

Introduction to Edge Impulse Studio CNN with Cifar-10

Marco Zennaro, PhD

TinyML4D Academic Network Co-Chair



All the credit goes to Prof. Marcelo Rovai:



WALC 2023
Applied AI

Introduction to Edge Impulse Studio CNN with Cifar-10

Prof. Marcelo J. Rovai
rovai@unifei.edu.br

UNIFEI - Federal University of Itajuba, Brazil
TinyML4D Academic Network Co-Chair



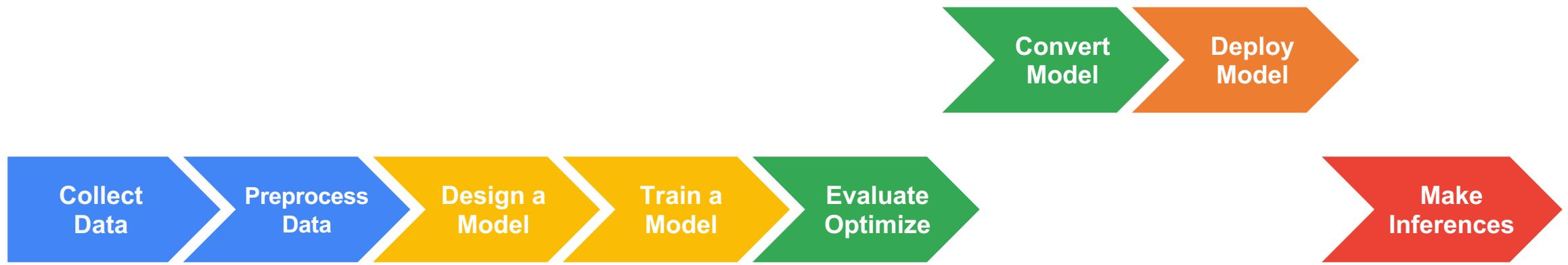
TINYML4D

Embedded Machine Learning (TinyML) Workflow Review

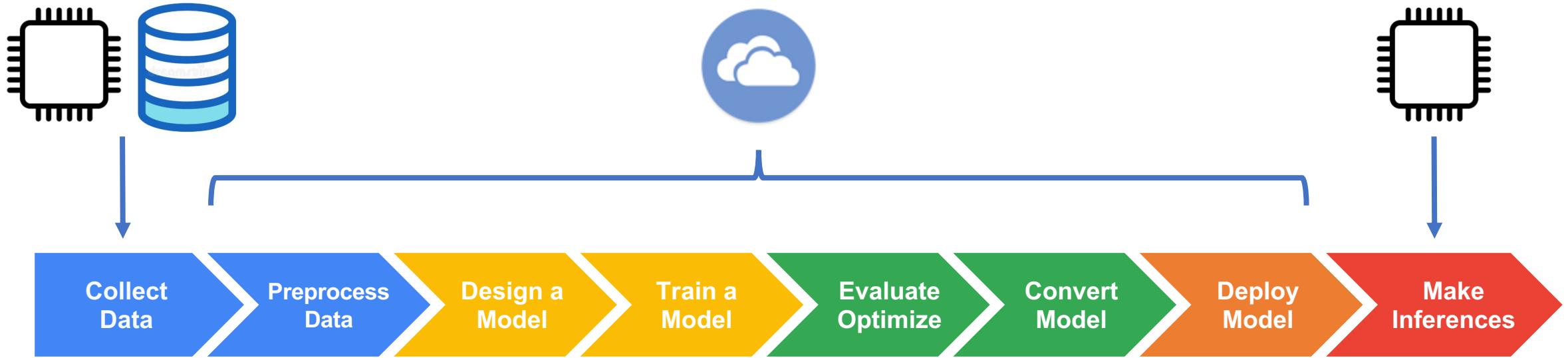
Machine Learning Workflow



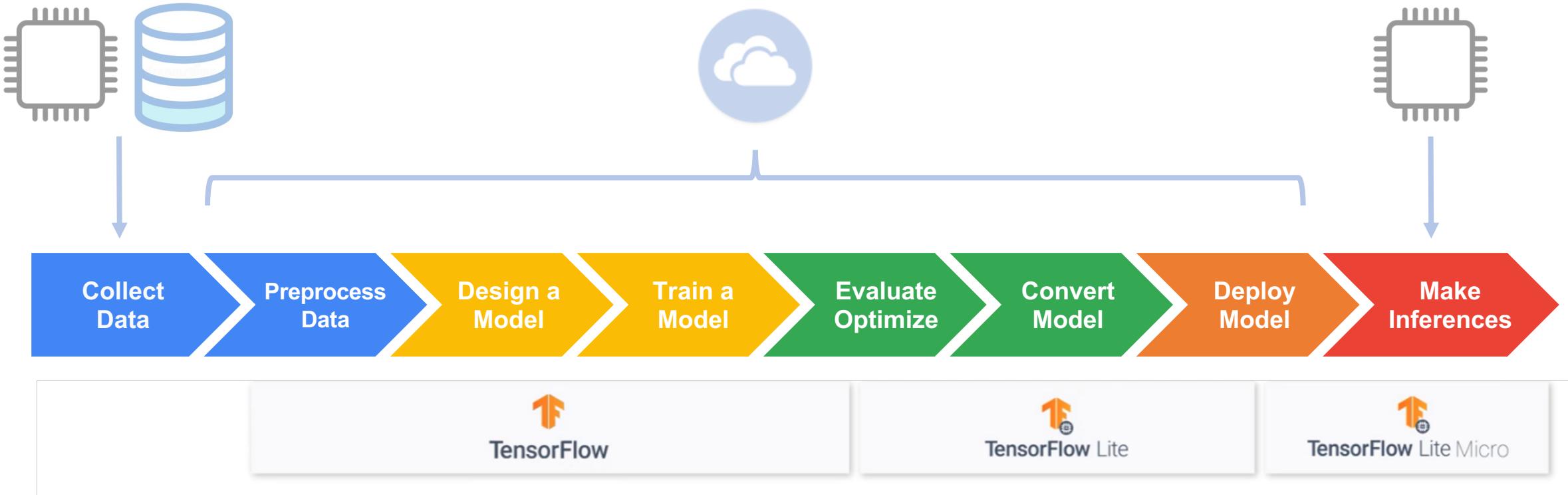
Tiny Machine Learning Workflow (“What”)



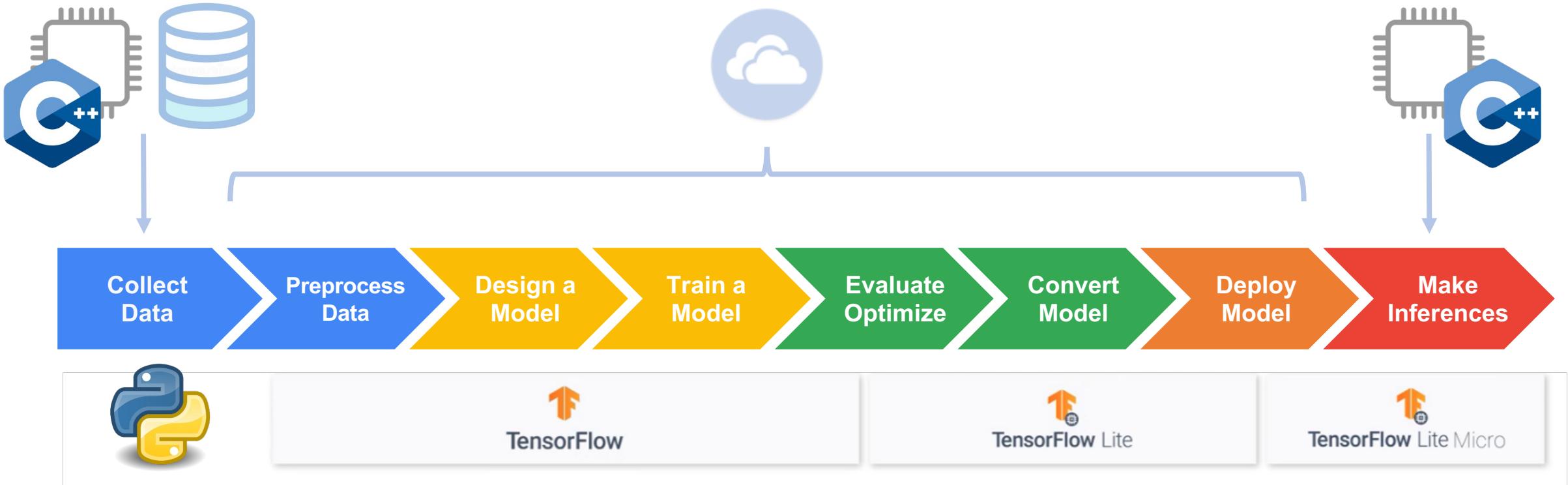
Tiny Machine Learning Workflow (“Where”)



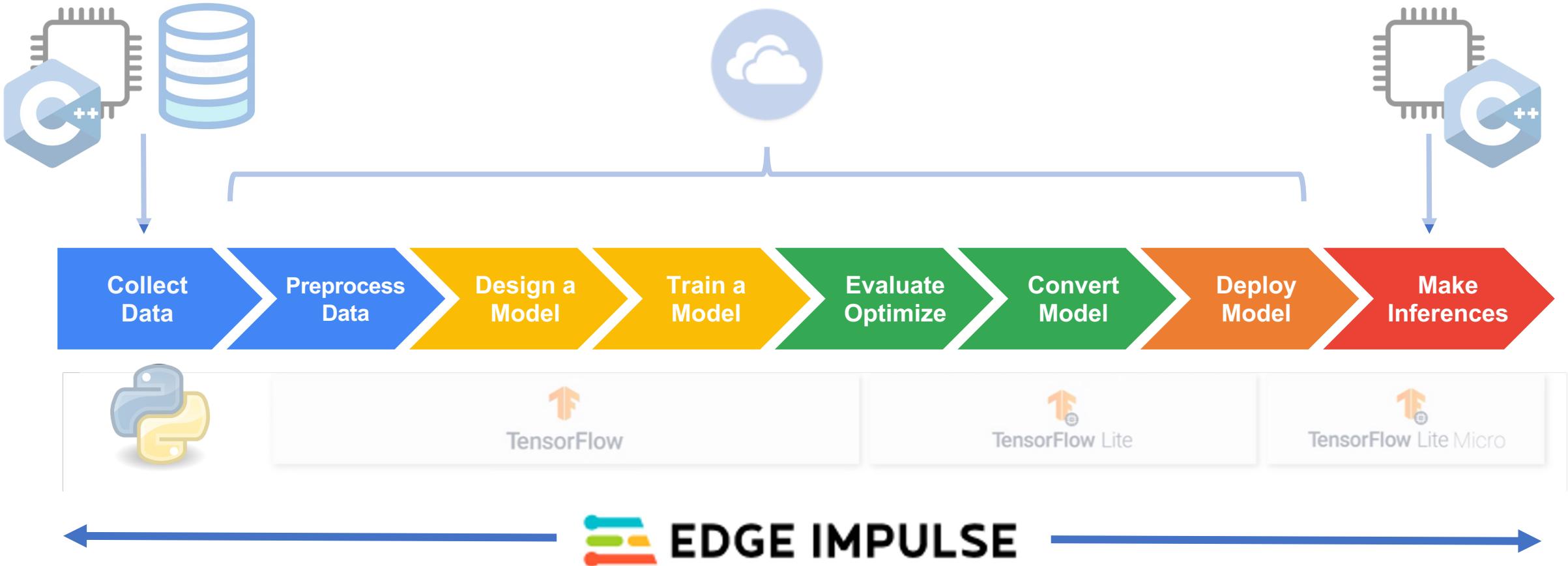
Machine Learning Workflow (“How”)

