Hardware Architectures for Embedded and Edge AI

Prof Manuel Roveri – manuel.roveri@polimi.it
Massimo Pavan – massimo.pavan@polimi.it

Exercise session 3 – Data collection with Edge Impulse

Requirements

- Chrome web browser
- Arduino CLI
  - Win: https://arduino.github.io/arduino-cli/0.31/installation/#download
  - Lin: https://lindevs.com/install-arduino-cli-on-ubuntu
- Data collection firmware
  - https://cdn.edgeimpulse.com/firmware/arduino-nano-33-ble-sense.zip
- You'll need an account at
  - https://www.edgeimpulse.com/

Outline of today

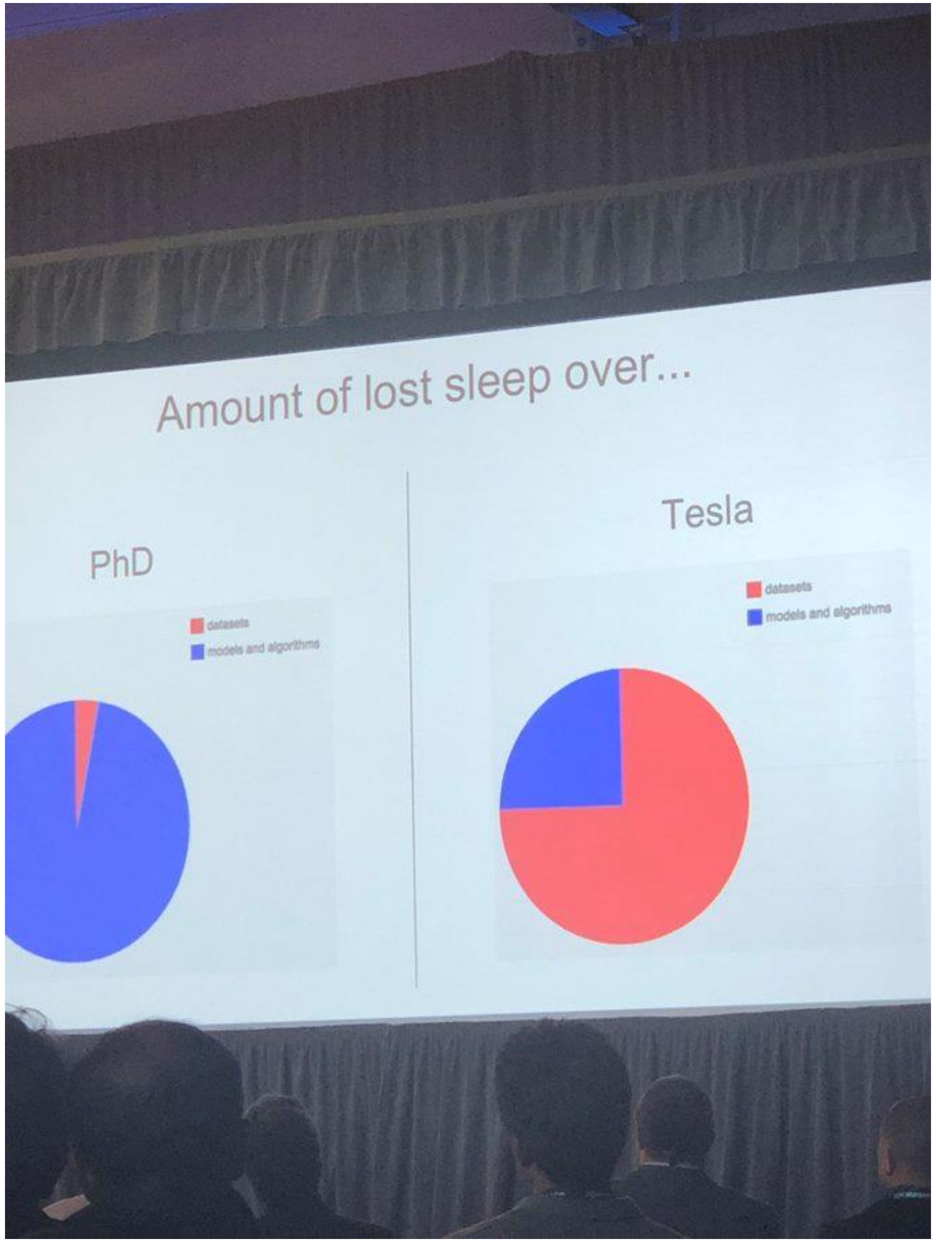
Collect a reasonably dimensioned dataset to perform object detection:

- Data collection
- Data labelling
- Data validation

We'll see the example of object detection/classification.

Is collecting data for object detection/classification a challenging task?

It is, for the same reasons that make object detection a challenging task!



https://petewarden.com/2018/05/28/why-you-need-to-improve-your-training-data-and-how-to-do-it/

Challenges: illumination

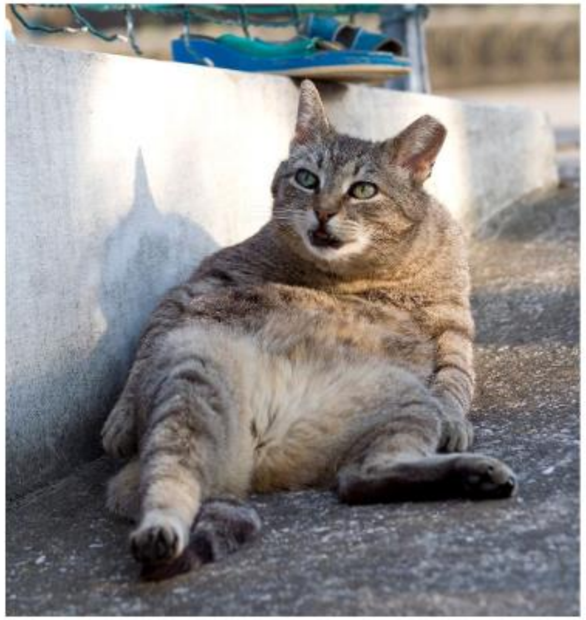
This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

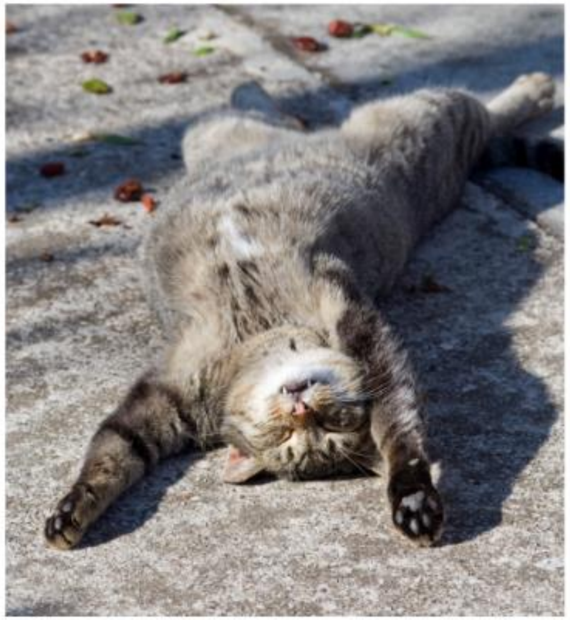
This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

Challenges: deformation

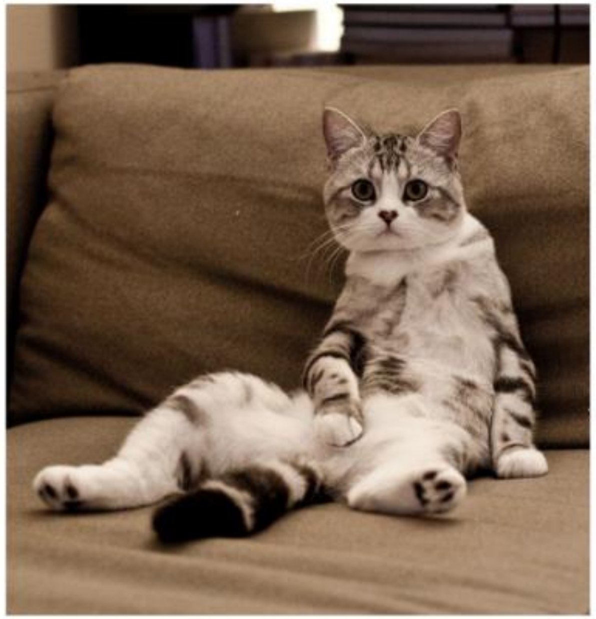


This image by Umberto Salvagnin is licensed under CC-BY 2.0



This image by Umberto Salvagnin is licensed under CC-BY 2.0

This image by sare bear is licensed under CC-BY 2.0



This image by Tom Thai is licensed under CC-BY 2.0

Challenges: occlusion

Challenges: view point change

Camera movement

Challenges: inter-class variability

Challenges: scale variation

Dataset construction for image detection/classification

- A deep learning algorithm can address all this challenges only if many possible examples of these challenges are present in the dataset.
- Thus, a generic image detection/classification algorithm meant to run in unkown conditions, should be trained on data with:
  - Different illuminations, view points
  - Occluded and deformed targets
  - Examples of different objects of the same class

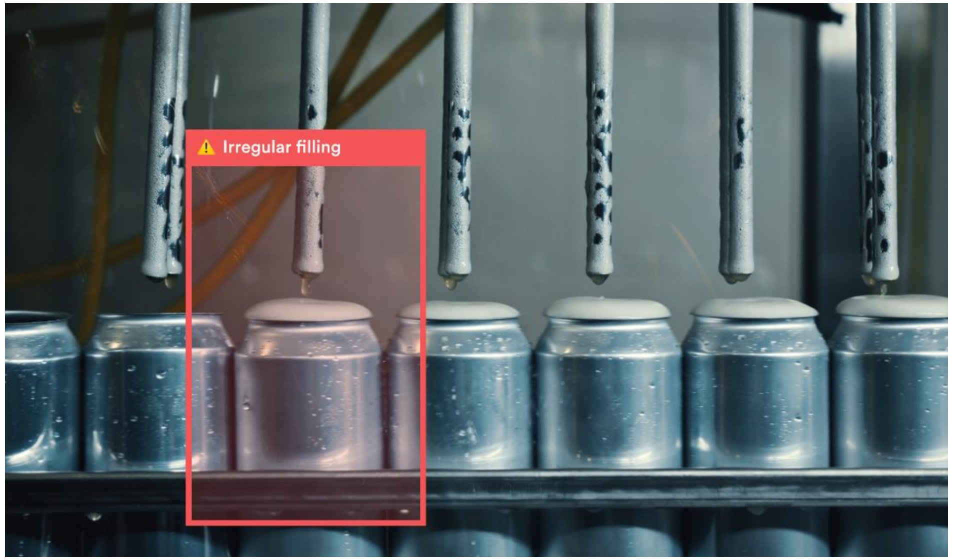
Image classification in the context of tinyML

Very often in the context of tinyML you do not have to deal with all the presented challenges at the same time:

- You can collect (some) data from the sensors that will be used on the device, in the environment in which it will be deployed.
- This can simplify a lot the learning phase for the algorithm.
- It's possible to start from simple conditions, then augment the capabilities of your algorithm by adding new data collected under new conditions.
- In some applications, it may be possible to even make the object that you want to detect more evident (e.g., coloring it, using objects with unusual shapes…)

