tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

"Tiny machine learning with Arduino"

Massimo Banzi - Arduino

UK Area Group – April 20, 2021

TINY ML

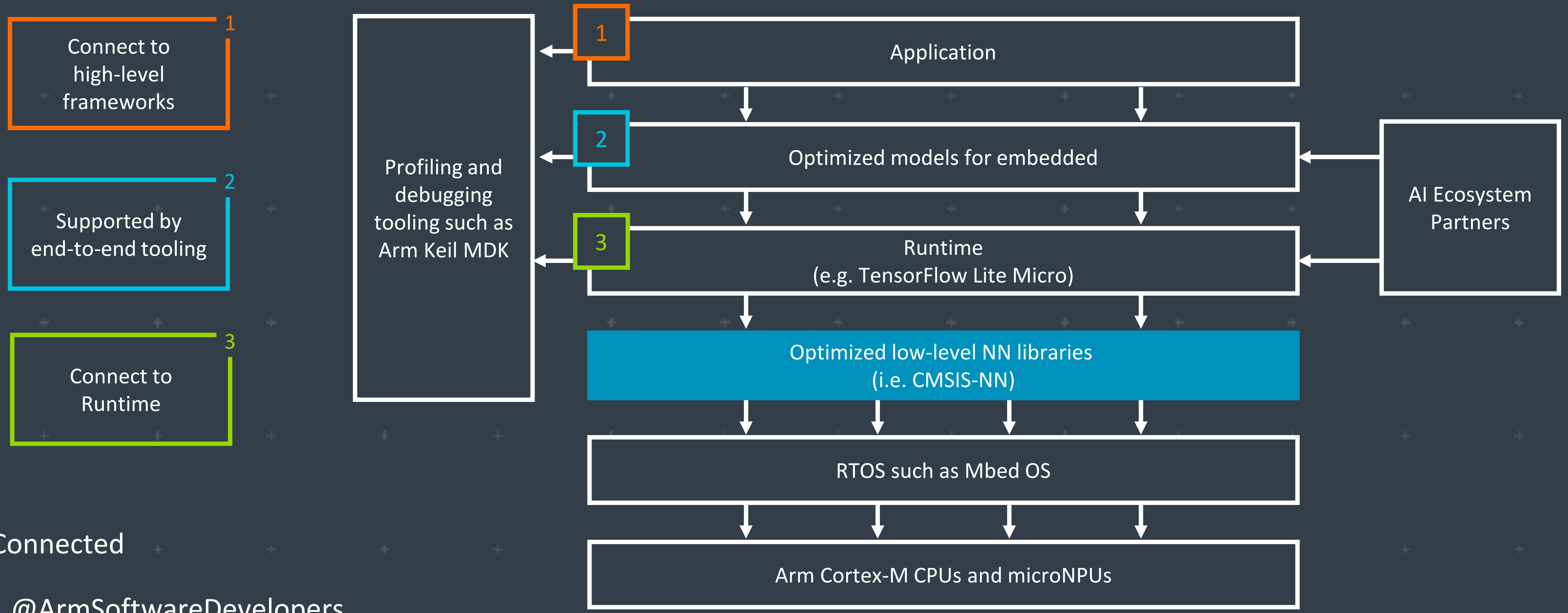
www.tinyML.org

tinyML Talks Sponsors

tinyML Strategic Partner

Additional Sponsorships available – contact Olga@tinyML.org for info

Arm: The Software and Hardware Foundation for tinyML



Stay Connected

@ArmSoftwareDevelopers

@ArmSoftwareDev

Resources: developer.arm.com/solutions/machine-learning-on-arm

Deeplite

WE USE AI TO MAKE OTHER AI FASTER, SMALLER AND MORE POWER EFFICIENT

Automatically compress SOTA models like MobileNet to <200KB with **little to no drop in accuracy** for inference on resource-limited MCUs

Reduce model optimization trial & error from weeks to days using Deeplite's **design space exploration**

APPLY NOW

Deploy more models to your device without sacrificing performance or battery life with our **easy-to-use software**

BECOME BETA USER bit.ly/testdeeplite

mobilityXlab arm AI 100 CBINSIGHTS

TinyML for all developers

C++ library

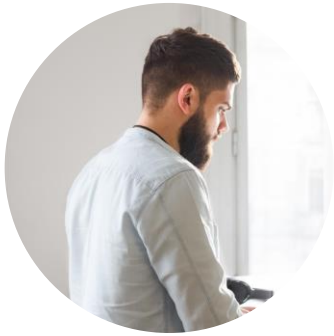
Arduino library

WebAssembly

Dataset

Acquire valuable
training data securely

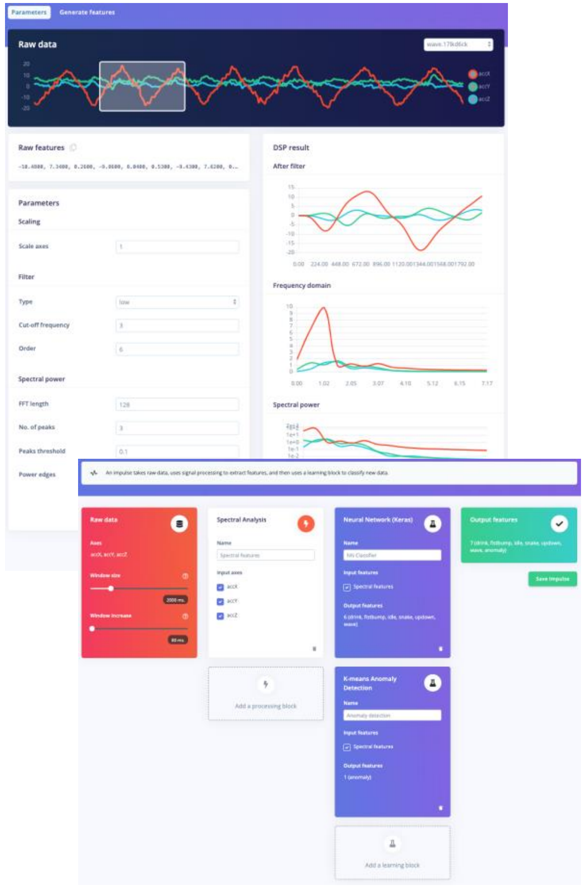
Enrich data and train
ML algorithms



Edge Device

Real sensors in real time
Open source SDK

Impulse



Embedded and edge
compute deployment
options

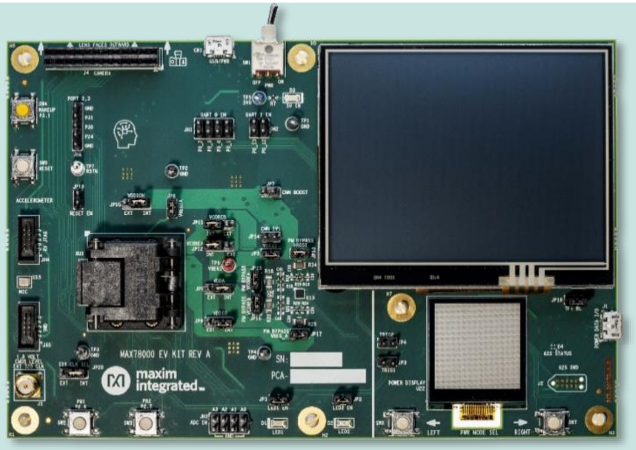
Test impulse with
real-time device
data flows

Test

www.edgeimpulse.com

Maxim Integrated: Enabling Edge Intelligence

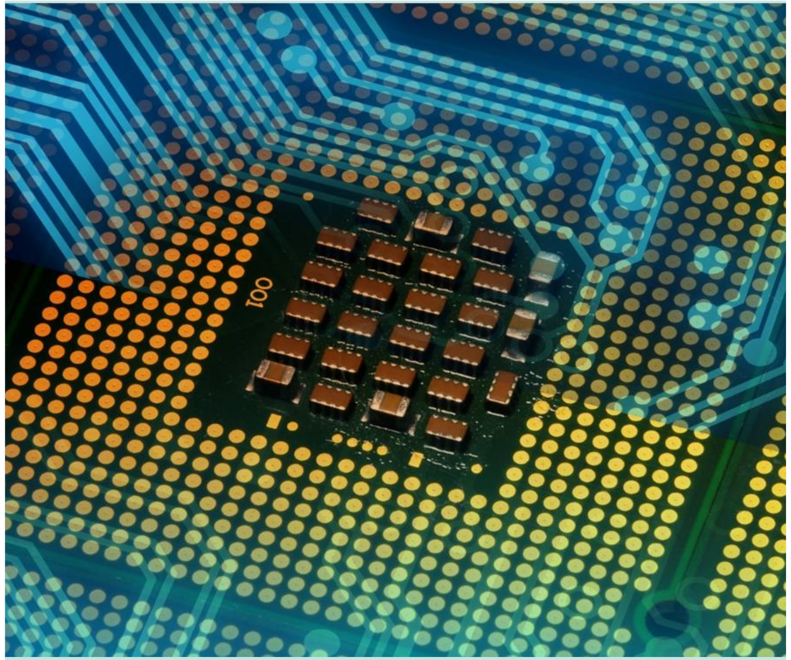
Advanced AI Acceleration IC



The new MAX78000 implements AI inferences at low energy levels, enabling complex audio and video inferencing to run on small batteries. Now the edge can see and hear like never before.

www.maximintegrated.com/MAX78000

Low Power Cortex M4 Micros



Large (3MB flash + 1MB SRAM) and small (256KB flash + 96KB SRAM, 1.6mm x 1.6mm) Cortex M4 microcontrollers enable algorithms and neural networks to run at wearable power levels.

www.maximintegrated.com/microcontrollers

Sensors and Signal Conditioning

Health sensors measure PPG and ECG signals critical to understanding vital signs. Signal chain products enable measuring even the most sensitive signals.

www.maximintegrated.com/sensors

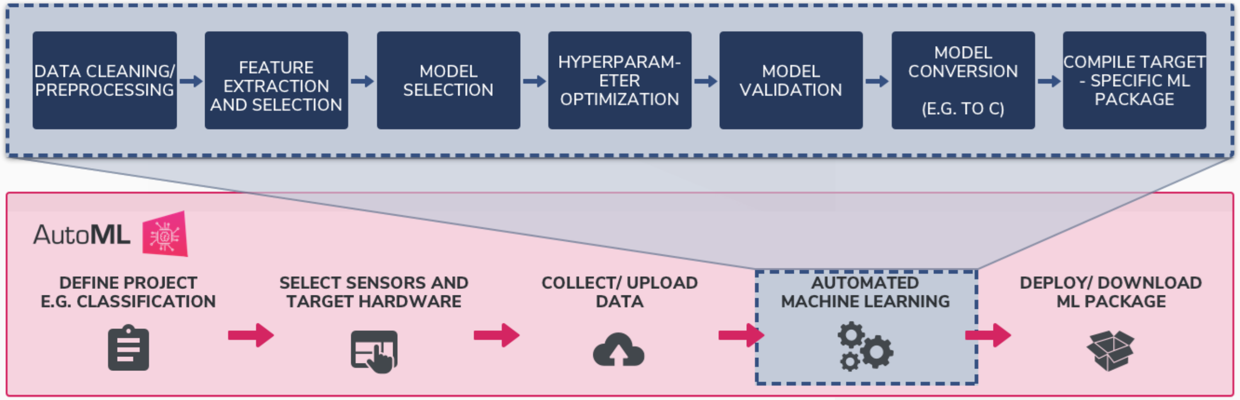
Qeexo AutoML

Automated Machine Learning Platform that builds tinyML solutions for the Edge using sensor data

Key Features

- Supports 17 ML methods:
  - Multi-class algorithms: GBM, XGBoost, Random Forest, Logistic Regression, Gaussian Naive Bayes, Decision Tree, Polynomial SVM, RBF SVM, SVM, CNN, RNN, CRNN, ANN
  - Single-class algorithms: Local Outlier Factor, One Class SVM, One Class Random Forest, Isolation Forest
- Labels, records, validates, and visualizes time-series sensor data
- On-device inference optimized for low latency, low power consumption, and small memory footprint applications
- Supports Arm® Cortex™- M0 to M4 class MCUs

End-to-End Machine Learning Platform



For more information, visit: www.qeexo.com

Target Markets/Applications

- Industrial Predictive Maintenance
- Smart Home
- Wearables
- Automotive
- Mobile
- IoT

Reality AI®

Add Advanced Sensing to your Product with Edge AI / TinyML

https://reality.ai info@reality.ai @SensorAI Reality AI

Pre-built Edge AI sensing modules, plus tools to build your own

Reality AI solutions

- Prebuilt sound recognition models for indoor and outdoor use cases
- Solution for industrial anomaly detection
- Pre-built automotive solution that lets cars "see with sound"

Reality AI Tools® software

- Build prototypes, then turn them into real products
- Explain ML models and relate the function to the physics
- Optimize the hardware, including sensor selection and placement

SynSense

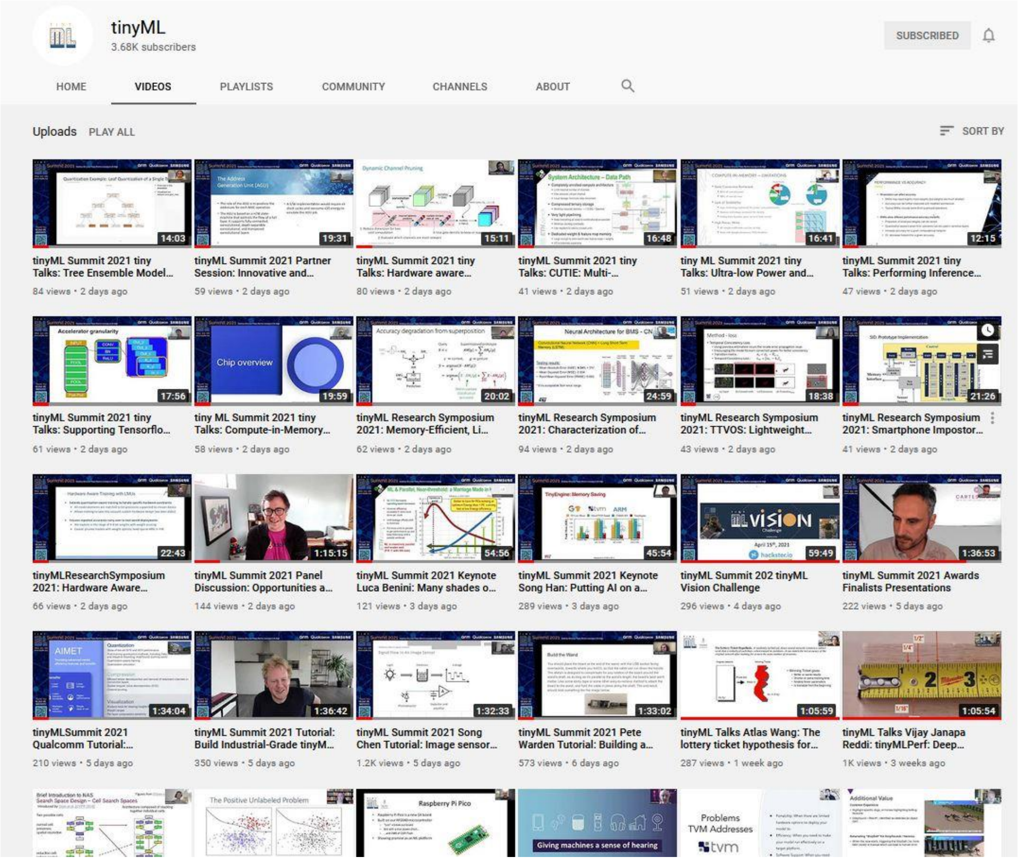
SynSense builds **sensing and inference** hardware for **ultra-low-power** (sub-mW) **embedded, mobile and edge** devices. We design systems for **real-time always-on smart sensing**, for audio, vision, IMUs, bio-signals and more.

https://SynSense.ai

Successful tinyML Summit 2021:

- **5** days of tutorials, talks, panels, breakouts, symposium
  - **4** tutorials
  - **6** keynotes & **6** plenary tinyTalks (more in breakouts)
  - **2** panel discussions
  - **5** disruptive news presentations
  - **17** breakout/partner sessions
  - **6** Best Product and Innovation Award Finalists & Presentations
  - **89** Speakers
- **5006** registered attendees representing:
  - **104** countries, **1000+** companies and **400+** academic institutions
- **26** Sponsoring companies

www.youtube.com/tinyML with 150+ videos



TINY ML EMEA

June 7-10, 2021 (virtual, but LIVE)

Deadline for abstracts: May 1

Sponsorships are being accepted: sponsorships@tinyML.org

TINY ML

Next tinyML Talks

Date	Presenter	Topic / Title
Tuesday, April 27	**Michael Jo and Xingheng Lin** Rose-Hulman Institute of Technology	Train-by-weight (TBW): Accelerated Deep Learning by Data Dimensionality Reduction

Webcast start time is 8 am Pacific time

Please contact talks@tinyml.org if you are interested in presenting

Local UK Committee

Alessandro Grande
Developer advocate & ecosystem manager, Arm

Dominic Binks
VP Technology, Audio Analytic

Gian Marco Iodice
ML Techlead, Arm

Neil Cooper
VP Marketing, Audio Analytic

Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums   youtube.com/tinyml

Please use the Q&A window for your questions

Massimo Banzi

Massimo Banzi is the co-founder of the Arduino project. He is an Interaction Designer, Educator and Open Source Hardware advocate. He has worked as a consultant for clients such as: Prada, Artemide, Persol, Whirlpool, V&A Museum and Adidas. Massimo started the first FabLab in Italy which led to the creation of Officine Arduino, a FabLab/Makerspace based in Torino. He spent 4 years at the Interaction Design Institue Ivrea as Associate Professor. Massimo has taught workshops and has been a guest speaker at institutions allover the world. Before joining IDII he was CTO for the Seat Ventures incubator. He spent many years working as a software architect, both in Milan and London, on projects for clients like Italia Online, Sapient, Labour Party, BT, MCI WorldCom, SmithKlineBeecham, Storagetek, BSkyB and boo.com. Massimo is also the author of “Getting Started with Arduino” published by O’Reilly. He is a regular contributor to the Italian edition of Wired Magazine and Che Futuro, an online magazine about innovation. He currently teaches Interaction Design at SUPSI Lugano in the south of Switzerland and is a visiting professor at CIID in Copenhagen.

