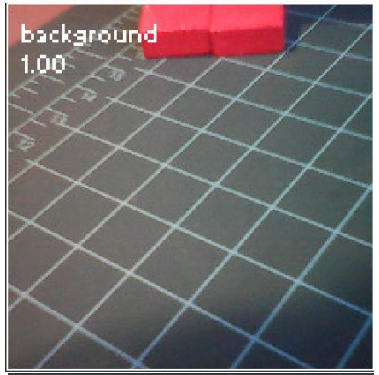
Serial Terminal

```
**********
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
```

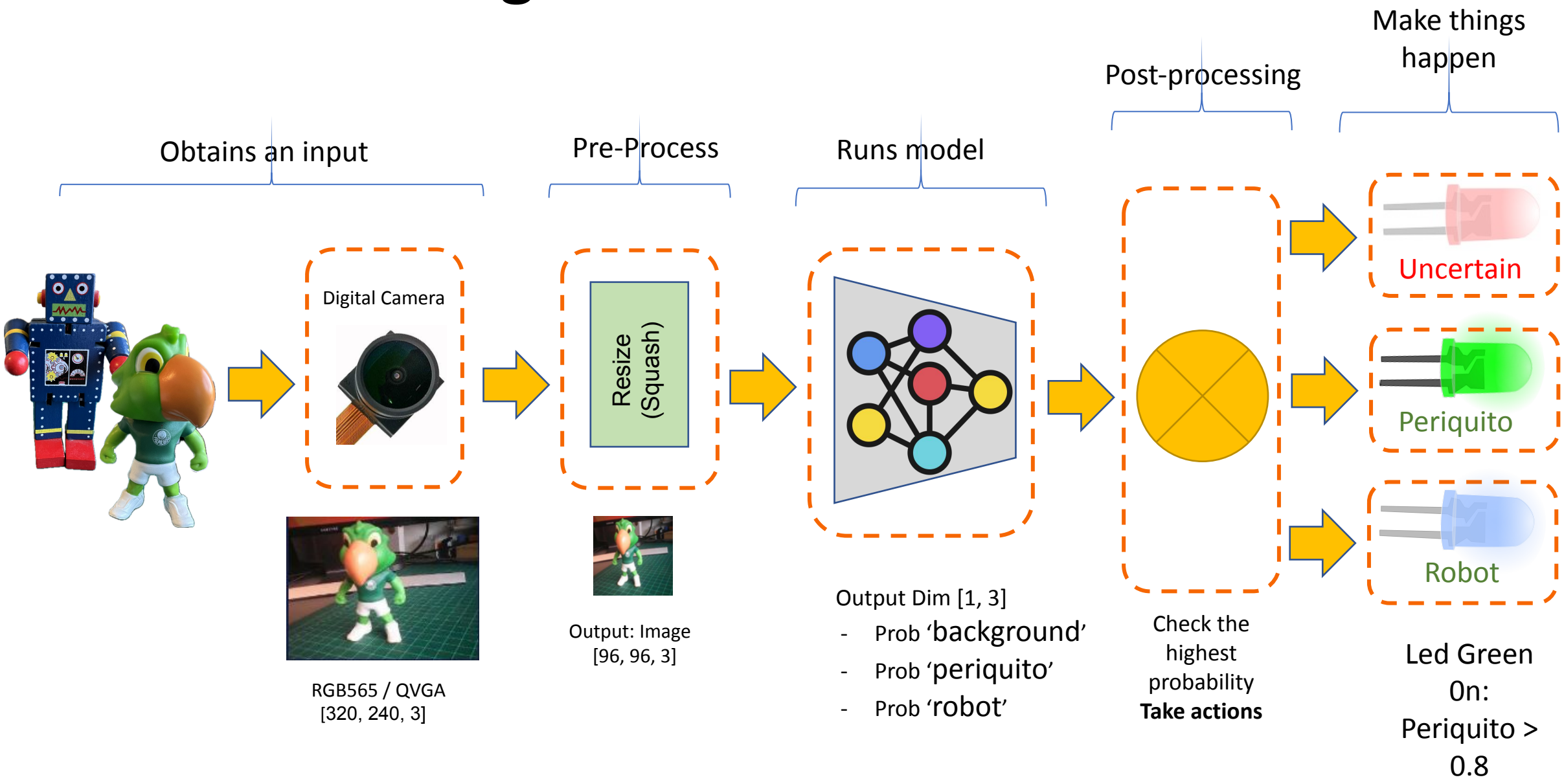


Serial Terminal

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

```python
while(True):
    clock.tick()  # Starts tracking elapsed time.

    img = sensor.snapshot()

    # default settings just do one detection... change them to search the image...
    for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        fps = clock.fps()
        lat = clock.avg()

        print("**********\nPrediction:")
        img.draw_rectangle(obj.rect())
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        max_val = predictions_list[0][1]
        max_lbl = 'background'
        for i in range(len(predictions_list)):
            val = predictions_list[i][1]
            lbl = predictions_list[i][0]
            #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))

            if val > max_val:
                max_val = val
                max_lbl = lbl

    # Print label with the highest probability
    if max_val < 0.5:
        max_lbl = 'uncertain'

    #print("%s with a prob of %f" % (max_lbl, max_val))
    print("{} with a prob of {:.2f}".format(max_lbl, max_val))
    # print(clock.fps(), "fps")
    print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))

    # Draw label with highest probability to image viewer
    img.draw_string(
        10, 10,
        max_lbl + "\n{:.2f}".format(max_val),
        mono_space = False,
        scale=2
        )

```

Post-Processing with LEDs:



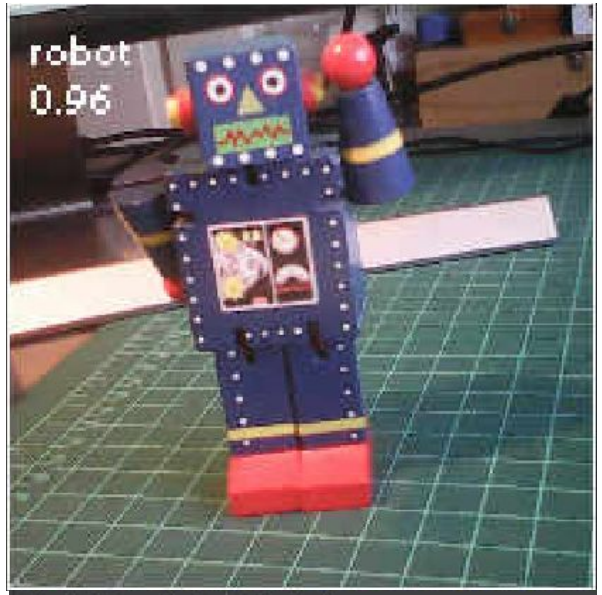
Post-Processing with LEDs:

nicla_image_classification_LED.py

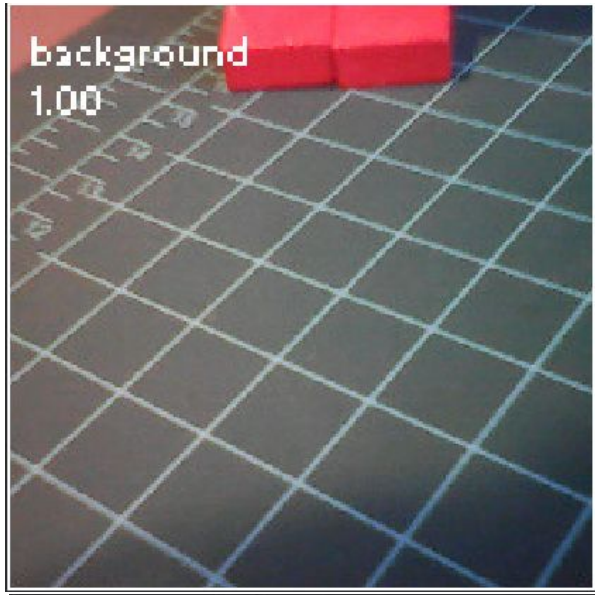
```python
# Marcelo Rovai - NICLA Vision - Image Classification with LEDs
# Adapted from Edge Impulse - OpenMV Image Classification Example
# @24Aug23

import sensor, image, time, os, tf, uos, gc, pyb

ledRed = pyb.LED(1)
ledGre = pyb.LED(2)
ledBlu = pyb.LED(3)

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

ledRed.off()
ledGre.off()
ledBlu.off()

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)

clock = time.clock()


def setLEDs(max_lbl):

    if max_lbl == 'uncertain':
        ledRed.on()
        ledGre.off()
        ledBlu.off()

    if max_lbl == 'periquito':
        ledRed.off()
        ledGre.on()
        ledBlu.off()

    if max_lbl == 'robot':
        ledRed.off()
        ledGre.off()
        ledBlu.on()

    if max_lbl == 'background':
        ledRed.off()
        ledGre.off()
        ledBlu.off()
```

```python


while(True):
    clock.tick()  # Starts tracking elapsed time.

    img = sensor.snapshot()

    # default settings just do one detection... change them to search the image...
    for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
        fps = clock.fps()
        lat = clock.avg()

        print("**********\nPrediction:")
        img.draw_rectangle(obj.rect())
        # This combines the labels and confidence values into a list of tuples
        predictions_list = list(zip(labels, obj.output()))

        max_val = predictions_list[0][1]
        max_lbl = 'background'
        for i in range(len(predictions_list)):
            val = predictions_list[i][1]
            lbl = predictions_list[i][0]
            #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))

            if val > max_val:
                max_val = val
                max_lbl = lbl

    # Print label and turn on LED with the highest probability
    if max_val < 0.8:
        max_lbl = 'uncertain'

    setLEDs(max_lbl)


    #print("%s with a prob of %f" % (max_lbl, max_val))
    print("{} with a prob of {:.2f}".format(max_lbl, max_val))
    # print(clock.fps(), "fps")
    print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))

    # Draw label with highest probability to image viewer
    img.draw_string(
        10, 10,
        max_lbl + "\n{:.2f}".format(max_val),
        mono_space = False,
        scale=2
        )


