



POLITECNICO
MILANO 1863



POLITECNICO
MILANO 1863

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

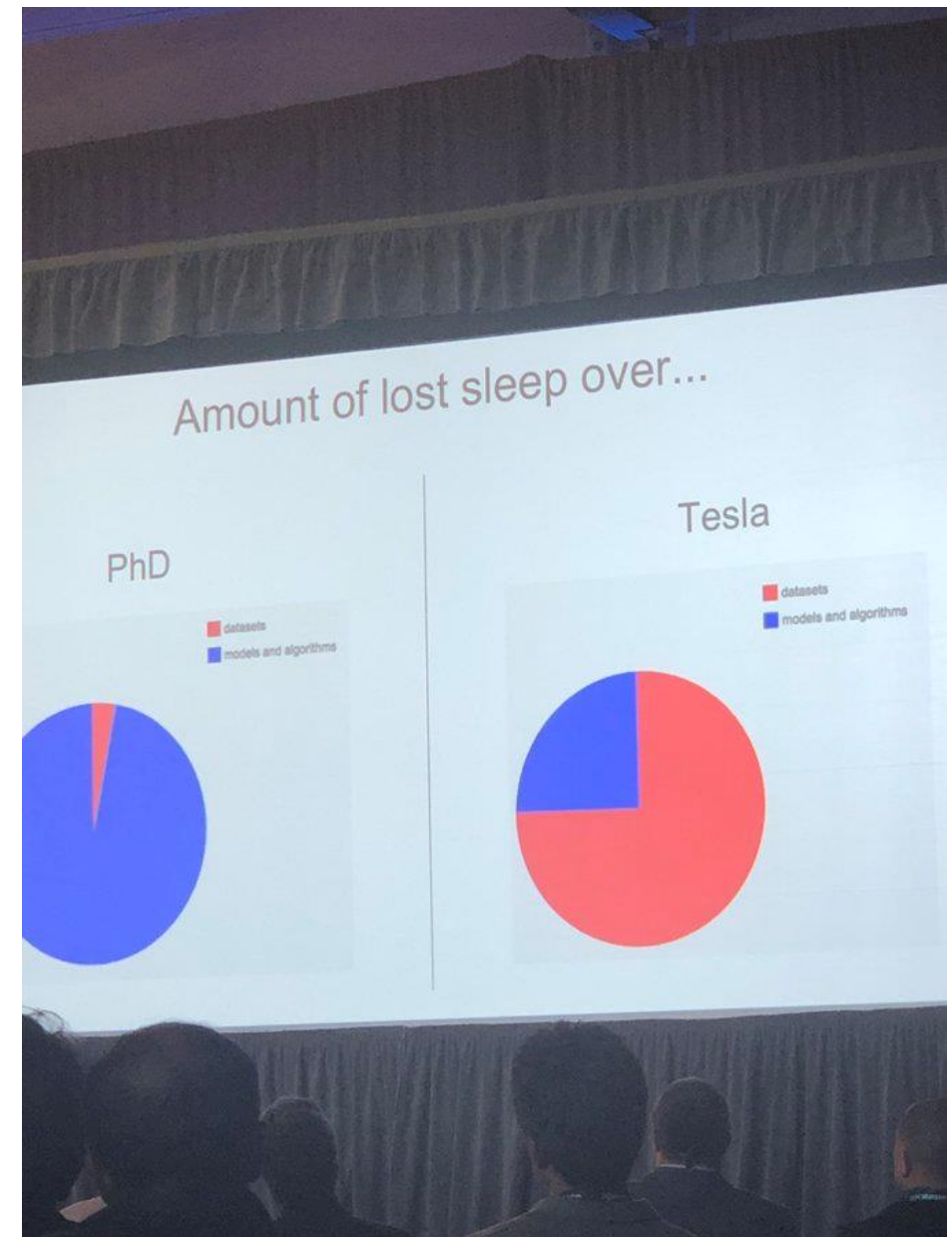
Collect a reasonably dimensioned dataset to perform object detection:

- 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



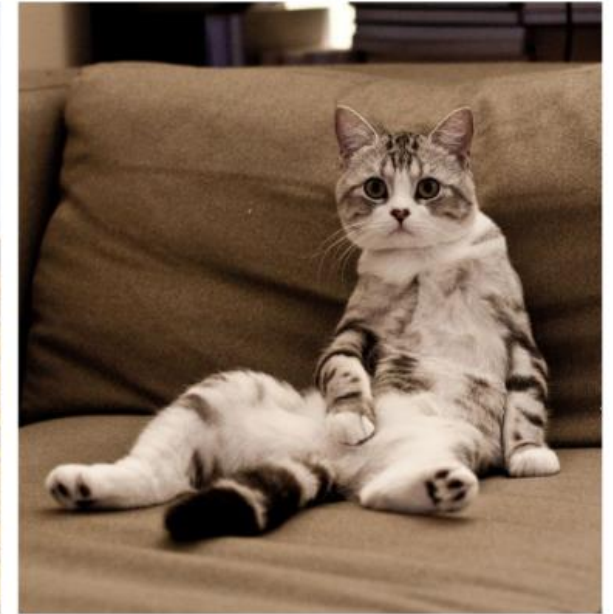
This image by Umberto Salvagnin
is licensed under [CC-BY 2.0](#)