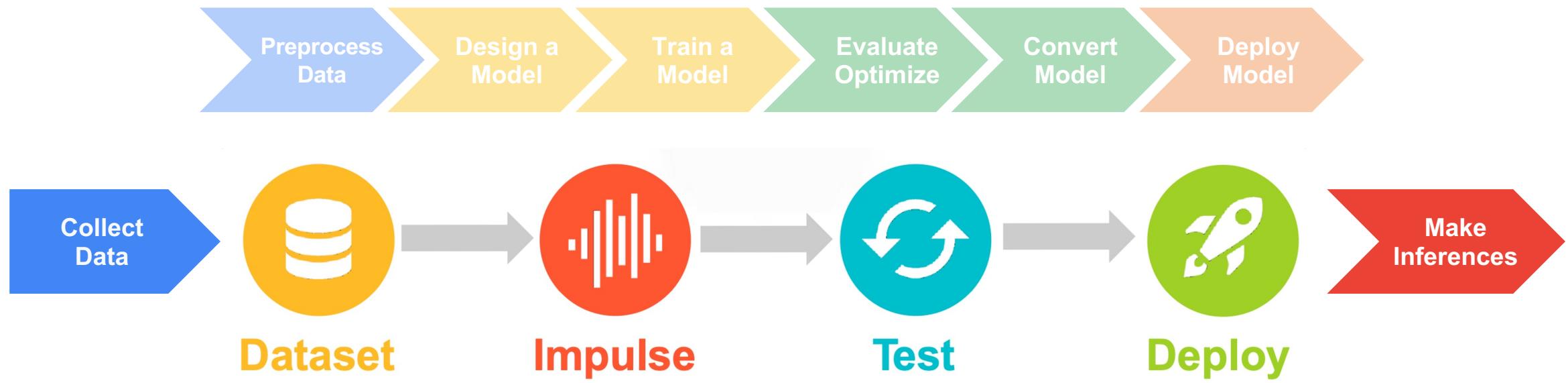


Machine Learning Workflow (“How”)

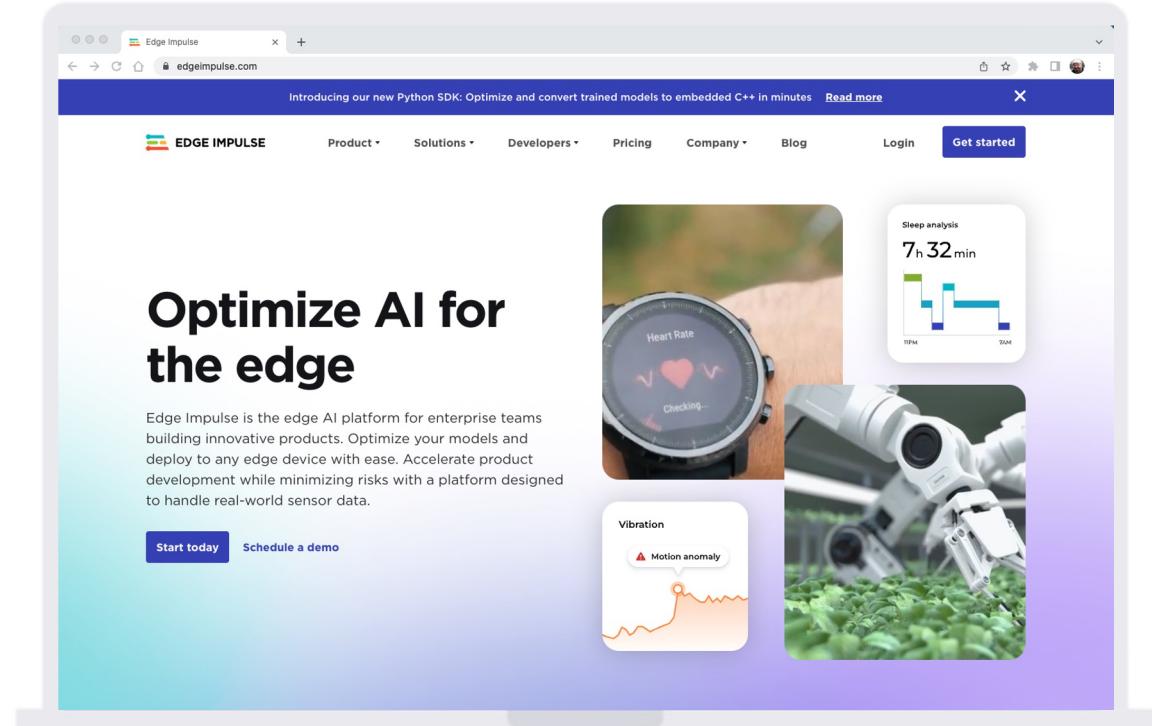
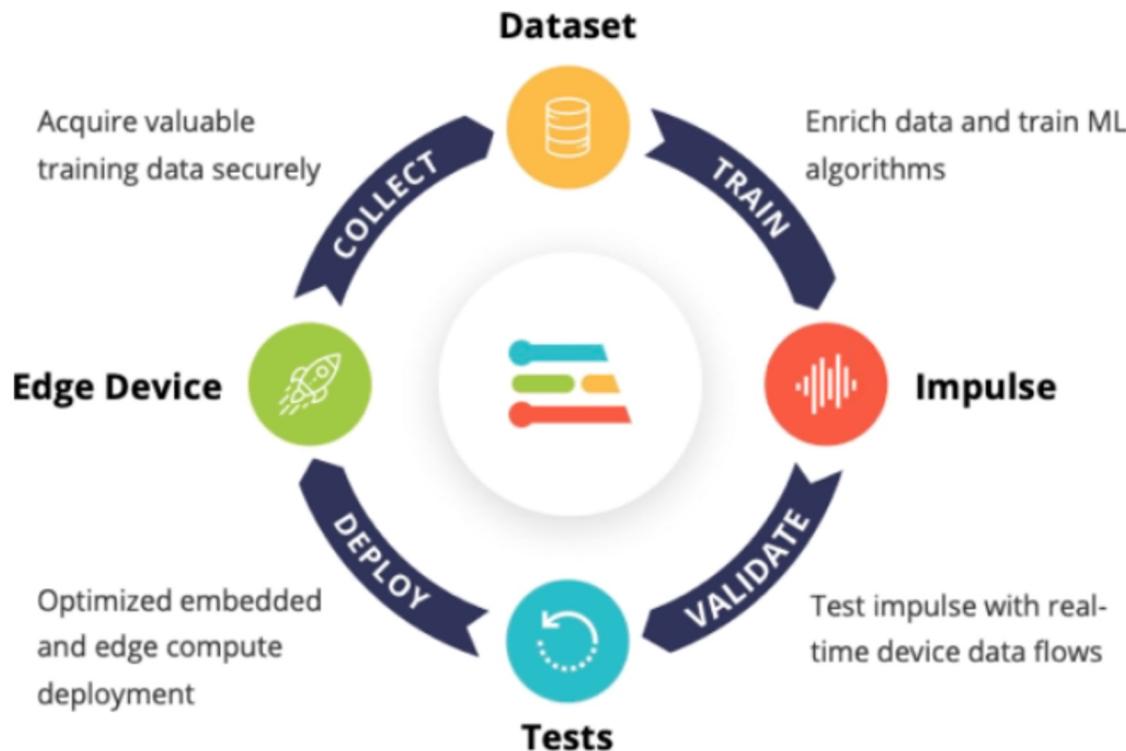


Machine Learning Workflow (“How”)





EI Studio - Embedded ML platform (“AutoML”)