TinyML with Arduino

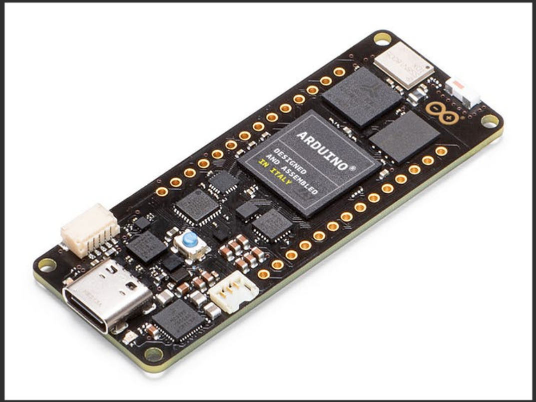
Massimo Banzi

20.04.2021

Introduction

In this talk we walk through some examples of TinyML on Arduino devices to show how simple it is to build working examples of ML with minimal coding using three different platforms compatible with Arduino.

Arduino



Enabling anyone
to innovate by
making complex
technologies
simple to use.

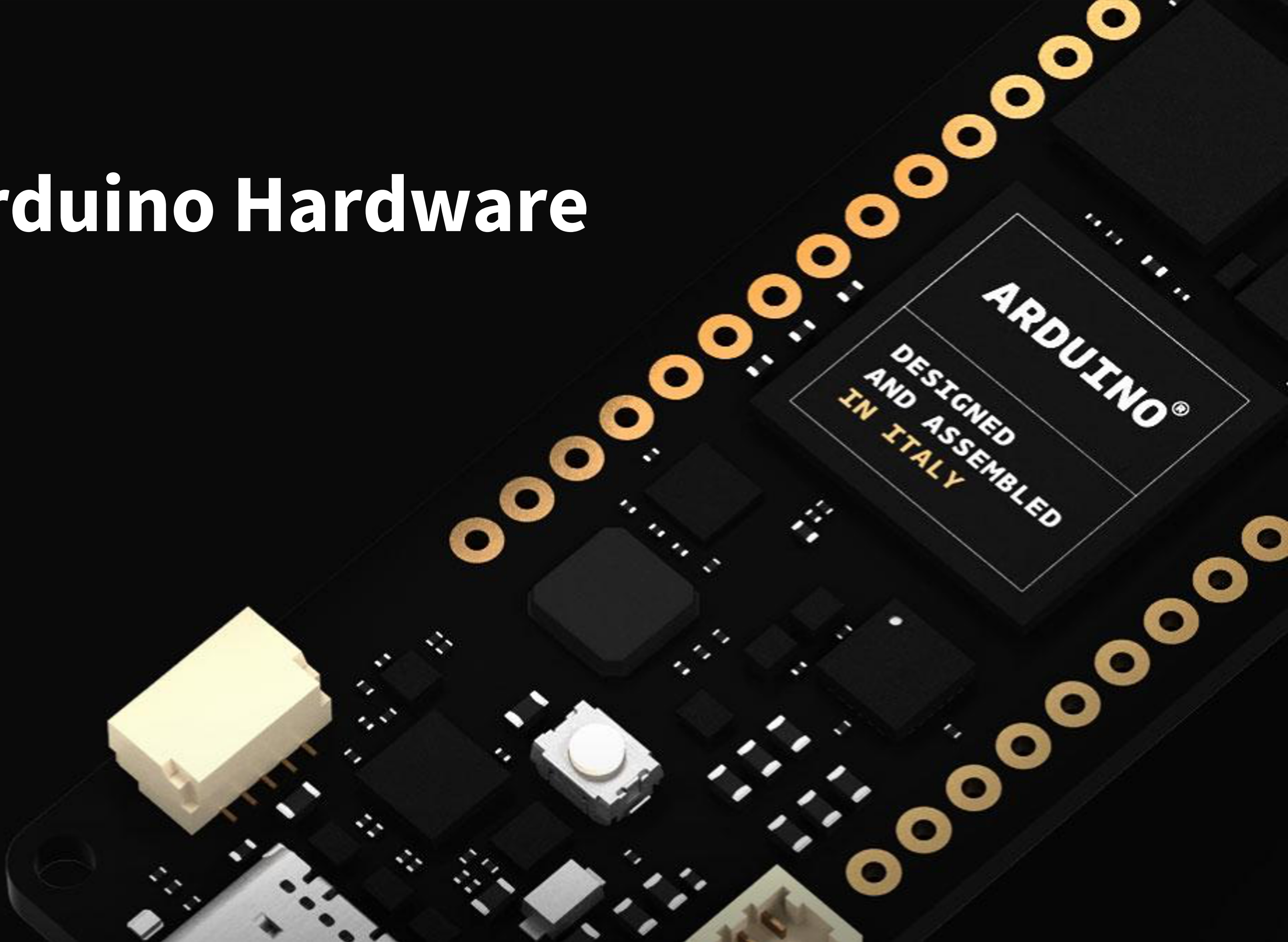
29M+	Users/Year on arduino.cc
900K+	Arduino Forum Users
17M+	IDE downloads/year
1.2M+	Users of Arduino Create Cloud
4000+	Libraries Available for Arduino

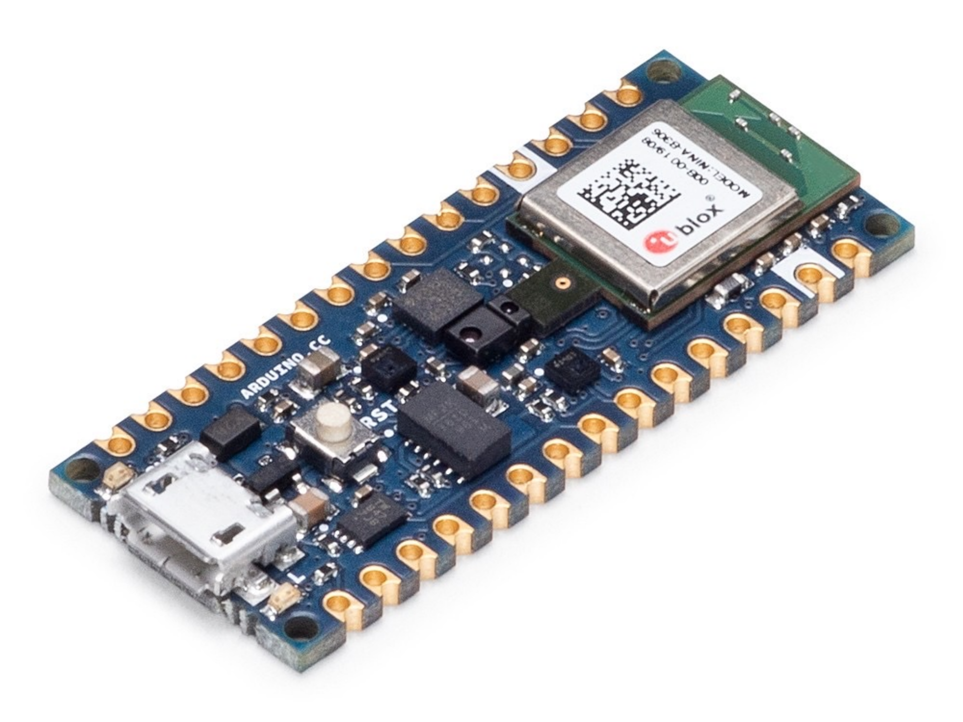
923K **302K** **408K**

Arduino Today

Easy for Beginners, Fast for professionals

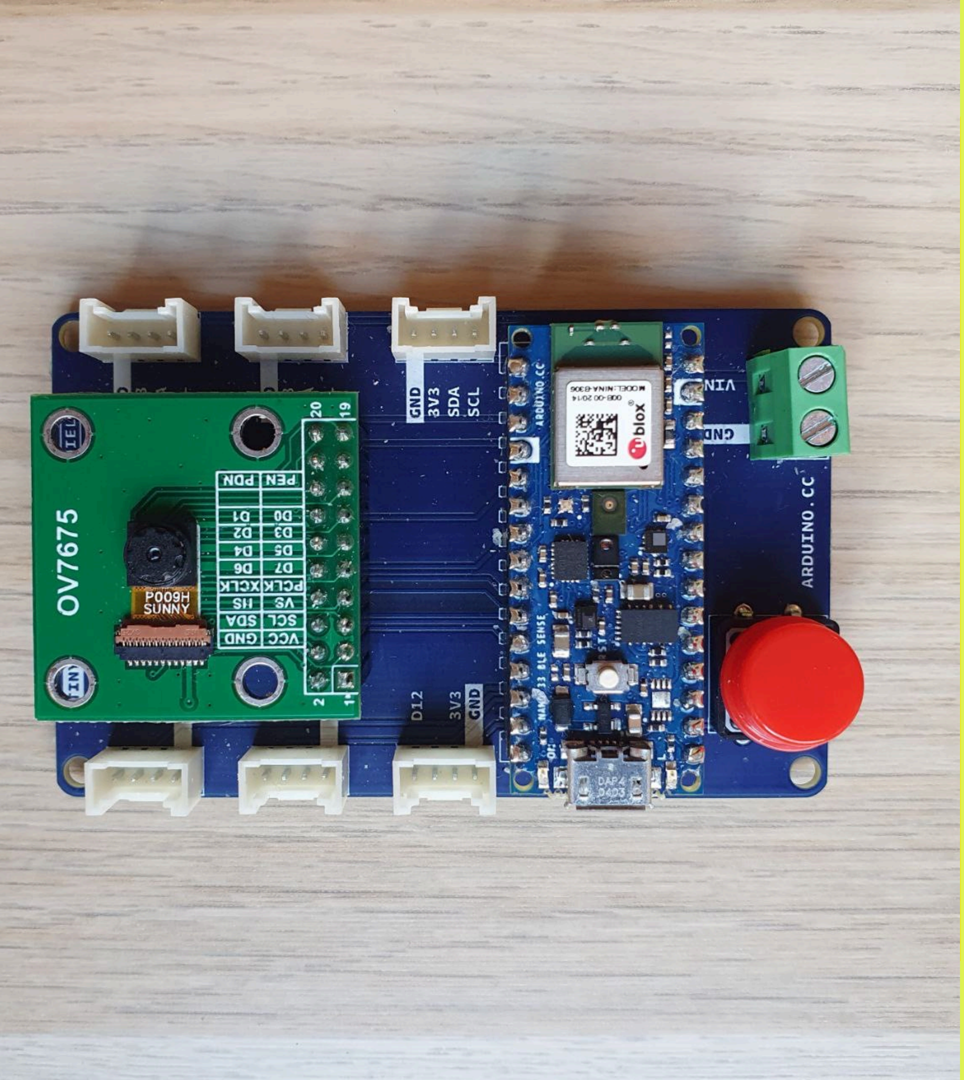
Arduino Hardware





Arduino Nano BLE Sense

- Nano form factor
- nRF52840 Arm Cortex M4f @ 64MHz
- 1MByte Flash
- 256Kb Ram
- Bluetooth 5
- Sensors: IMU, Microphone, Gesture, Light, Proximity, Pressure, Temperature , Humidity



Tiny Machine Learning Kit

- VGA Camera and Nano BLE Sense
- Used for the EdX course on TinyML


ARDUINO
TINY MACHINE
LEARNING KIT
In partnership with:
edX
TensorFlow

edX Courses Programs & Degrees Schools & Partners Search online courses edX for Business

Catalog › Data Science Courses › HarvardX's Tiny Machine Learning (TinyML)

The Future of ML is Tiny and Bright

HARVARD UNIVERSITY

Professional Certificate in

Tiny Machine Learning (TinyML)

I'm interested

https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning

- Fundamentals of machine learning and embedded devices.

TinyML

Home Lectures Final Projects Assignments HarvardX FAQ

CS249r: Tiny Machine Learning

Applied Machine Learning for Embedded IoT Devices

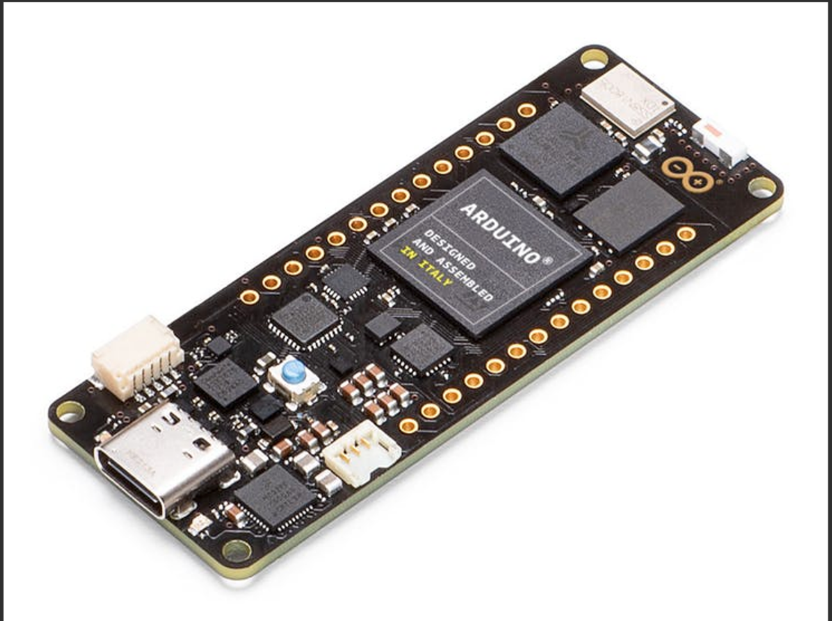
Welcome

Course Overview

Tiny Machine Learning (TinyML) is an introductory course at the intersection of Machine Learning and Embedded IoT Devices. The pervasiveness of ultra-low-power embedded devices, coupled with the introduction of embedded machine learning frameworks like TensorFlow Lite for Microcontrollers, will enable the mass proliferation of AI-powered IoT devices. The explosive growth in machine learning and the ease of use of platforms like TensorFlow (TF) make it an indispensable topic of study for modern computer science *and* electrical engineering students.