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



This image is CC0 1.0 public domain



This image is CC0 1.0 public domain

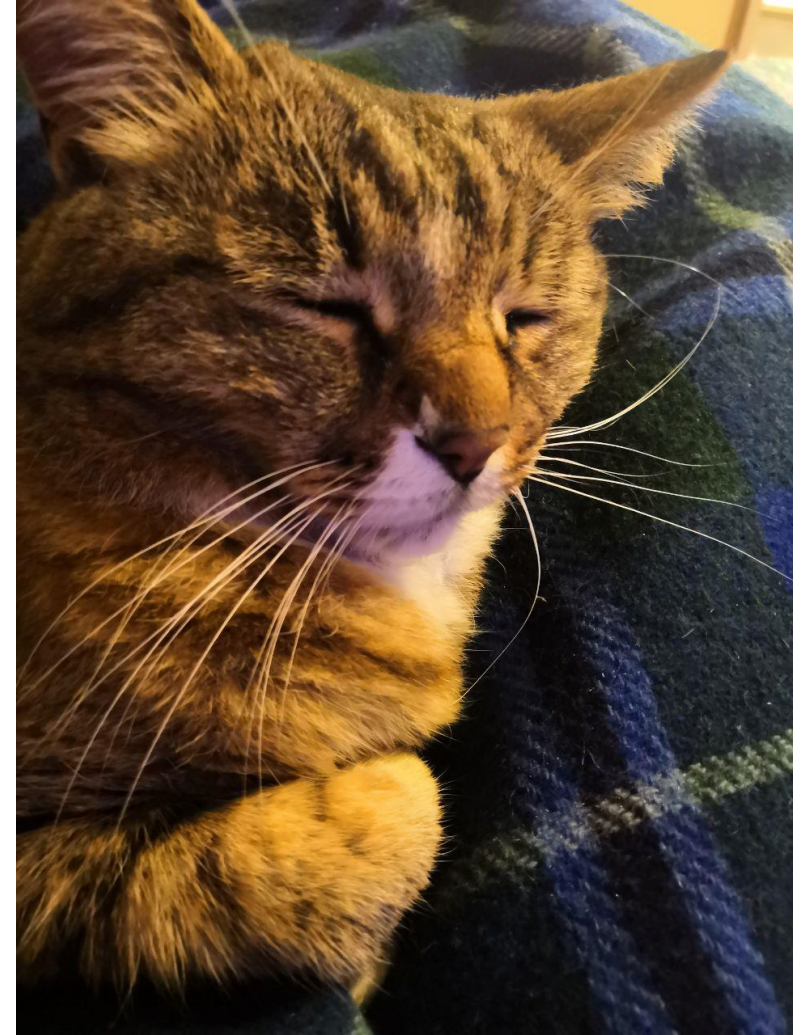


This image by jonsson is licensed under CC-BY 2.0

Challenges: view point change



Camera movement



Challenges: inter-class variability



This image is CC0 1.0 public domain

Challenges: scale variation



Dataset construction for image detection/classification

- A deep learning algorithm can address all these 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 unknown conditions, should be trained on data with:
 - Different illuminations, view points
 - Occluded and deformed targets
 - Examples of different objects of the same class

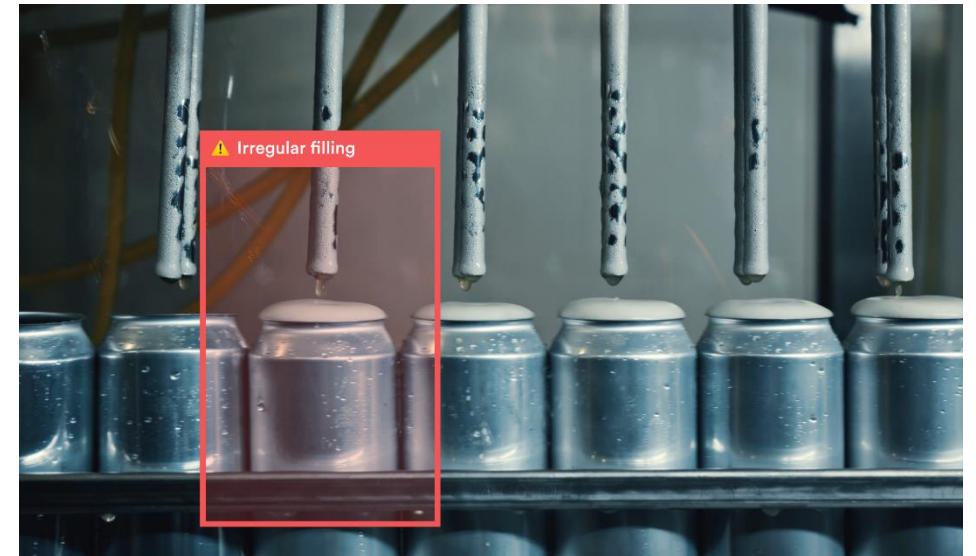
Image classification in the context of tinyML

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 (especially 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%

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

Connect device

Connect using WebUSB


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

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% ⓘ



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



DATA COLLECTED
95 items

TRAIN / TEST SPLIT
86% / 14%

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

Device ?
F9:52:E7:F7:EF:BA

Label
banana

Sensor
Camera (128x96)

Camera feed

Start sampling

The picture you'll take

DATA COLLECTED
95 items

TRAIN / TEST SPLIT
86% / 14%

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

Device

F9:52:E7:F7:EF:BA

Label

banana

Sensor

Camera (128x96)