Learn more at <http://edgeimpulse.com>



Cifar10

Edge Impulse Studio



Dataset: <https://github.com/YoongiKim/CIFAR-10-images>

El Studio Public Project: <https://studio.edgeimpulse.com/public/51070/latest>

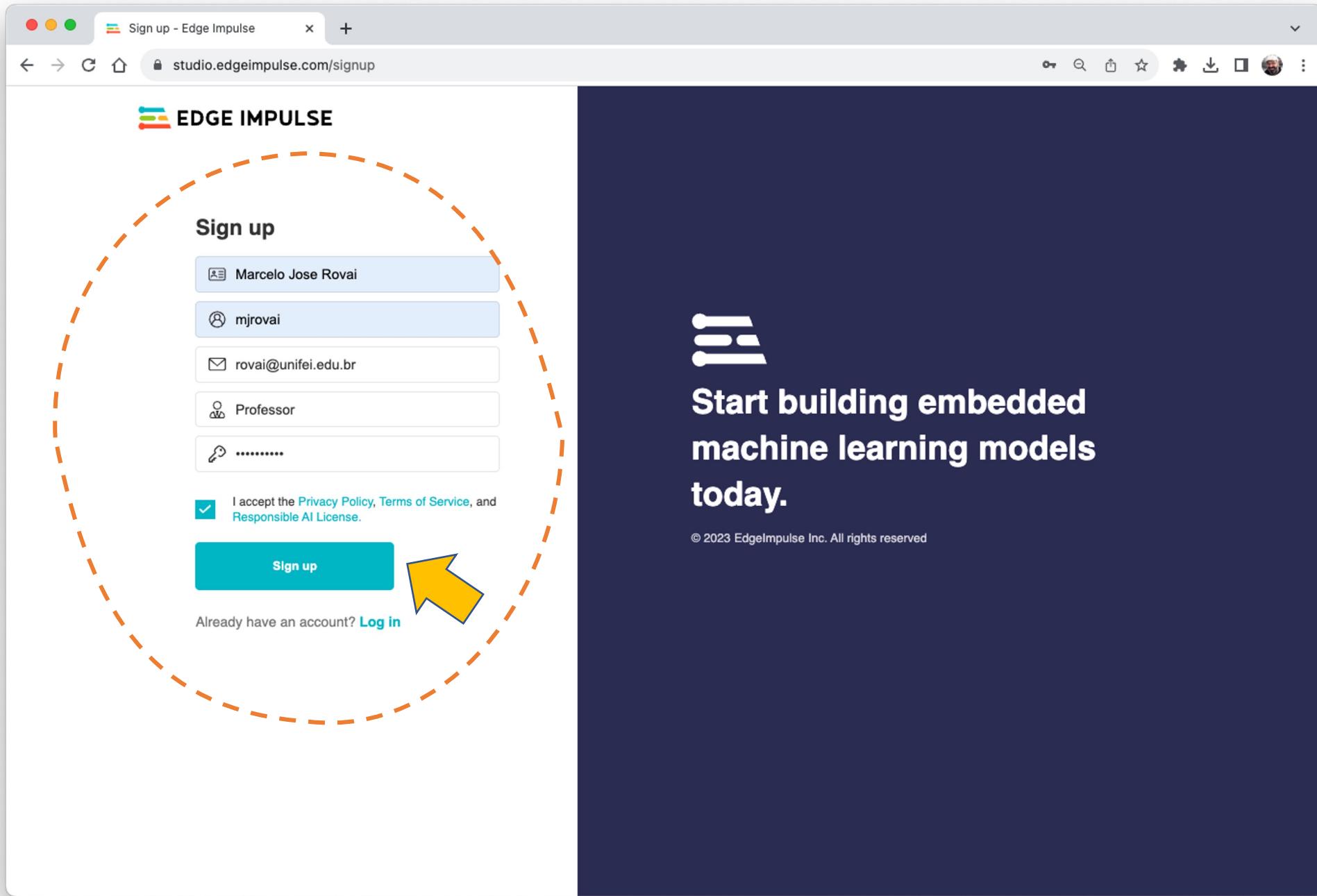
The screenshot shows the homepage of the Edge Impulse website. At the top, there's a navigation bar with links for 'Product', 'Solutions', 'Developers', 'Pricing', 'Company', and 'Blog'. On the right side of the header, there are 'Login' and 'Get started' buttons. A yellow arrow labeled '1' points to the 'edgeimpulse.com' URL in the browser's address bar. Another yellow arrow labeled '2' points to the 'Get started' button.

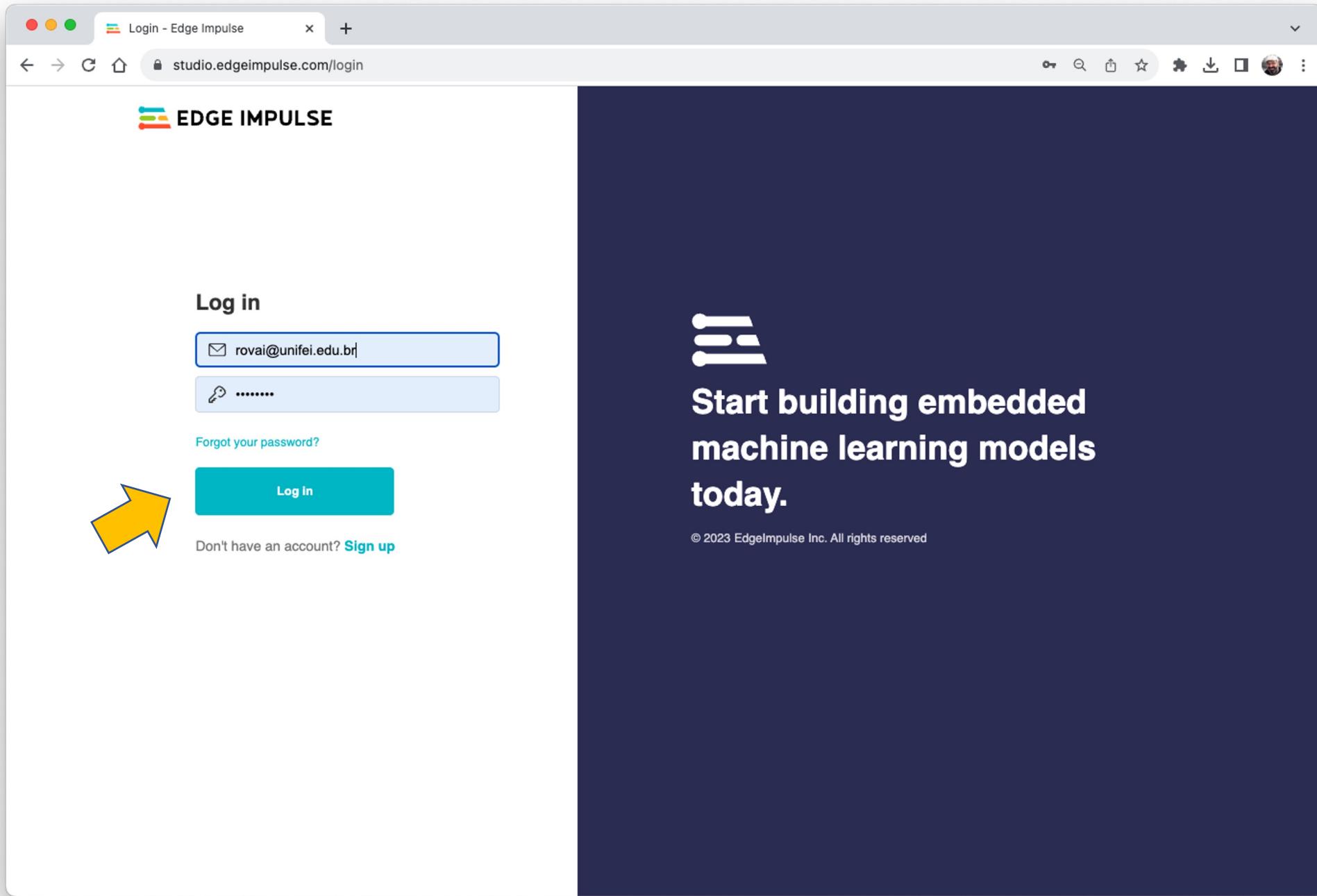
Build. Train. Optimize. AI for the edge.

Build datasets, train models, and optimize libraries to run on any edge device, from extremely low-power MCUs to efficient Linux CPU targets and GPUs.

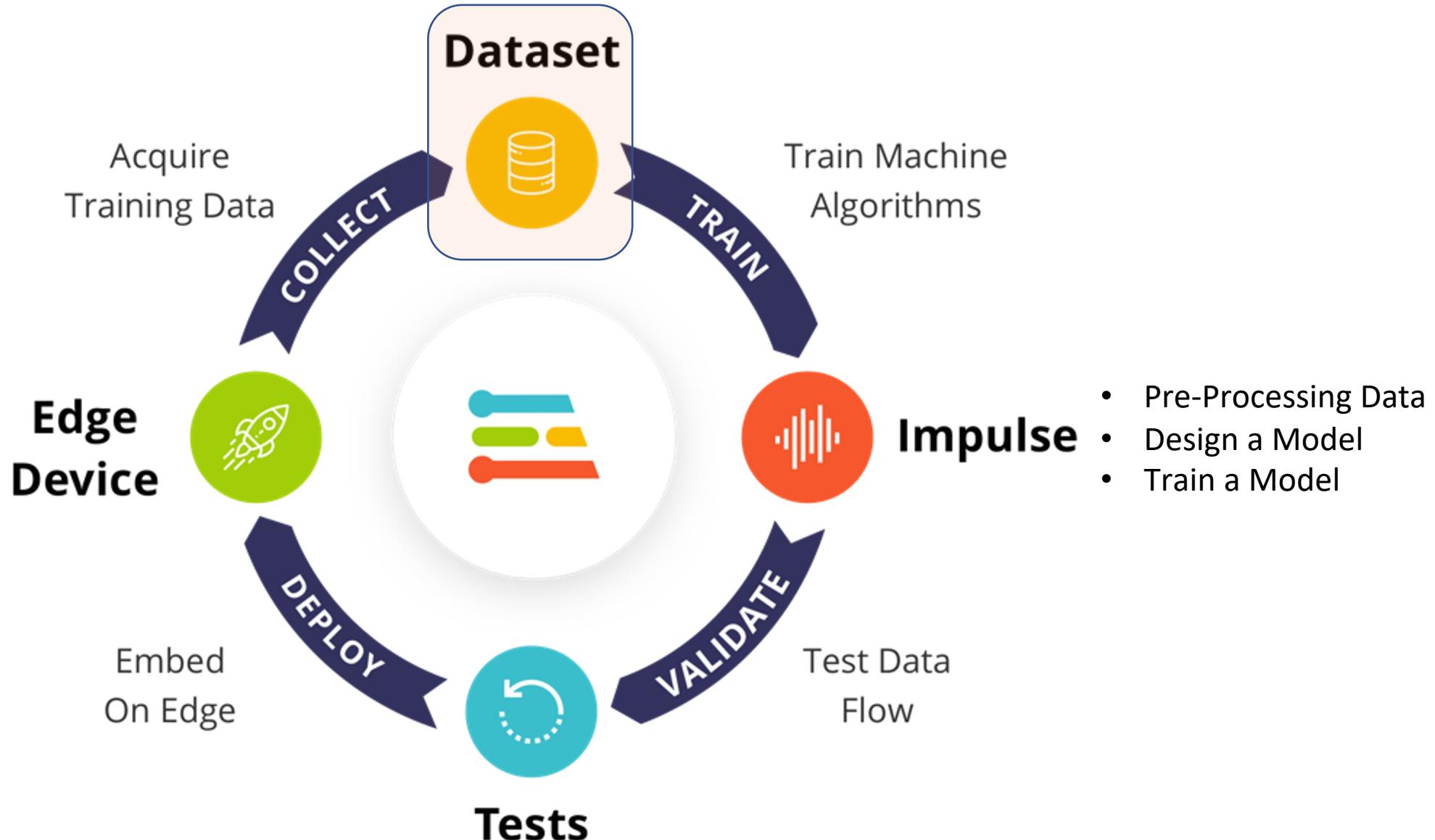
[Get Started](#) [Schedule a demo](#)

Webinar November 9th: 'Fast Track AI to the Edge with NVIDIA and Edge Impulse' [Register here](#)



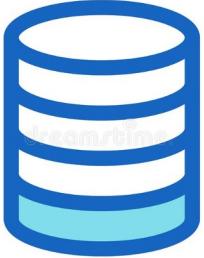


The screenshot shows the Edge Impulse studio interface. On the left, there's a sidebar with a user profile picture of a man with a beard, the name "MJRoBot", and the subtitle "(Marcelo Rovai)". Below this is a section titled "Organizations" with a purple button labeled "EIE". The main area has tabs for "Projects" (which is selected) and "Custom ML blocks". A modal window titled "Create a new project" is open. It contains a text input field with the placeholder "Enter the name for your new project:" and the value "Cifar10_Image_Classification" (which is highlighted with an orange border). Below this is a section to "Choose your project type:" with two options: "Developer" (selected, indicated by a blue radio button) and "Enterprise" (indicated by a white radio button). Under "Developer", it says "20 min job limit, 4GB or 4 hours of data, limited collaboration." Under "Enterprise", it says "No job or data size limits, higher performance, custom blocks." There's also a dropdown menu "Create under organization:" set to "Edge Impulse Experts". At the bottom of the modal is a green "Create new project" button. In the background, several project cards are visible, including "MJRoBot (Marcelo Rovai) / Cifar10_Image_Classification", "MJRoBot (Marcelo Rovai) / video_tinyml_raw", "MJRoBot (Marcelo Rovai) / Pico_Motion_Detection" (marked as PUBLIC), and "MJRoBot (Marcelo Rovai) / Loi_kovis_low_meetup". A yellow arrow points from the bottom right towards the "Create new project" button.