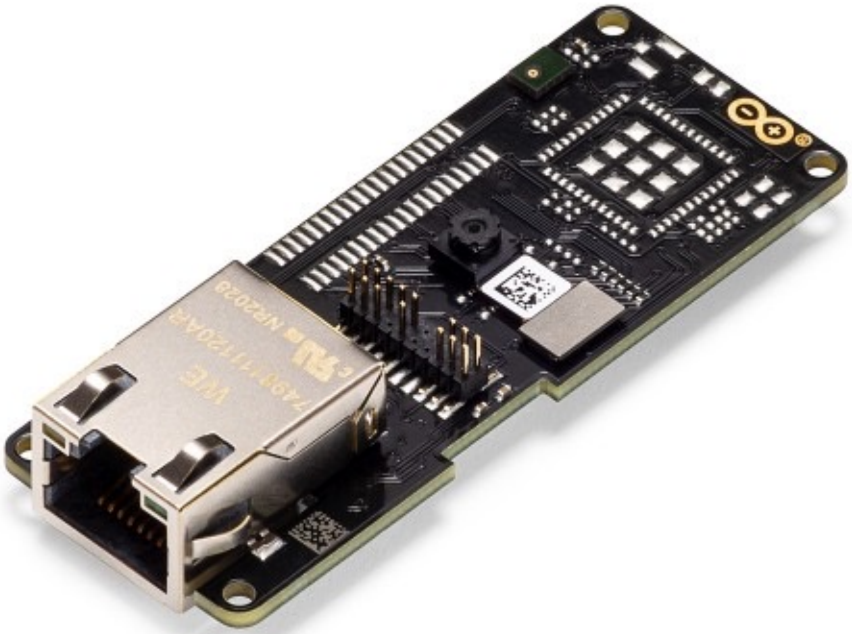
https://sites.google.com/g.harvard.edu/tinyml/home

The course will cover all things related to machine learning on embedded devices. The topics range from tinyML applications and algorithms to the design of the frameworks built



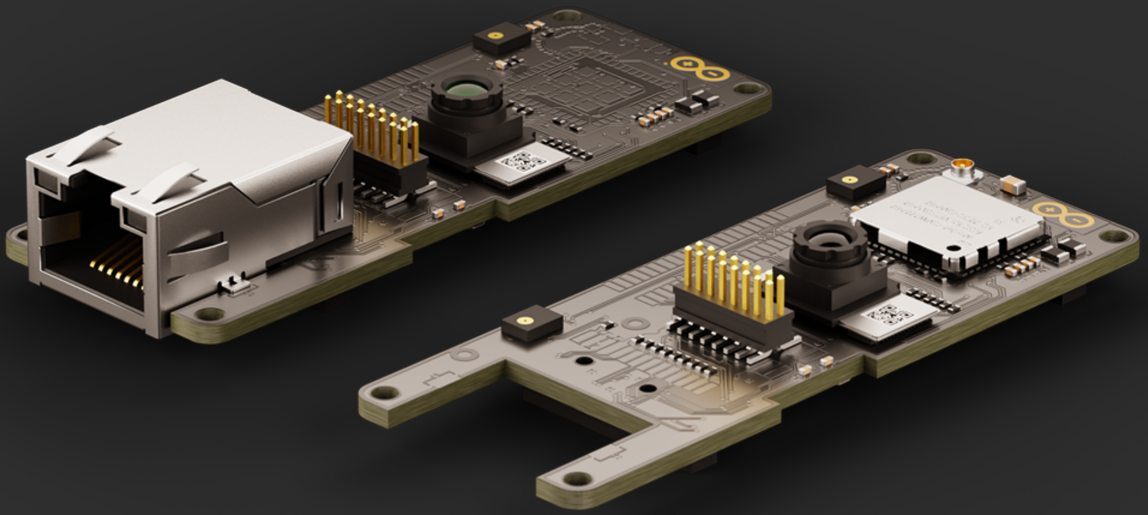
Portenta H7

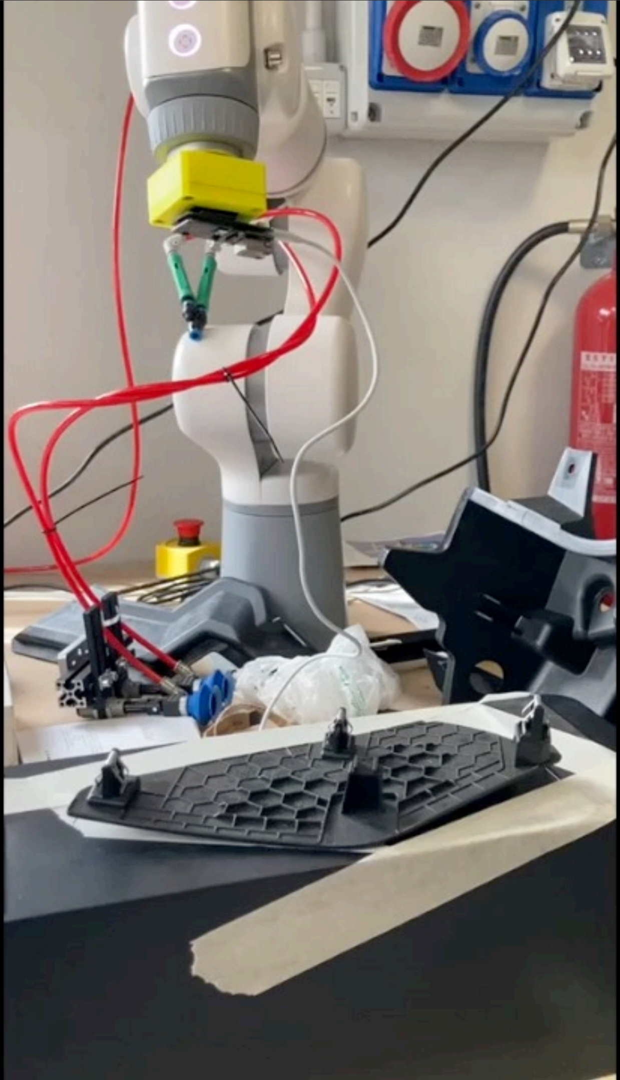
- Portenta form factor
- Dual core
  - Arm Cortex M7@480MHz
  - Arm Cortex M4@240MHz
- up to 64 MByte SDRAM
- up to 128 MByte QSPI Flash
- Common Criteria Crypto
- WiFi b/g/n 65Mbps + BT 5.1
- 100Mbit Ethernet
- CAN
- USB-C with DisplayPort output



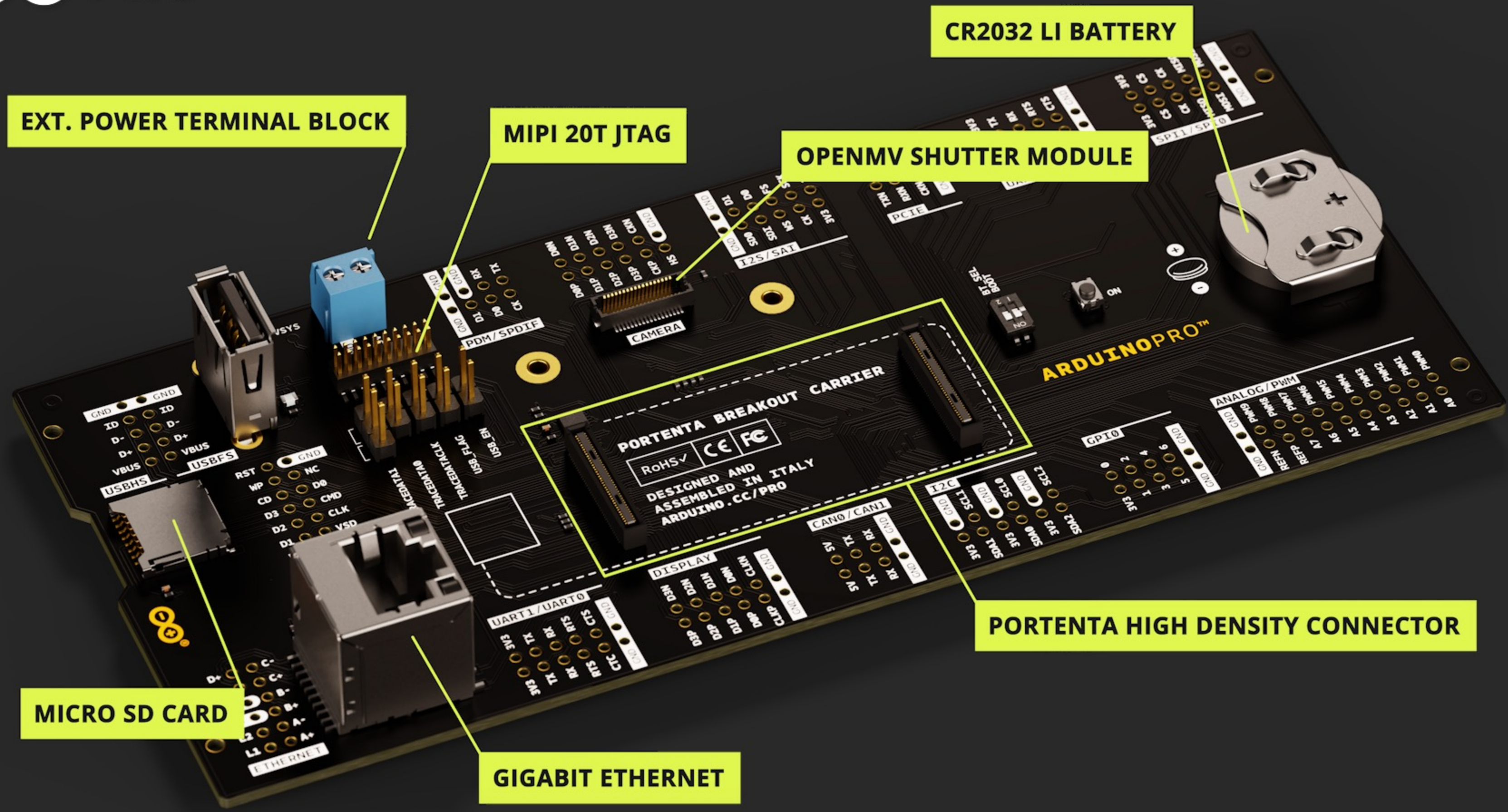
Portenta Vision Shield

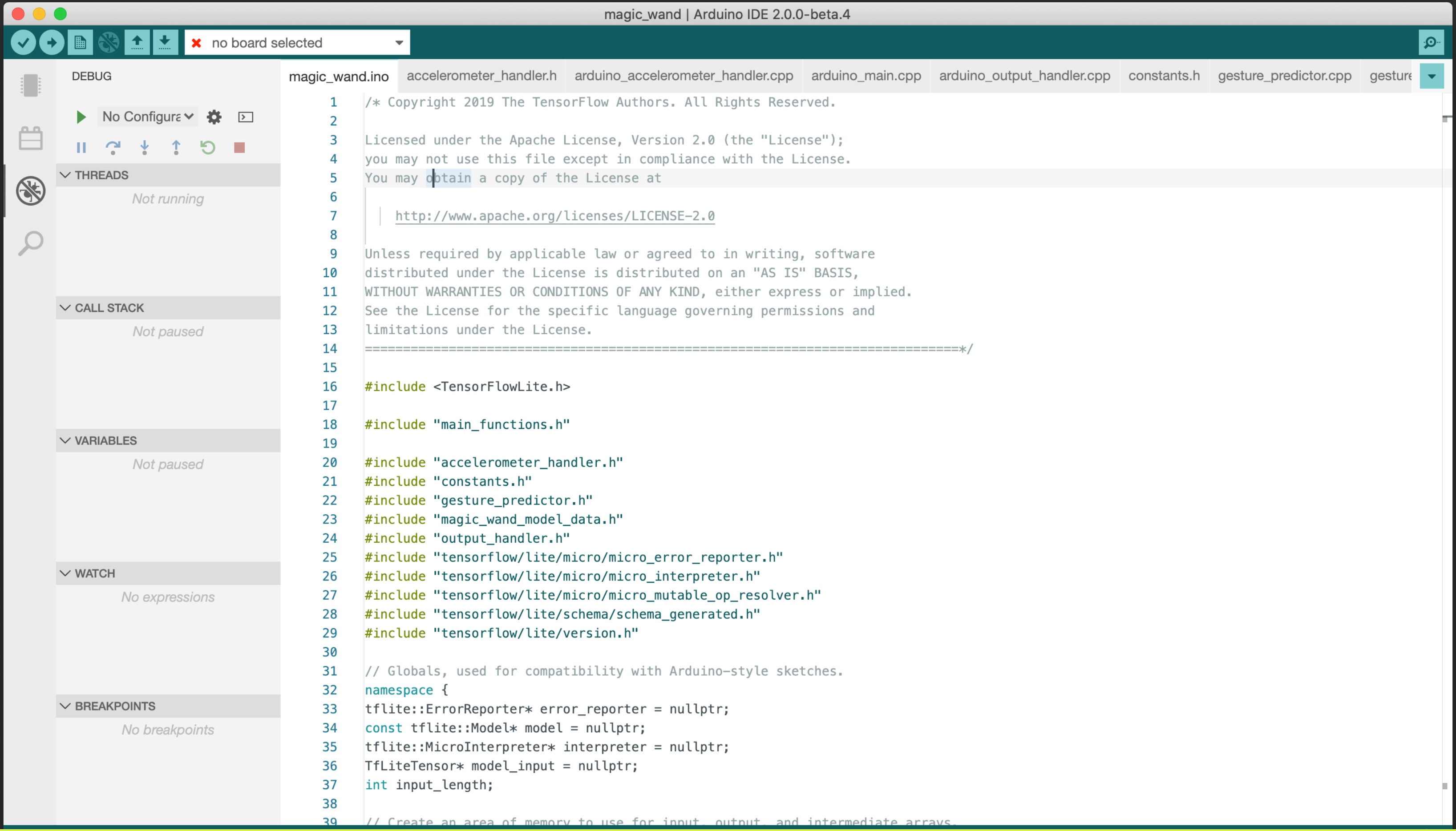
- Portenta form factor
- Low power cameras
  - Himax HM-01B0
  - Himax HM-0360
- Dual microphone (beamforming)
- Ethernet
- Lora
- SD Card



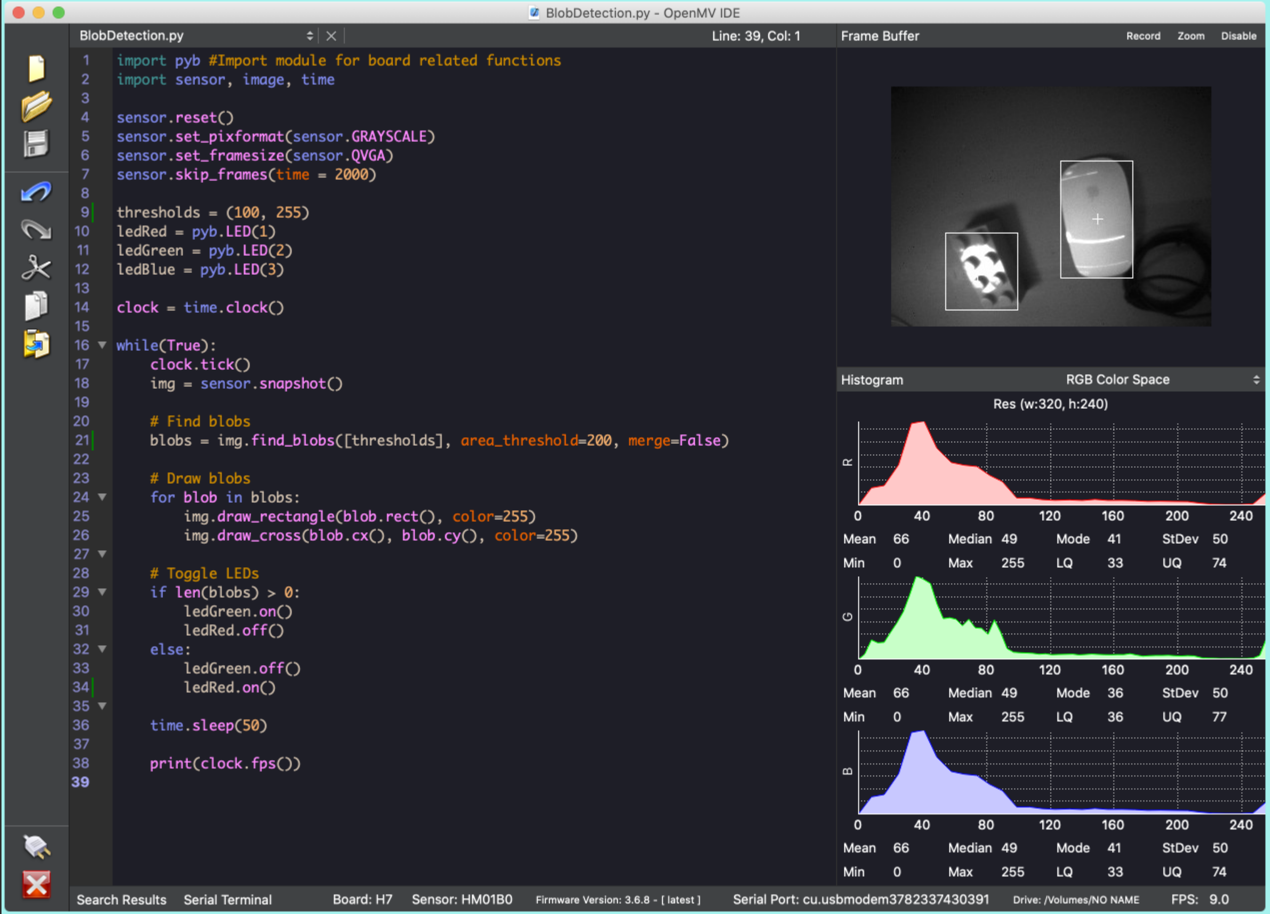


PRO


CR2032 LI BATTERY
EXT. POWER TERMINAL BLOCK
MIPI 20T JTAG
OPENMV SHUTTER MODULE
PORTENTA BREAKOUT CARRIER
DESIGNED AND ASSEMBLED IN ITALY
ARDUINO.CC/PRO
ARDUINO PRO
PORTENTA HIGH DENSITY CONNECTOR
MICRO SD CARD
GIGABIT ETHERNET