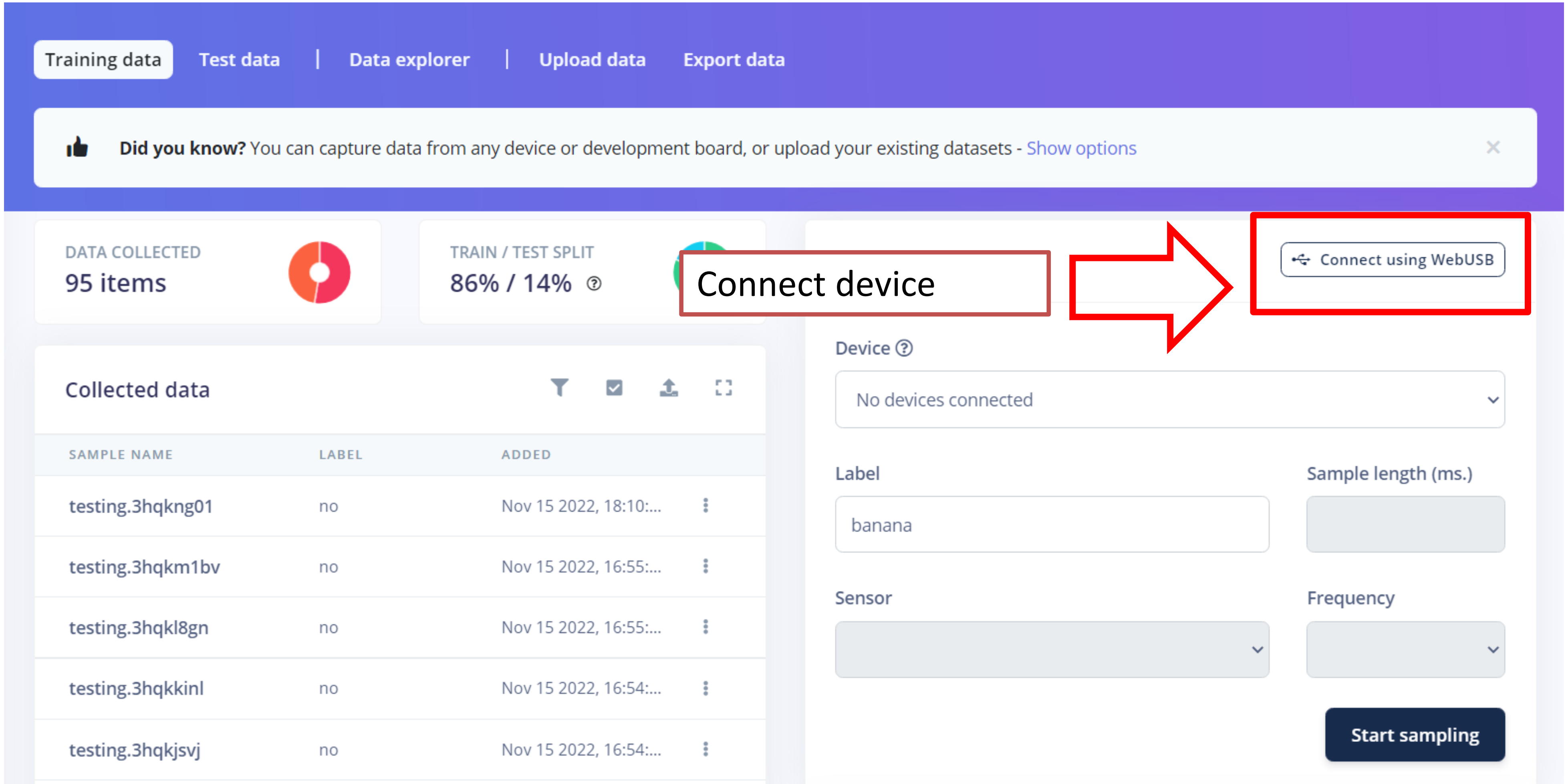
Why this approach

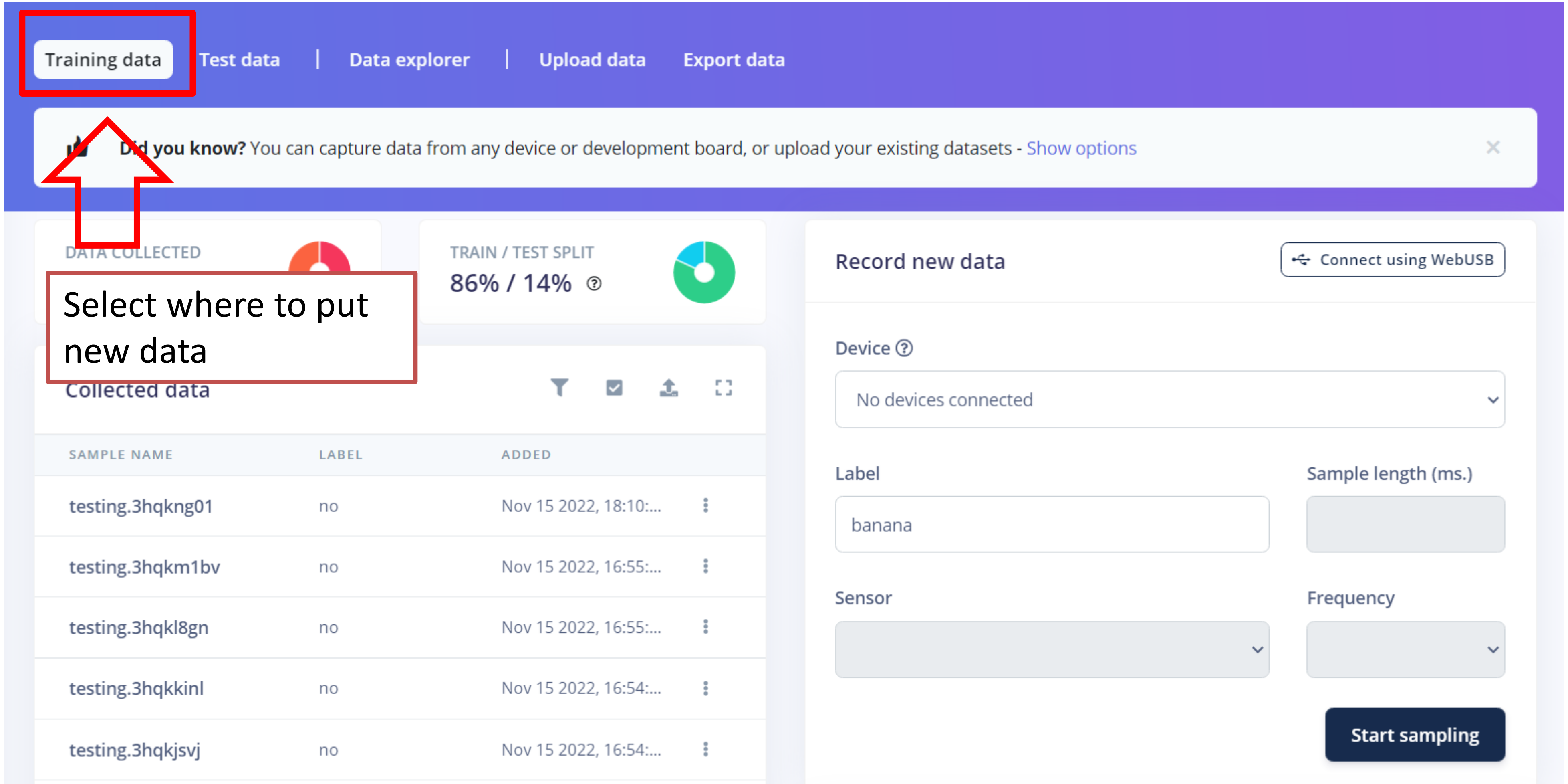
- A complete model that can work in any condition on tasks of arbitrary complexity with reasonable performances may be too memory/energy demanding for our target devices, and difficult to obtain (expecially in the time of an exercise session ☺)
- We can exploit the fact that we are collecting the data with the same device that will run the algorithm, in the same condition
- An algorithm that works in some well defined conditions can still be extremely useful (e.g.: computer vision in the industry, anomaly detection …)