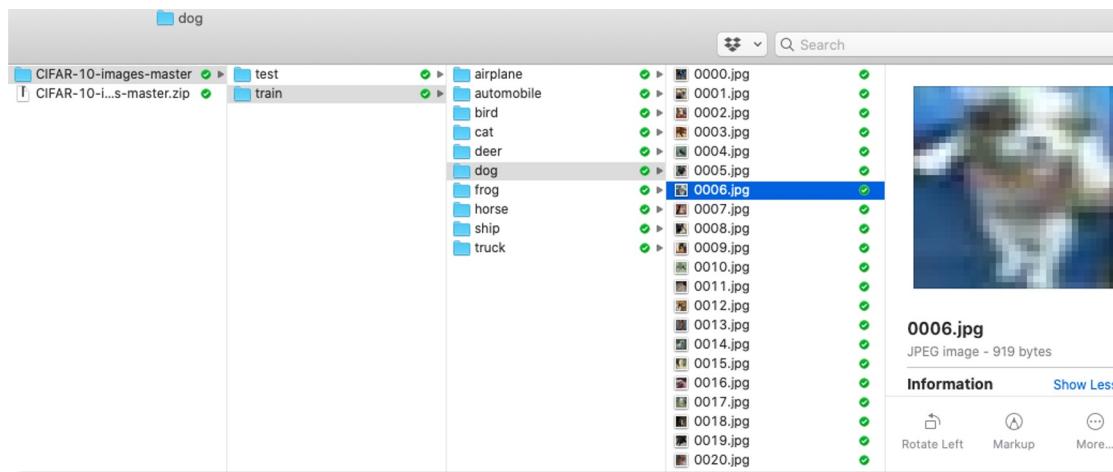


Download Dataset

<https://github.com/YoongiKim/CIFAR-10-images>



The screenshot shows a GitHub repository page for 'CIFAR-10-images'. The repository has 1 branch and 0 tags. It contains files: 'test' (Upload), 'train' (Upload), and 'README.md' (Create README.md). A modal window is open over the repository details, showing options: 'Clone' (HTTPS, SSH, GitHub CLI), 'About' (CIFAR-10 raw jpeg images, Readme), 'Releases' (No releases published), and 'Packages' (No packages published). A 'Download ZIP' button is highlighted with a yellow arrow. The main content area displays the 'README.md' file which says 'CIFAR-10 raw jpeg images' and 'You can just clone this repository to use dataset.'



Collect Data

Upload data - Cifar10_Image_C

stUDIO.edgeimpulse.com/studio/51070/upload

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- NN Classifier

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

UPLOAD DATA (CIFAR10_IMAGE_CLASSIFICATION)

Upload existing data

You can upload existing data to your project in the [Data Acquisition Format](#) (CBOR, JSON, CSV), or as WAV, JPG or PNG files.

Select files 

No file chosen

Upload into category

Automatically split between training and testing ?

Training 

Testing

Label

Infer from filename ?

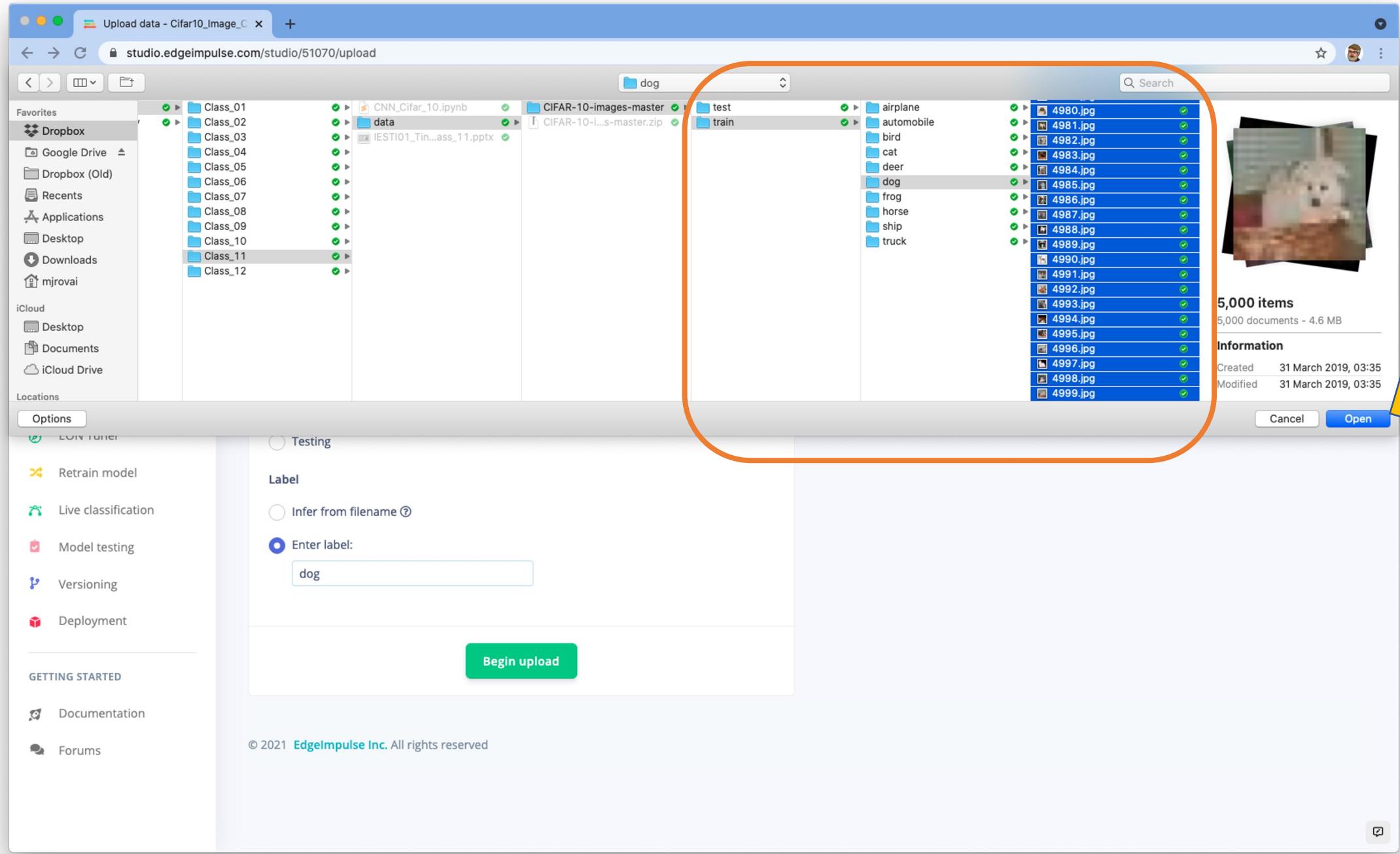
Enter label: 

dog

Begin upload

© 2021 EdgeImpulse Inc. All rights reserved

MJRoBot (Marcelo Rovai)



Collect Data

Upload data - Cifar10_Image_C

stUDIO.edgeimpulse.com/studio/51070/upload

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- NN Classifier

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

UPLOAD DATA (CIFAR10_IMAGE_CLASSIFICATION)

Upload existing data

You can upload existing data to your project in the [Data Acquisition Format](#) (CBOR, JSON, CSV), or as WAV, JPG or PNG files.

Select files

[Choose Files](#) 5000 files

Upload into category

Automatically split between training and testing ?

Training

Testing

Label

Infer from filename ?

Enter label:

dog

Begin upload



© 2021 EdgeImpulse Inc. All rights reserved

MJRoBot (Marcelo Rovai)

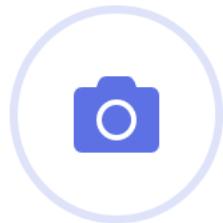


IMAGE DATA DETECTED!

Are you building an object detection project?

You can change this choice under "Dashboard > Labeling method".

The form consists of two rectangular buttons. The left button is blue with the word "Yes" in white. The right button is dark blue with the word "No" in white. An orange square outline surrounds the "No" button, indicating it is the selected option.

Collect Data

EDGE IMPULSE

Dataset Data explorer Data sources | CSV Wizard

DATA COLLECTED
12,002 items

TRAIN / TEST SPLIT
83% / 17%

Dataset

Training (10,000) Test (2,002)

Apply filters

By label

cat (5000)

dog (5000)

By name

Enter a sample name

By signature validity

Valid & invalid signatures

Enabled & disabled samples

Enabled & disabled samples

Delete (0) Edit labels (0) Move to test set (0) Enable (0) Disable (0)

<input type="checkbox"/>	SAMPLE NAME	LABEL	ADDED
<input type="checkbox"/>	4999.jpg.2hfb34u0	dog	Oct 09 2021, 11:1...
<input type="checkbox"/>	4997.jpg.2hfb34n1	dog	Oct 09 2021, 11:1...
<input type="checkbox"/>	4991.jpg.2hfb34mm	dog	Oct 09 2021, 11:1...
<input type="checkbox"/>	4993.jpg.2hfb34n8	dog	Oct 09 2021, 11:1...

Collect data

Connect a device to start building your dataset.

RAW DATA
4999.jpg.2hfb34u0

Metadata

No metadata.

