African Regional Workshop on SciTinyML: Scientific Use of Machine Learning on Low-Power Devices

25-29 April 2022
Online

Further information:
http://indico.ictp.it/event/9792/
smr3708@ictp.it

ICTP
TINY ML

Convolutions for Hands-on Computer Vision

Brian Plancher
Harvard John A. Paulson School of Engineering and Applied Sciences
Barnard College, Columbia University
brianplancher.com

Quick Disclaimer:

Today will be **both** **too fast**

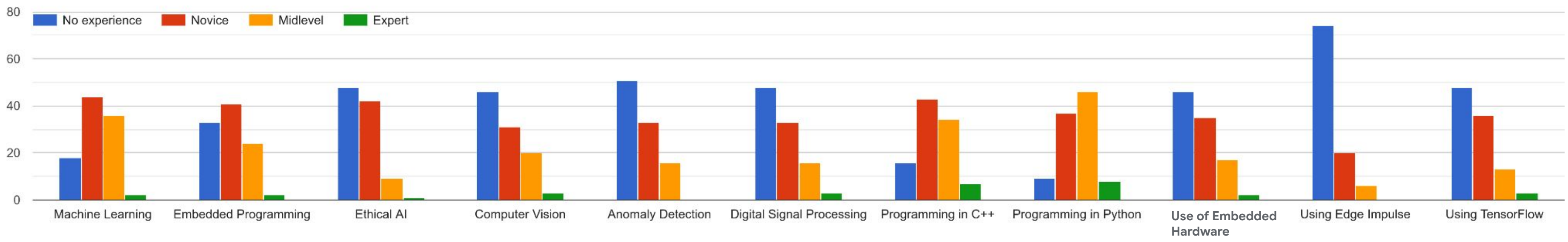
and **too slow**!

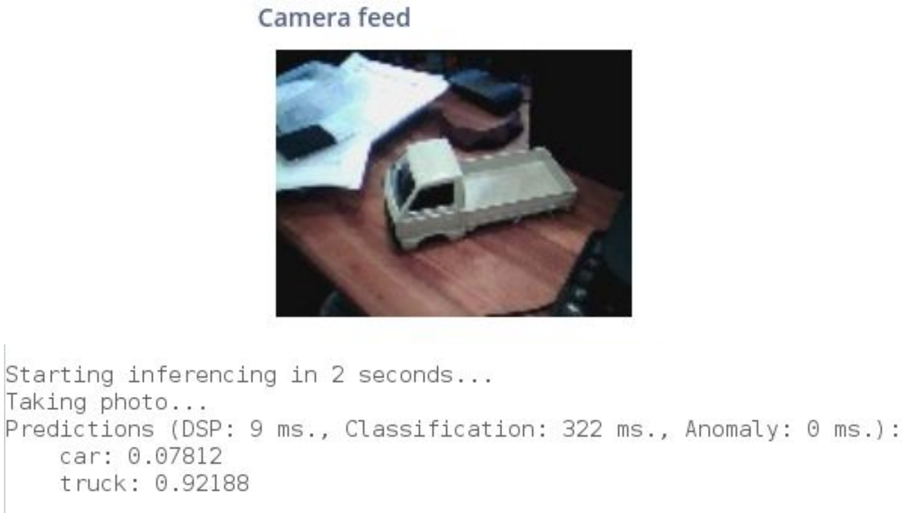
Quick Disclaimer:

Today will be **both too fast**

and **too slow**!

Do you have experience in?





```
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification: 322 ms., Anomaly: 0 ms.):
    car: 0.07812
    truck: 0.92188
```

By the end of today: **Hands-on Computer Vision** **(Object Classification)**

We will explore the **science** behind computer vision and **collect data** and **train** our own custom model to recognize objects using **Edge Impulse**

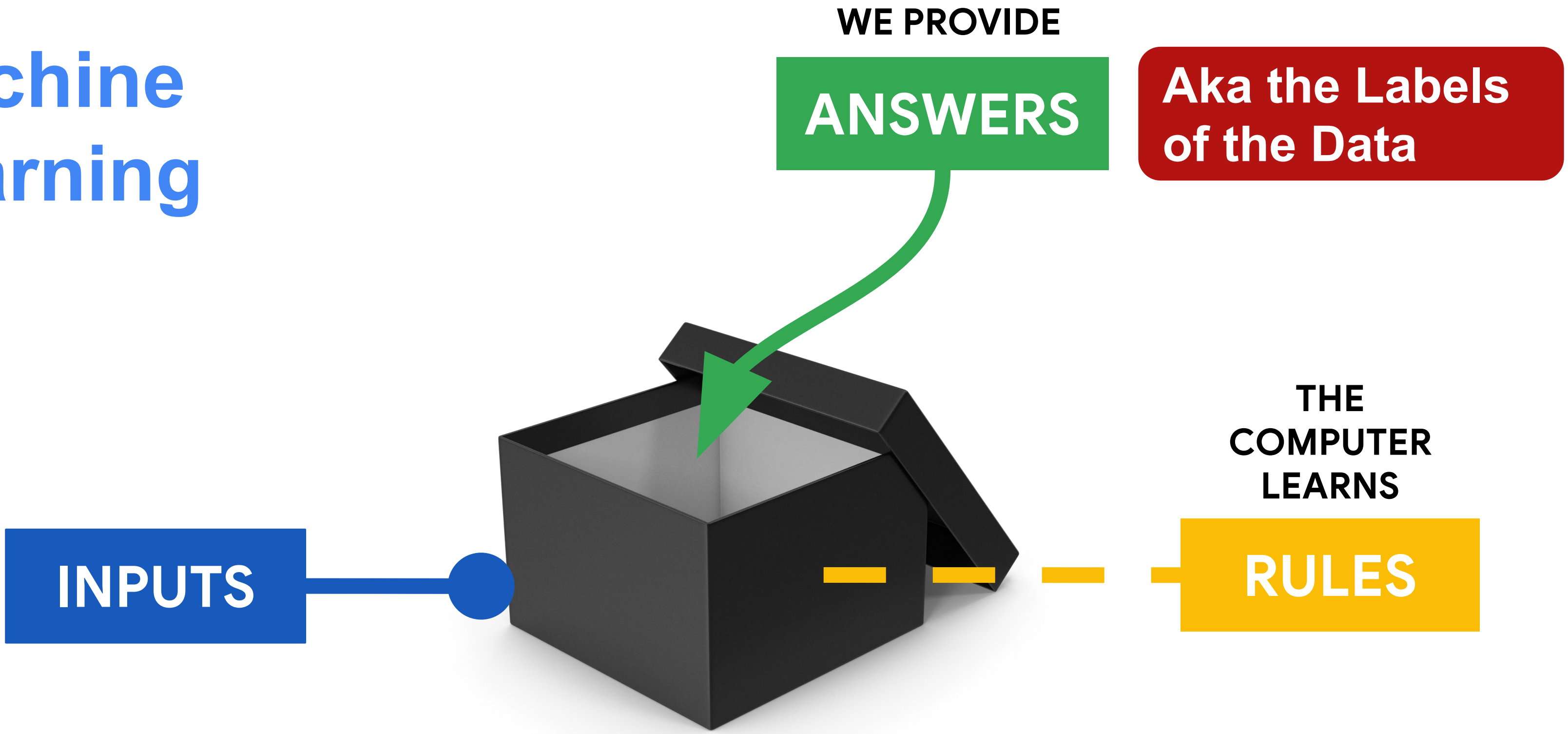
Today’s Agenda

- Introduction to Computer Vision
- Hands-on Computer Vision: Thing Translator
- Building an Object Detection Dataset
- Training our Model using Transfer Learning
- Deploying our Model onto our Arduino
- Summary

Today’s Agenda

- **Introduction to Computer Vision**
- Hands-on Computer Vision: Thing Translator
- Building an Object Detection Dataset
- Training our Model using Transfer Learning
- Deploying our Model onto our Arduino
- Summary

Machine Learning



WE PROVIDE

ANSWERS

Aka the Labels of the Data

THE COMPUTER LEARNS

RULES

INPUTS

Let’s try to figure out **what** she’s doing?

```
01010101001000110101
01010100101001001010
10101011010100101001
```

walking

```
11110101001001010101
01010010100101010100
11010110010101001111
```

running

```
00001110101110101101
01010111101011010101
11010111111001001011
```

biking

```
01111110101110101010
10101110101011010101
11111111100100001110
```

golfing

Let’s try to figure out **what** she’s doing?

```
010101010010001101010
010101001010010010100
101010110101001010101001
```

walking

```
111101010010010101010
010100101001010101000
110101100101010011111
```

running

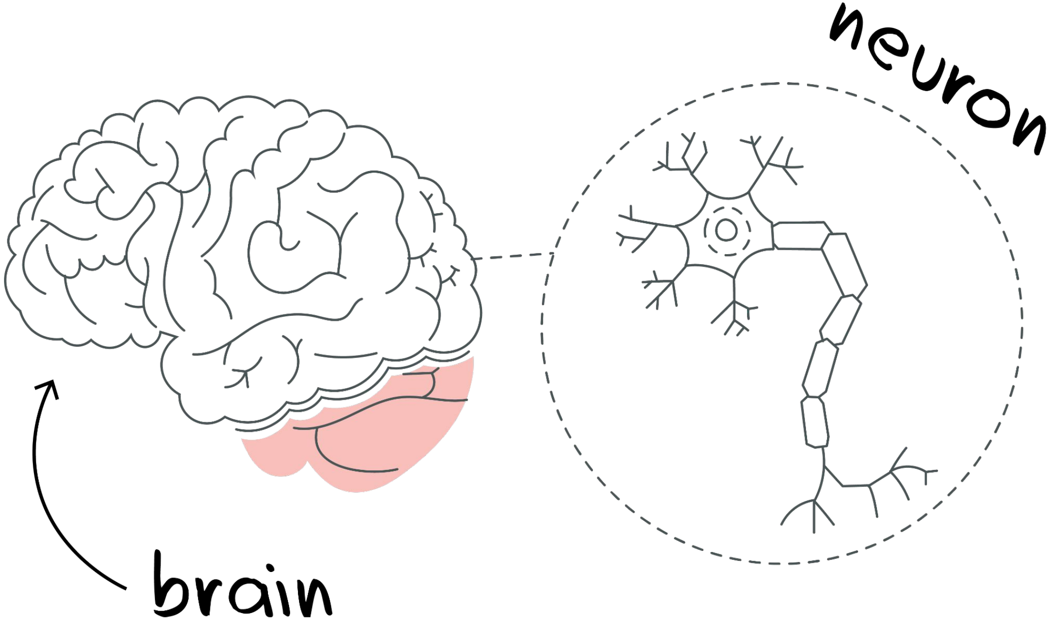
```
000011101011101011010
010101111010110101010
110101111110010010111
```

biking

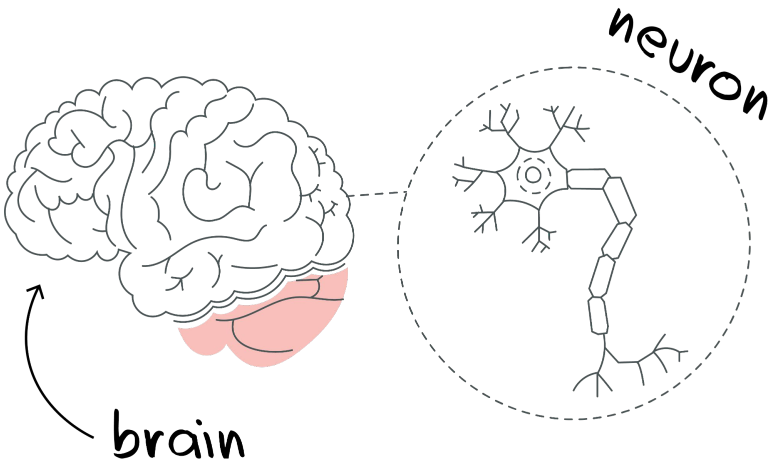
```
011111101011101010100
101011101010110101010
111111111001000011100
```

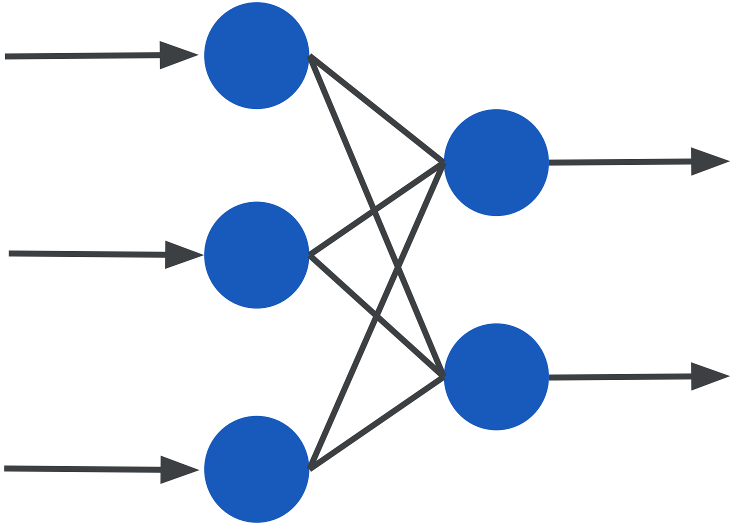
golfing

What is a **neural network**?



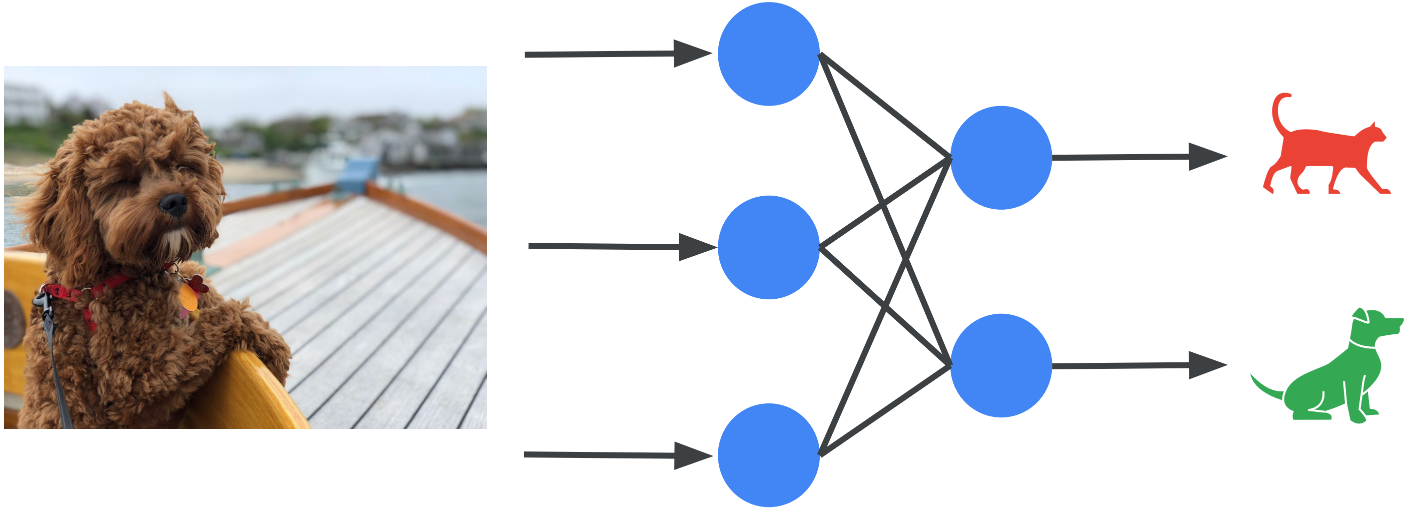
What is a **neural network**?





artificial

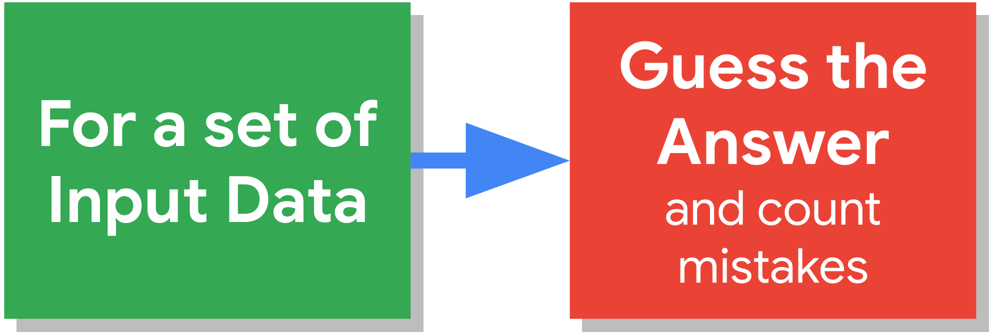
What is a **neural network**?



Training the machine 🏋️‍♂️

For a set of Input Data

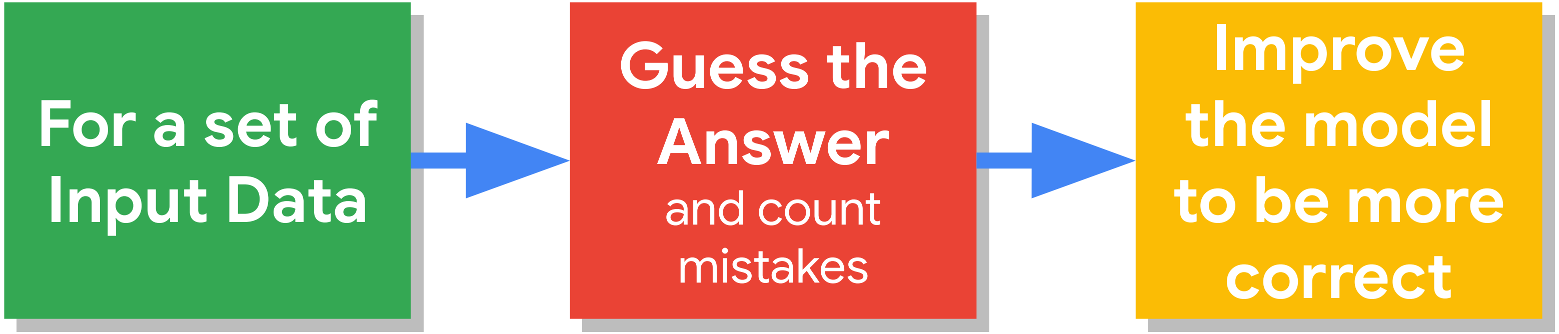
Training the machine



For a set of Input Data

Guess the Answer and count mistakes

Training the machine 🏋️

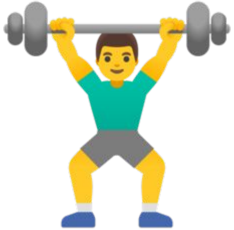


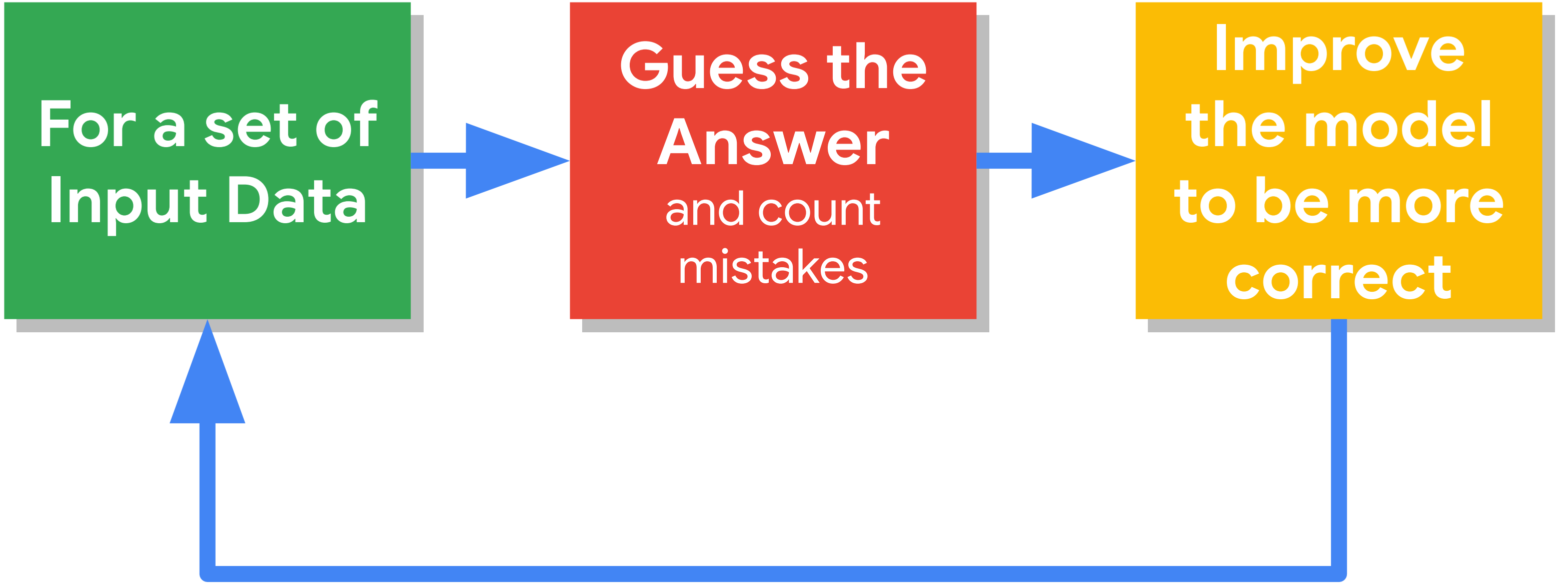
For a set of Input Data

Guess the Answer and count mistakes

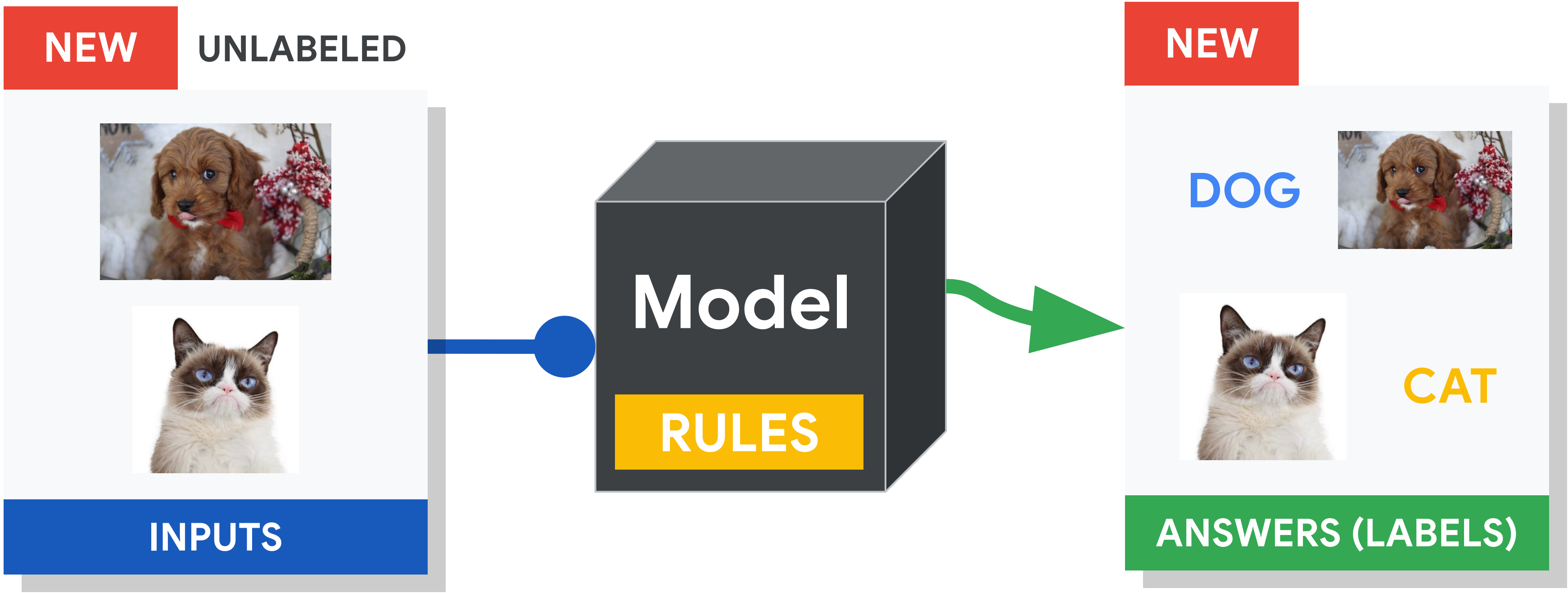
Improve the model to be more correct

Training the machine

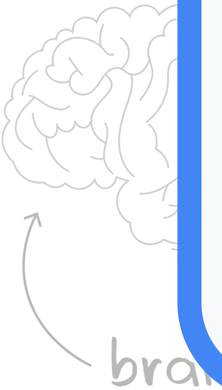



For a set of Input Data
Guess the Answer and count mistakes
Improve the model to be more correct

After it's **learned** use it for **inference**:



What is a **neural network**?



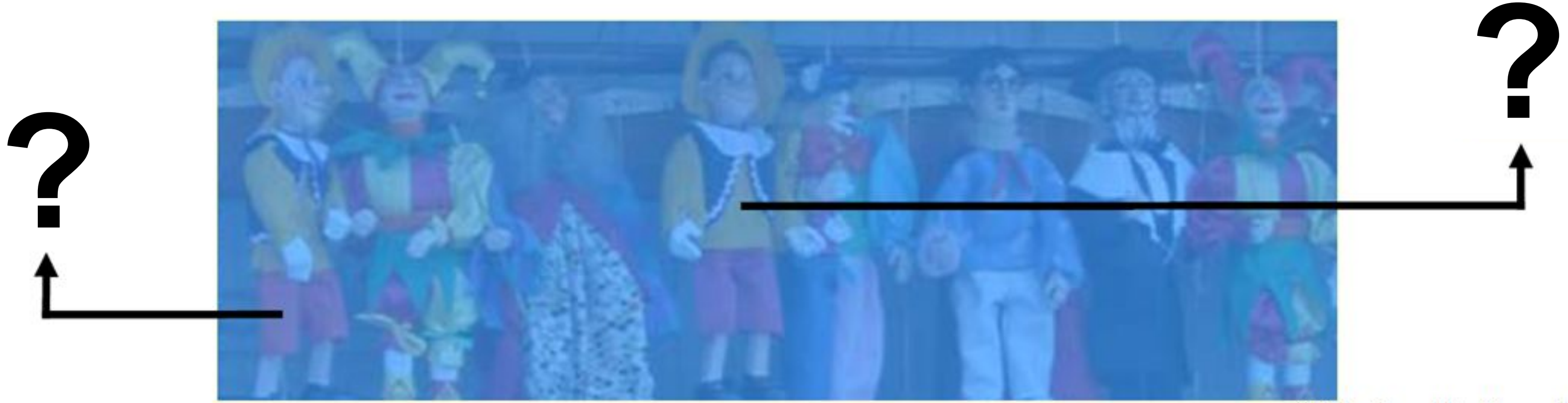
To learn more about the **math behind neural network training** there is a nice series of videos here: [3Blue1Brown Neural Networks Playlist]

artificial

Computer Vision is Hard

Computer Vision is Hard

What color are the pants and the shirt?

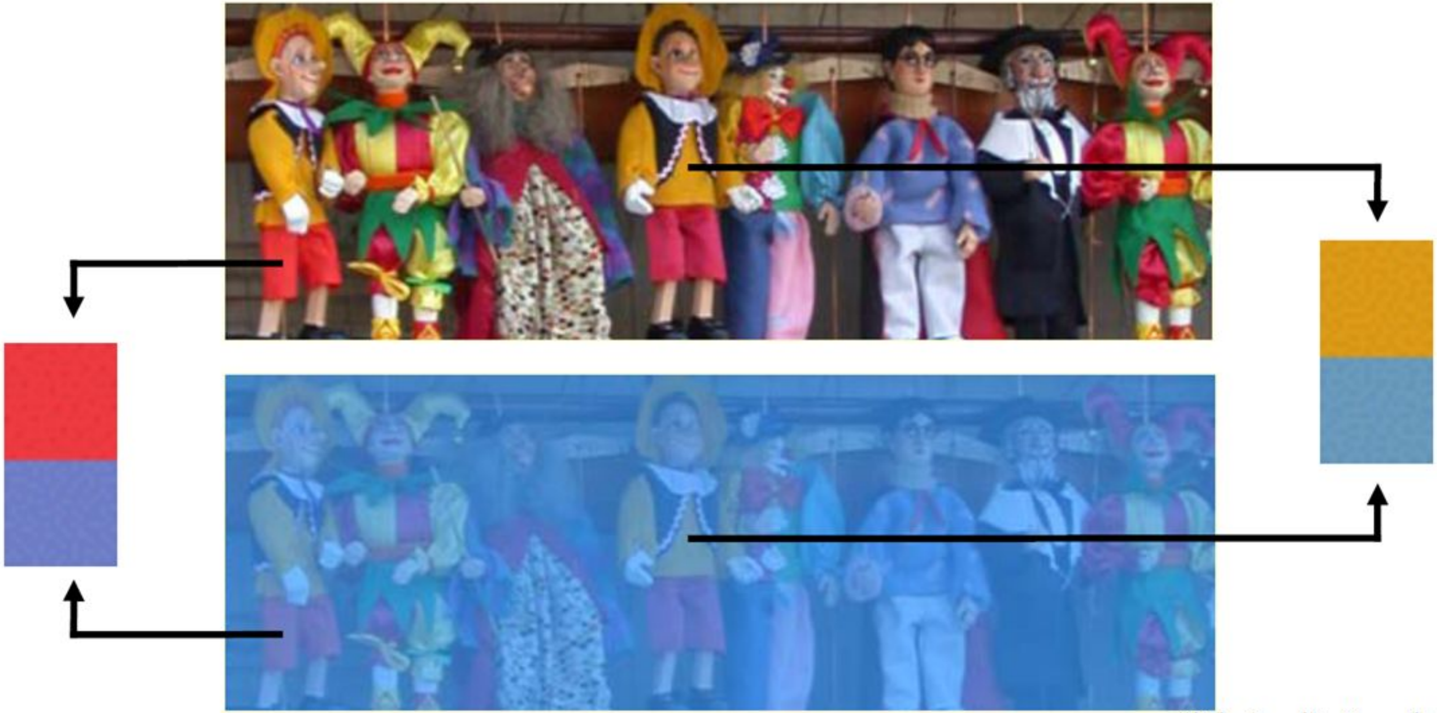


Slide Credit: Hamilton Chong

Computer Vision is Hard

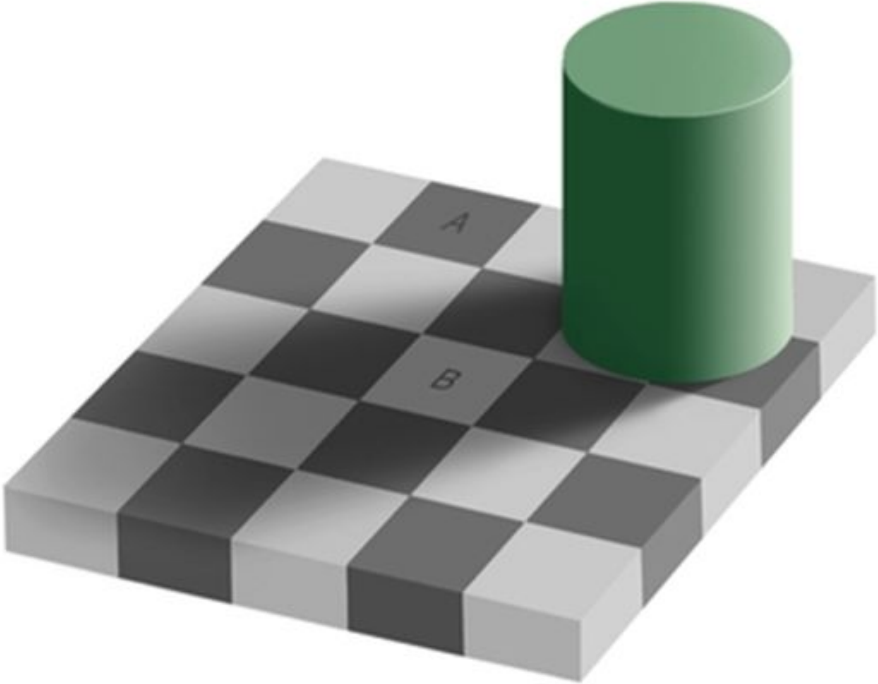
Slide Credit: Hamilton Chong

Computer Vision is Hard



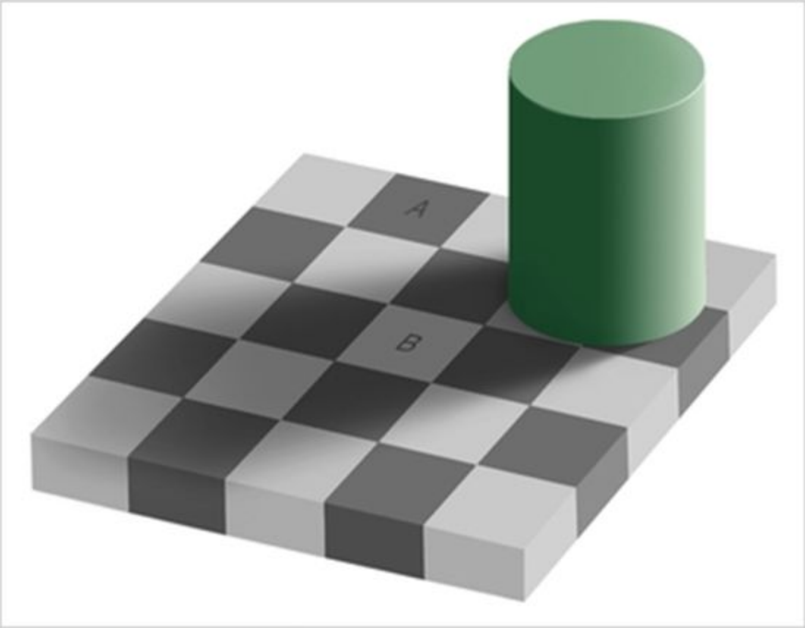
Slide Credit: Hamilton Chong

Computer Vision is Hard

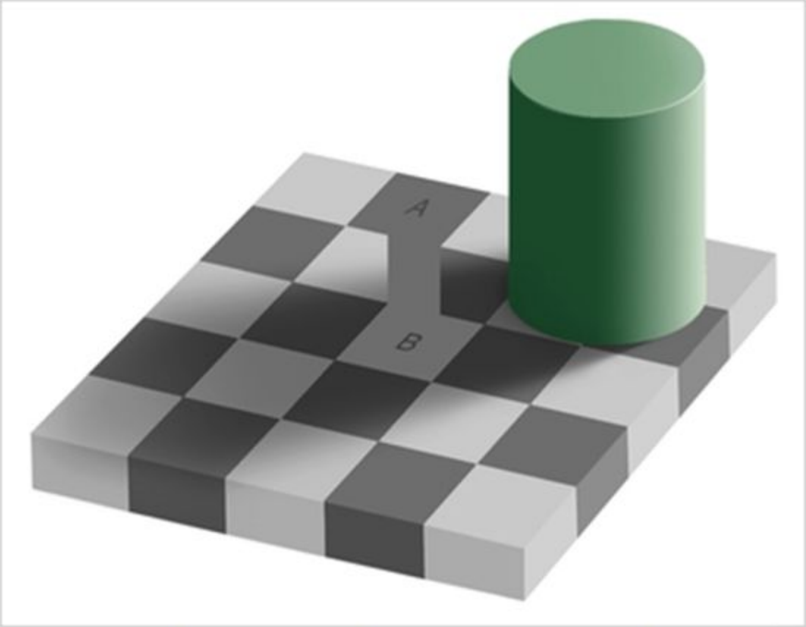


Is square A or B darker in color?

Computer Vision is Hard

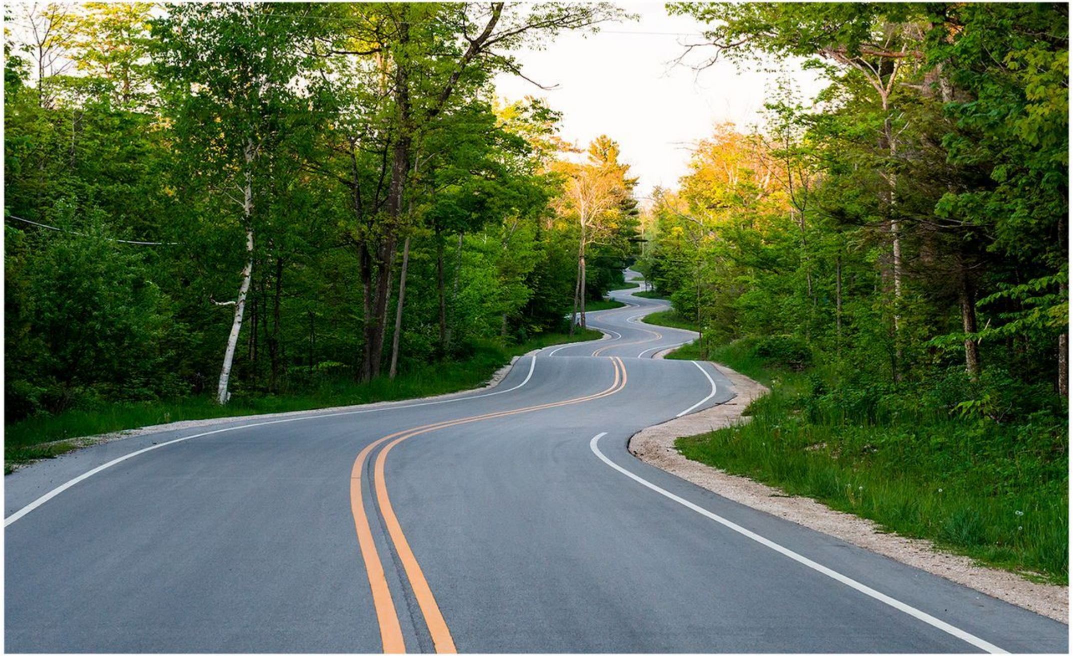


Areas of the image A and B are the same color

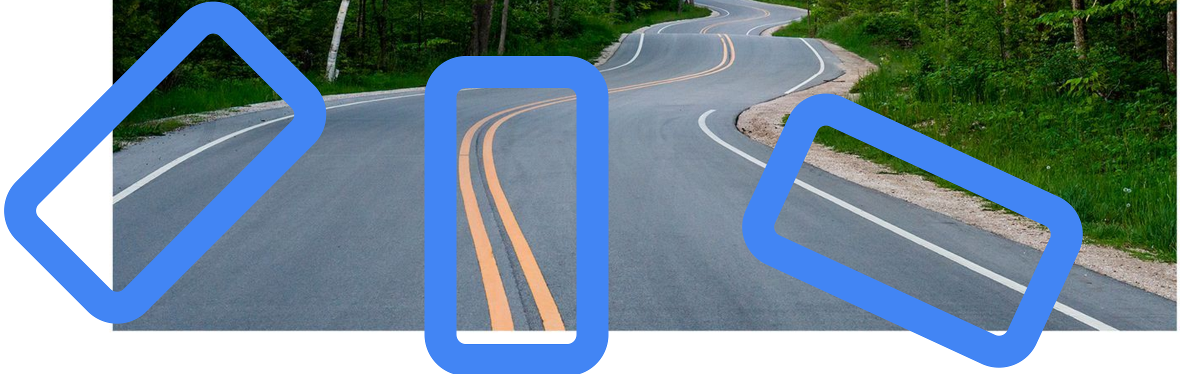


A rectangle of the same color has been drawn connecting the two areas of the image

What **Features** of the image might be important for self driving cars?



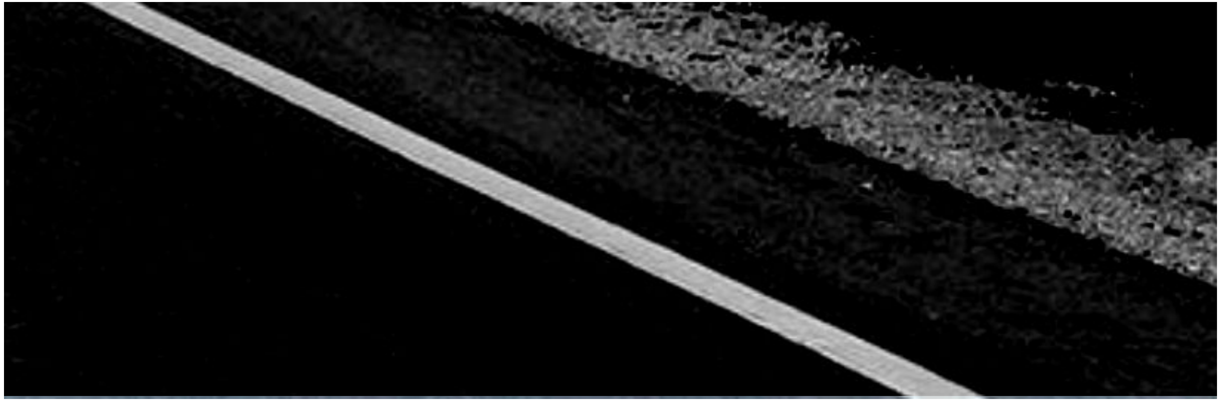
What **Features** of the image might be important for self driving cars?



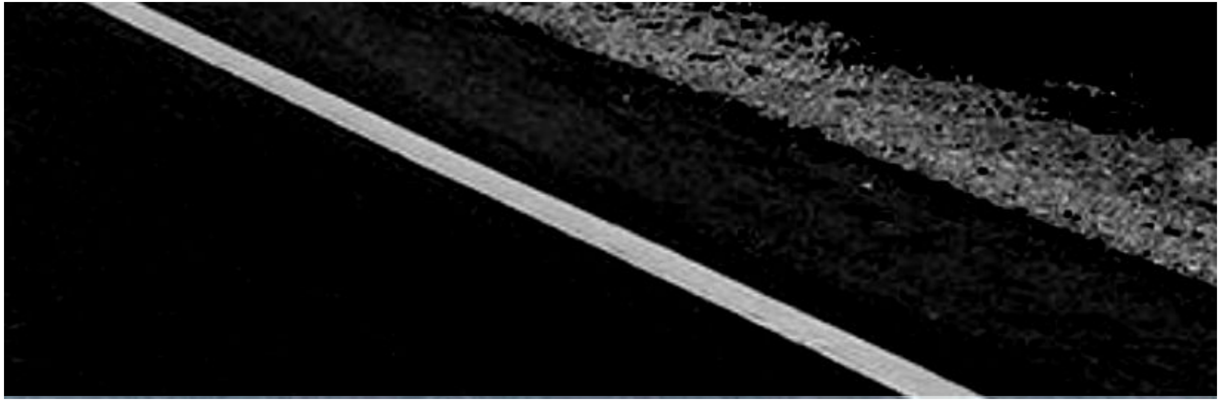
Maybe straight lines to see the lanes of the road?

How might we find these features?

How might we find these features?

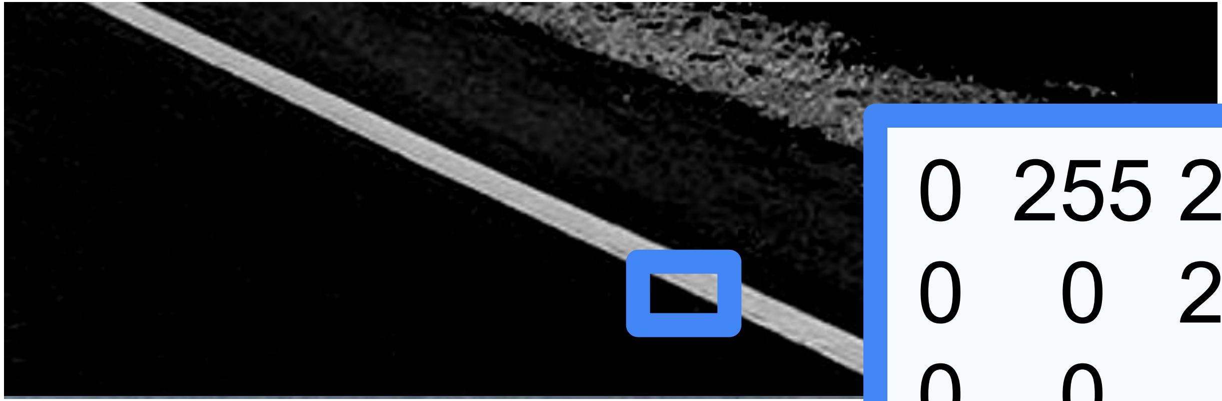


How might we find these features?



Black: 0
White: 255

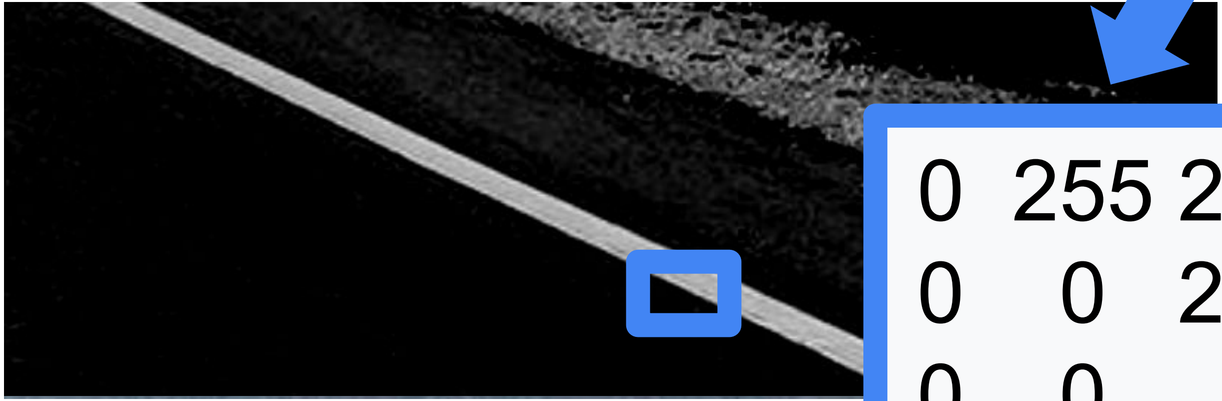
How might we find these features?



0	255	255	255	255
0	0	255	200	255
0	0	0	255	255
0	0	0	0	255
0	0	0	0	0

Black: 0
White: 255

How might we find these features?



Look for a Big Change!

0	255	255	255	255
0	0	255	200	255
0	0	0	255	255
0	0	0	0	255
0	0	0	0	0

Black: 0
White: 255

How might we find these features?