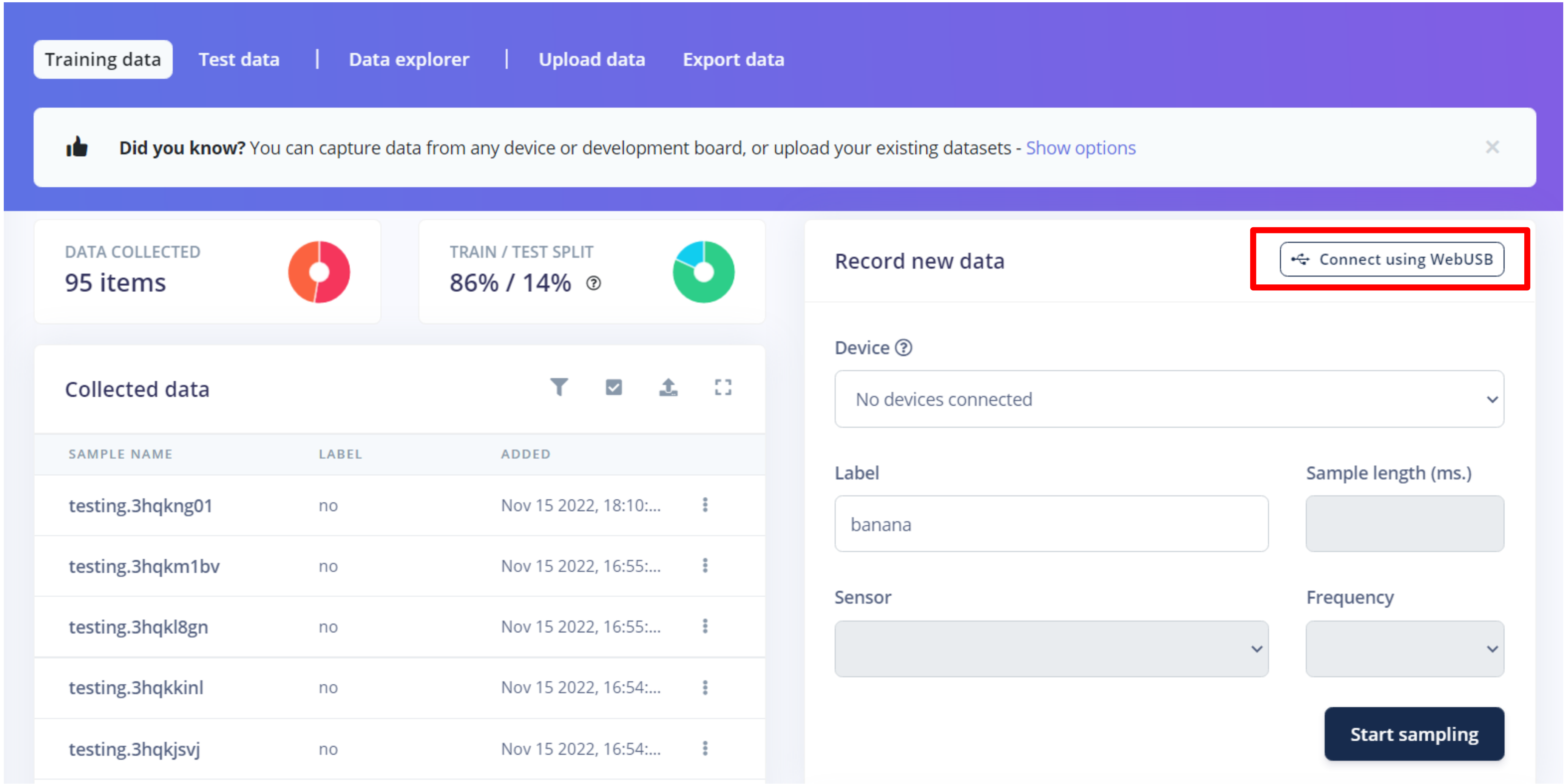


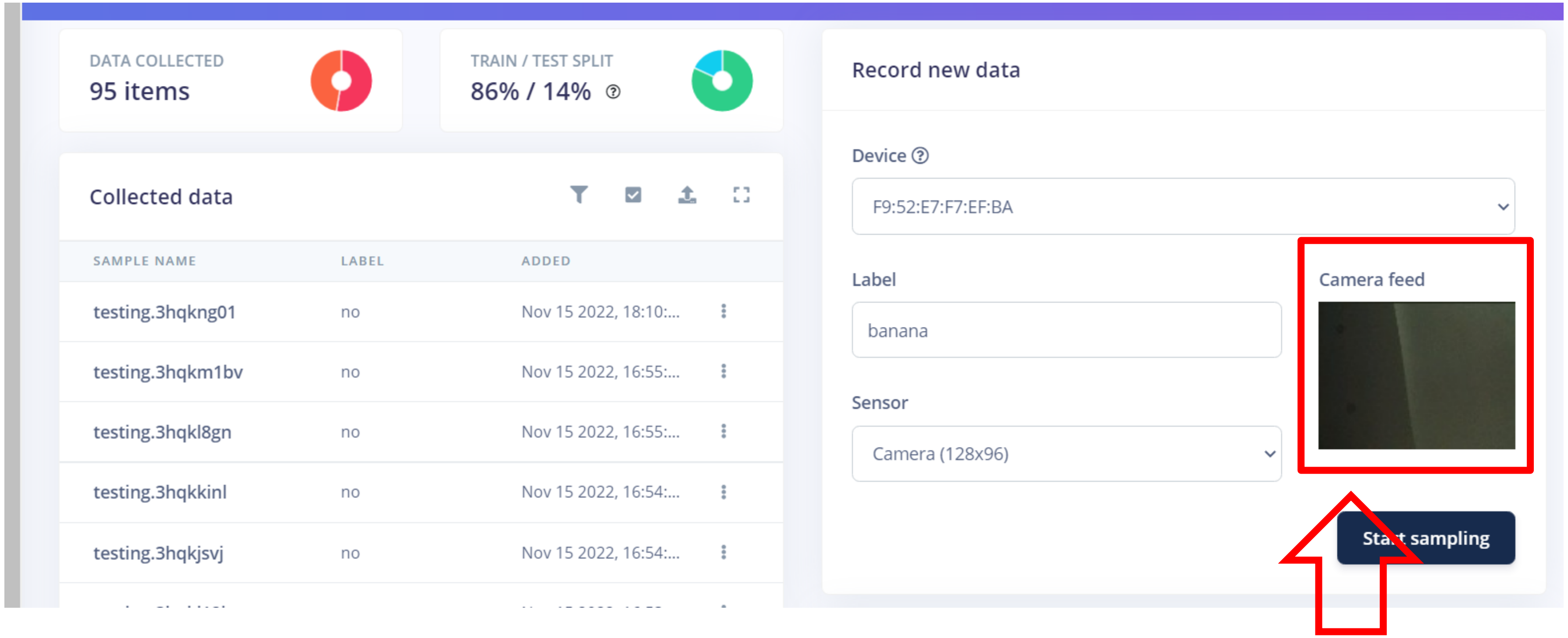
POLITECNICO MILANO 1863

Data collection with edge impulse

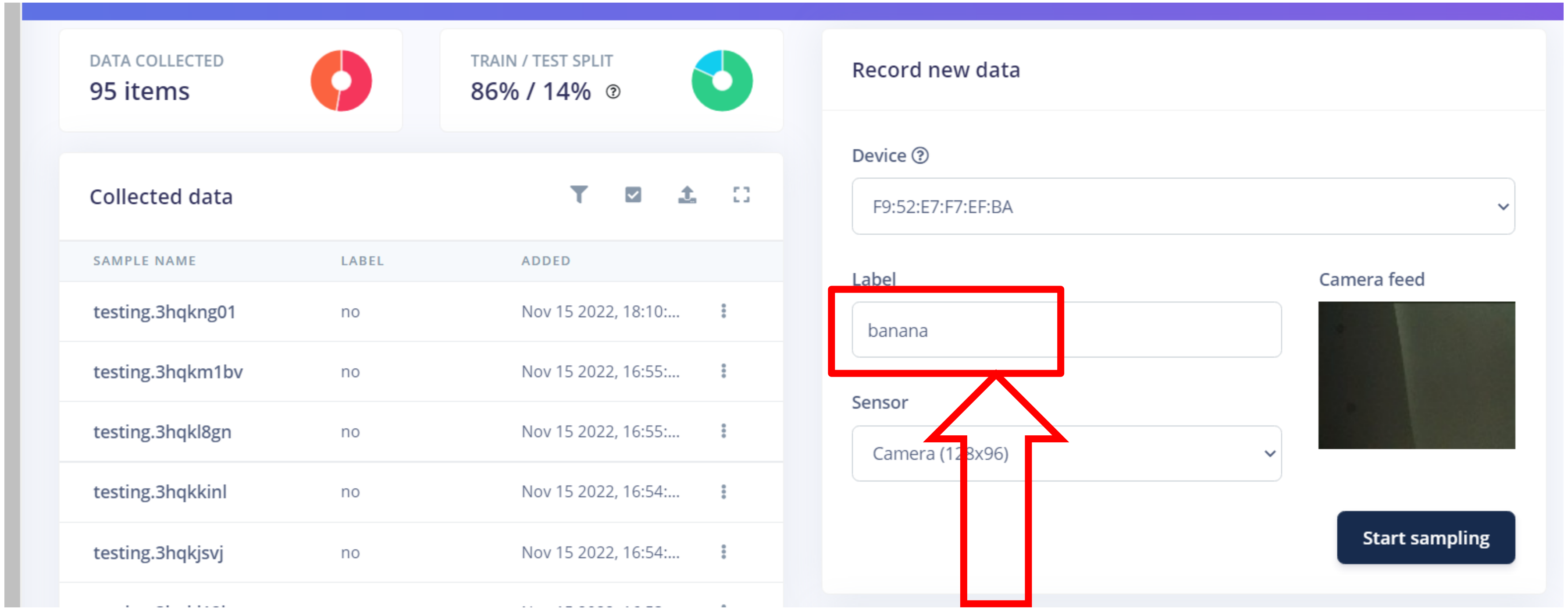

Training data
Test data
Data explorer
Upload data
Export data
Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options
DATA COLLECTED
95 items
TRAIN / TEST SPLIT
86% / 14%
Connect device
Connect using WebUSB
Collected data
SAMPLE NAME
LABEL
ADDED
testing.3hqkng01
no
Nov 15 2022, 18:10:...
testing.3hqkm1bv
no
Nov 15 2022, 16:55:...
testing.3hqkl8gn
no
Nov 15 2022, 16:55:...
testing.3hqkkinl
no
Nov 15 2022, 16:54:...
testing.3hqkjsvj
no
Nov 15 2022, 16:54:...
Device
No devices connected
Label
banana
Sample length (ms.)
Sensor
Frequency
Start sampling


Training data
Test data
Data explorer
Upload data
Export data
Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options
DATA COLLECTED
TRAIN / TEST SPLIT
86% / 14%
Select where to put new data
Collected data
SAMPLE NAME
LABEL
ADDED
testing.3hqkng01
no
Nov 15 2022, 18:10:...
testing.3hqkm1bv
no
Nov 15 2022, 16:55:...
testing.3hqkl8gn
no
Nov 15 2022, 16:55:...
testing.3hqkkinl
no
Nov 15 2022, 16:54:...
testing.3hqkjsvj
no
Nov 15 2022, 16:54:...
Record new data
Connect using WebUSB
Device
No devices connected
Label
banana
Sample length (ms.)
Sensor
Frequency
Start sampling


Training data
Test data
Data explorer
Upload data
Export data
Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options
DATA COLLECTED
95 items
TRAIN / TEST SPLIT
86% / 14%
Record new data
Connect using WebUSB
Collected data
SAMPLE NAME
LABEL
ADDED
testing.3hqkng01
no
Nov 15 2022, 18:10:...
testing.3hqkm1bv
no
Nov 15 2022, 16:55:...
testing.3hqkl8gn
no
Nov 15 2022, 16:55:...
testing.3hqkkinl
no
Nov 15 2022, 16:54:...
testing.3hqkjsvj
no
Nov 15 2022, 16:54:...
Device
No devices connected
Label
banana
Sample length (ms.)
Sensor
Frequency
Start sampling