```


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

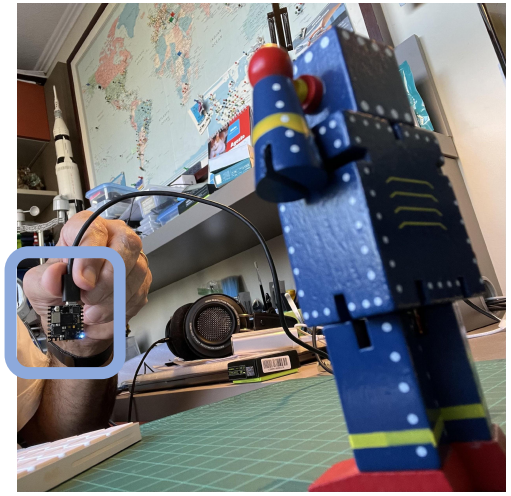


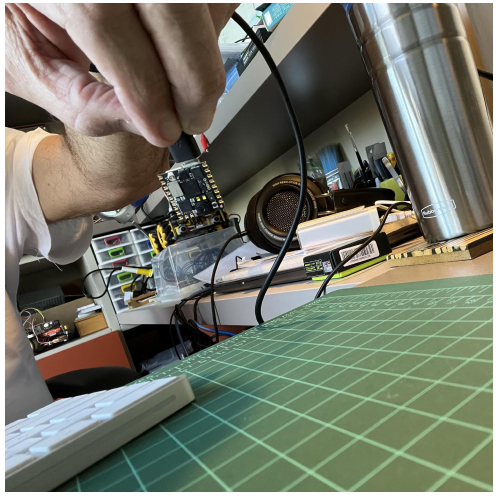

```
Serial Terminal
**********
Prediction:
robot with a prob of 0.96
FPS: 7.34 fps ==> latency: 136 ms
```

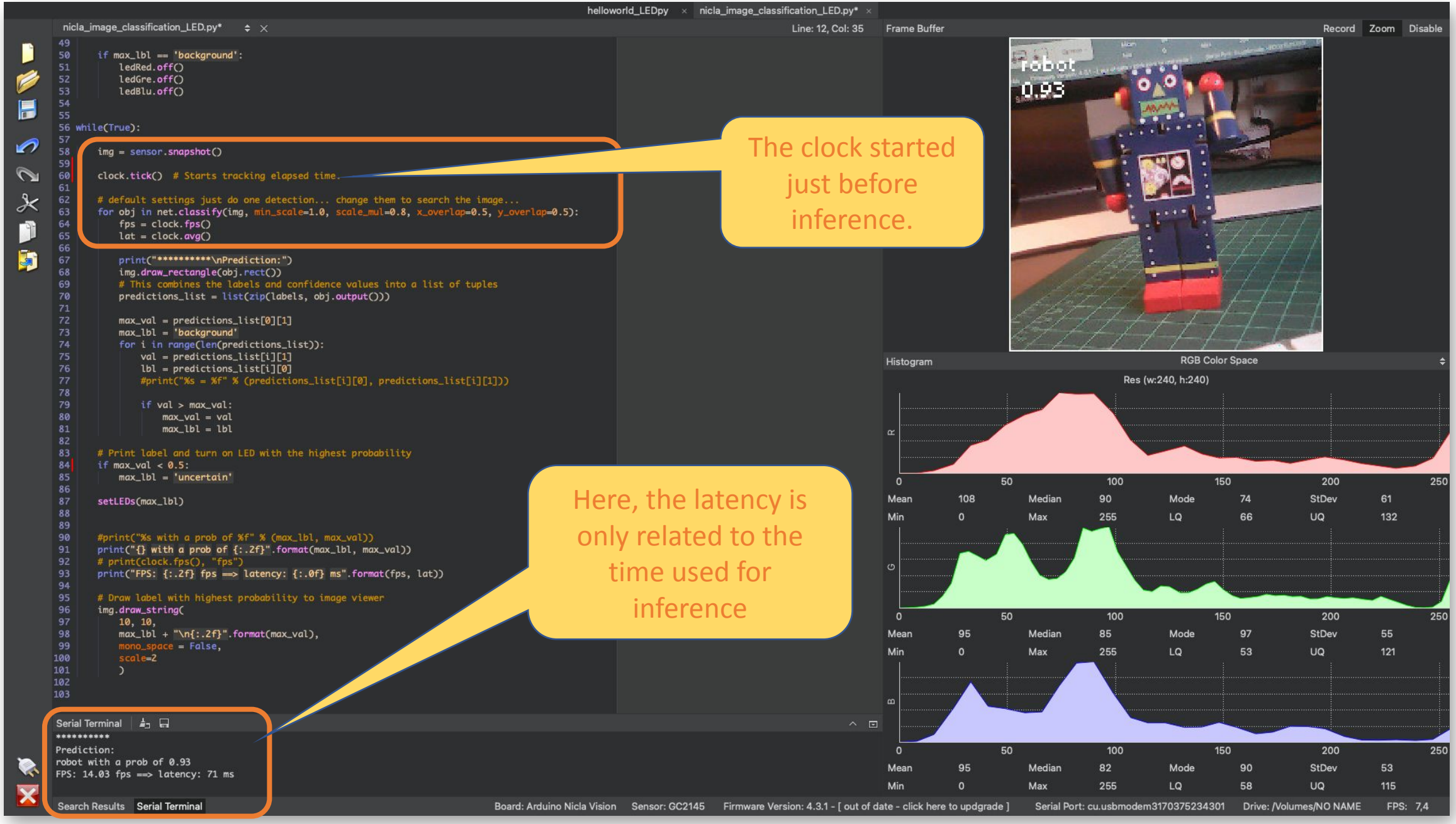



```
Serial Terminal
**********
Prediction:
background with a prob of 1.00
FPS: 7.34 fps ==> latency: 136 ms
```