Convolutions

How might we find these features?
Convolutions

Original Image

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

How might we find these features?
Convolutions

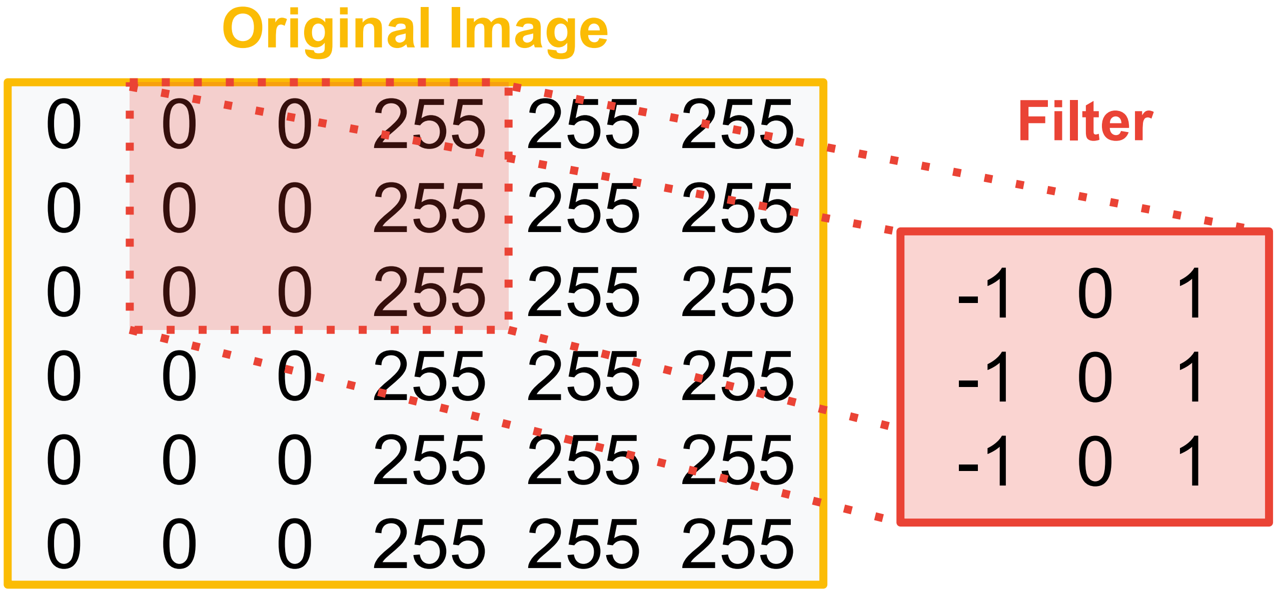
Original Image

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

Filter

-1	0	1
-1	0	1
-1	0	1

How might we find these features?
Convolutions



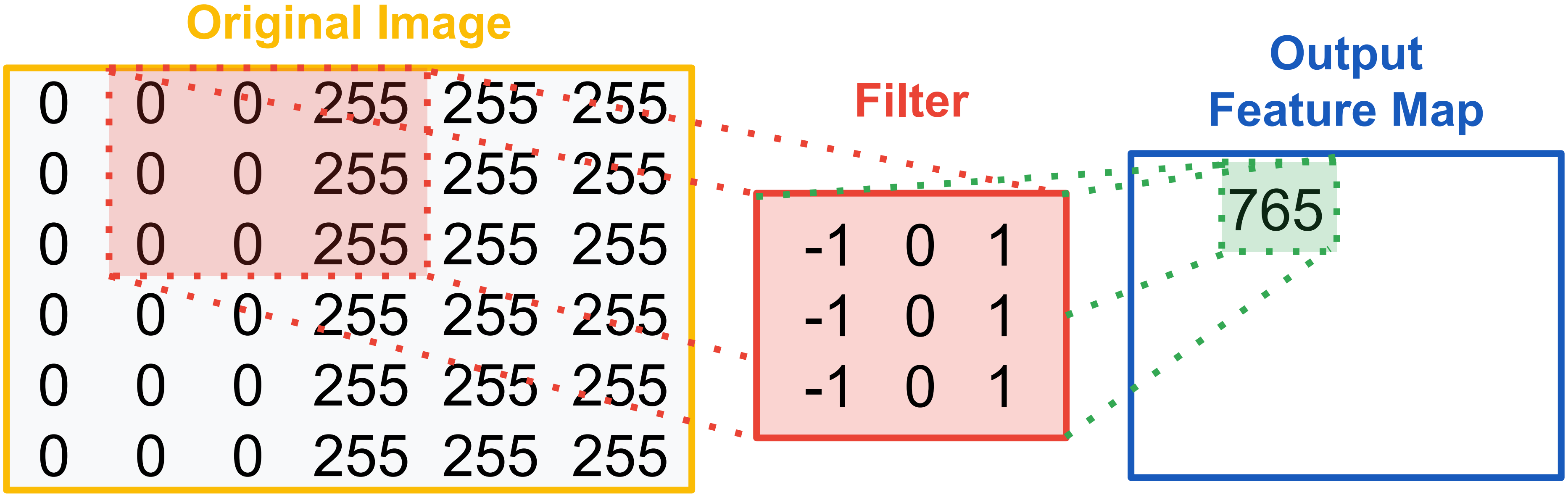
Original Image

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

Filter

-1	0	1
-1	0	1
-1	0	1

How might we find these features?
Convolutions



Original Image

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

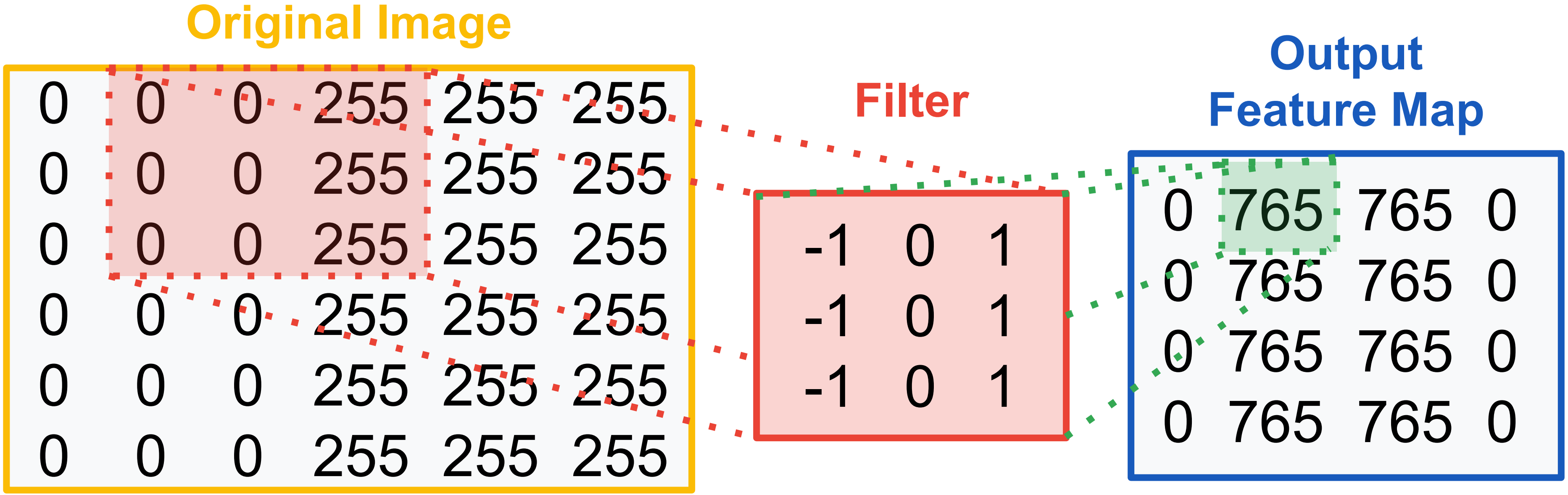
Filter

-1	0	1
-1	0	1
-1	0	1

Output Feature Map

765

How might we find these features?
Convolutions



Original Image

0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255
0	0	0	255	255	255

Filter

-1	0	1
-1	0	1
-1	0	1

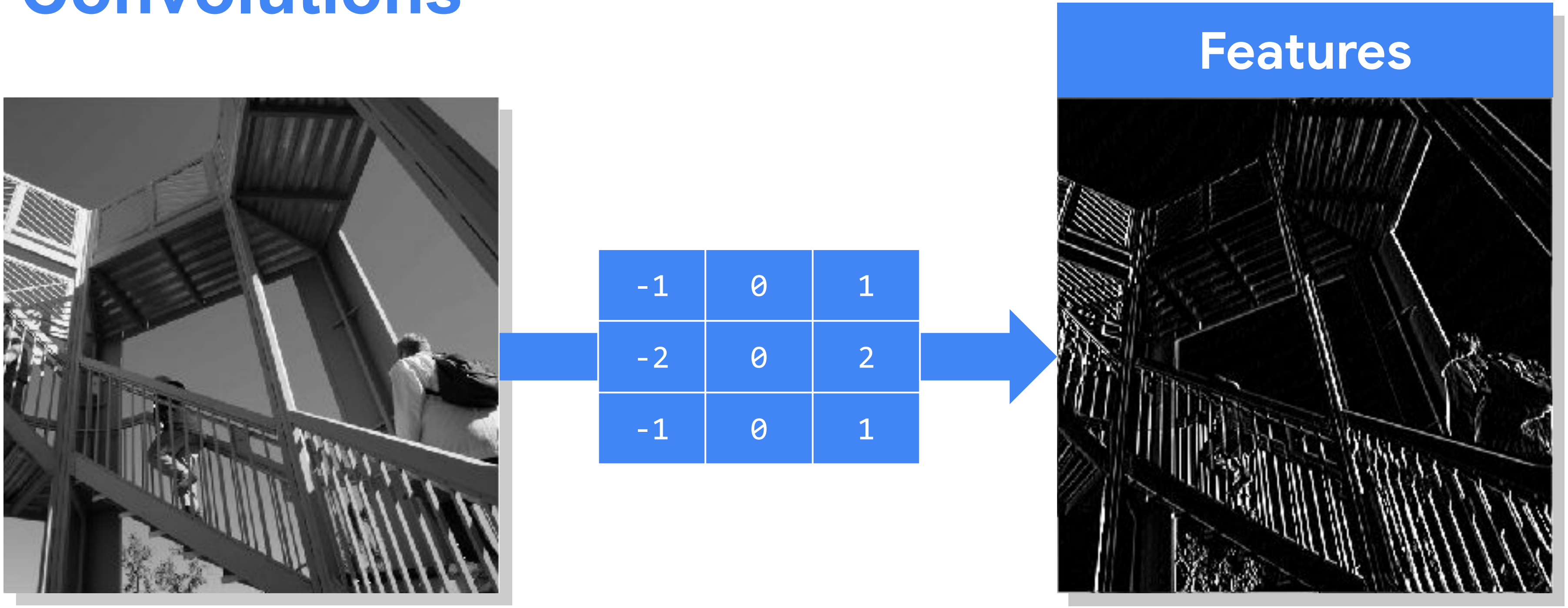
Output Feature Map

0	765	765	0
0	765	765	0
0	765	765	0
0	765	765	0

Colab Link

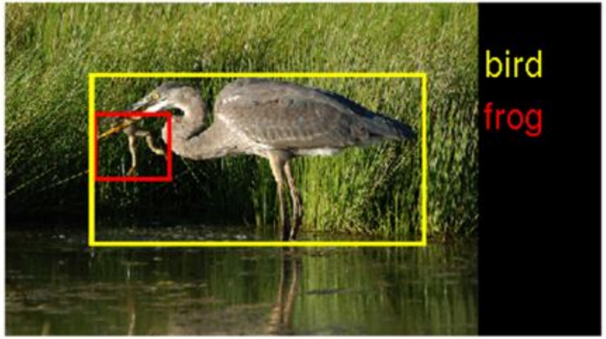
How might we find these features?

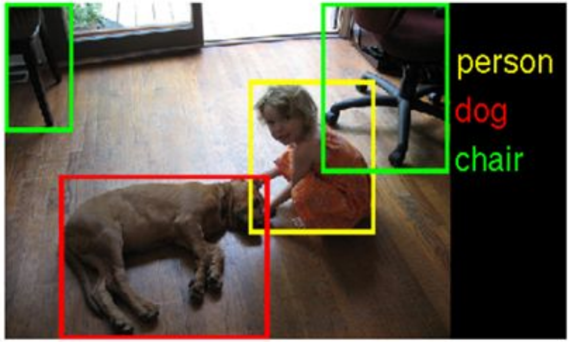
Convolutions

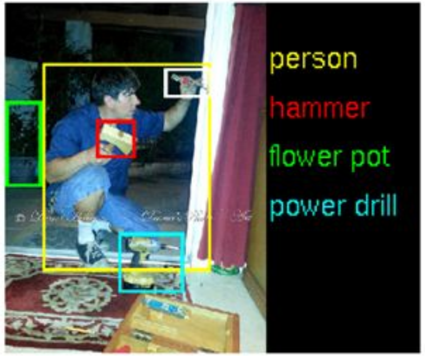


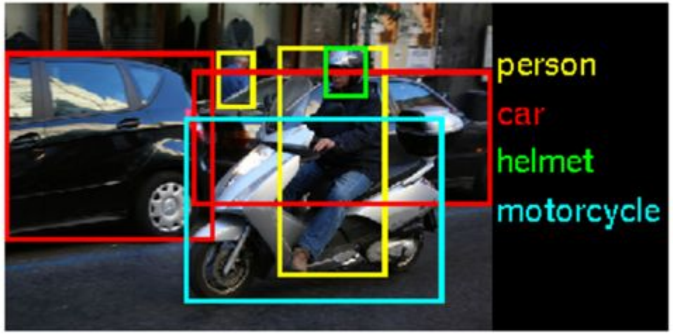
-1	0	1
-2	0	2
-1	0	1

What features are needed for Object Detection?

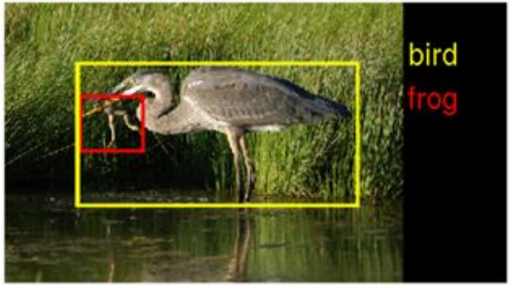


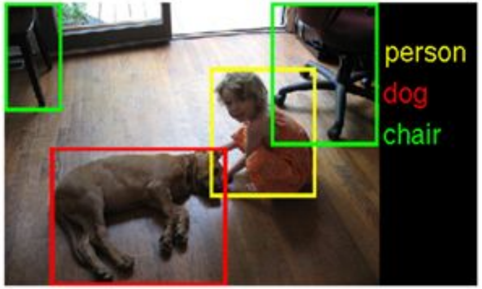


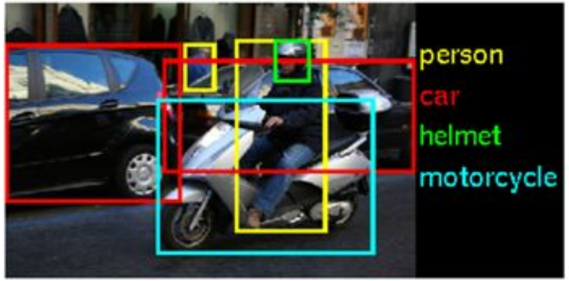




What features are needed for Object Detection?

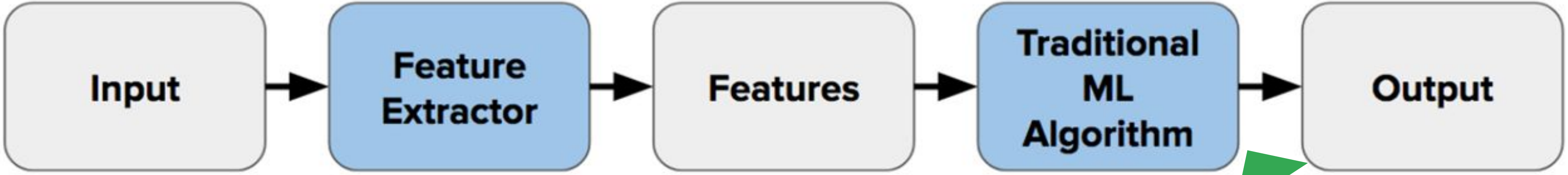






The ImageNet Challenge provided 1.2 million examples of 1,000 **labeled** items and challenged algorithms to learn from the data and then was tested on another 100,000 images

What features are needed for Object Detection?

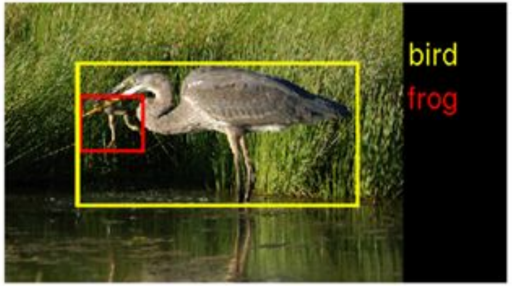


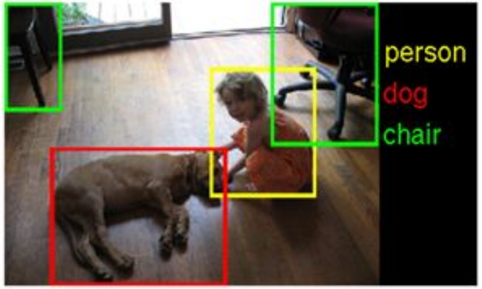
Traditional Machine Learning Flow

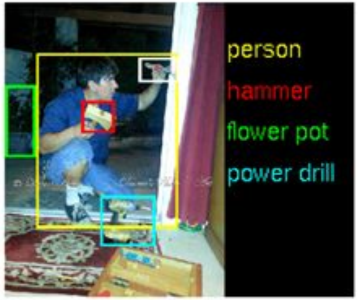
Regression, Clustering, etc.

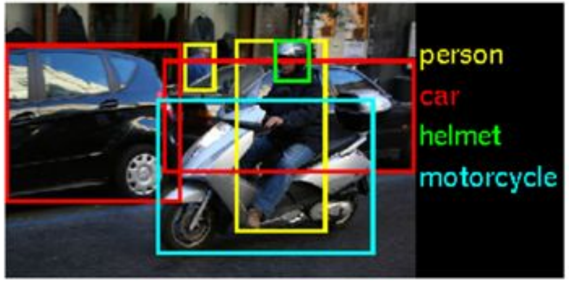
Vertical Lines, Horizontal Lines, Changes in Color, Changes in Focus, etc.

What features are needed for Object Detection?





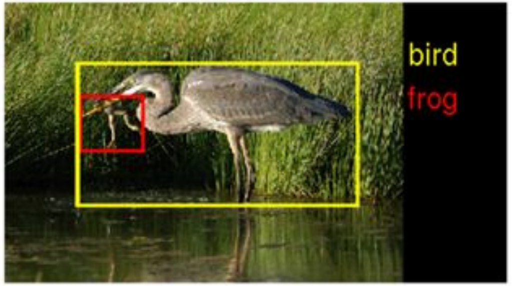


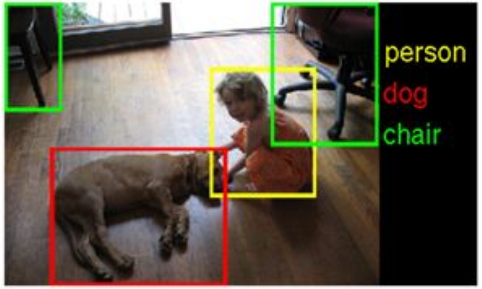


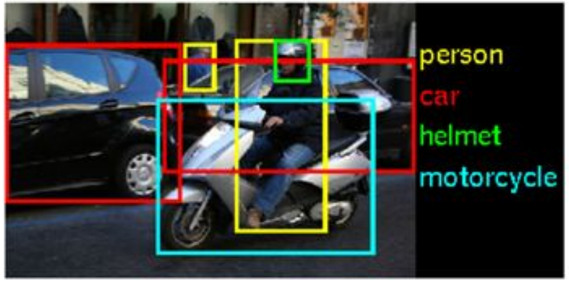
In 2010 teams had
75-50% error

In 2011 teams had
75-25% error

What features are needed for Object Detection?

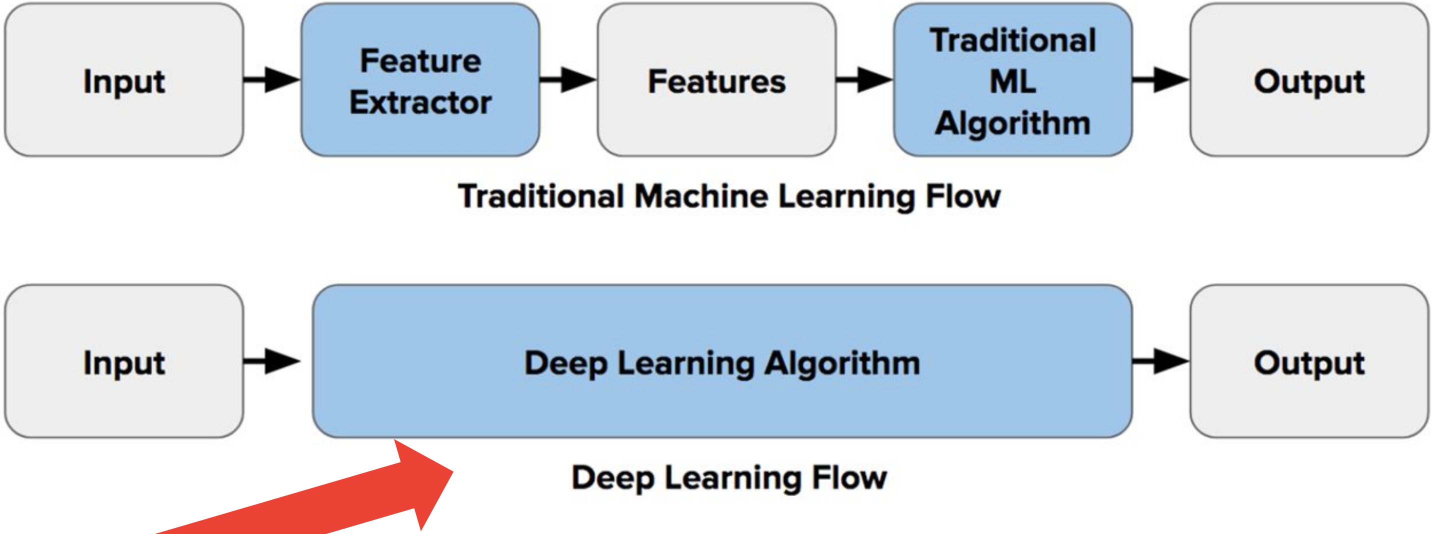






In 2012 still no team had less than 25% error barrier except **AlexNet at 15%**

What features are needed for Object Detection?



Traditional Machine Learning Flow

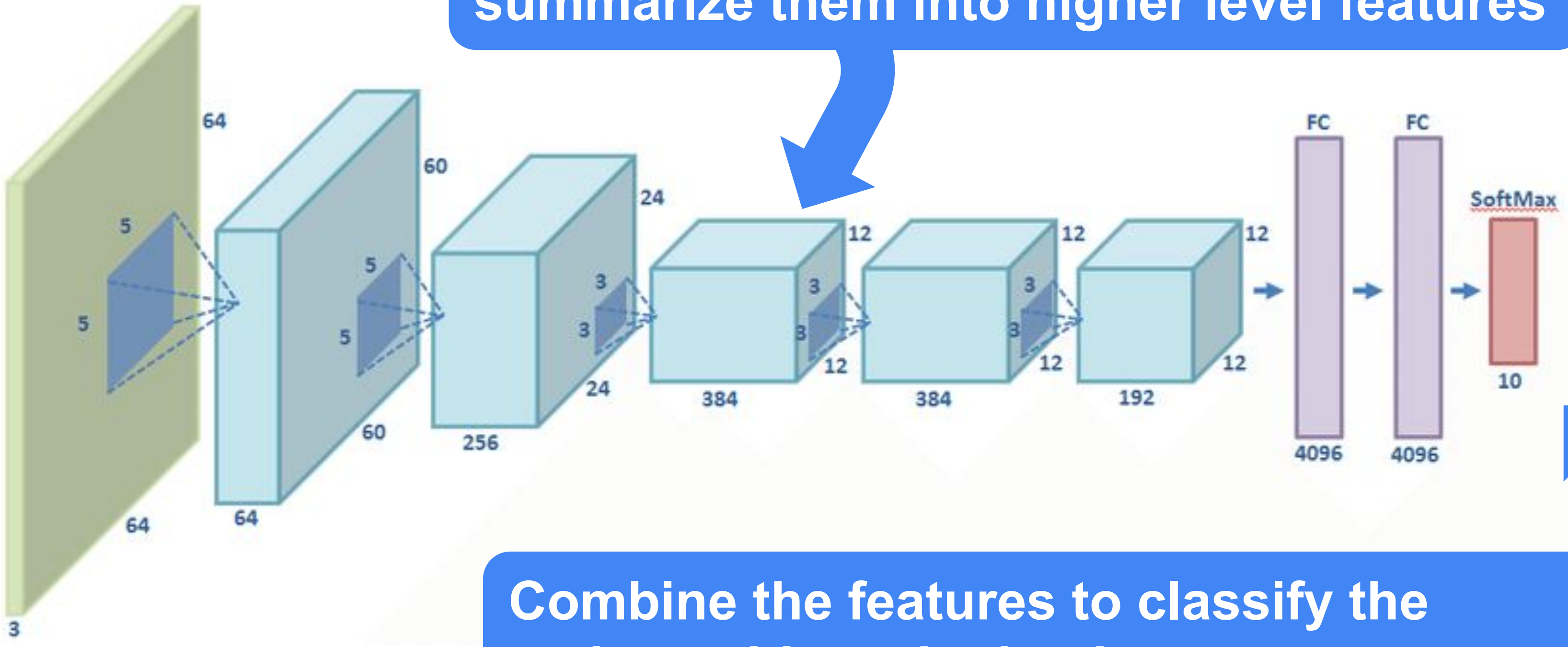
Deep Learning Flow

Let the computer figure out its own features and how to combine them!

AlexNet Paper

AlexNet

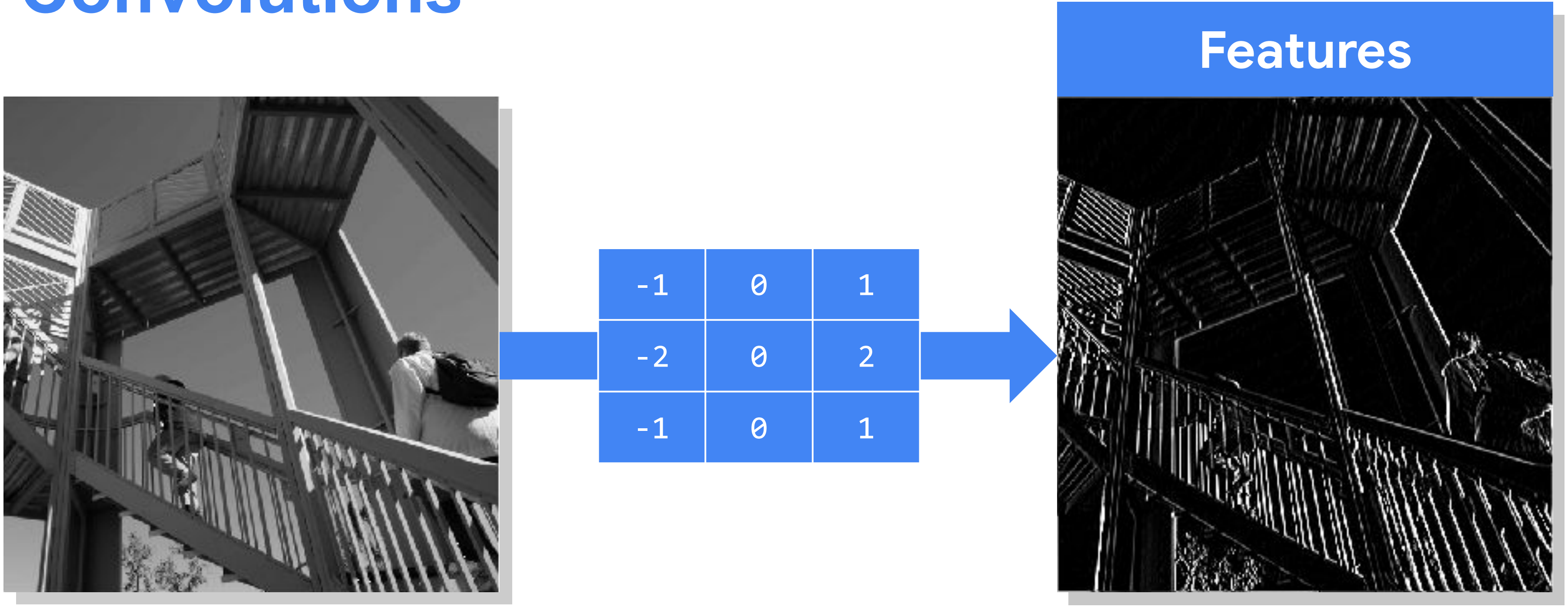
Use convolutions to find features and the summarize them into higher level features



Combine the features to classify the various objects in the dataset

Colab Link

How might we find these features?
Convolutions



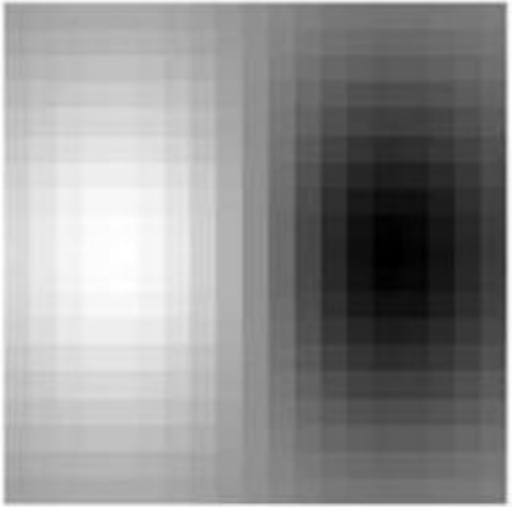
Features

-1	0	1
-2	0	2
-1	0	1

How might we find these features?

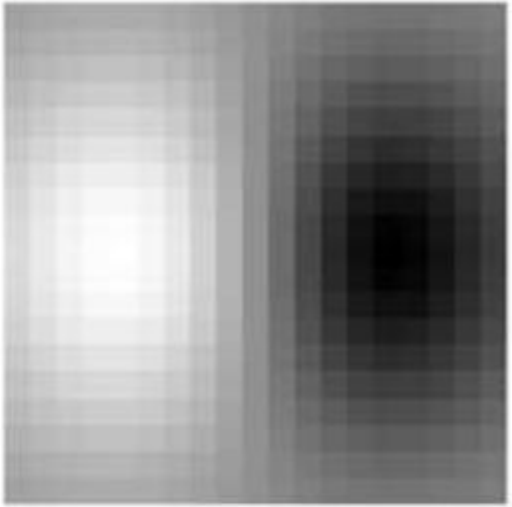
Convolutions

-1	0	1
-2	0	2
-1	0	1



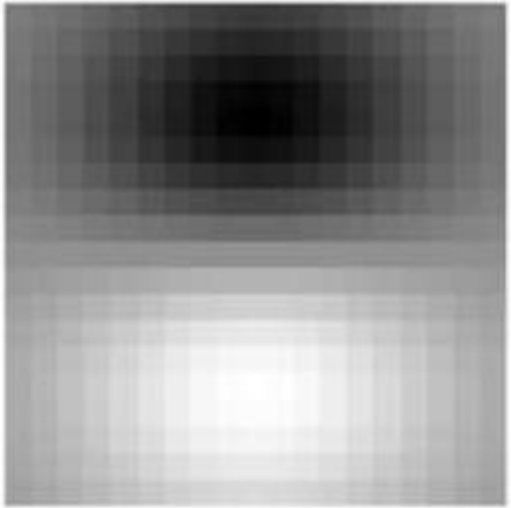
How might we find these features?

Convolutions



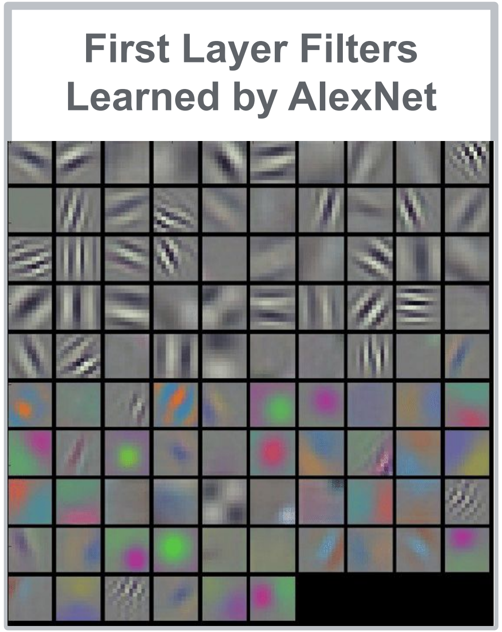
How might we find these features?

Convolutions

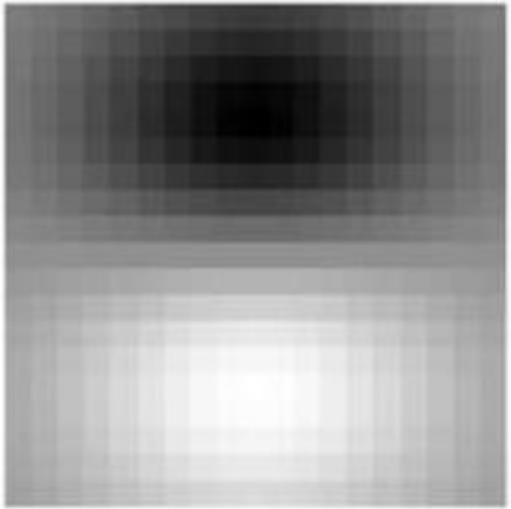


How might we find these features?

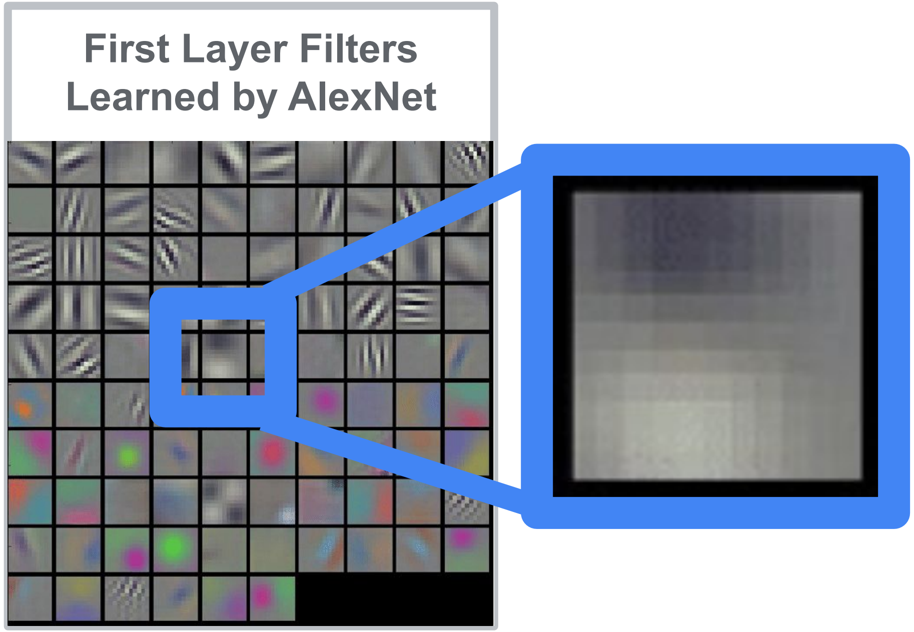
Convolutions

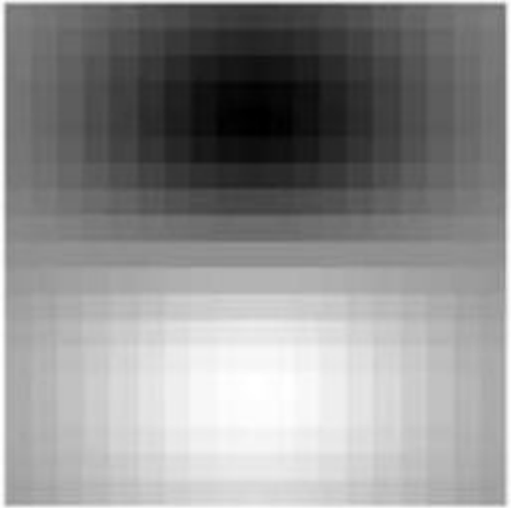


First Layer Filters
Learned by AlexNet



How might we find these features?
Convolutions

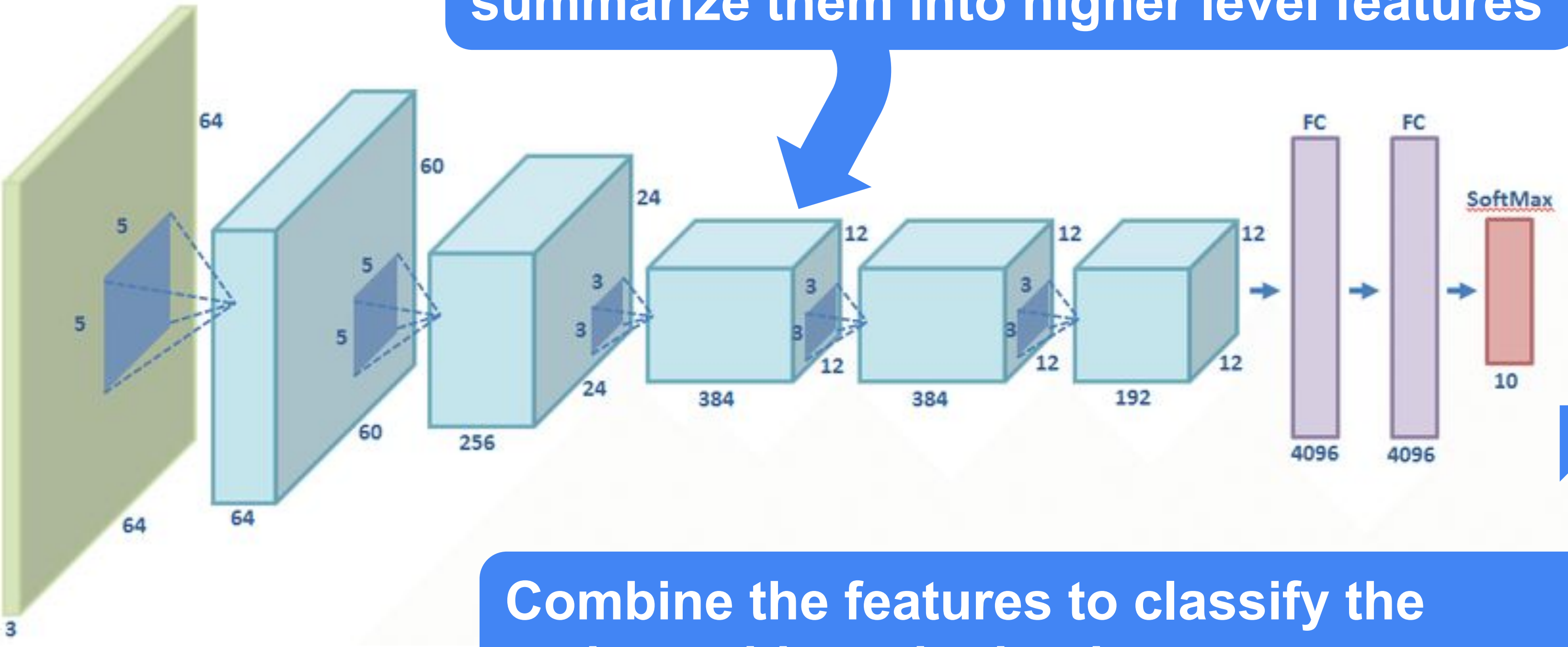

First Layer Filters
Learned by AlexNet



AlexNet Paper

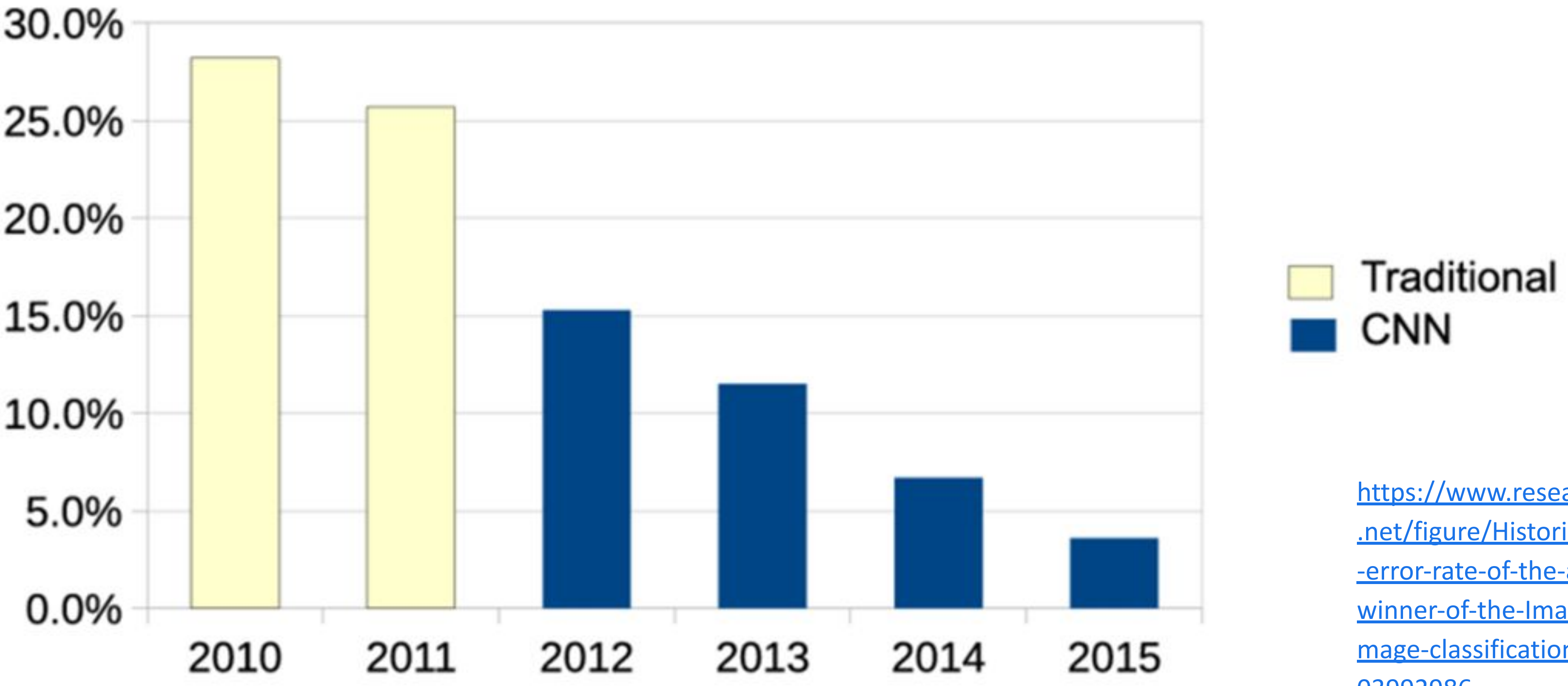
AlexNet

Use convolutions to find features and the summarize them into higher level features



Combine the features to classify the various objects in the dataset

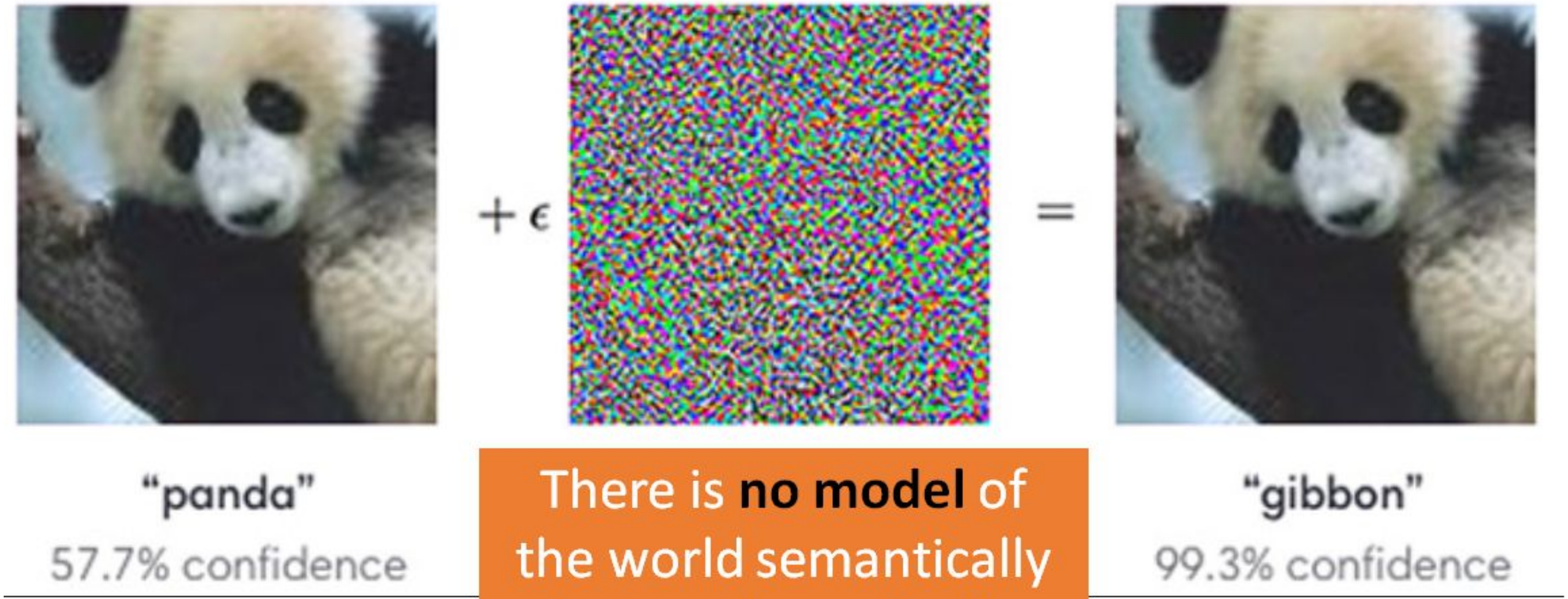
What features are needed for Object Detection?



https://www.researchgate.net/figure/Historical-top5-error-rate-of-the-annual-winner-of-the-ImageNet-image-classification_fig7_303992986

A word of caution...

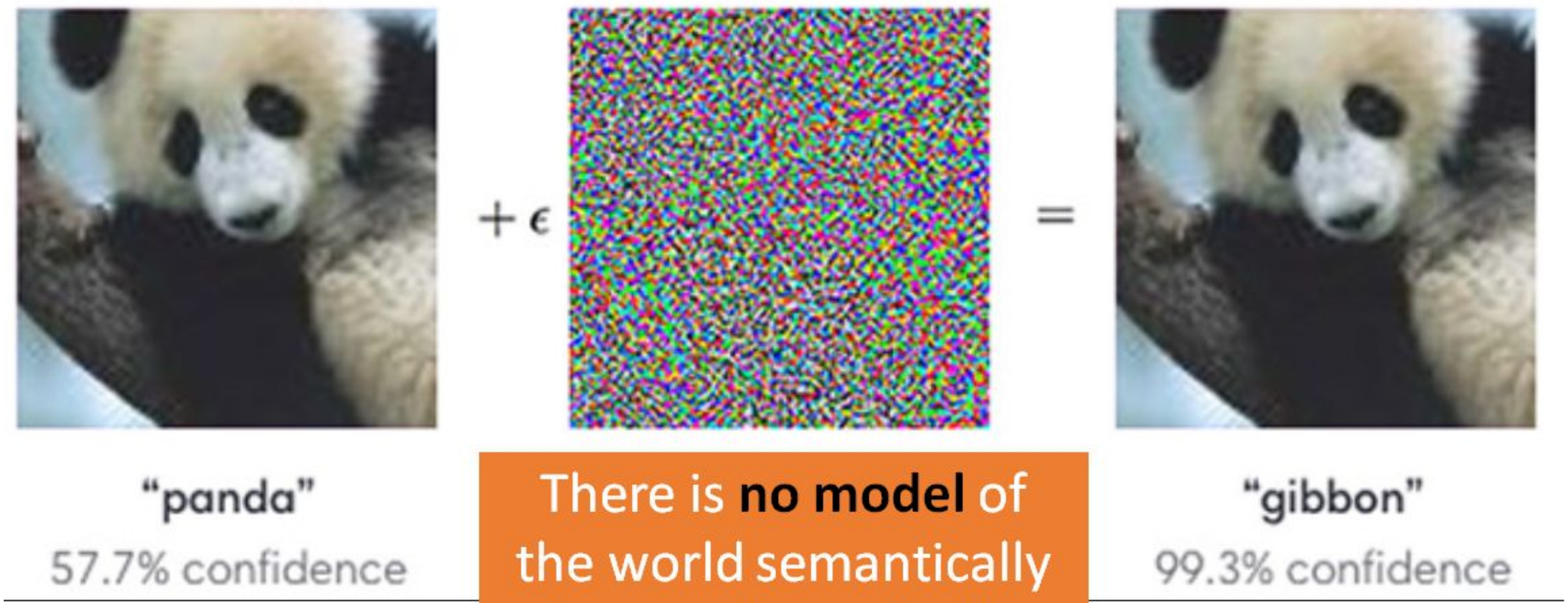
Ackerman “Hacking the Brain With Adversarial Images”



There is **no model** of the world semantically just mathematically https://www.vox.com/future-perfect/2019/4/8/18297410/ai-tesla-self-driving-cars-adversarial-machine-learning

A word of caution...