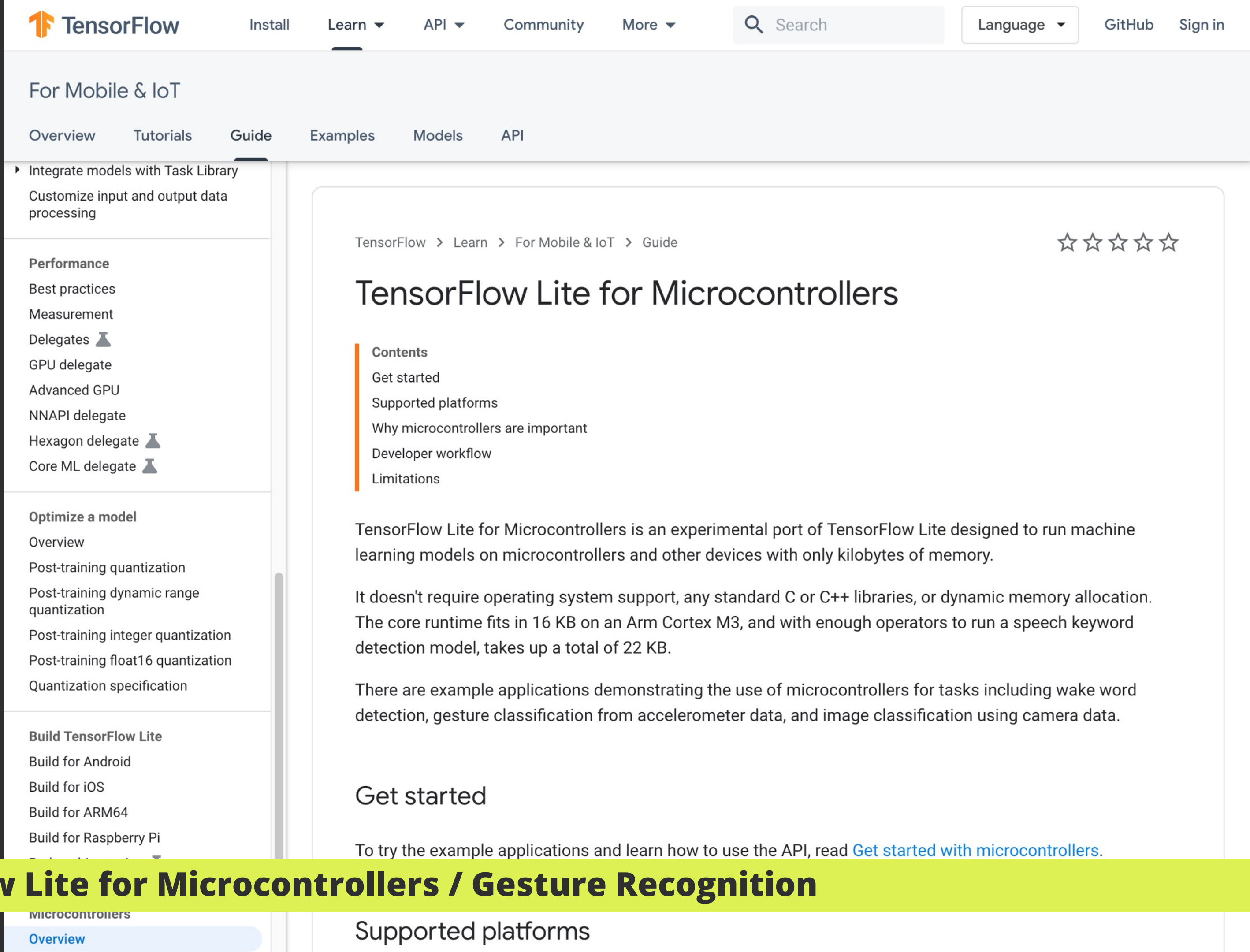


Arduino IDE 2.0 Beta

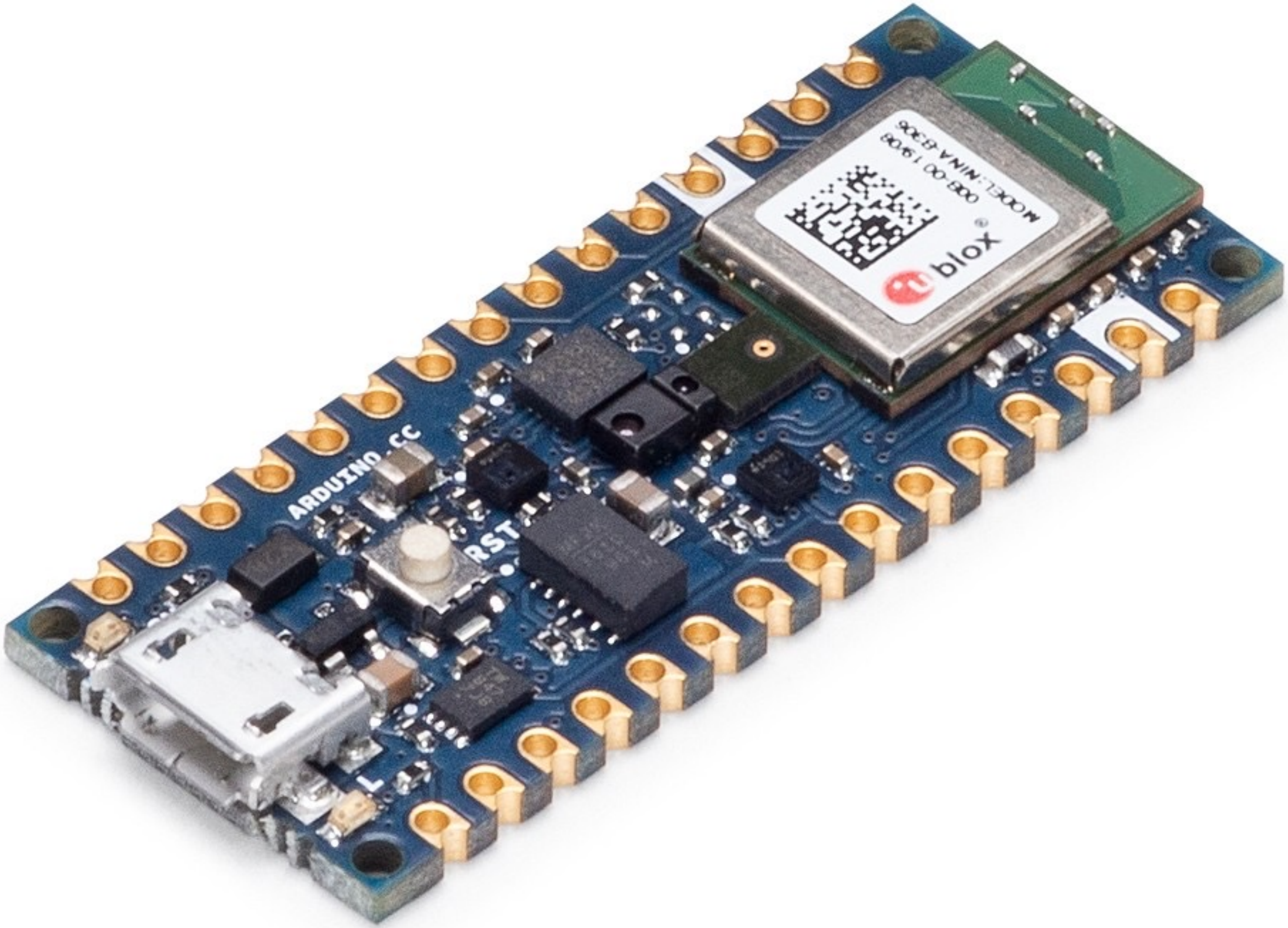

BlobDetection.py - OpenMV IDE
BlobDetection.py
Line: 39, Col: 1
Frame Buffer
Record
Zoom
Disable
import pyb #Import module for board related functions
import sensor, image, time
sensor.reset()
sensor.set_pixformat(sensor.GRAYSCALE)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 2000)
thresholds = (100, 255)
ledRed = pyb.LED(1)
ledGreen = pyb.LED(2)
ledBlue = pyb.LED(3)
clock = time.clock()
while(True):
clock.tick()
img = sensor.snapshot()
# Find blobs
blobs = img.find_blobs([thresholds], area_threshold=200, merge=False)
# Draw blobs
for blob in blobs:
img.draw_rectangle(blob.rect(), color=255)
img.draw_cross(blob.cx(), blob.cy(), color=255)
# Toggle LEDs
if len(blobs) > 0:
ledGreen.on()
ledRed.off()
else:
ledGreen.off()
ledRed.on()
time.sleep(50)
print(clock.fps())
Histogram
RGB Color Space
Res (w:320, h:240)
R
0 40 80 120 160 200 240
Mean 66 Median 49 Mode 41 StDev 50
Min 0 Max 255 LQ 33 UQ 74
G
0 40 80 120 160 200 240
Mean 66 Median 49 Mode 36 StDev 50
Min 0 Max 255 LQ 36 UQ 77
B
0 40 80 120 160 200 240
Mean 66 Median 49 Mode 41 StDev 50
Min 0 Max 255 LQ 33 UQ 74
Search Results
Serial Terminal
Board: H7
Sensor: HM01B0
Firmware Version: 3.6.8 - [latest]
Serial Port: cu.usbmodem3782337430391
Drive: /Volumes/NO NAME
FPS: 9.0

OpenMV IDE

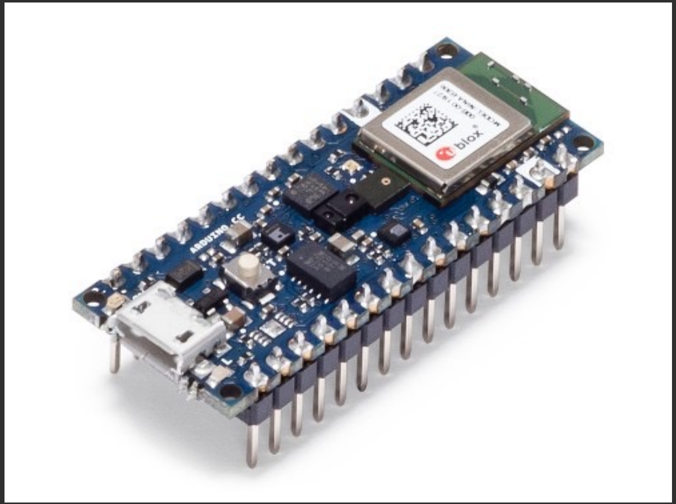
Examples
Arduino Tiny ML



TensorFlow Lite for Microcontrollers / Gesture Recognition


ARDUINO.CC
RST
L
blox

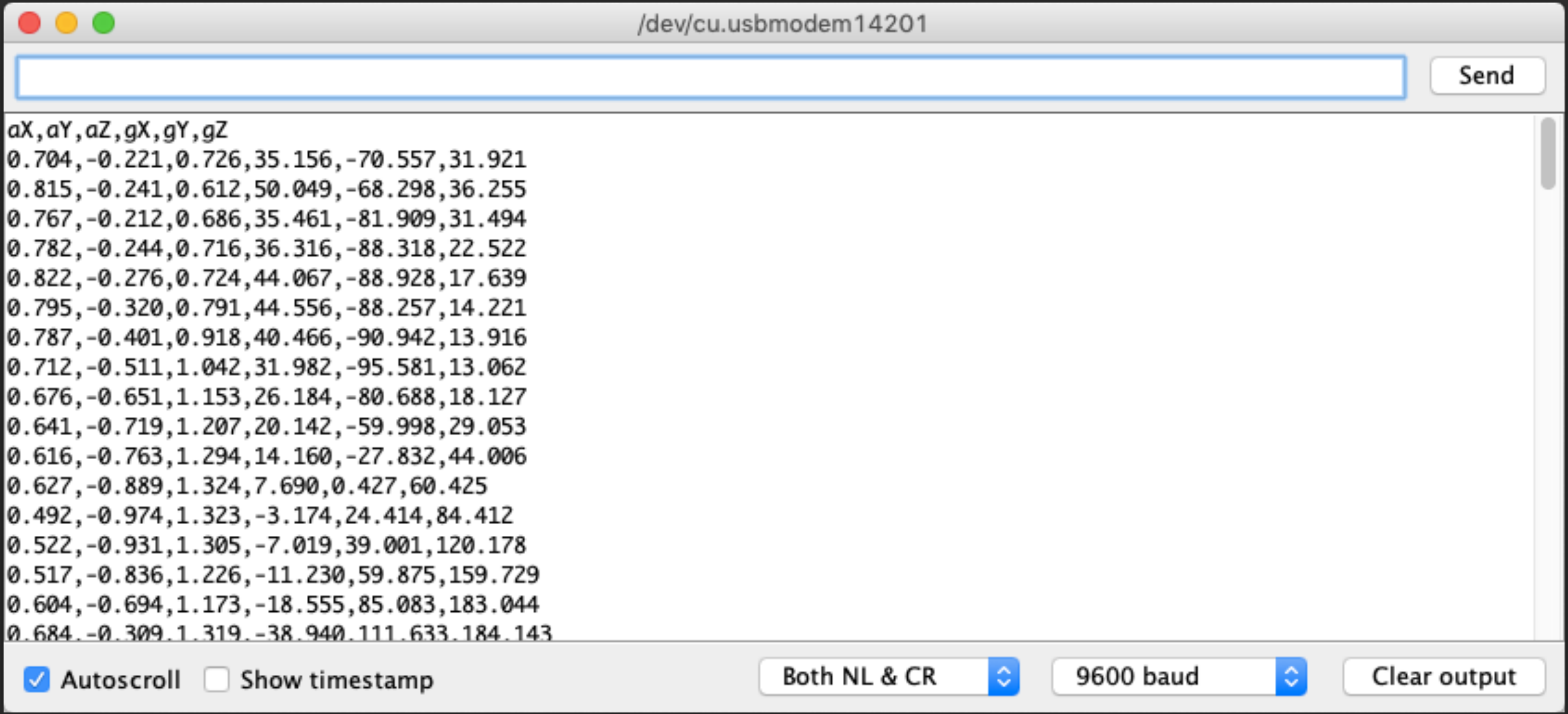
Gesture Recognition

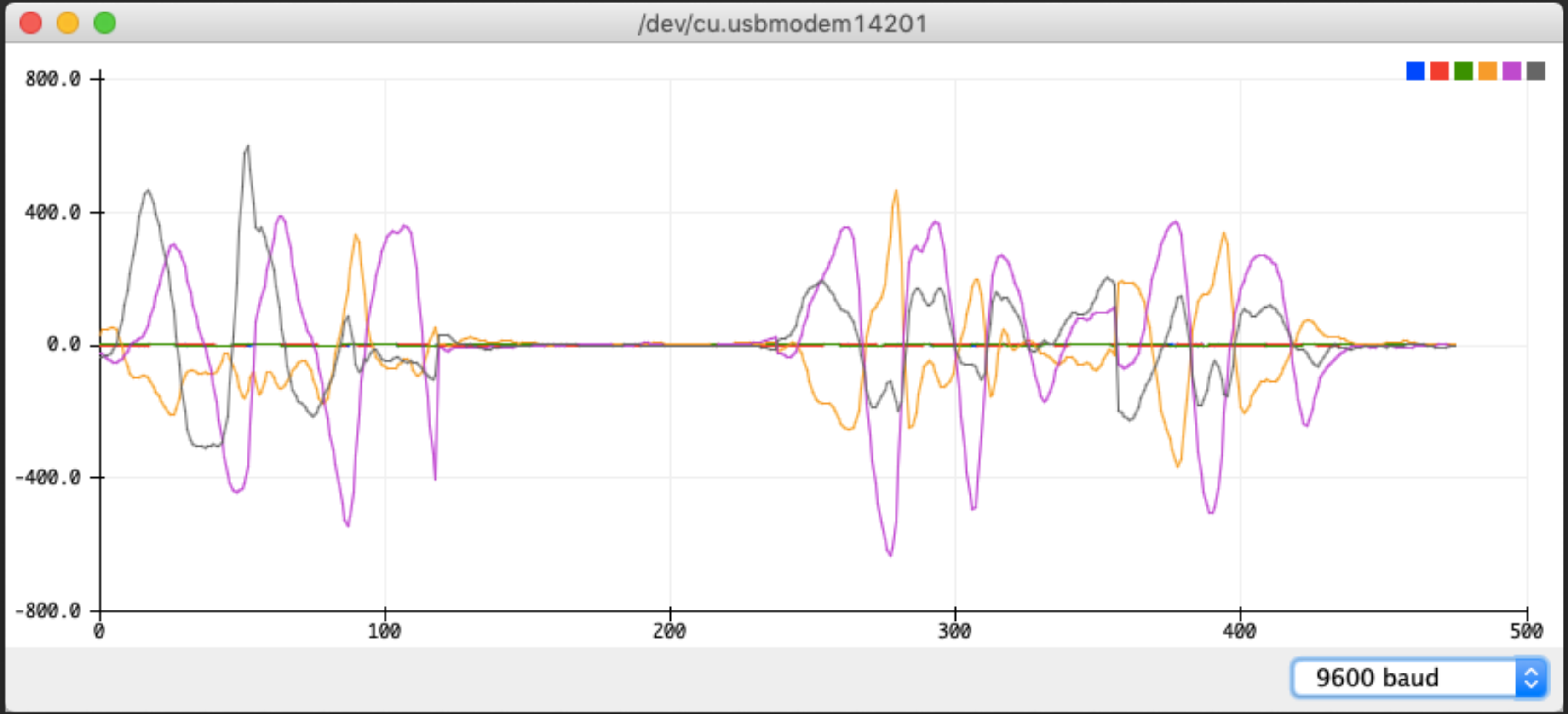


We’re going to use the code provided here:
https://github.com/arduino/ArduinoTensorFlowLiteTutorials/tree/master/GestureToEmoji