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



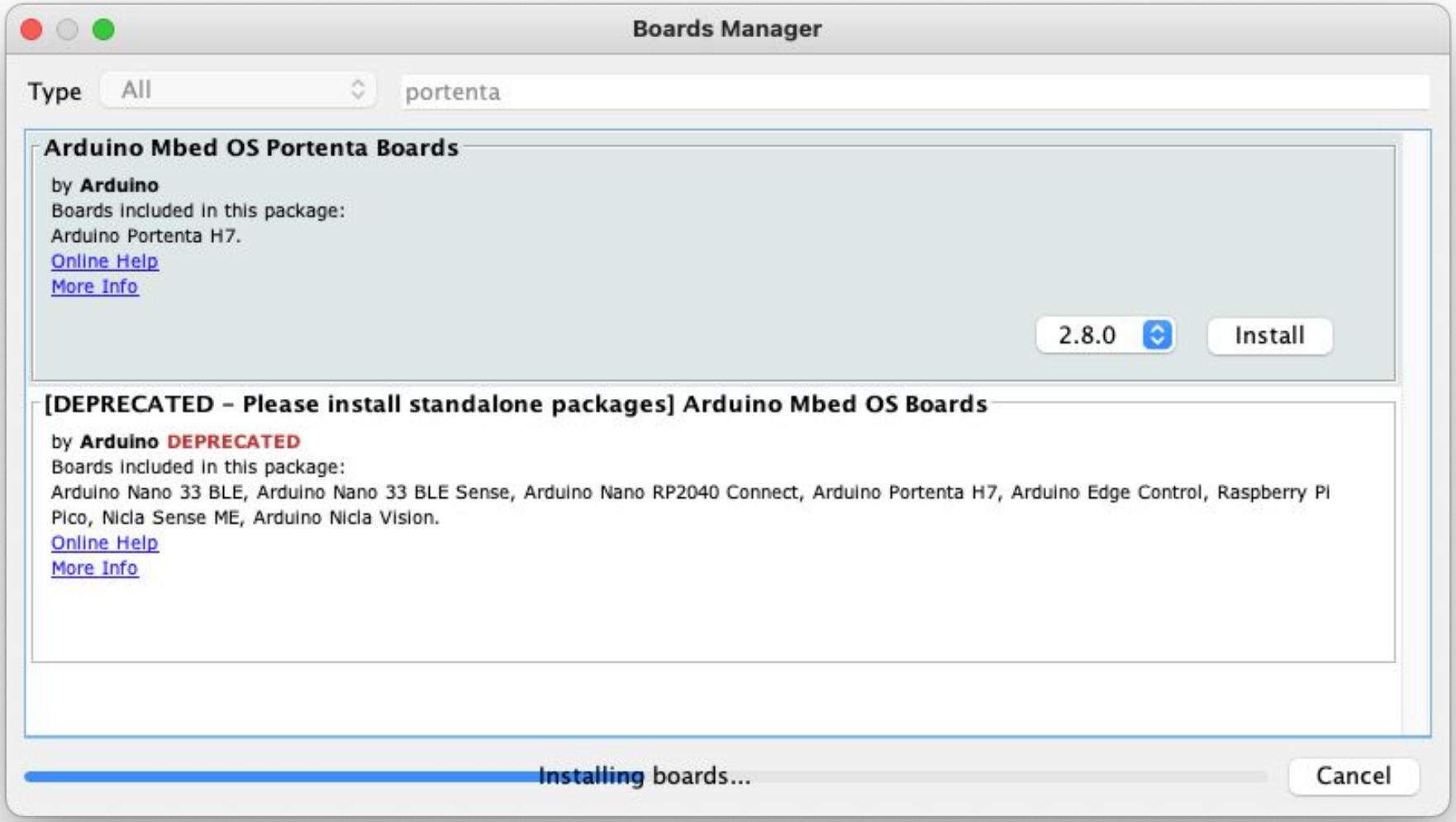





```
49
50      if max_lbl == 'background':
51          ledRed.off()
52          ledGre.off()
53          ledBlu.off()
54
55
56  while(True):
57
58      img = sensor.snapshot()
59
60      clock.tick()  # Starts tracking elapsed time.
61
62      # default settings just do one detection... change them to search the image...
63      for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
64          fps = clock.fps()
65          lat = clock.avg()
66
67          print("**********\nPrediction:")
68          img.draw_rectangle(obj.rect())
69          # This combines the labels and confidence values into a list of tuples
70          predictions_list = list(zip(labels, obj.output()))
71
72          max_val = predictions_list[0][1]
73          max_lbl = 'background'
74          for i in range(len(predictions_list)):
75              val = predictions_list[i][1]
76              lbl = predictions_list[i][0]
77              #print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
78
79              if val > max_val:
80                  max_val = val
81                  max_lbl = lbl
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
```

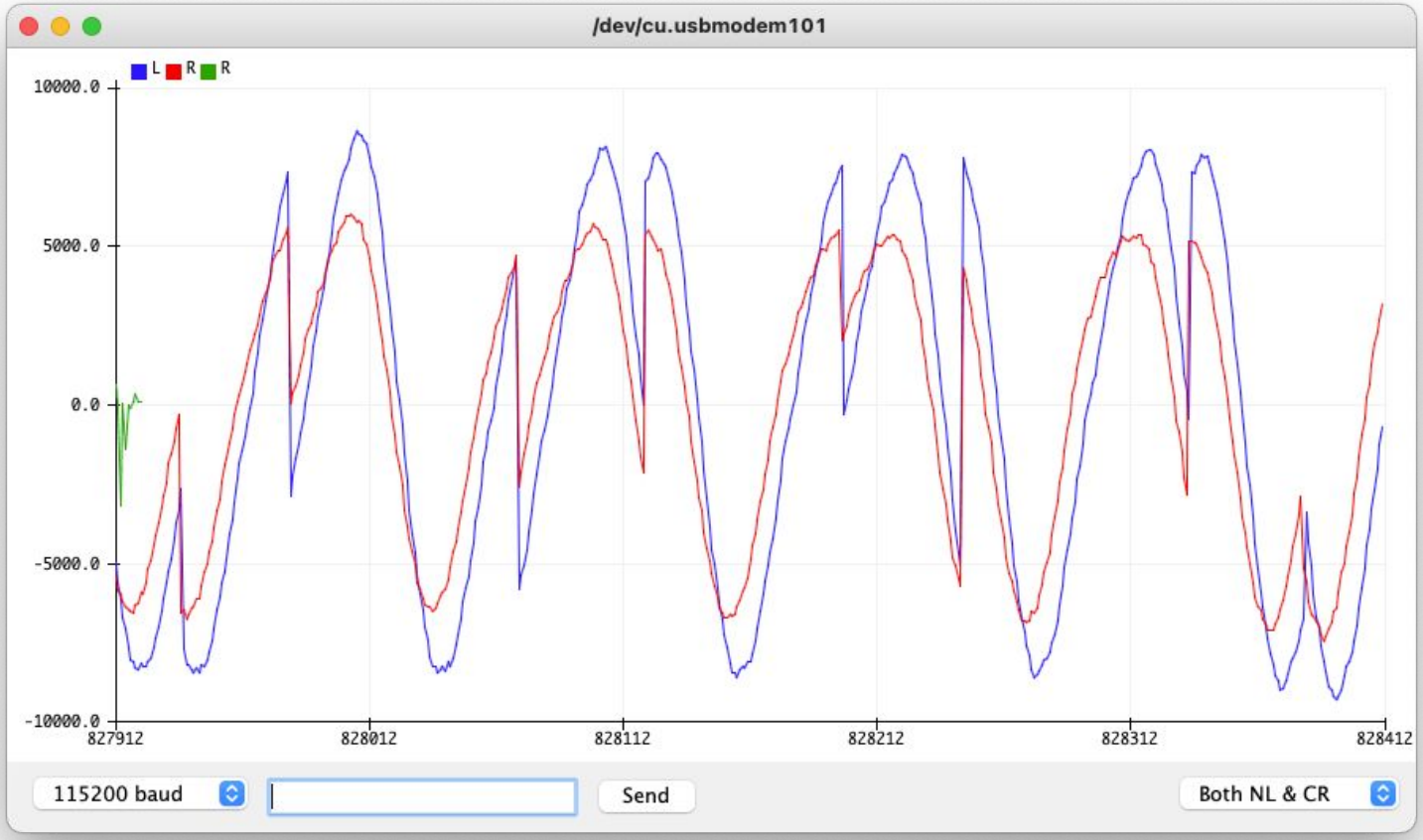
R	Mean	108	Median	90	Mode	74	StDev	61
	Min	0	Max	255	LQ	66	UQ	132
G	Mean	95	Median	85	Mode	97	StDev	55
	Min	0	Max	255	LQ	53	UQ	121
B	Mean	95	Median	82	Mode	90	StDev	53
	Min	0	Max	255	LQ	58	UQ	115