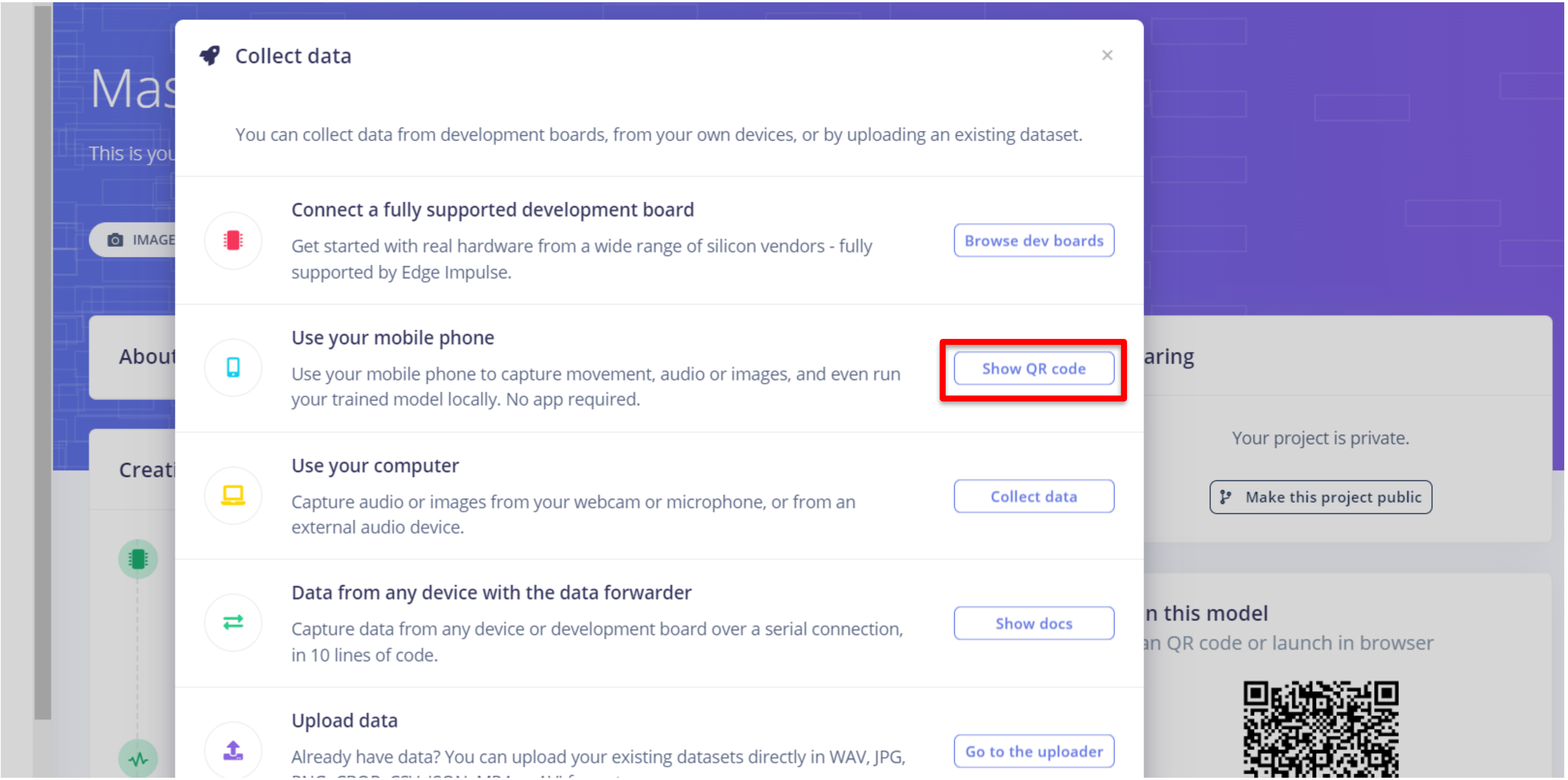


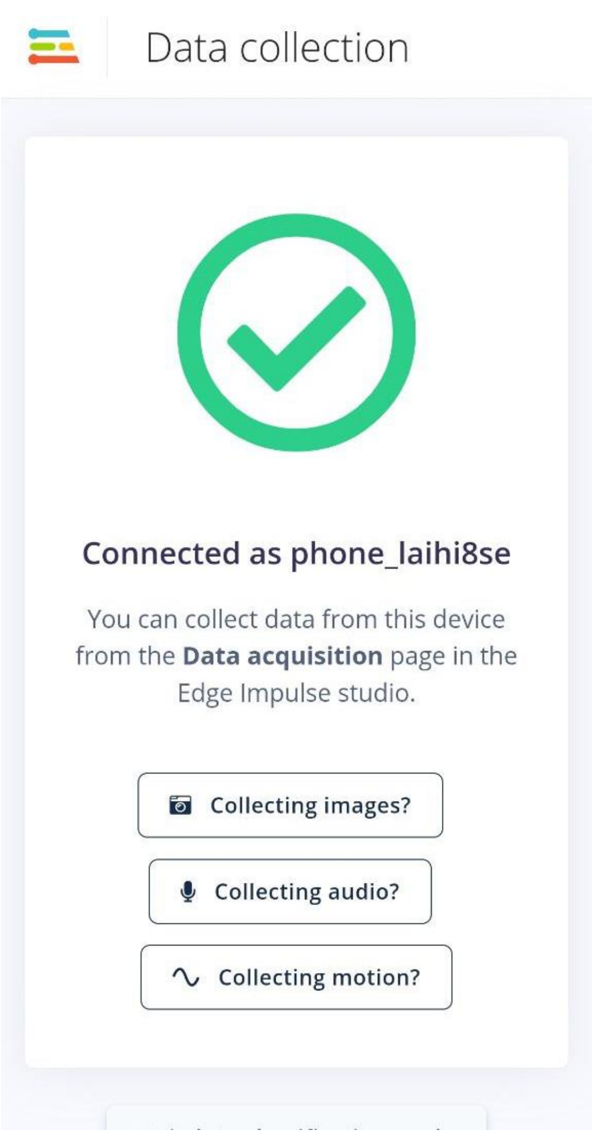
The picture you'll take

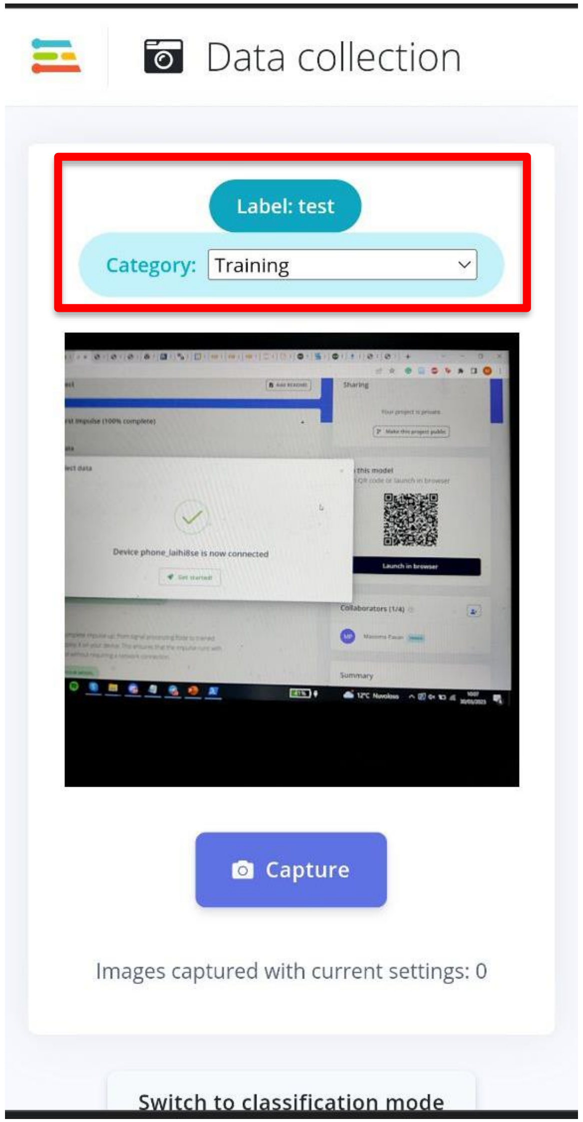


The label to assign to the picture

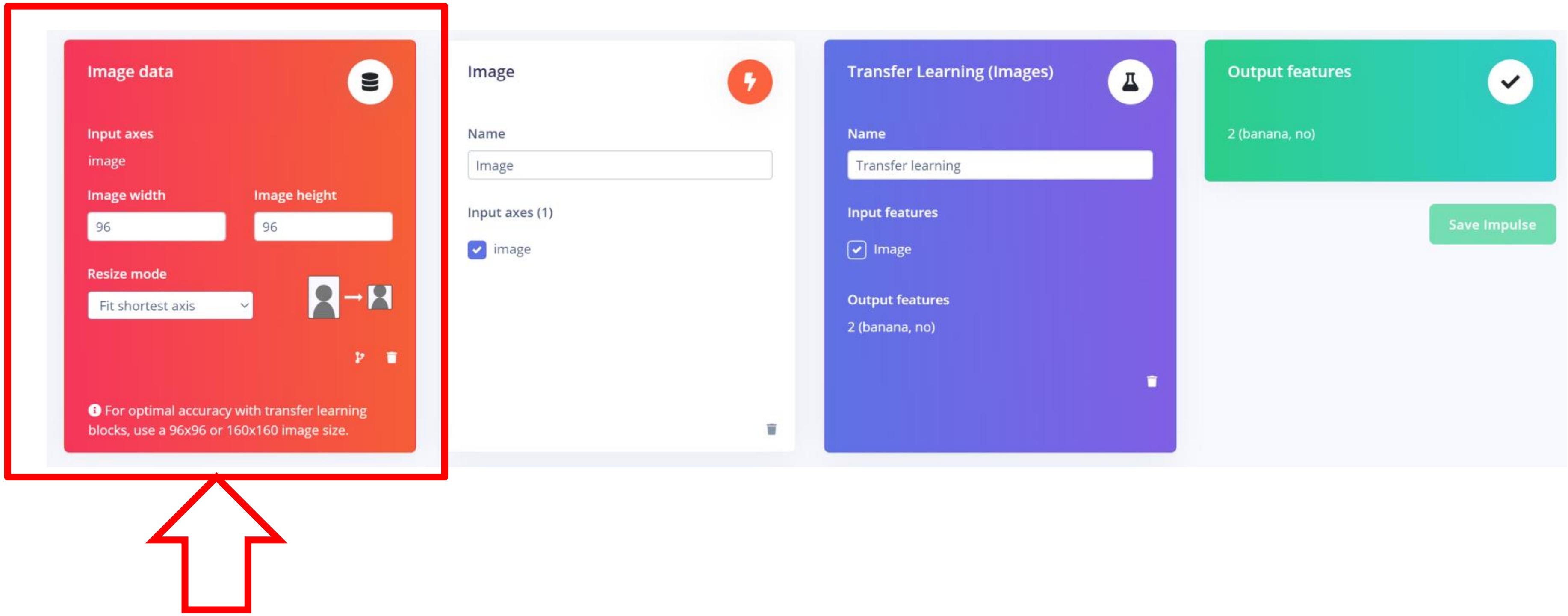
Use mobile phone to faster collect the data (but they'll be different!)




Data collection
Connected as phone_laihi8se
You can collect data from this device from the Data acquisition page in the Edge Impulse studio.
Collecting images?
Collecting audio?
Collecting motion?

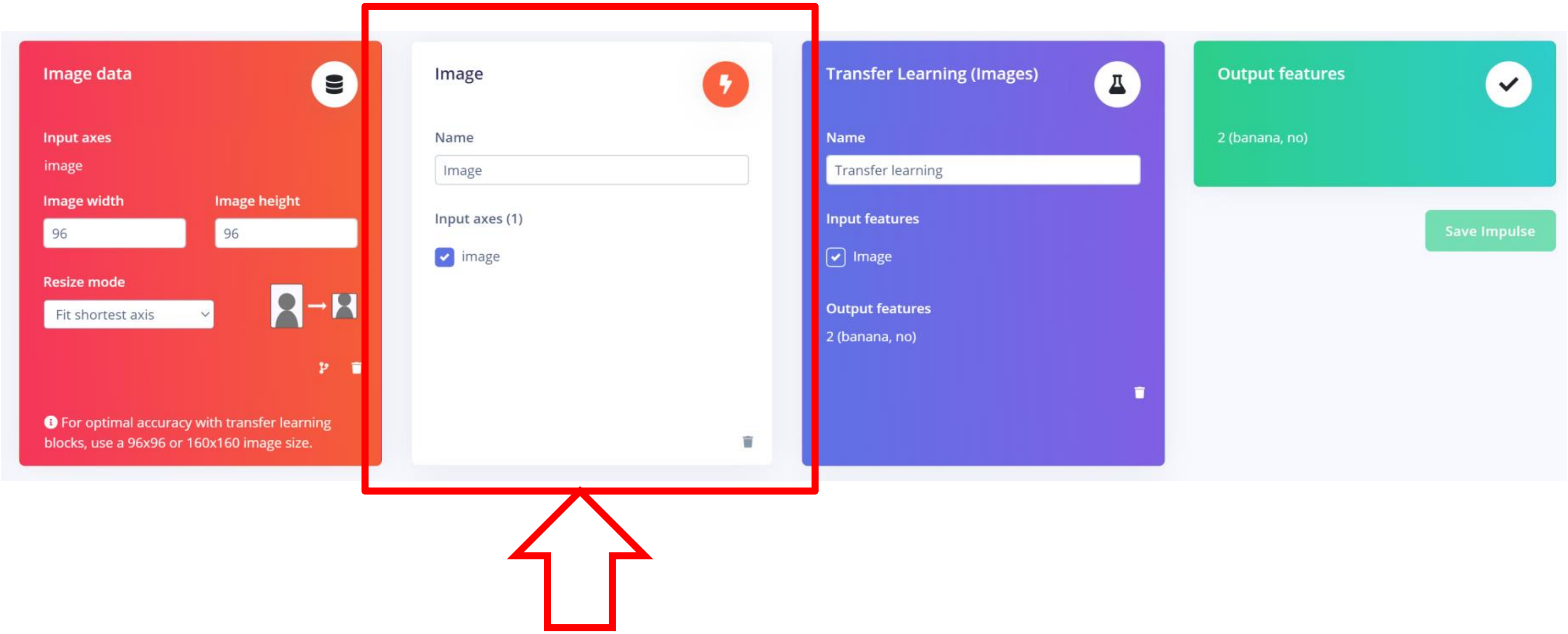

Data collection
Label: test
Category: Training
Capture
Images captured with current settings: 0
Switch to classification mode