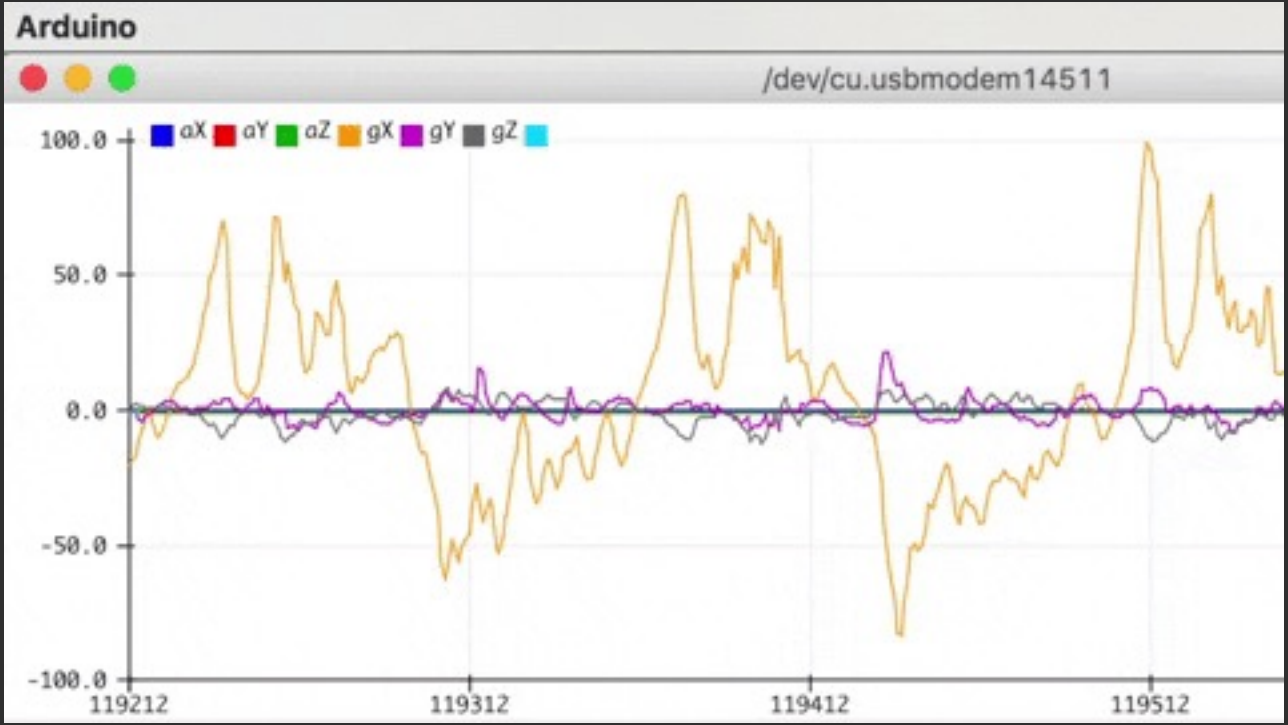
Visualize the IMU Data

- Open **ArduinoSketches/IMU_Capture/IMU_Capture.ino** in the Arduino IDE.
- Compile the sketch and upload it to the board: Sketch -> Upload
- Open the Serial Monitor: Tools -> Serial Monitor
- Press the button, IMU data will be captured and outputted for 1 second
- Close the Serial Monitor window
- Open the Serial Plotter: Tools -> Serial Plotter
- Press the button, and perform a gesture
- You'll see a graph of the data capture
- Repeat capturing various gestures to get a sense of what the training data will look like
- Close the Serial Plotter


/dev/cu.usbmodem14201
Send
aX,aY,aZ,gX,gY,gZ
0.704,-0.221,0.726,35.156,-70.557,31.921
0.815,-0.241,0.612,50.049,-68.298,36.255
0.767,-0.212,0.686,35.461,-81.909,31.494
0.782,-0.244,0.716,36.316,-88.318,22.522
0.822,-0.276,0.724,44.067,-88.928,17.639
0.795,-0.320,0.791,44.556,-88.257,14.221
0.787,-0.401,0.918,40.466,-90.942,13.916
0.712,-0.511,1.042,31.982,-95.581,13.062
0.676,-0.651,1.153,26.184,-80.688,18.127
0.641,-0.719,1.207,20.142,-59.998,29.053
0.616,-0.763,1.294,14.160,-27.832,44.006
0.627,-0.889,1.324,7.690,0.427,60.425
0.492,-0.974,1.323,-3.174,24.414,84.412
0.522,-0.931,1.305,-7.019,39.001,120.178
0.517,-0.836,1.226,-11.230,59.875,159.729
0.604,-0.694,1.173,-18.555,85.083,183.044
0.684,-0.309,1.319,-38.940,111.633,184.143
Autoscroll
Show timestamp
Both NL & CR
9600 baud
Clear output



TensorFlow Lite for Microcontrollers / Gesture Recognition



TensorFlow Lite for Microcontrollers / Gesture Recognition

Gather Training Data

- Press the reset button on the board
- Open the Serial Monitor: Tools -> Serial Monitor
- Make a punch gesture with the board in your hand - you should see the sensor data log in the Serial Monitor
- Repeat 10 times to gather more data
- Copy and paste the data from the serial output to new text file called punch.csv using your favorite text editor
- Close the Serial Monitor
- Press the reset button on the board
- Open the Serial Monitor: Tools -> Serial Monitor
- Make a flex gesture with the board in your hand
- Repeat 10 times
- Copy and paste the serial output to new text file flex.csv using your favorite text editor

arduino tinyml workshop.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

Table of contents Code snippets Files

Tiny ML on Arduino

Gesture recognition tutorial

- Sandeep Mistry - Arduino
- Don Coleman - Chariot Solutions

https://github.com/arduino/ArduinoTensorFlowLiteTutorials/

Setup Python Environment

The next cell sets up the dependencies in required for the notebook, run it.

```
[ ] # Setup environment
    !apt-get -qq install xxd
    !pip install pandas numpy matplotlib
    !pip install tensorflow==2.0.0-rc1
```

Upload Data

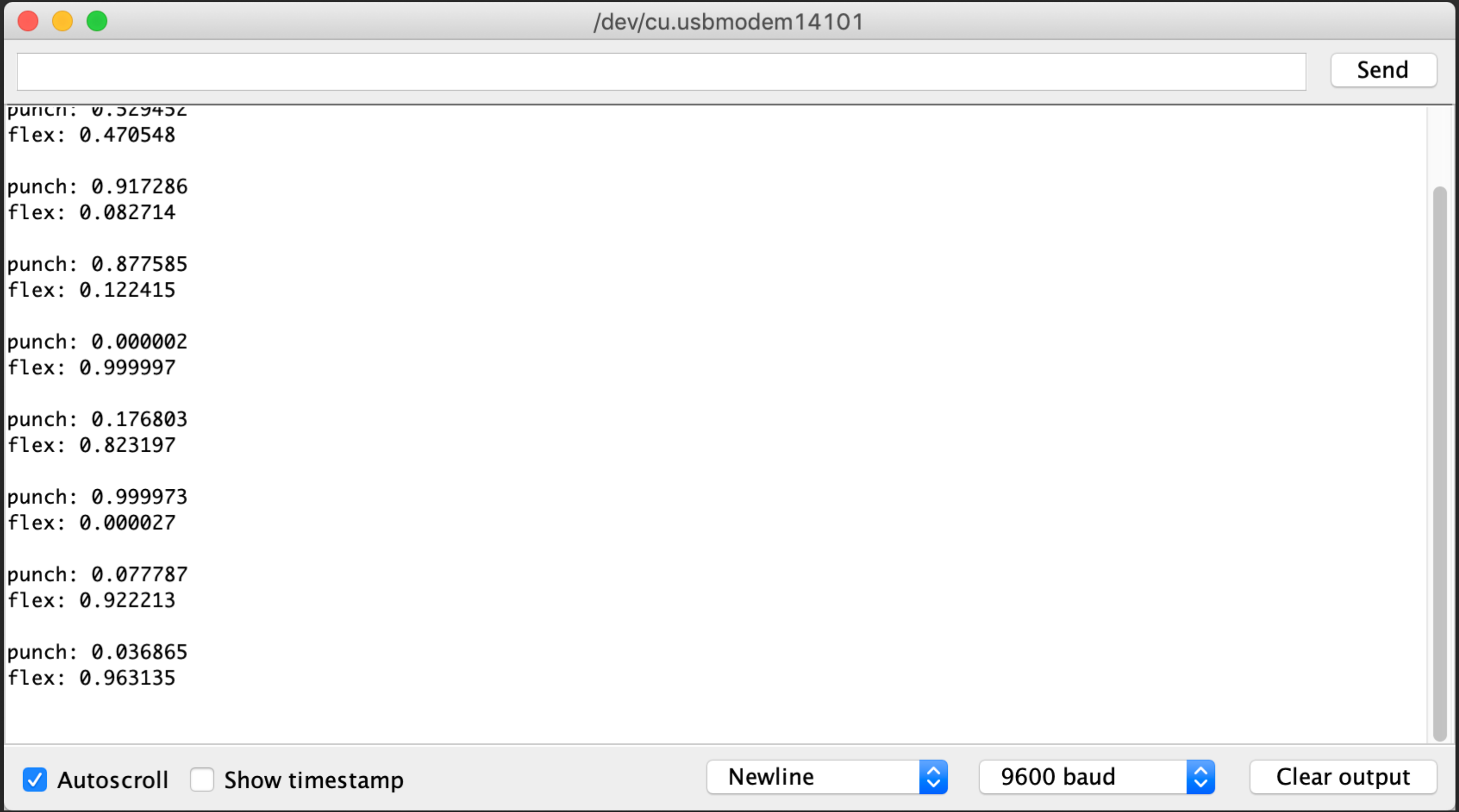
1. Open the panel on the left side of Colab by clicking on the >
2. Select the files tab

Machine Learning

- Go to this colab:
- https://colab.research.google.com/github/arduino/ArduinoTensorFlowLiteTutorials/blob/master/GestureToEmoji/arduino_tinyml_workshop.ipynb
- Drag punch.csv and flex.csv files from your computer to the tab to upload them into colab.
- Follow the instructions on the Notebook and you'll end up with "model.h" file you can import into arduino

Classifying IMU Data

- Open ArduinoSketches/IMU_Classifier/IMU_Classifier.ino in the Arduino IDE.
- Switch to the model.h tab
- Replace the contents of model.h with the version you downloaded from Colab
- Upload the sketch: Sketch -> Upload
- Open the Serial Monitor: Tools -> Serial Monitor
- Press the button, and perform a gesture
- The confidence of each gesture will be printed to the Serial Monitor (0 -> low confidence, 1 -> high confidence)

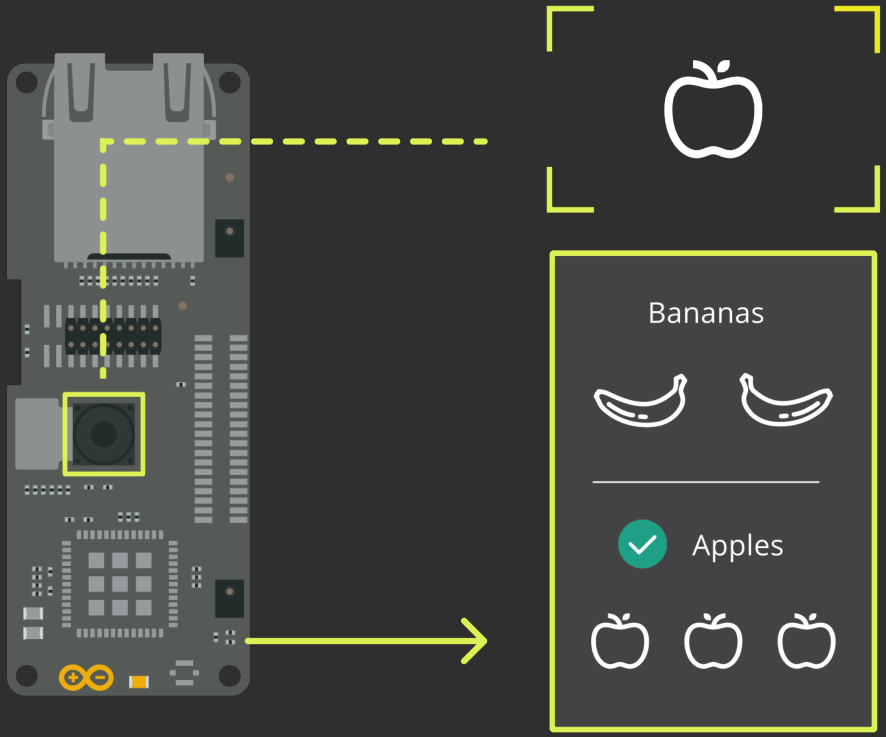

/dev/cu.usbmodem14101
Send
flex: 0.470548
punch: 0.917286
flex: 0.082714
punch: 0.877585
flex: 0.122415
punch: 0.000002
flex: 0.999997
punch: 0.176803
flex: 0.823197
punch: 0.999973
flex: 0.000027
punch: 0.077787
flex: 0.922213
punch: 0.036865
flex: 0.963135
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

Gesture Controlled USB Emoji Keyboard

- The Emoji_Button example, ArduinoSketchs/Emoji_Button/Emoji_Button.ino, shows how to create a USB keyboard that prints and emoji character. (This only works on Linux and macos, so if you're running Windows, find a friend to work on this exercise.)

- Try combining the Emoji_Button example with the IMU_Classifier sketch to create an gesture controlled emoji keyboard. 👊

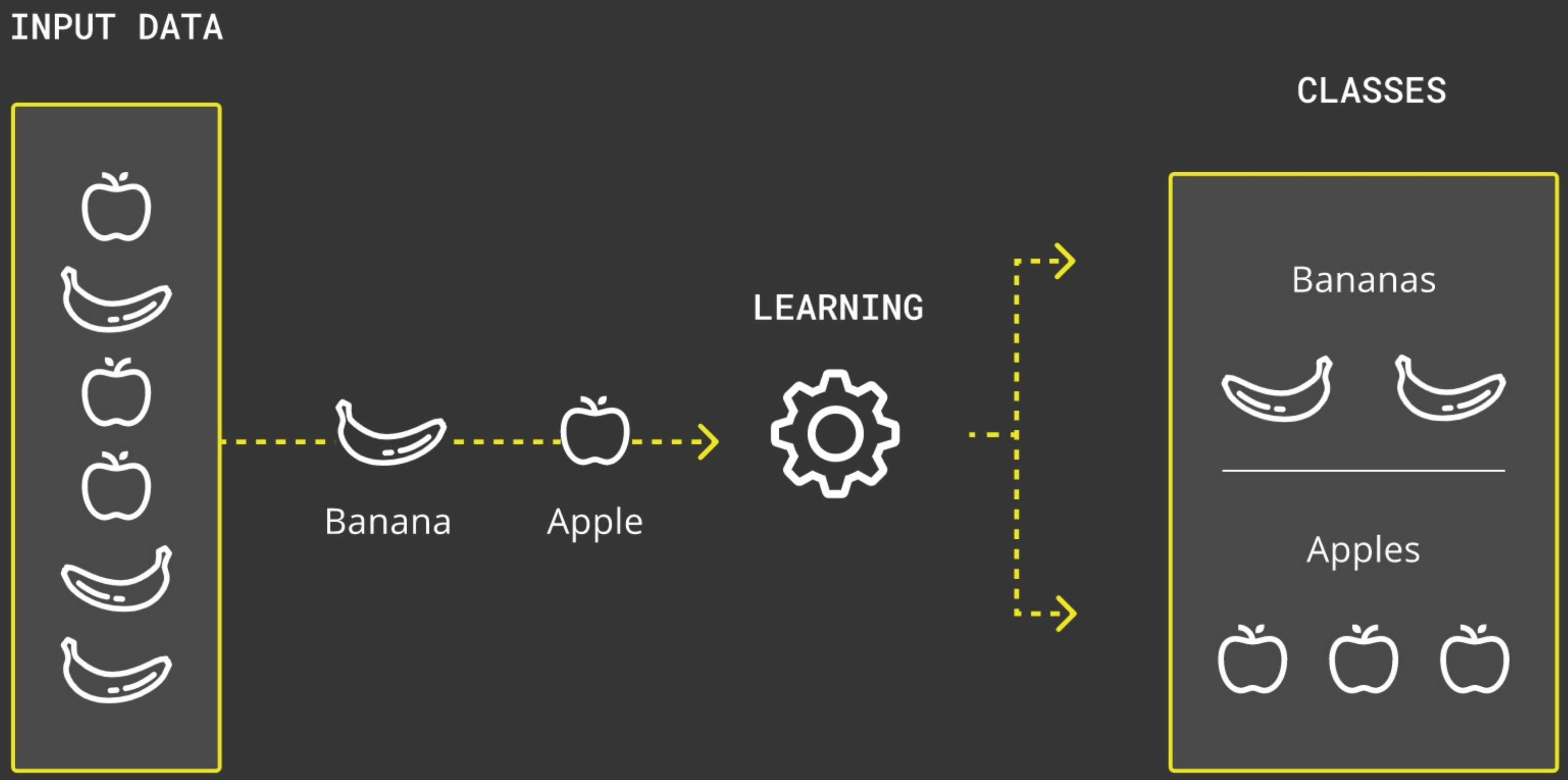
TensorFlow Lite for Microcontrollers / Gesture Recognition