Ackerman “Hacking the Brain With Adversarial Images”



There is **no model** of the world semantically just mathematically

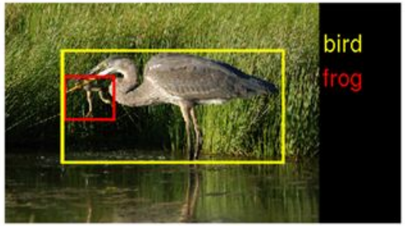
Today’s Agenda

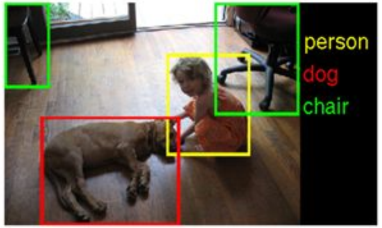
- Introduction to Computer Vision
- **Hands-on Computer Vision: Thing Translator**
- Building an Object Detection Dataset
- Training our Model using Transfer Learning
- Deploying our Model onto our Arduino
- Summary

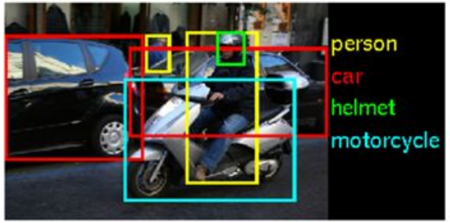
The **Thing Translator**

Open On Your Phone

https://thing-translator.appspot.com/





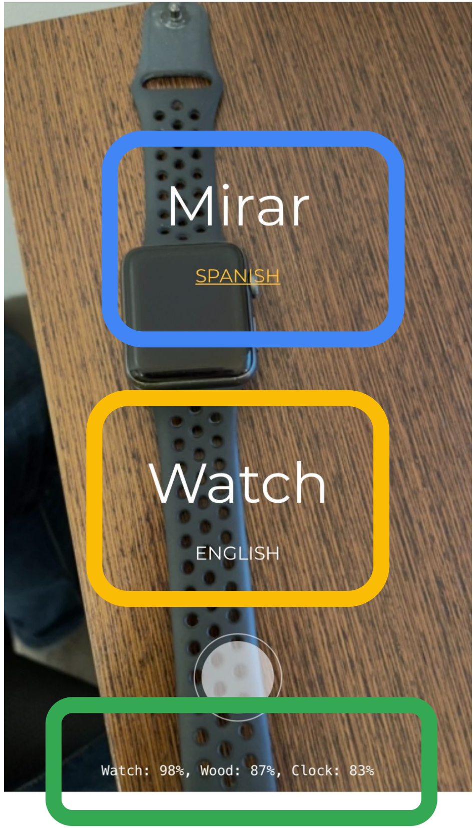


+

The **Thing Translator**

https://thing-translator.appspot.com/

Open On Your Phone

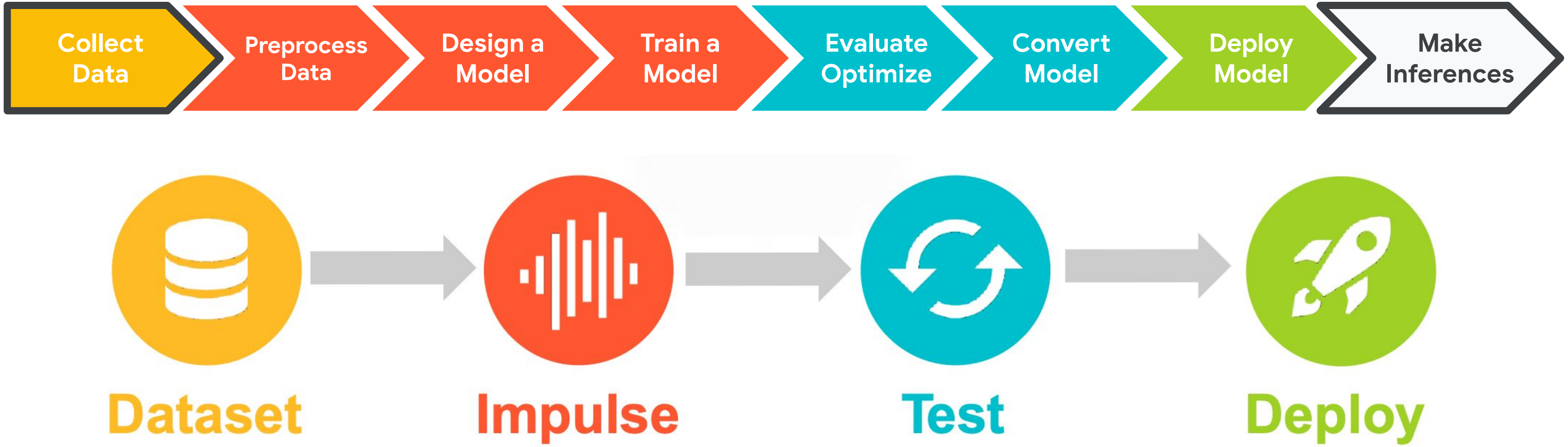


Watch: 98%, Wood: 87%, Clock: 83%

Today’s Agenda

- Introduction to Computer Vision
- Hands-on Computer Vision: Thing Translator
- **Building an Object Detection Dataset**
- Training our Model using Transfer Learning
- Deploying our Model onto our Arduino
- Summary

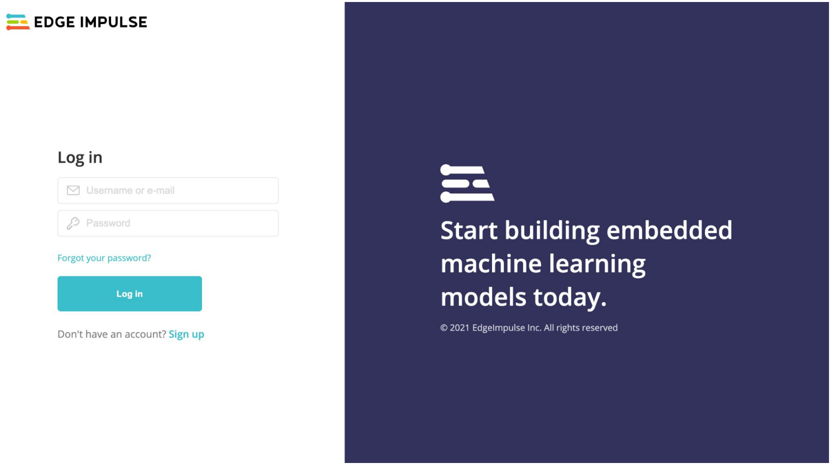
The TinyML Workflow using Edge Impulse



Create an Edge Impulse Account

1. Create an Edge Impulse account: https://studio.edgeimpulse.com/signup

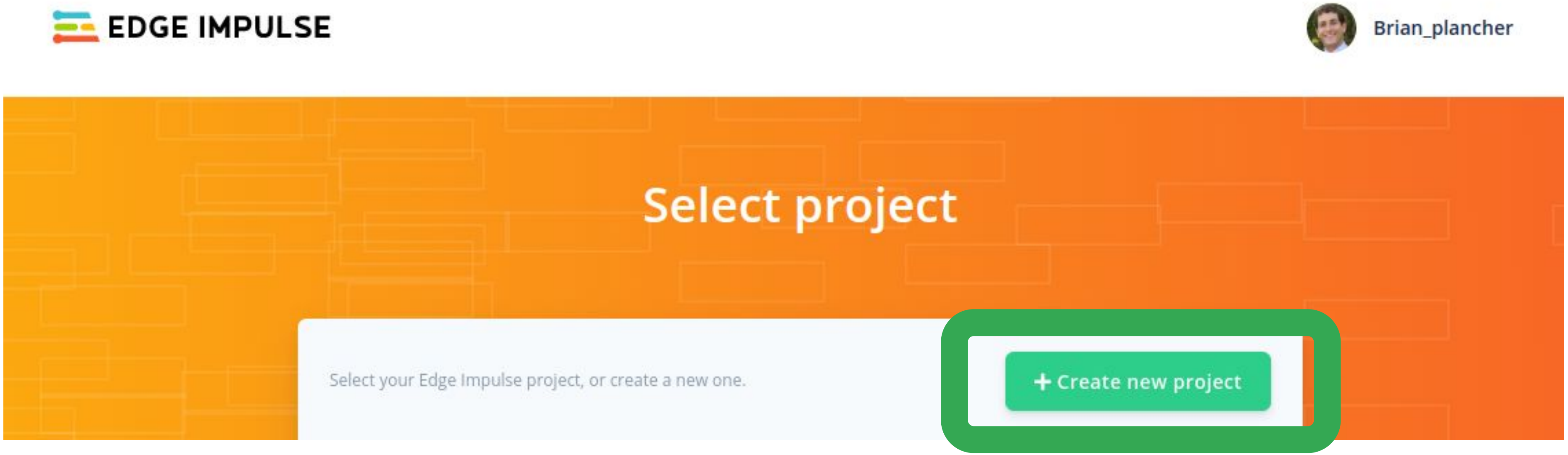
2. Validate your email by clicking the link in the email sent to your account's email address



Don't have an account? Sign up

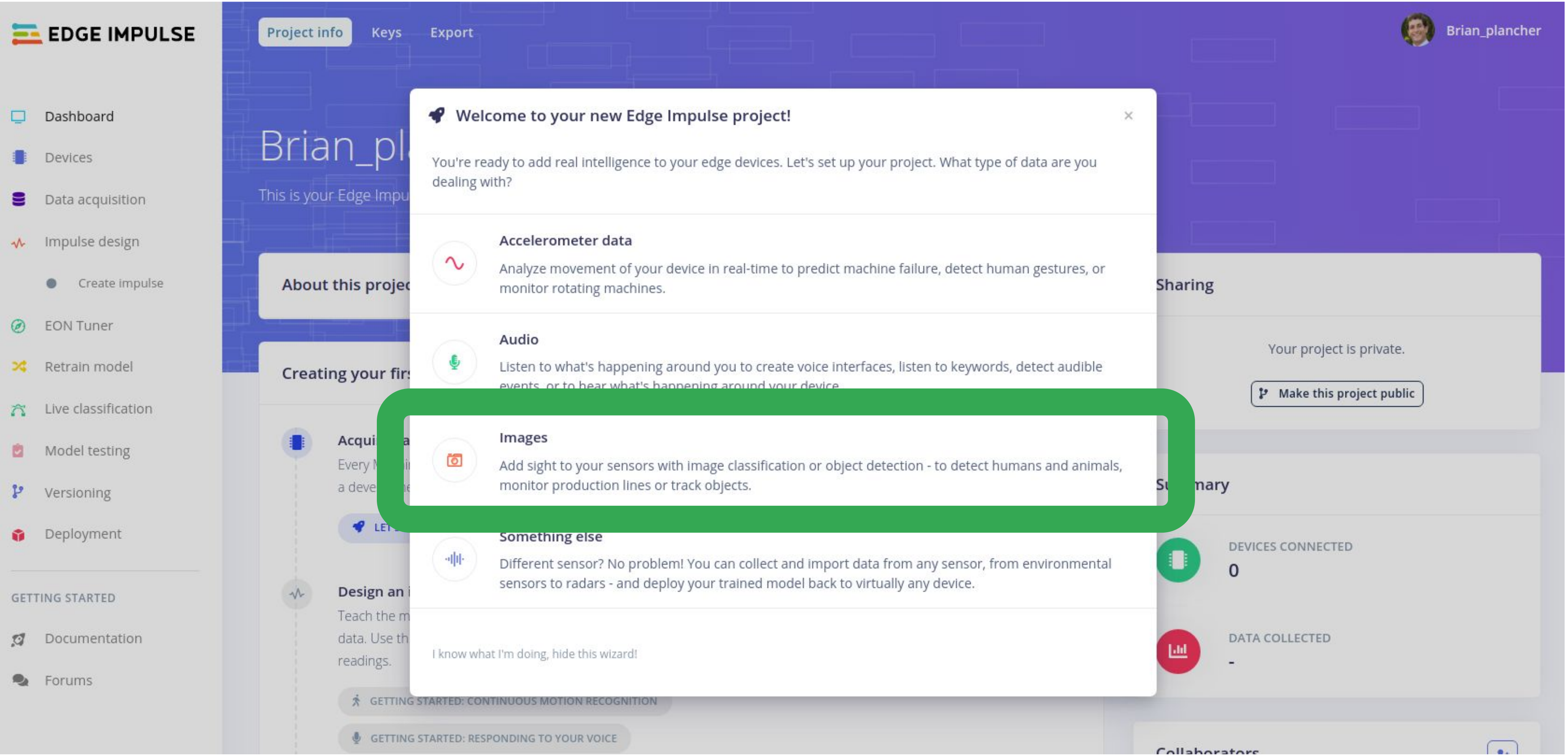
Start building embedded machine learning models today.

© 2021 EdgeImpulse Inc. All rights reserved



Select project

Select your Edge Impulse project, or create a new one.



Welcome to your new Edge Impulse project!

You're ready to add real intelligence to your edge devices. Let's set up your project. What type of data are you dealing with?

Accelerometer data
Analyze movement of your device in real-time to predict machine failure, detect human gestures, or monitor rotating machines.

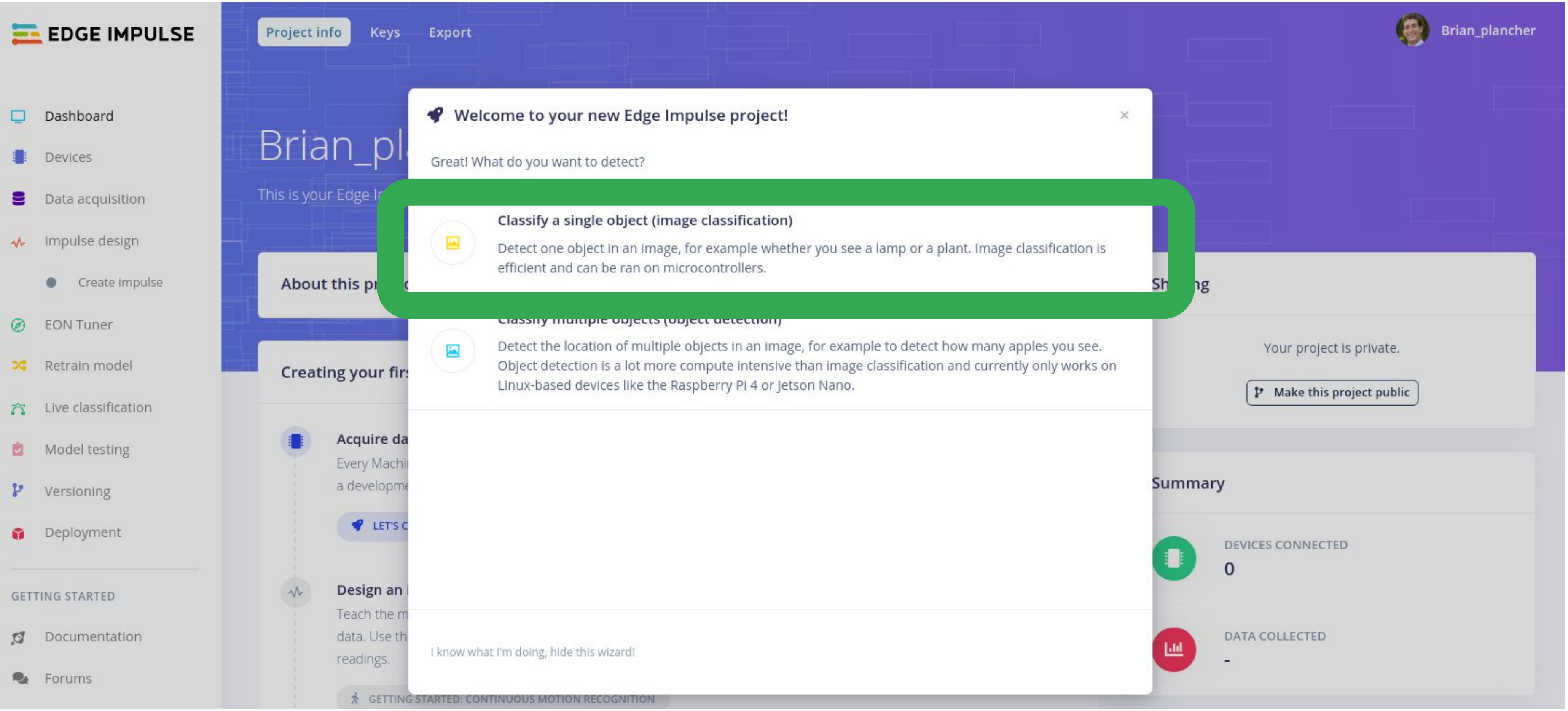
Audio
Listen to what's happening around you to create voice interfaces, listen to keywords, detect audible events, or to hear what's happening around your device.

Images
Add sight to your sensors with image classification or object detection - to detect humans and animals, monitor production lines or track objects.

Something else
Different sensor? No problem! You can collect and import data from any sensor, from environmental sensors to radars - and deploy your trained model back to virtually any device.

I know what I'm doing, hide this wizard!

Your project is private.



Welcome to your new Edge Impulse project!

Great! What do you want to detect?

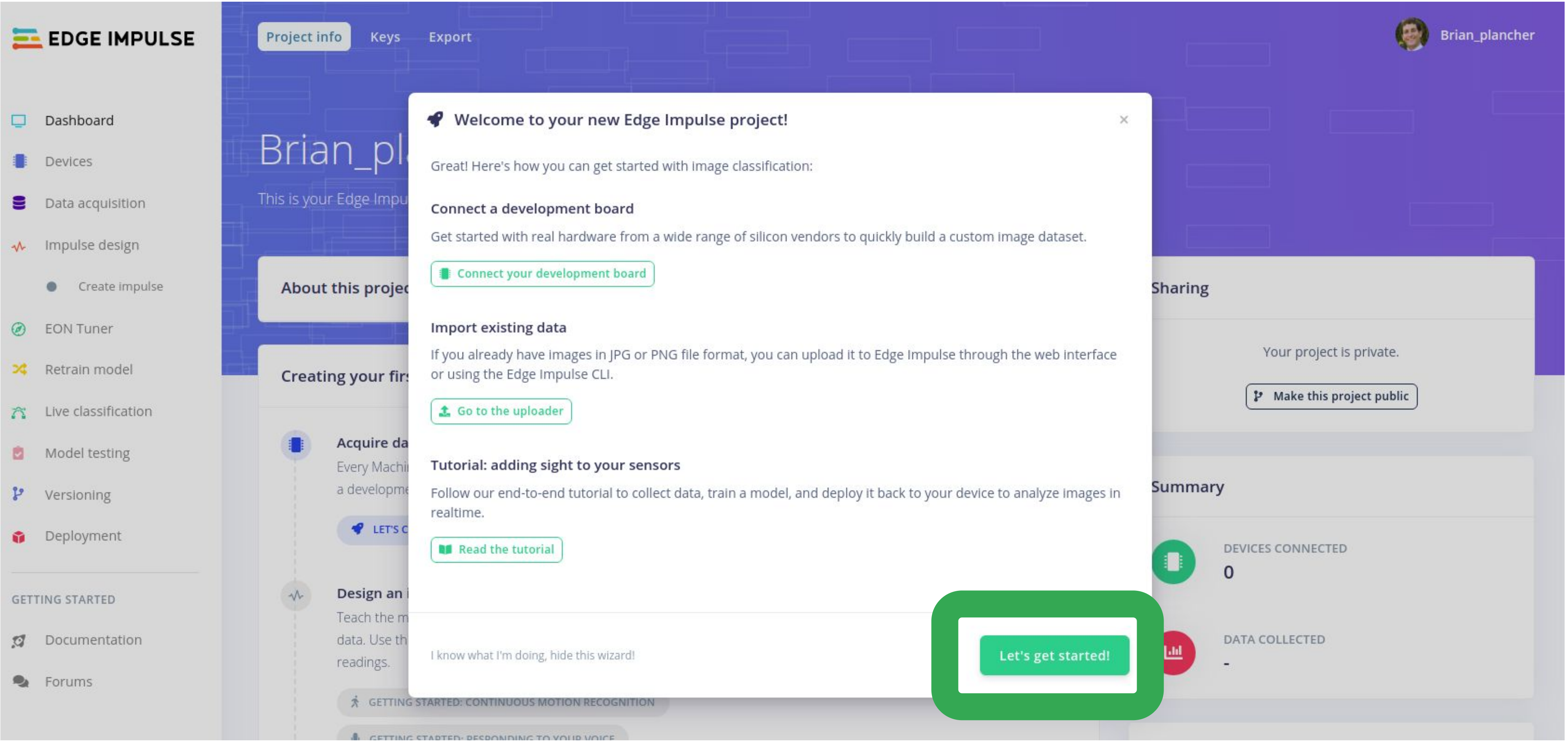
Classify a single object (image classification)

Detect one object in an image, for example whether you see a lamp or a plant. Image classification is efficient and can be ran on microcontrollers.

Classify multiple objects (object detection)

Detect the location of multiple objects in an image, for example to detect how many apples you see. Object detection is a lot more compute intensive than image classification and currently only works on Linux-based devices like the Raspberry Pi 4 or Jetson Nano.

I know what I'm doing, hide this wizard!



Welcome to your new Edge Impulse project!

Great! Here's how you can get started with image classification:

Connect a development board

Get started with real hardware from a wide range of silicon vendors to quickly build a custom image dataset.

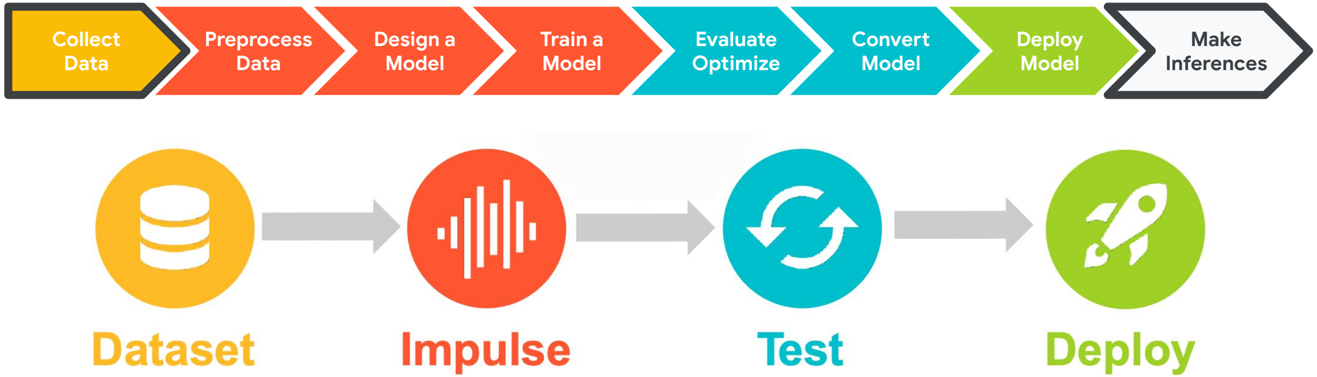
Import existing data

If you already have images in JPG or PNG file format, you can upload it to Edge Impulse through the web interface or using the Edge Impulse CLI.

Tutorial: adding sight to your sensors

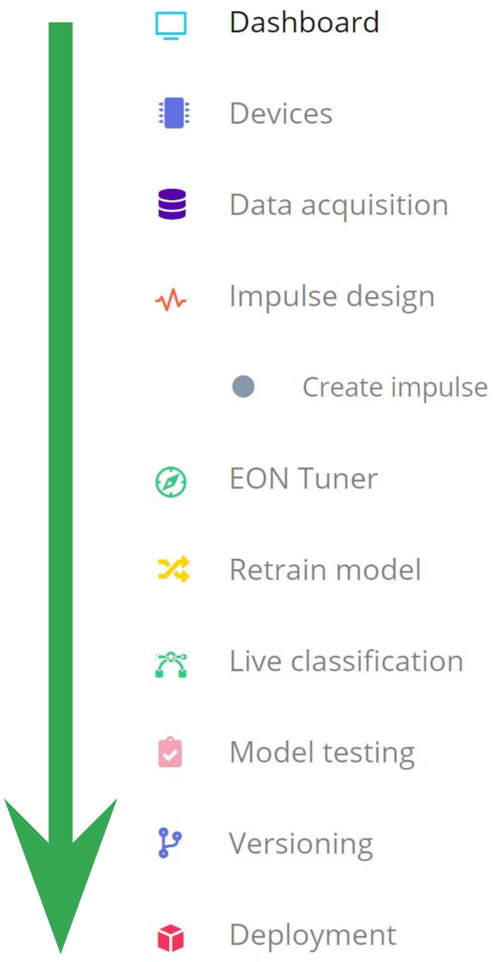
Follow our end-to-end tutorial to collect data, train a model, and deploy it back to your device to analyze images in realtime.

Edge Impulse Project Dashboard



Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences

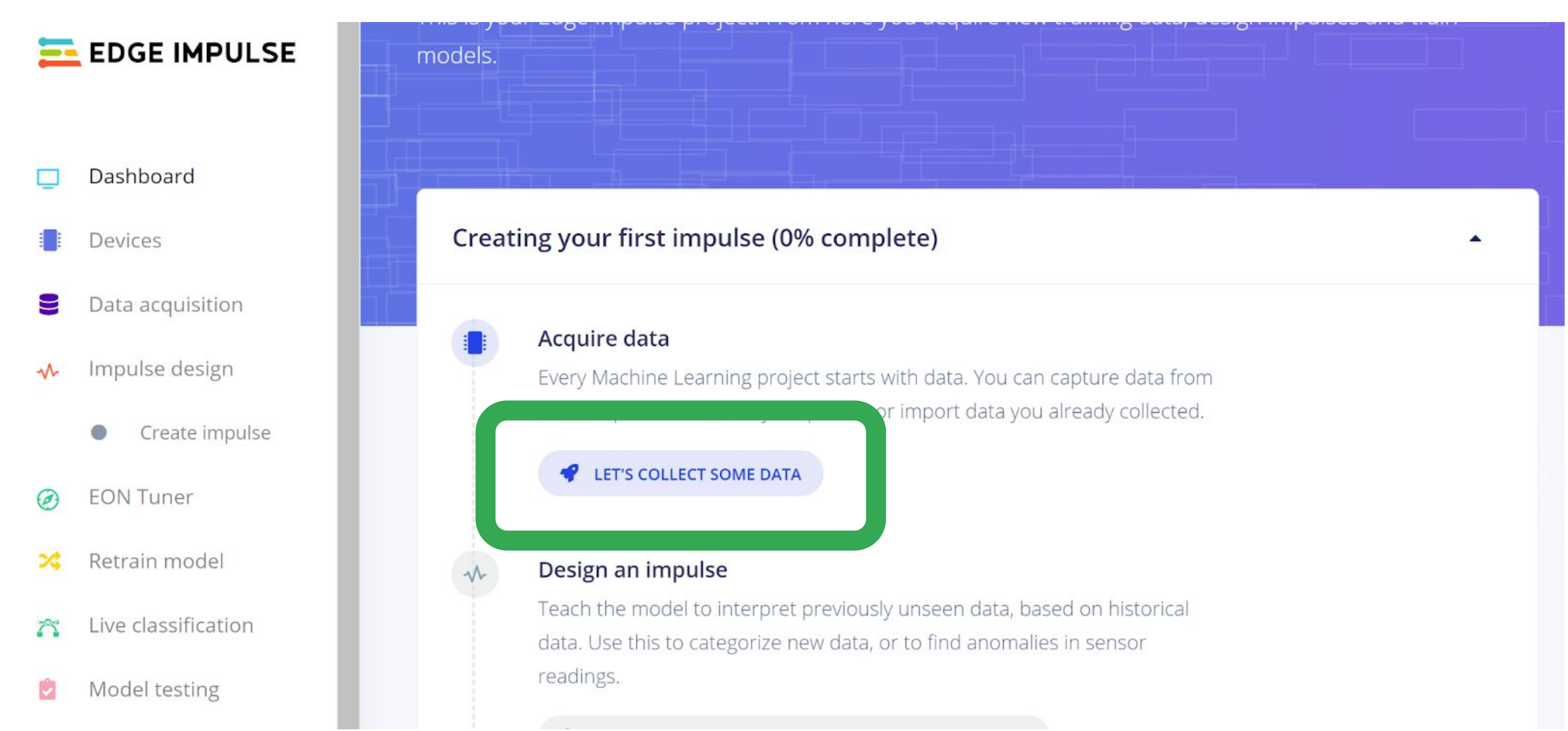
Dataset
Impulse
Test
Deploy



Activity: Create an Object Classification Dataset

Collect **~30 samples each** of the following classes of data:

- **Target Object #1**
- **Target Object #2**
- **(Optional) Target Object #3**



This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

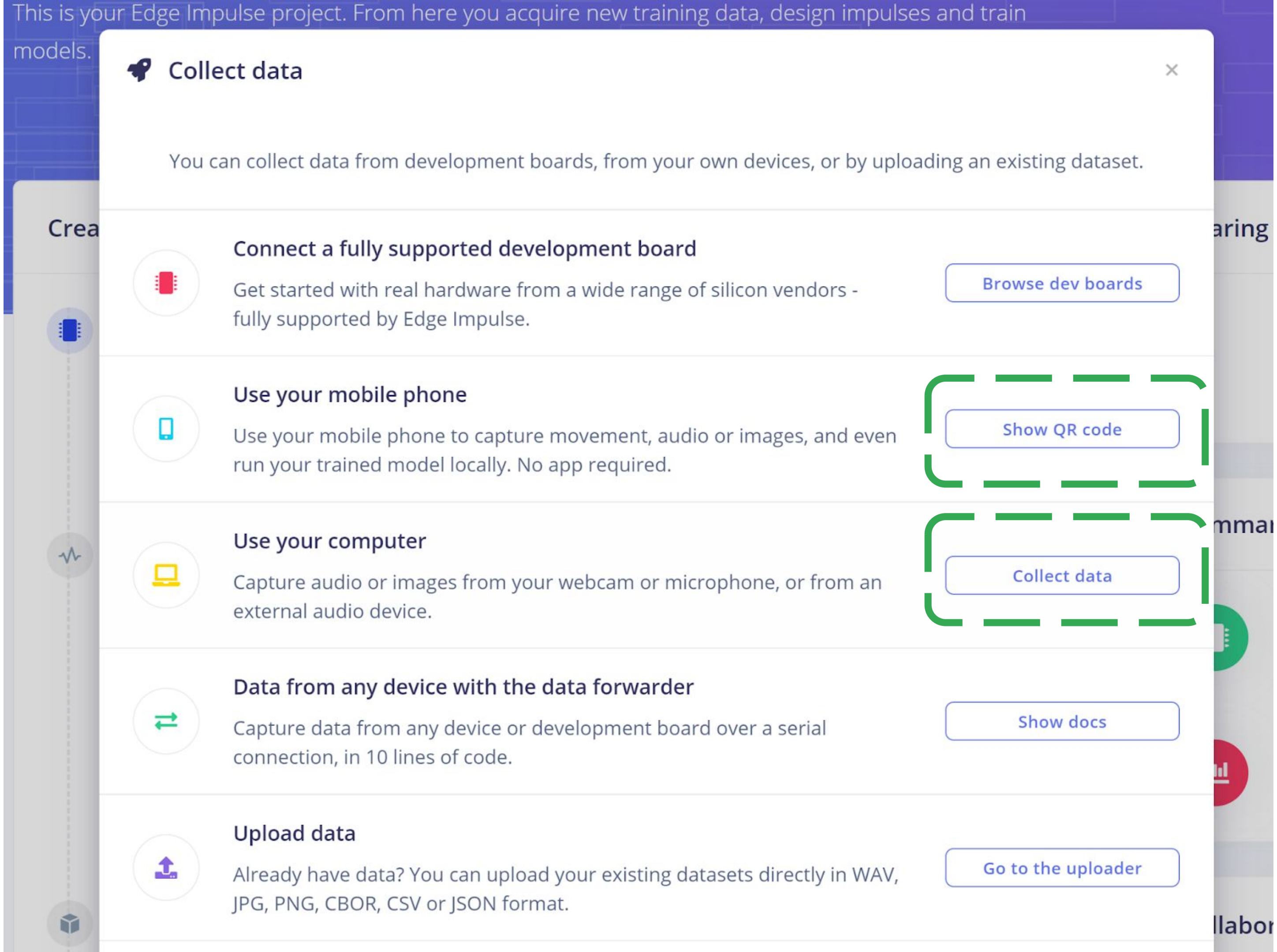
Creating your first impulse (0% complete)

Acquire data

Every Machine Learning project starts with data. You can capture data from … or import data you already collected.

Design an impulse

Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.



This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

Collect data

You can collect data from development boards, from your own devices, or by uploading an existing dataset.

Connect a fully supported development board

Get started with real hardware from a wide range of silicon vendors - fully supported by Edge Impulse.

Use your mobile phone

Use your mobile phone to capture movement, audio or images, and even run your trained model locally. No app required.

Use your computer

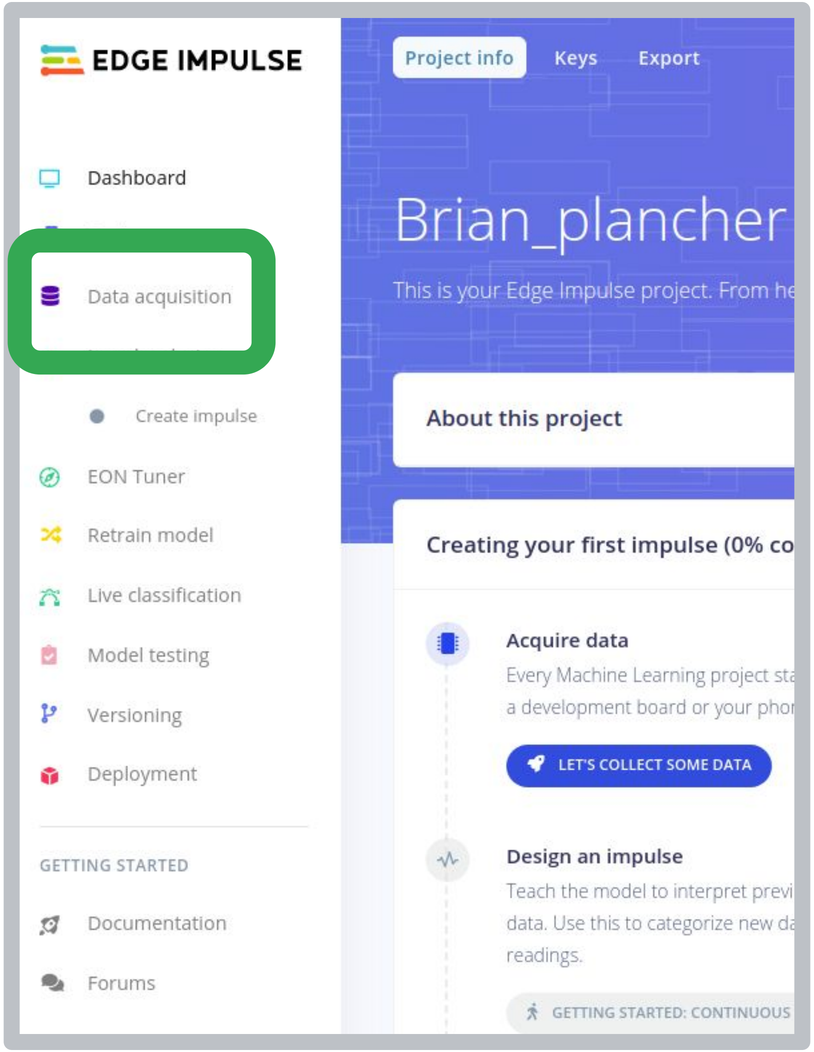
Capture audio or images from your webcam or microphone, or from an external audio device.

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.

Upload data

Already have data? You can upload your existing datasets directly in WAV, JPG, PNG, CBOR, CSV or JSON format.



EDGE IMPULSE

Project info Keys Export

Dashboard

Data acquisition

Create impulse

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

Brian_plancher

This is your Edge Impulse project. From he

About this project

Creating your first impulse (0% co

Acquire data

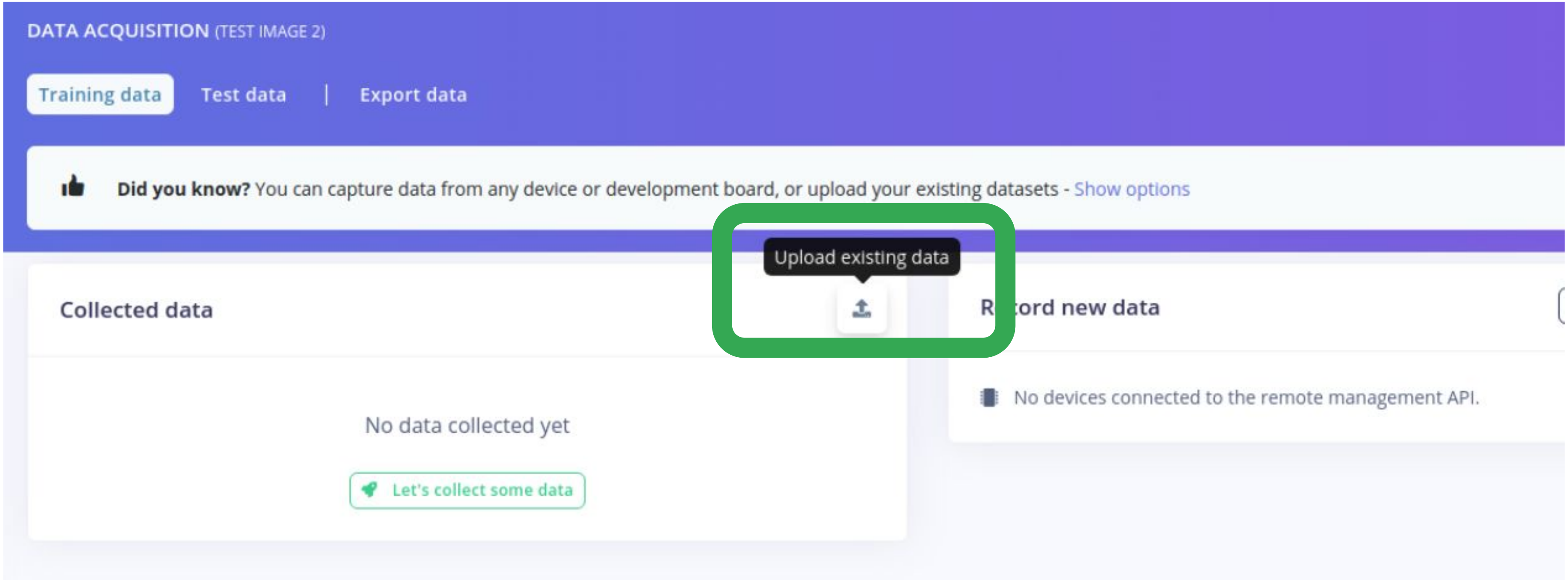
Every Machine Learning project sta
a development board or your phon

LET'S COLLECT SOME DATA

Design an impulse

Teach the model to interpret previ
data. Use this to categorize new da
readings.

GETTING STARTED: CONTINUOUS

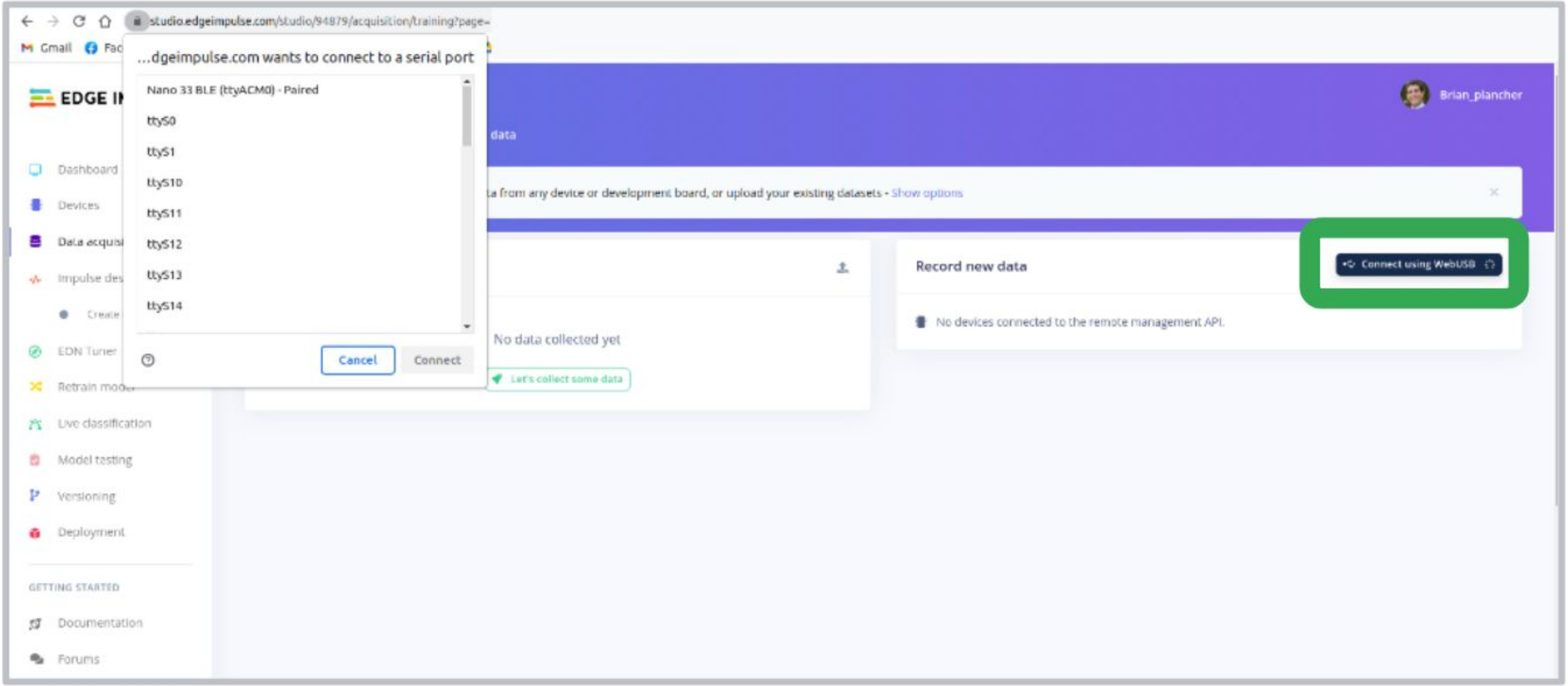


DATA ACQUISITION (TEST IMAGE 2)

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

No data collected yet

No devices connected to the remote management API.



studio.edgeimpulse.com/studio/94879/acquisition/training?page=

...dgeimpulse.com wants to connect to a serial port

Nano 33 BLE (ttyACM0) - Paired

ttyS0

ttyS1

ttyS10

ttyS11

ttyS12

ttyS13

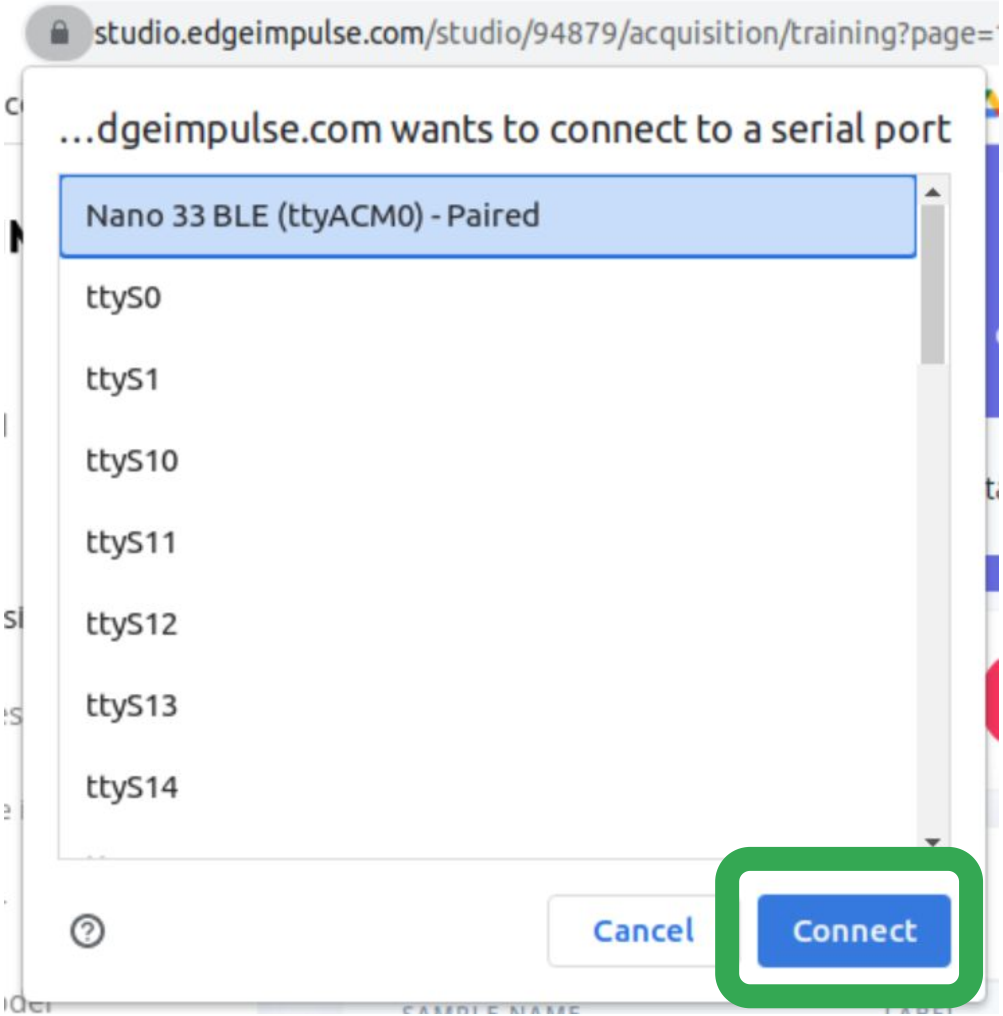
ttyS14

a from any device or development board, or upload your existing datasets - Show options

Record new data

No devices connected to the remote management API.

No data collected yet



studio.edgeimpulse.com/studio/94879/acquisition/training?page=

…dgeimpulse.com wants to connect to a serial port

Nano 33 BLE (ttyACM0) - Paired

ttyS0

ttyS1

ttyS10

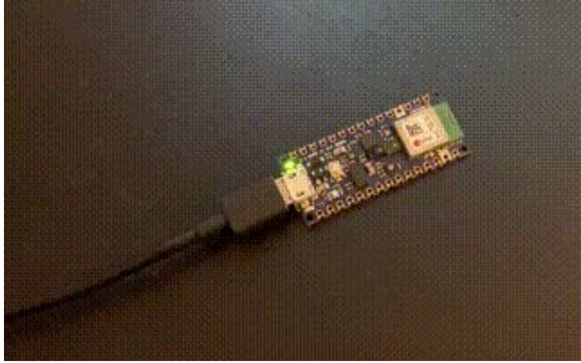
ttyS11

ttyS12

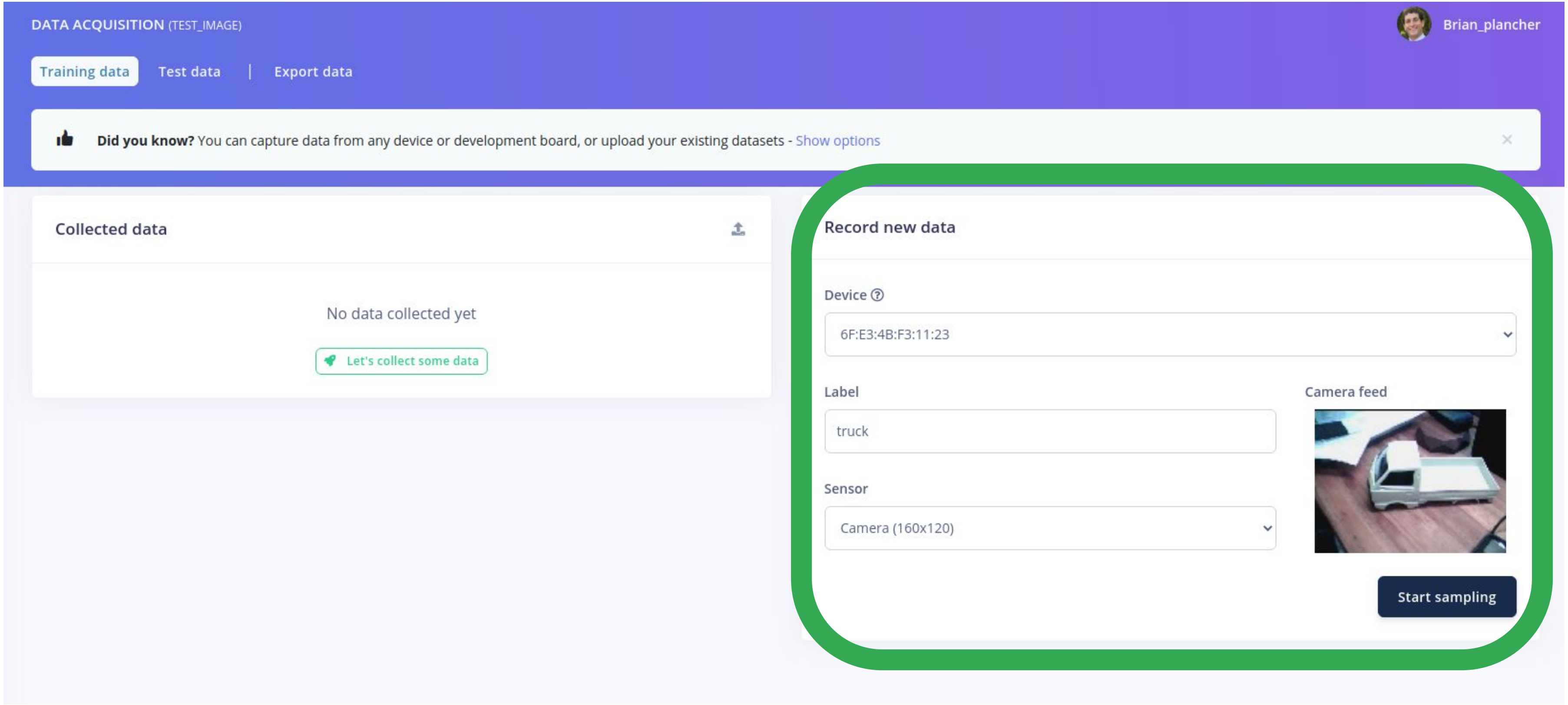
ttyS13

ttyS14

You may need to re-flash the EI Firmware!



1. Double tap RESET to enter bootloader mode
2. Download the firmware: bit.ly/EI-Nano33-Firmware
3. Run the flash script for your operating system (`flash_windows.bat`, `flash_mac.command` or `flash_linux.sh`).
4. Wait until flashing is complete, and press the RESET button once to launch the new firmware.