Cifar10_Image_Classification

MJRoBot (Marcelo Rovai) / Cifar10_Image_Classification

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.](#)

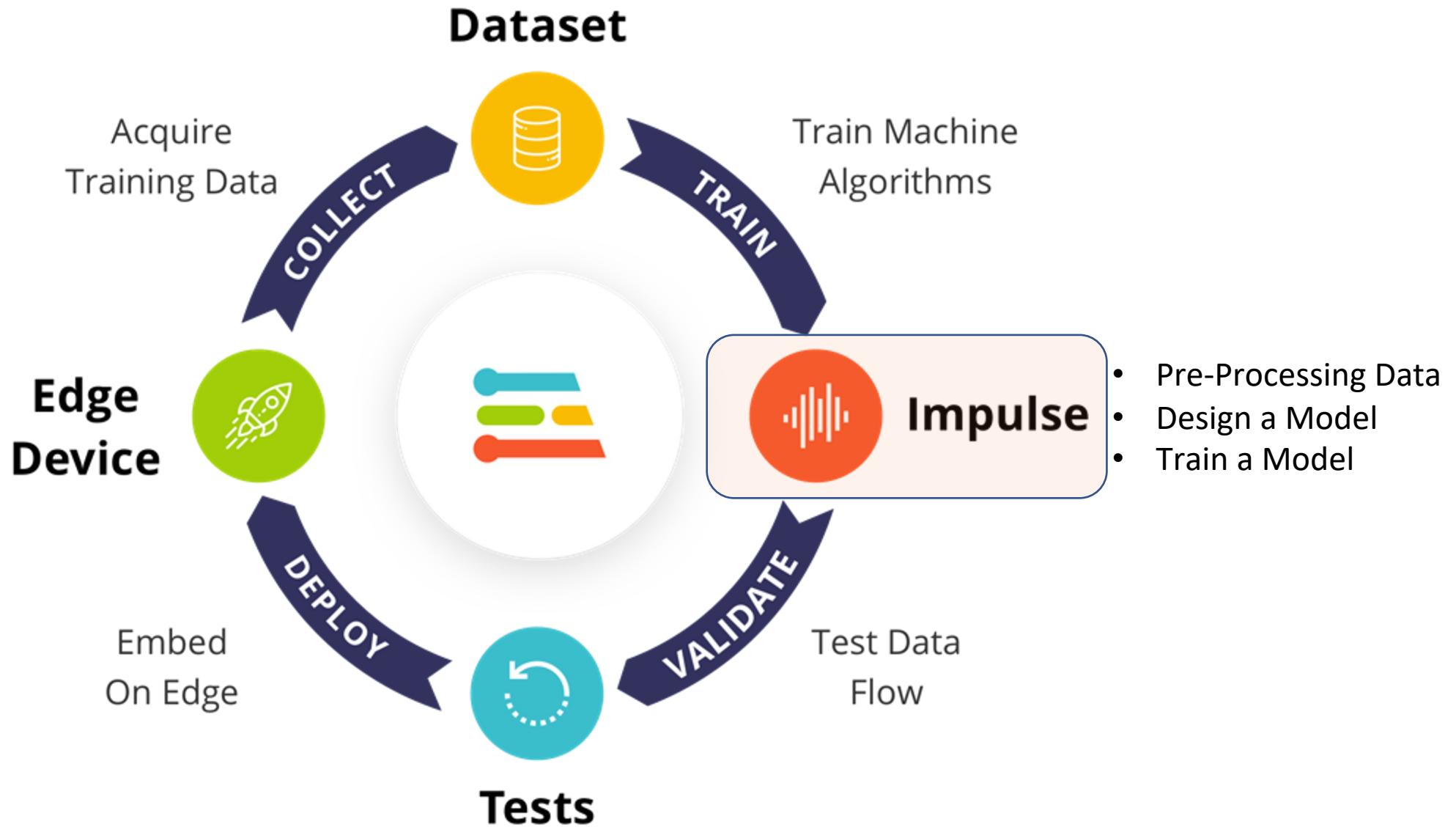
How should we generate the data explorer?

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

Generate data explorer



Create Impulse

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

Image data

Axes: 32x32x3

Image width: 32 Image height: 32

Resize mode: Fit longest axis

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

Image

Name: Image

Input axes: image

Classification (Keras)

Name: NN Classifier

Input features: Image

Output features: 2 (cat, dog)

Output features

2 (cat, dog)

Save Impulse

© 2021 EdgeImpulse Inc. All rights reserved

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

GETTING STARTED
Documentation
Forums

MJRoBot (Marcelo Rovai)

Preprocess Data

Image - Cifar10_Image_Classifi x +

studio.edgeimpulse.com/studio/51070/dsp/image/3

EDGE IMPULSE

IMAGE (CIFAR10_IMAGE_CLASSIFICATION)

#1 Click to set a description for this version

Parameters Generate features

Raw data

4999.jpg.2hfb34u0 (dog) ▾

Raw features

0xa9aaa4, 0xafb0aa, 0xabaca6, 0xa9aaa4, 0xaebe1a8, 0xabaee5, 0xa6a9a0, 0xabaee5, 0xabaee3, 0xaaad...

Parameters

Image

Color depth

RGB

Save parameters

DSP result

Image

Processed features

0.6627, 0.6667, 0.6431, 0.6863, 0.6902, 0.6667, 0.6706, 0.6745, 0.6510, 0.6627, 0.6667, 0.6431, ...

On-device performance

PROCESSING TIME 9 ms.

PEAK RAM USAGE 4 KB

© 2021 EdgeImpulse Inc. All rights reserved

MJRoBot (Marcelo Rovai)

Preprocess Data

IMAGE (CIFAR10_IMAGE_CLASSIFICATION)
#1 Click to set a description for this version

Parameters Generate features

Training set

Data in training set 10,000 items

Classes 2 (cat, dog)

Generate features

Feature explorer (10,000 samples)

X Axis Y Axis Z Axis

Visualization layer 1 Visualization layer 2 Visualization layer 3

cat dog

8.5
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0

Visualization layer 3
Visualization layer 2
Visualization layer 1

On-device performance

PROCESSING TIME 9 ms.

PEAK RAM USAGE 4 KB

© 2021 Edgimpulse Inc. All rights reserved

MJRoBot (Marcelo Rovai)

Design a Model

Train a Model

Cifar10_Image_Classification

studio.edgeimpulse.com/studio/51070/learning/keras/6

#2 ▾ Model - Same as done in class (IESTI01) ★ Primary version

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- NN Classifier

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

Neural Network settings

Training settings

Number of training cycles ② 10

Learning rate ② 0.0005

Advanced training settings

Validation set size ② 20 %

Split train/validation set on metadata key ②

Batch size ② 32

Auto-balance dataset ②

Profile int8 model ②

Neural network architecture

Input layer (3,072 features)

2D conv / pool layer (32 filters, 3 kernel size, 1 layer)

2D conv / pool layer (64 filters, 3 kernel size, 1 layer)

Flatten layer

Dense layer (64 neurons)

Add an extra layer

Output layer (2 classes)

Start training

Training output CPU (0)

Model

Model version: ② Quantized (int8)

Last training performance (validation set)

ACCURACY 74.4%

LOSS 0.56

Confusion matrix (validation set)

	CAT	DOG
CAT	80.6%	19.4%
DOG	32.1%	67.9%
F1 SCORE	0.76	0.72

Feature explorer (full training set) ②

Visualization layer 1

Visualization layer 2

Visualization layer 3

Legend:

- cat - correct
- dog - correct
- cat - incorrect
- dog - incorrect

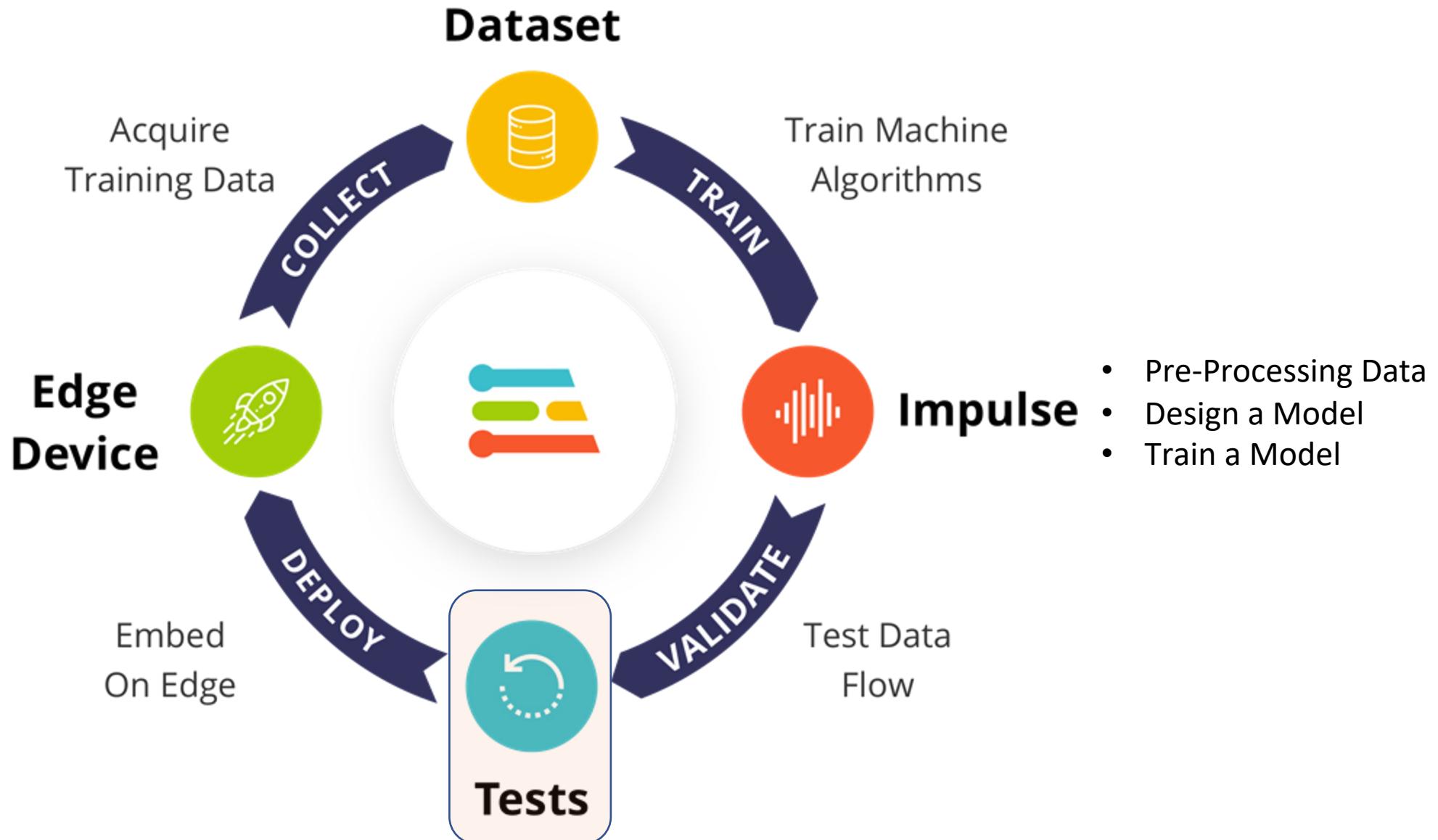
On-device performance ②

INFERENCING TIME 93 ms.