Camera feed

Start sampling

The label to assign to the picture

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

The image shows a 'Collect data' dialog box with a close button (X) in the top right corner. The dialog contains the following sections:

- Connect a fully supported development board**
Get started with real hardware from a wide range of silicon vendors - fully supported by Edge Impulse.
[Browse dev boards](#)
- Use your mobile phone**
Use your mobile phone to capture movement, audio or images, and even run your trained model locally. No app required.
[Show QR code](#) (This button is highlighted with a red rectangle)
- Use your computer**
Capture audio or images from your webcam or microphone, or from an external audio device.
[Collect data](#)
- Data from any device with the data forwarder**
Capture data from any device or development board over a serial connection, in 10 lines of code.
[Show docs](#)
- Upload data**
Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, MP3, CSV, JSON, MP4, and more.
[Go to the uploader](#)

In the background, a sidebar on the left shows options like 'Mas', 'This is you', 'IMAGE', 'About', and 'Creat'. The main background area on the right includes a 'Your project is private.' message with a 'Make this project public' button, and a section titled 'n this model' with a QR code and the text 'an QR code or launch in browser'.



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?

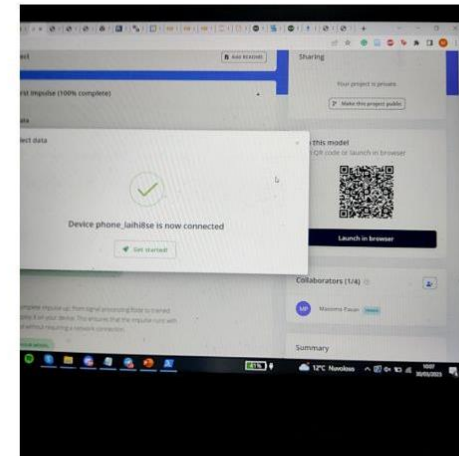
Collecting motion?



Data collection

Label: test

Category: Training



Capture

Images captured with current settings: 0

Switch to classification mode

Impulse design

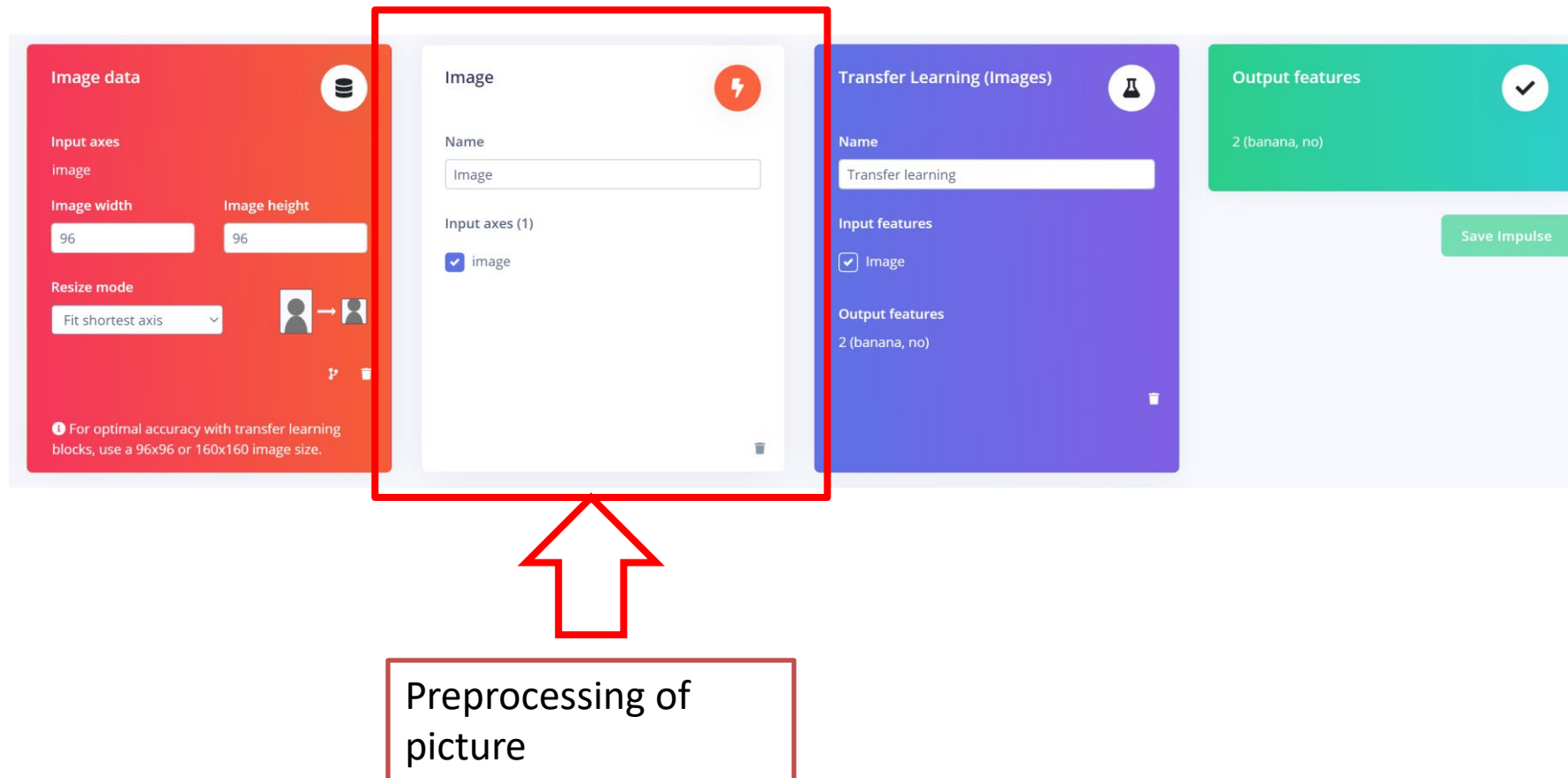
The screenshot displays the Impulse design interface with four blocks in a sequence:

- Image data** (Red block):
 - Input axes: image
 - Image width: 96
 - Image height: 96
 - Resize mode: Fit shortest axis
 - Icon: Two people with an arrow between them
 - Tip: For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.
- Image** (White block):
 - Name: Image
 - Input axes (1): ☒ image
- Transfer Learning (Images)** (Purple block):
 - Name: Transfer learning
 - Input features: ☒ Image
 - Output features: 2 (banana, no)
- Output features** (Teal block):
 - 2 (banana, no)
 - Save Impulse button

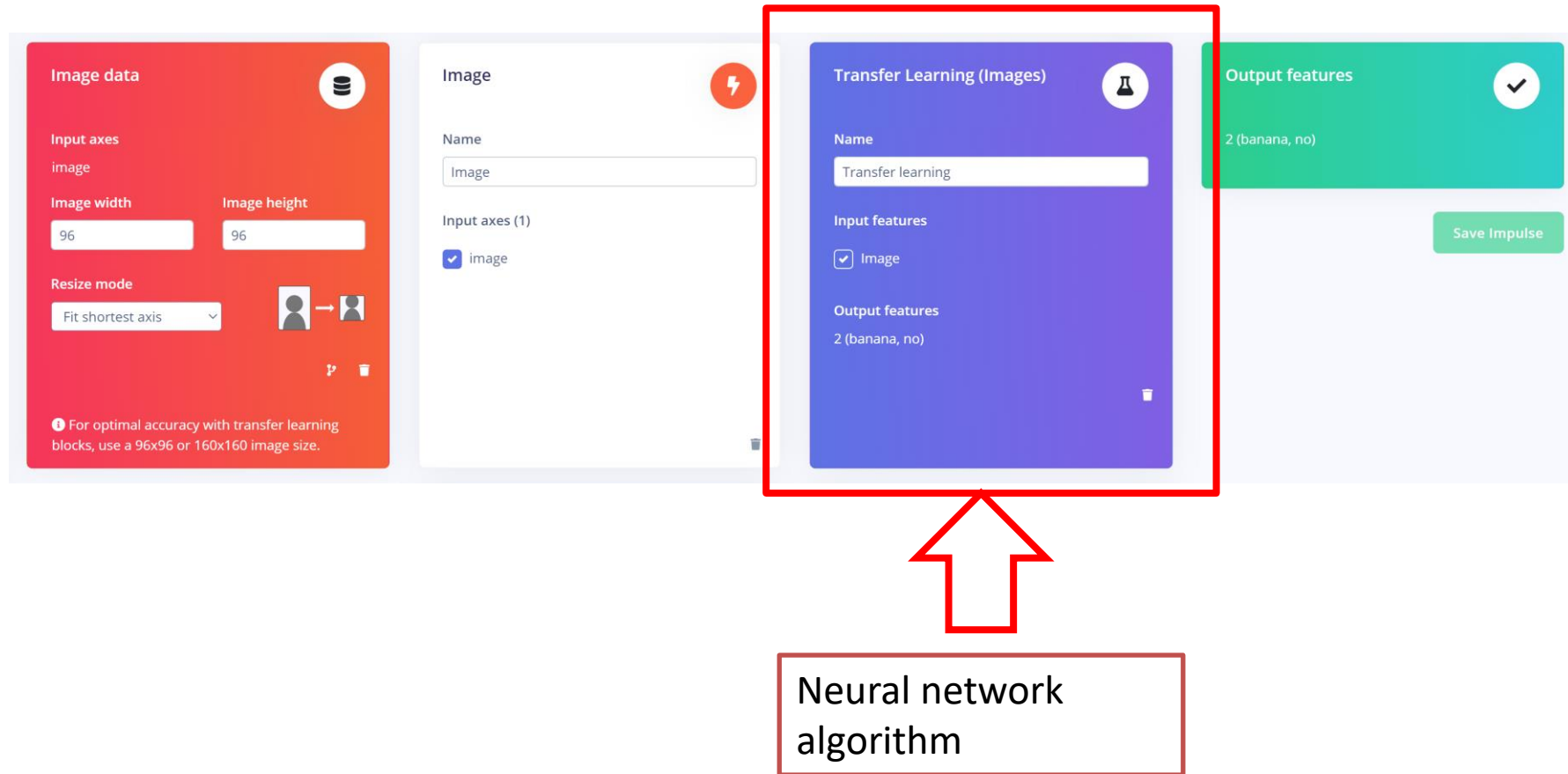
A red box highlights the 'Image data' block, and a red arrow points from the text box below to it.