DATA ACQUISITION (TEST_IMAGE)

Brian_plancher

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

Collected data

No data collected yet

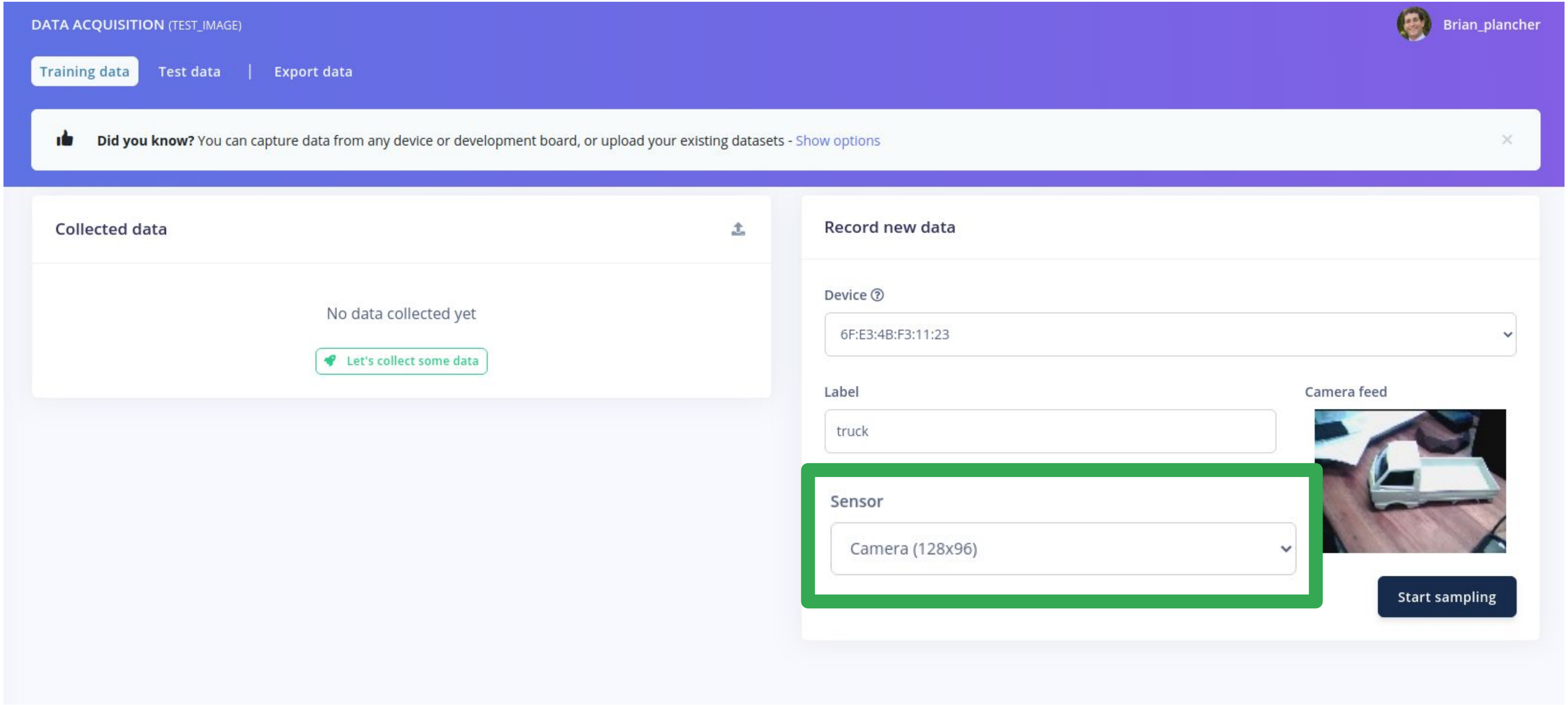
Record new data

Device: 6F:E3:4B:F3:11:23

Label: truck

Sensor: Camera (160x120)

Camera feed



DATA ACQUISITION (TEST_IMAGE)

Brian_plancher

Training data Test data | Export data

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

Collected data

No data collected yet

Let's collect some data

Record new data

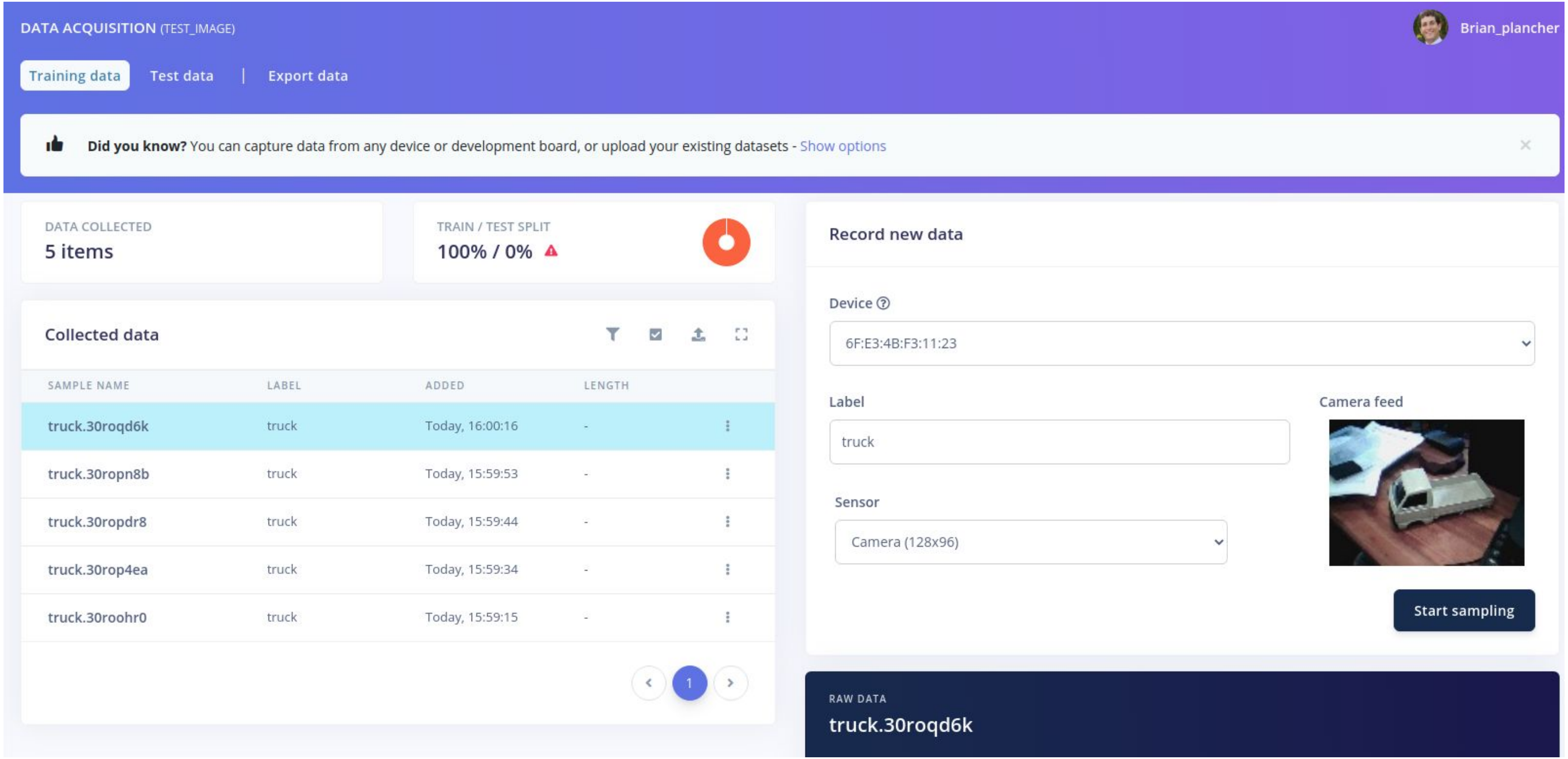
Device: 6F:E3:4B:F3:11:23

Label: truck

Camera feed

Sensor: Camera (128x96)

Start sampling



DATA ACQUISITION (TEST_IMAGE)

Brian_plancher

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

DATA COLLECTED
5 items

TRAIN / TEST SPLIT
100% / 0%

Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30roqd6k	truck	Today, 16:00:16	-
truck.30ropn8b	truck	Today, 15:59:53	-
truck.30ropdr8	truck	Today, 15:59:44	-
truck.30rop4ea	truck	Today, 15:59:34	-
truck.30roohr0	truck	Today, 15:59:15	-

Record new data

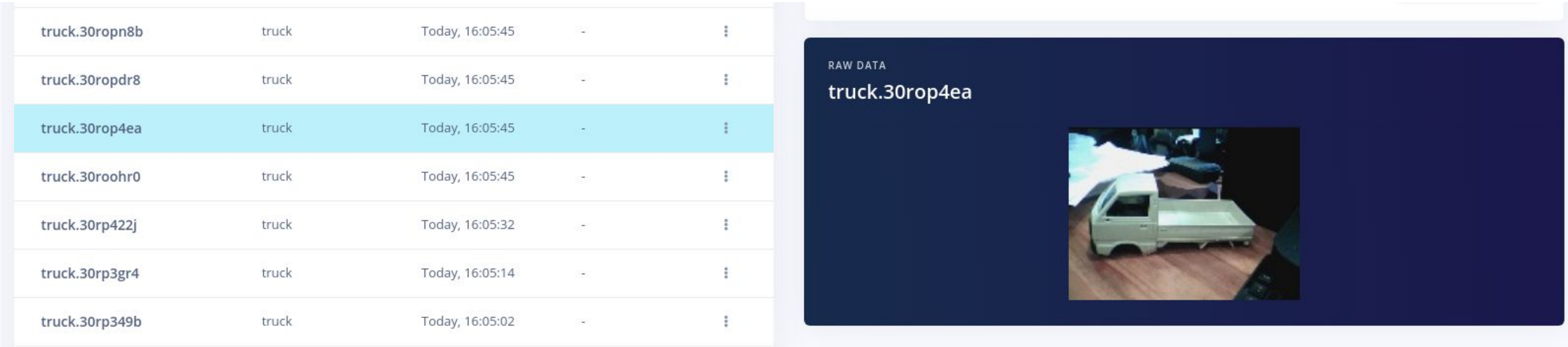
Device: 6F:E3:4B:F3:11:23

Label: truck

Sensor: Camera (128x96)

Camera feed

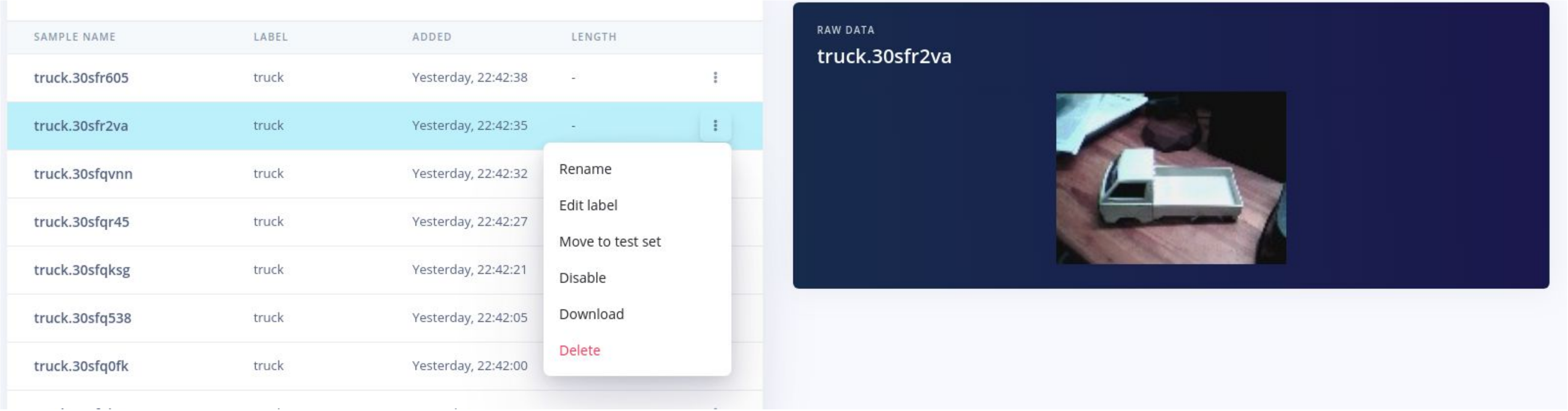
RAW DATA
truck.30roqd6k



truck.30ropn8b	truck	Today, 16:05:45	-
truck.30ropdr8	truck	Today, 16:05:45	-
truck.30rop4ea	truck	Today, 16:05:45	-
truck.30roohr0	truck	Today, 16:05:45	-
truck.30rp422j	truck	Today, 16:05:32	-
truck.30rp3gr4	truck	Today, 16:05:14	-
truck.30rp349b	truck	Today, 16:05:02	-

RAW DATA

truck.30rop4ea



SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30sfr605	truck	Yesterday, 22:42:38	-
truck.30sfr2va	truck	Yesterday, 22:42:35	-
truck.30sfqvnn	truck	Yesterday, 22:42:32	
truck.30sfqr45	truck	Yesterday, 22:42:27	
truck.30sfqksg	truck	Yesterday, 22:42:21	
truck.30sfq538	truck	Yesterday, 22:42:05	
truck.30sfq0fk	truck	Yesterday, 22:42:00	

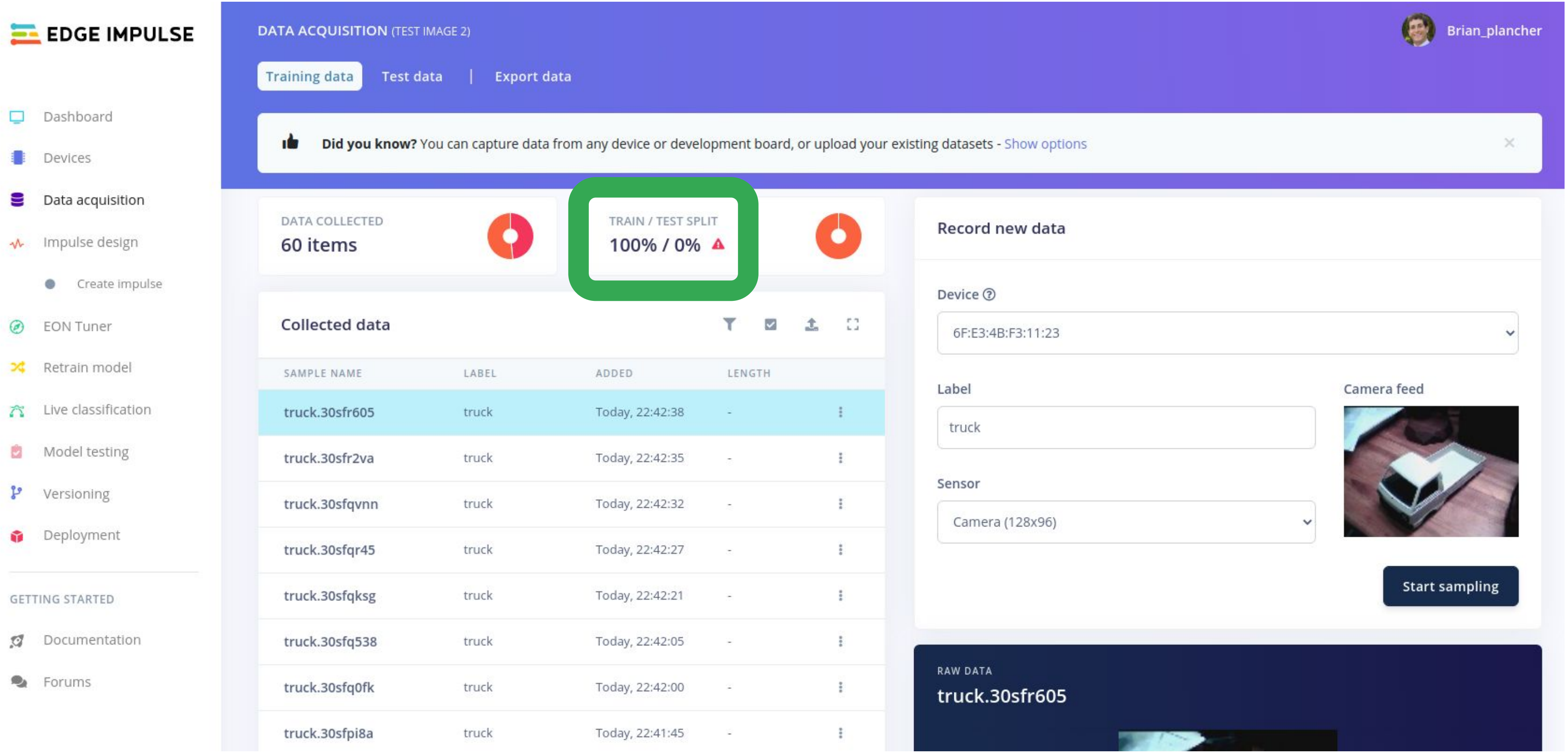
Activity: Create an Object Classification Dataset

Collect **~30 samples each** of the following classes of data:

- **Target Object #1**
- **Target Object #2**
- **(Optional) Target Object #3**

Download the firmware:
bit.ly/EI-Nano33-Firmware

```
flash_windows.bat
flash_mac.command
flash_linux.sh
```



EDGE IMPULSE

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

GETTING STARTED

- Documentation
- Forums

DATA ACQUISITION (TEST IMAGE 2)

Brian_plancher

Training data | Test 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
60 items

TRAIN / TEST SPLIT
100% / 0% ⚠

Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30sfr605	truck	Today, 22:42:38	-
truck.30sfr2va	truck	Today, 22:42:35	-
truck.30sfqvnn	truck	Today, 22:42:32	-
truck.30sfqr45	truck	Today, 22:42:27	-
truck.30sfqksg	truck	Today, 22:42:21	-
truck.30sfq538	truck	Today, 22:42:05	-
truck.30sfq0fk	truck	Today, 22:42:00	-
truck.30sfpi8a	truck	Today, 22:41:45	-

Record new data

Device: 6F:E3:4B:F3:11:23

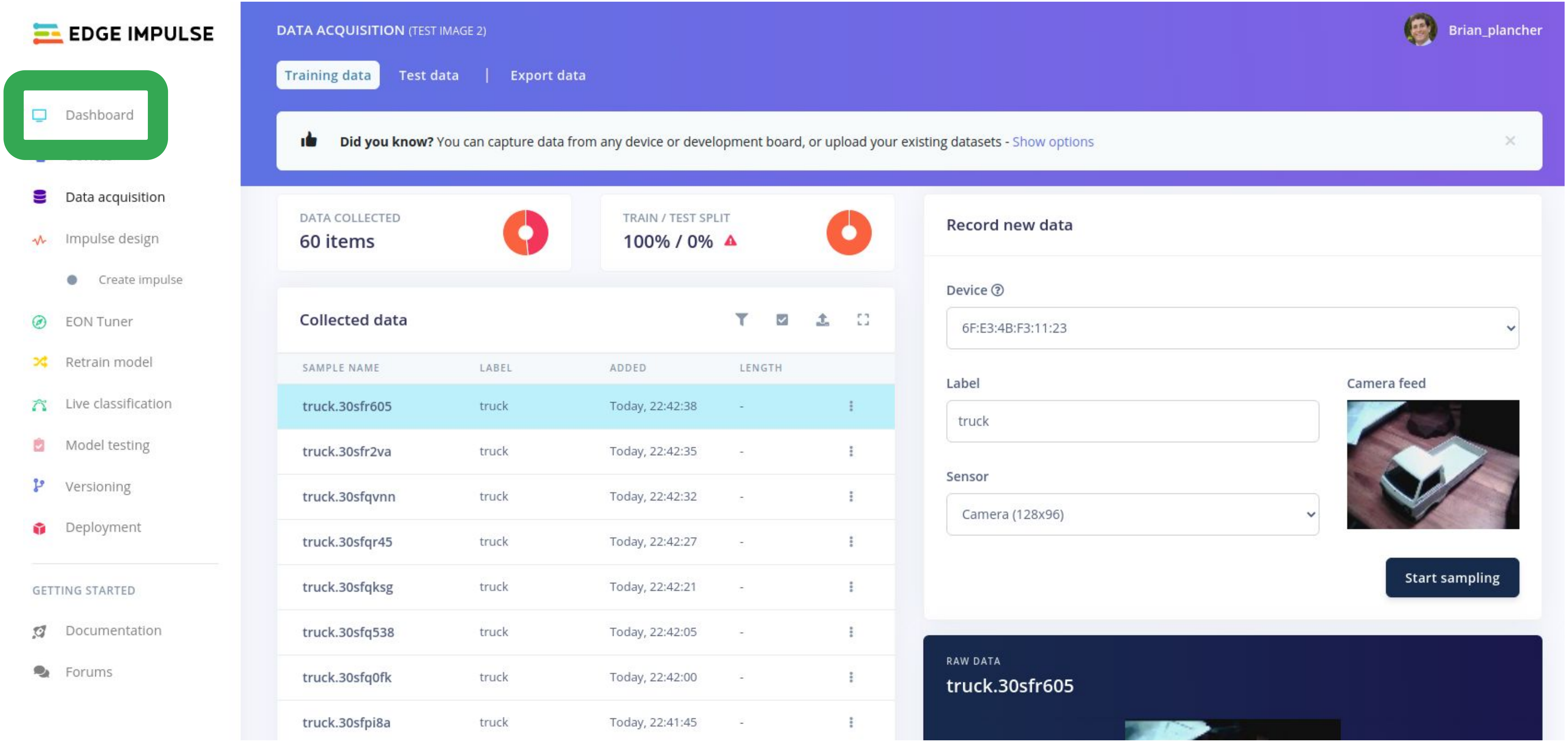
Label: truck

Sensor: Camera (128x96)

Camera feed

Start sampling

RAW DATA
truck.30sfr605



EDGE IMPULSE

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

GETTING STARTED

- Documentation
- Forums

DATA ACQUISITION (TEST IMAGE 2)

Brian_plancher

Training data | Test 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
60 items

TRAIN / TEST SPLIT
100% / 0%

Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30sfr605	truck	Today, 22:42:38	-
truck.30sfr2va	truck	Today, 22:42:35	-
truck.30sfqvnn	truck	Today, 22:42:32	-
truck.30sfqr45	truck	Today, 22:42:27	-
truck.30sfqksg	truck	Today, 22:42:21	-
truck.30sfq538	truck	Today, 22:42:05	-
truck.30sfq0fk	truck	Today, 22:42:00	-
truck.30sfpi8a	truck	Today, 22:41:45	-

Record new data

Device: 6F:E3:4B:F3:11:23

Label: truck

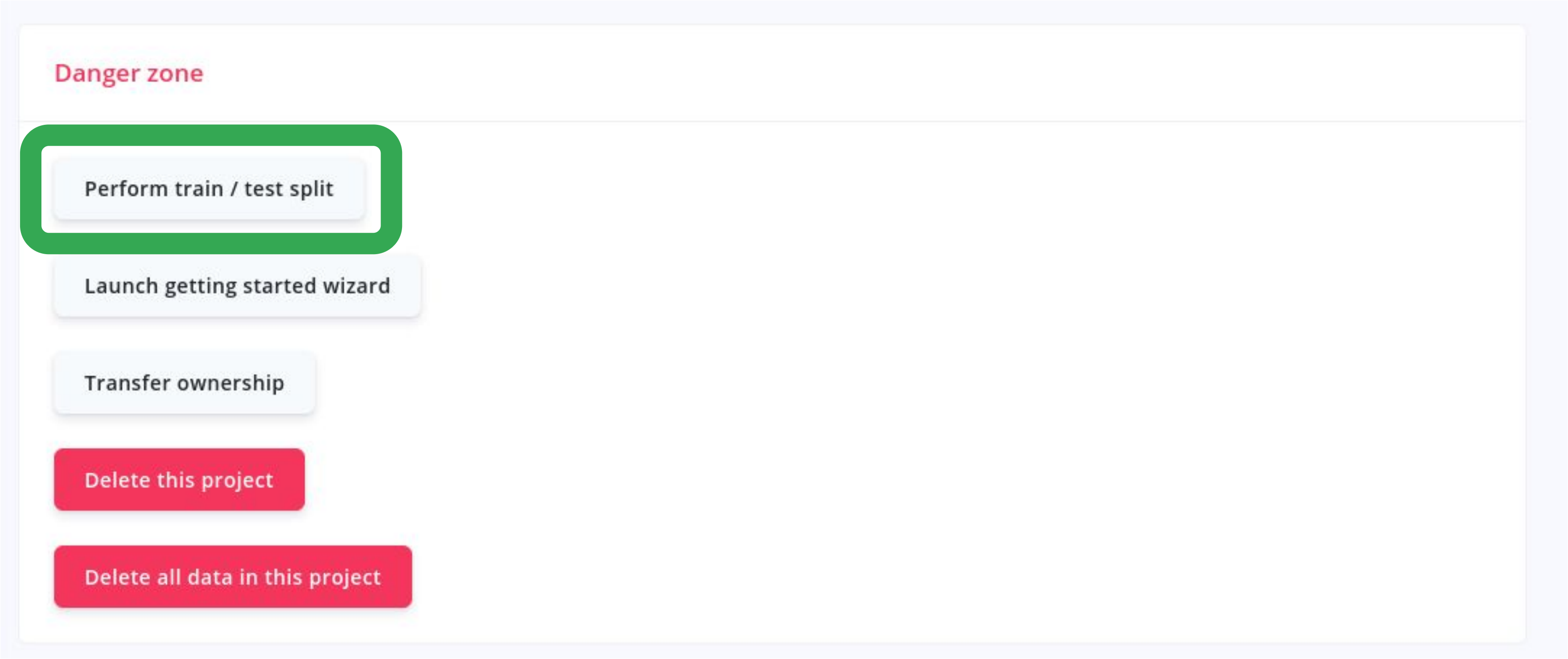
Sensor: Camera (128x96)

Camera feed

Start sampling

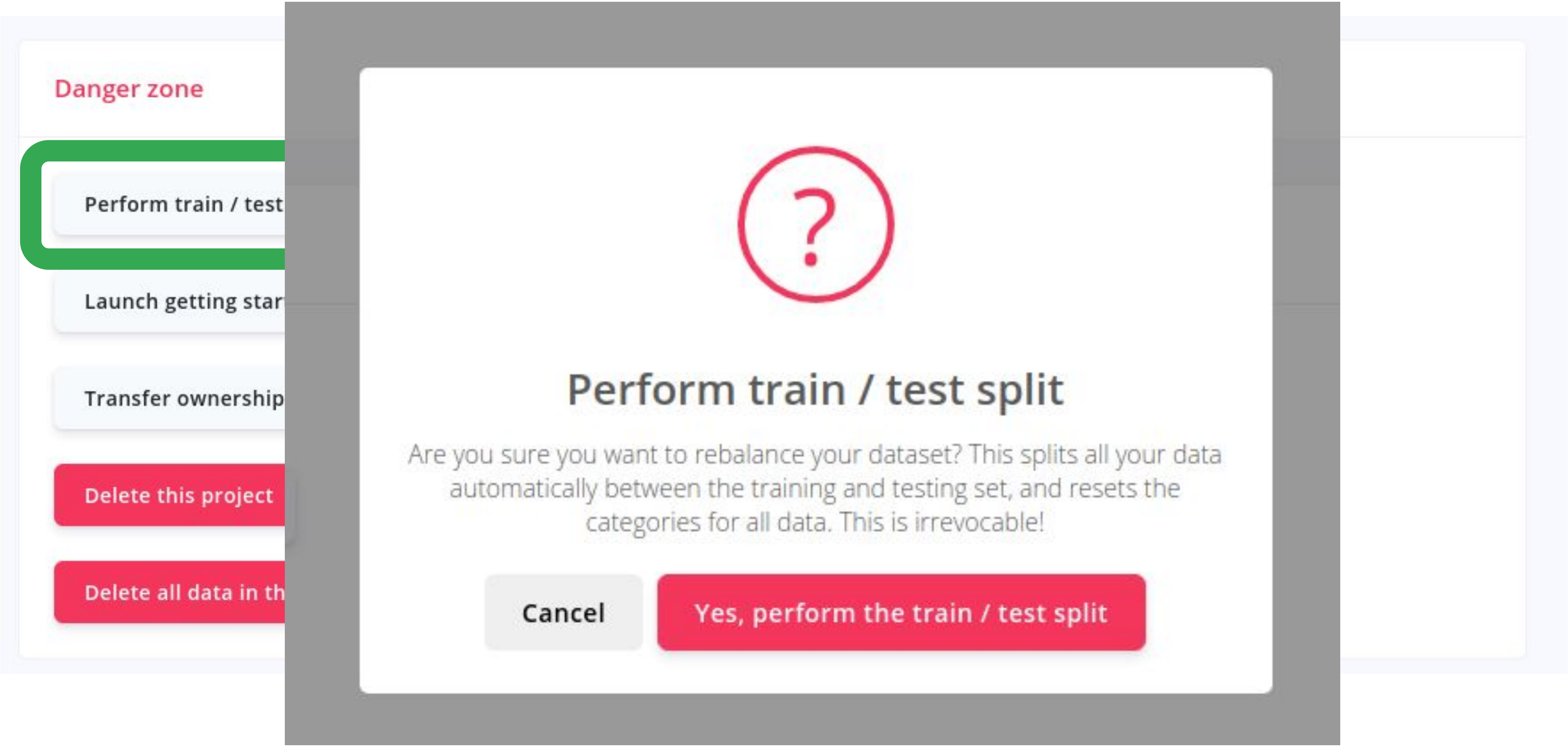
RAW DATA
truck.30sfr605

Scroll Down to the Bottom



Danger zone
Perform train / test split
Launch getting started wizard
Transfer ownership
Delete this project
Delete all data in this project

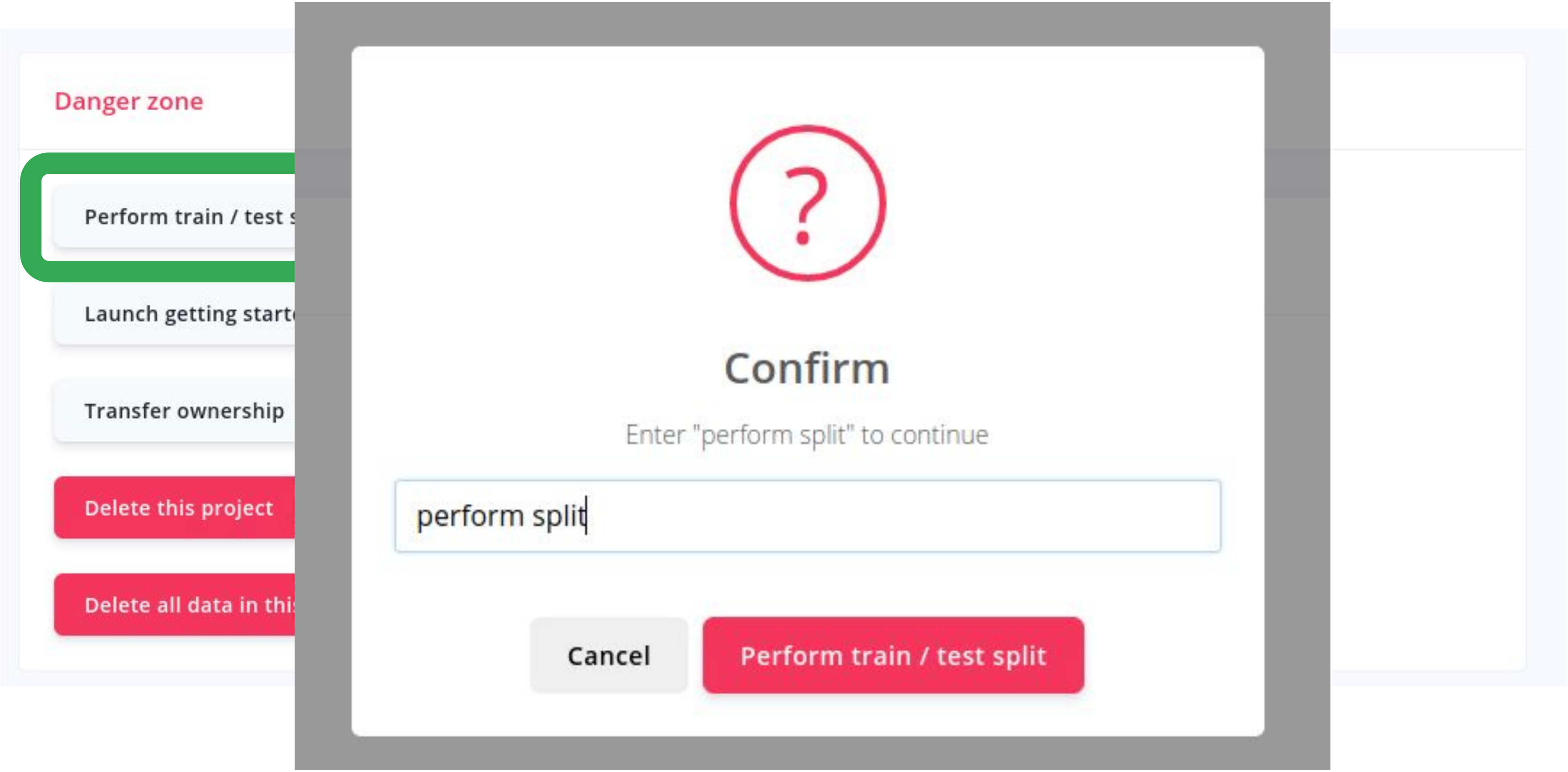
Scroll Down to the Bottom



Perform train / test split

Are you sure you want to rebalance your dataset? This splits all your data automatically between the training and testing set, and resets the categories for all data. This is irrevocable!

Scroll Down to the Bottom

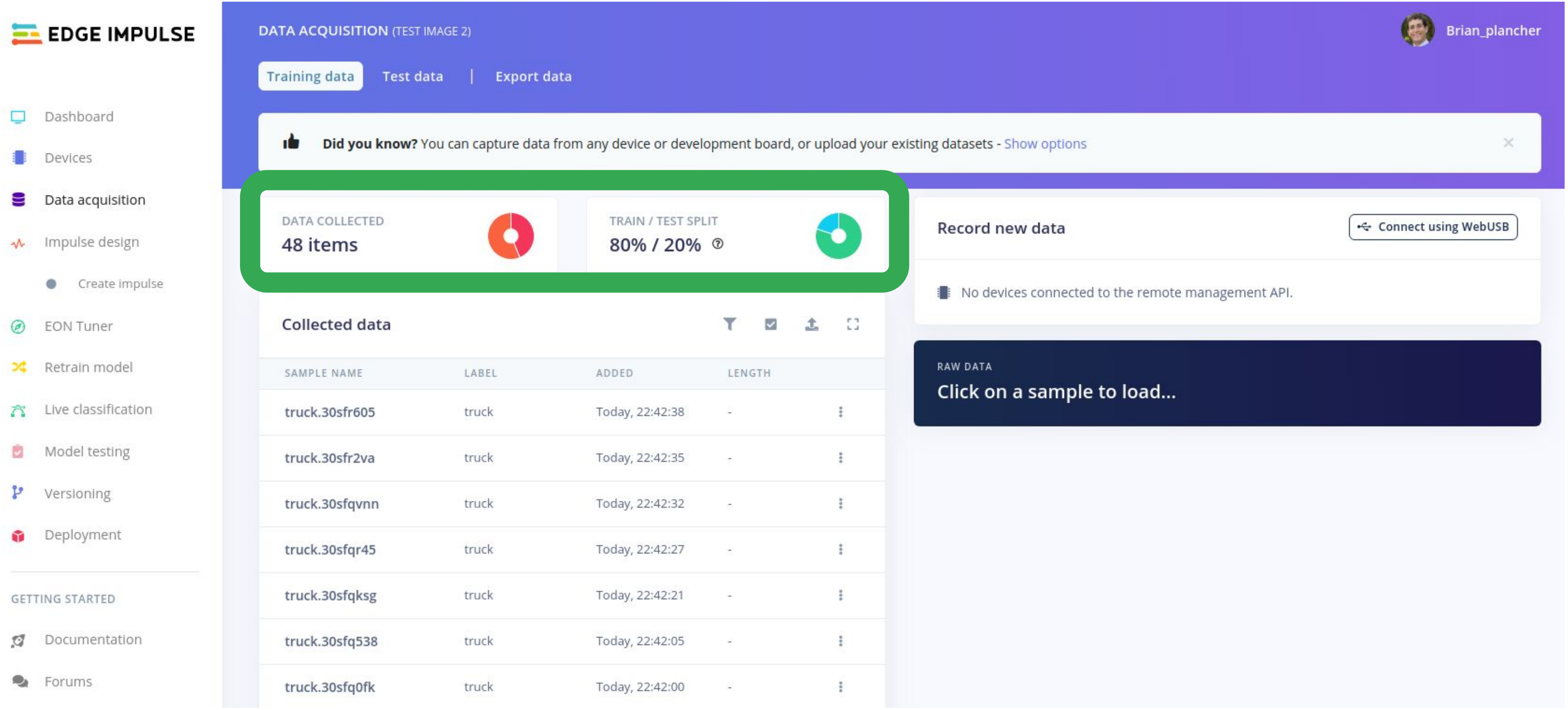


Danger zone

Confirm

Enter "perform split" to continue

perform split



DATA ACQUISITION (TEST IMAGE 2)

Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options

DATA COLLECTED
48 items

TRAIN / TEST SPLIT
80% / 20%

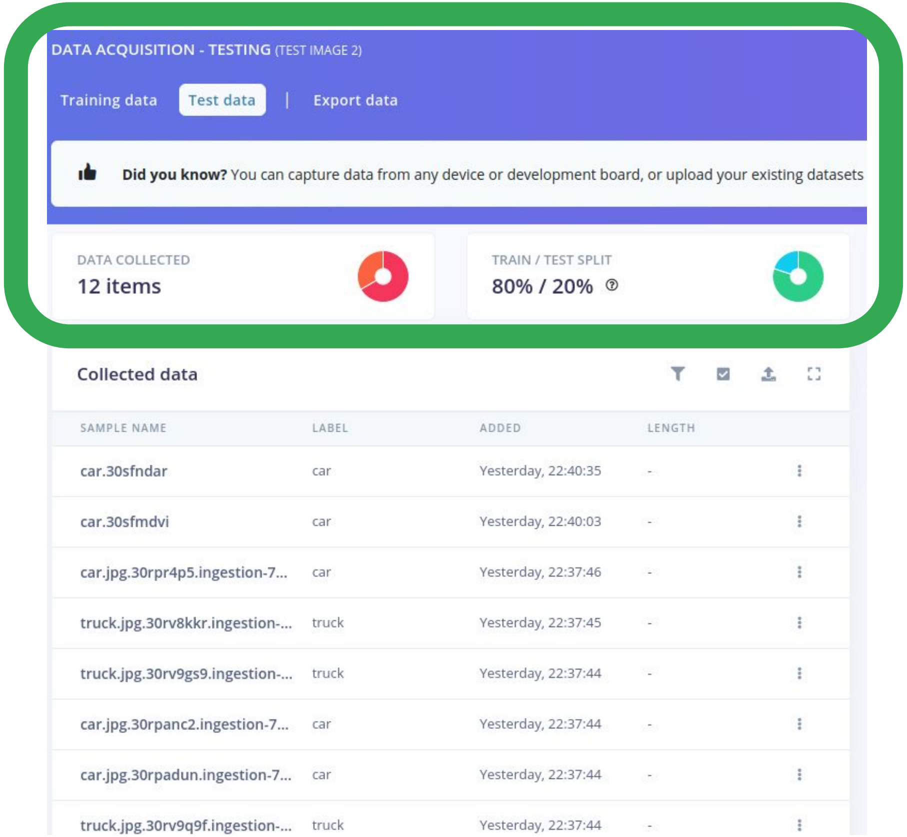
Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30sfr605	truck	Today, 22:42:38	-
truck.30sfr2va	truck	Today, 22:42:35	-
truck.30sfqvnn	truck	Today, 22:42:32	-
truck.30sfqr45	truck	Today, 22:42:27	-
truck.30sfqksg	truck	Today, 22:42:21	-
truck.30sfq538	truck	Today, 22:42:05	-
truck.30sfq0fk	truck	Today, 22:42:00	-

Record new data

No devices connected to the remote management API.

RAW DATA
Click on a sample to load...



DATA ACQUISITION - TESTING (TEST IMAGE 2)

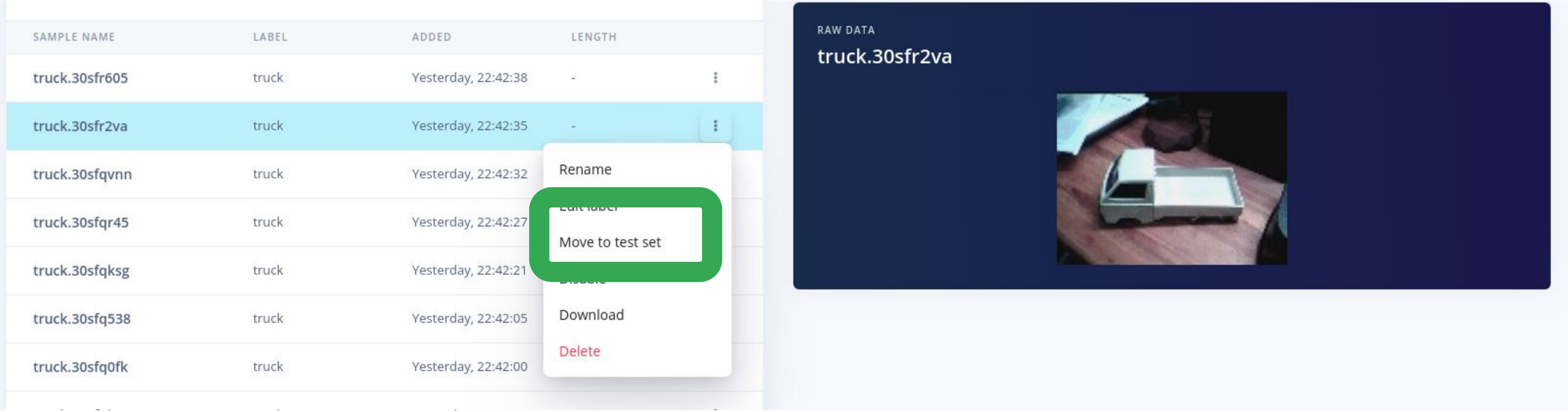
Did you know? You can capture data from any device or development board, or upload your existing datasets

DATA COLLECTED
12 items

TRAIN / TEST SPLIT
80% / 20%

Collected data

SAMPLE NAME	LABEL	ADDED	LENGTH
car.30sfndar	car	Yesterday, 22:40:35	-
car.30sfmdvi	car	Yesterday, 22:40:03	-
car.jpg.30rpr4p5.ingestion-7...	car	Yesterday, 22:37:46	-
truck.jpg.30rv8kkr.ingestion-...	truck	Yesterday, 22:37:45	-
truck.jpg.30rv9gs9.ingestion-...	truck	Yesterday, 22:37:44	-
car.jpg.30rpanc2.ingestion-7...	car	Yesterday, 22:37:44	-
car.jpg.30rpadun.ingestion-7...	car	Yesterday, 22:37:44	-
truck.jpg.30rv9q9f.ingestion-...	truck	Yesterday, 22:37:44	-

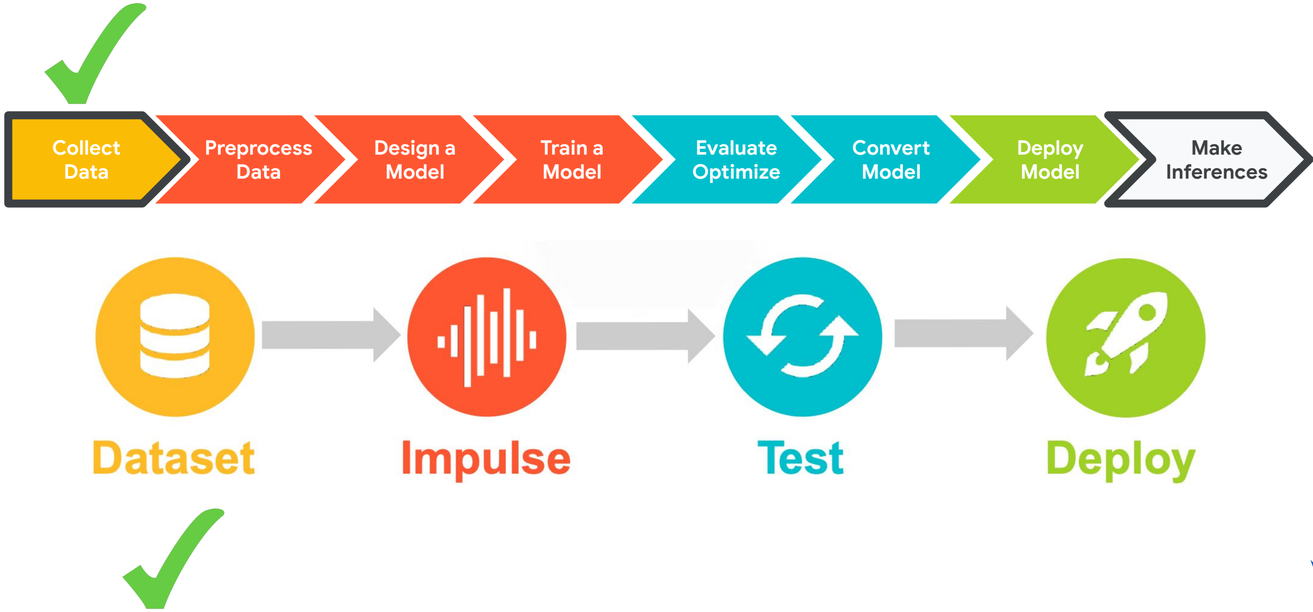


SAMPLE NAME	LABEL	ADDED	LENGTH
truck.30sfr605	truck	Yesterday, 22:42:38	-
truck.30sfr2va	truck	Yesterday, 22:42:35	-
truck.30sfqvnn	truck	Yesterday, 22:42:32	
truck.30sfqr45	truck	Yesterday, 22:42:27	
truck.30sfqksg	truck	Yesterday, 22:42:21	
truck.30sfq538	truck	Yesterday, 22:42:05	
truck.30sfq0fk	truck	Yesterday, 22:42:00	

Today’s Agenda

- Introduction to Computer Vision
- Hands-on Computer Vision: Thing Translator
- Building an Object Detection Dataset
- **Training our Model using Transfer Learning**
- Deploying our Model onto our Arduino
- Summary

Edge Impulse Project Dashboard



Collect Data

Preprocess Data

Design a Model

Train a Model

Evaluate Optimize

Convert Model

Deploy Model

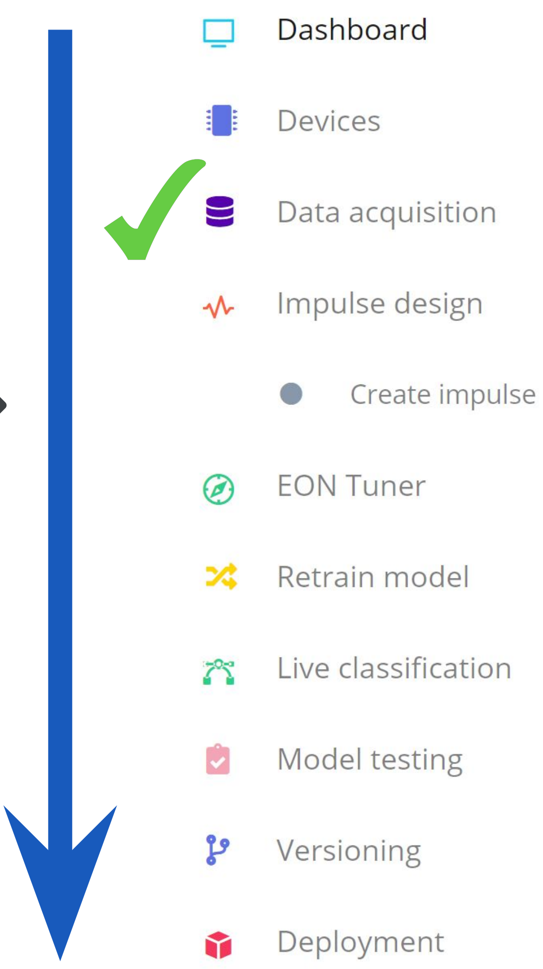
Make Inferences

Dataset

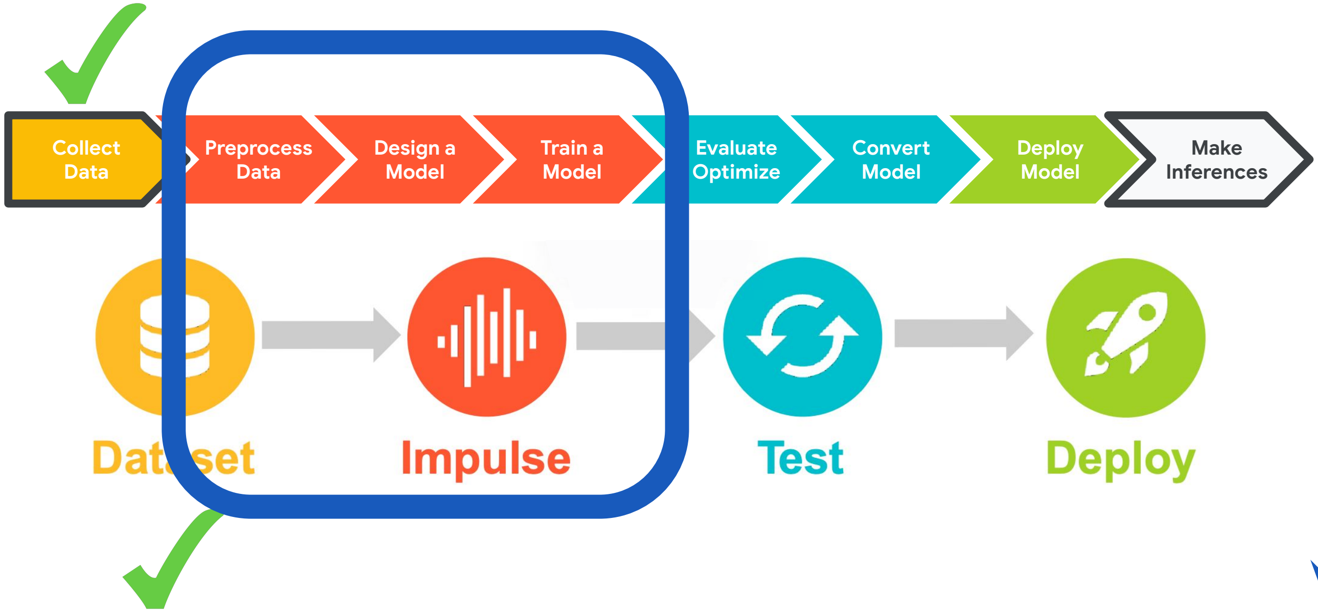
Impulse

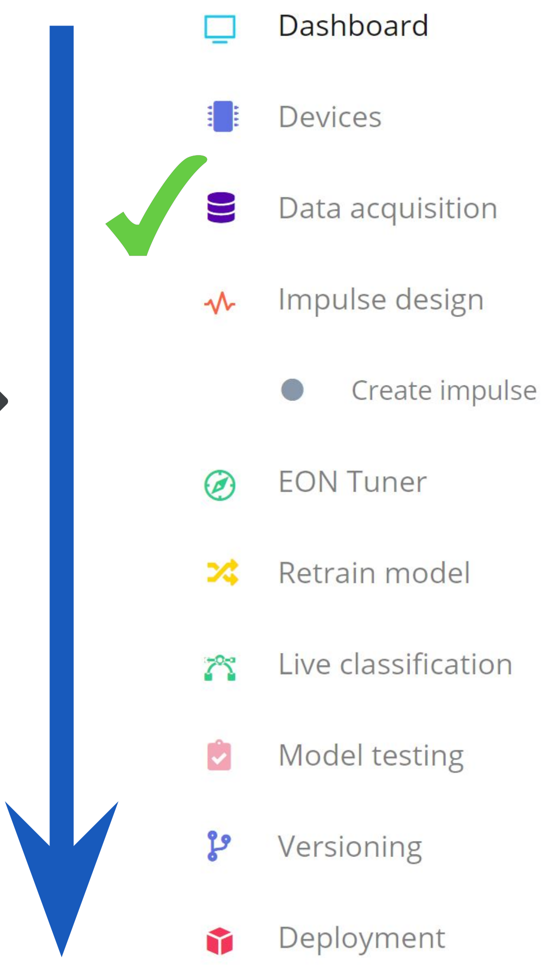
Test

Deploy

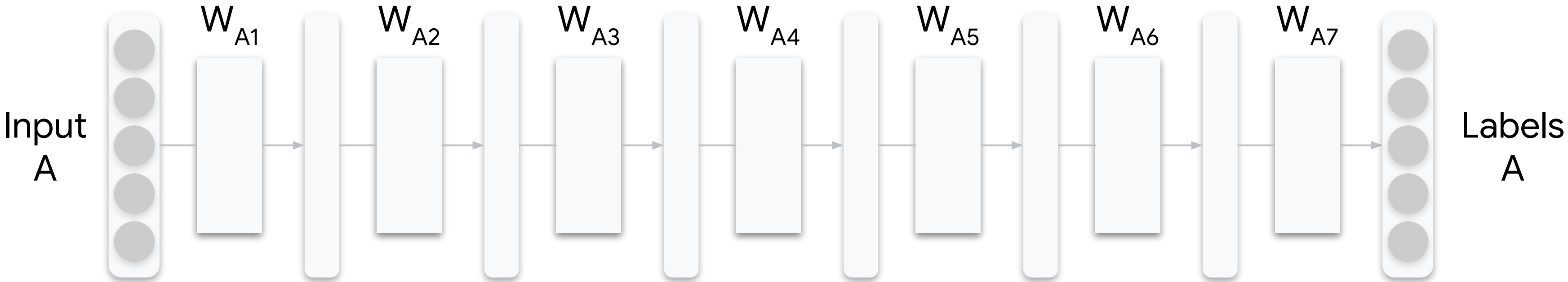


Edge Impulse Project Dashboard

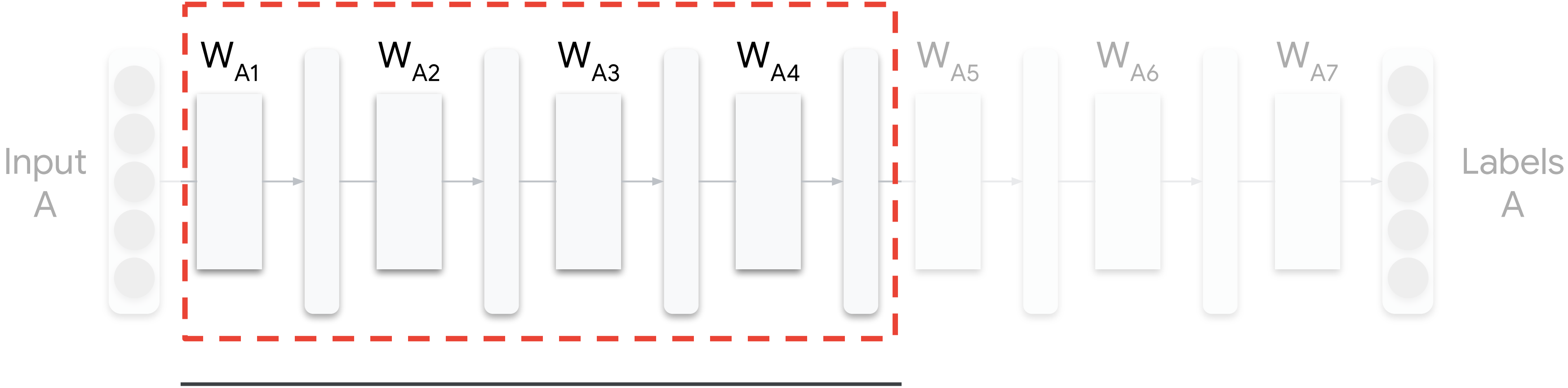




Transfer Learning: Saving time and computational resources

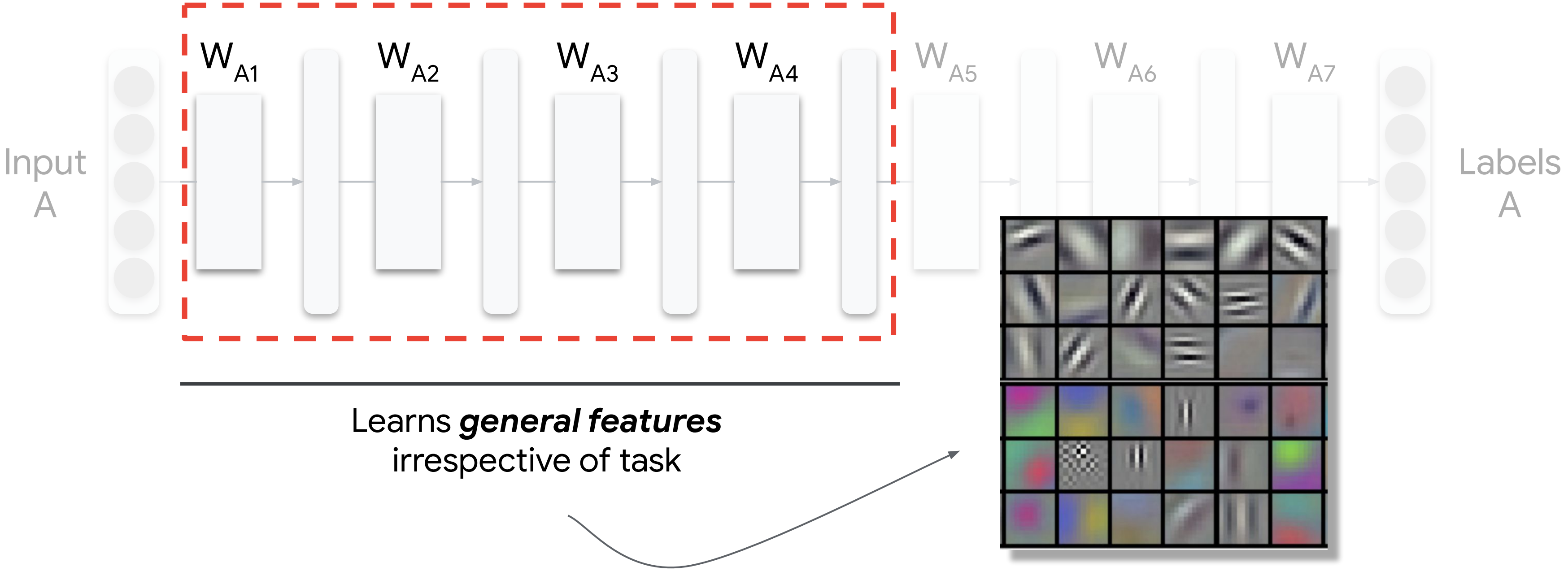


Transfer Learning: Saving time and computational resources



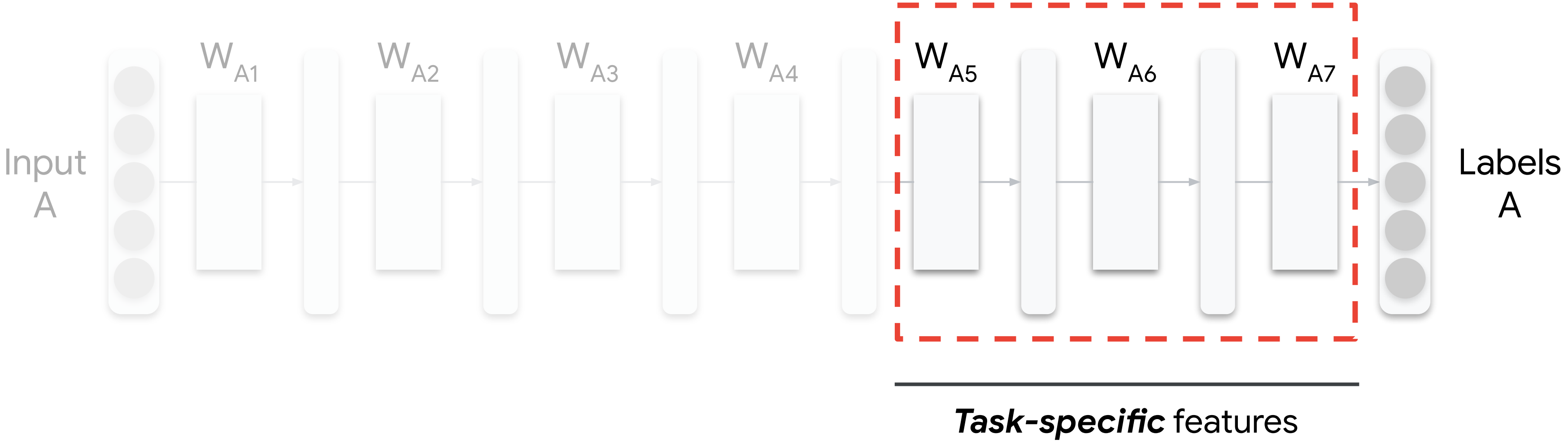
Learns ***general features***
irrespective of task