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



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

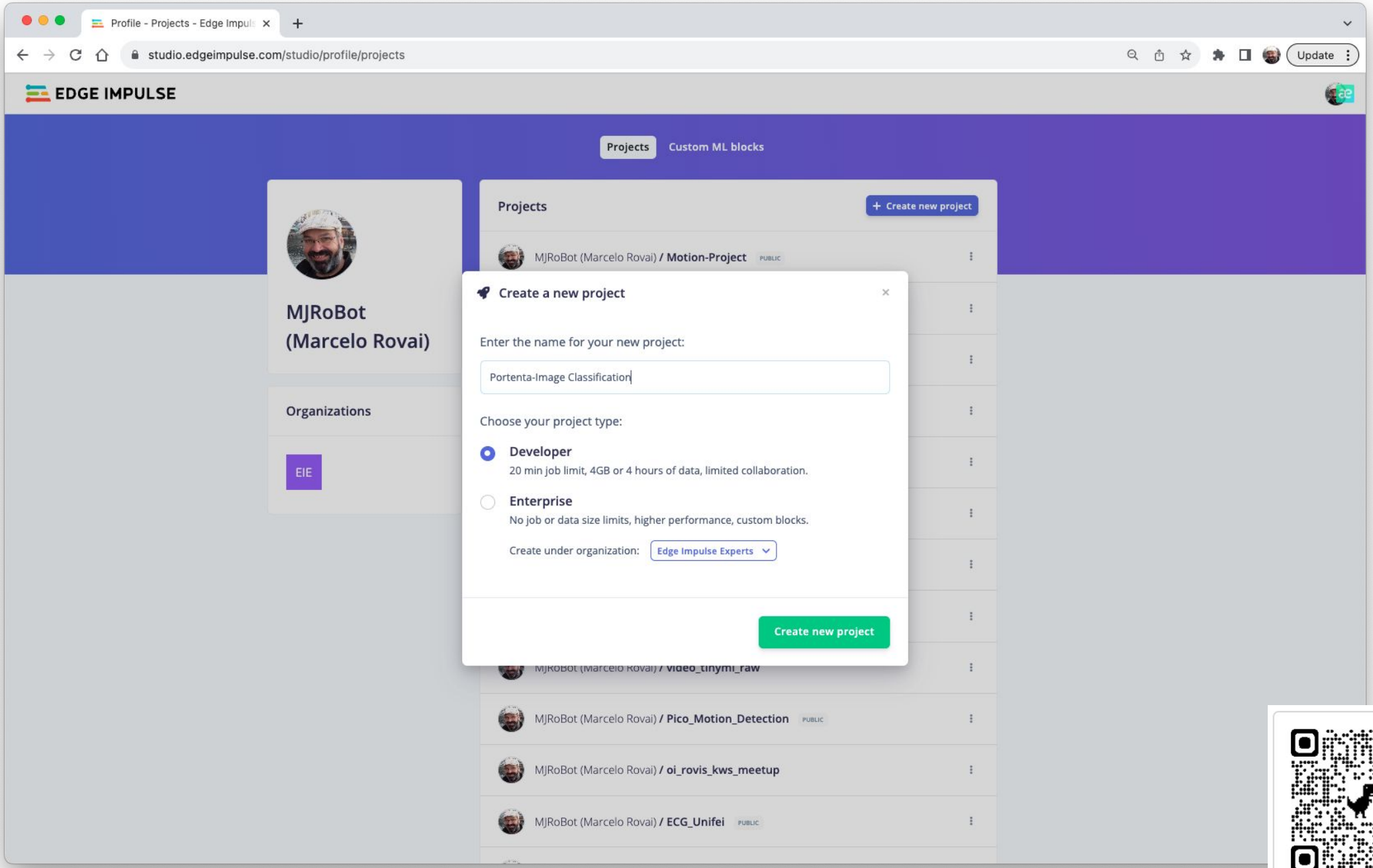
Microphones Test



Go to Examples > PDM > PDMSerialPlotter

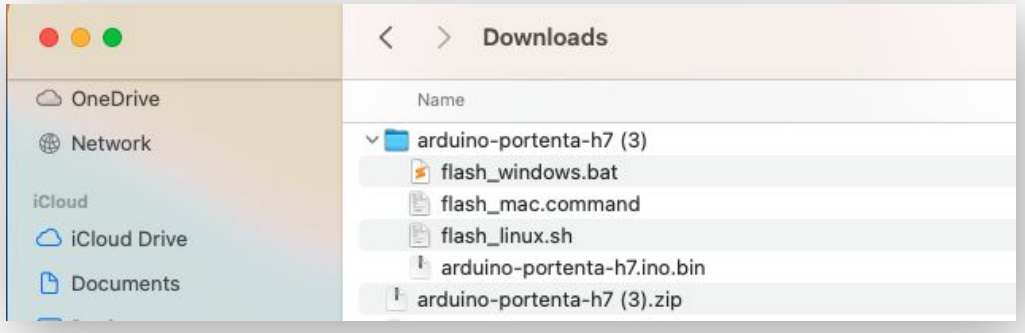
and test the 2 channels (left and right) and sample freq. of 32KHz

```
// 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 Impuls
studio.edgeimpulse.com/studio/profile/projects
Update
EDGE IMPULSE
Projects
Custom ML blocks
MJRoBot (Marcelo Rovai)
Organizations
EIE
Projects
+ Create new project
MJRoBot (Marcelo Rovai) / Motion-Project PUBLIC
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) / Pico_Motion_Detection PUBLIC
MJRoBot (Marcelo Rovai) / oi_rovis_kws_meetup
MJRoBot (Marcelo Rovai) / ECG_Unifei PUBLIC

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 — 108×53

```
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\ \(3\)/flas
h_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

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

Update

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

GETTING STARTED

- Documentation
- Forums

#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

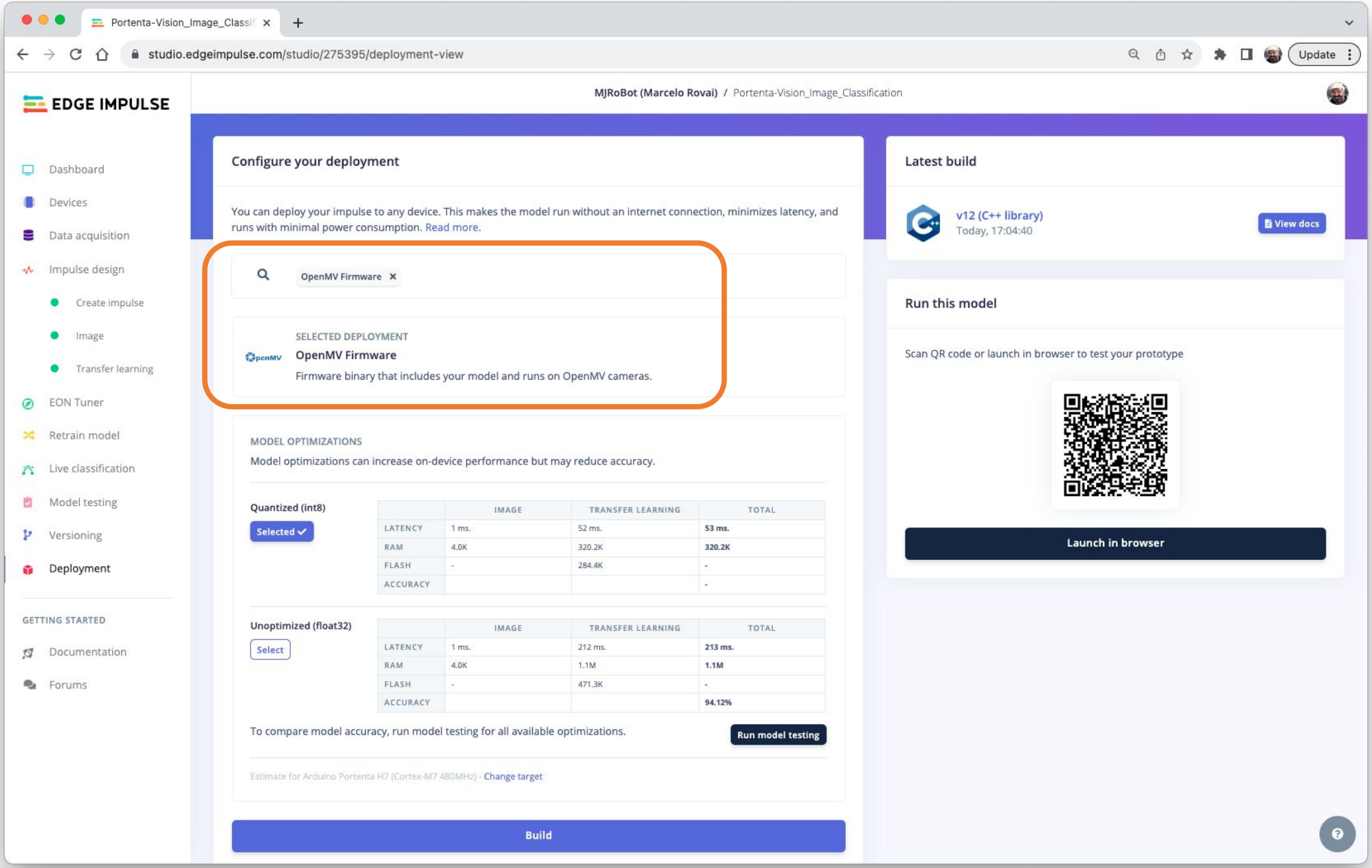
0.2736, 0.2780, 0.2879, 0.2970, 0.3016, 0.3177, 0.3327, 0.3339, 0.3460, 0.3522, 0.3583, 0.3628,…

On-device performance

PROCESSING TIME
1 ms.