Resize of the picture taken

Impulse design



Impulse design



Impulse design

The screenshot displays the Impulse AI design interface, which is a drag-and-drop environment for building machine learning models. The interface is divided into several panels:

- Image data (Red panel):** Contains input axes (image), image width (96), image height (96), and a resize mode dropdown (Fit shortest axis). A note at the bottom states: "For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size."
- Image (White panel):** Contains a name field (Image) and an input axes list (image).
- Transfer Learning (Images) (Blue panel):** Contains a name field (Transfer learning), an input features list (Image), and an output features list (2 (banana, no)).
- Output features (Green panel):** Contains the output features list (2 (banana, no)) and a Save Impulse button.

A red box highlights the Output features panel, and a red arrow points to it from a text box below.

Output: 2 probability values (one for each class)

Neural network algorithm

Neural Network settings

Training settings

Number of training cycles ②

30

Learning rate ②

0.0005

Validation set size ②

15 %

Auto-balance dataset ②

☒

Data augmentation ②

☐

Neural network architecture

Input layer

MobileNetV1 96x96 0.25 (final layer: 64 neurons, 0.1 dropout)

Choose a different model

Output layer (2 classes)

Training output

Model

Model version: ②

Quantized (int8)

Last training performance (validation set)

ACCURACY

80.0%

LOSS

0,60

Confusion matrix (validation set)

	BANANA	NO
BANANA	90.9%	9.1%
NO	50%	50%
F1 SCORE	0.87	0.57

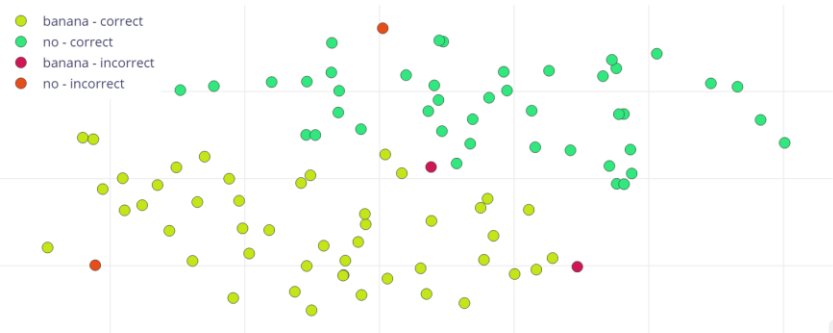
Data explorer (full training set) ②

● banana - correct

● no - correct

● banana - incorrect

● no - incorrect



How long to train
(more training, more
risk of overfitting)

Neural network algorithm

Neural Network settings

Training settings

Number of training cycles ②

Learning rate ②

Validation set size ②

%


Auto-balance dataset ② ☒

Data augmentation ② ☒

Neural network architecture

Input layer

Validation set


MobileNetV1 96x96 0.25 (final layer: 64 neurons, 0.1 dropout)

Choose a different model


Output layer (2 classes)


Training output

Model

Model version: ② Quantized (int8) ▾

Last training performance (validation set)

 **ACCURACY**
80.0%

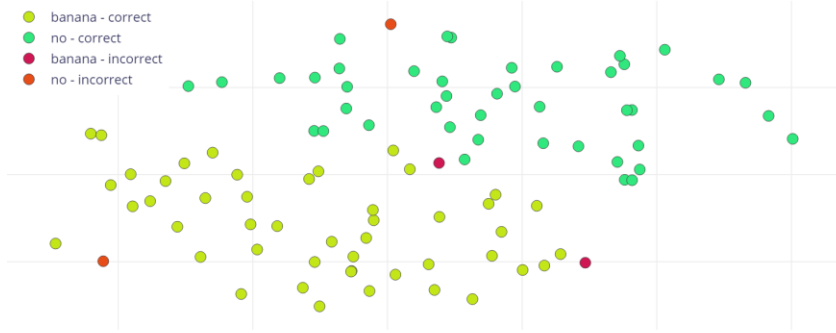
 **LOSS**
0,60

Confusion matrix (validation set)

	BANANA	NO
BANANA	90.9%	9.1%
NO	50%	50%
F1 SCORE	0.87	0.57

Data explorer (full training set) ②

● banana - correct
● no - correct
● banana - incorrect
● no - incorrect



Neural network algorithm

Neural Network settings

Training settings

Number of training cycles ②

Learning rate ②


Validation set size ② %

Auto-balance dataset ② ☒

Data augmentation ② ☒

Neural network architecture

Input layer (27,648 features)


MobileNetV1 96x96 0.25 (final layer: 64 neurons, 0.1 dropout)

[Choose a different model](#)


Output layer (2 classes)


Training output

Model

Model version: ② Quantized (int8) ▾

Last training performance (validation set)

 **ACCURACY**
80.0%

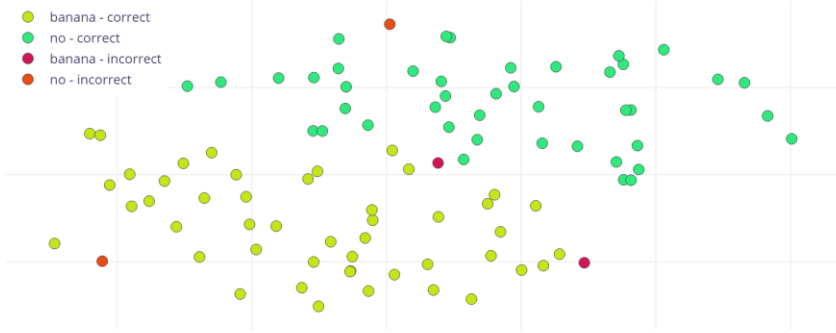
 **LOSS**
0,60

Confusion matrix (validation set)