Transfer Learning: Saving time and computational resources

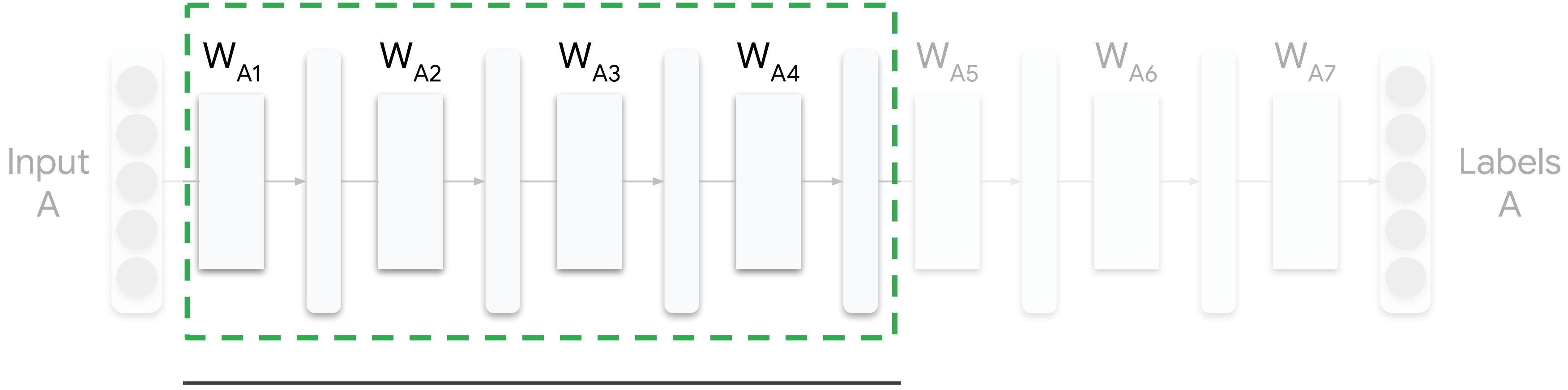


Learns ***general features***
irrespective of task

Transfer Learning: Saving time and computational resources



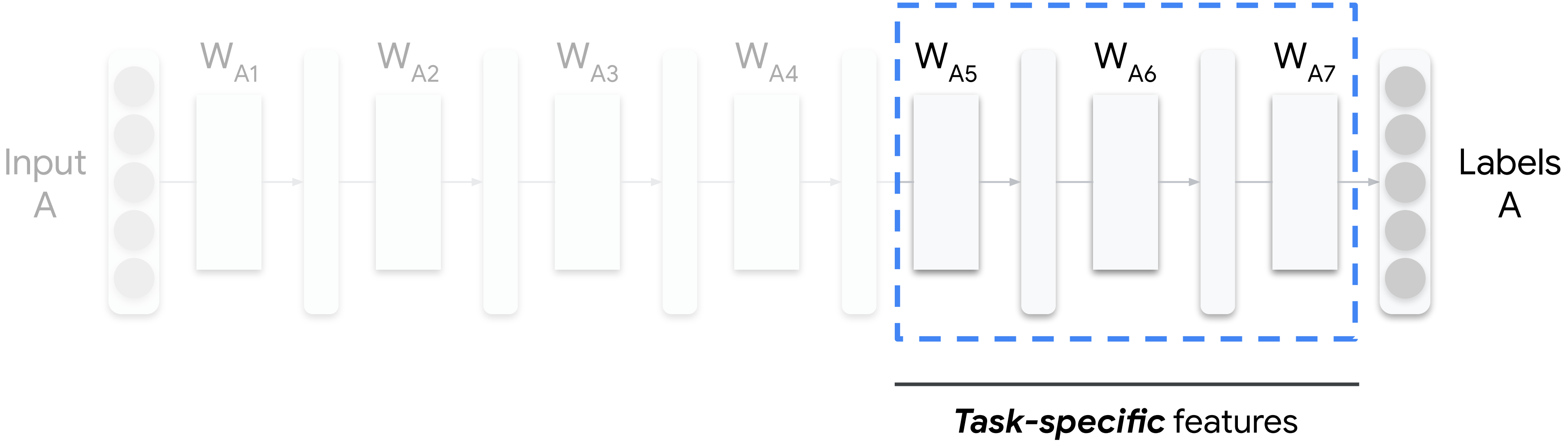
Transfer Learning: Saving time and computational resources



Learns ***general features*** irrespective of task

Reuse (freeze general feature extraction)

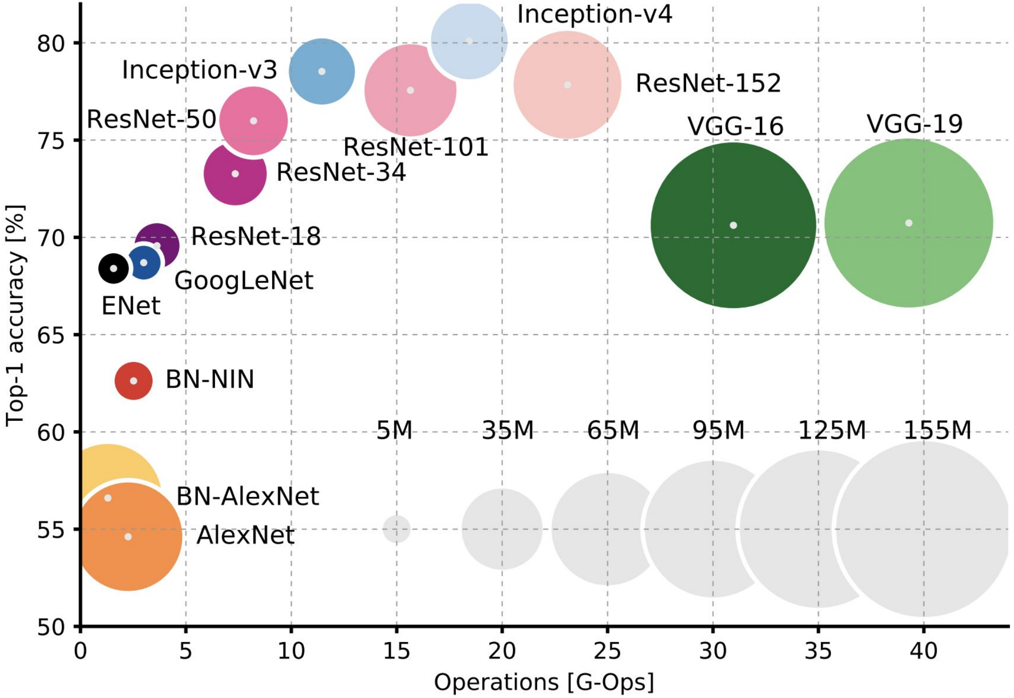
Transfer Learning: Saving time and computational resources



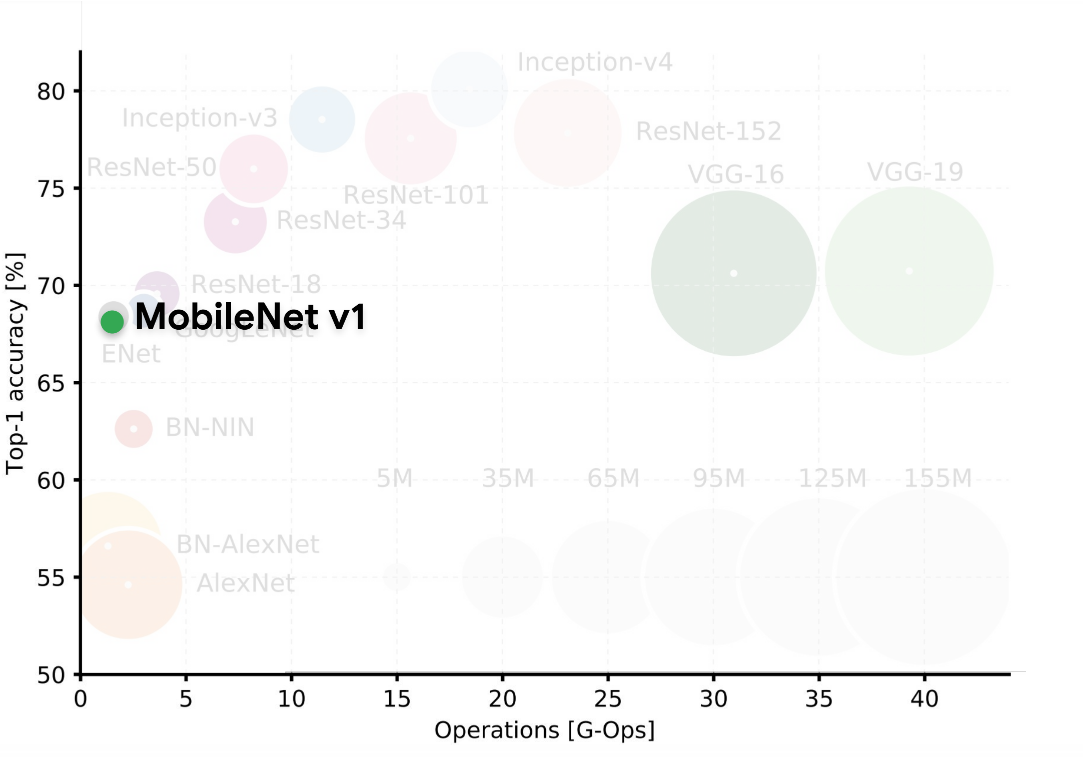
Train only last few layers

So **what model** should we transfer from?

Model **Evolution**



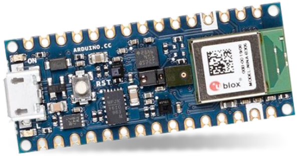
Model **Evolution**



MobileNet v1

Model	Size	Top-1 Accuracy
MobileNet v1	***16 MB***	0.713

Fine for mobile phones with GB of RAM, but 64X microcontroller RAM



Our board [Course 3 Kit] only has **256KB** of RAM (memory)

Further **Optimizations**

Multiply-Accumulates

α	**Image Size**	**MACs** (millions)	**Params** (millions)	**Top-1 Accuracy**
1	224	569	4.24	**70.7**
1	128	186	4.14	64.1
0.75	224	317	2.59	68.4
0.75	128	104	2.59	61.8
0.5	224	150	1.34	64.0
0.5	128	49	1.34	56.2
0.25	224	41	0.47	50.6
0.25	128	14	0.47	41.2

Further **Optimizations**

Multiply-Accumulates

α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	224	569	4.24	**70.7**
1	128	186	4.14	64.1
0.75	224	317	2.59	**68.4**
0.75	128	104	2.59	61.8
0.5	224	150	1.34	64.0
0.5	128	49	1.34	56.2
0.25	224	41	0.47	50.6
0.25	128	14	0.47	41.2

Further **Optimizations**

Multiply-Accumulates

α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	224	569	**4.24**	**70.7**
1	128	186	4.14	64.1
0.75	224	317	**2.59**	**68.4**
0.75	128	104	2.59	61.8
0.5	224	150	1.34	64.0
0.5	128	49	1.34	56.2
0.25	224	41	0.47	50.6
0.25	128	14	0.47	41.2

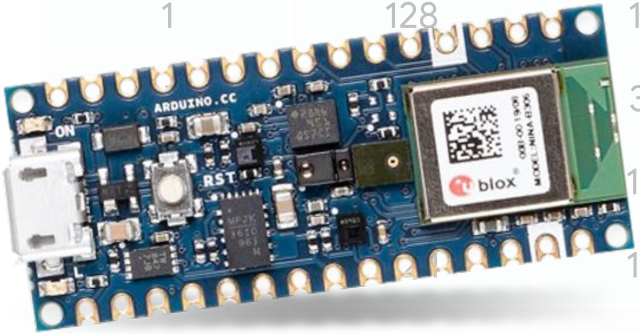
Further **Optimizations**

Multiply-Accumulates

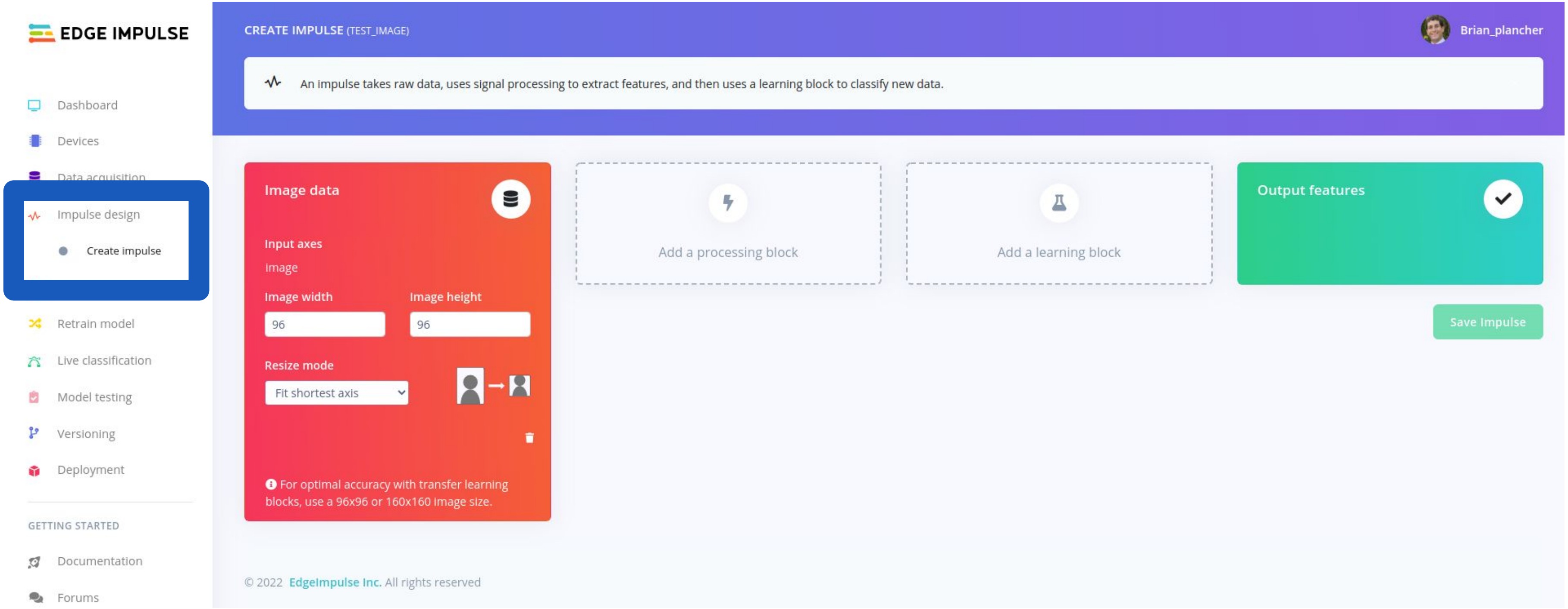
α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	**224**	**569**	**4.24**	**70.7**
1	**128**	**186**	**4.14**	**64.1**
0.75	224	317	2.59	68.4
0.75	128	104	2.59	61.8
0.5	224	150	1.34	64.0
0.5	128	49	1.34	56.2
0.25	224	41	0.47	50.6
0.25	128	14	0.47	41.2

Further **Optimizations**

α	Image Size	MACs (millions)	Params (millions)	Top-1 Accuracy
1	224	569	4.24	70.7
1	128	186	[illegible]	[illegible]
[illegible]	[illegible]	317	[illegible]	[illegible]
[illegible]	[illegible]	104	[illegible]	[illegible]
[illegible]	[illegible]	150	1.34	64.0
0.5	128	49	1.34	56.2
0.25	224	41	0.47	50.6
0.25	128	14	0.47	41.2



We will need to **both** reduce alpha and the image size!



CREATE IMPULSE (TEST_IMAGE)

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

Image data

Input axes
image

Image width: 96

Image height: 96

Resize mode: Fit shortest axis

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

© 2022 EdgeImpulse Inc. All rights reserved

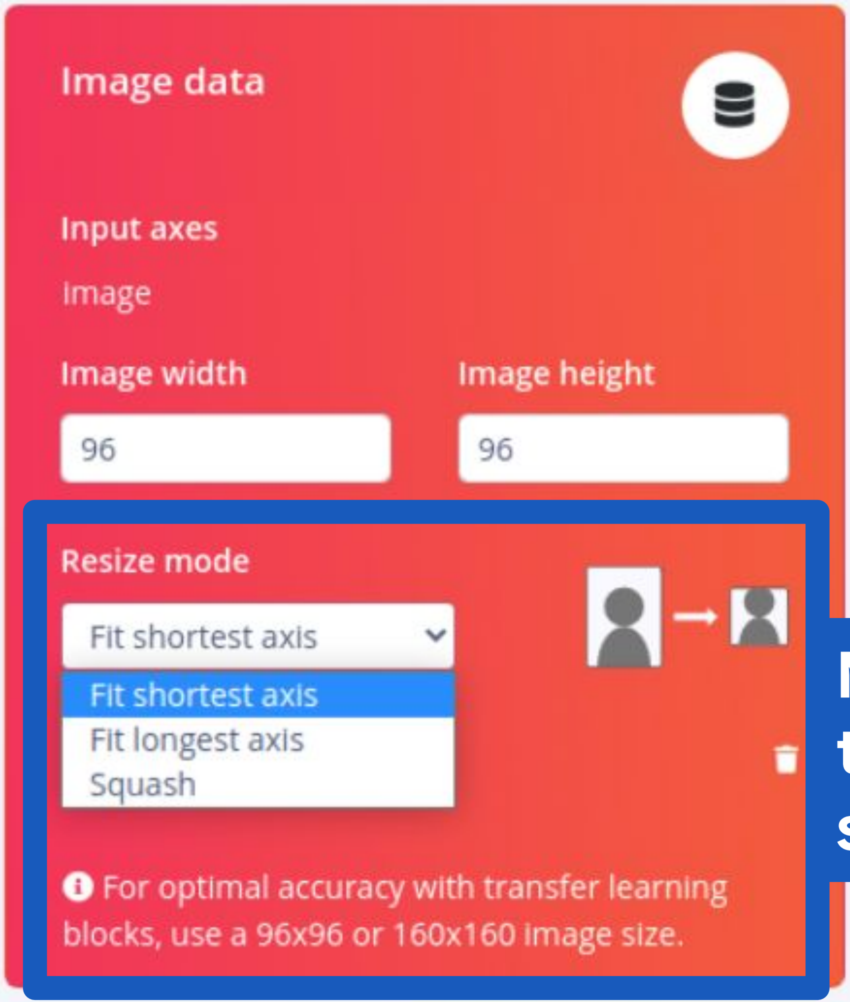


Image data

Input axes

image

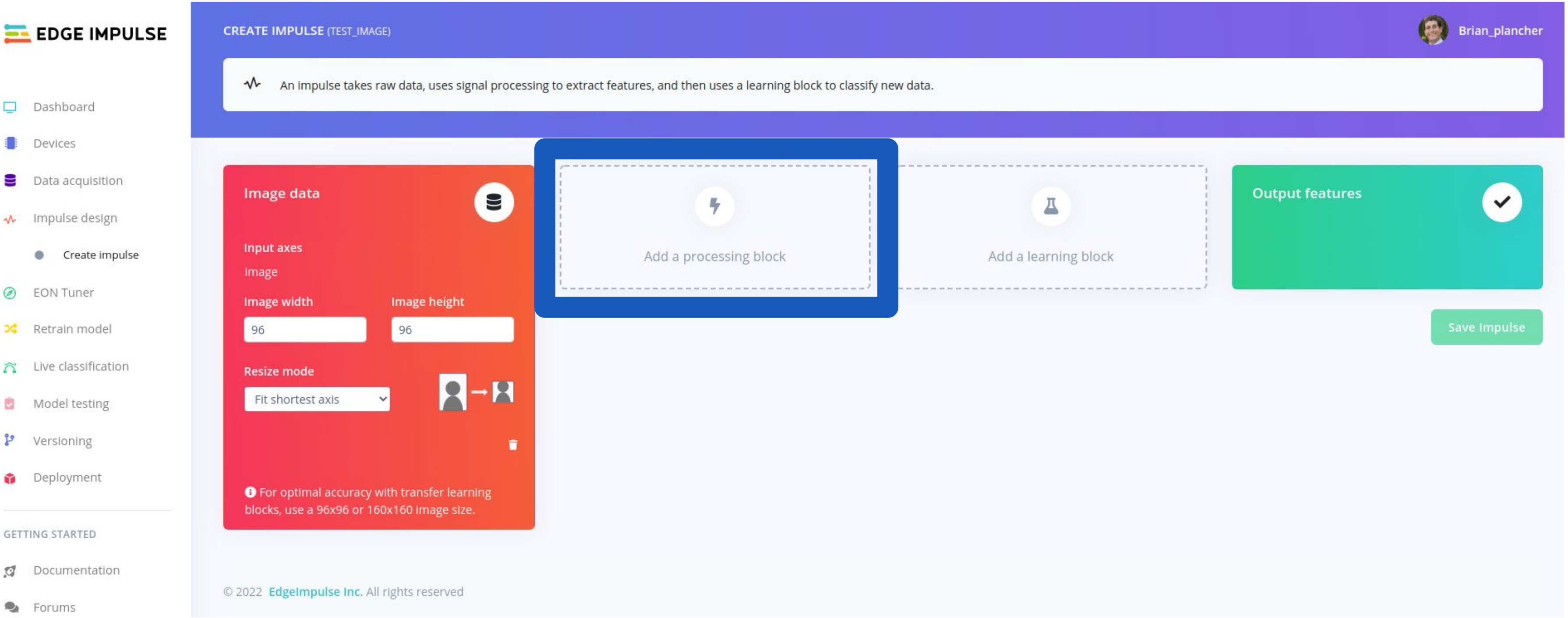
Image width: 96

Image height: 96

Resize mode: Fit shortest axis

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

MobileNet is trained on square images!



EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

- Create impulse

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

CREATE IMPULSE (TEST_IMAGE)

Brian_plancher

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

Image data

Input axes

image

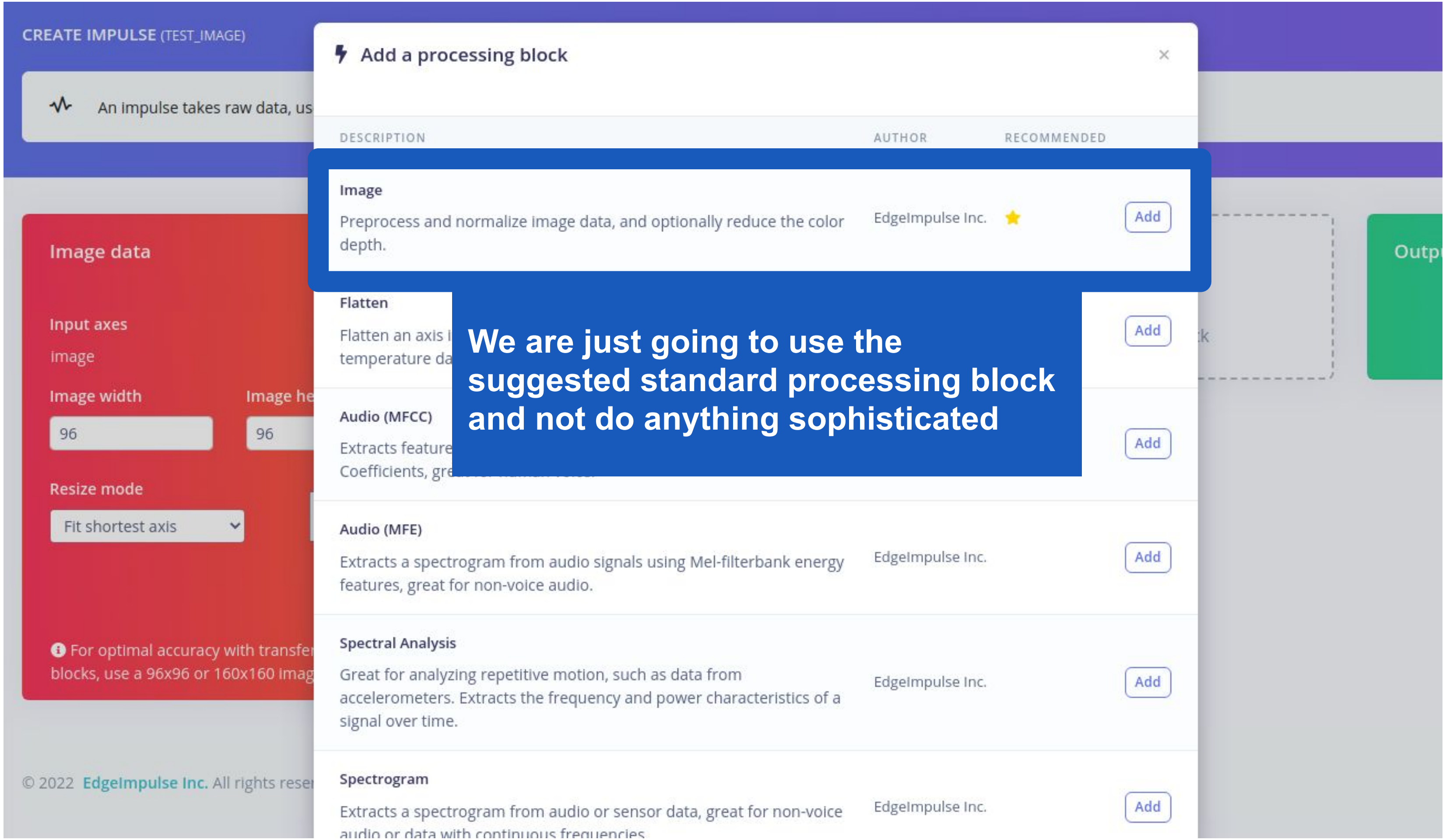
Image width	Image height
96	96

Resize mode

Fit shortest axis

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

© 2022 EdgeImpulse Inc. All rights reserved



CREATE IMPULSE (TEST_IMAGE)

An impulse takes raw data, us

Image data

Input axes

image

Image width

96

Image he

96

Resize mode

Fit shortest axis

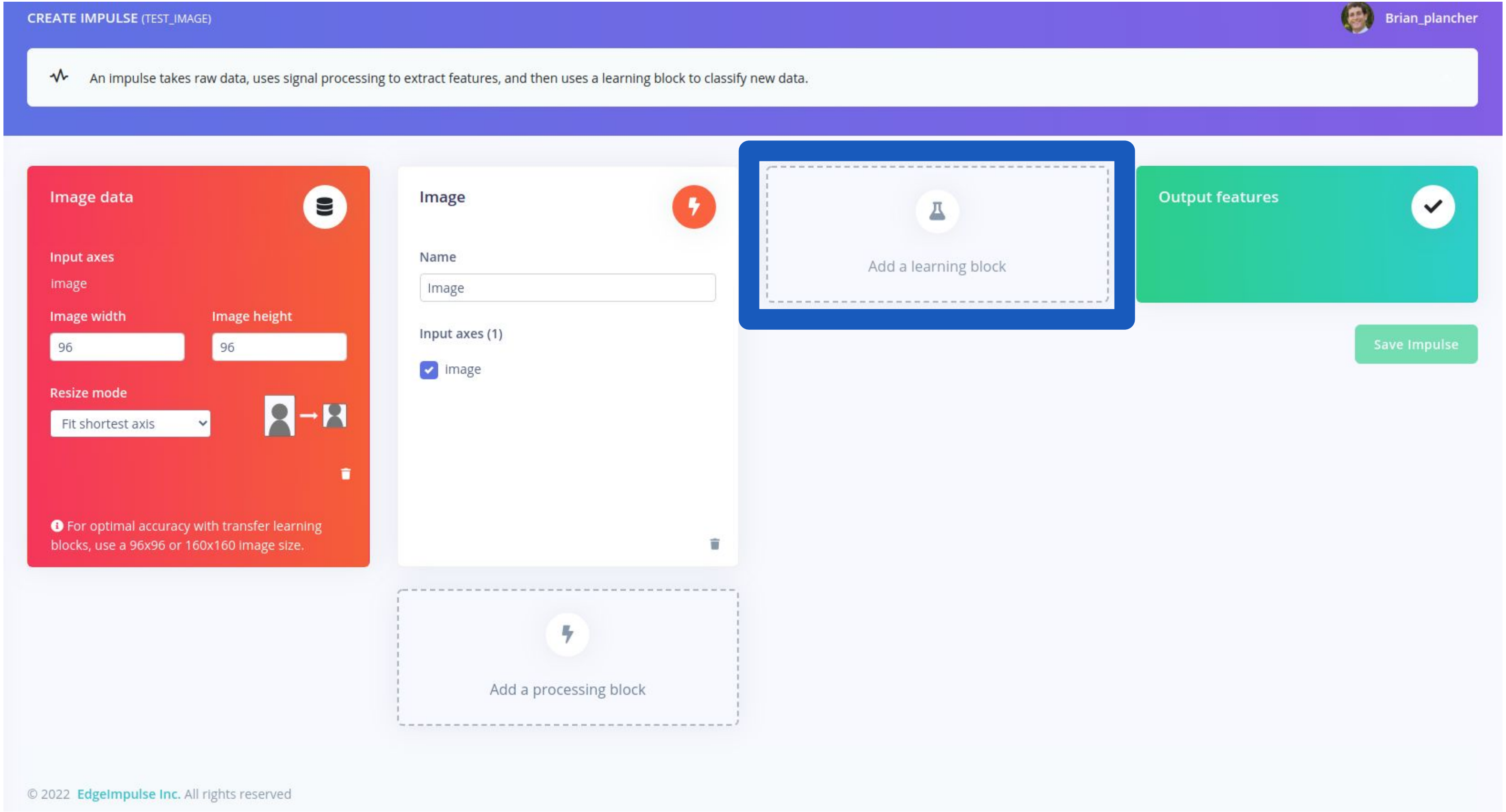
For optimal accuracy with transfe blocks, use a 96x96 or 160x160 imag

© 2022 EdgeImpulse Inc. All rights rese

Add a processing block

DESCRIPTION	AUTHOR	RECOMMENDED	
Image Preprocess and normalize image data, and optionally reduce the color depth.	EdgeImpulse Inc.	★	Add
Flatten Flatten an axis i temperature da			Add
Audio (MFCC) Extracts feature Coefficients, gre			Add
Audio (MFE) Extracts a spectrogram from audio signals using Mel-filterbank energy features, great for non-voice audio.	EdgeImpulse Inc.		Add
Spectral Analysis Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	EdgeImpulse Inc.		Add
Spectrogram Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies	EdgeImpulse Inc.		Add

We are just going to use the suggested standard processing block and not do anything sophisticated



CREATE IMPULSE (TEST_IMAGE)

Brian_plancher

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

Image data

Input axes

image

Image width: 96

Image height: 96

Resize mode: Fit shortest axis

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

Image

Name: Image

Input axes (1)

☑ image

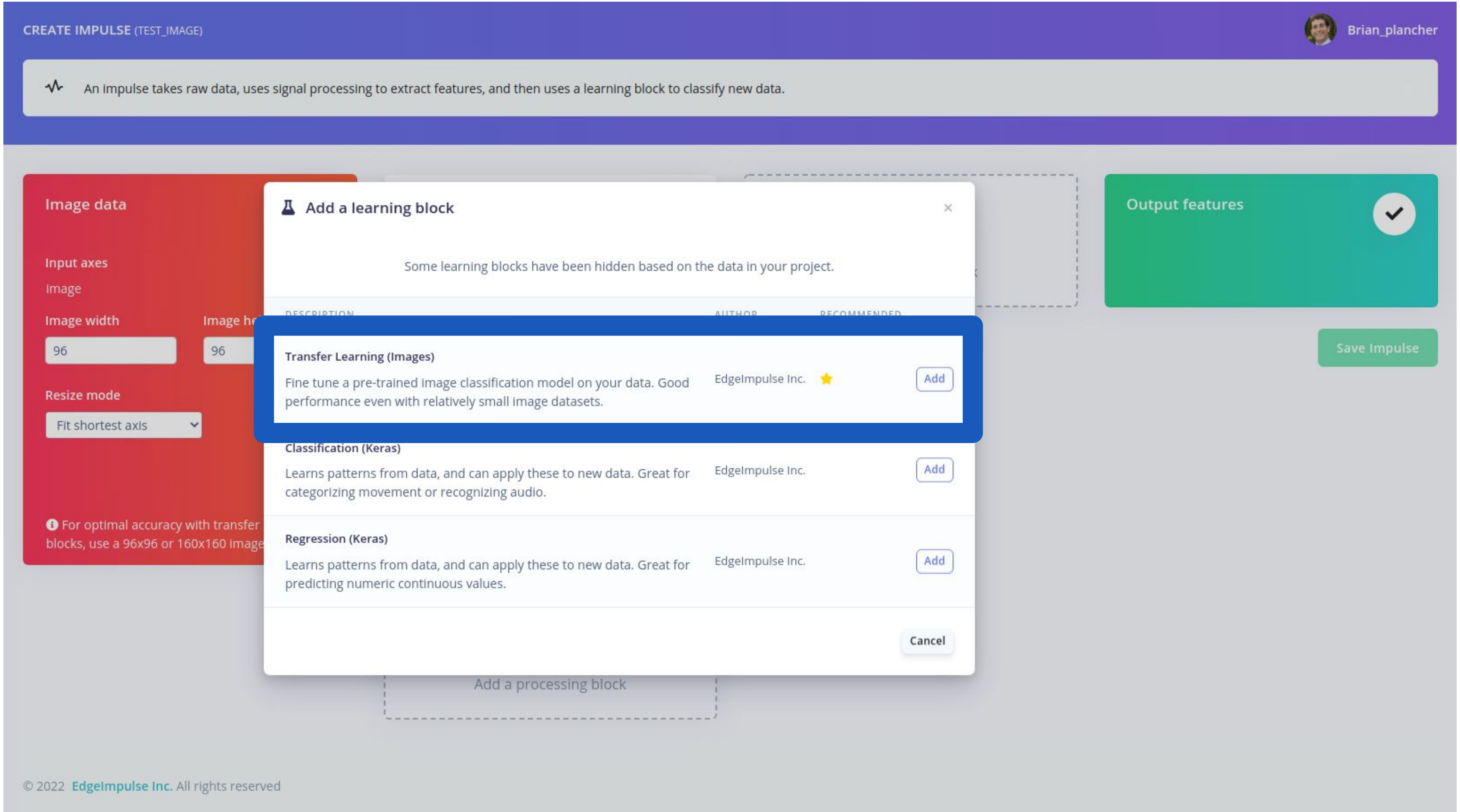
Add a processing block

Add a learning block

Output features

Save Impulse

© 2022 EdgeImpulse Inc. All rights reserved



CREATE IMPULSE (TEST_IMAGE)

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

Image data

Input axes: image

Image width: 96

Image he[ight]: 96

Resize mode: Fit shortest axis

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

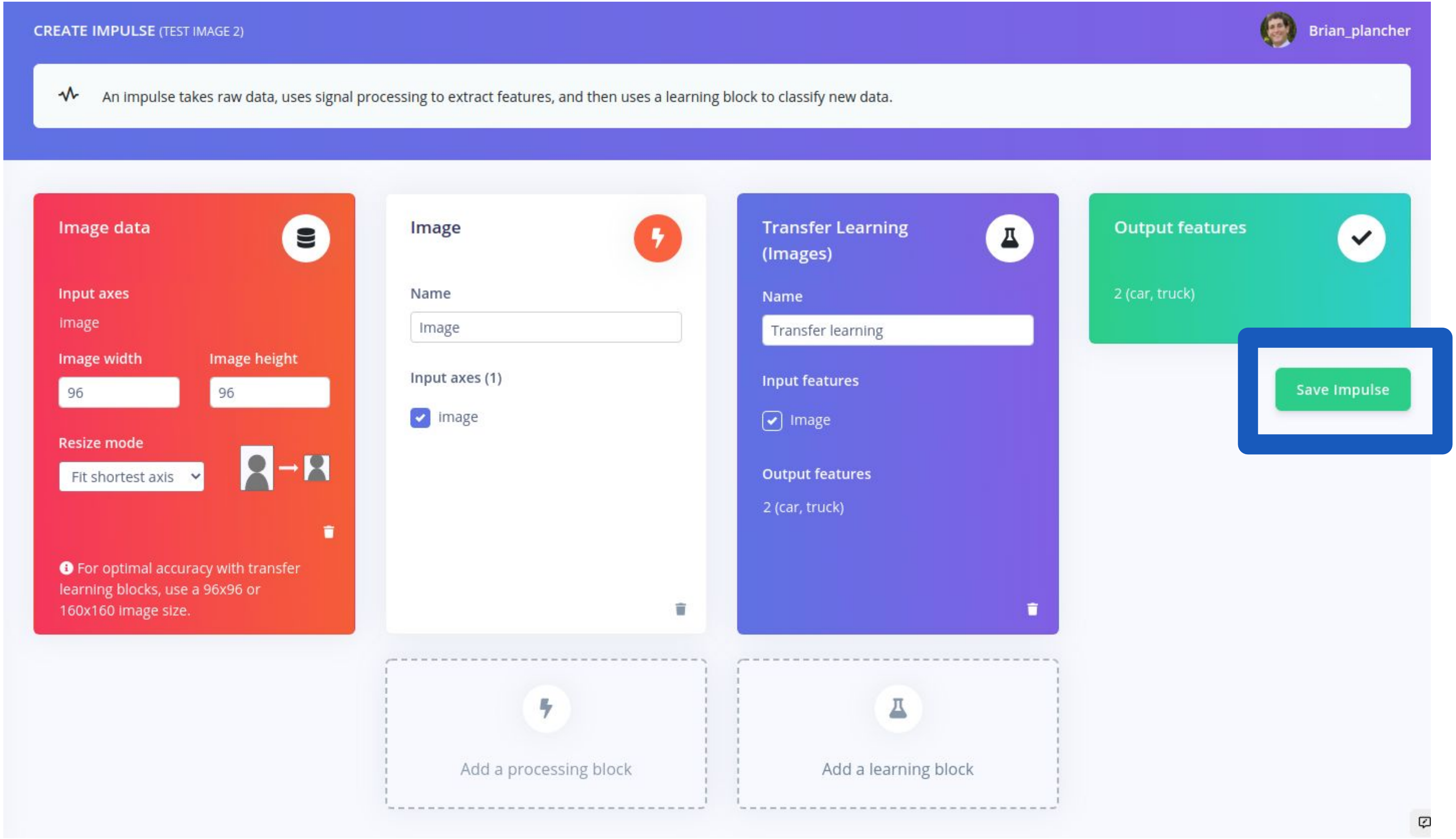
Output features

Add a learning block

Some learning blocks have been hidden based on the data in your project.

DESCRIPTION	AUTHOR	RECOMMENDED	
Transfer Learning (Images) Fine tune a pre-trained image classification model on your data. Good performance even with relatively small image datasets.	EdgeImpulse Inc.	★	Add
Classification (Keras) Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.	EdgeImpulse Inc.		Add
Regression (Keras) Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.	EdgeImpulse Inc.		Add

© 2022 EdgeImpulse Inc. All rights reserved



CREATE IMPULSE (TEST IMAGE 2)

Brian_plancher

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

Image data

Input axes: image

Image width: 96

Image height: 96

Resize mode: Fit shortest axis

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

Image

Name: Image

Input axes (1)

☑ image

Transfer Learning (Images)

Name: Transfer learning

Input features

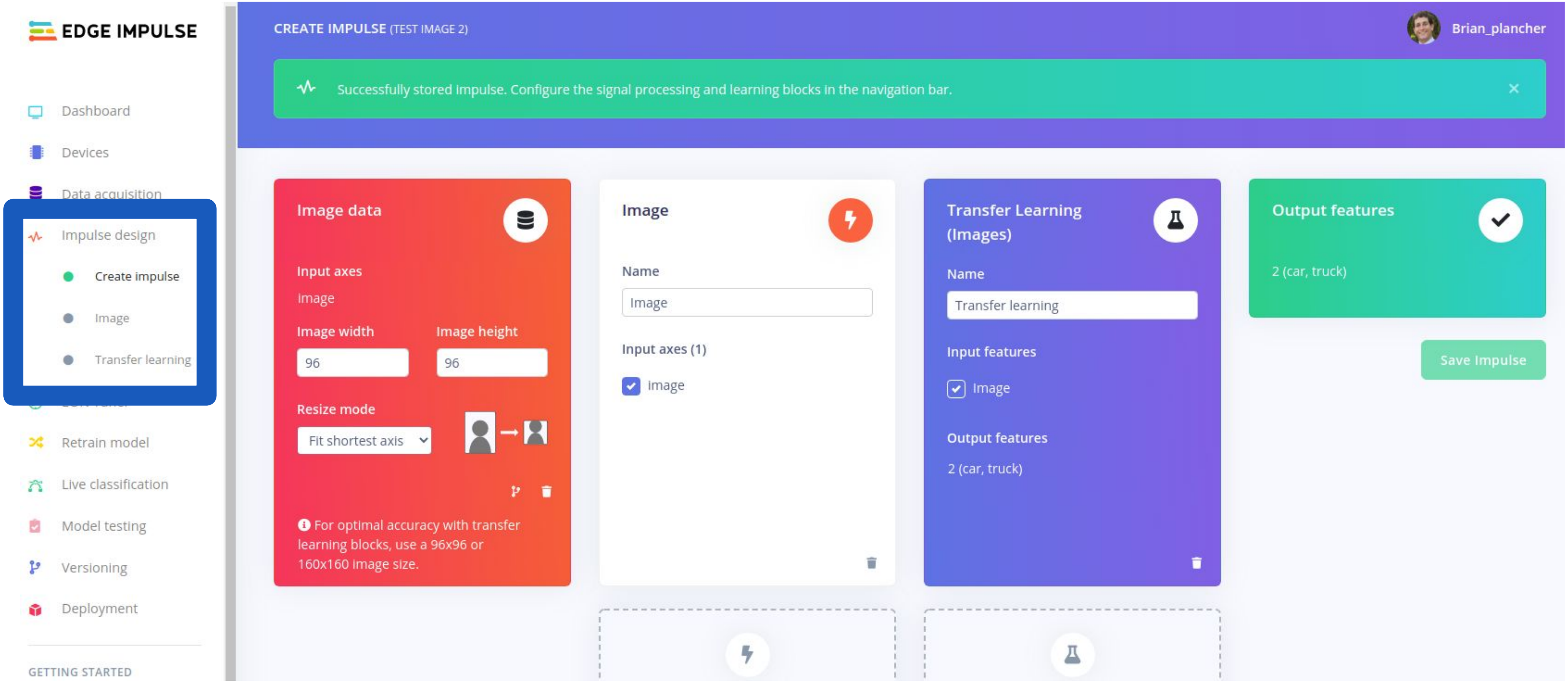
☑ Image

Output features

2 (car, truck)

Output features

2 (car, truck)



CREATE IMPULSE (TEST IMAGE 2)

Successfully stored impulse. Configure the signal processing and learning blocks in the navigation bar.