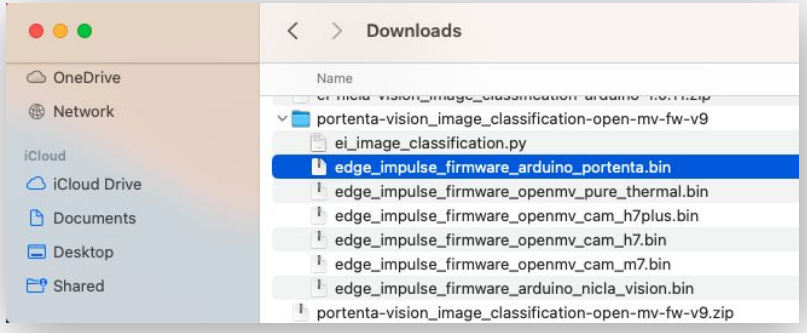
PEAK RAM USAGE
4 KB

© 2023 EdgeImpulse Inc. All rights reserved

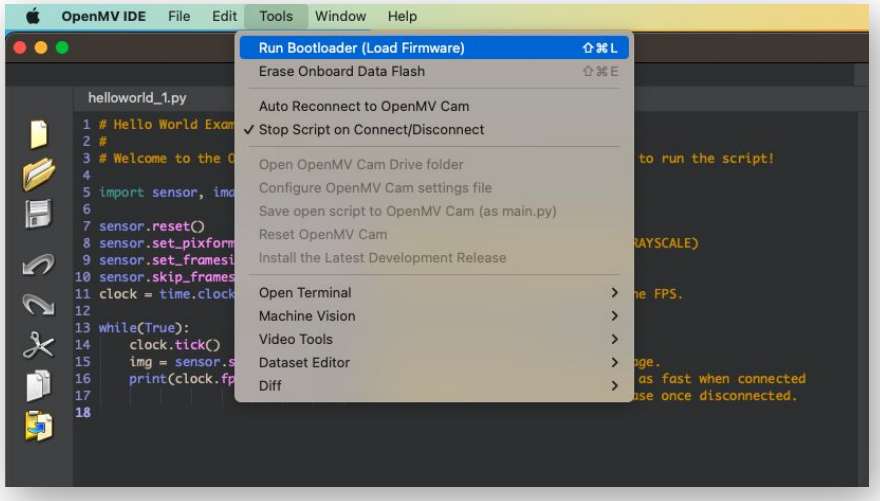

Portenta-Vision_Image_Classif
studio.edgeimpulse.com/studio/275395/deployment-view
Update
EDGE IMPULSE
MJRoBot (Marcelo Rovai) / Portenta-Vision_Image_Classification
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Image
Transfer learning
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
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. 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
Launch in browser

Deploying model on OpenMV IDE

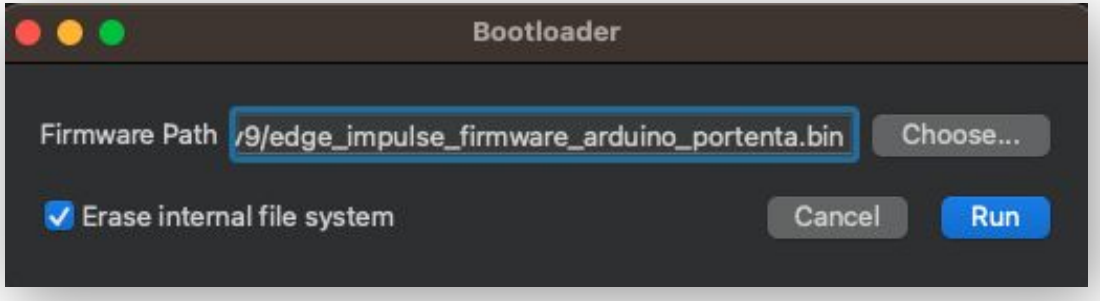
1. On your computer, you will find a ZIP file. Open it:

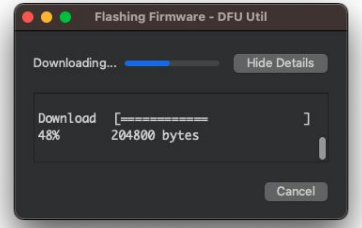


2. Use the Bootloader tool on OpenMV IDE to load it on your board:



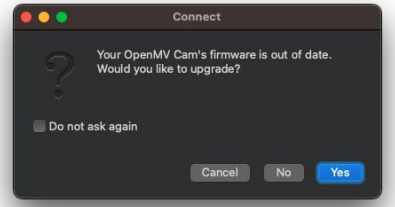
3. Select the appropriate file (.bin for Portenta) and press [Run]:





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*

```python
# Marcelo Rovai - NICLA Vision - Image Classification with LEDs
# Adapted from Edge Impulse - OpenMV Image Classification Example
# @24Aug23

import sensor, image, time, os, tf, uos, gc, pyb

ledRed = pyb.LED(1)
ledGre = pyb.LED(2)
ledBlu = pyb.LED(3)

sensor.reset()                         # Reset and initialize the sensor.
sensor.set_pixformat(sensor.GRAYSCALE)    # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)      # Set frame size to QVGA (320x240)
sensor.set_windowing((240, 240))       # Set 240x240 window.
sensor.skip_frames(time=2000)          # Let the camera adjust.

net = None
labels = None

ledRed.off()
ledGre.off()
ledBlu.off()

try:
    # Load built in model
    labels, net = tf.load_builtin_model('trained')
except Exception as e:
    raise Exception(e)

clock = time.clock()


def setLEDs(max_lbl):

    if max_lbl == 'uncertain':
        ledRed.on()
        ledGre.off()
        ledBlu.off()

    if max_lbl == 'periquito':
        ledRed.off()
        ledGre.on()
        ledBlu.off()

    if max_lbl == 'robot':
        ledRed.off()
        ledGre.off()
        ledBlu.on()

    if max_lbl == 'background':
        ledRed.off()
        ledGre.off()
        ledBlu.off()


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)

helloworld_LEDpy | nicla_image_classification_LED.py*

nicla_image_classification_LED.py* — Line: 86, Col: 21

```
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
```

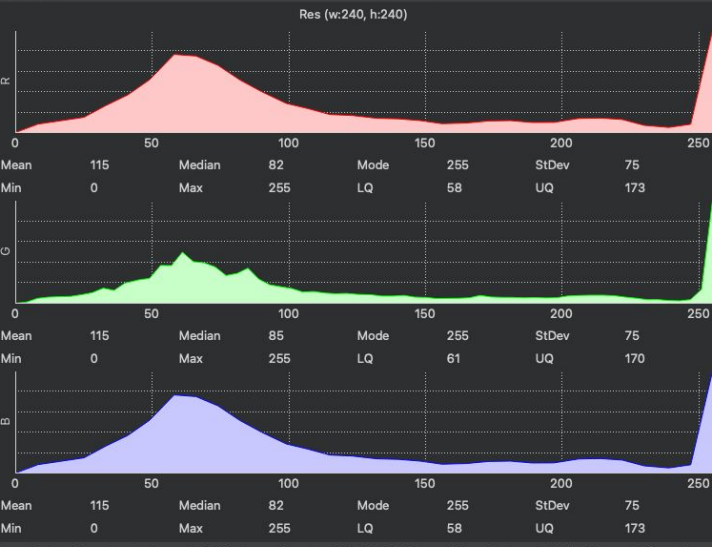
Serial Terminal

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

Frame Buffer — Record | Zoom | Disable

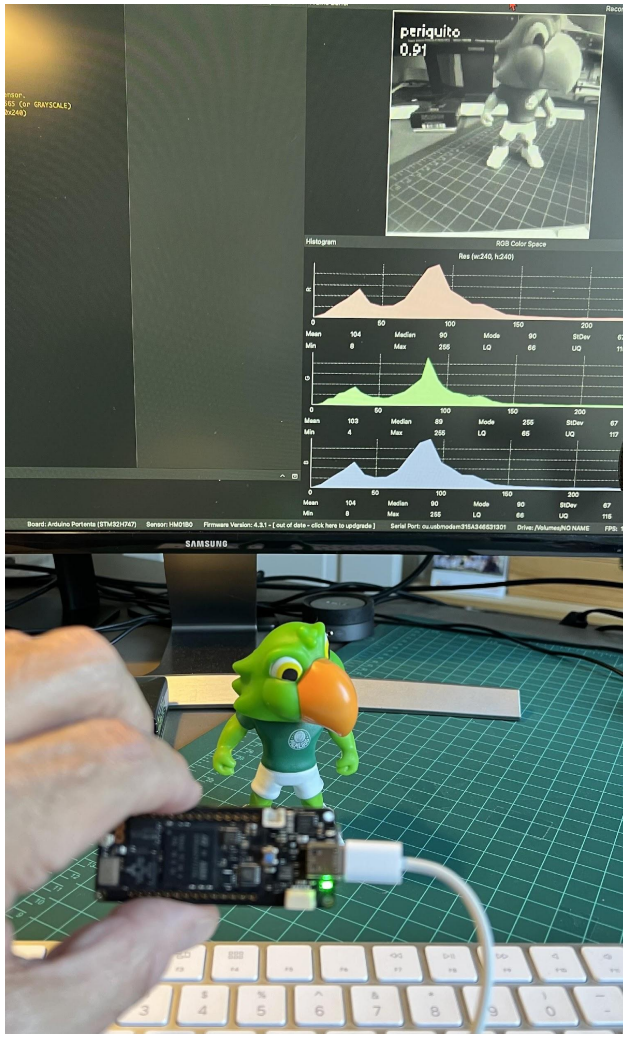
Histogram — RGB Color Space

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

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



helloworld_LEDpy | nicla_image_classification_LED.py*

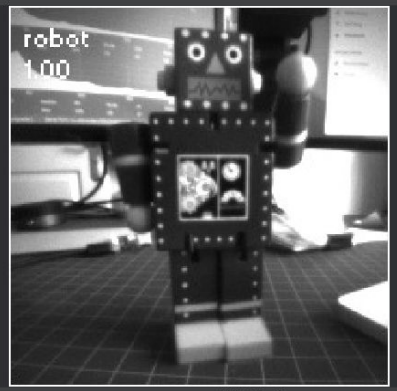
nicla_image_classification_LED.py* — Line: 86, Col: 21