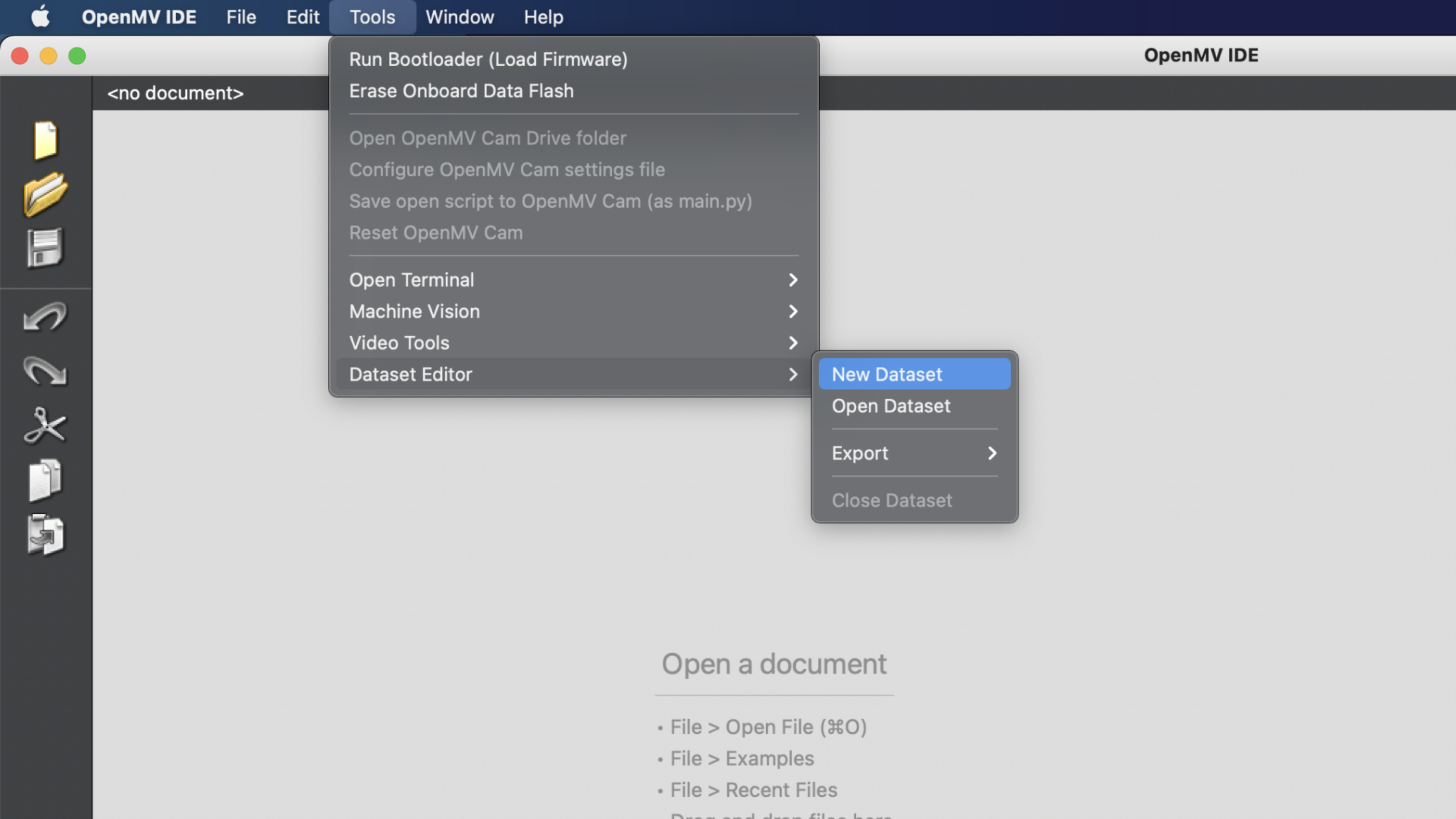


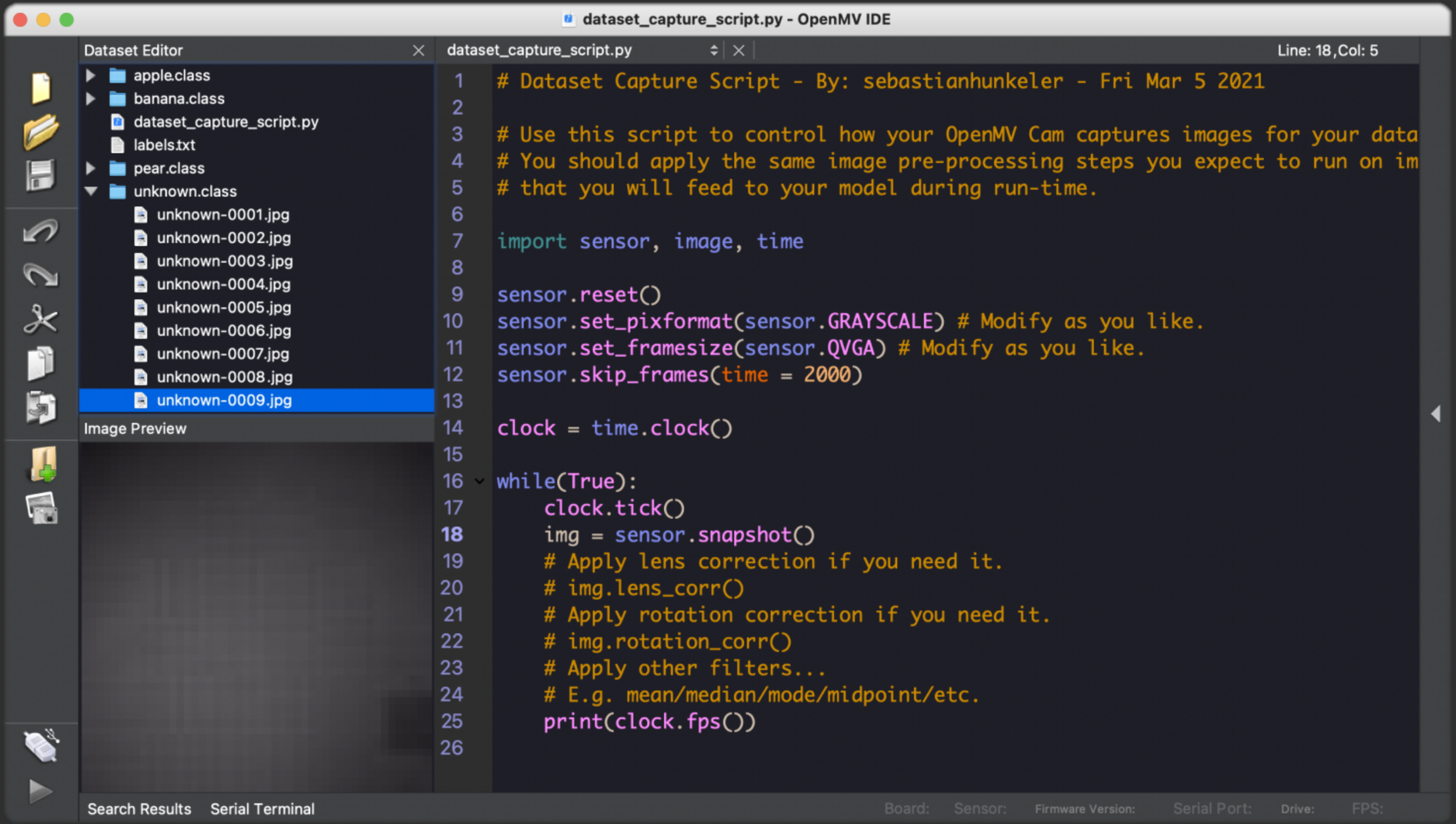
OpenMV / Tensorflow / Portenta Vision Shield / Image Recognition

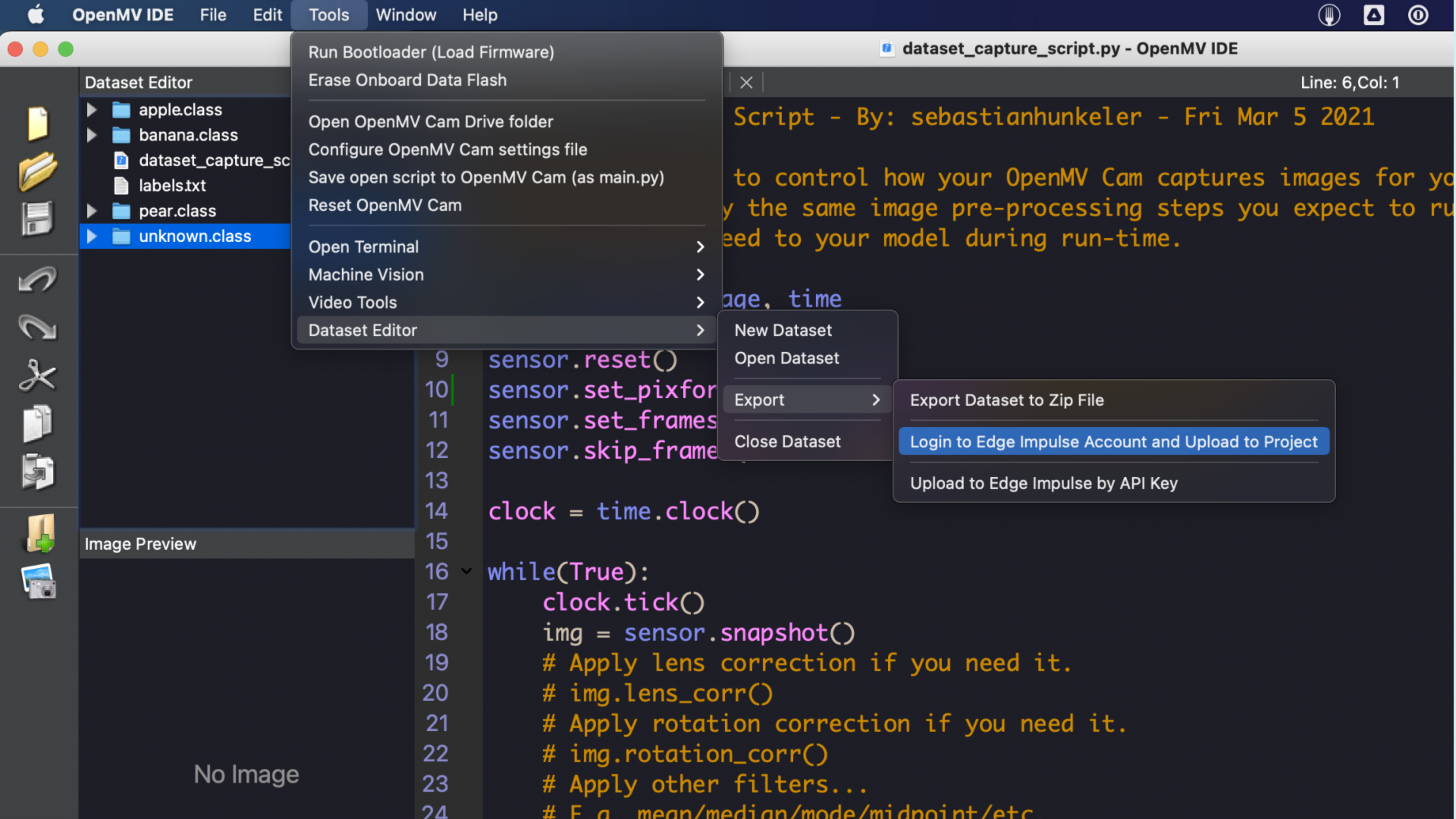
REQUIRED HARDWARE AND SOFTWARE

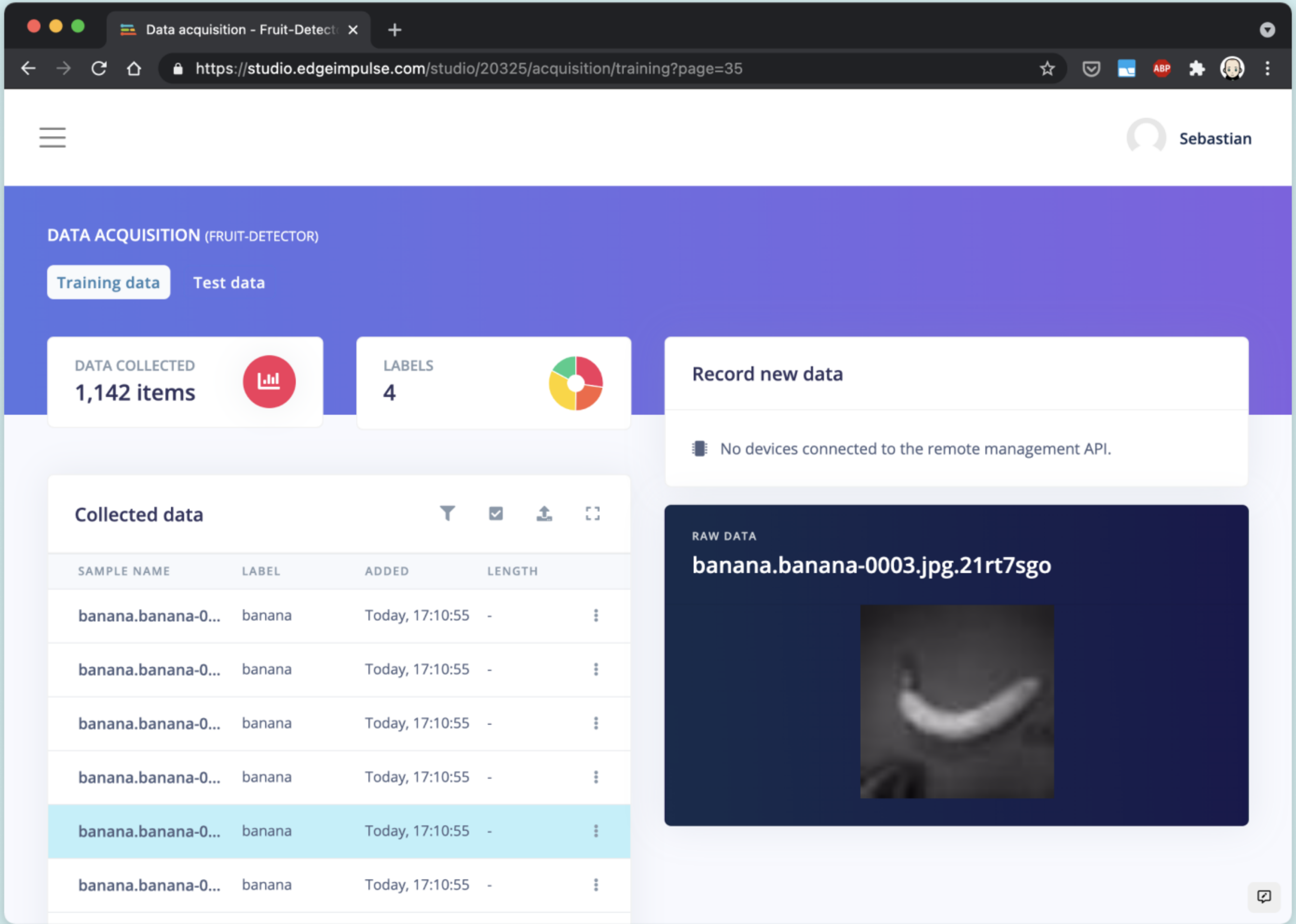
- Portenta H7 board
- Portenta Vision Shield - LoRa or Portenta Vision Shield - Ethernet
- USB-C cable (either USB-A to USB-C or USB-C to USB-C)
- An Edge Impulse account for training the ML model
- Fruits (or other objects) to create the classification model 🍏🍌🍐