Image data

Input axes: image

Image width: 96

Image height: 96

Resize mode: Fit shortest axis

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

Image

Name: Image

Input axes (1)

☑ image

Transfer Learning (Images)

Name: Transfer learning

Input features

☑ Image

Output features

2 (car, truck)

Output features

2 (car, truck)

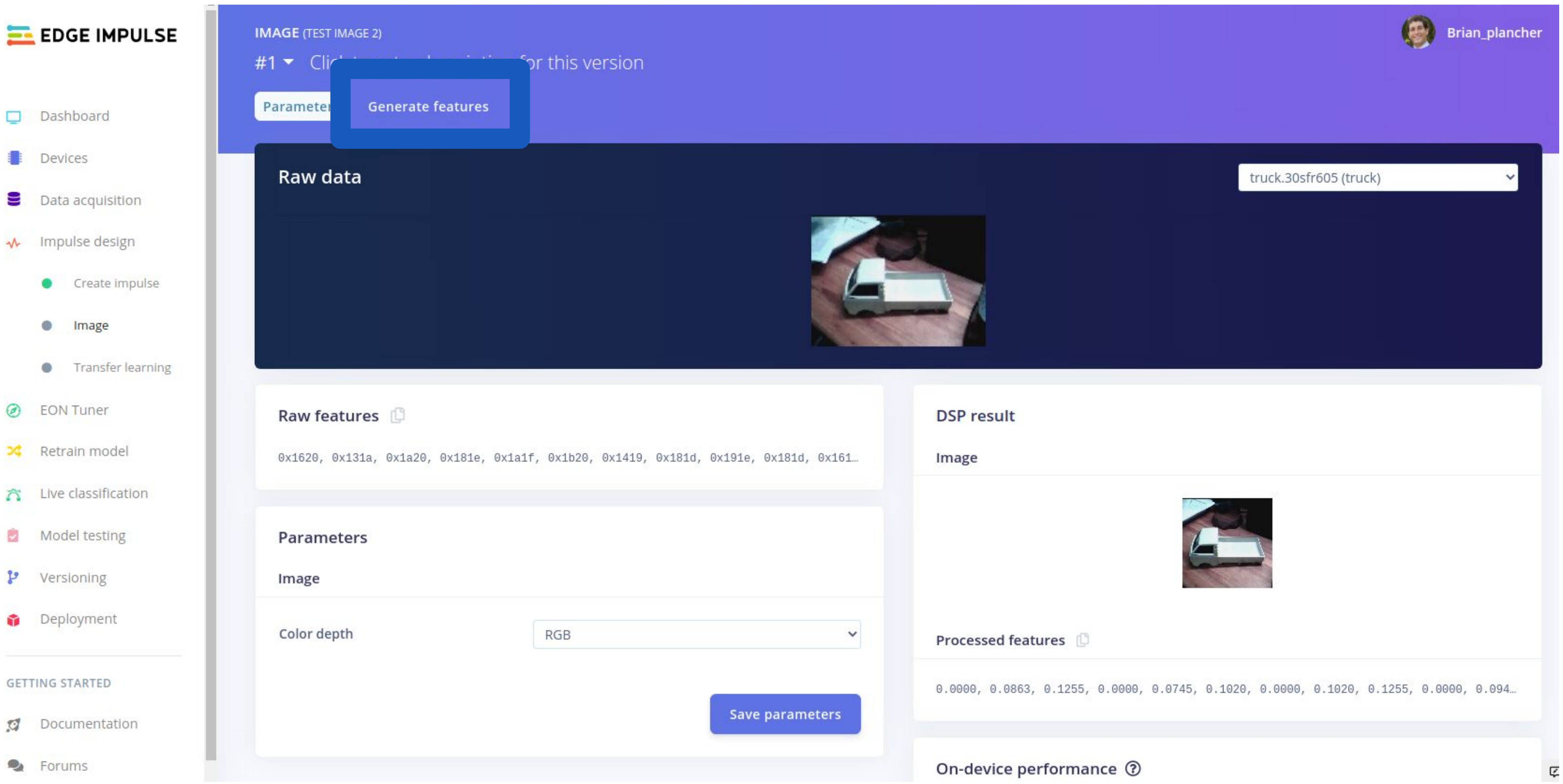


IMAGE (TEST IMAGE 2)

#1 Cli for this version

Raw data

truck.30sfr605 (truck)

Raw features

```
0x1620, 0x131a, 0x1a20, 0x181e, 0x1a1f, 0x1b20, 0x1419, 0x181d, 0x191e, 0x181d, 0x161…
```

Parameters

Image

Color depth: RGB

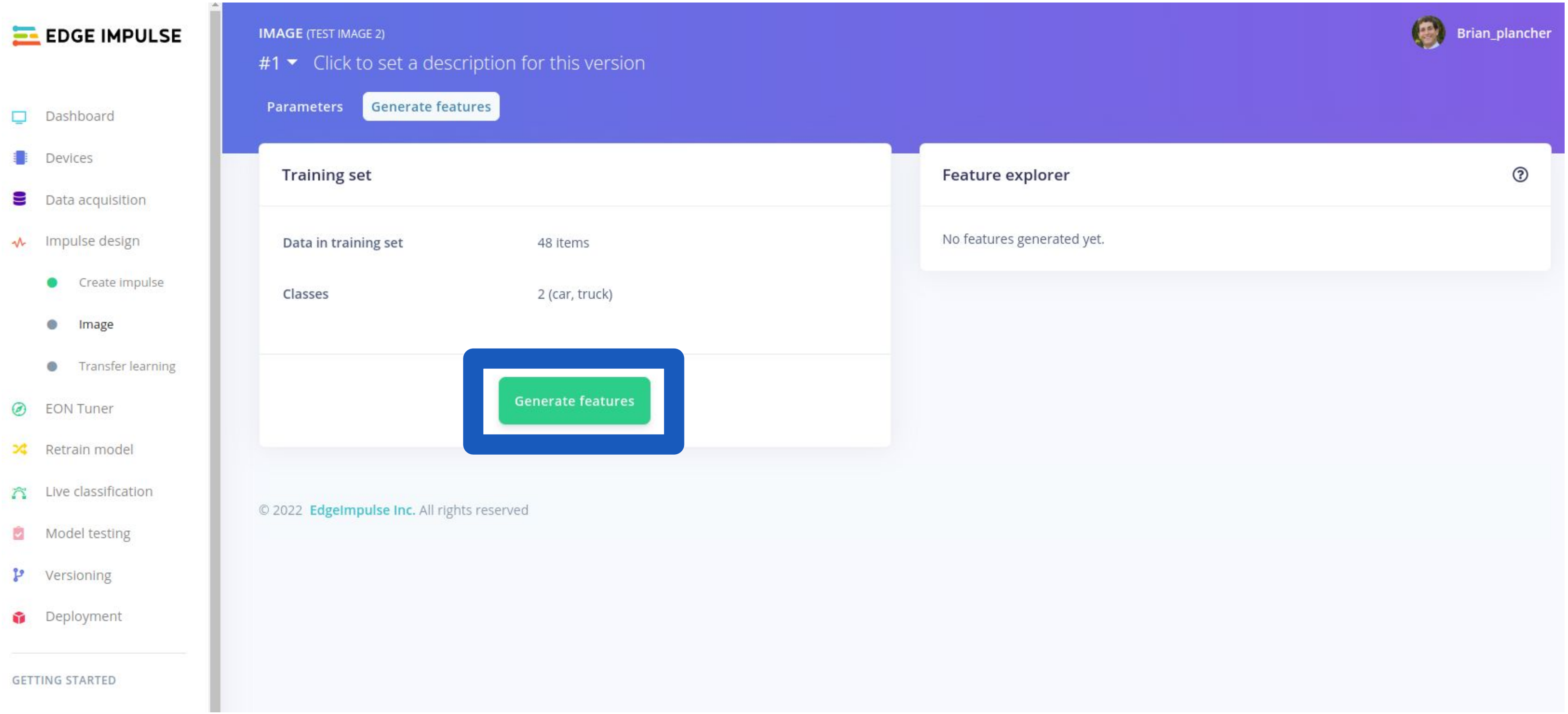
DSP result

Image

Processed features

```
0.0000, 0.0863, 0.1255, 0.0000, 0.0745, 0.1020, 0.0000, 0.1020, 0.1255, 0.0000, 0.094…
```

On-device performance



EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

Create impulse

Image

Transfer learning

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

IMAGE (TEST IMAGE 2)

#1 Click to set a description for this version

Parameters | Generate features

Brian_plancher

Training set

Data in training set	48 items
Classes	2 (car, truck)

Generate features

Feature explorer

No features generated yet.

© 2022 EdgeImpulse Inc. All rights reserved

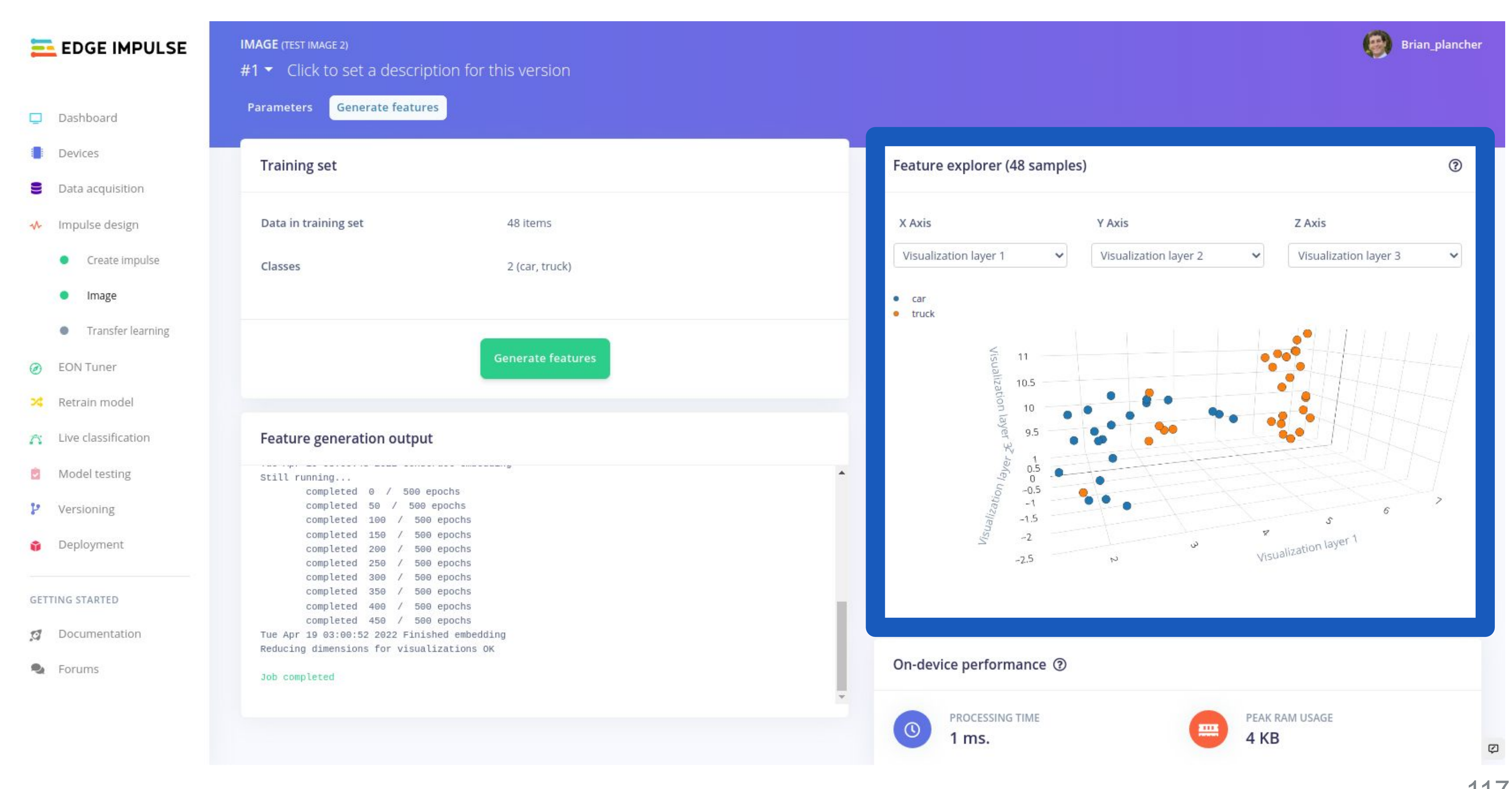


IMAGE (TEST IMAGE 2)

#1 Click to set a description for this version

Training set

Data in training set	48 items
Classes	2 (car, truck)

Feature generation output

```
Still running...
        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
        completed  400  /  500 epochs
        completed  450  /  500 epochs
Tue Apr 19 03:00:52 2022 Finished embedding
Reducing dimensions for visualizations OK

Job completed
```

Feature explorer (48 samples)

X Axis: Visualization layer 1

Y Axis: Visualization layer 2

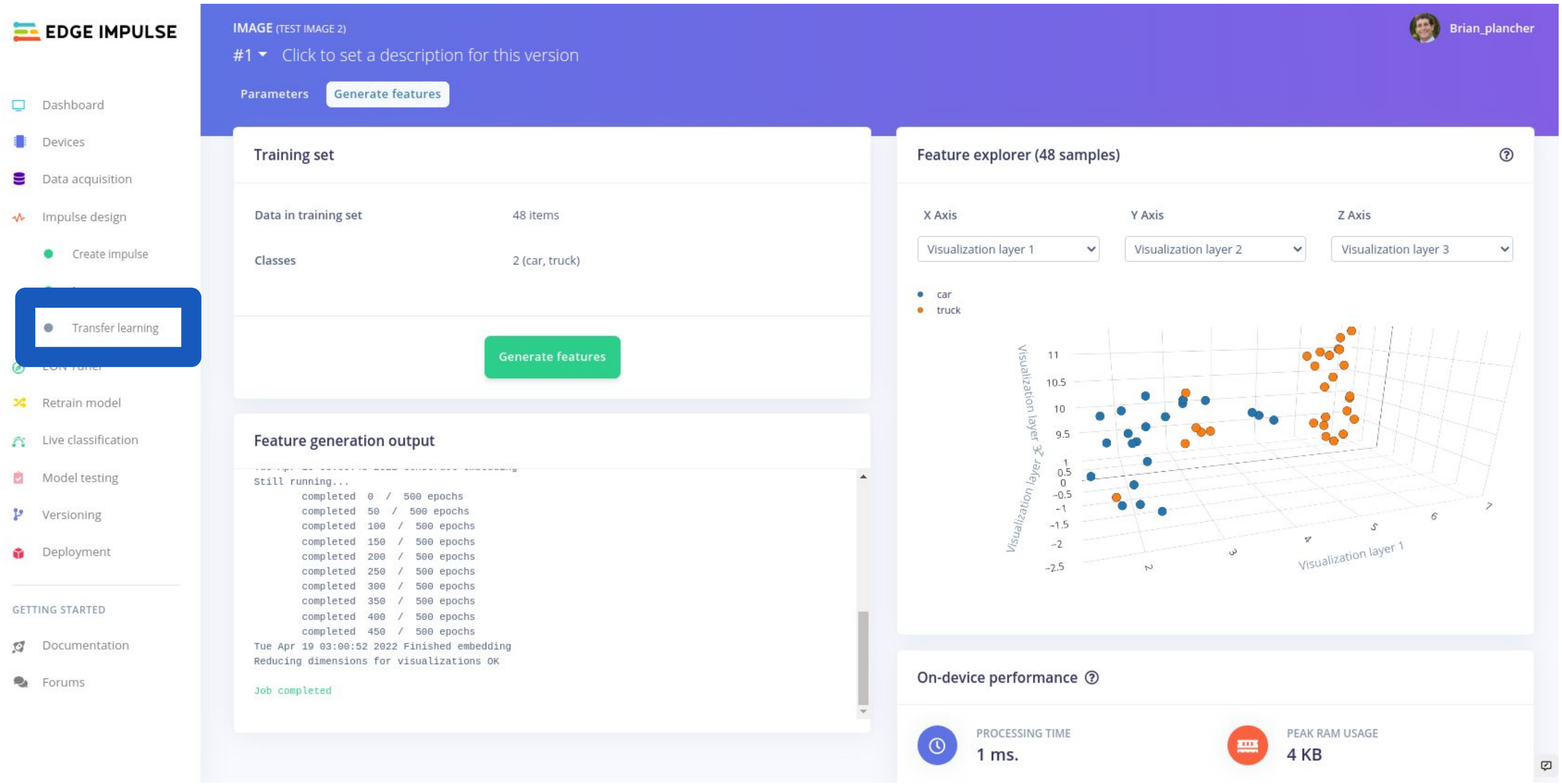
Z Axis: Visualization layer 3

● car
● truck

On-device performance

PROCESSING TIME
1 ms.

PEAK RAM USAGE
4 KB



Training set

Data in training set: 48 items

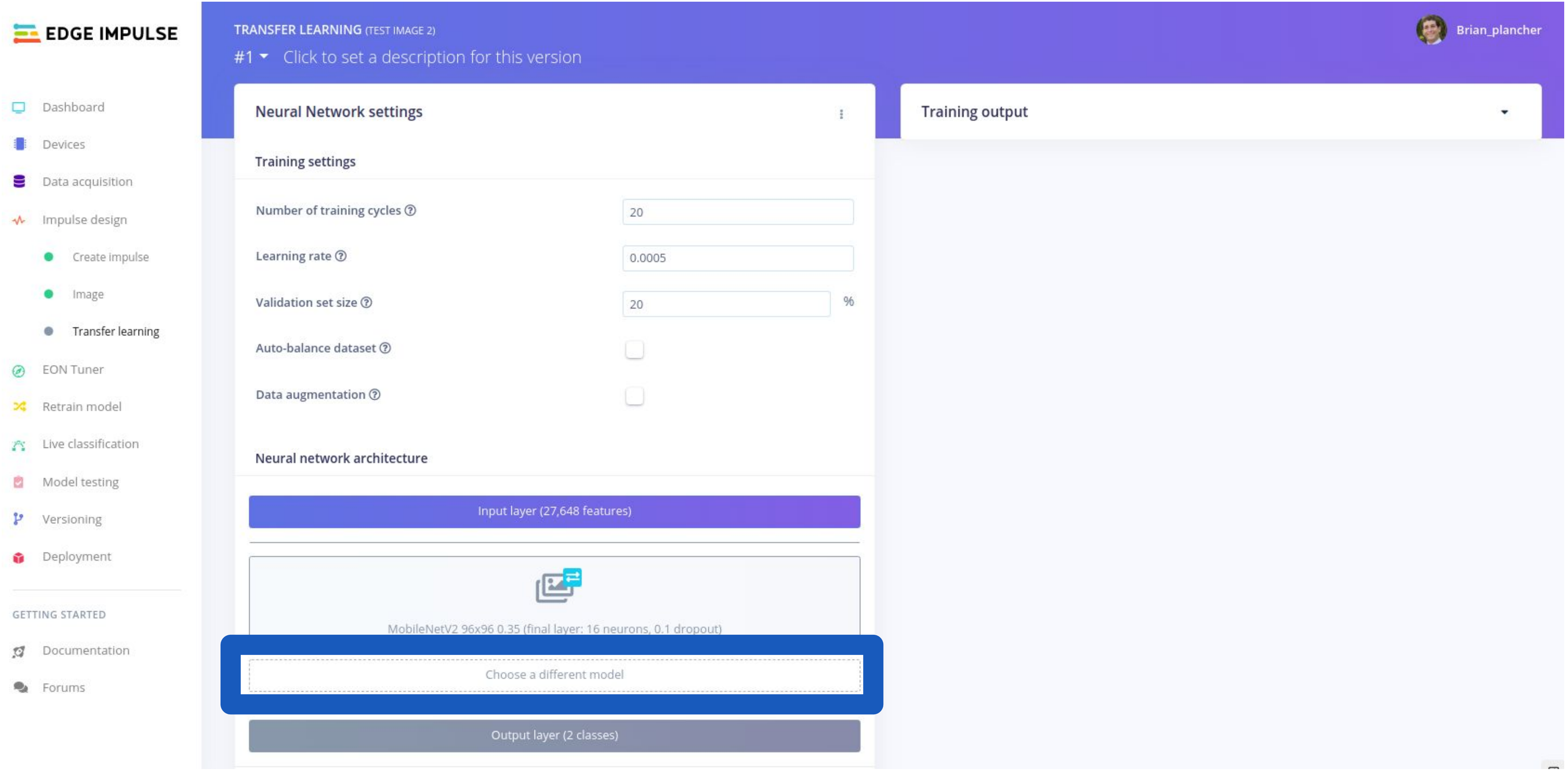
Classes: 2 (car, truck)

Feature generation output

```
Still running...
        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
        completed  400  /  500 epochs
        completed  450  /  500 epochs
Tue Apr 19 03:00:52 2022 Finished embedding
Reducing dimensions for visualizations OK

Job completed
```

PROCESSING TIME: 1 ms.

PEAK RAM USAGE: 4 KB



TRANSFER LEARNING (TEST IMAGE 2)

#1 Click to set a description for this version

Neural Network settings

Training settings

Number of training cycles: 20

Learning rate: 0.0005

Validation set size: 20 %

Auto-balance dataset: □

Data augmentation: □

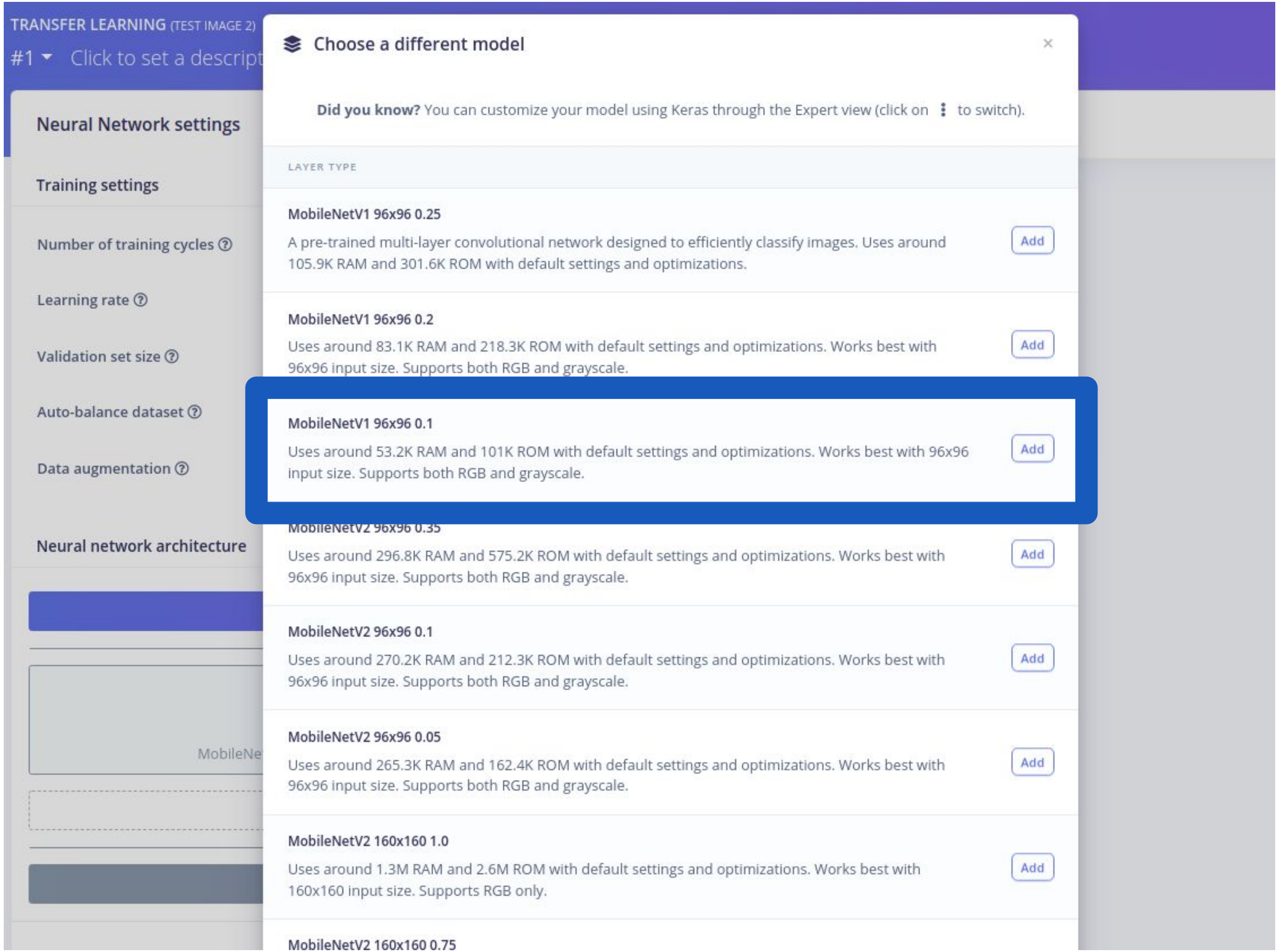
Neural network architecture

Input layer (27,648 features)

MobileNetV2 96x96 0.35 (final layer: 16 neurons, 0.1 dropout)

Output layer (2 classes)

Training output



Choose a different model

Did you know? You can customize your model using Keras through the Expert view (click on ⋮ to switch).

LAYER TYPE

MobileNetV1 96x96 0.25
A pre-trained multi-layer convolutional network designed to efficiently classify images. Uses around 105.9K RAM and 301.6K ROM with default settings and optimizations. Add

MobileNetV1 96x96 0.2
Uses around 83.1K RAM and 218.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV1 96x96 0.1
Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

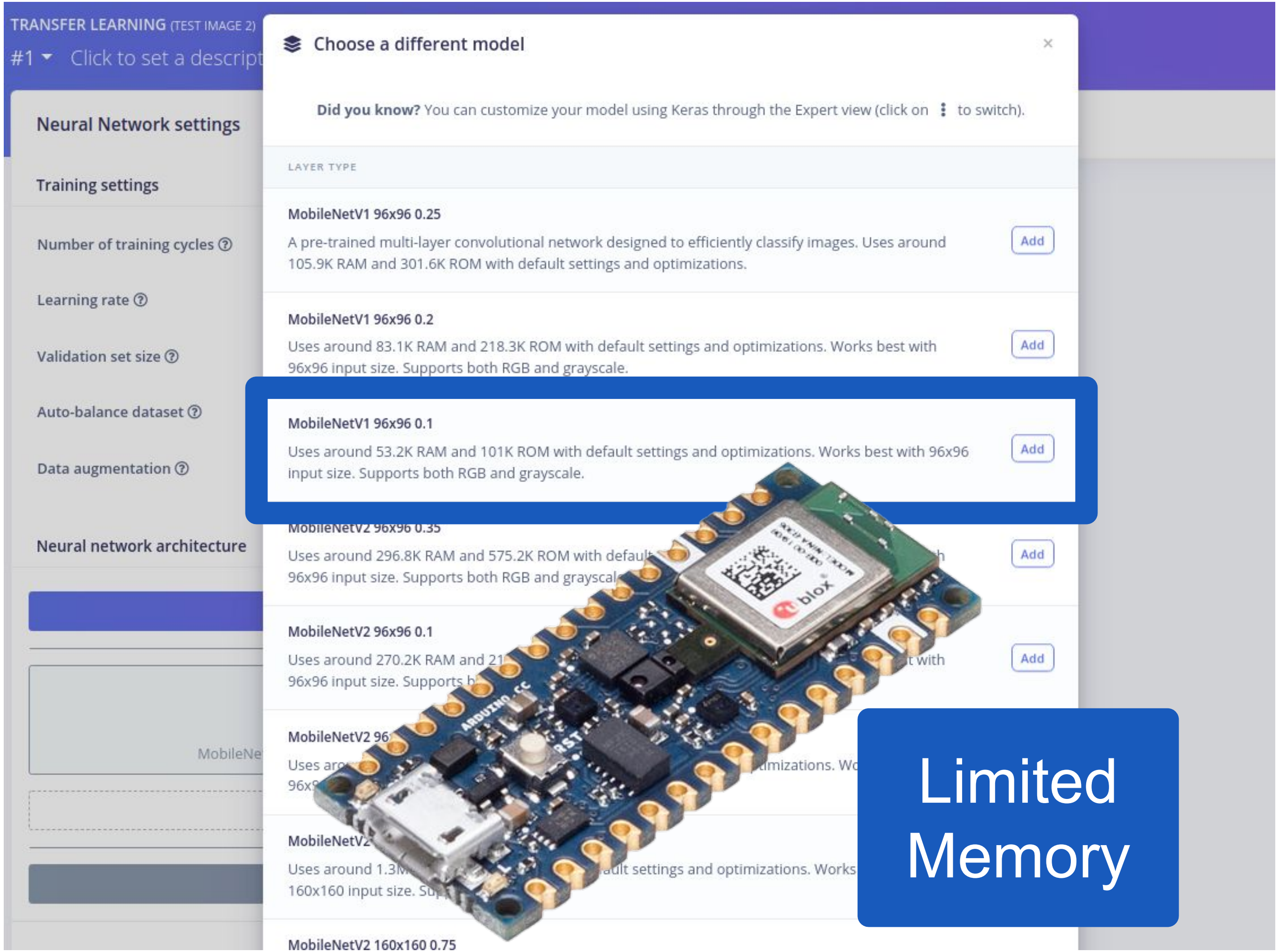
MobileNetV2 96x96 0.35
Uses around 296.8K RAM and 575.2K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV2 96x96 0.1
Uses around 270.2K RAM and 212.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV2 96x96 0.05
Uses around 265.3K RAM and 162.4K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV2 160x160 1.0
Uses around 1.3M RAM and 2.6M ROM with default settings and optimizations. Works best with 160x160 input size. Supports RGB only. Add

MobileNetV2 160x160 0.75



TRANSFER LEARNING (TEST IMAGE 2)

#1 ▾ Click to set a descript

Neural Network settings

Training settings

Number of training cycles ⓘ

Learning rate ⓘ

Validation set size ⓘ

Auto-balance dataset ⓘ

Data augmentation ⓘ

Neural network architecture

MobileNe

Choose a different model ×

Did you know? You can customize your model using Keras through the Expert view (click on ⋮ to switch).

LAYER TYPE

MobileNetV1 96x96 0.25

A pre-trained multi-layer convolutional network designed to efficiently classify images. Uses around 105.9K RAM and 301.6K ROM with default settings and optimizations. Add

MobileNetV1 96x96 0.2

Uses around 83.1K RAM and 218.3K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV1 96x96 0.1

Uses around 53.2K RAM and 101K ROM with default settings and optimizations. Works best with 96x96 input size. Supports both RGB and grayscale. Add

MobileNetV2 96x96 0.35

Uses around 296.8K RAM and 575.2K ROM with defaul h 96x96 input size. Supports both RGB and grayscal Add

MobileNetV2 96x96 0.1

Uses around 270.2K RAM and 21 t with 96x96 input size. Supports b Add

MobileNetV2 96

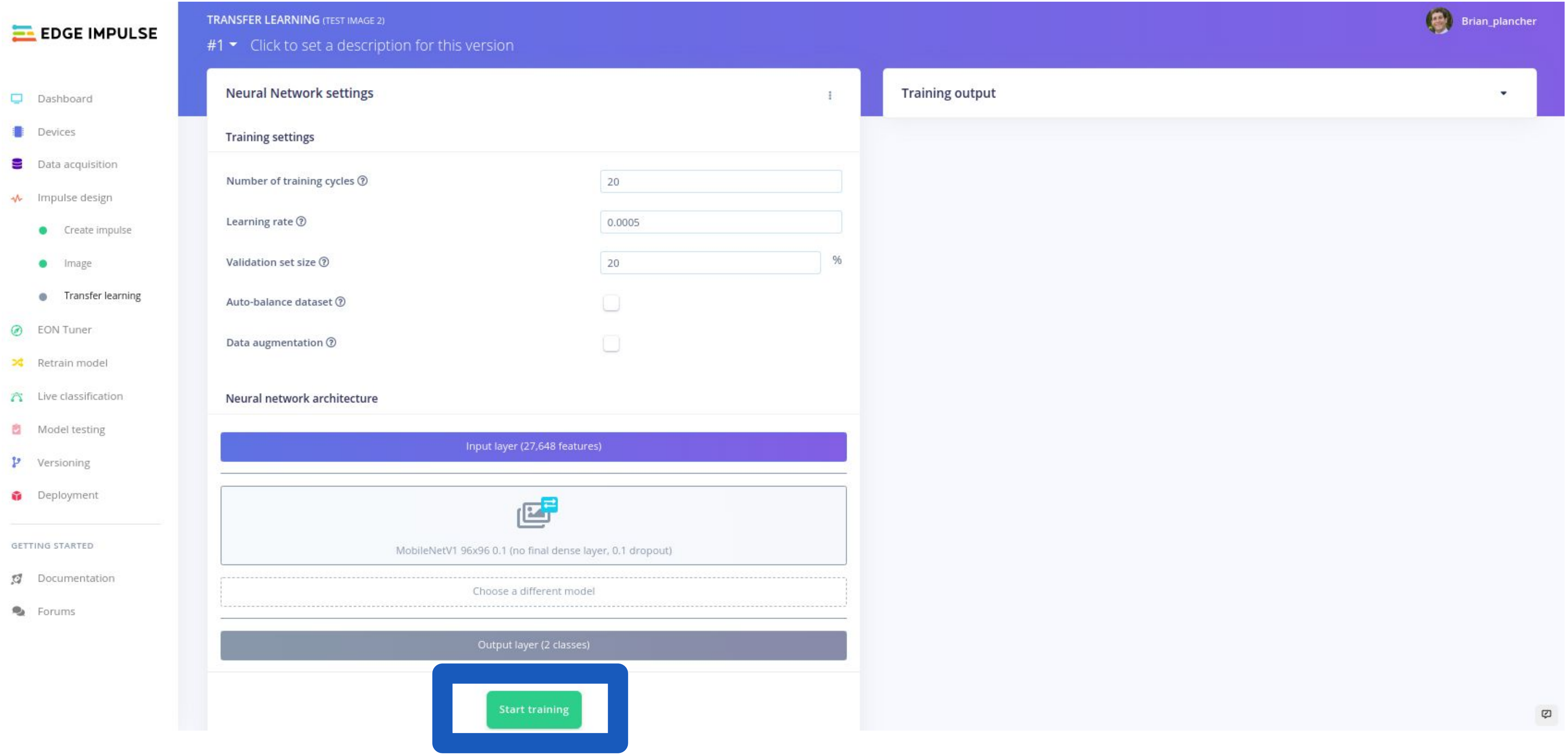
Uses ar timizations. Wo 96x

MobileNetV2

Uses around 1.3M ult settings and optimizations. Works 160x160 input size. Su

MobileNetV2 160x160 0.75

Limited Memory



Neural Network settings

Training settings

Number of training cycles	20
Learning rate	0.0005
Validation set size	20 %
Auto-balance dataset	□
Data augmentation	□

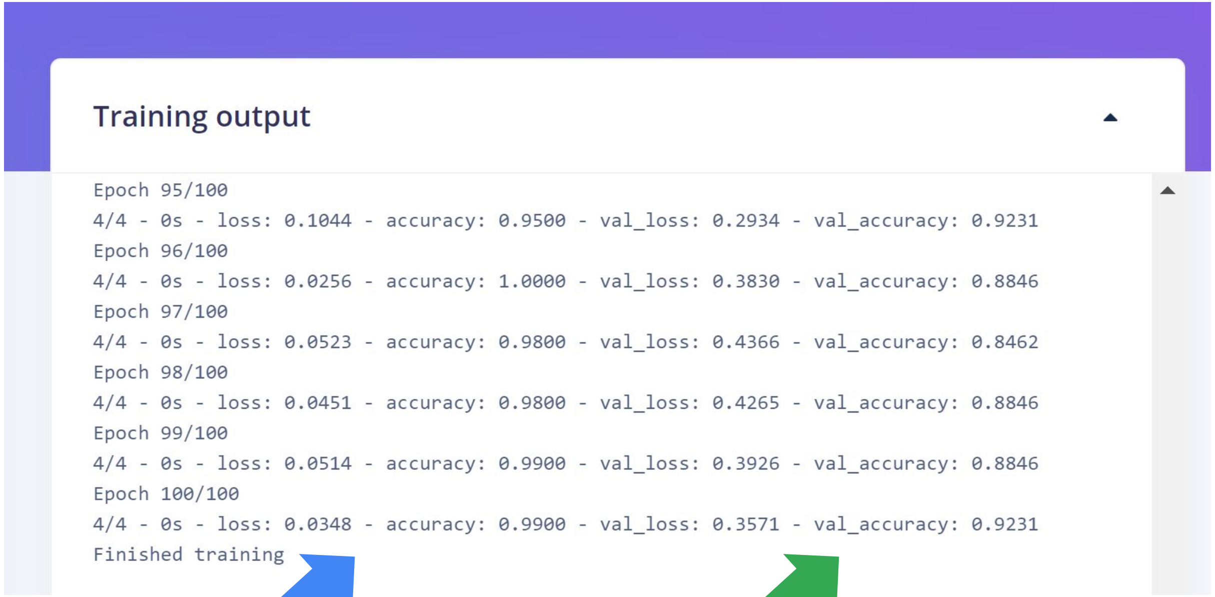
Neural network architecture

Input layer (27,648 features)

MobileNetV1 96x96 0.1 (no final dense layer, 0.1 dropout)

Choose a different model

Output layer (2 classes)



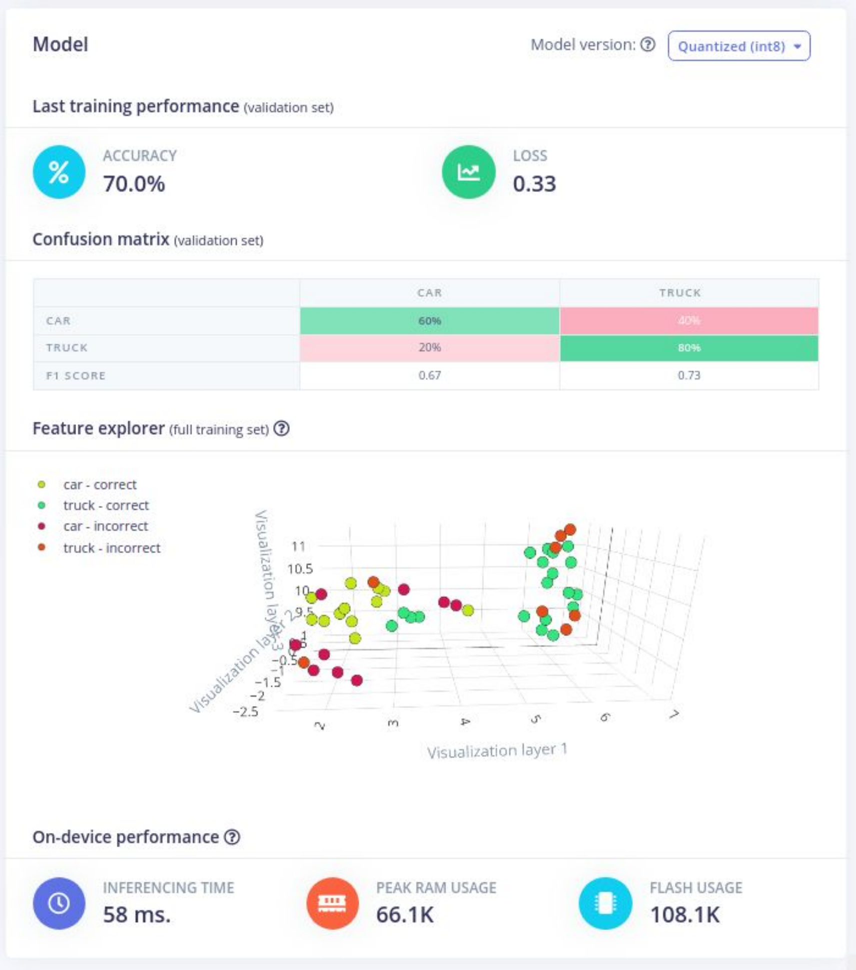
Training output

```
Epoch 95/100
4/4 - 0s - loss: 0.1044 - accuracy: 0.9500 - val_loss: 0.2934 - val_accuracy: 0.9231
Epoch 96/100
4/4 - 0s - loss: 0.0256 - accuracy: 1.0000 - val_loss: 0.3830 - val_accuracy: 0.8846
Epoch 97/100
4/4 - 0s - loss: 0.0523 - accuracy: 0.9800 - val_loss: 0.4366 - val_accuracy: 0.8462
Epoch 98/100
4/4 - 0s - loss: 0.0451 - accuracy: 0.9800 - val_loss: 0.4265 - val_accuracy: 0.8846
Epoch 99/100
4/4 - 0s - loss: 0.0514 - accuracy: 0.9900 - val_loss: 0.3926 - val_accuracy: 0.8846
Epoch 100/100
4/4 - 0s - loss: 0.0348 - accuracy: 0.9900 - val_loss: 0.3571 - val_accuracy: 0.9231
Finished training
```

Training Set

Validation Set

Final Test Accuracy



Model

Model version: ⓘ Quantized (int8) ▾

Last training performance (validation set)

ACCURACY 70.0%

LOSS 0.33

Confusion matrix (validation set)

	CAR	TRUCK
CAR	60%	40%
TRUCK	20%	80%
F1 SCORE	0.67	0.73

Feature explorer (full training set) ⓘ

- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

On-device performance ⓘ

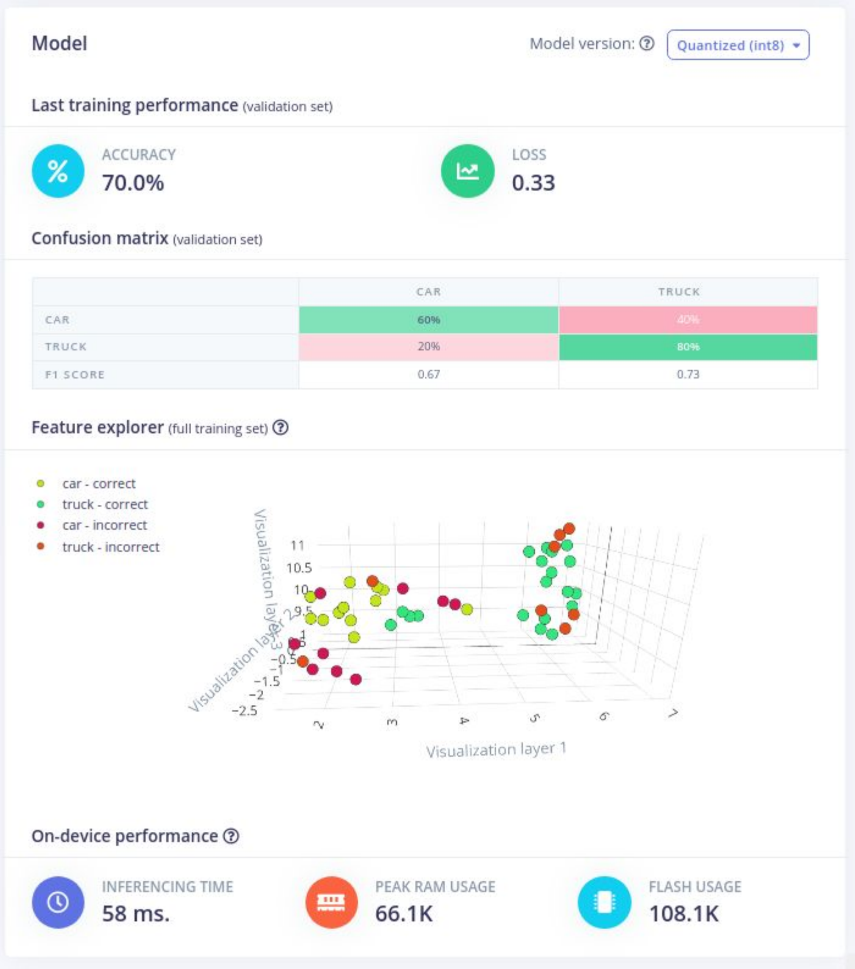
INFERENCING TIME 58 ms.

PEAK RAM USAGE 66.1K

FLASH USAGE 108.1K

Final Test Accuracy

Accuracy Breakdown



Model

Model version: ⓘ Quantized (int8) ▾

Last training performance (validation set)

ACCURACY
70.0%

LOSS
0.33

Confusion matrix (validation set)

	CAR	TRUCK
CAR	60%	40%
TRUCK	20%	80%
F1 SCORE	0.67	0.73

Feature explorer (full training set) ⓘ

- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

On-device performance ⓘ

INFERENCING TIME
58 ms.

PEAK RAM USAGE
66.1K

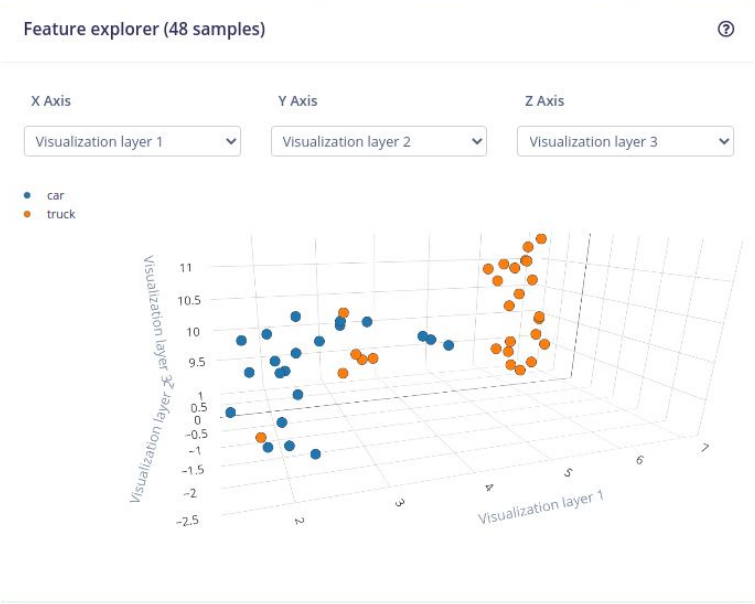
FLASH USAGE
108.1K

Confusion Matrix

	Actually Object 1	Actually Object 2
Predicted Object 1	**# of Correct Object 1**	***# of Error***
Predicted Object 2	***# of Error***	**# of Correct Object 2**

Final Test Accuracy

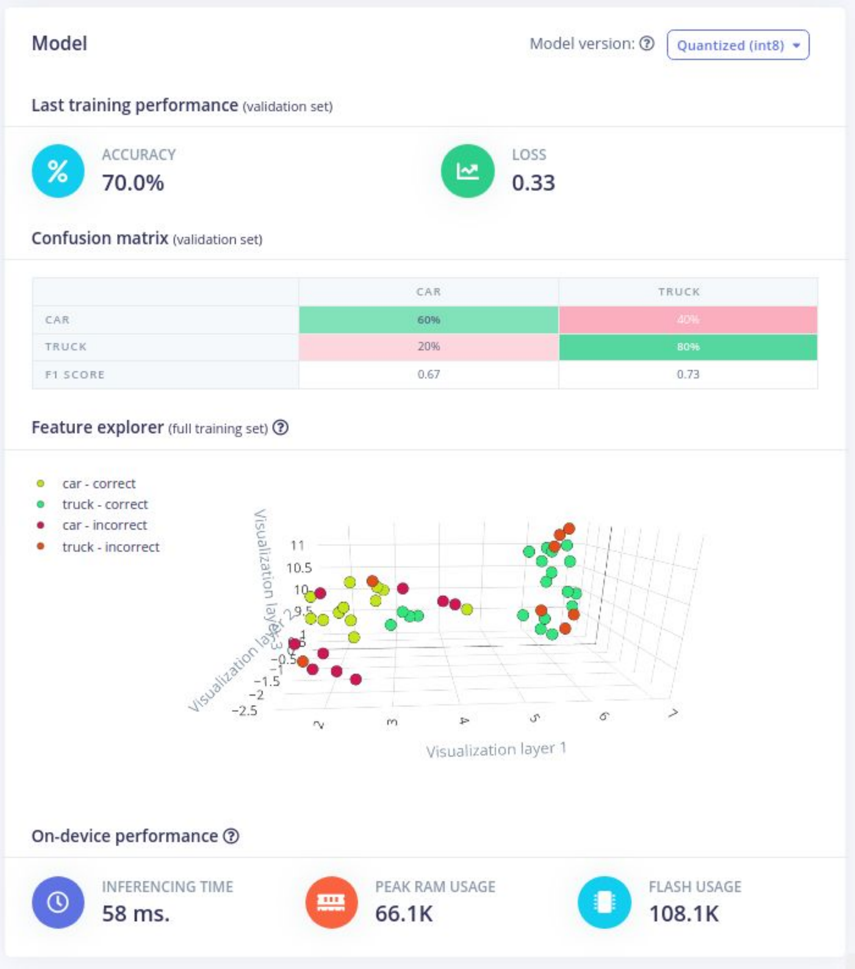
Accuracy Breakdown



Feature explorer (48 samples)

X Axis	Y Axis	Z Axis
Visualization layer 1	Visualization layer 2	Visualization layer 3

- ● car
- ● truck



Model

Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY 70.0%

LOSS 0.33

Confusion matrix (validation set)

	CAR	TRUCK
CAR	60%	40%
TRUCK	20%	80%
F1 SCORE	0.67	0.73

Feature explorer (full training set)

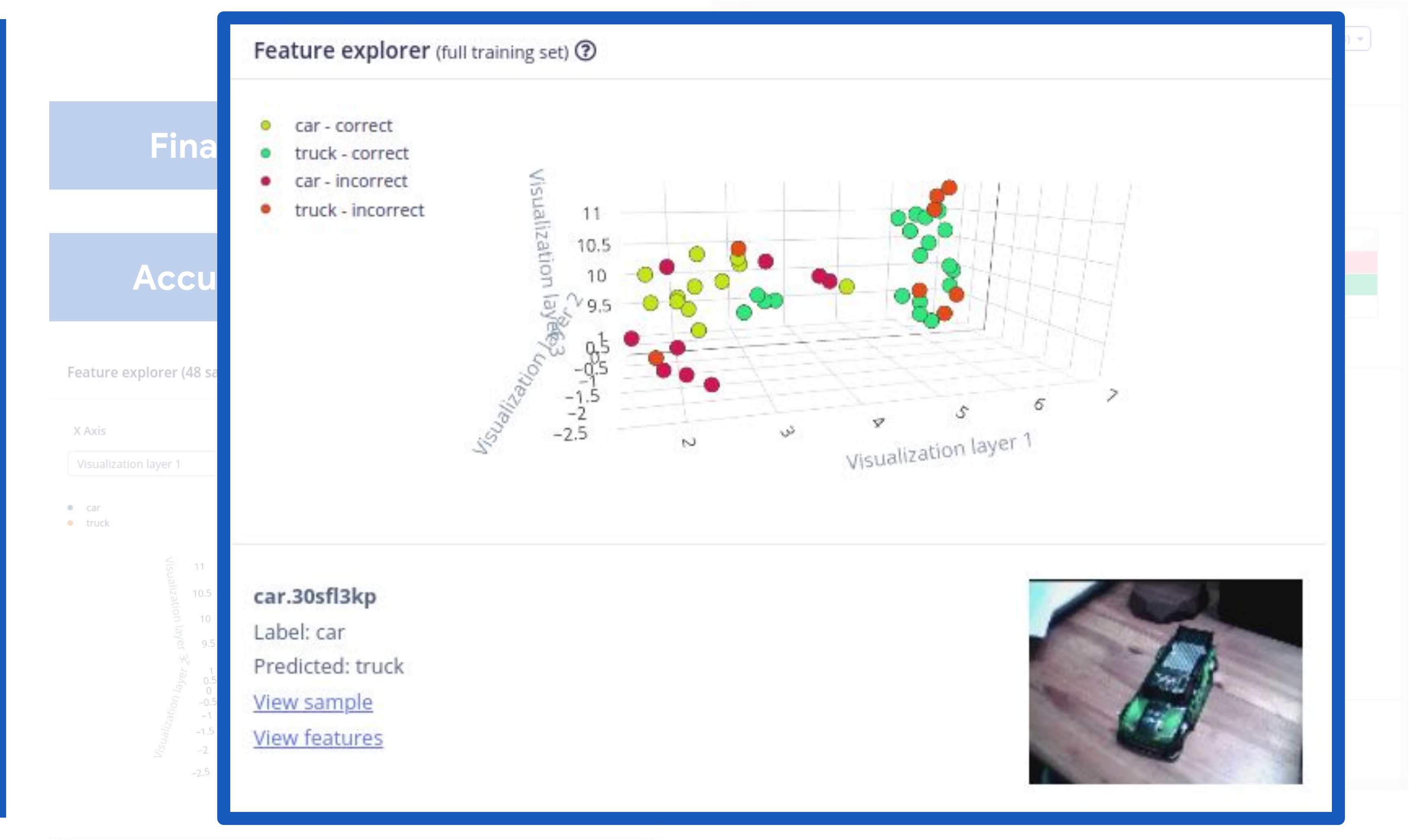
- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

On-device performance

INFERENCING TIME 58 ms.