PEAK RAM USAGE 44.7K

FLASH USAGE 308.2K

Target: Arduino Nicla Vision (Cortex-M7 480MHz)



Model testing - Cifar10_Image... Upload data - Cifar10 Classific... | +

studio.edgeimpulse.com/studio/51070/validation

EDGE IMPULSE

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

Evaluate Optimize

Test data

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

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT	
testing.2hfe5uat	testing	-	1 cat	⋮	
0999.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0997.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0998.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0996.jpg.2hfb7a...	cat	-	0%	1 uncertain	⋮
0993.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0995.jpg.2hfb7a...	cat	-	0%	1 dog	⋮
0991.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0994.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0992.jpg.2hfb7a...	cat	-	100%	1 cat	⋮
0990.jpg.2hfb79...	cat	-	100%	1 cat	⋮
0988.jpg.2hfb79...	cat	-	100%	1 cat	⋮
0985.jpg.2hfb79...	cat	-	0%	1 uncertain	⋮

Model testing output

Classifying data for NN classifier...
 Copying features from processing blocks...
 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...
 Job started
 Classifying data for NN Classifier OK

Job completed

Model testing results

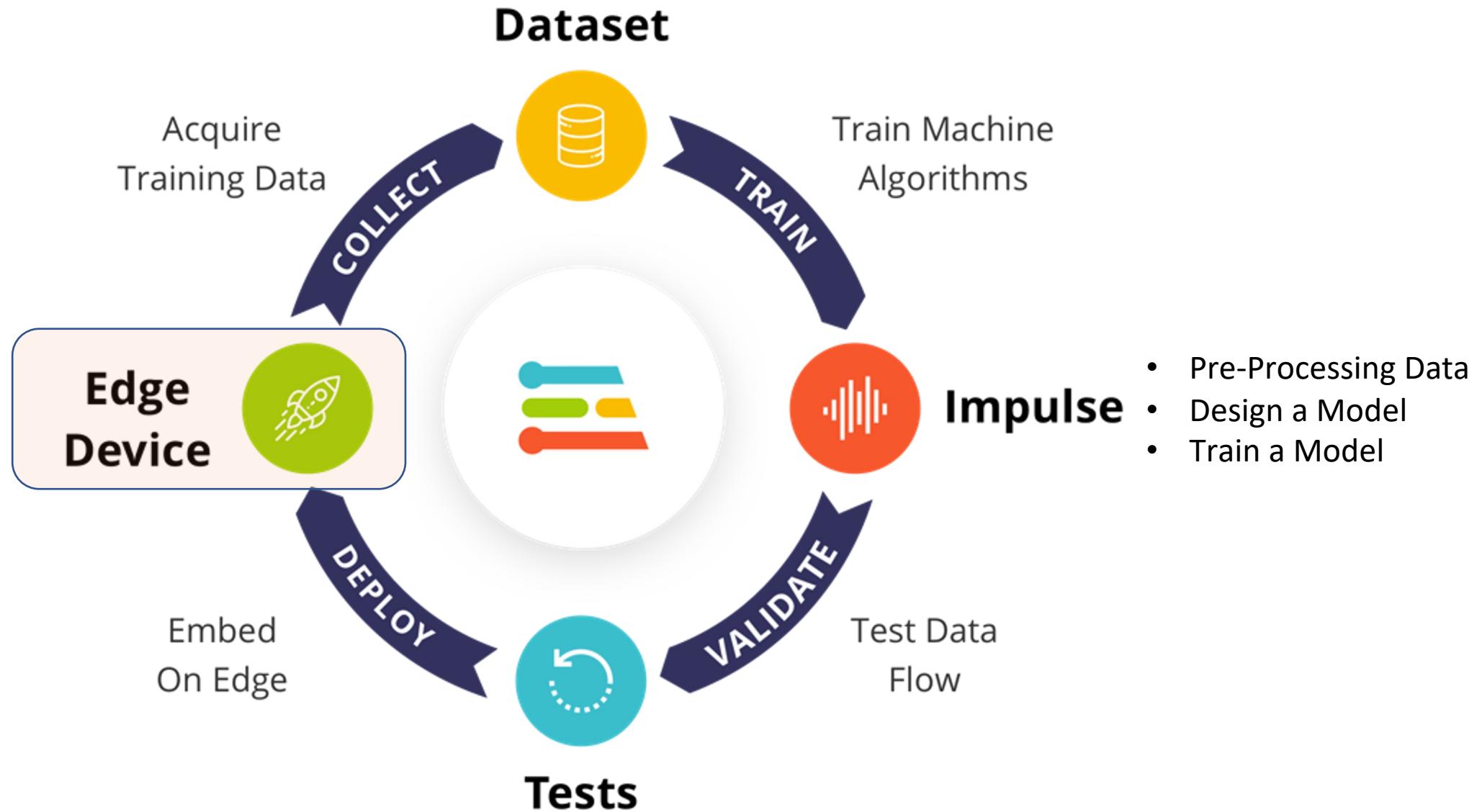
ACCURACY **64.95%** %

	CAT	DOG	UNCERTAIN
CAT	67.7%	16%	16.3%
DOG	21.2%	62.2%	16.6%
F1 SCORE	0.72	0.70	

Feature explorer ⓘ

Legend:

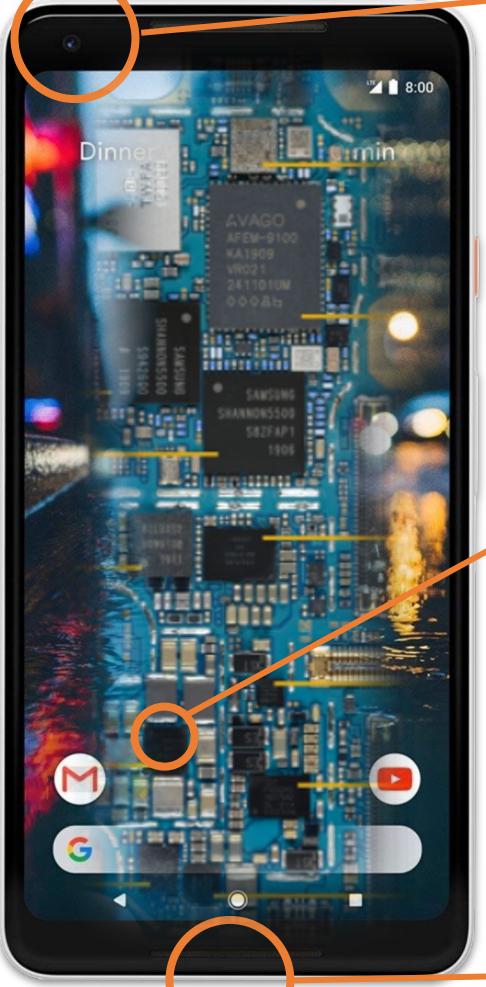
- cat - correct
- dog - correct
- cat - incorrect
- dog - incorrect
- testing



Edge Device



& Sensors



Camera



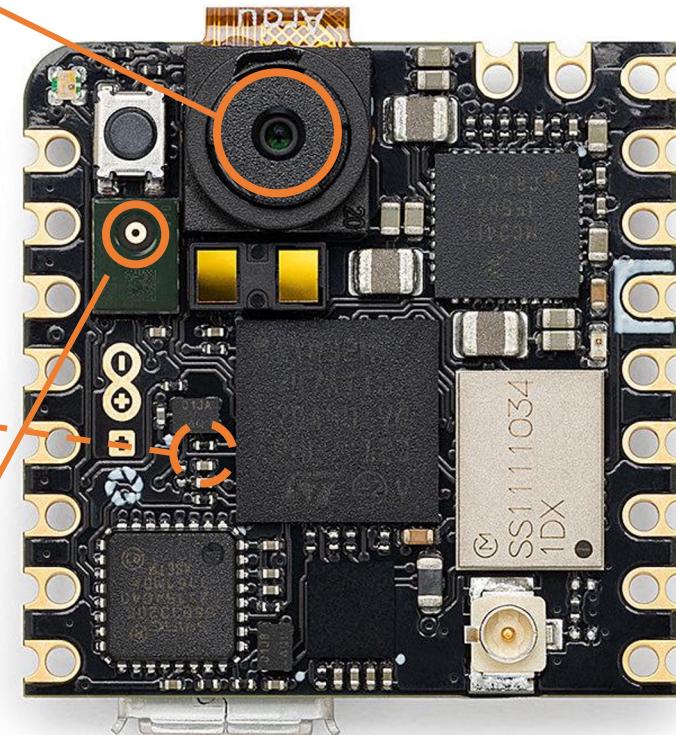
Accelerometer



Gyroscope



Microphone



Deploy Model

The screenshot shows the Edge Impulse studio deployment interface. On the left, a sidebar lists various project management and development tools. The main area is titled "Configure your deployment" and "Run this model". The "Run this model" section features a QR code and a "Launch in browser" button, which is highlighted with an orange rounded rectangle. Below this, there's a "DEFAULT DEPLOYMENT" section for a "C++ library" and a "MODEL OPTIMIZATIONS" section for "Quantized (int8)" and "Unoptimized (float32)". A large blue "Build" button is at the bottom.

Run this model

Scan QR code or launch in browser to test your prototype

DEFAULT DEPLOYMENT

C++ library

A portable C++ library with no external dependencies, which can be compiled with any modern C++ compiler.