Impulse design



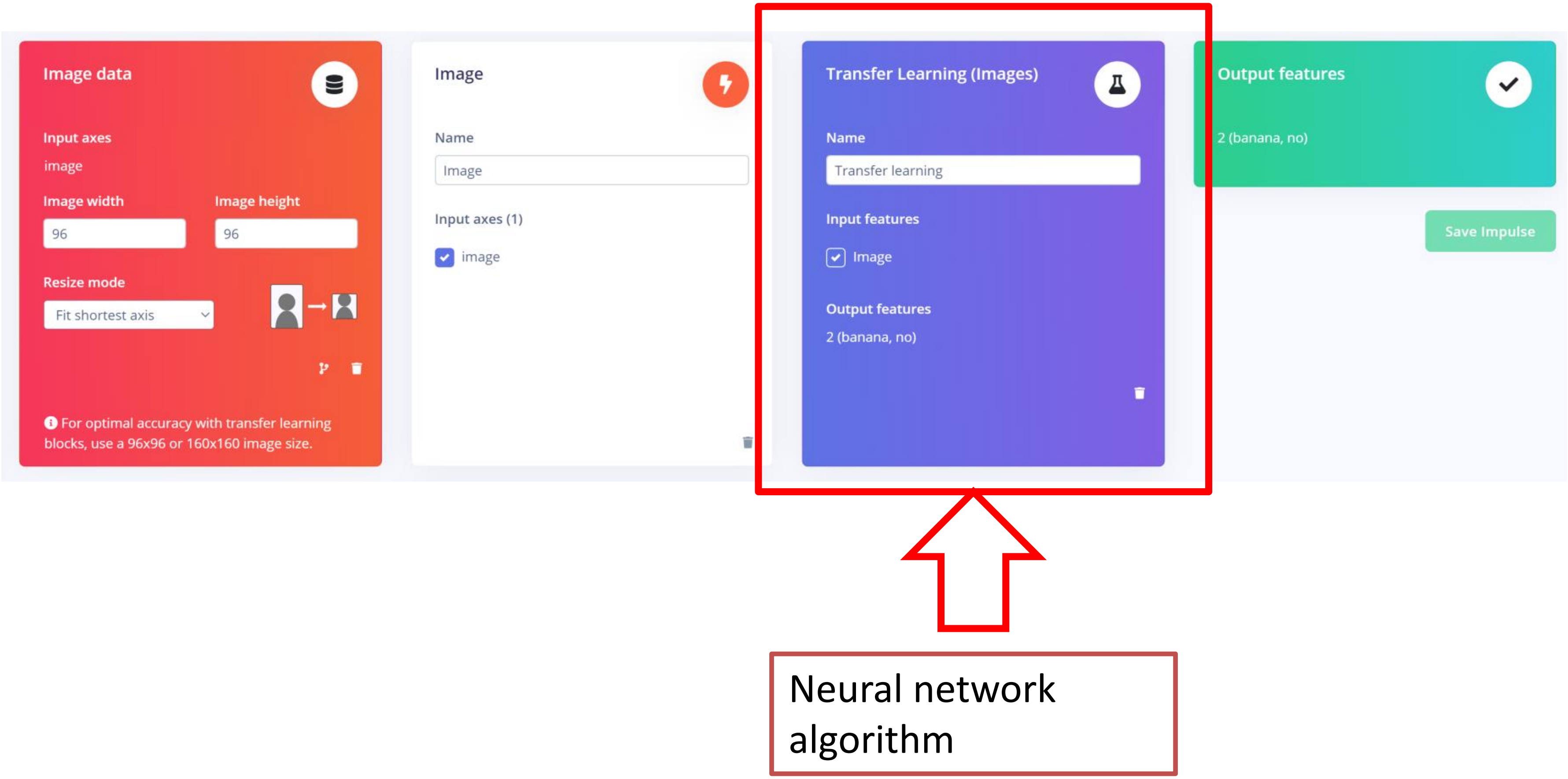
Resize of the picture taken

Impulse design

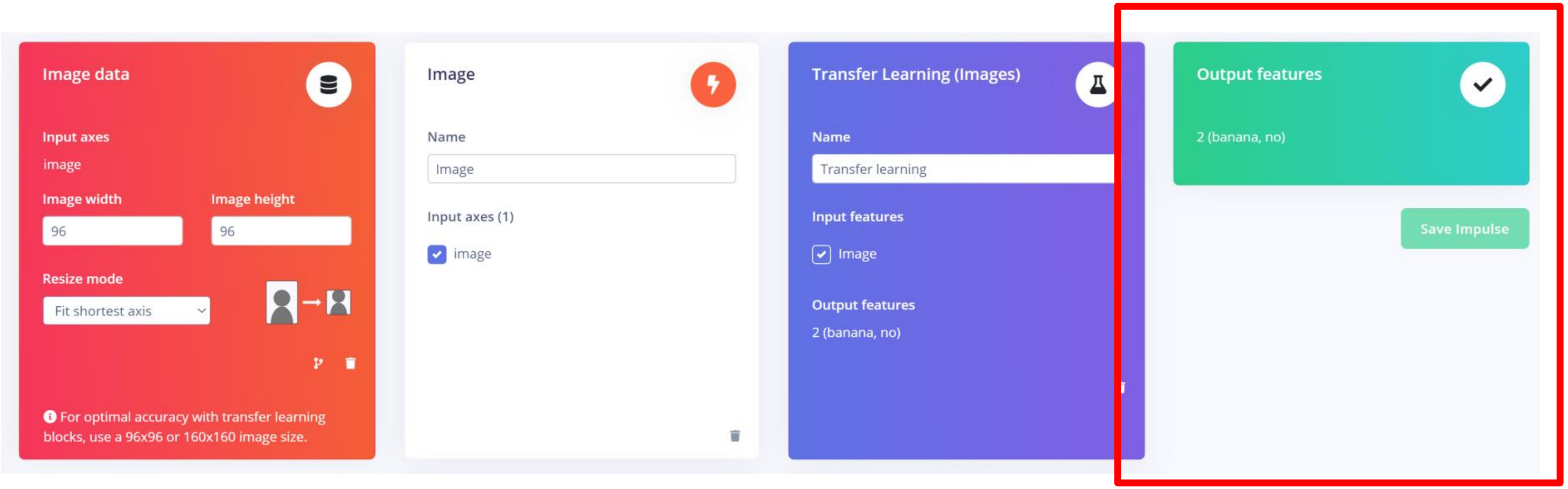


Preprocessing of picture

Impulse design

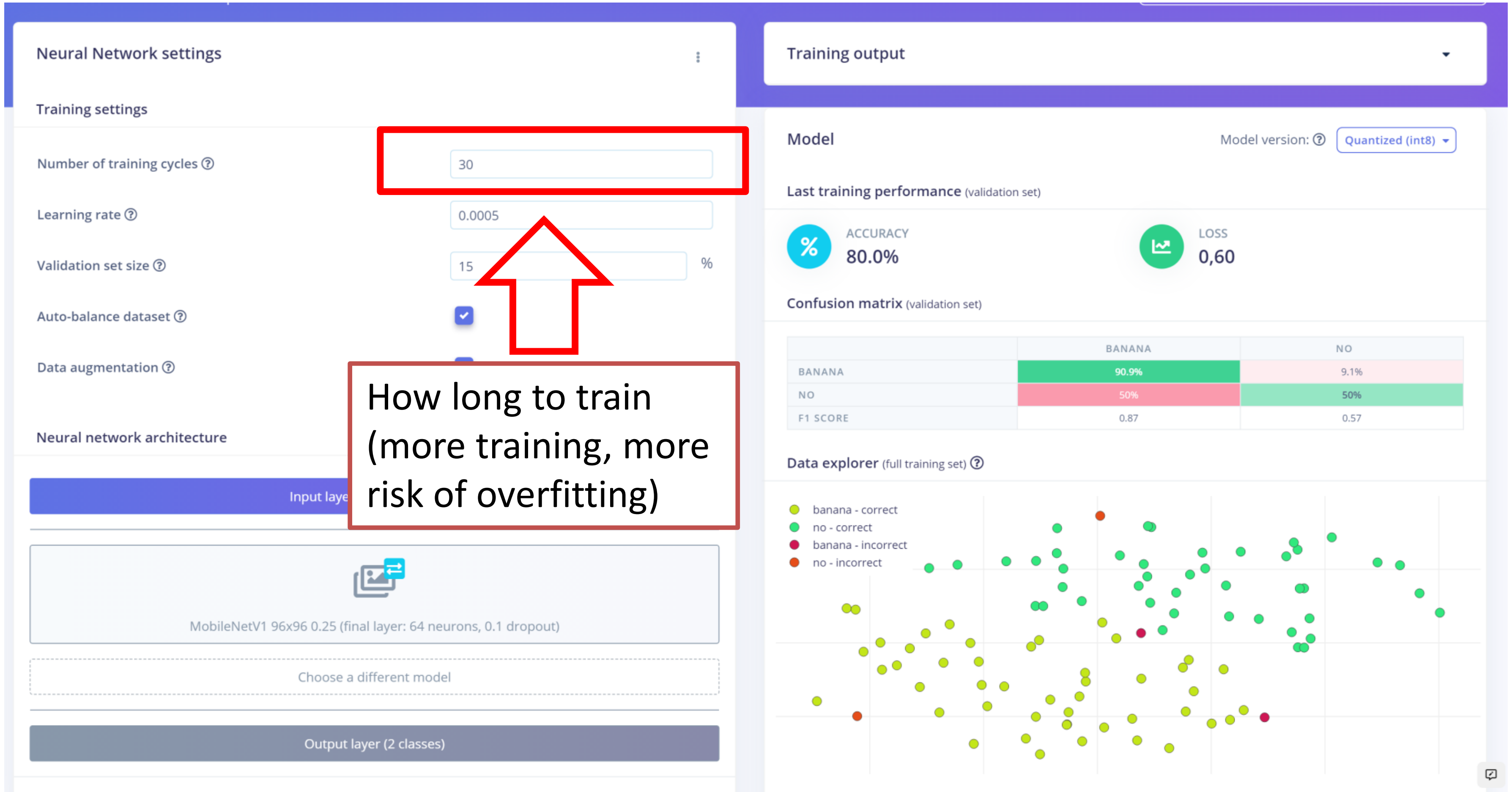


Impulse design

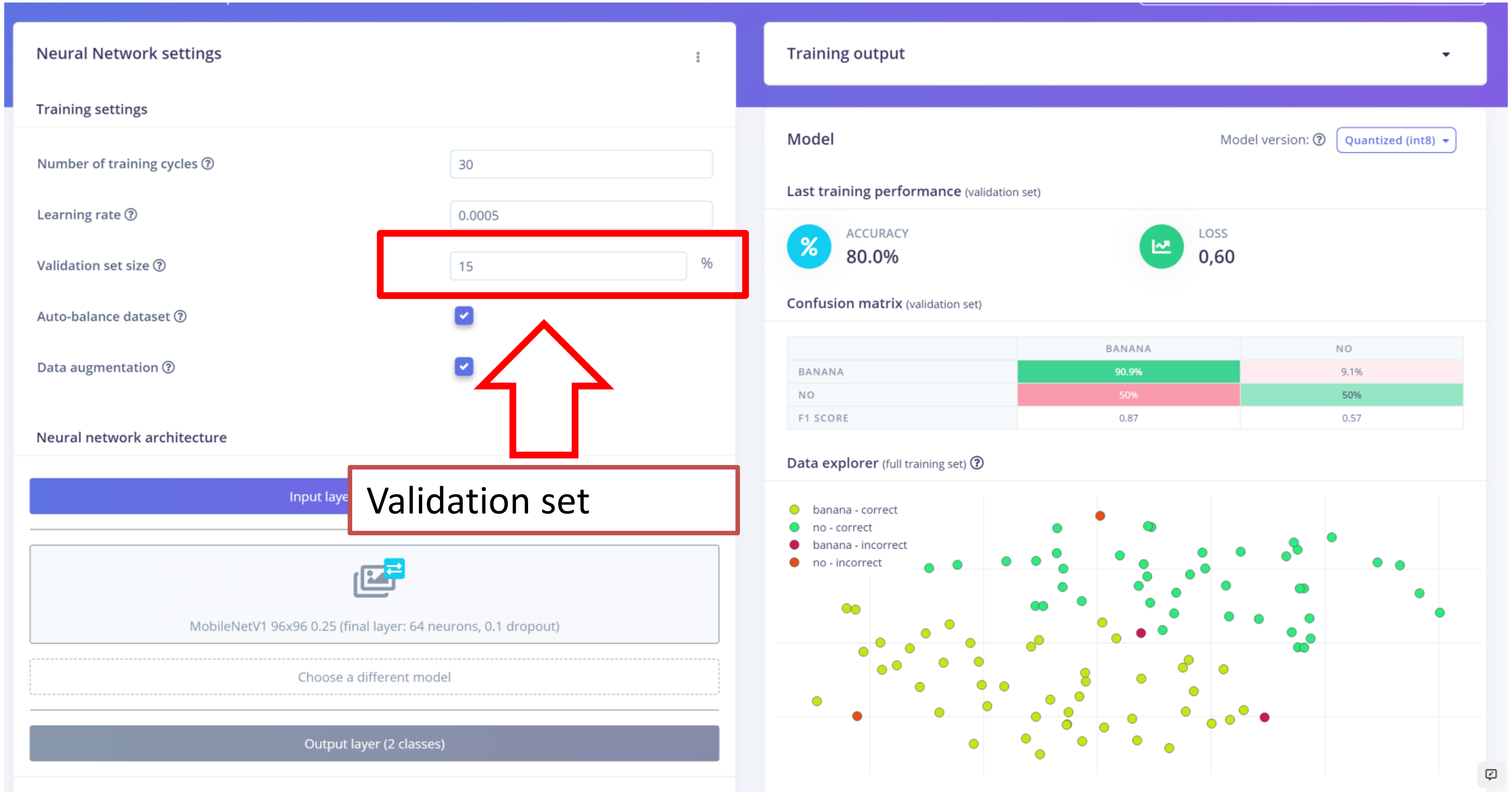


Output: 2 probability values (one for each class)