PEAK RAM USAGE 66.1K

FLASH USAGE 108.1K



Feature explorer (full training set) ?

car.30sfl3kp

Label: car

Predicted: truck

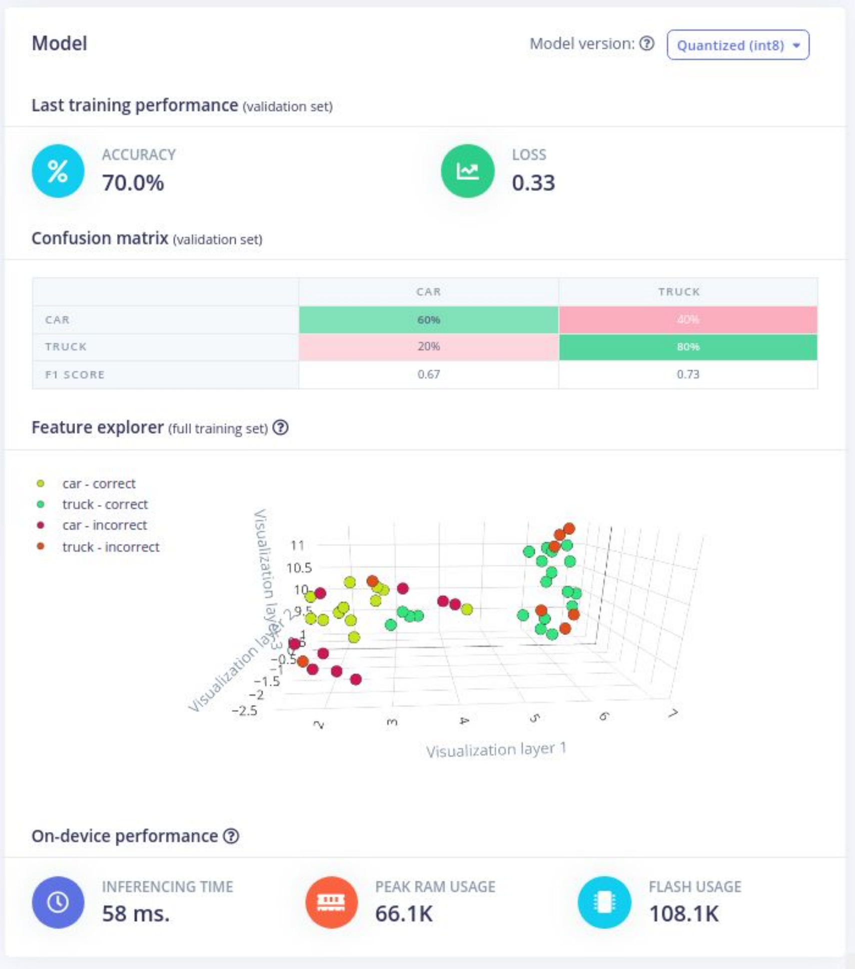
View sample

View features

Final Test Accuracy

Accuracy Breakdown

Memory and Time



Model

Model version: ⓘ Quantized (int8) ▾

Last training performance (validation set)

ACCURACY 70.0%

LOSS 0.33

Confusion matrix (validation set)

	CAR	TRUCK
CAR	60%	40%
TRUCK	20%	80%
F1 SCORE	0.67	0.73

Feature explorer (full training set) ⓘ

- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

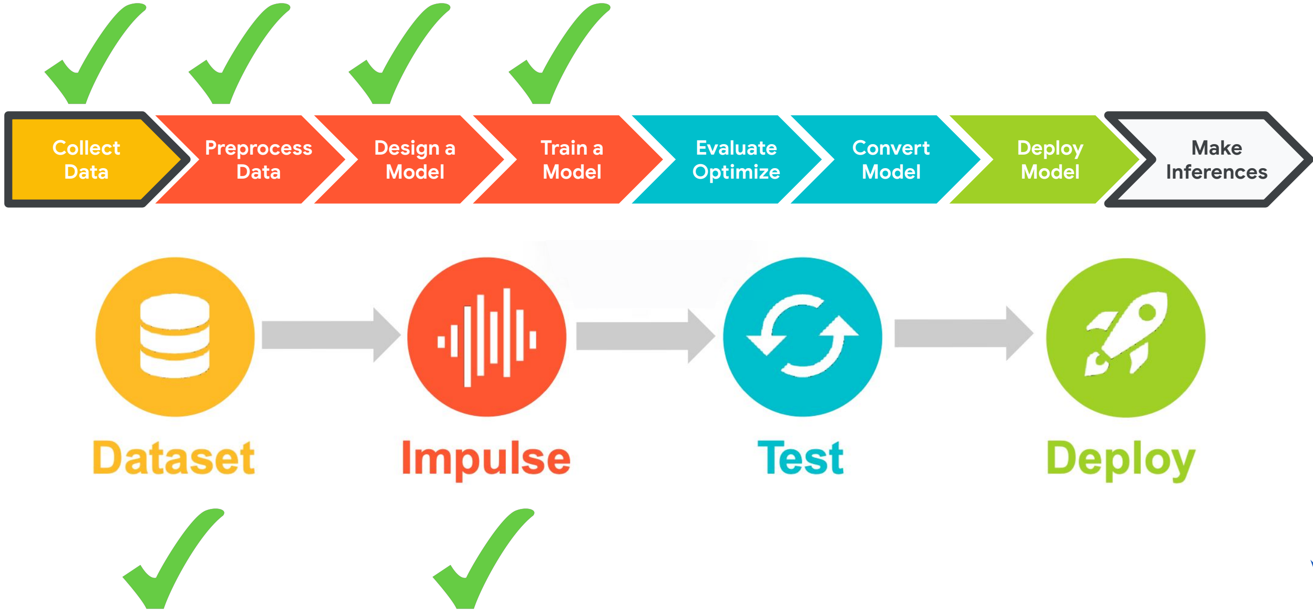
On-device performance ⓘ

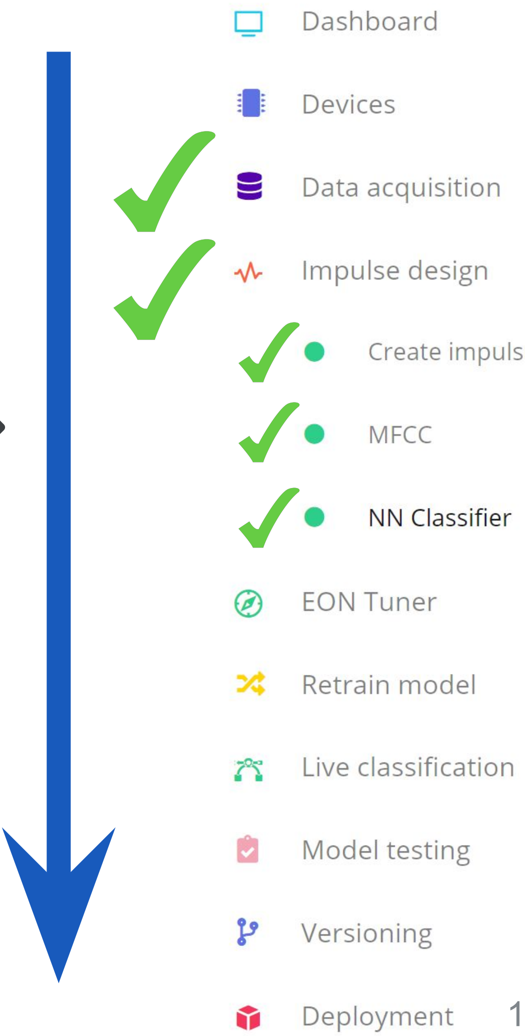
INFERENCING TIME 58 ms.

PEAK RAM USAGE 66.1K

FLASH USAGE 108.1K

Edge Impulse Project Dashboard

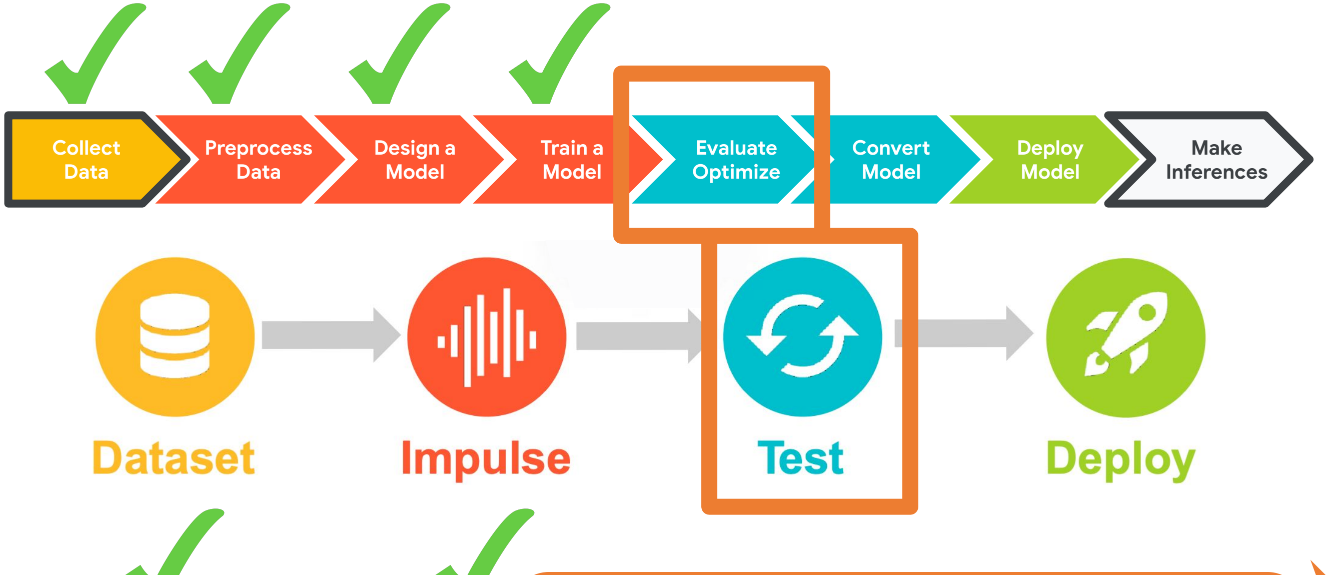




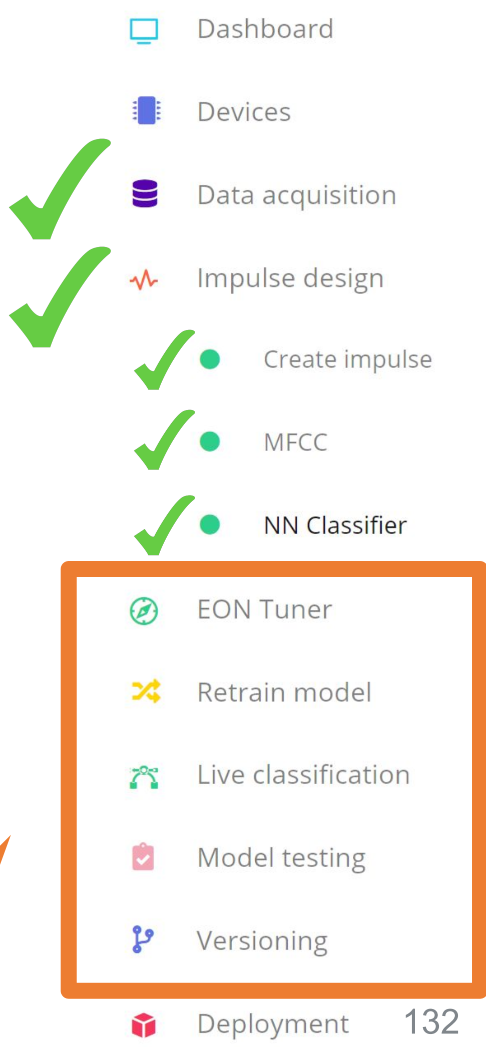
Today’s Agenda

- Introduction to Computer Vision
- Hands-on Computer Vision: Thing Translator
- Building an Object Detection Dataset
- Training our Model using Transfer Learning
- **Deploying our Model onto our Arduino**
- Summary

Edge Impulse Project Dashboard

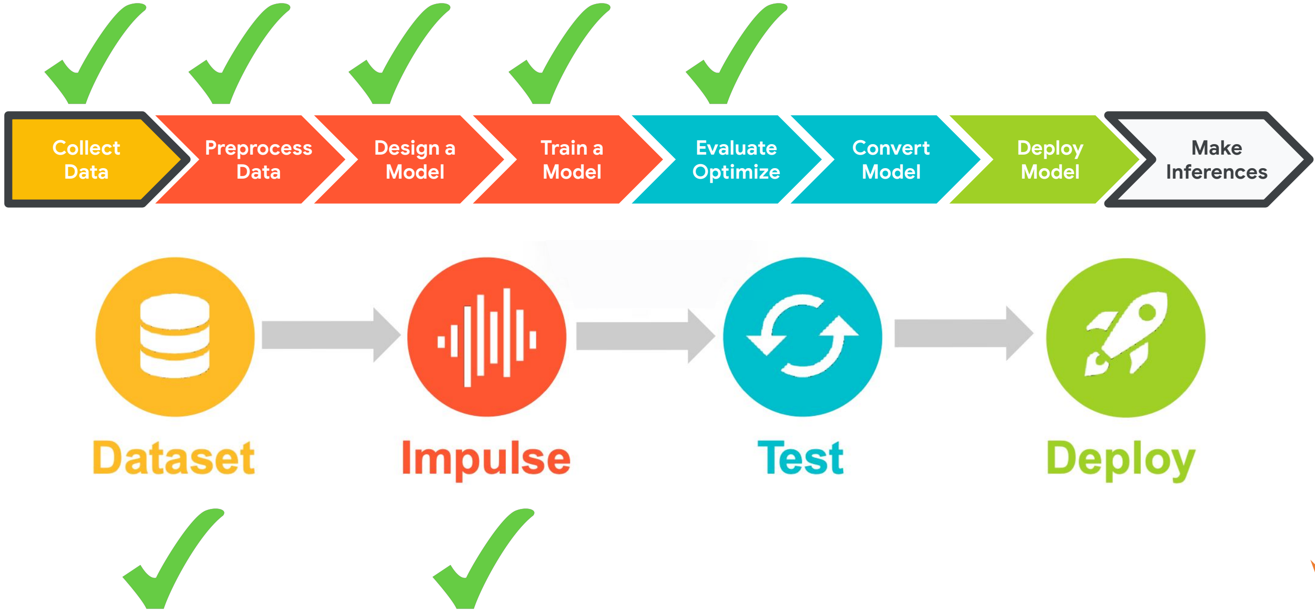


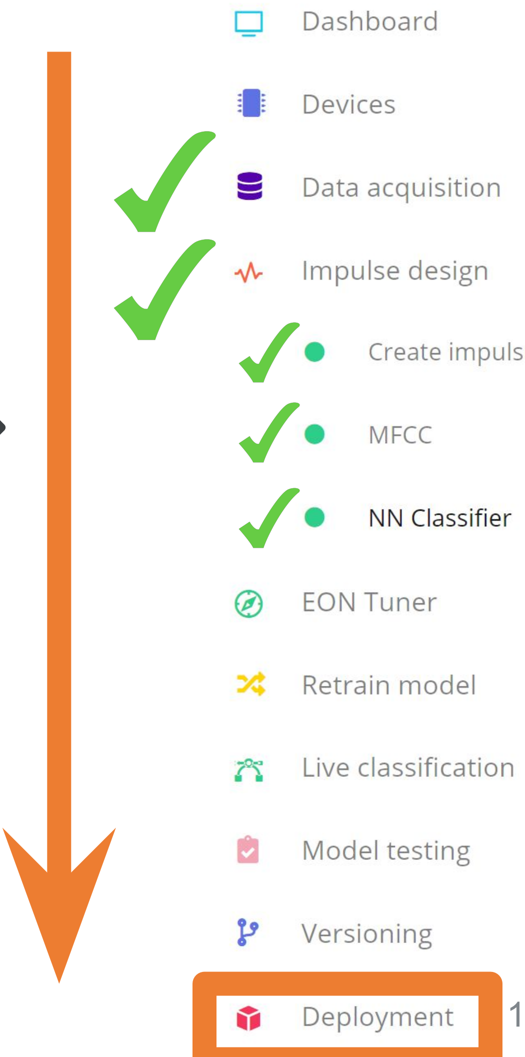
https://www.edgeimpulse.com/blog/introducing-the-eon-tuner-edge-impulses-new-automl-tool-for-embedded-machine-learning

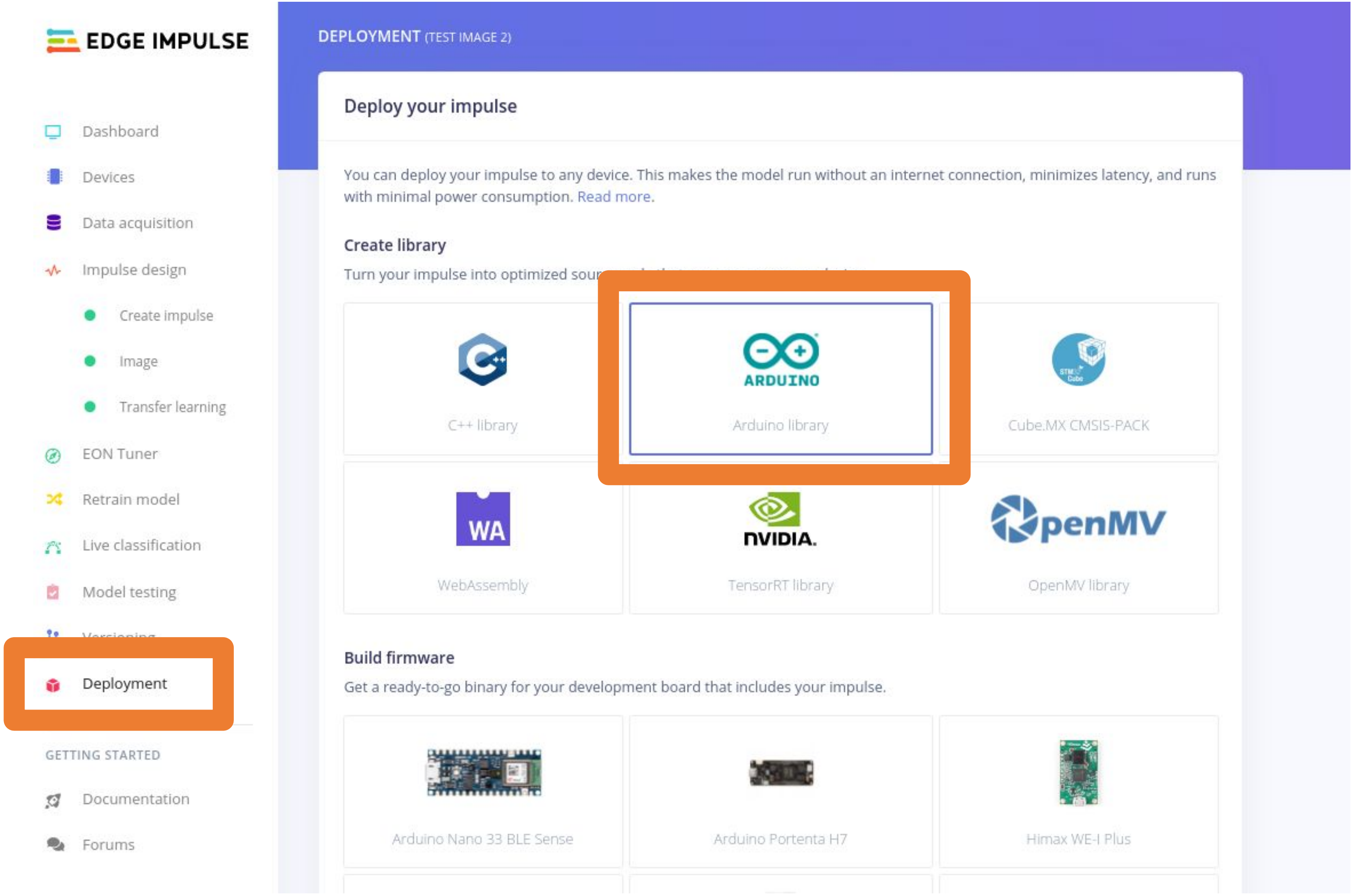


- Dashboard
- Devices
- ✓ Data acquisition
- ✓ Impulse design
  - ✓ Create impulse
  - ✓ MFCC
  - ✓ NN Classifier
- EON Tuner
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

Edge Impulse Project Dashboard







Deploy your impulse

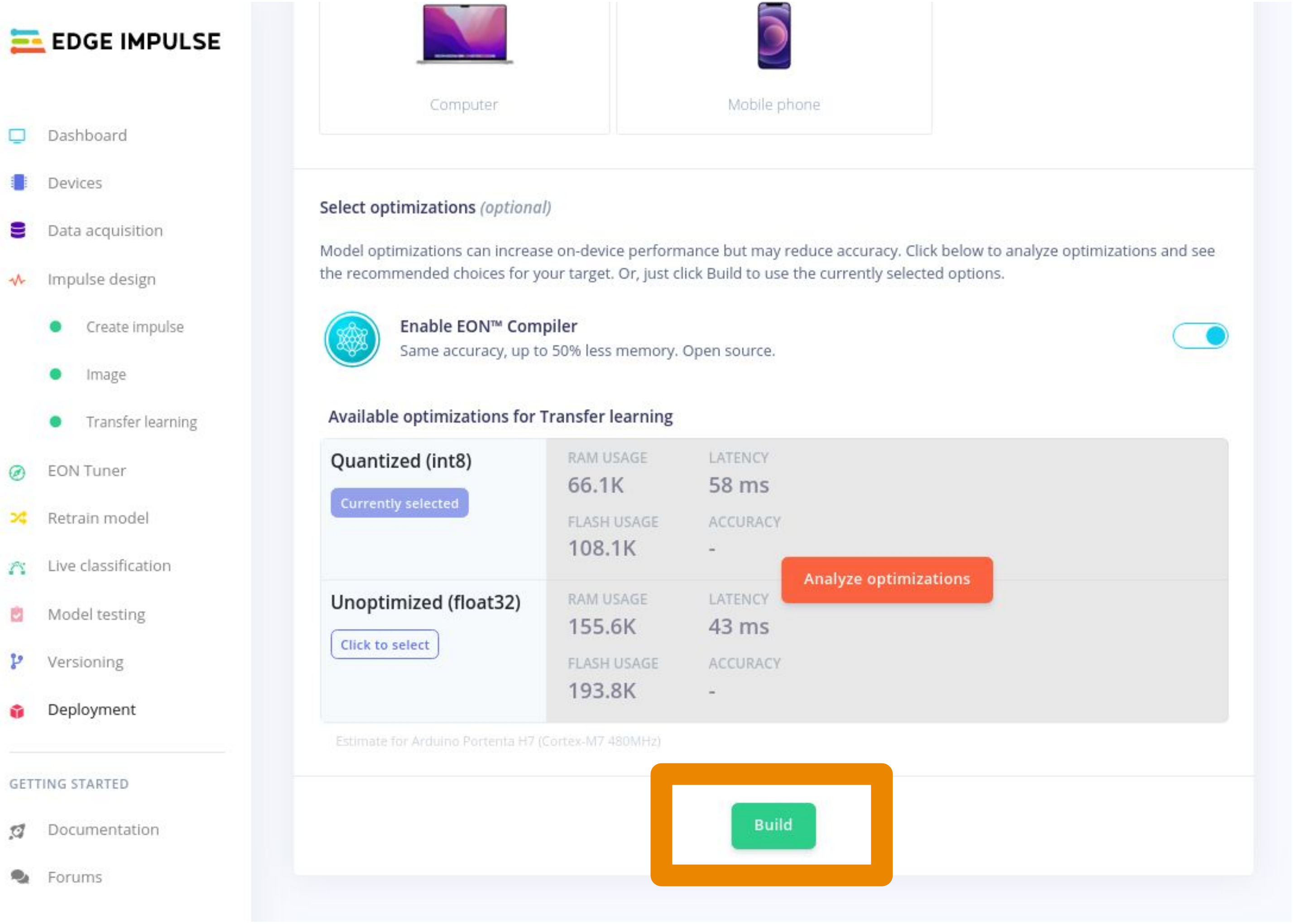
You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Create library

Turn your impulse into optimized sour

Build firmware

Get a ready-to-go binary for your development board that includes your impulse.



Select optimizations *(optional)*

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

Enable EON™ Compiler
Same accuracy, up to 50% less memory. Open source.

Available optimizations for Transfer learning

	RAM USAGE	LATENCY
Quantized (int8) Currently selected	66.1K	58 ms
	FLASH USAGE	ACCURACY
	108.1K	-
Unoptimized (float32) Click to select	RAM USAGE	LATENCY
	155.6K	43 ms
	FLASH USAGE	ACCURACY
	193.8K	-

Estimate for Arduino Portenta H7 (Cortex-M7 480MHz)

Quantization

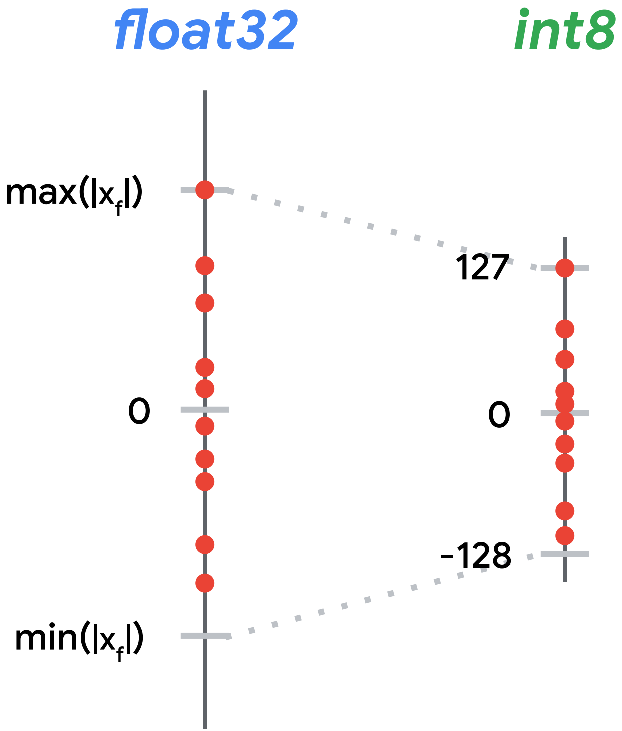
Reduces the precision of numbers used in a model which results in:

- smaller model size
- faster computation

```
max: 3.40282e+38
min: 1.17549e-38
```

Reducing the Precision

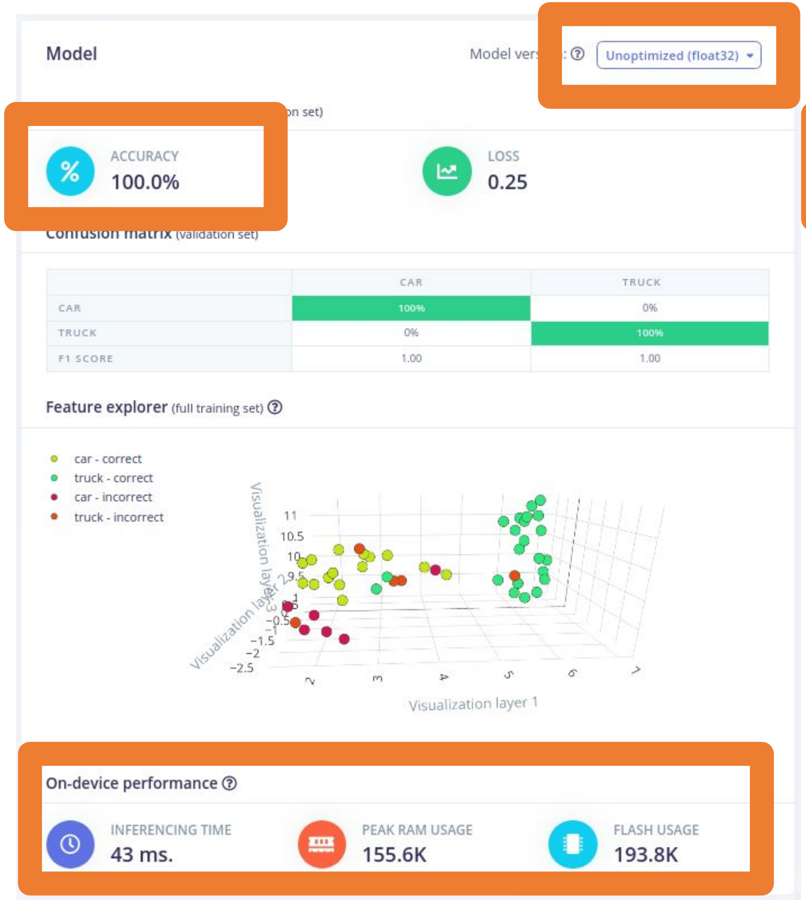
4 bytes per model parameter



1 byte per model parameter

Tradeoff

	Floating-point Baseline	After Quantization	Accuracy Drop
MobileNet v1 1.0 224	71.03%	69.57%	▼1.46%
MobileNet v2 1.0 224	70.77%	70.20%	▼0.57%
Resnet v1 50	76.30%	75.95%	▼0.35%



Model

Model vers : ⓘ Unoptimized (float32) ▾

on set)

ACCURACY 100.0%

LOSS 0.25

Confusion matrix (validation set)

	CAR	TRUCK
CAR	100%	0%
TRUCK	0%	100%
F1 SCORE	1.00	1.00

Feature explorer (full training set) ⓘ

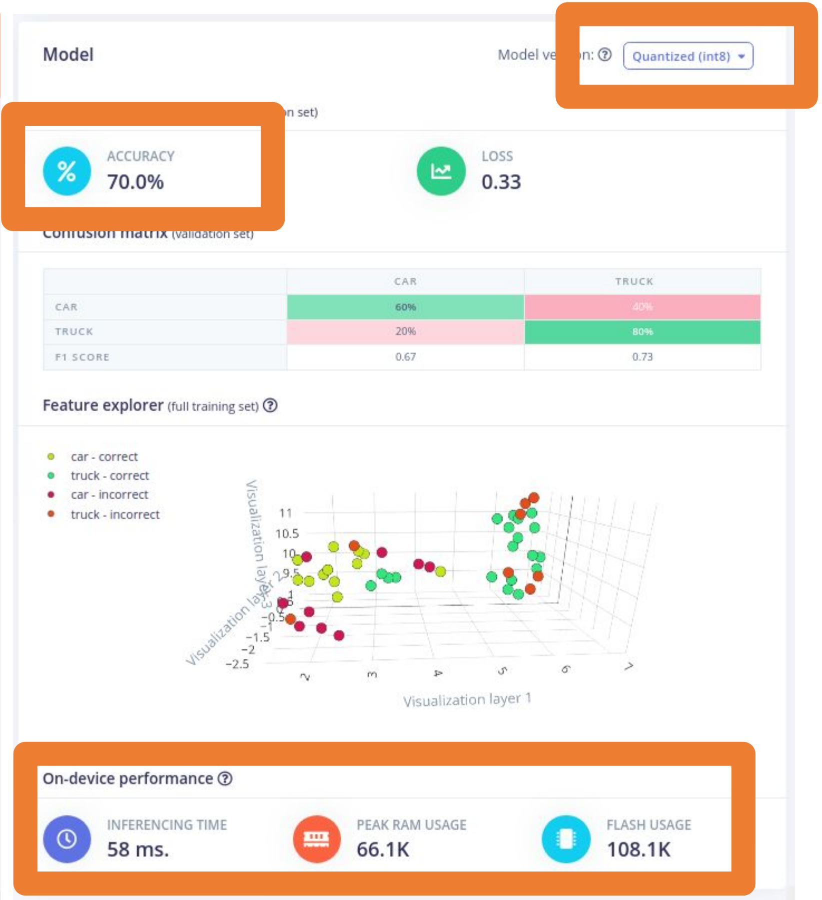
- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

On-device performance ⓘ

INFERENCING TIME 43 ms.

PEAK RAM USAGE 155.6K

FLASH USAGE 193.8K



Model

Model ve n: ⓘ Quantized (int8) ▾

n set)

ACCURACY 70.0%

LOSS 0.33

Confusion matrix (validation set)

	CAR	TRUCK
CAR	60%	40%
TRUCK	20%	80%
F1 SCORE	0.67	0.73

Feature explorer (full training set) ⓘ

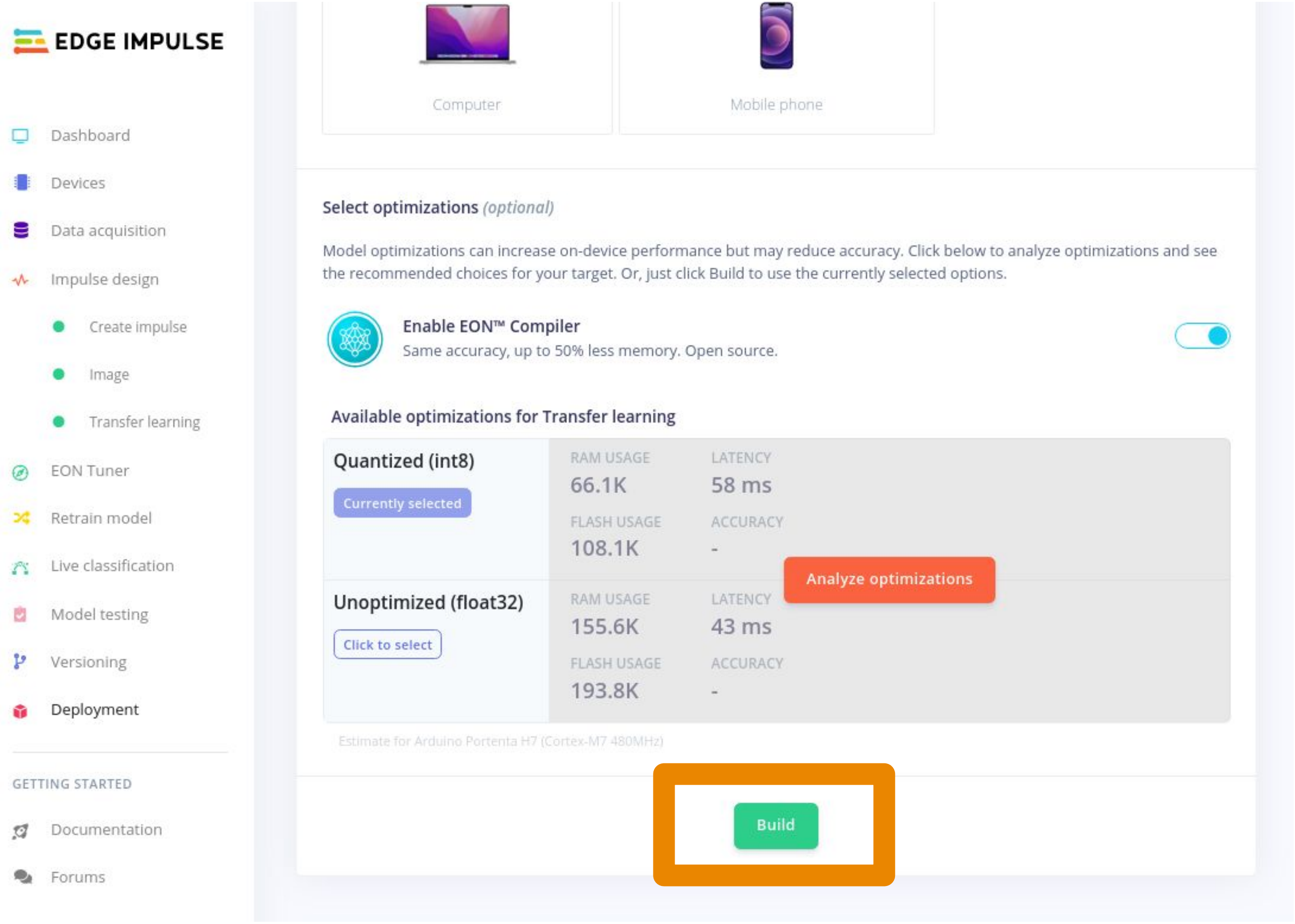
- ● car - correct
- ● truck - correct
- ● car - incorrect
- ● truck - incorrect

On-device performance ⓘ

INFERENCING TIME 58 ms.

PEAK RAM USAGE 66.1K

FLASH USAGE 108.1K



Select optimizations *(optional)*

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

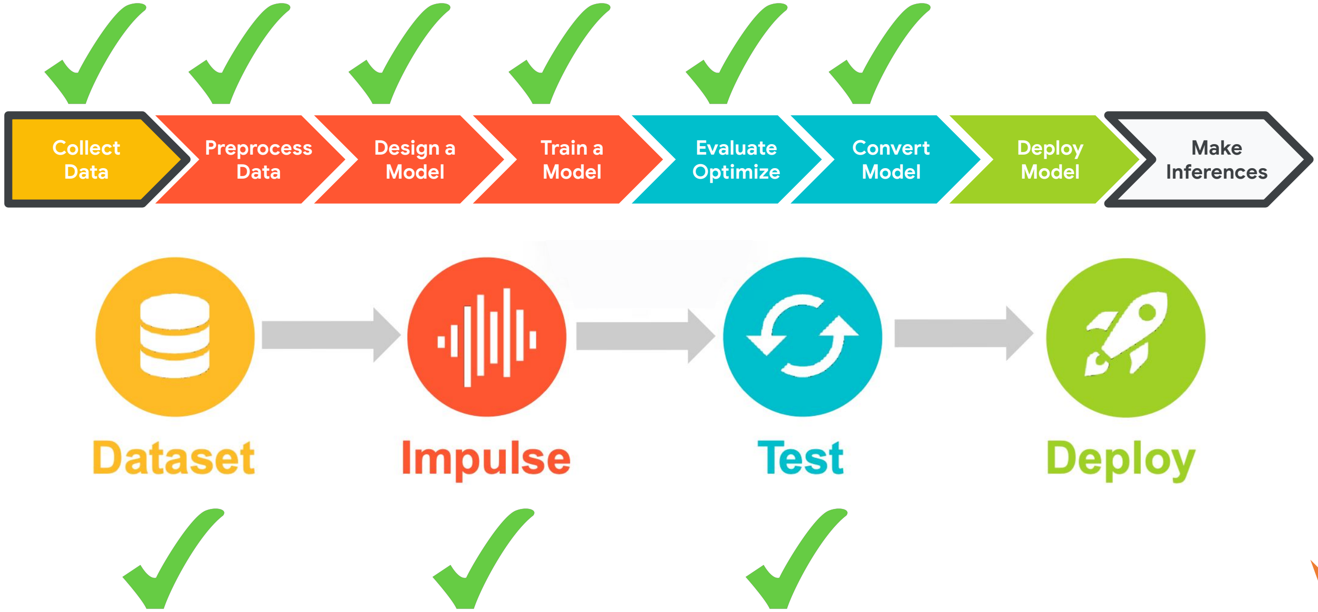
Enable EON™ Compiler
Same accuracy, up to 50% less memory. Open source.

Available optimizations for Transfer learning

Quantized (int8) Currently selected	RAM USAGE 66.1K FLASH USAGE 108.1K	LATENCY 58 ms ACCURACY -
Unoptimized (float32) Click to select	RAM USAGE 155.6K FLASH USAGE 193.8K	LATENCY 43 ms ACCURACY -

Estimate for Arduino Portenta H7 (Cortex-M7 480MHz)

Edge Impulse Project Dashboard



Collect Data

Preprocess Data

Design a Model

Train a Model

Evaluate Optimize

Convert Model

Deploy Model

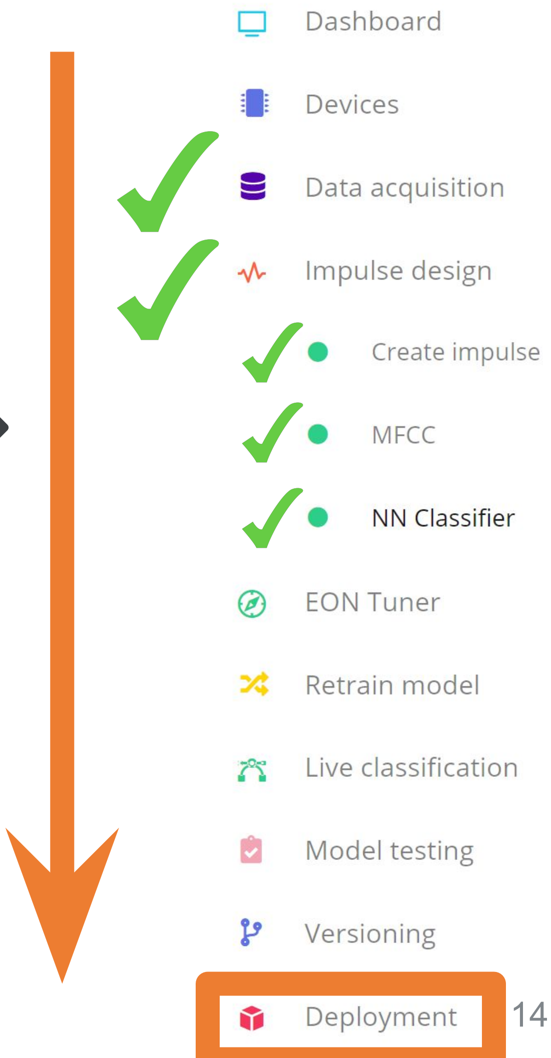
Make Inferences

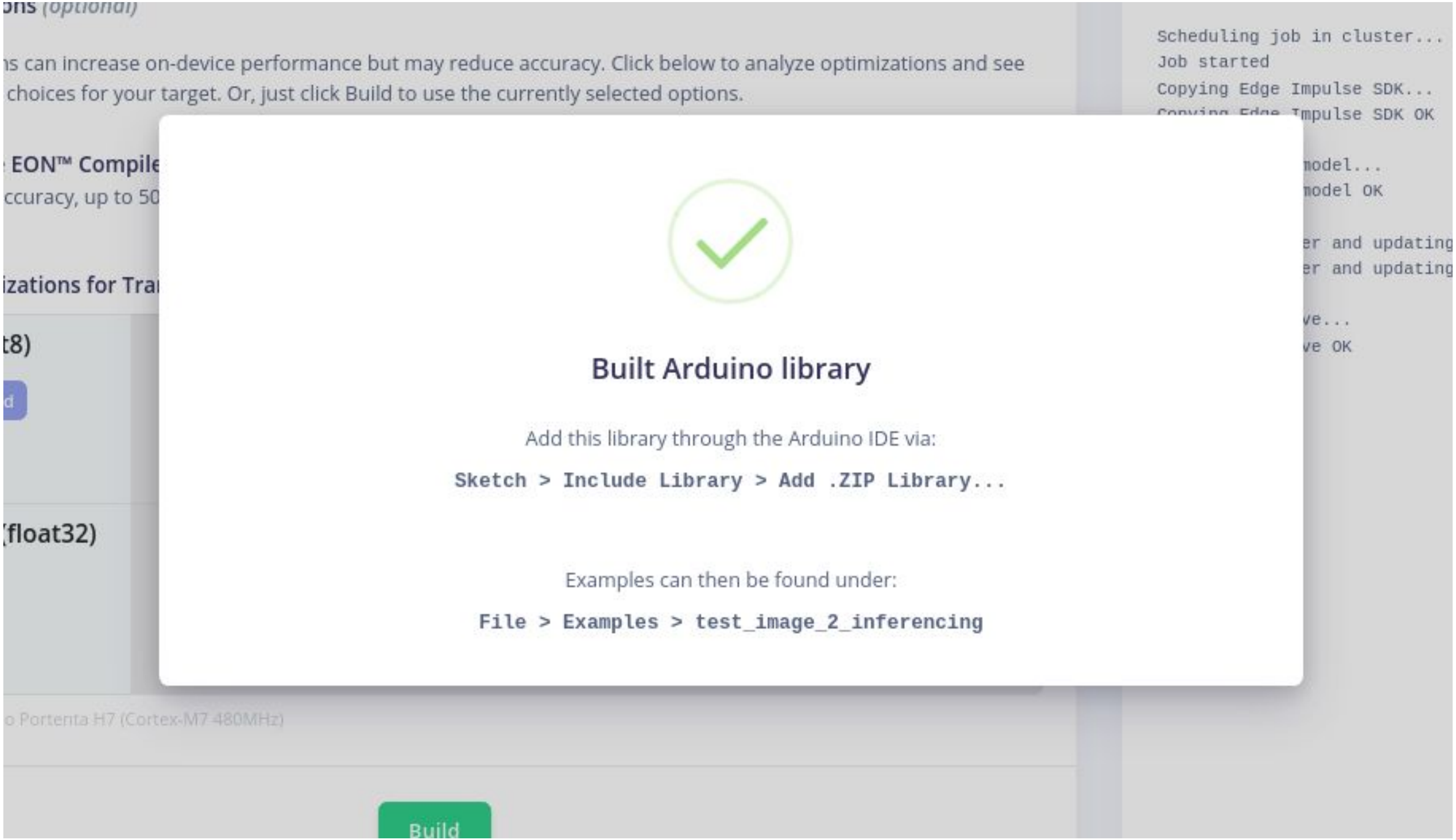
Dataset

Impulse

Test

Deploy





ons *(optional)*

ns can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see choices for your target. Or, just click Build to use the currently selected options.

EON™ Compile
ccuracy, up to 50

izations for Tra

t8)

(float32)

o Portenta H7 (Cortex-M7 480MHz)

```
Scheduling job in cluster...
Job started
Copying Edge Impulse SDK...
Copying Edge Impulse SDK OK
model...
model OK
er and updating
er and updating
ve...
ve OK
```

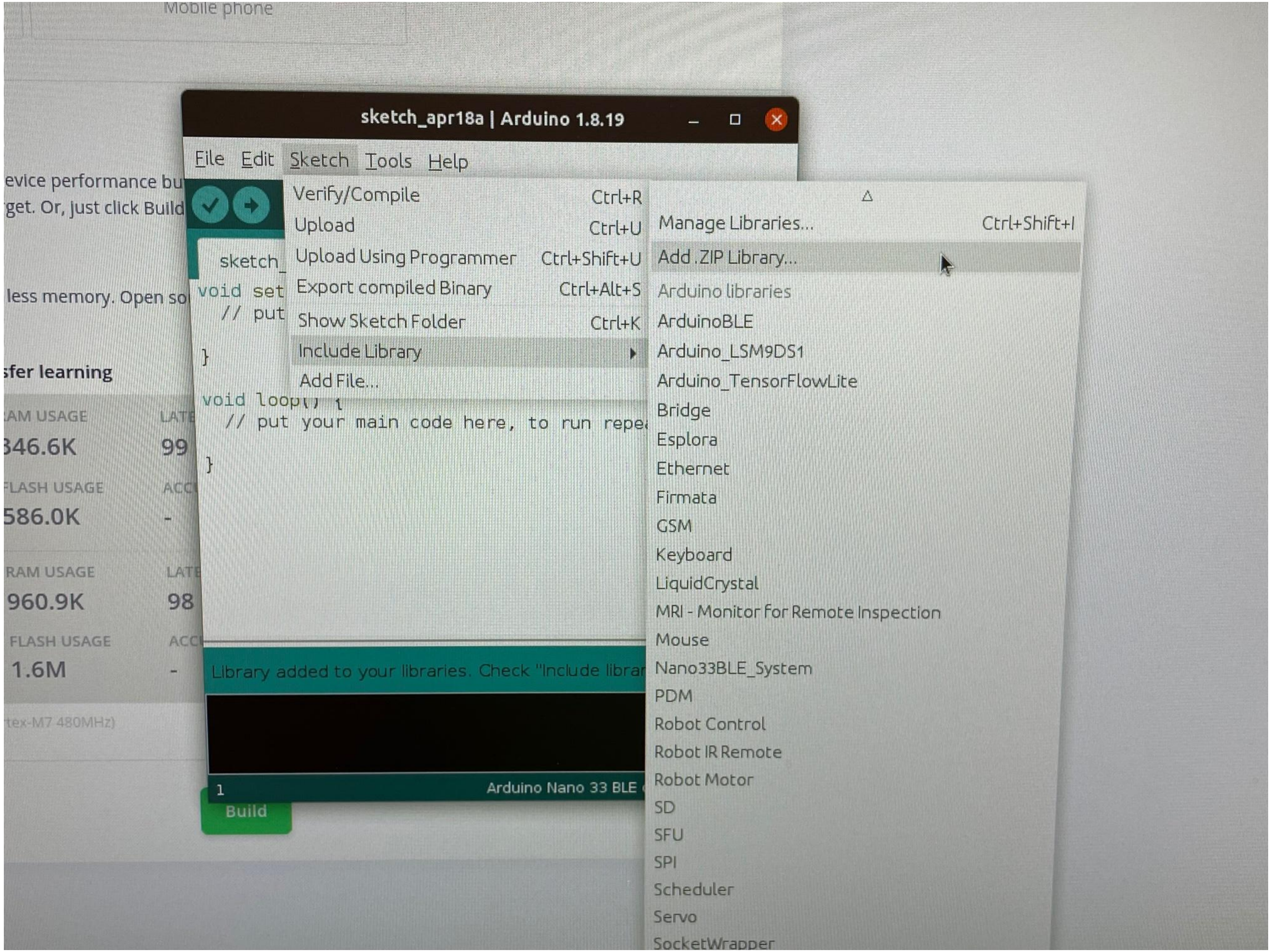
Built Arduino library

Add this library through the Arduino IDE via:

`Sketch > Include Library > Add .ZIP Library...`

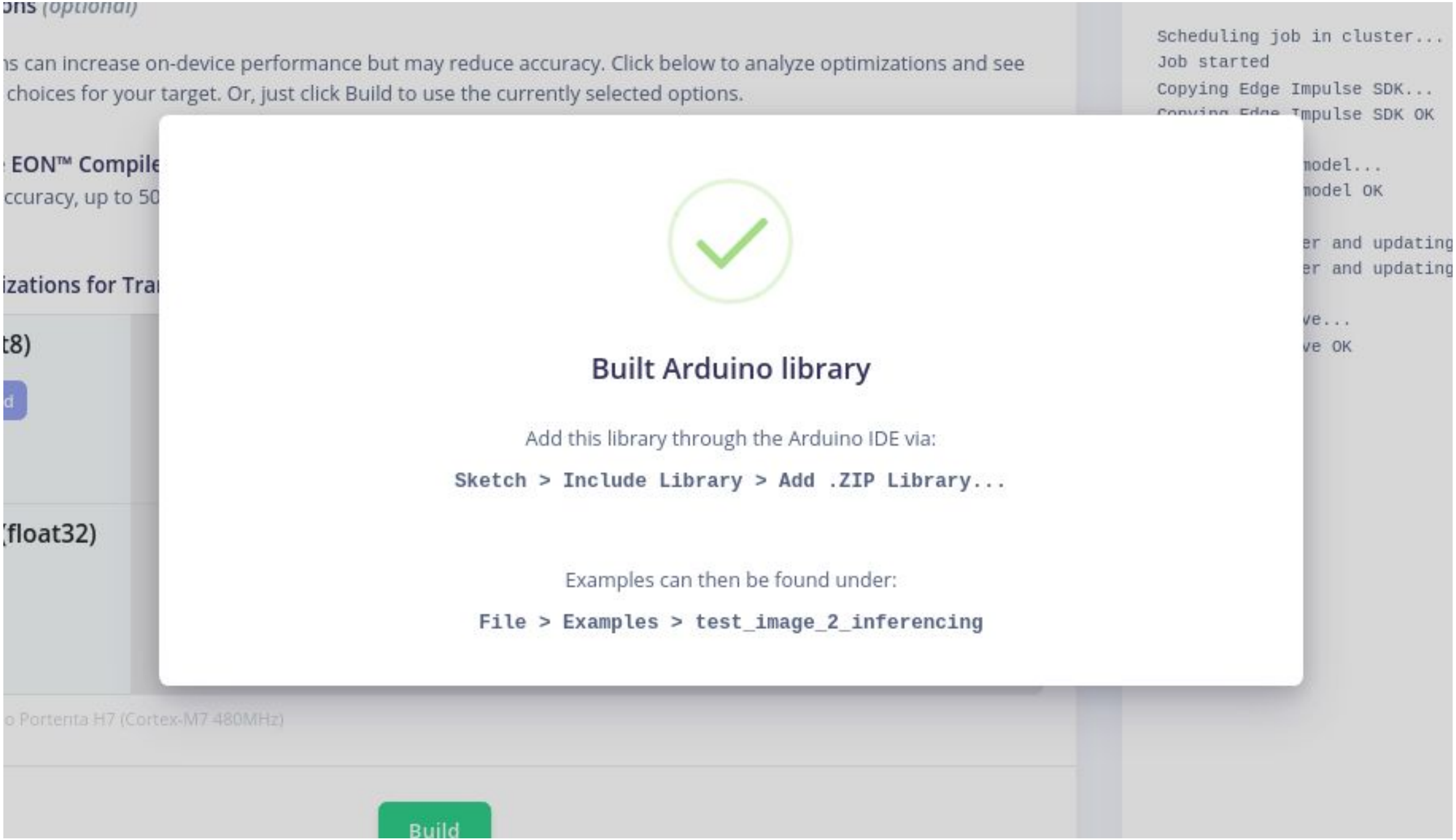
Examples can then be found under:

`File > Examples > test_image_2_inferencing`



```
void set
  // put

}

void loop() {
  // put your main code here, to run repea

}
```

Library added to your libraries. Check "Include librar



ons *(optional)*

ns can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see choices for your target. Or, just click Build to use the currently selected options.