```
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

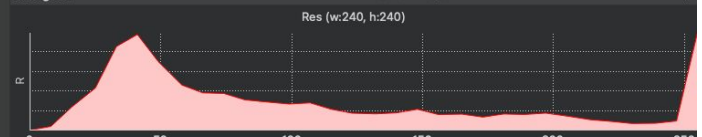
```
**********
Prediction:
robot with a prob of 1.00
FPS: 14.31 fps ==> latency: 70 ms
```

Frame Buffer — Record | Zoom | Disable

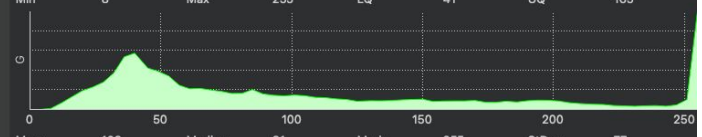


Histogram — RGB Color Space

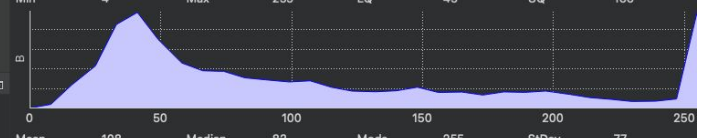
Res (w:240, h:240)



Mean	108	Median	82	Mode	255	StDev	77
Min	8	Max	255	LQ	41	UQ	165

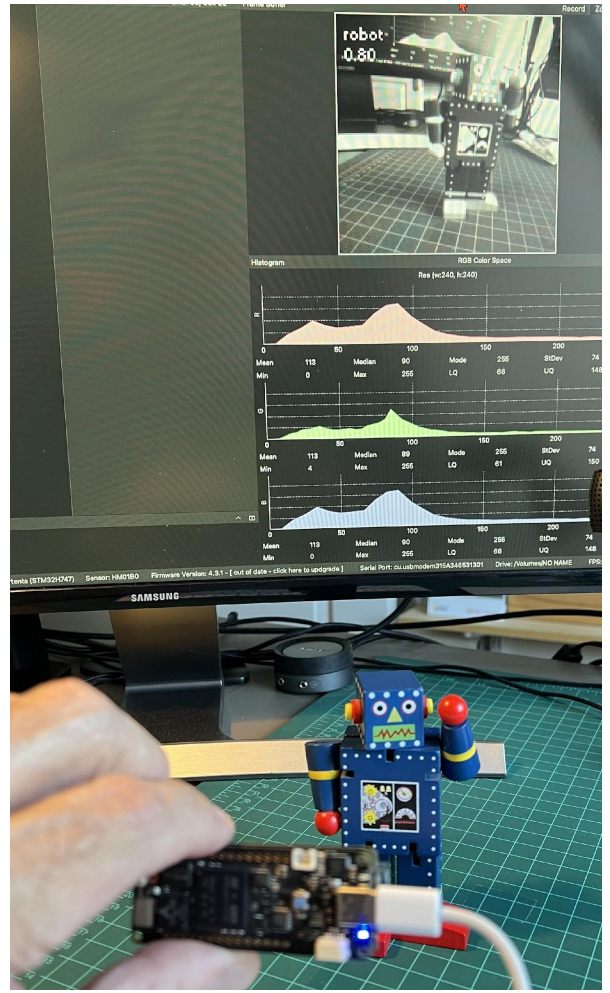


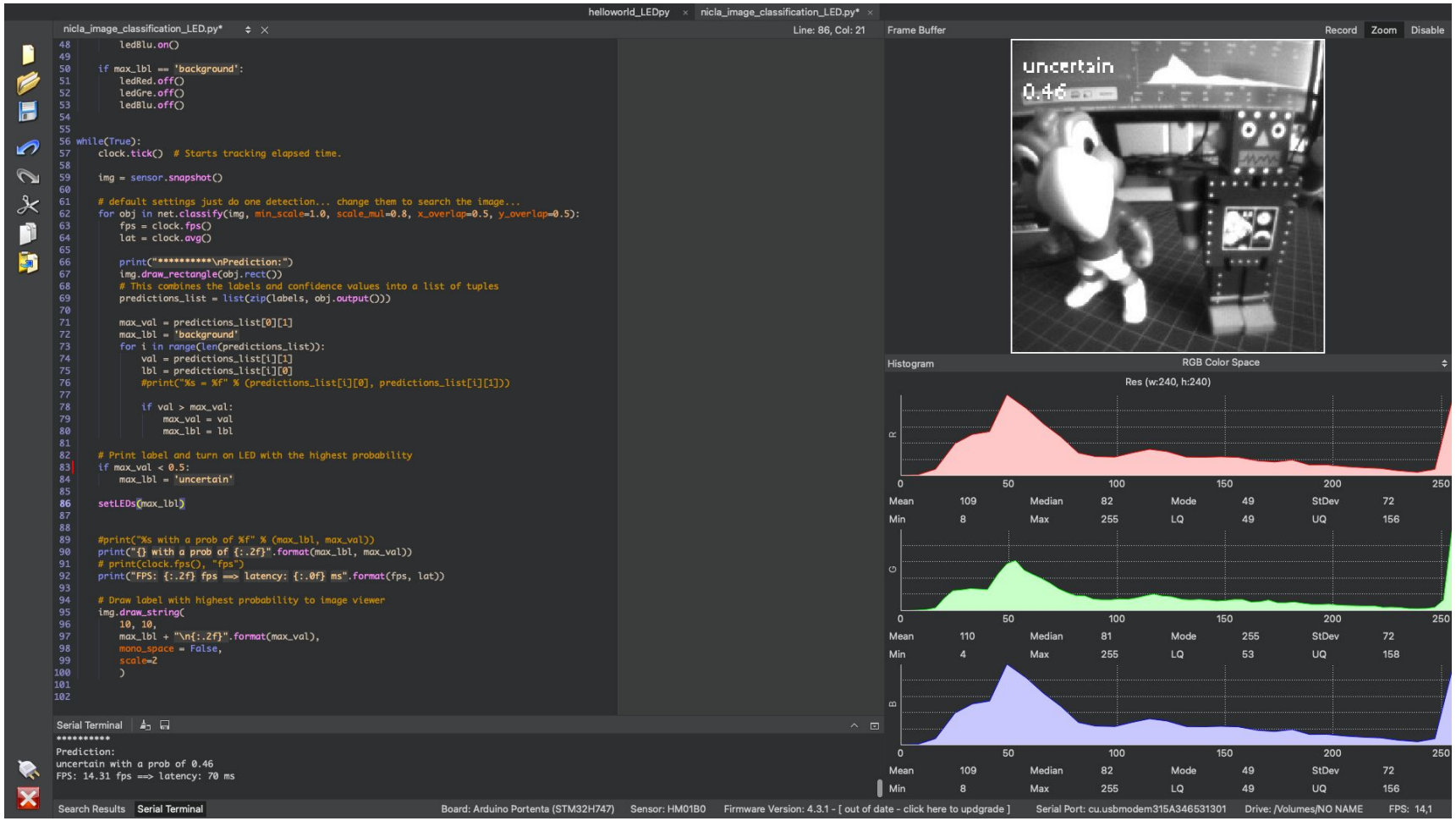
Mean	108	Median	81	Mode	255	StDev	77
Min	4	Max	255	LQ	45	UQ	166