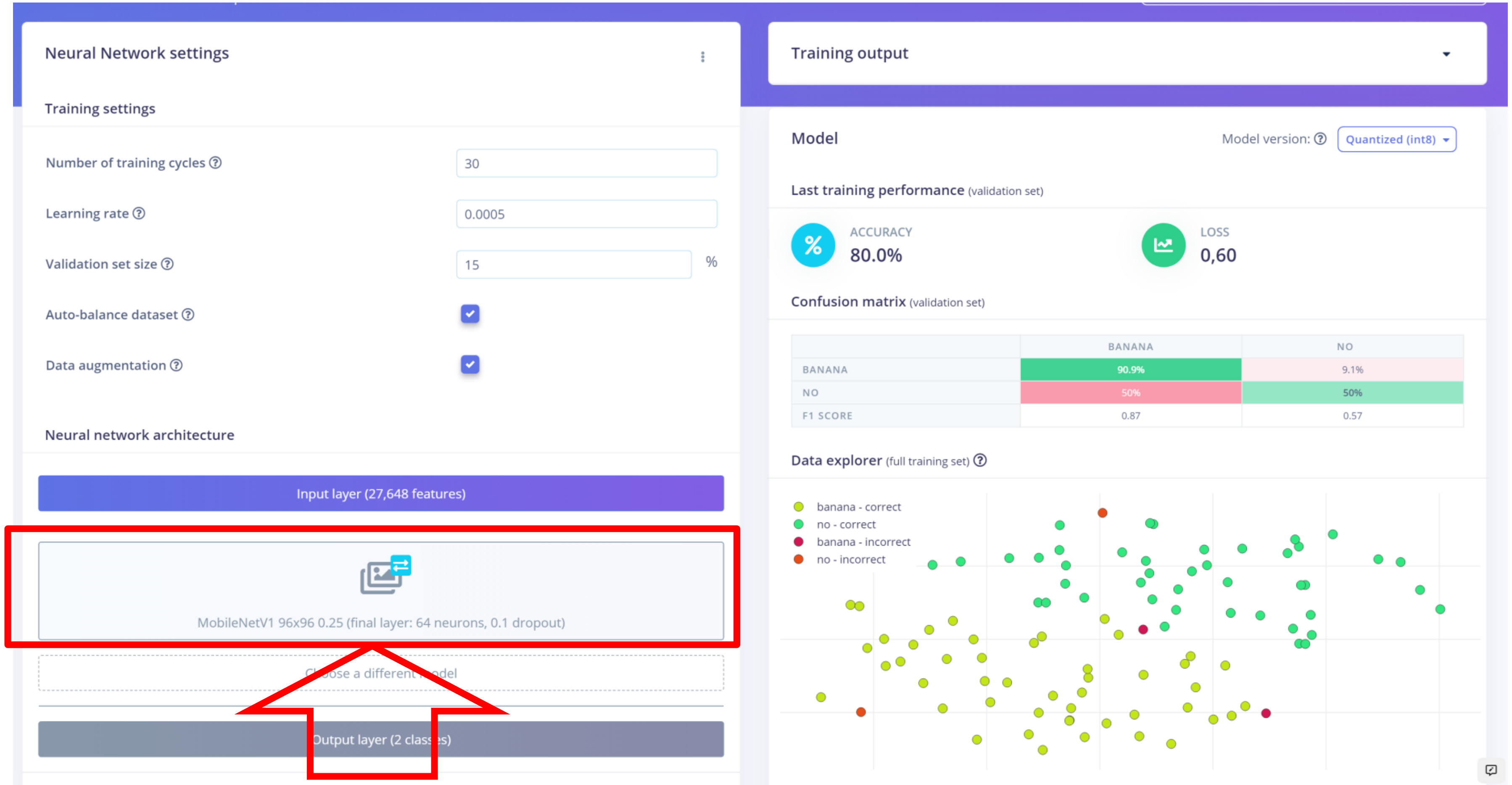
Neural network algorithm



Neural network algorithm

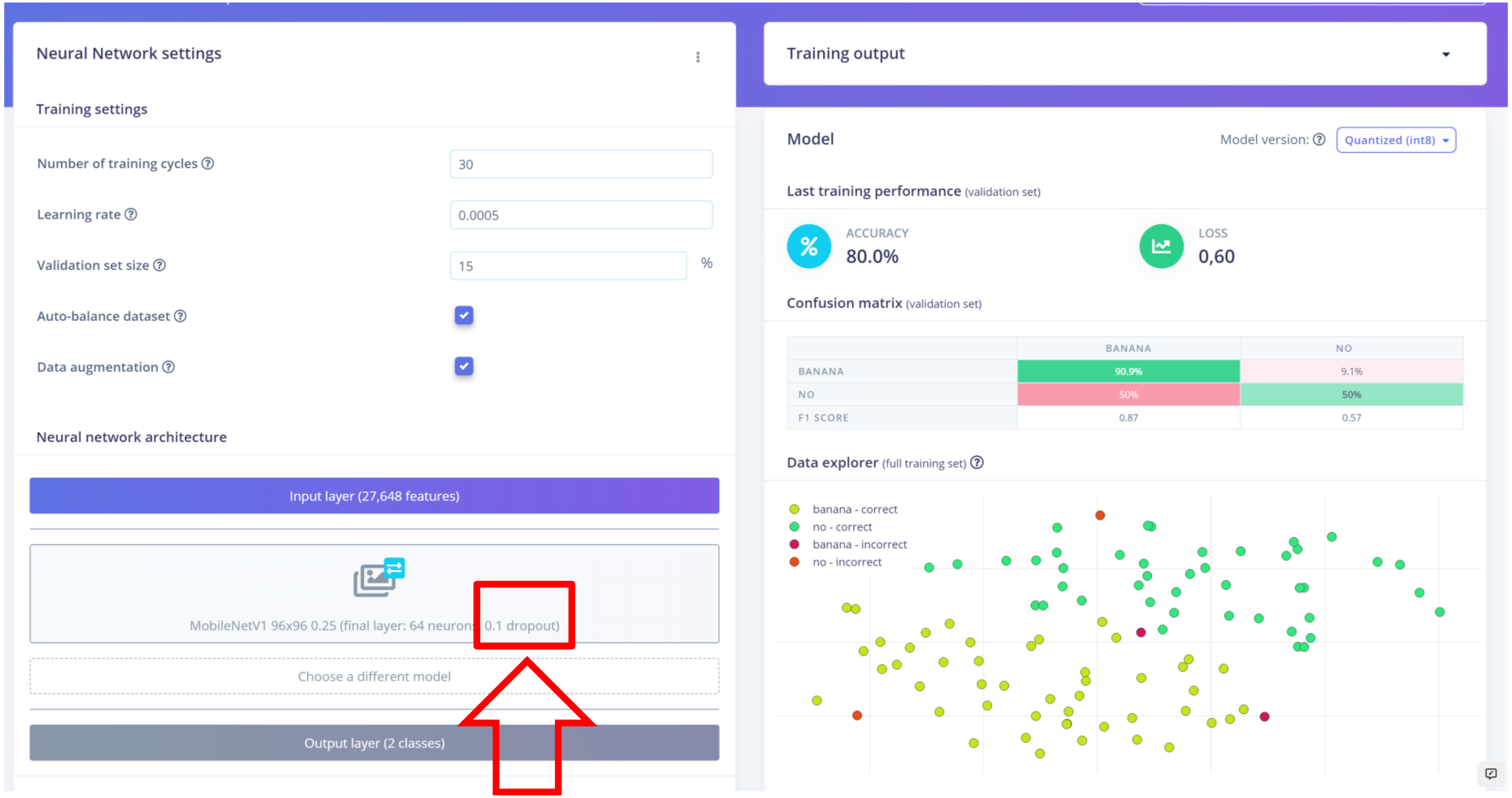


Neural network algorithm



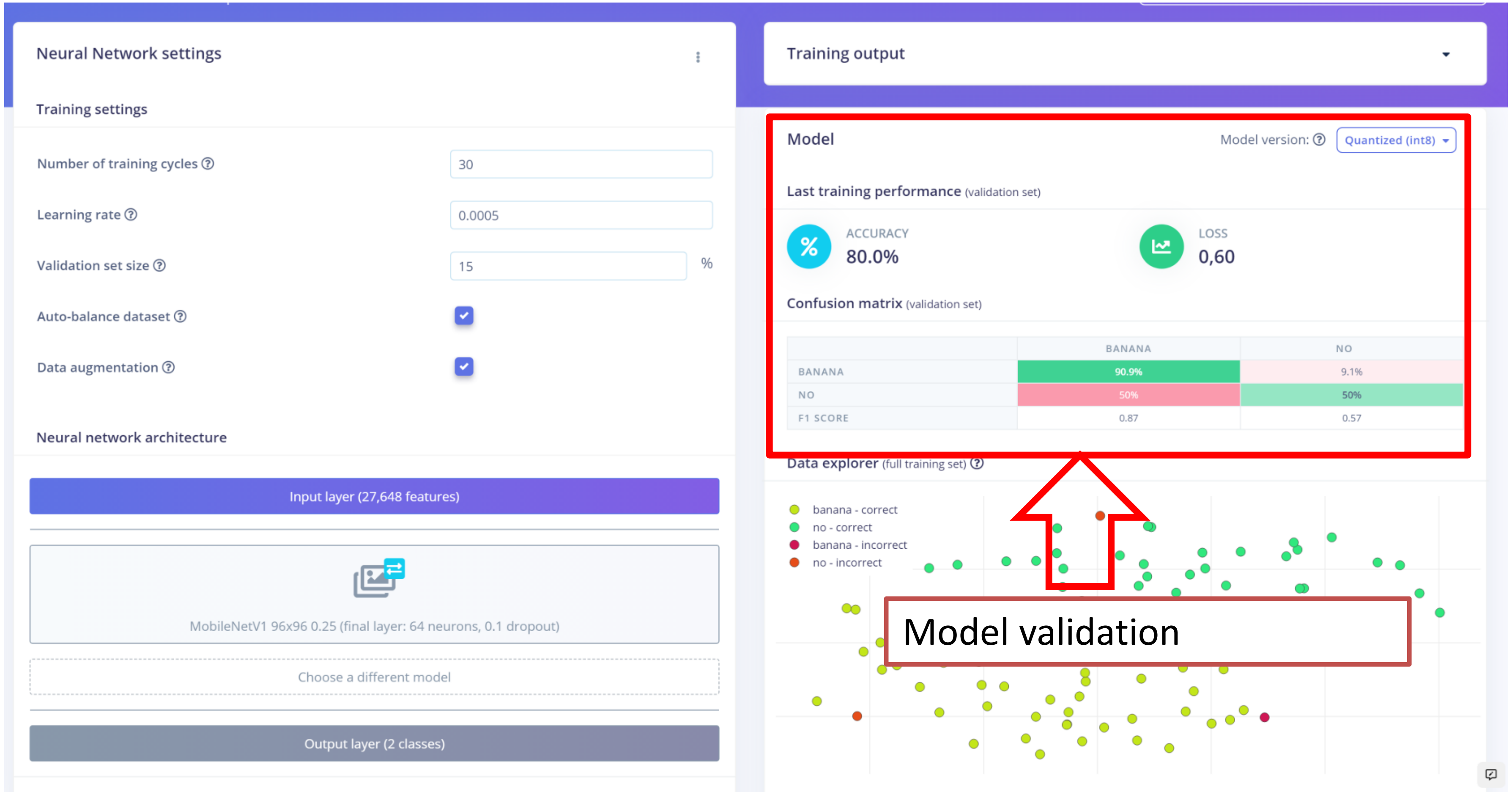
Model used (larger model works better, but may not fit) -> use «MobilenetV1 96x96 0.1»