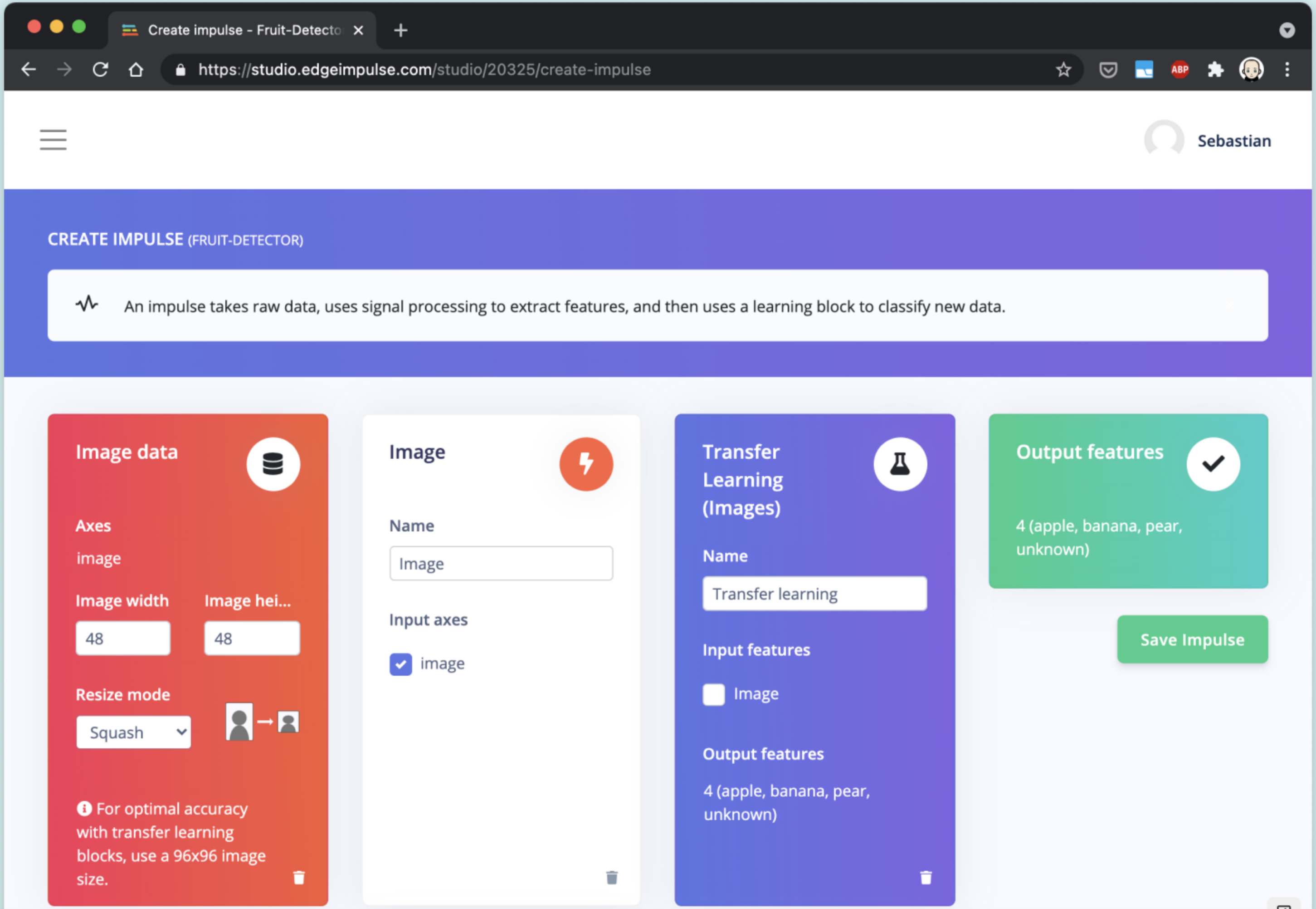

INPUT DATA
Banana
Apple
LEARNING
CLASSES
Bananas
Apples


OpenMV IDE
File
Edit
Tools
Window
Help
OpenMV IDE
<no document>
Run Bootloader (Load Firmware)
Erase Onboard Data Flash
Open OpenMV Cam Drive folder
Configure OpenMV Cam settings file
Save open script to OpenMV Cam (as main.py)
Reset OpenMV Cam
Open Terminal
Machine Vision
Video Tools
Dataset Editor
New Dataset
Open Dataset
Export
Close Dataset
Open a document
• File > Open File (⌘O)
• File > Examples
• File > Recent Files


dataset_capture_script.py - OpenMV IDE
Dataset Editor
apple.class
banana.class
dataset_capture_script.py
labels.txt
pear.class
unknown.class
unknown-0001.jpg
unknown-0002.jpg
unknown-0003.jpg
unknown-0004.jpg
unknown-0005.jpg
unknown-0006.jpg
unknown-0007.jpg
unknown-0008.jpg
unknown-0009.jpg
Image Preview
dataset_capture_script.py
Line: 18,Col: 5
# Dataset Capture Script - By: sebastianhunkeler - Fri Mar 5 2021
# Use this script to control how your OpenMV Cam captures images for your data
# You should apply the same image pre-processing steps you expect to run on im
# that you will feed to your model during run-time.
import sensor, image, time
sensor.reset()
sensor.set_pixformat(sensor.GRAYSCALE) # Modify as you like.
sensor.set_framesize(sensor.QVGA) # Modify as you like.
sensor.skip_frames(time = 2000)
clock = time.clock()
while(True):
clock.tick()
img = sensor.snapshot()
# Apply lens correction if you need it.
# img.lens_corr()
# Apply rotation correction if you need it.
# img.rotation_corr()
# Apply other filters...
# E.g. mean/median/mode/midpoint/etc.
print(clock.fps())
Search Results
Serial Terminal
Board:
Sensor:
Firmware Version:
Serial Port:
Drive:
FPS:


OpenMV IDE
File
Edit
Tools
Window
Help
dataset_capture_script.py - OpenMV IDE
Dataset Editor
apple.class
banana.class
dataset_capture_sc
labels.txt
pear.class
unknown.class
Image Preview
No Image
Run Bootloader (Load Firmware)
Erase Onboard Data Flash
Open OpenMV Cam Drive folder
Configure OpenMV Cam settings file
Save open script to OpenMV Cam (as main.py)
Reset OpenMV Cam
Open Terminal
Machine Vision
Video Tools
Dataset Editor
New Dataset
Open Dataset
Export
Close Dataset
Export Dataset to Zip File
Login to Edge Impulse Account and Upload to Project
Upload to Edge Impulse by API Key
Line: 6,Col: 1
Script - By: sebastianhunkeler - Fri Mar 5 2021
to control how your OpenMV Cam captures images for yo
y the same image pre-processing steps you expect to ru
eed to your model during run-time.
age, time
sensor.reset()
sensor.set_pixfor
sensor.set_frames
sensor.skip_frame
clock = time.clock()
while(True):
clock.tick()
img = sensor.snapshot()
# Apply lens correction if you need it.
# img.lens_corr()
# Apply rotation correction if you need it.
# img.rotation_corr()
# Apply other filters...


Data acquisition - Fruit-Detect
https://studio.edgeimpulse.com/studio/20325/acquisition/training?page=35
Sebastian
DATA ACQUISITION (FRUIT-DETECTOR)
Training data
Test data
DATA COLLECTED
1,142 items
LABELS
4
Record new data
No devices connected to the remote management API.
Collected data
SAMPLE NAME
LABEL
ADDED
LENGTH
banana.banana-0... banana Today, 17:10:55 -
banana.banana-0... banana Today, 17:10:55 -
banana.banana-0... banana Today, 17:10:55 -
banana.banana-0... banana Today, 17:10:55 -
banana.banana-0... banana Today, 17:10:55 -
banana.banana-0... banana Today, 17:10:55 -
RAW DATA
banana.banana-0003.jpg.21rt7sgo


Create impulse - Fruit-Detecto
https://studio.edgeimpulse.com/studio/20325/create-impulse
Sebastian
CREATE IMPULSE (FRUIT-DETECTOR)
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Image data
Axes
image
Image width
48
Image hei...
48
Resize mode
Squash
For optimal accuracy with transfer learning blocks, use a 96x96 image size.
Image
Name
Image
Input axes
image
Transfer Learning (Images)
Name
Transfer learning
Input features
Image
Output features
4 (apple, banana, pear, unknown)
Output features
4 (apple, banana, pear, unknown)
Save Impulse

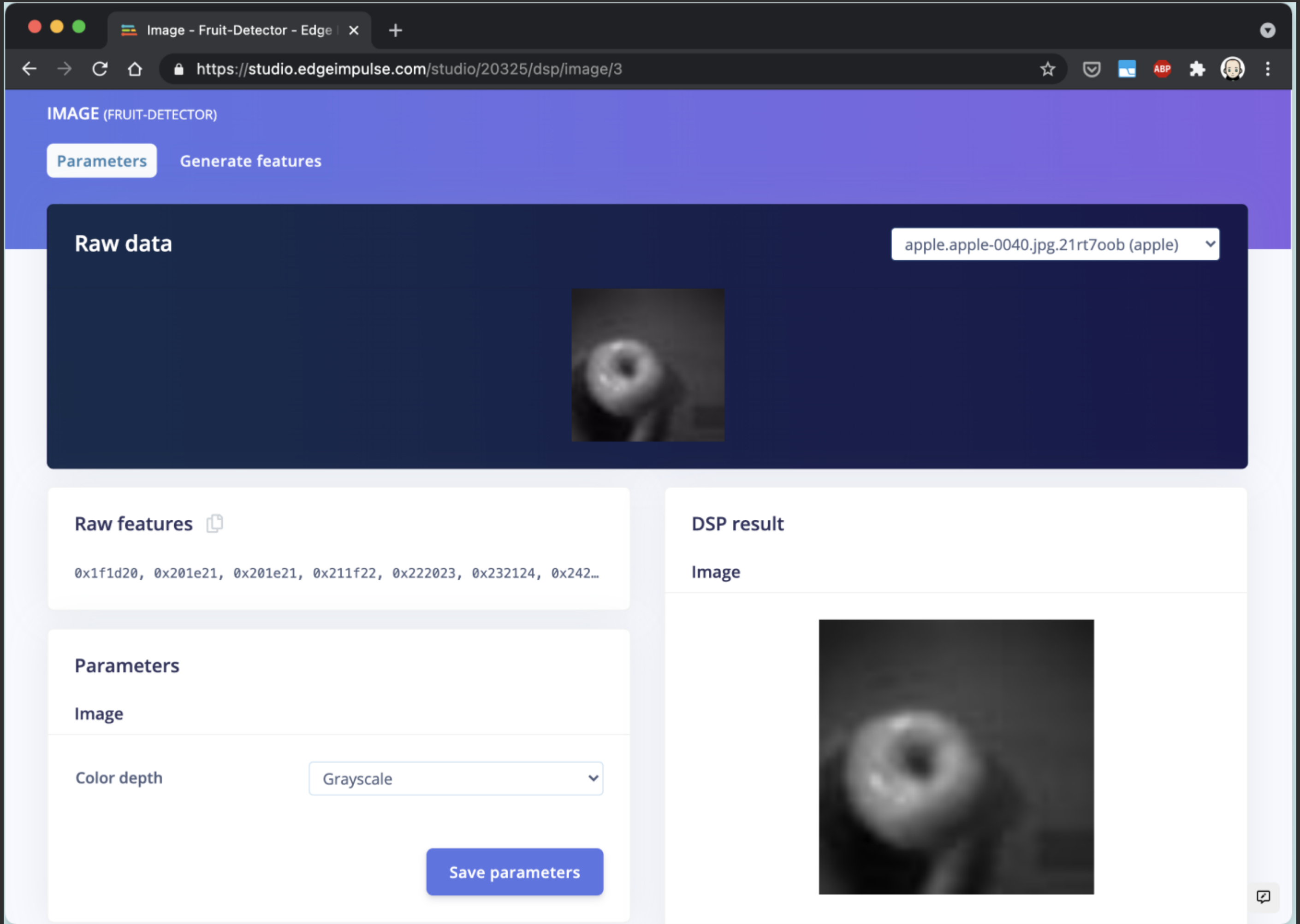


Image - Fruit-Detector - Edge I
https://studio.edgeimpulse.com/studio/20325/dsp/image/3
IMAGE (FRUIT-DETECTOR)
Parameters
Generate features
Raw data
apple.apple-0040.jpg.21rt7oob (apple)
Raw features
0x1f1d20, 0x201e21, 0x201e21, 0x211f22, 0x222023, 0x232124, 0x242…
DSP result
Image
Parameters
Image
Color depth
Grayscale
Save parameters

Image - Fruit-Detector - Edge I

https://studio.edgeimpulse.com/studio/20325/dsp/image/3/generate-features

Training set

Data in training set	1,142 items
Classes	4 (apple, banana, pear, unknown)

Generate features

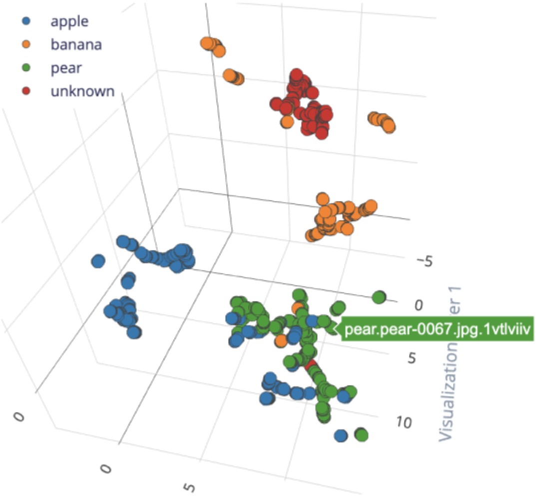
Feature generation output

```
Mon Mar 29 16:07:20 2021 Construct embedding
        completed  0  /  500 epochs
        completed  50  /  500 epochs
        completed  100  /  500 epochs
        completed  150  /  500 epochs
        completed  200  /  500 epochs
        completed  250  /  500 epochs
        completed  300  /  500 epochs
        completed  350  /  500 epochs
Still running...
        completed  400  /  500 epochs
        completed  450  /  500 epochs
Mon Mar 29 16:07:25 2021 Finished embedding
Reducing dimensions for visualizations OK

Job completed
```

Feature explorer (1,142 samples)