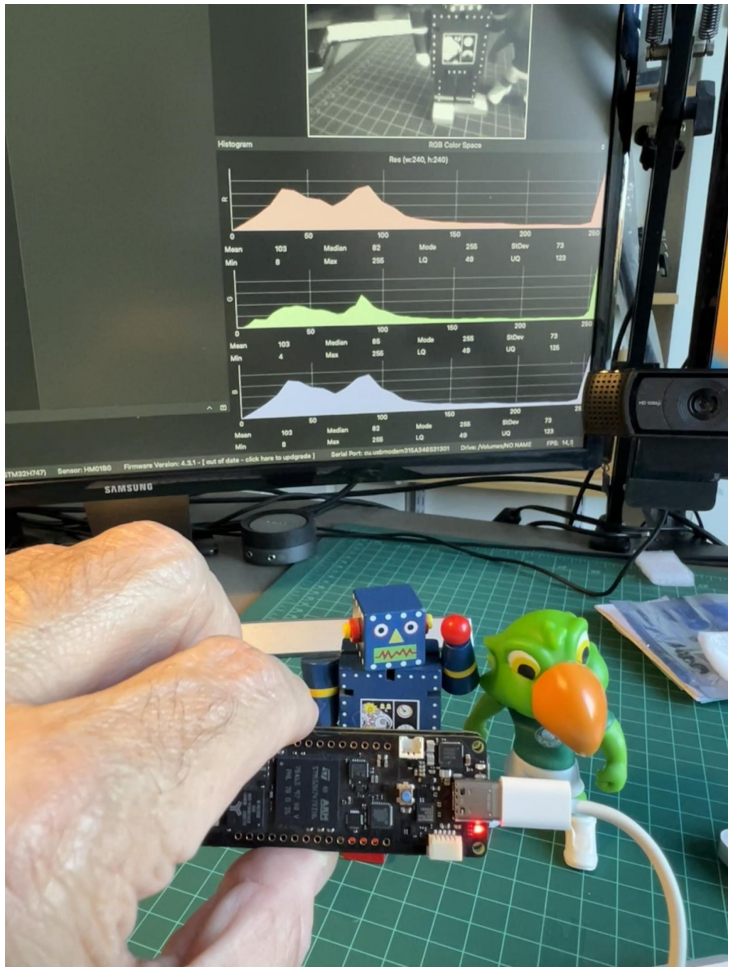


Mean	108	Median	82	Mode	255	StDev	77
Min	8	Max	255	LQ	41	UQ	165

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




helloworld_LEDpy
nicla_image_classification_LED.py*
nicla_image_classification_LED.py*
Line: 86, Col: 21
ledBlu.on()
if max_lbl == 'background':
ledRed.off()
ledGre.off()
ledBlu.off()
while(True):
clock.tick() # Starts tracking elapsed time.
img = sensor.snapshot()
# default settings just do one detection... change them to search the image...
for obj in net.classify(img, min_scale=1.0, scale_mul=0.8, x_overlap=0.5, y_overlap=0.5):
fps = clock.fps()
lat = clock.avg()
print("**********\nPrediction:")
img.draw_rectangle(obj.rect())
# This combines the labels and confidence values into a list of tuples
predictions_list = list(zip(labels, obj.output()))
max_val = predictions_list[0][1]
max_lbl = 'background'
for i in range(len(predictions_list)):
val = predictions_list[i][1]
lbl = predictions_list[i][0]
#print("%s = %f" % (predictions_list[i][0], predictions_list[i][1]))
if val > max_val:
max_val = val
max_lbl = lbl
# Print label and turn on LED with the highest probability
if max_val < 0.5:
max_lbl = 'uncertain'
setLEDs(max_lbl)
#print("%s with a prob of %f" % (max_lbl, max_val))
print("{} with a prob of {:.2f}".format(max_lbl, max_val))
# print(clock.fps(), "fps")
print("FPS: {:.2f} fps ==> latency: {:.0f} ms".format(fps, lat))
# Draw label with highest probability to image viewer
img.draw_string(
10, 10,
max_lbl + "\n{:.2f}".format(max_val),
mono_space = False,
scale=2
)
Serial Terminal
****
Prediction:
uncertain with a prob of 0.46
FPS: 14.31 fps ==> latency: 70 ms
Frame Buffer
Record
Zoom
Disable
uncertain
0.46
Histogram
RGB Color Space
Res (w:240, h:240)
R
Mean 109 Median 82 Mode 49 StDev 72
Min 8 Max 255 LQ 49 UQ 156
G
Mean 110 Median 81 Mode 255 StDev 72
Min 4 Max 255 LQ 53 UQ 158
B
Mean 109 Median 82 Mode 49 StDev 72
Min 8 Max 255 LQ 49 UQ 156
Search Results
Serial Terminal
Board: Arduino Portenta (STM32H747)
Sensor: HM01B0
Firmware Version: 4.3.1 - [out of date - click here to updgrade]
Serial Port: cu.usbmodem315A346531301
Drive: /Volumes/NO NAME
FPS: 14,1


SAMSUNG

helloworld_LEDpy | nicla_image_classification_LED.py*

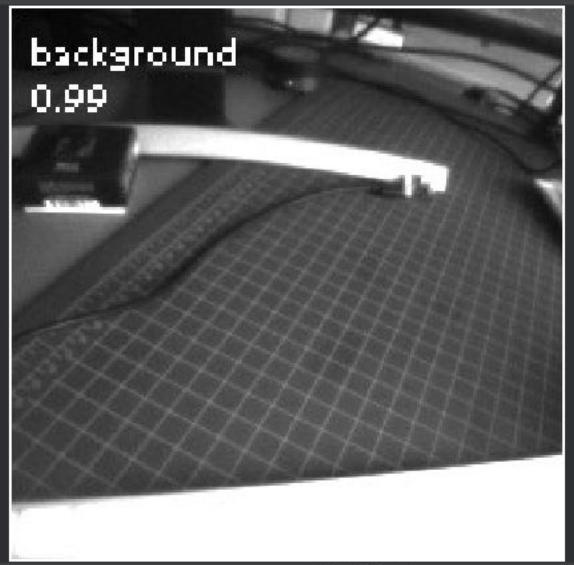
nicla_image_classification_LED.py* — Line: 86, Col: 21

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

Frame Buffer — Record | Zoom | Disable



Histogram — RGB Color Space

Res (w:240, h:240)

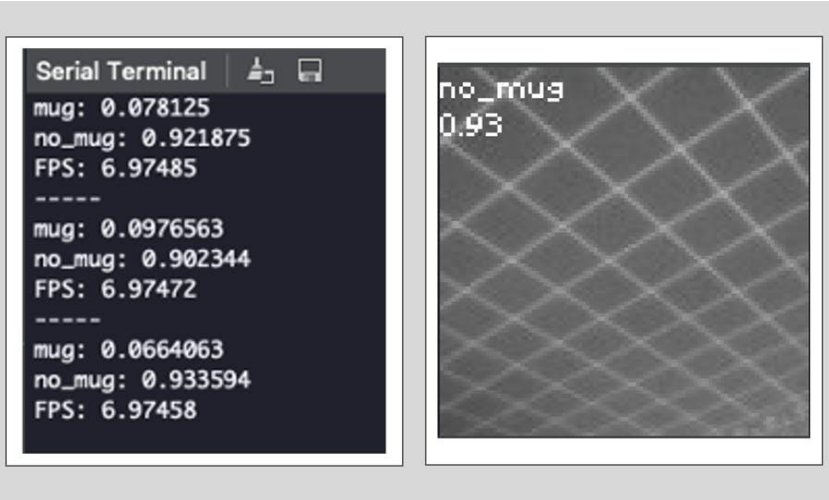
Search Results | Serial Terminal | Board: Arduino Portenta (STM32H747) | Sensor: HM01B0 | Firmware Version: 4.3.1 - [out of date - click here to updgrade] | Serial Port: cu.usbmodem315A346531301 | Drive: /Volumes/NO NAME | FPS: 14,3