MODEL OPTIMIZATIONS

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

Enable EON™ Compiler Same accuracy, up to 50% less memory. [Learn more](#)

Quantized (int8)

	IMAGE	NN CLASSIFIER	TOTAL
LATENCY	1 ms.	93 ms.	94 ms.
RAM	4.0K	44.7K	44.7K
FLASH	-	308.2K	-
ACCURACY			-

Unoptimized (float32)

	IMAGE	NN CLASSIFIER	TOTAL
LATENCY	1 ms.	206 ms.	207 ms.
RAM	4.0K	163.4K	163.4K
FLASH	-	1.1M	-
ACCURACY			64.95%

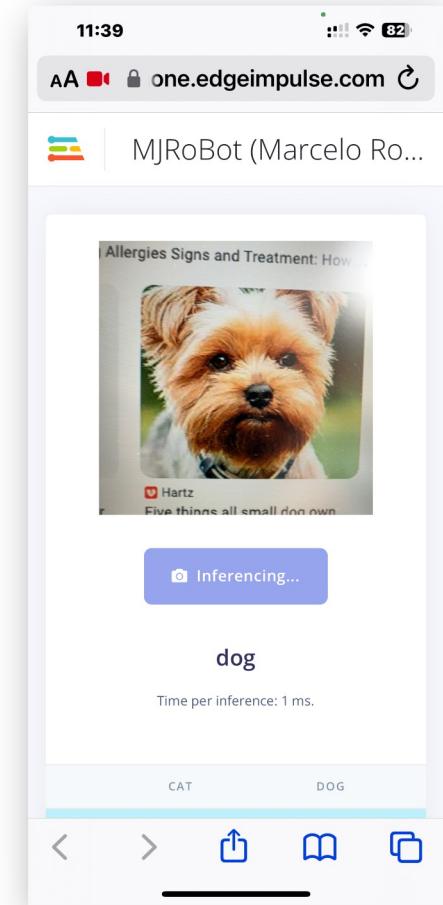
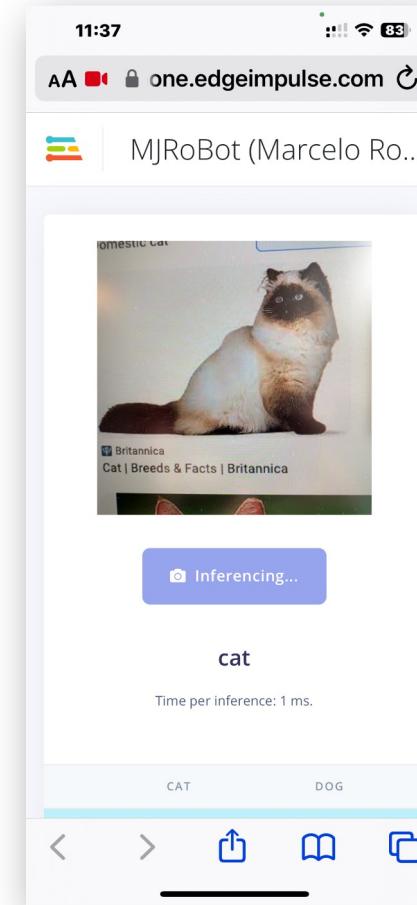
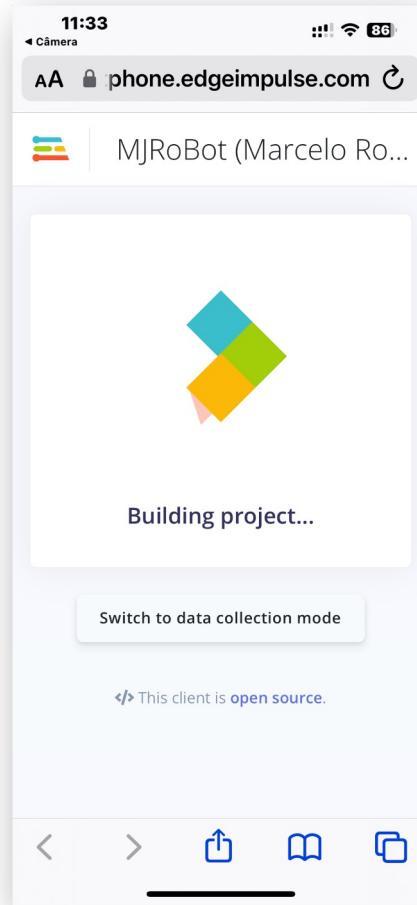
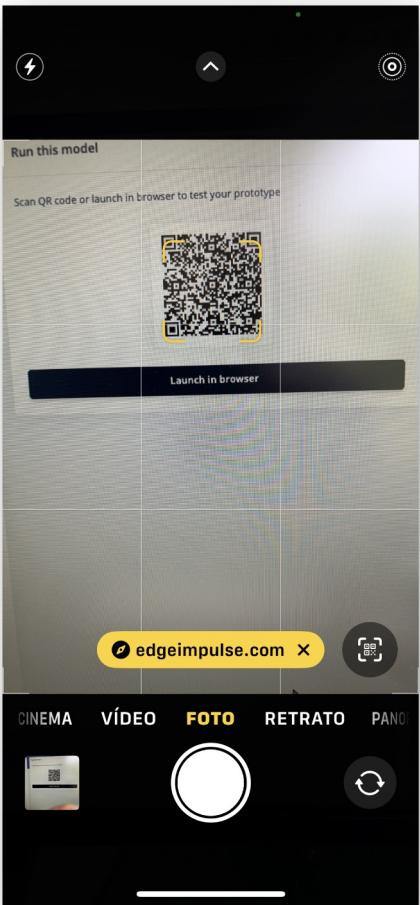
To compare model accuracy, run model testing for all available optimizations. [Run model testing](#)

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

Build



Make Inferences



El Studio “under the hood”



NN Classifier - Cifar10_Image_... X Upload data - Cifar10 Classific... X G computer icon - Pesquisa Goog... X +

studio.edgeimpulse.com/studio/51070/learning/keras/6

EDGE IMPULSE

NN CLASSIFIER (CIFAR10_IMAGE_CLASSIFICATION)
#2 ▾ Model - Same as done in class (IESTI01) ★ Primary version

MJRoBot (Marcelo Rovai)

Neural Network settings

Training settings

Neural network architecture

```

1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten,
4     Reshape, MaxPooling1D, MaxPooling2D, BatchNormalization, TimeDistributed
5 from tensorflow.keras.optimizers import Adam
6
7 # model architecture
8 model = Sequential()
9 model.add(Conv2D(32, kernel_size=3, activation='relu', kernel_constraint=tf.keras
    .constraints.MaxNorm(1), padding='same'))
10 model.add(MaxPooling2D(pool_size=2, strides=2, padding='same'))
11 model.add(Conv2D(64, kernel_size=3, activation='relu', kernel_constraint=tf.keras
    .constraints.MaxNorm(1), padding='same'))
12 model.add(MaxPooling2D(pool_size=2, strides=2, padding='same'))
13 model.add(Flatten())
14 model.add(Dense(64, activation='relu',
    activity_regularizer=tf.keras.regularizers.l1(0.0001)))
15 model.add(Dense(classes, activation='softmax', name='y_pred'))
16
17 # this controls the learning rate
18 opt = Adam(lr=0.0005, beta_1=0.9, beta_2=0.999)
19 # this controls the batch size, or you can manipulate the tf.data.Dataset objects
    yourself
20 BATCH_SIZE = 32
21 train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
22 validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)
23 callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count))
24
25 # train the neural network
26 model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

```

Training output

Model Model version: ② Quantized (int8)

Last training performance (validation set)

ACCURACY 74.4% **LOSS** 0.56

Confusion matrix (validation set)

	CAT	DOG
CAT	80.6%	19.4%
DOG	32.1%	67.9%
F1 SCORE	0.76	0.72

Feature explorer (full training set) ②

- cat - correct
- dog - correct
- cat - incorrect
- dog - incorrect

On-device performance ②

INFERENCING TIME 1,048 ms. **PEAK RAM USAGE** 44.7K **FLASH USAGE** 308.2K

<https://studio.edgeimpulse.com/studio/51070/learning/keras/6>

Connect device for data capture
Mobile Phone

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse
- Image
- NN Classifier

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

Collect new data

Collect data directly from your phone, computer, device, or development board.

Scan QR code to connect to your phone

Connect to your computer

Connect your device or development board

© 2023 Edgeimpulse Inc. All rights reserved

?

studiostudio.edgeimpulse.com/studio/51070/devices#collect-new-data

MICROBOT (Marcelo Bouvier) / Cifar10_Images_Classification

+ Connect a new device

The screenshot illustrates the Edge Impulse studio interface across multiple devices. On the left, a desktop browser window shows the 'Devices' page with a list of available devices. A specific device, 'phone_lorah242', is highlighted and connected, as indicated by a prominent green checkmark and a message stating 'Device "phone_lorah242" is now connected'. A 'Get started!' button is visible below the message. The desktop interface includes a sidebar with various project management and development tools like 'Dashboard', 'Data acquisition', 'Impulse design', and 'Deployment'. On the right, two mobile devices are shown. Both devices display the 'Data collection' screen for their respective connected devices. The top mobile device shows 'phone.edgeimpulse.com' and the bottom one shows 'one.edgeimpulse.com'. Both screens feature a large green checkmark indicating a successful connection. Below the checkmark, text reads 'Connected as phone_lorah242' and 'You can collect data from this device from the Data acquisition page in the Edge Impulse studio.' At the bottom of each mobile screen, there are several blue icons for navigating between projects and managing files.

Using Edge Impulse and the Grove Vision AI Module V2



MJRoBot (Marcelo Rovai)

Published February 22, 2024 © Apache-2.0

Computer Vision at the Edge with Grove Vision AI Module V2

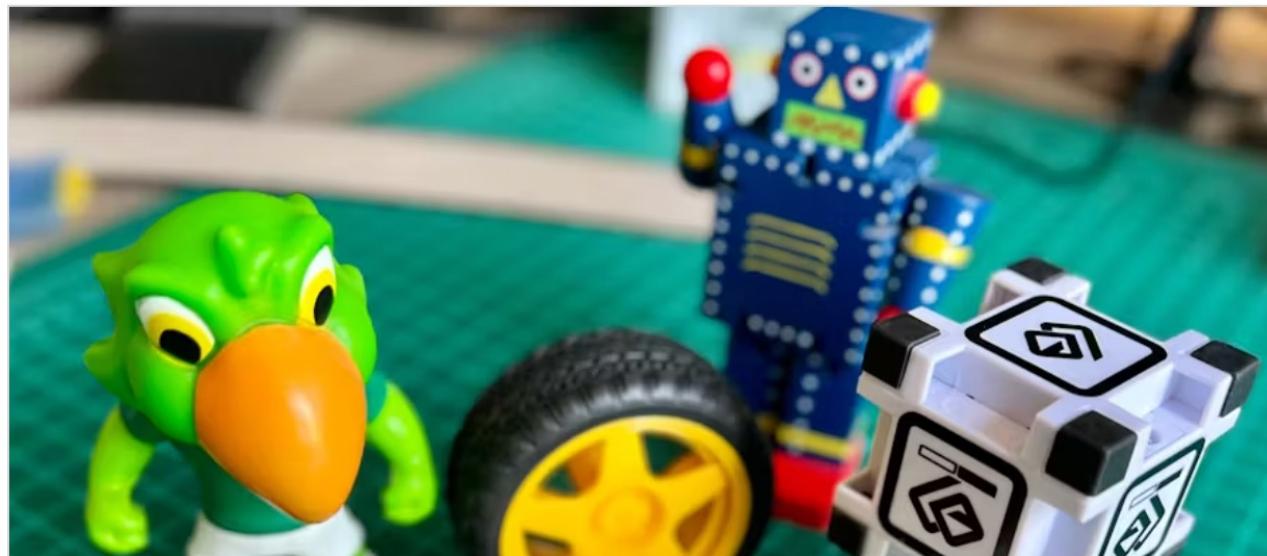
Exploring Computer Vision applications such as Image Classification, Object Detection, and Pose estimation.