X Axis	Y Axis	Z Axis
Visualization lay	Visualization lay	Visualization lay

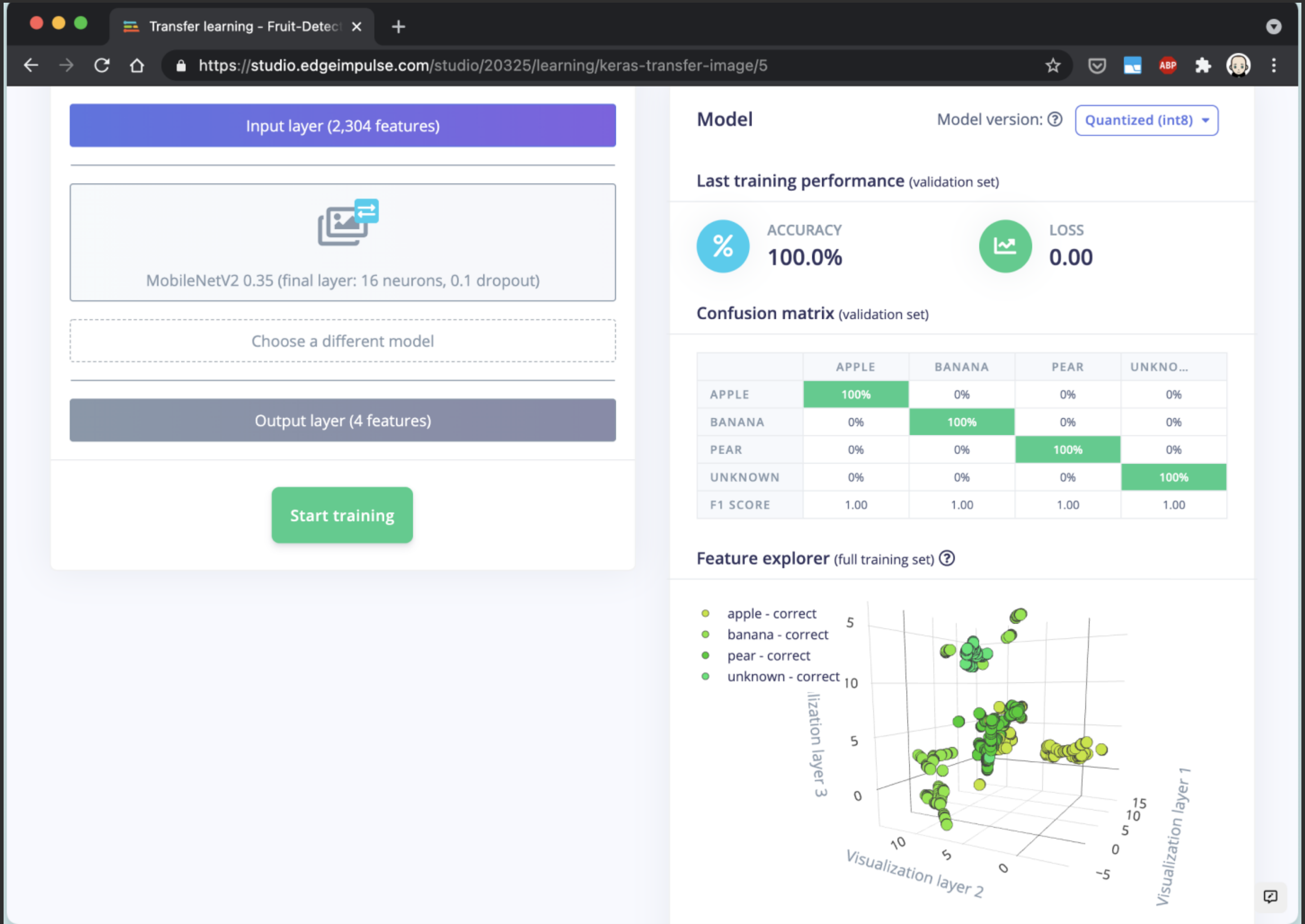


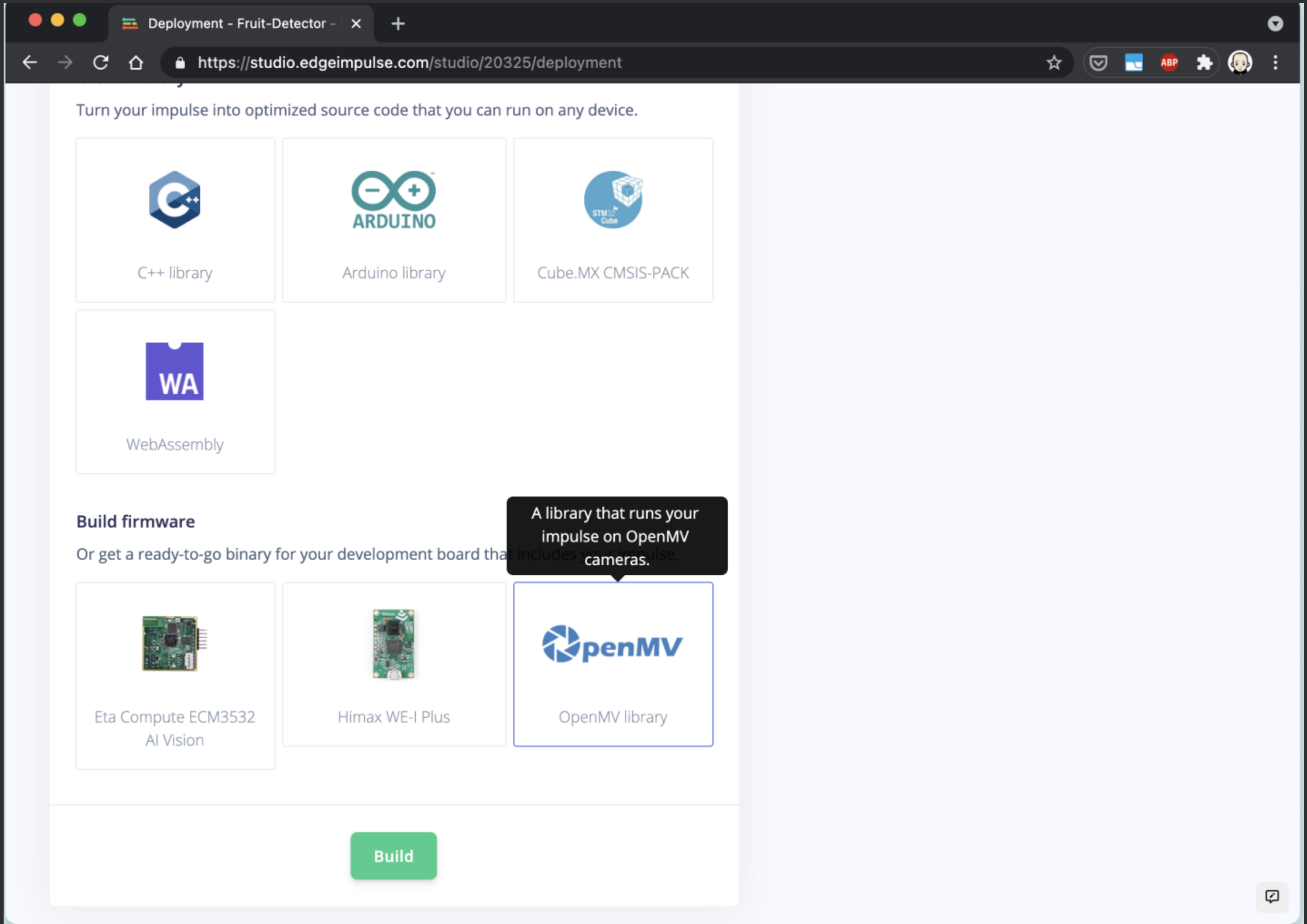
pear.pear-0067.jp...

Label: pear

View sample

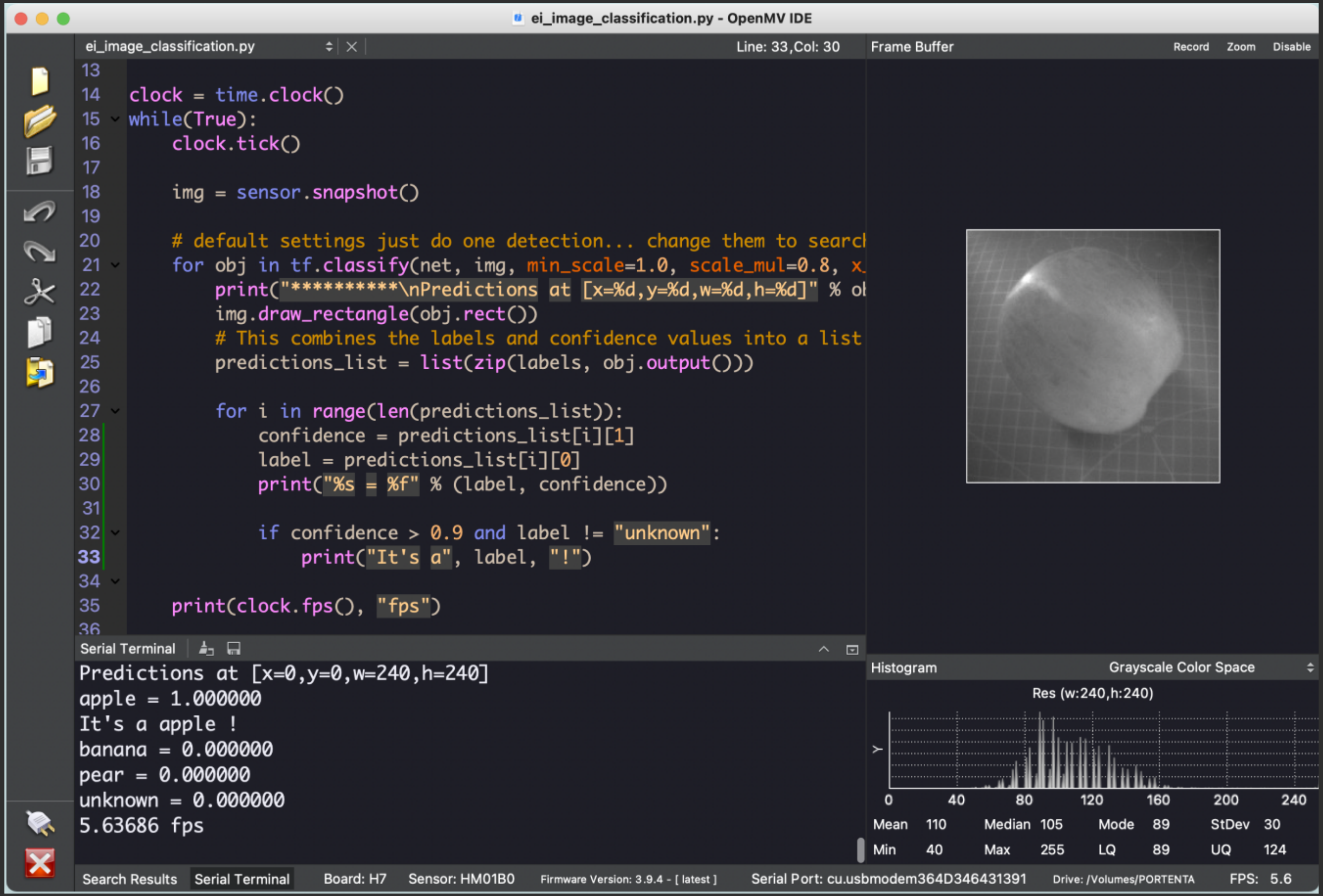
View features


Transfer learning - Fruit-Detect
https://studio.edgeimpulse.com/studio/20325/learning/keras-transfer-image/5
Input layer (2,304 features)
MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)
Choose a different model
Output layer (4 features)
Start training
Model
Model version:
Quantized (int8)
Last training performance (validation set)
ACCURACY
100.0%
LOSS
0.00
Confusion matrix (validation set)
APPLE BANANA PEAR UNKNO...
APPLE 100% 0% 0% 0%
BANANA 0% 100% 0% 0%
PEAR 0% 0% 100% 0%
UNKNOWN 0% 0% 0% 100%
F1 SCORE 1.00 1.00 1.00 1.00
Feature explorer (full training set)
apple - correct
banana - correct
pear - correct
unknown - correct
Visualization layer 1
Visualization layer 2
lization layer 3


Deployment - Fruit-Detector -
https://studio.edgeimpulse.com/studio/20325/deployment
Turn your impulse into optimized source code that you can run on any device.
C++ library
Arduino library
Cube.MX CMSIS-PACK
WebAssembly
Build firmware
Or get a ready-to-go binary for your development board that includes your impulse.
A library that runs your impulse on OpenMV cameras.
Eta Compute ECM3532 AI Vision
Himax WE-I Plus
OpenMV library
Build

```python
confidence = predictions_list[i][1]
label = predictions_list[i][0]
print("%s = %f" % (label[2:], confidence))

if confidence > 0.9 and label != "unknown":
    print("It's a ", label, "!")
```


ei_image_classification.py - OpenMV IDE
ei_image_classification.py
Line: 33, Col: 30
Frame Buffer
Record
Zoom
Disable
clock = time.clock()
while(True):
clock.tick()
img = sensor.snapshot()
# default settings just do one detection... change them to searcl
for obj in tf.classify(net, img, min_scale=1.0, scale_mul=0.8, x
print("**********\nPredictions at [x=%d,y=%d,w=%d,h=%d]" % ol
img.draw_rectangle(obj.rect())
# This combines the labels and confidence values into a list
predictions_list = list(zip(labels, obj.output()))
for i in range(len(predictions_list)):
confidence = predictions_list[i][1]
label = predictions_list[i][0]
print("%s = %f" % (label, confidence))
if confidence > 0.9 and label != "unknown":
print("It's a", label, "!")
print(clock.fps(), "fps")
Serial Terminal
Predictions at [x=0,y=0,w=240,h=240]
apple = 1.000000
It's a apple !
banana = 0.000000
pear = 0.000000
unknown = 0.000000
5.63686 fps
Histogram
Grayscale Color Space
Res (w:240,h:240)
0 40 80 120 160 200 240
Mean 110 Median 105 Mode 89 StDev 30
Min 40 Max 255 LQ 89 UQ 124
Search Results
Serial Terminal
Board: H7
Sensor: HM01B0
Firmware Version: 3.9.4 - [latest]
Serial Port: cu.usbmodem364D346431391
Drive: /Volumes/PORTENTA
FPS: 5.6

https://www.arduino.cc/pro/tutorials/portenta-h7/vs-openmv-ml https://bit.ly/portenta_person_detection

EDGE IMPULSE Home Log In

Guides > Responding to your voice Search ⌘K

DOCUMENTATION

- Getting Started
- API and SDK references
- What is embedded ML, anyway?

DEVELOPMENT BOARDS

- Overview
- ST B-L475E-IOT01A
- Arduino Nano 33 BLE Sense
- Eta Compute ECM3532 AI Sensor
- Eta Compute ECM3532 AI Vision
- OpenMV Cam H7 Plus
- Himax WE-I Plus
- Nordic Semi nRF52840 DK
- Nordic Semi nRF5340 DK
- SiLabs Thunderboard Sense 2
- Arduino Portenta H7 + Vision shield (preview)
- Raspberry Pi 4
- NVIDIA Jetson Nano
- Mobile phone
- Porting guide

EDGE IMPULSE FOR LINUX

- Edge Impulse for Linux
- Node.js SDK
- Go SDK

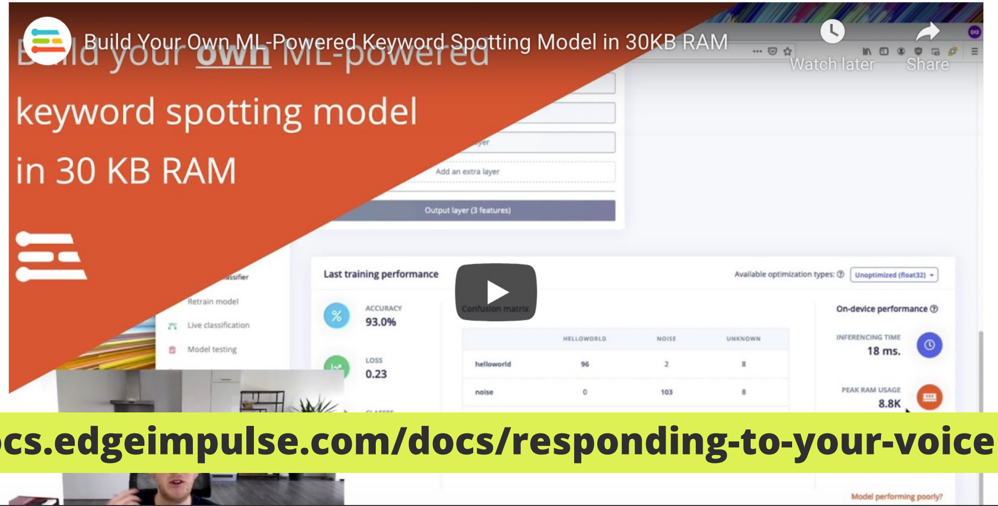
Responding to your voice

In this tutorial, you'll use machine learning to build a system that can recognize audible events, particularly your voice through *audio classification*. The system you create will work similarly to "Hey Siri" or "OK, Google" and is able to recognize keywords or other audible events, even in the presence of other background noise or background chatter.

You'll learn how to collect audio data from microphones, use signal processing to extract the most important information, and train a deep neural network that can tell you whether your keyword was heard in a given clip of audio. Finally, you'll deploy the system to an embedded device and evaluate how well it works.

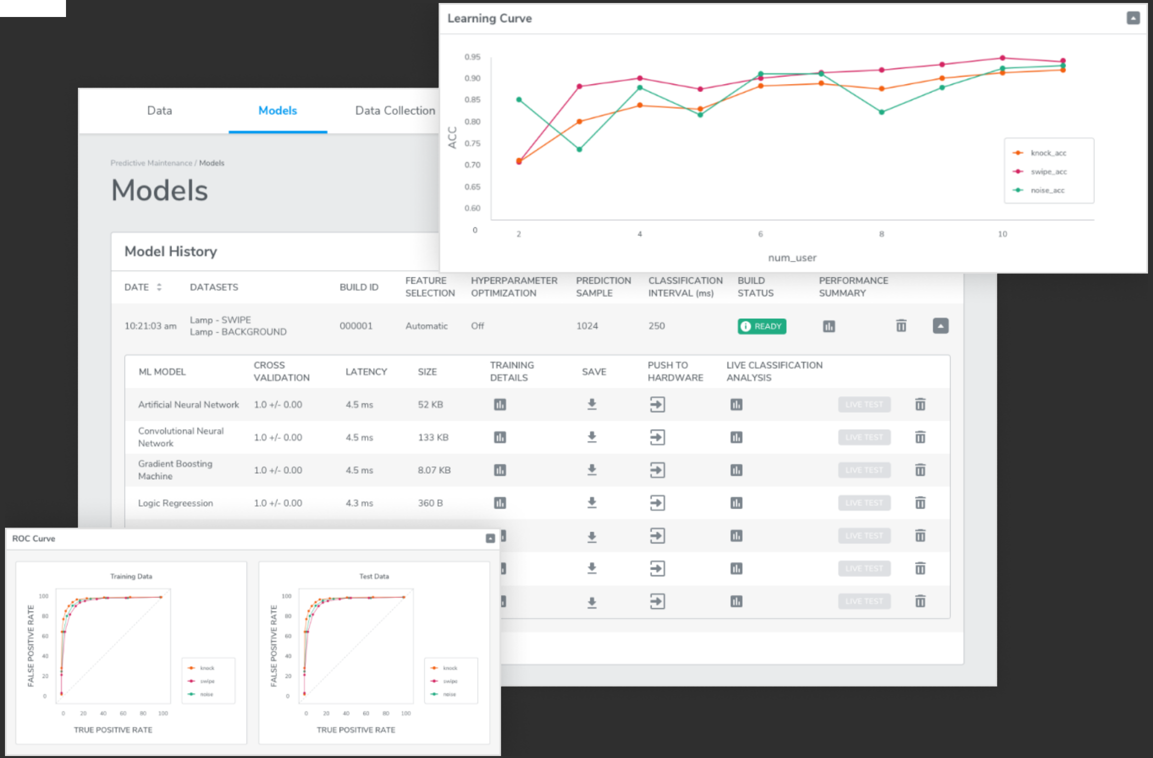
At the end of this tutorial, you'll have a firm understanding of how to classify audio using Edge Impulse.

There is also a video version of this tutorial:



Keyword Spotting / https://docs.edgeimpulse.com/docs/responding-to-your-voice

https://docs.edgeimpulse.com/docs



Qeexo / https://github.com/qeexo/Qeexo_AutoML_Arduino_Nano33BLE_pub

That's a wrap,
Thank you!

Massimo Banzi

askmassimo@arduino.cc

Copyright Notice