	BANANA	NO
BANANA	90.9%	9.1%
NO	50%	50%
F1 SCORE	0.87	0.57

Data explorer (full training set) ②

● banana - correct
● no - correct
● banana - incorrect
● no - incorrect



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

Neural network algorithm

Neural Network settings

Training settings

Number of training cycles ②

Learning rate ②

Validation set size ② %

Auto-balance dataset ② ☒

Data augmentation ② ☒

Neural network architecture

Input layer (27,648 features)

MobileNetV1 96x96 0.25 (final layer: 64 neurons, 0.1 dropout)

Choose a different model

Output layer (2 classes)

Training output

Model

Model version: ② Quantized (int8)

Last training performance (validation set)

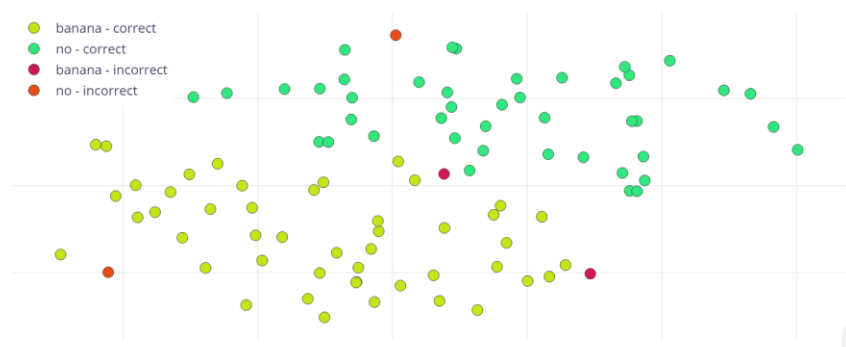
ACCURACY 80.0% **LOSS** 0,60

Confusion matrix (validation set)