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



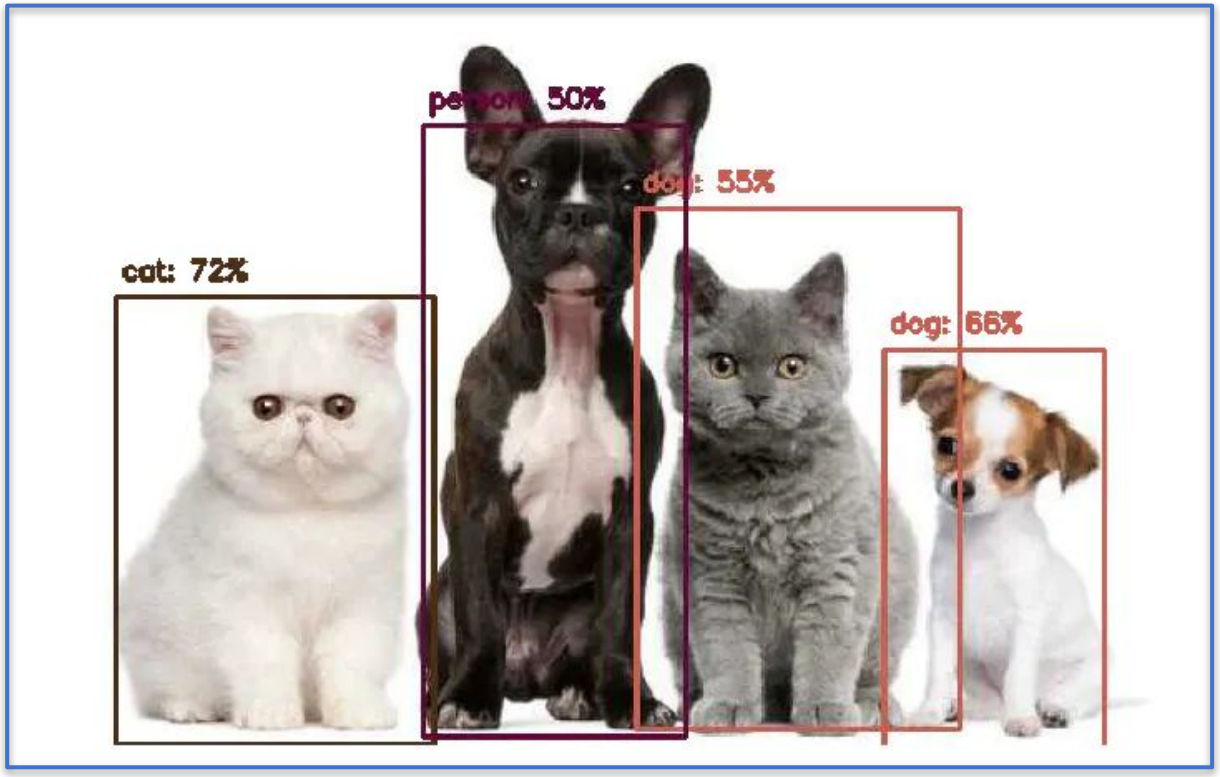
Computer Vision Main Types

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

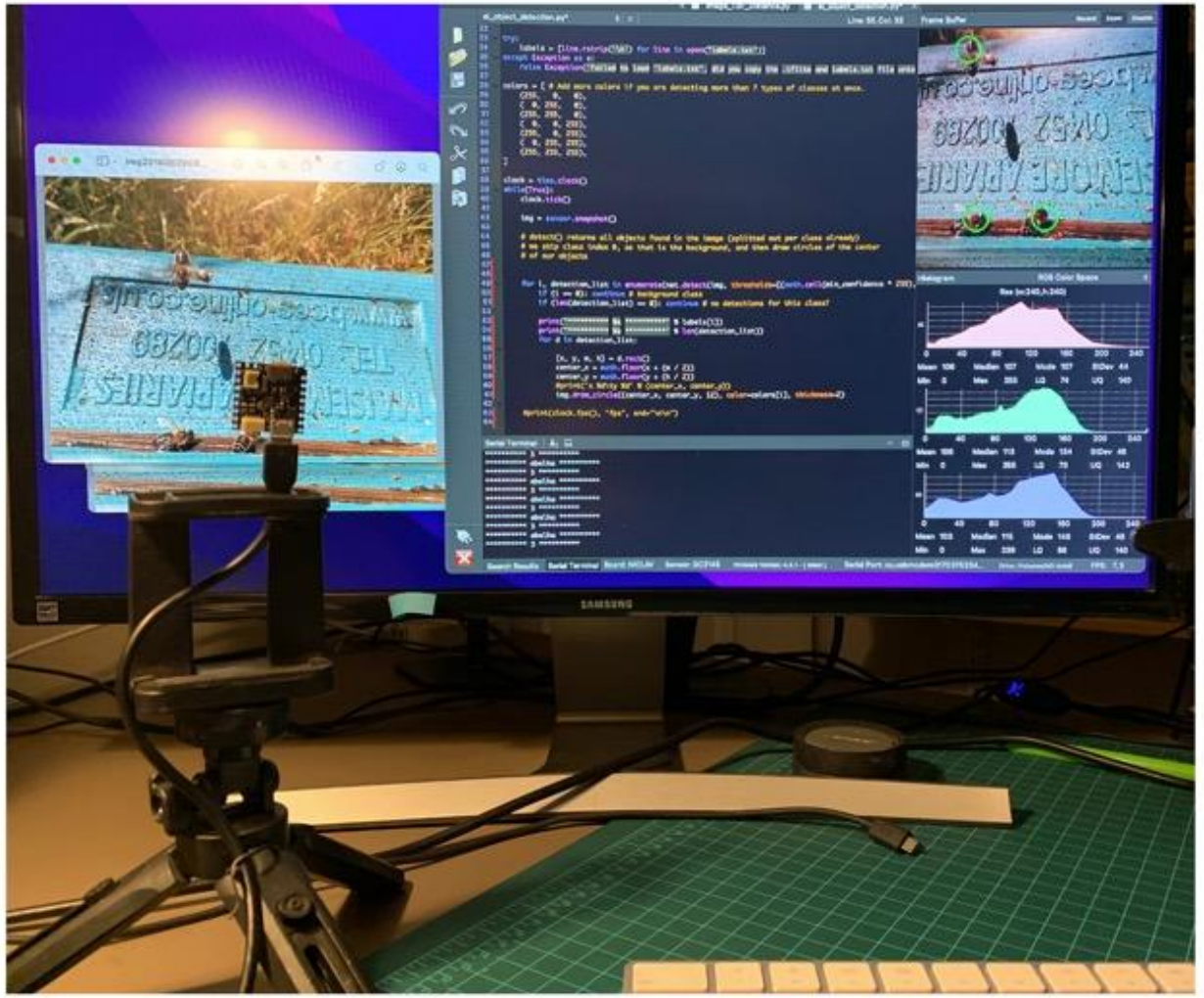
Cat: 70%

Dog: 80%

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



Detecting Objects using TinyML (FOMO)



OpenMV

MicroPython

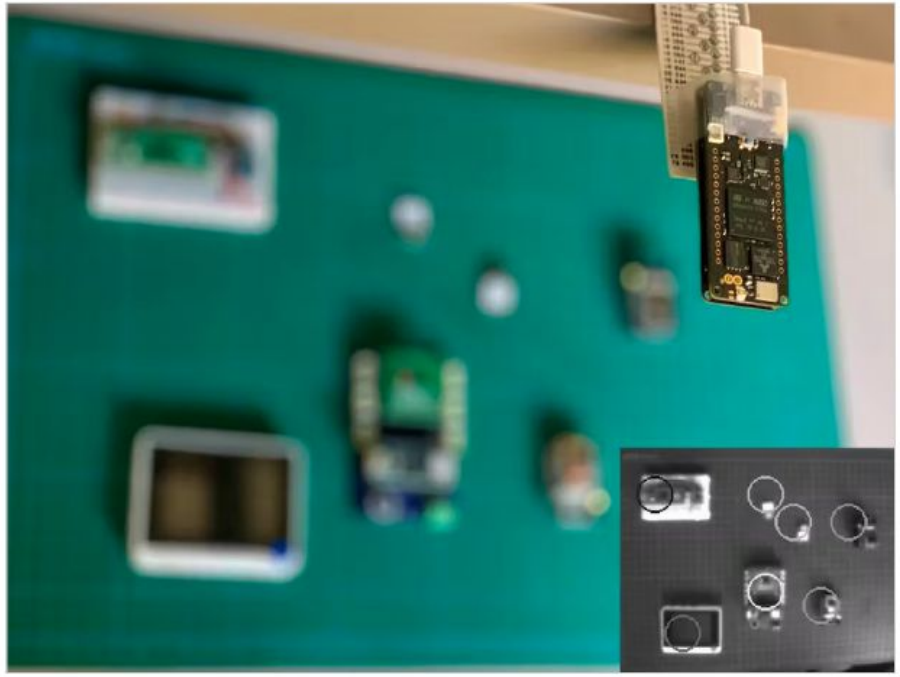
EDGE IMPULSE

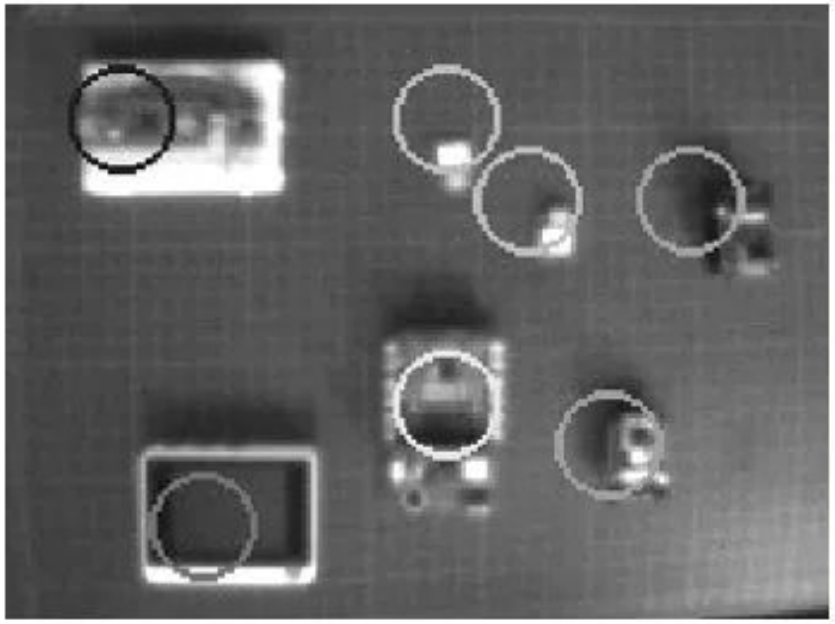
Detecting Objects using TinyML (FOMO)

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 31st, 2023	TBD	Video here when ready
May 25th, 2023	Thread here	Video here when ready
April 20th, 2023	Thread here	https://youtu.be/uoM_IjXjDFY
March 30th, 2023	thread here	https://youtu.be/UQ0l-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/e49pkjnIMlQ 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