Neural network algorithm

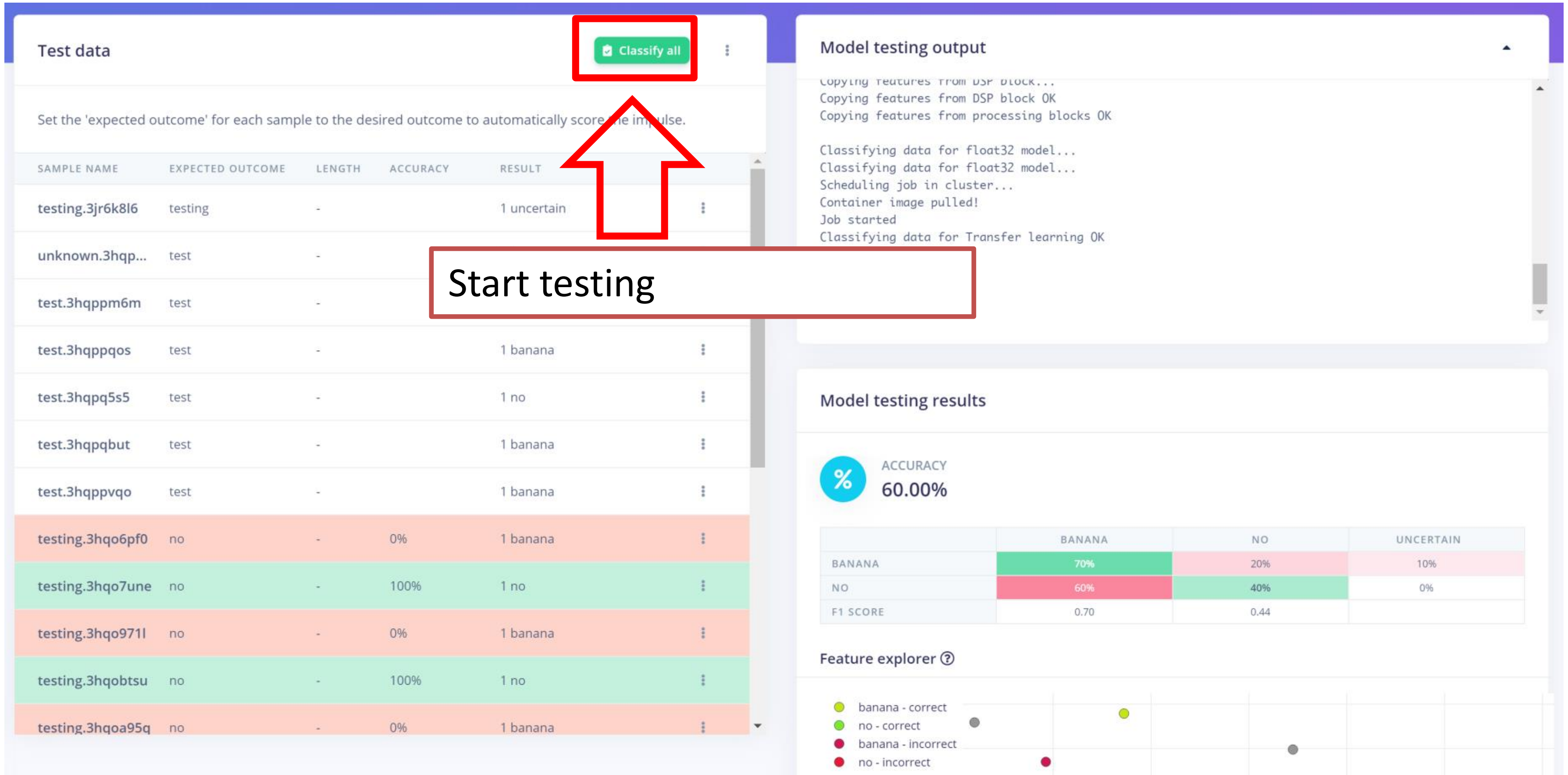


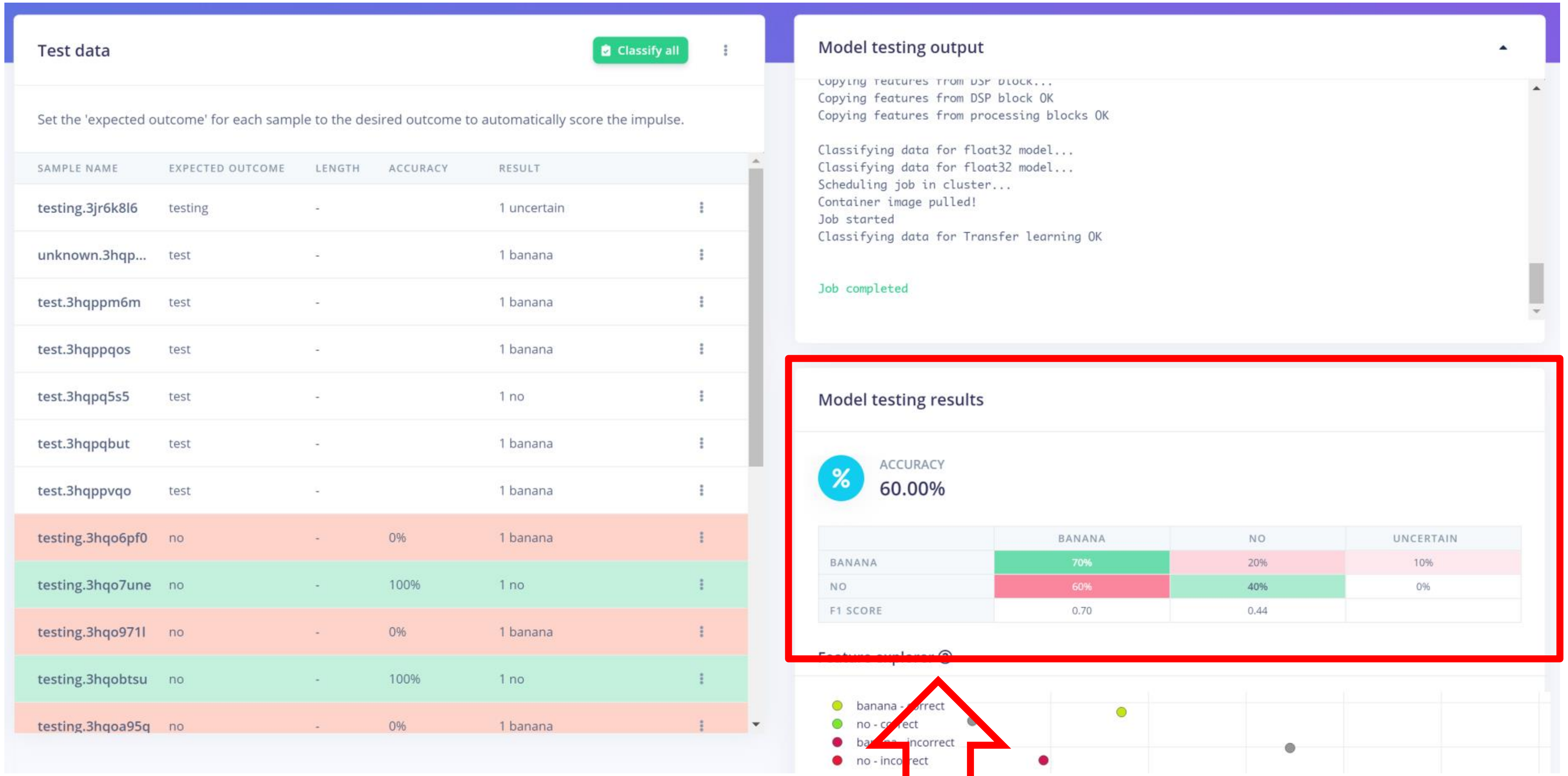
Dropout may help avoid overfitting

Neural network algorithm



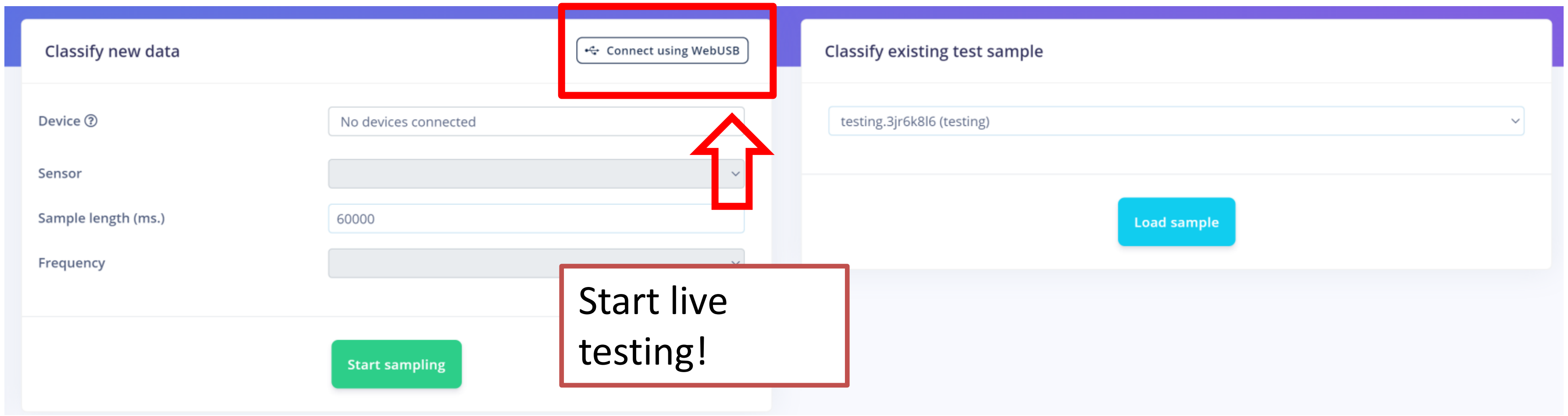
Testing the algorithm





model testing results

Live testing

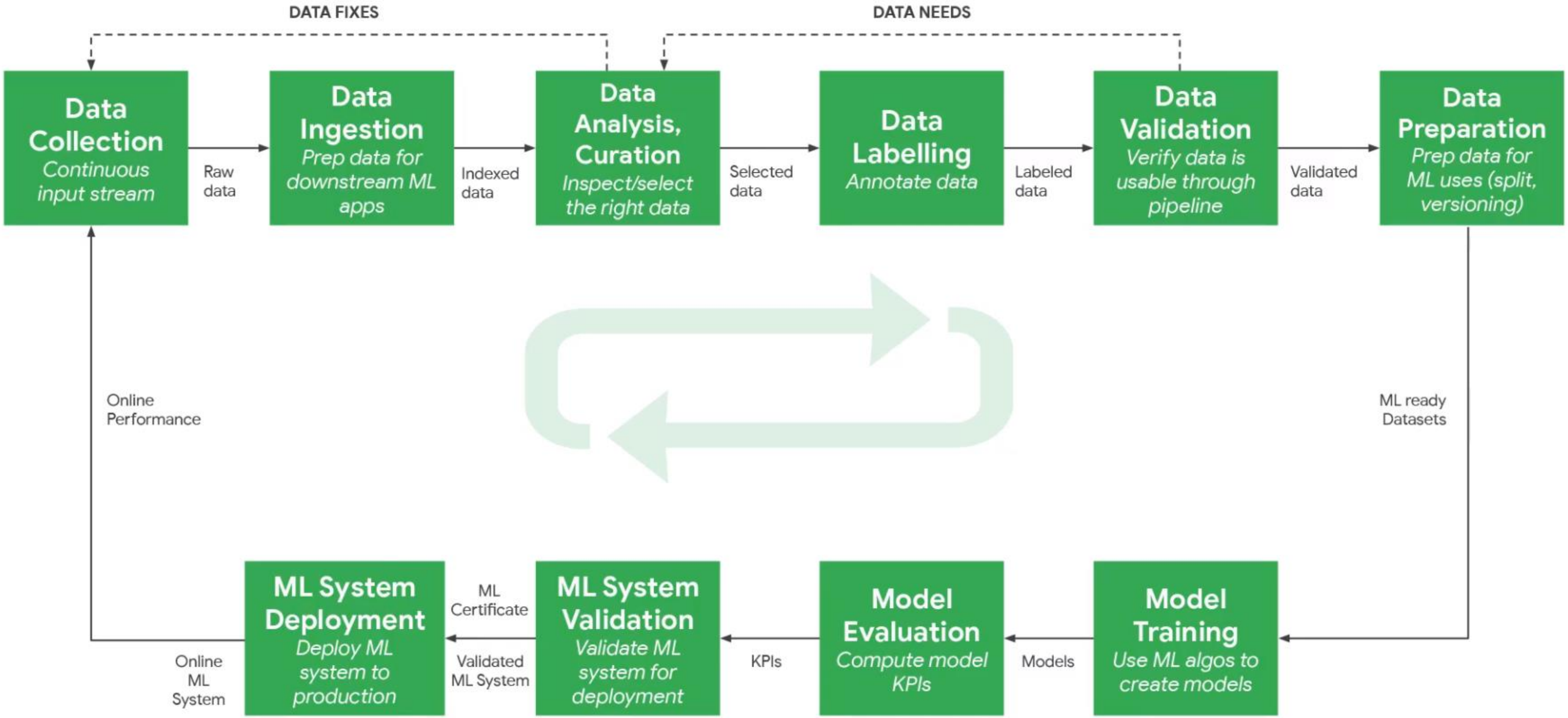


Start live testing!

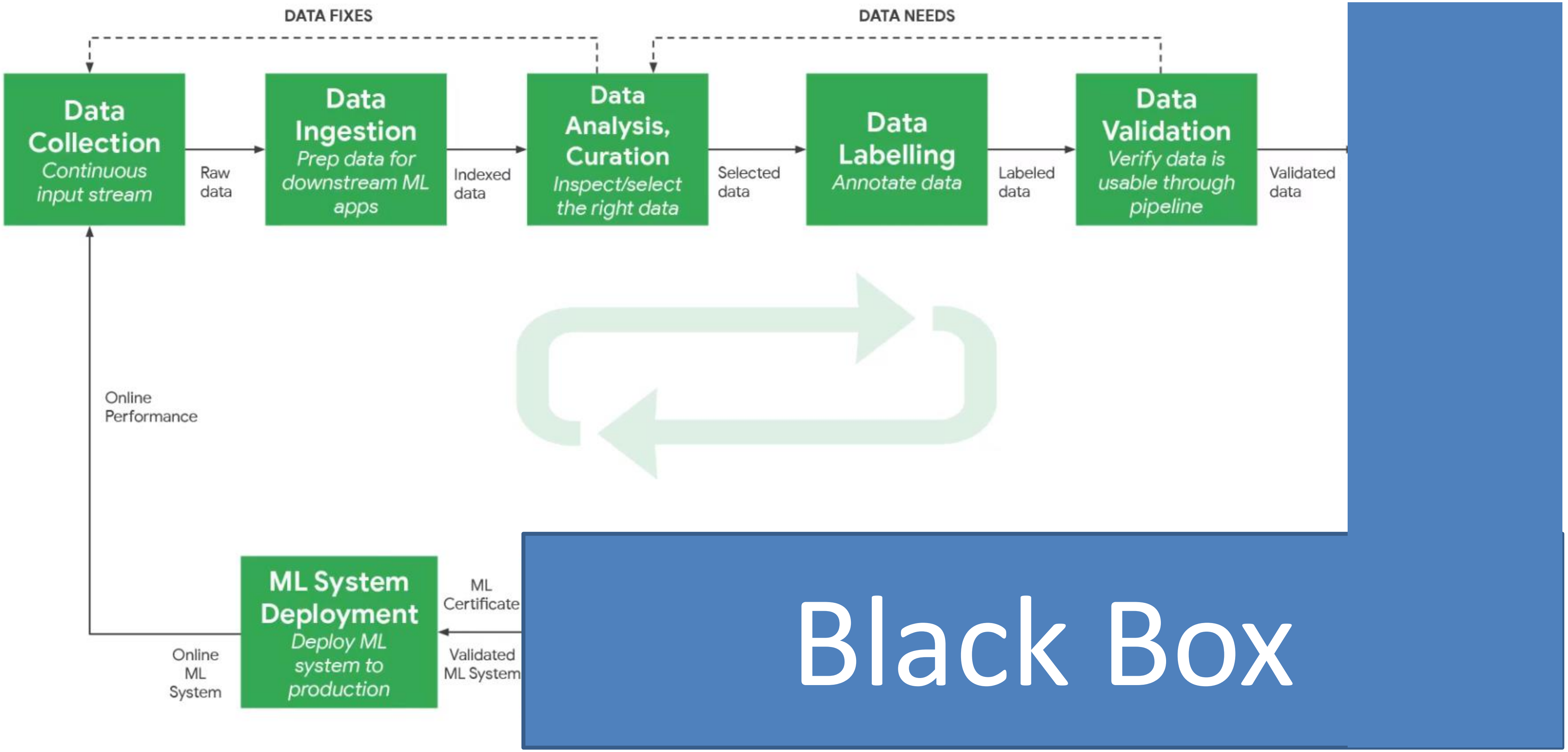
POLITECNICO
MILANO 1863

Hands on session

ML real life cycle



What we'll do today



Hands on session

- Collect the data for an object dectection algorithm:
  - Start from data of a single point of view, with the object fixed in one position
- Use the algorithm itself as a black box
  - Use standard configurations as seen in the slide
- Complicate the problem
  - Change the point of view, change the position of the object
  - Try occluding the object (do not capture it entirely, put a notebook covering half of it)
  - Deformate it!
  - Remember to give the algorithm an **equal number of object and other**!
- Start again from the beginning

Appendix

Credits and reference

- "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers", Daniel Situnayake, Pete Warden, O'Reilly Media, Inc.
- Online course:
  - https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning
- A lot more material on TinyML:
  - http://tinyml.seas.harvard.edu/
- https://petewarden.com/2018/05/28/why-you-need-to-improve-your-training-data-and-how-to-do-it/
- Cats pictures and challenges in OD/C: https://github.com/maxis42/Convolutional-Neural-Networks-Stanford-CS231n/blob/master/Lectures/lecture2.pdf and lectures of the Artificial Neural Network and Deep Learning course by prof. Boracchi
- Special thanks to M. Azzolini, M. Aldrighetti, E. Martello, and F. Zanotelli, who let me use the pictures that they collected for their project at CFP G. Veronesi at Rovereto

https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-mcu-targets/arduino-nano-33-ble-sense

https://cdn.edgeimpulse.com/firmware/arduino-nano-33-ble-sense.zip