	BANANA	NO
BANANA	90.9%	9.1%
NO	50%	50%
F1 SCORE	0.87	0.57

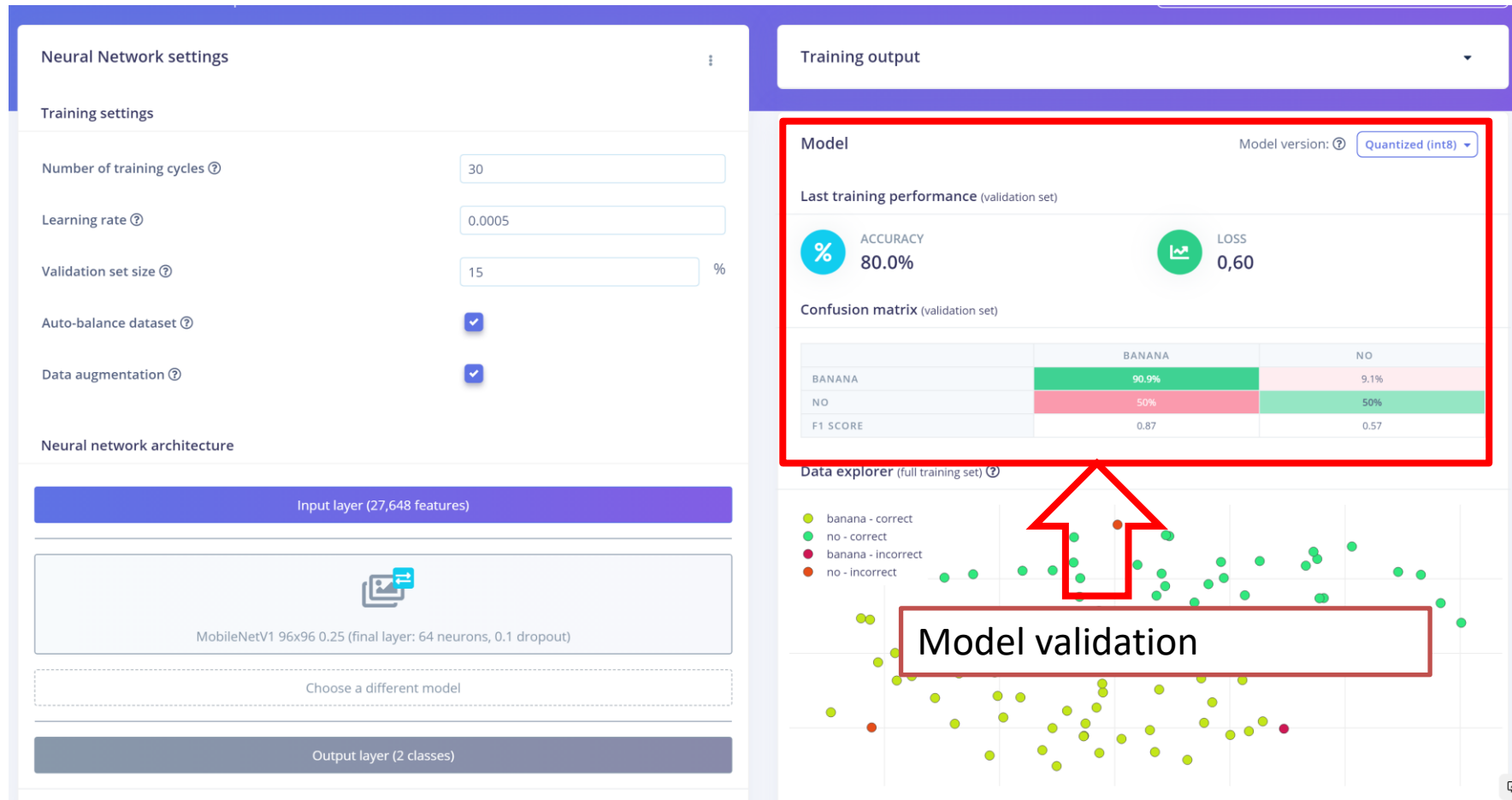
Data explorer (full training set) ②

● banana - correct
● no - correct
● banana - incorrect
● no - incorrect



Dropout may help avoid overfitting

Neural network algorithm



Testing the algorithm

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
testing.3jr6k8l6	testing	-		1 uncertain
unknown.3hqp...	test	-		
test.3hqppm6m	test	-		
test.3hqppqos	test	-		1 banana
test.3hqppq5s5	test	-		1 no
test.3hqppqbut	test	-		1 banana
test.3hqppvqo	test	-		1 banana
testing.3hqo6pf0	no	-	0%	1 banana
testing.3hqo7une	no	-	100%	1 no
testing.3hqo971l	no	-	0%	1 banana
testing.3hqobtsu	no	-	100%	1 no
testing.3hqoa95q	no	-	0%	1 banana

Model testing output

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

Classifying data for float32 model...
Classifying data for float32 model...
Scheduling job in cluster...
Container image pulled!
Job started
Classifying data for Transfer learning OK

Start testing

Model testing results

%

ACCURACY

60.00%

	BANANA	NO	UNCERTAIN
BANANA	70%	20%	10%
NO	60%	40%	0%
F1 SCORE	0.70	0.44	

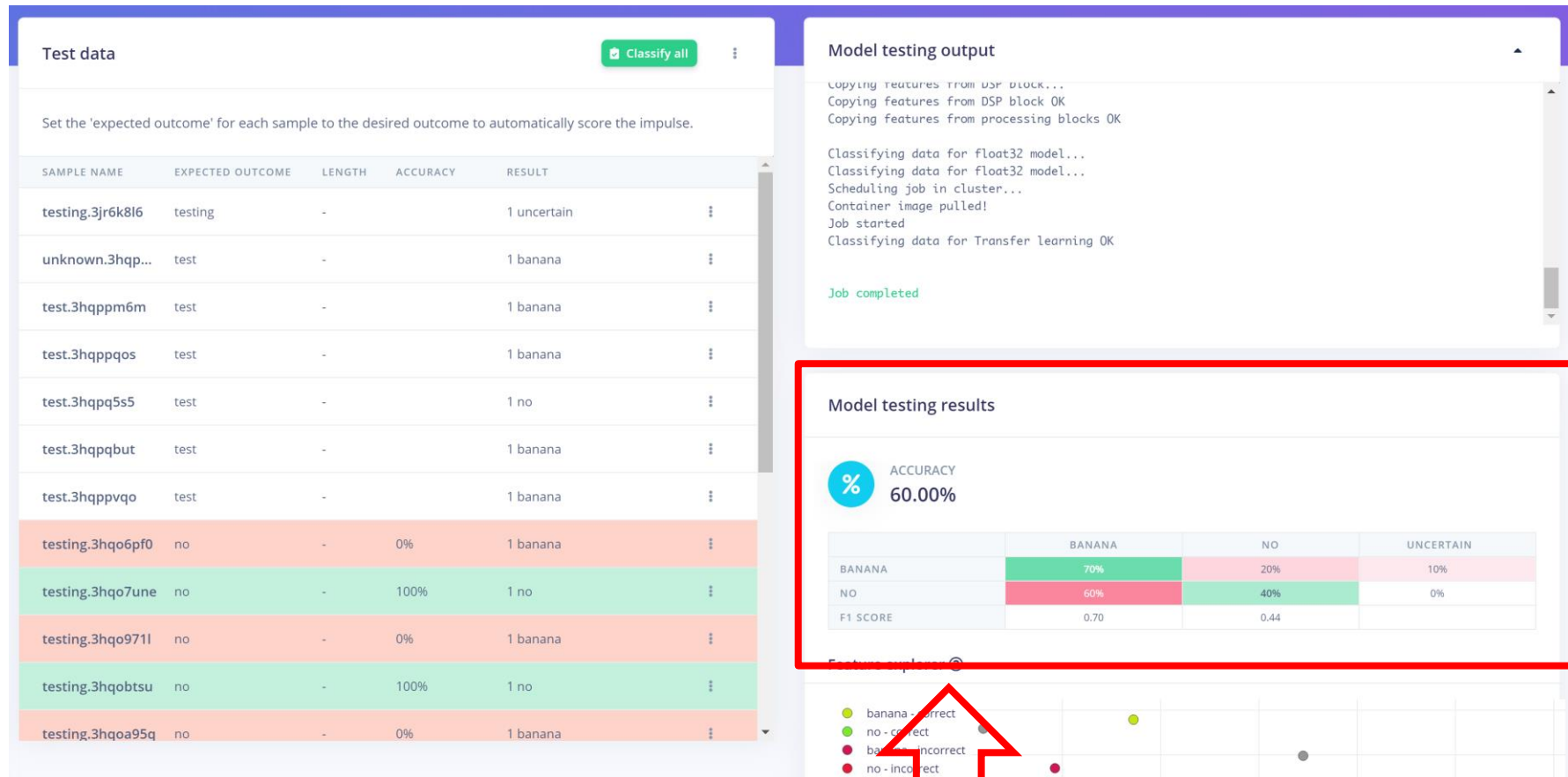
Feature explorer ?