⌚ Intermediate

📘 Full instructions provided

🕒 8 hours

👁 2,658



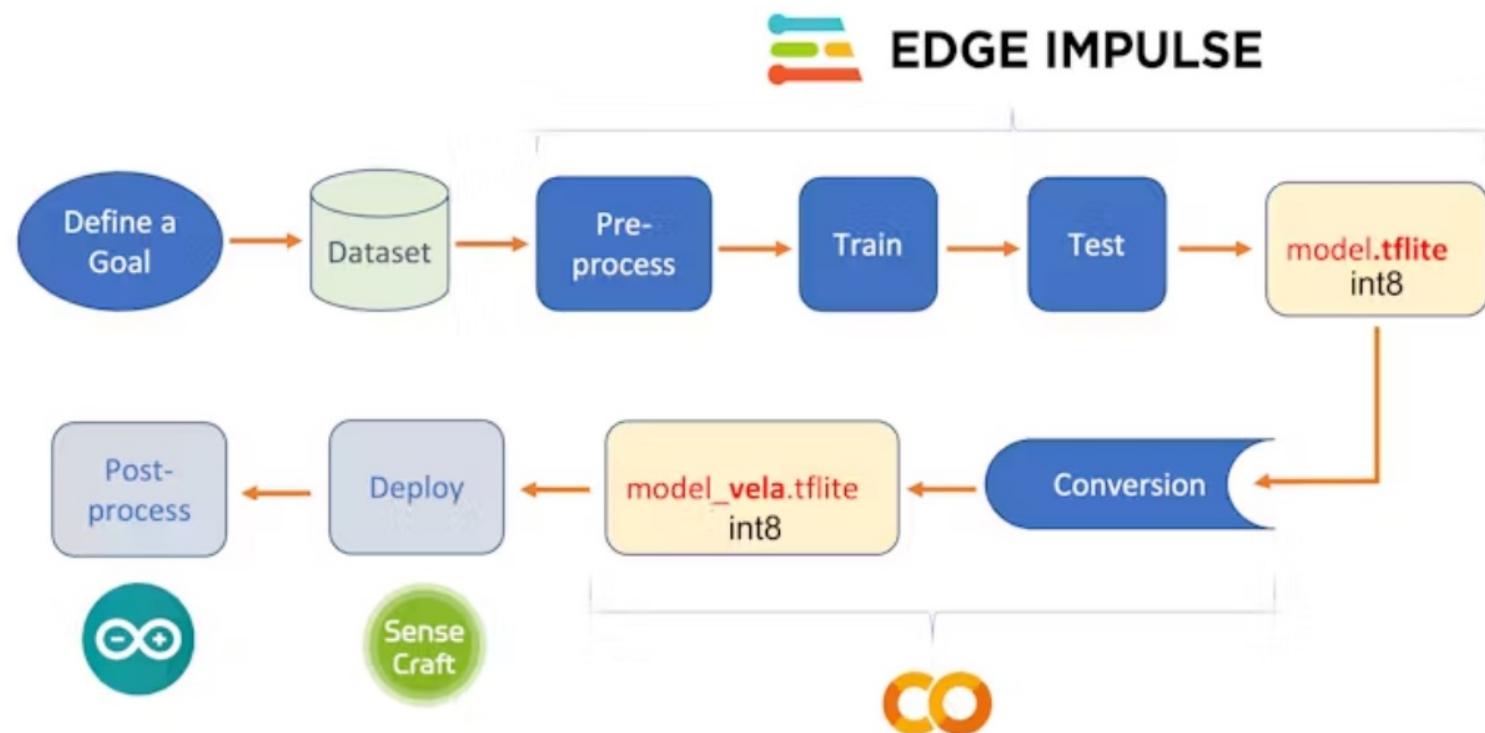
<https://www.hackster.io/mjrobot/computer-vision-at-the-edge-with-grove-vision-ai-module-v2-0003c7?f=1>

4. An Image Classification Project from Scratch

Introduction

So far, we have explored several computer vision models previously uploaded by Seeed Studio. Let's develop our CV project from scratch, starting with image classification.

Below, we can see the project's main steps and where we will do it:



Now, we will use [Vela](#), a tool to compile our quantized int8 TensorFlow Lite neural network model (`ei-periquito-vs-robot-int8.tflite`) into an optimized version that can run on an embedded system containing an [Arm Ethos-U NPU](#) (`ei-periquito-vs-robot-int8_vela.tflite`).

The optimized model will contain TensorFlow Lite Custom operators for those parts of the model that the Ethos-U NPU can accelerate. Parts of the model that cannot be accelerated are left unchanged and will run on the Cortex-M series CPU using an appropriate kernel (such as the [Arm](#) optimized [CMSIS-NN](#) kernels). After compilation, the optimized model can only be run on an Ethos-U NPU embedded system, such as the Grove AI Vision (V2).

On a Google CoLab, run the lines below:

```
!pip install ethos-u-vela
!!vela --version
```

We should get: `3.10.0`

Deploy model on the SenseCraft Web-Toolkit

On SenseCraft-Web-Toolkit, use the blue button at the bottom of the page:

[Upload Custom AI Model]. A window will pop up. Enter the Model file that you downloaded to your computer from Google Colab, choose a Model Name, and enter with labels (ID:Object):

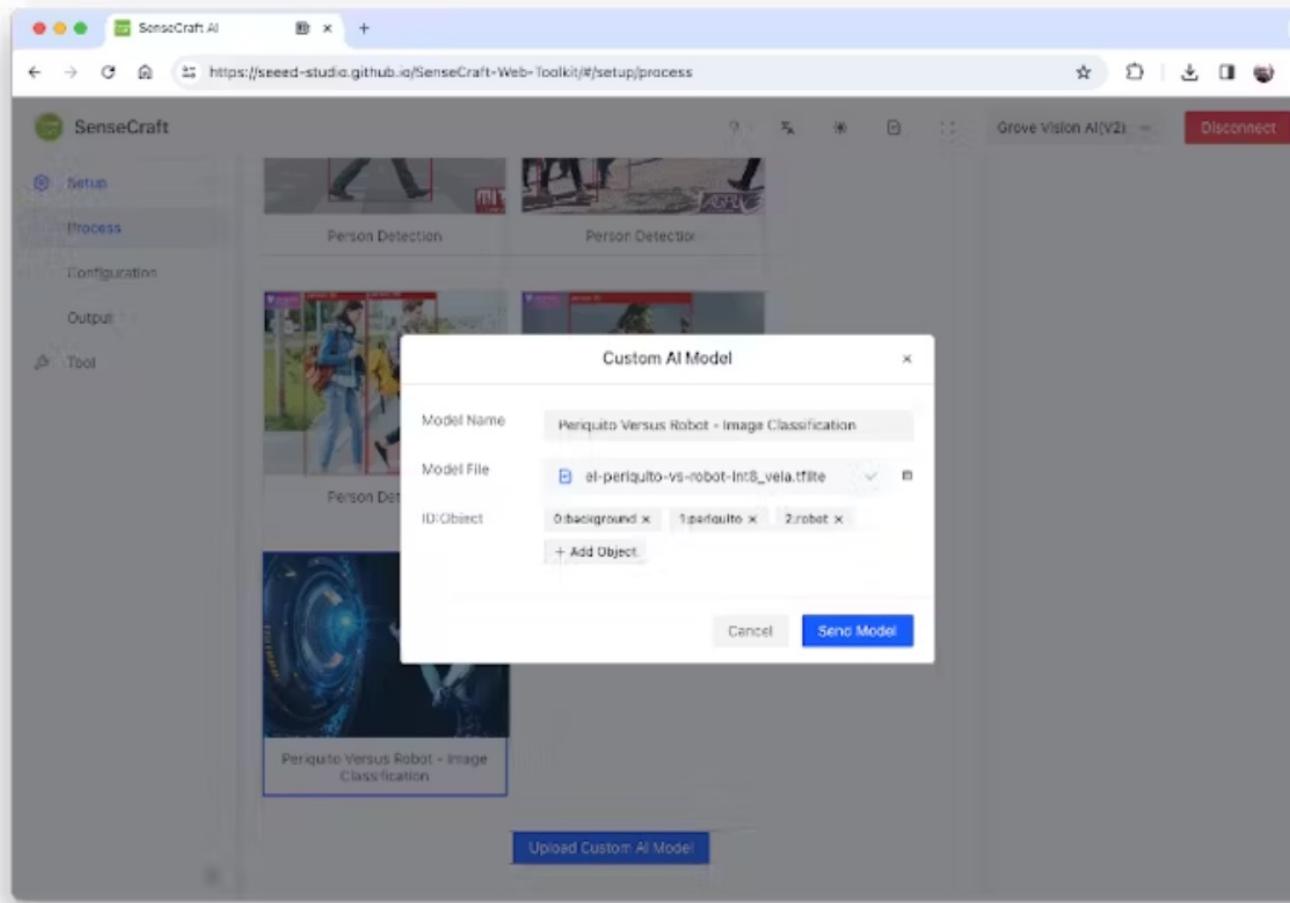


Image Classification (non-official) Benchmark

Several development boards can be used for embedded machine learning (tinyML), and the most common ones (so far) for Computer Vision applications (with low energy), are the ESP32 CAM, the Seeed XIAO ESP32S3 Sense, the Arduinos Nicla Vison, and Portenta.

Using the opportunity, a similarly trained model MobilenetV2 96x96, but with an alpha 0.1, was deployed on the ESP-CAM, the XIAO, and Portenta (in this one, the model was trained using grayscaled images to be compatible with its camera). Here is the result:



Thanks