● banana - correct

● no - correct

● banana - incorrect

● no - incorrect



model testing results

Live testing

The screenshot shows a web interface with two main panels. The left panel, titled 'Classify new data', contains a 'Connect using WebUSB' button (highlighted with a red rectangle), a 'Device' dropdown (showing 'No devices connected'), a 'Sensor' dropdown, a 'Sample length (ms.)' input (set to 60000), and a 'Frequency' dropdown. A green 'Start sampling' button is at the bottom. The right panel, titled 'Classify existing test sample', has a dropdown menu (showing 'testing_3jr6k8l6 (testing)') and a blue 'Load sample' button. A red arrow points from a text box labeled 'Start live testing!' to the 'Connect using WebUSB' button.

Classify new data

Connect using WebUSB

Device ? No devices connected

Sensor

Sample length (ms.) 60000

Frequency

Start sampling

Classify existing test sample

testing_3jr6k8l6 (testing)

Load sample

Start live testing!



POLITECNICO
MILANO 1863



POLITECNICO
MILANO 1863

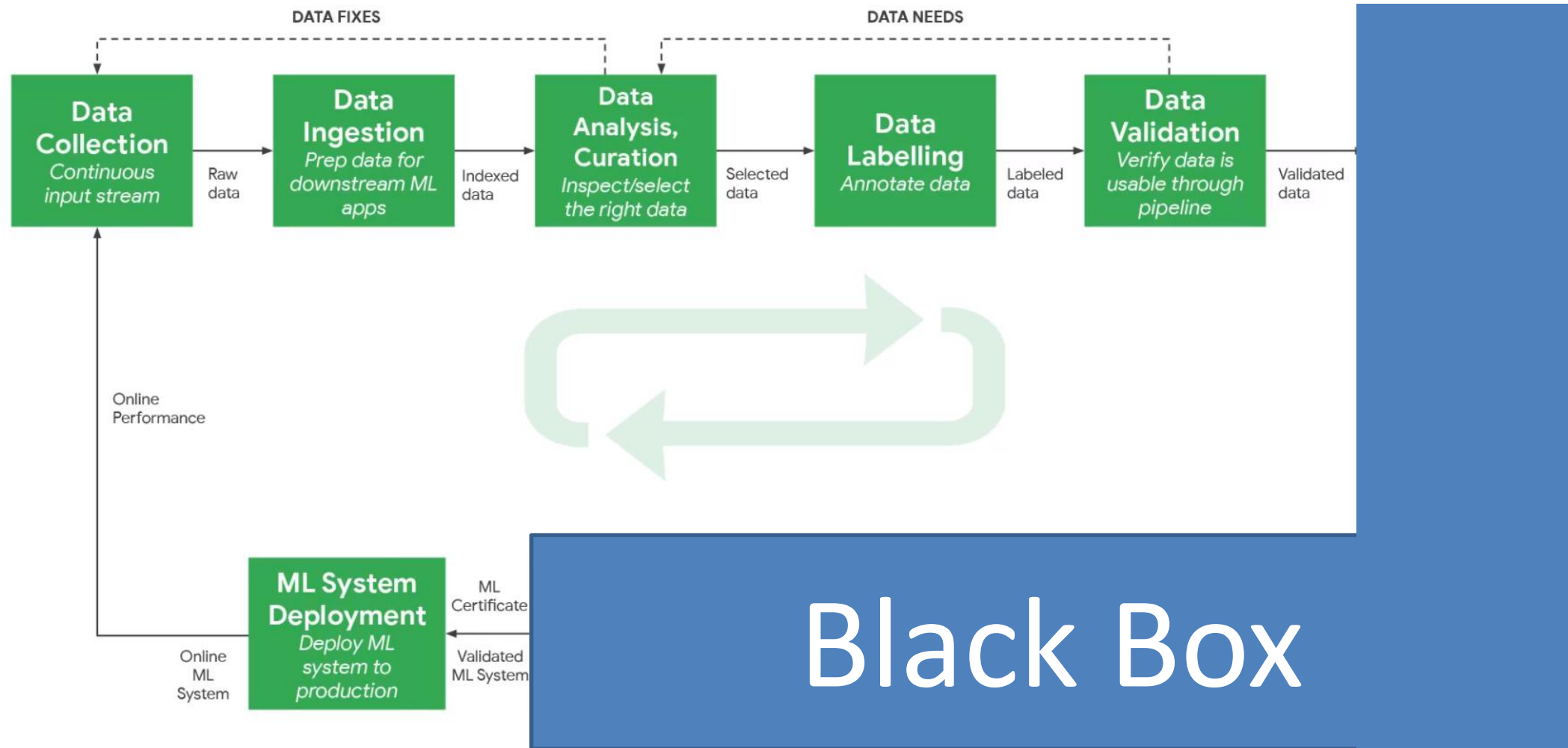
Hands on session

34



What we'll do today

35



Hands on session

- Collect the data for an object detection 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



POLITECNICO
MILANO 1863



POLITECNICO
MILANO 1863

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://tinymml.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>