EON™ Compile

ccuracy, up to 50

izations for Tra

t8)

(float32)

o Portenta H7 (Cortex-M7 480MHz)

```
Scheduling job in cluster...
Job started
Copying Edge Impulse SDK...
Copying Edge Impulse SDK OK
model...
model OK
er and updating
er and updating
ve...
ve OK
```

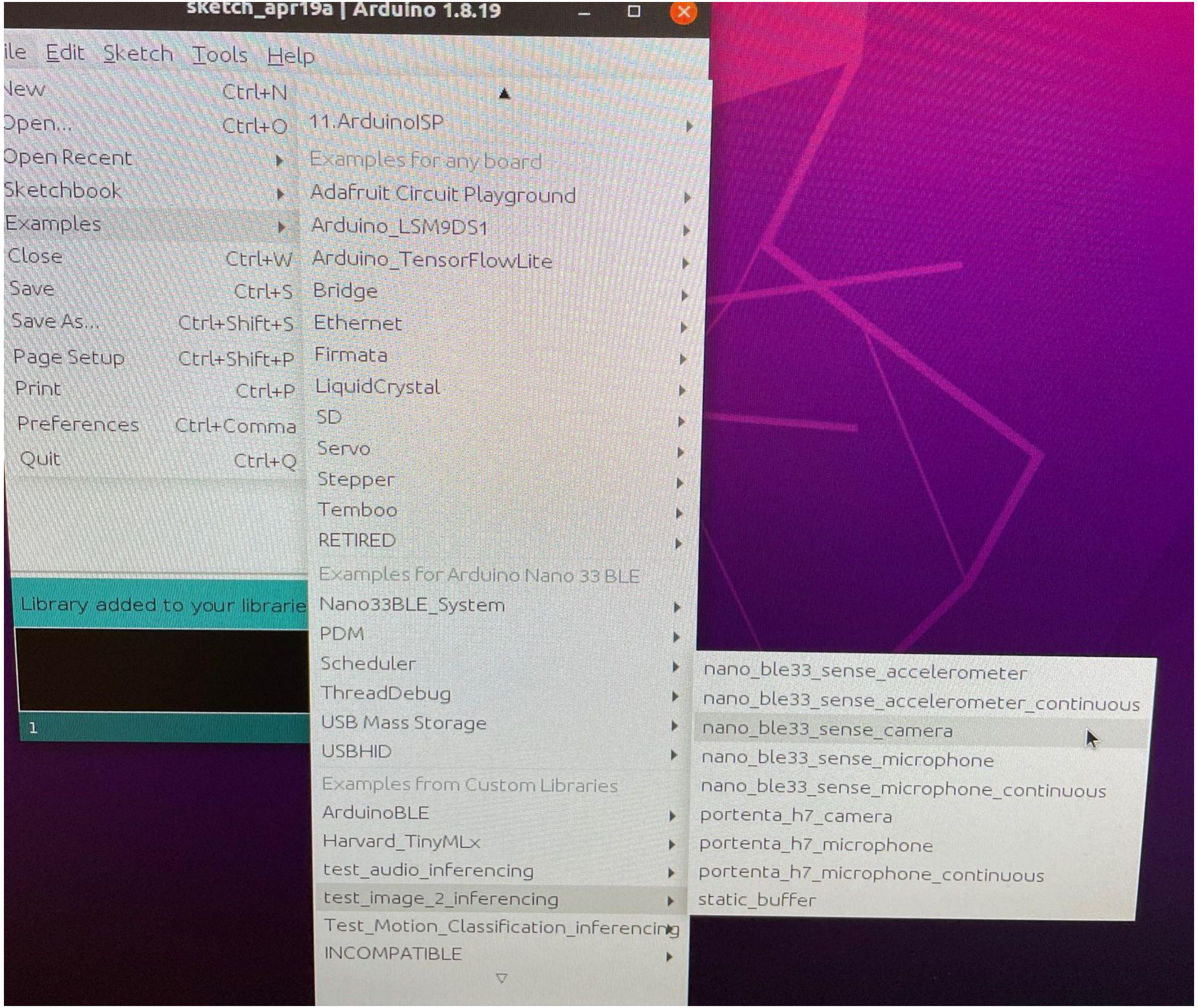
Built Arduino library

Add this library through the Arduino IDE via:

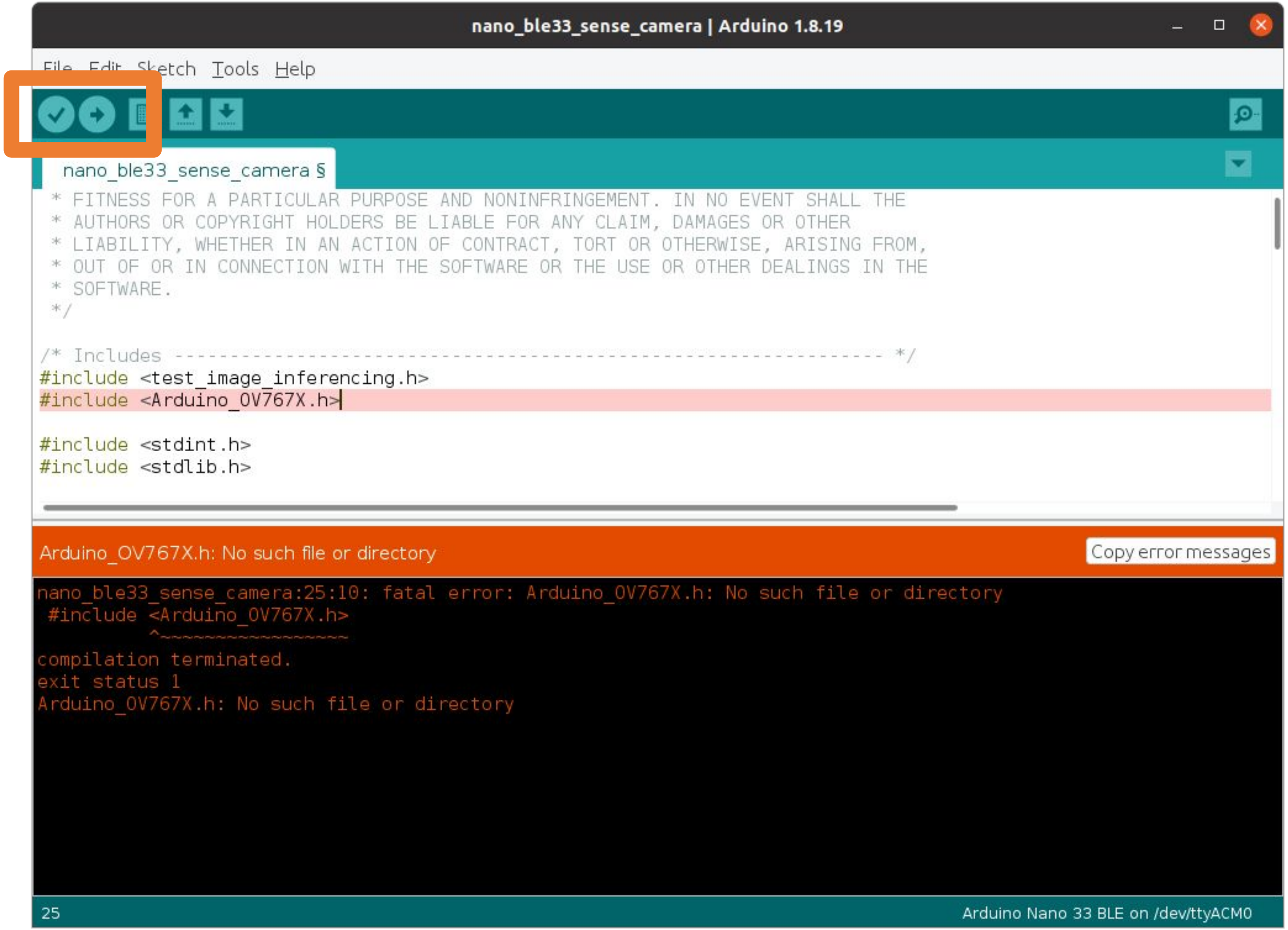
```
Sketch > Include Library > Add .ZIP Library...
```

Examples can then be found under:

```
File > Examples > test_image_2_inferencing
```



Library added to your librarie

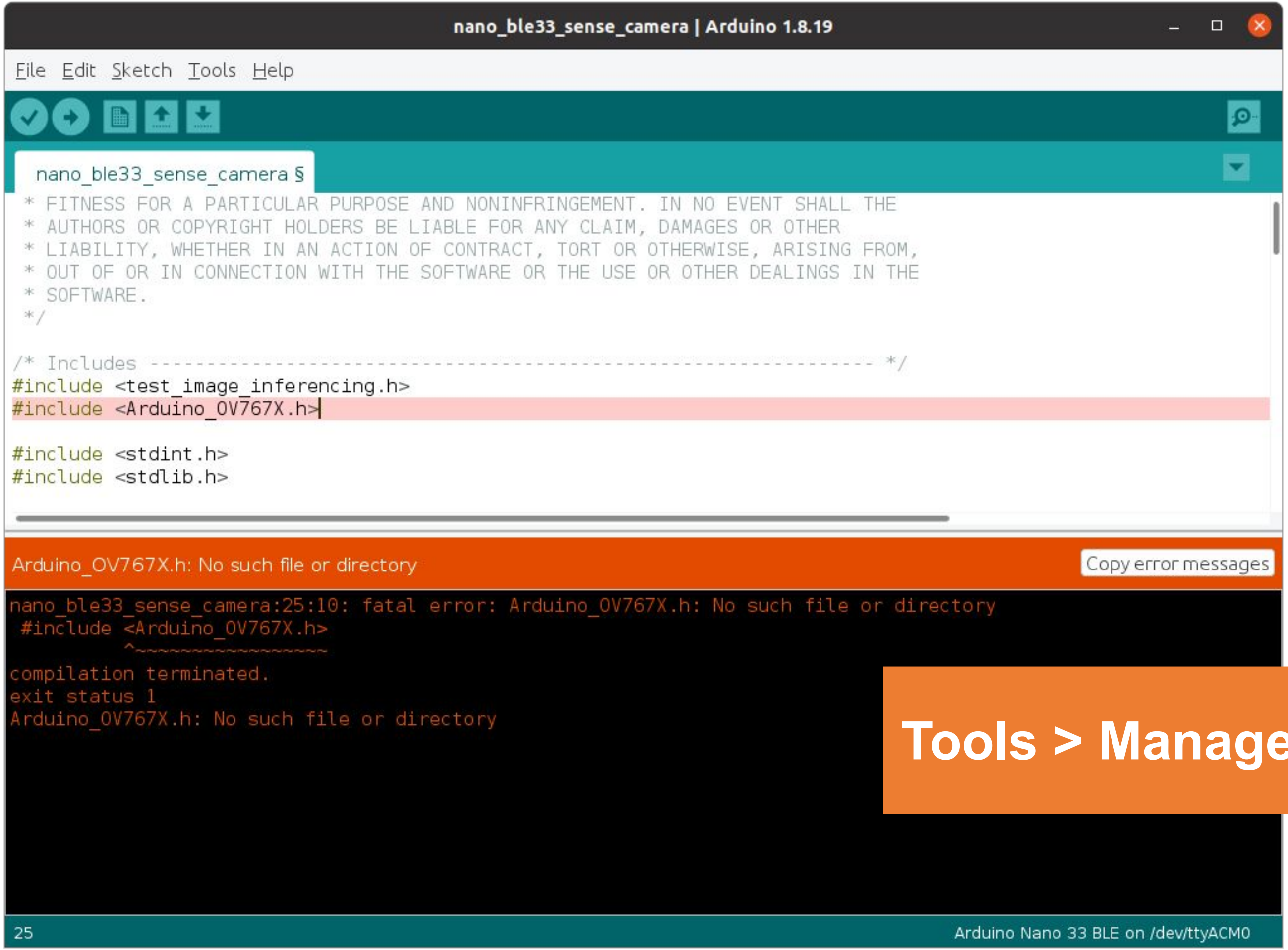


nano_ble33_sense_camera | Arduino 1.8.19
File Edit Sketch Tools Help
nano_ble33_sense_camera §
Copy error messages
25
Arduino Nano 33 BLE on /dev/ttyACM0

```
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

/* Includes ---------------------------------------------------------------- */
#include <test_image_inferencing.h>
#include <Arduino_OV767X.h>

#include <stdint.h>
#include <stdlib.h>
```

Arduino_OV767X.h: No such file or directory

```
nano_ble33_sense_camera:25:10: fatal error: Arduino_OV767X.h: No such file or directory
 #include <Arduino_OV767X.h>
          ^~~~~~~~~~~~~~~~~~
compilation terminated.
exit status 1
Arduino_OV767X.h: No such file or directory
```

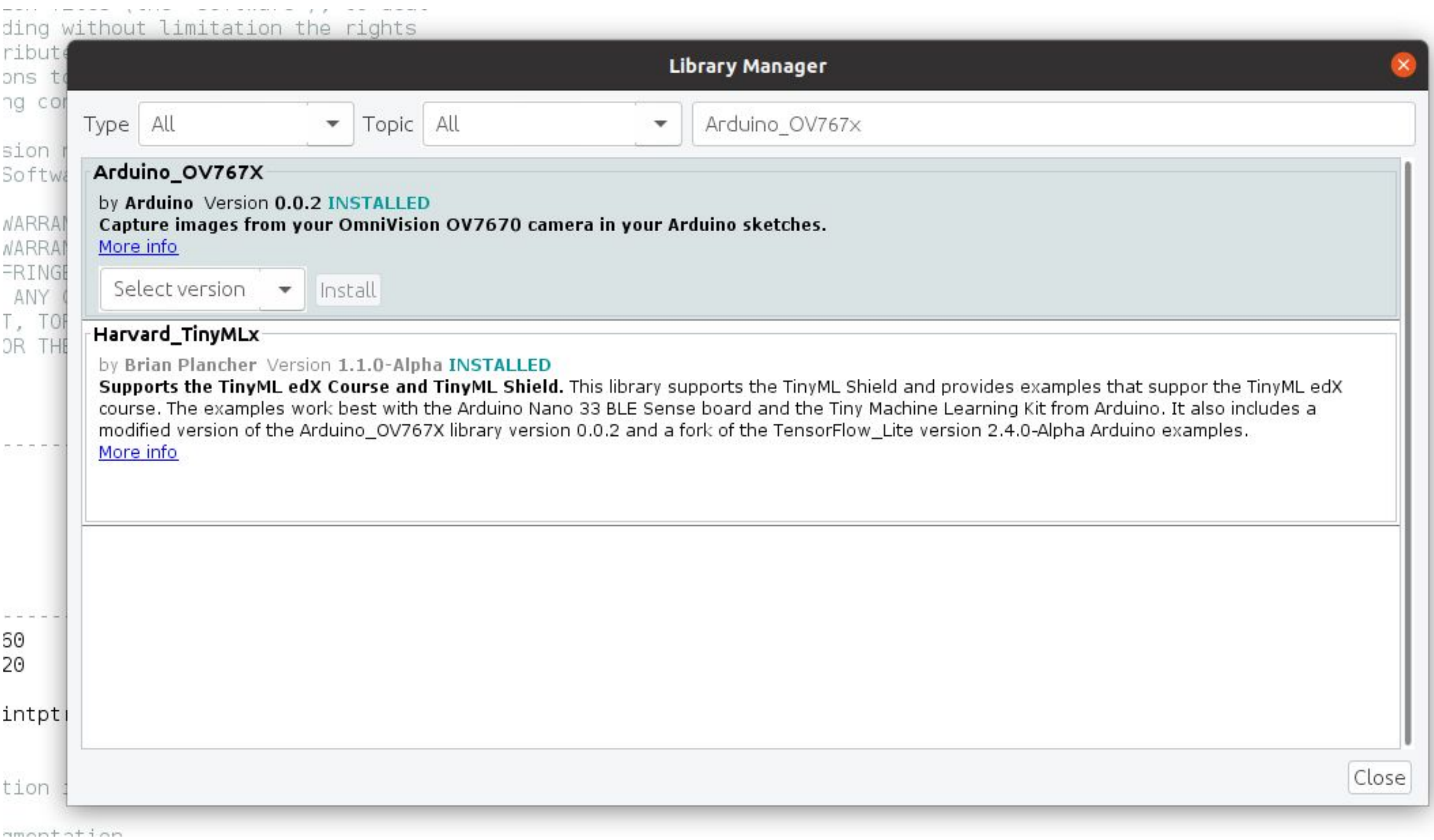

nano_ble33_sense_camera | Arduino 1.8.19
File Edit Sketch Tools Help
nano_ble33_sense_camera §
Copy error messages
25
Arduino Nano 33 BLE on /dev/ttyACM0

```
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

/* Includes ---------------------------------------------------------------- */
#include <test_image_inferencing.h>
#include <Arduino_OV767X.h>

#include <stdint.h>
#include <stdlib.h>
```

Arduino_OV767X.h: No such file or directory

```
nano_ble33_sense_camera:25:10: fatal error: Arduino_OV767X.h: No such file or directory
 #include <Arduino_OV767X.h>
          ^~~~~~~~~~~~~~~~~~
compilation terminated.
exit status 1
Arduino_OV767X.h: No such file or directory
```

Tools > Manage Libraries



Library Manager

Arduino_OV767x

Arduino_OV767X

by **Arduino** Version **0.0.2** INSTALLED

Capture images from your OmniVision OV7670 camera in your Arduino sketches.

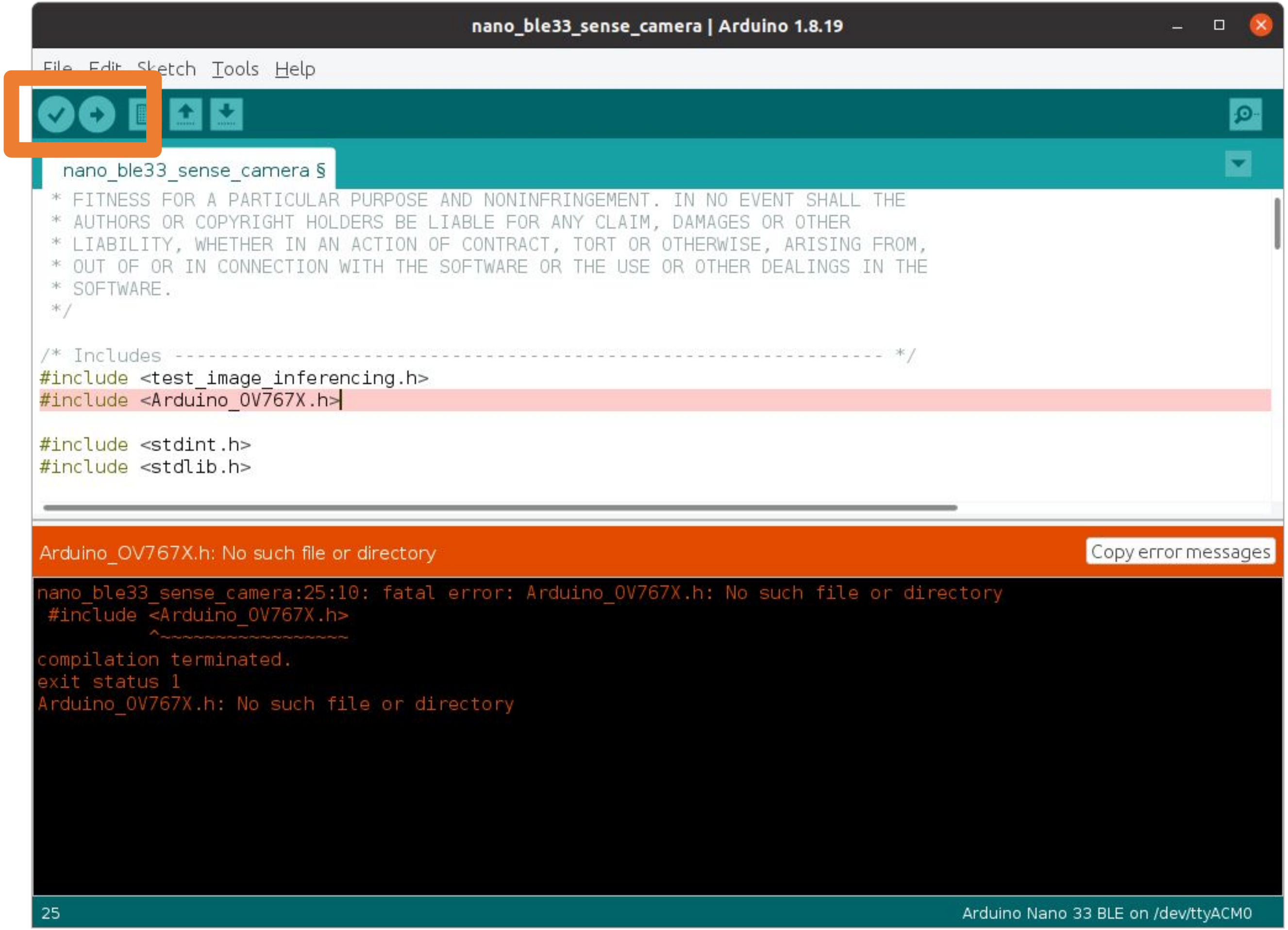
More info

Harvard_TinyMLx

by **Brian Plancher** Version **1.1.0-Alpha** INSTALLED

Supports the TinyML edX Course and TinyML Shield. This library supports the TinyML Shield and provides examples that suppor the TinyML edX course. The examples work best with the Arduino Nano 33 BLE Sense board and the Tiny Machine Learning Kit from Arduino. It also includes a modified version of the Arduino_OV767X library version 0.0.2 and a fork of the TensorFlow_Lite version 2.4.0-Alpha Arduino examples.

More info

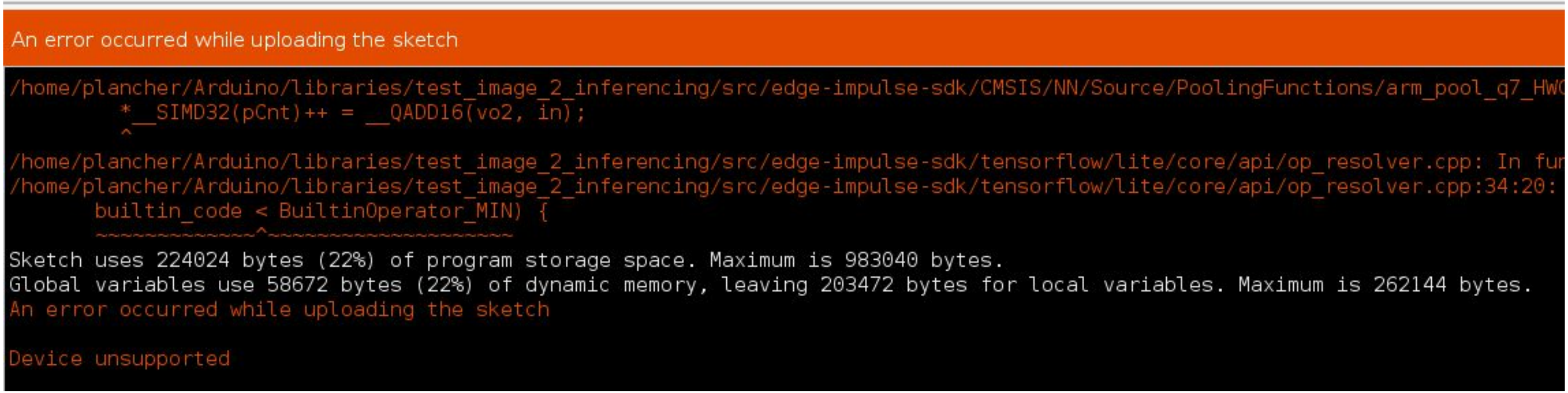


nano_ble33_sense_camera | Arduino 1.8.19
File Edit Sketch Tools Help
nano_ble33_sense_camera §

```
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
 * SOFTWARE.
 */

/* Includes ---------------------------------------------------------------- */
#include <test_image_inferencing.h>
#include <Arduino_OV767X.h>

#include <stdint.h>
#include <stdlib.h>
```

Arduino_OV767X.h: No such file or directory

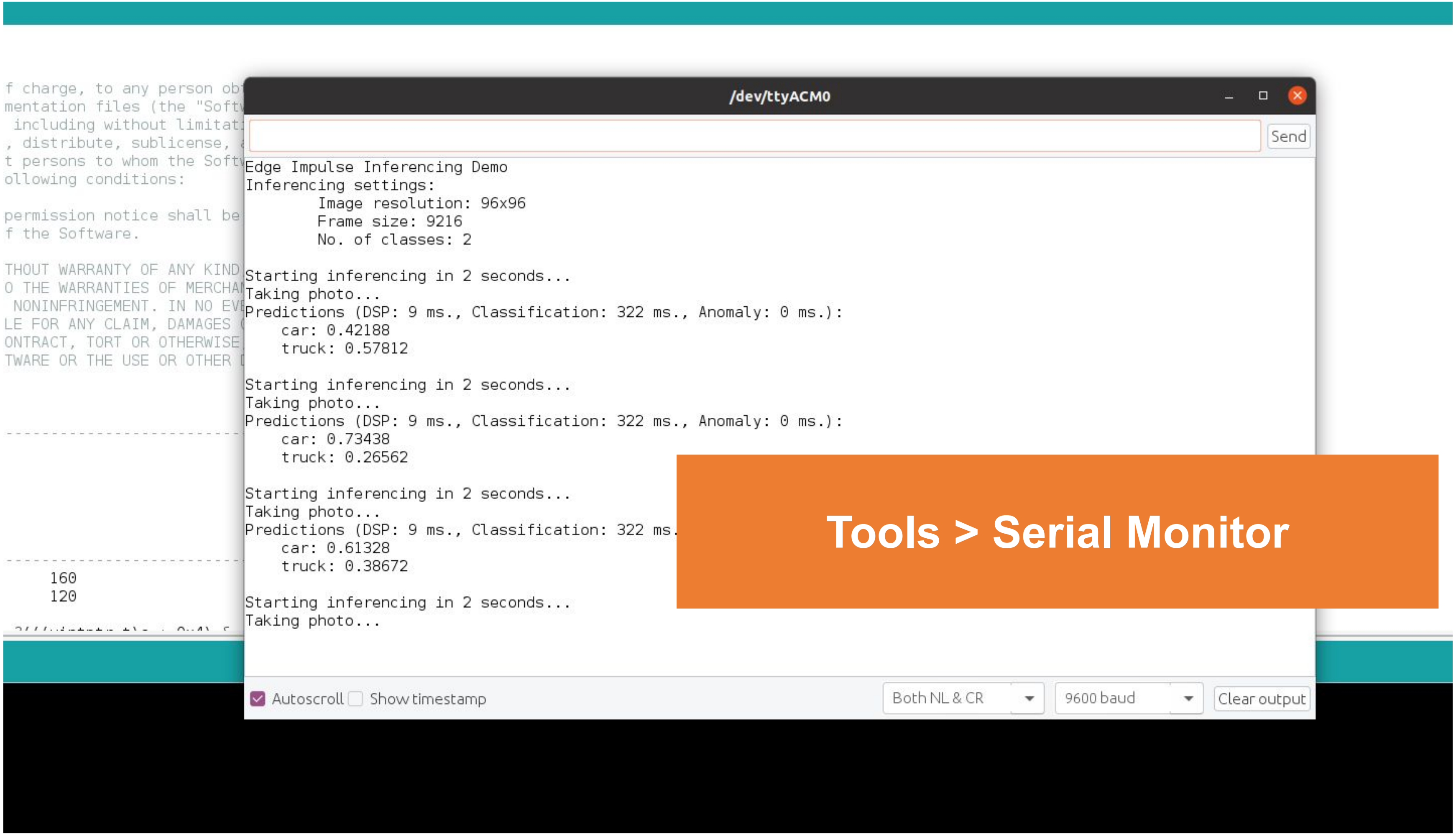
```
nano_ble33_sense_camera:25:10: fatal error: Arduino_OV767X.h: No such file or directory
 #include <Arduino_OV767X.h>
          ^~~~~~~~~~~~~~~~~~
compilation terminated.
exit status 1
Arduino_OV767X.h: No such file or directory
```



An error occurred while uploading the sketch

```
/home/plancher/Arduino/libraries/test_image_2_inferencing/src/edge-impulse-sdk/CMSIS/NN/Source/PoolingFunctions/arm_pool_q7_HWC
         *__SIMD32(pCnt)++ = __QADD16(vo2, in);
         ^

/home/plancher/Arduino/libraries/test_image_2_inferencing/src/edge-impulse-sdk/tensorflow/lite/core/api/op_resolver.cpp: In fu
/home/plancher/Arduino/libraries/test_image_2_inferencing/src/edge-impulse-sdk/tensorflow/lite/core/api/op_resolver.cpp:34:20:
       builtin_code < BuiltinOperator_MIN) {
       ~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~
Sketch uses 224024 bytes (22%) of program storage space. Maximum is 983040 bytes.
Global variables use 58672 bytes (22%) of dynamic memory, leaving 203472 bytes for local variables. Maximum is 262144 bytes.
An error occurred while uploading the sketch

Device unsupported
```

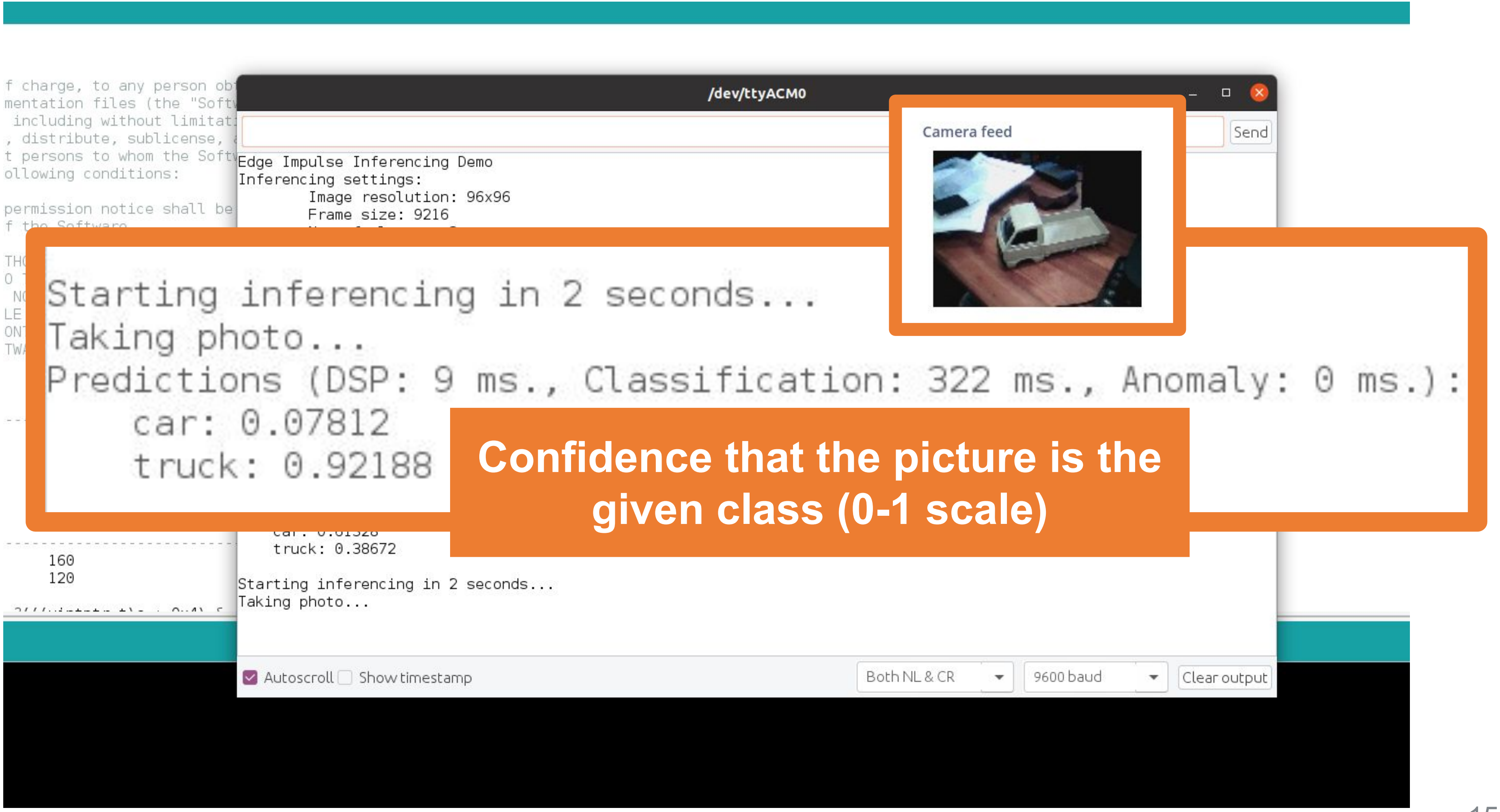
Double Tap RESET for Bootloader Mode!



```
f charge, to any person ob
mentation files (the "Soft
 including without limitat
, distribute, sublicense,
t persons to whom the Soft
ollowing conditions:

permission notice shall be
f the Software.

THOUT WARRANTY OF ANY KIND
O THE WARRANTIES OF MERCHAN
 NONINFRINGEMENT. IN NO EV
LE FOR ANY CLAIM, DAMAGES
ONTRACT, TORT OR OTHERWISE
TWARE OR THE USE OR OTHER

    160
    120
```

```
Edge Impulse Inferencing Demo
Inferencing settings:
        Image resolution: 96x96
        Frame size: 9216
        No. of classes: 2

Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification: 322 ms., Anomaly: 0 ms.):
    car: 0.42188
    truck: 0.57812

Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification: 322 ms., Anomaly: 0 ms.):
    car: 0.73438
    truck: 0.26562

Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification: 322 ms.
    car: 0.61328
    truck: 0.38672

Starting inferencing in 2 seconds...
Taking photo...
```

Tools > Serial Monitor



```
f charge, to any person ob
mentation files (the "Soft
 including without limitat
, distribute, sublicense, 
t persons to whom the Soft
ollowing conditions:

permission notice shall be
f the Software

THE
O
 NO
LE
ONT
TWA

--

-----------------------------
    160
    120
```

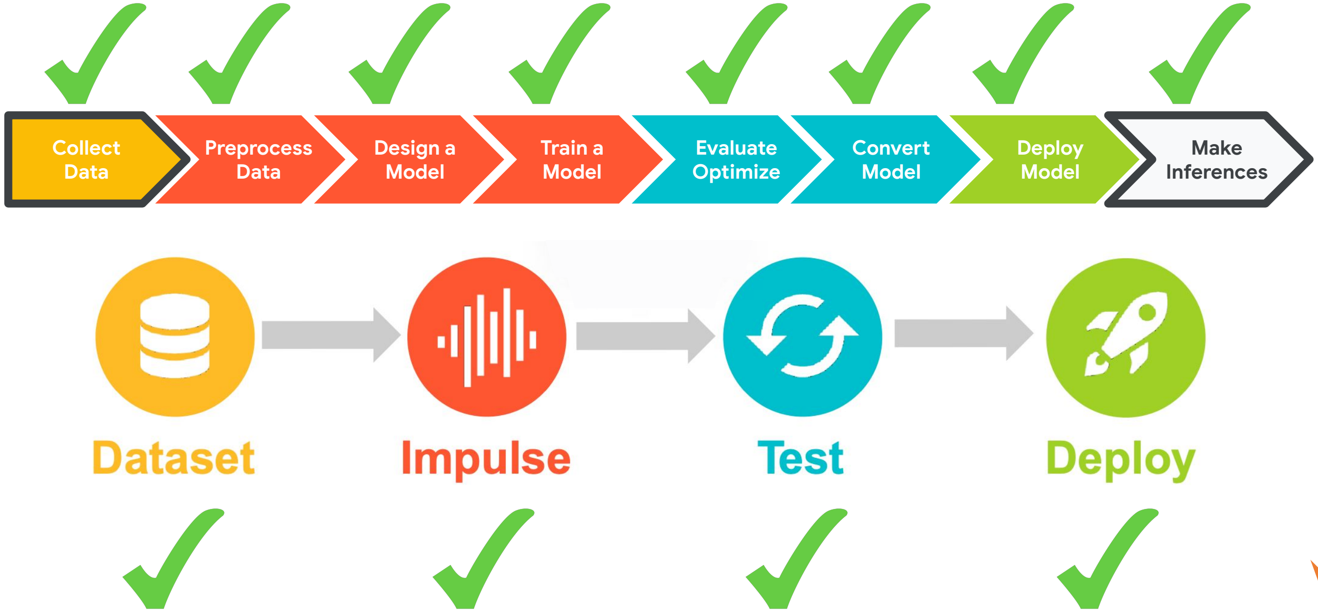
```
Edge Impulse Inferencing Demo
Inferencing settings:
        Image resolution: 96x96
        Frame size: 9216
```

```
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification: 322 ms., Anomaly: 0 ms.):
    car: 0.07812
    truck: 0.92188
```

Confidence that the picture is the given class (0-1 scale)

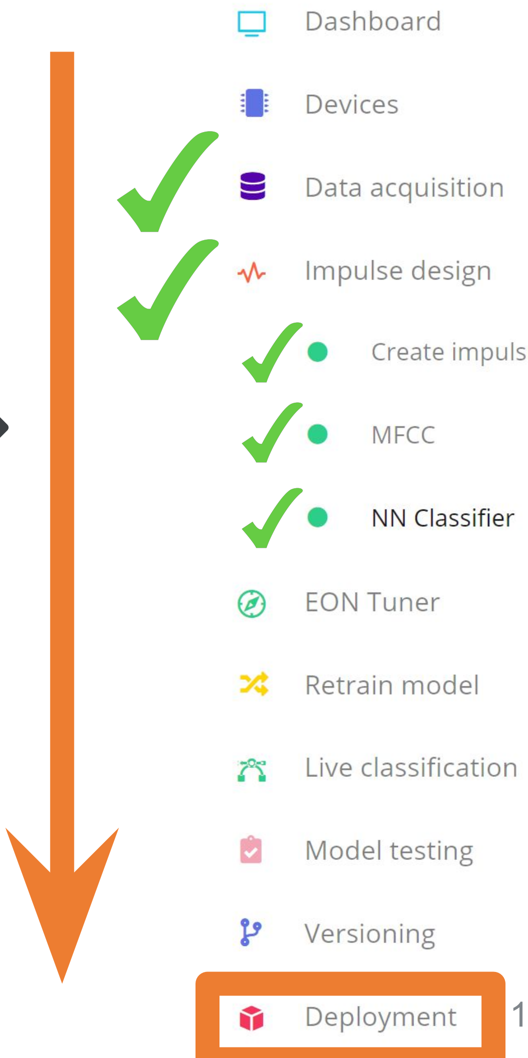
```
    car: 0.01328
    truck: 0.38672

Starting inferencing in 2 seconds...
Taking photo...
```

Edge Impulse Project Dashboard



Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences

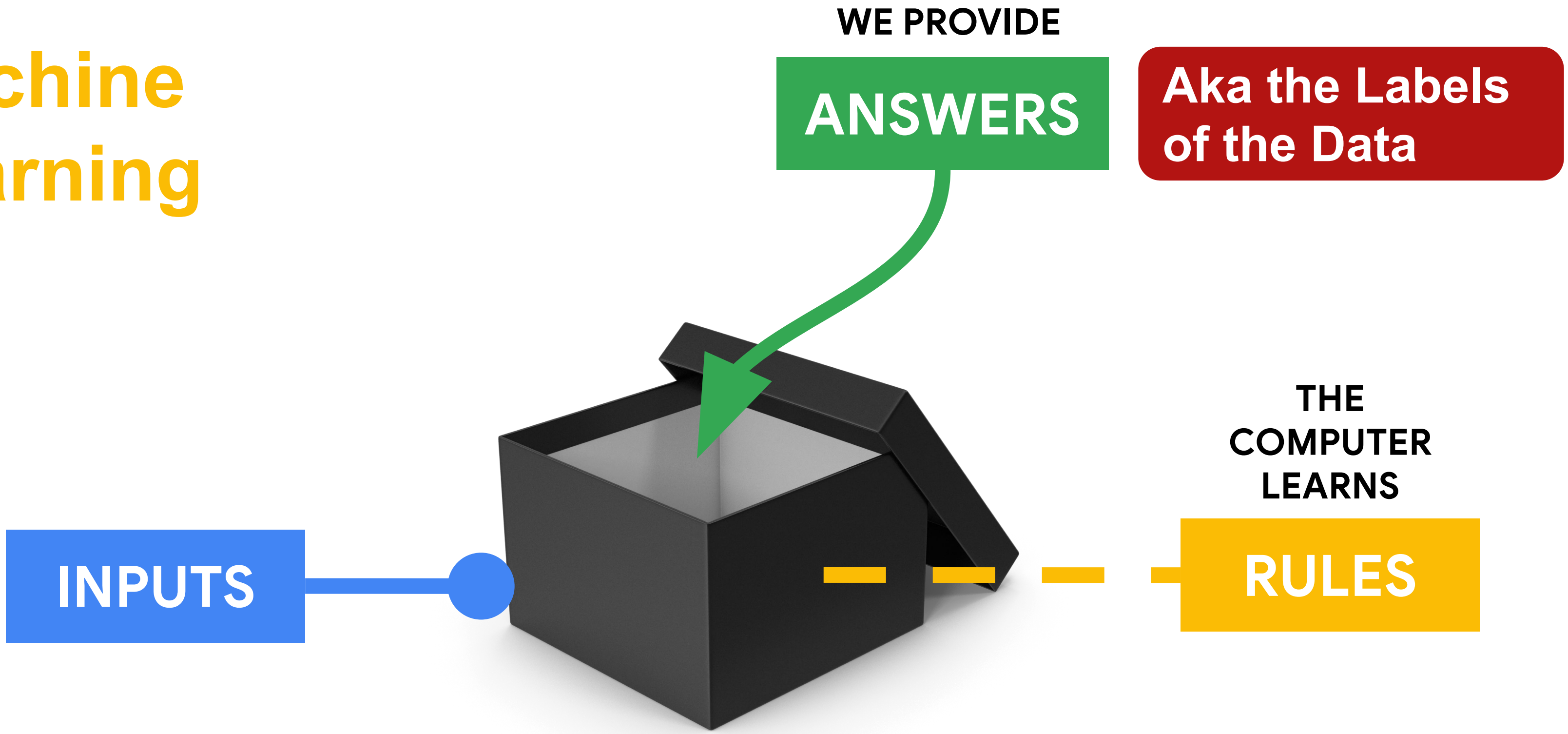
Dataset → Impulse → Test → Deploy



Today’s Agenda

- Introduction to Computer Vision
- Hands-on Computer Vision: Thing Translator
- Building an Object Detection Dataset
- Training our Model using Transfer Learning
- Deploying our Model onto our Arduino
- **Summary**

Machine Learning



WE PROVIDE

ANSWERS

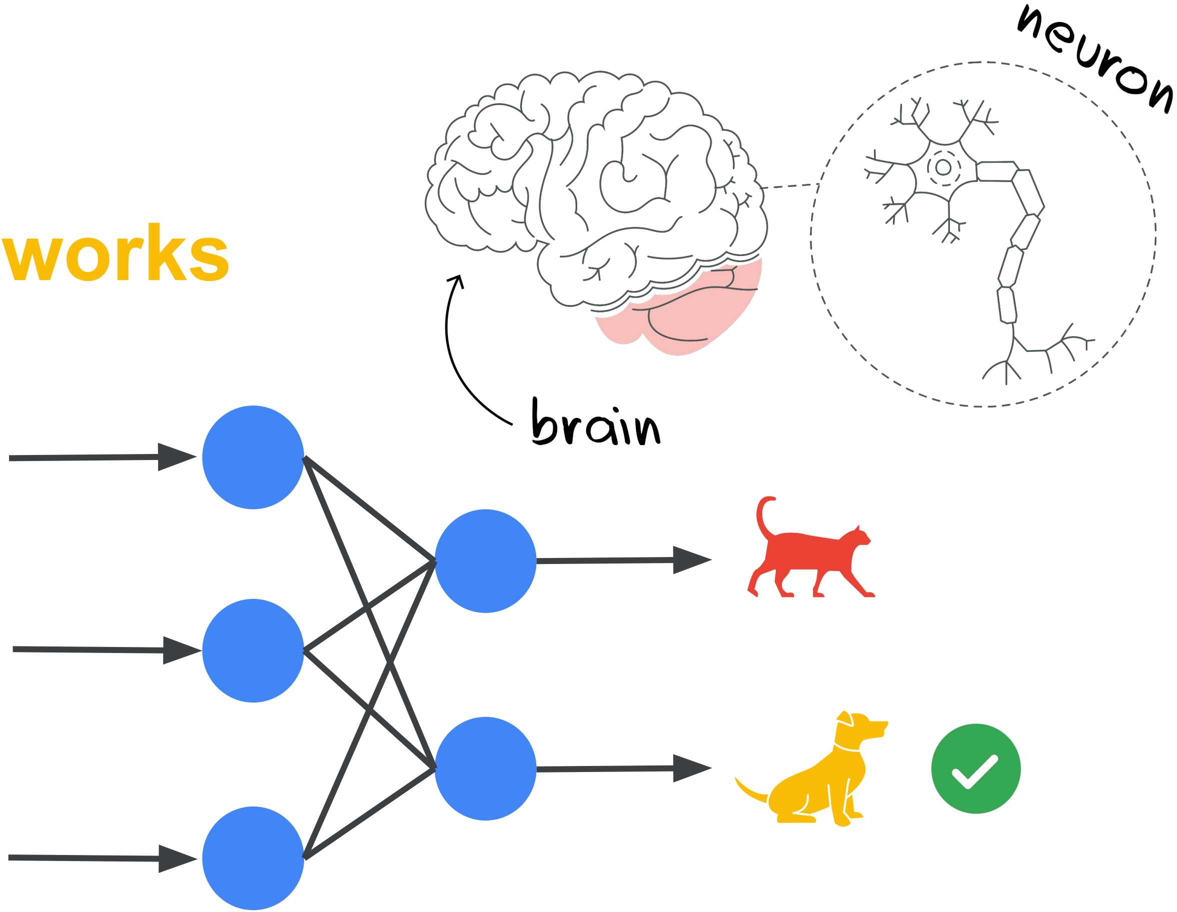
Aka the Labels of the Data

THE COMPUTER LEARNS

RULES

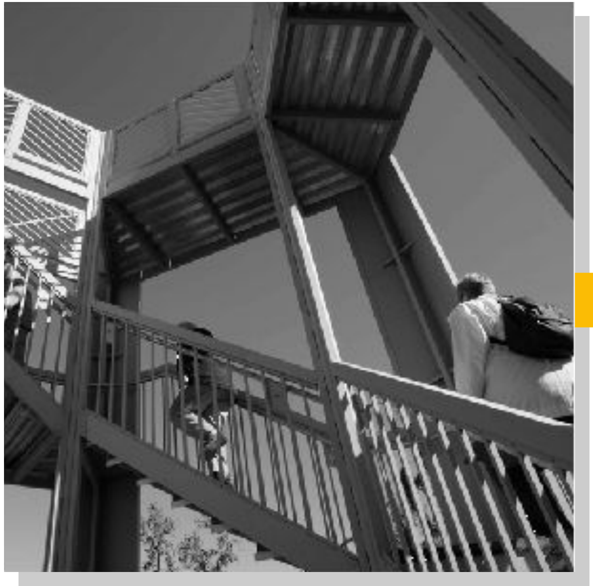
INPUTS

Deep Learning with **Neural Networks**



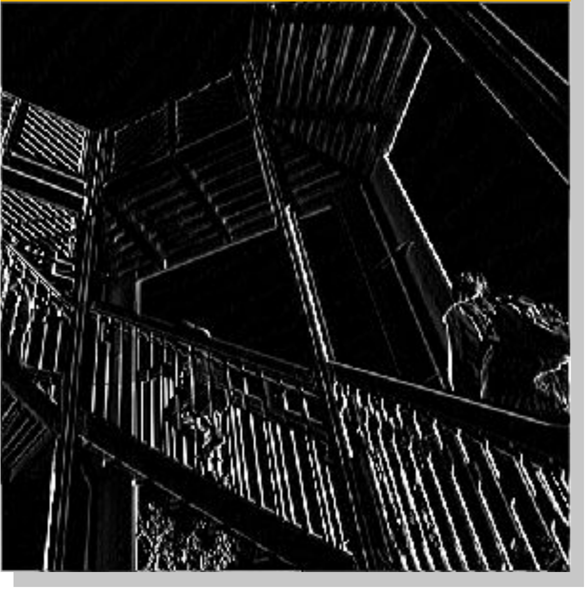
Colab Link

Features can be found with **Convolutions**



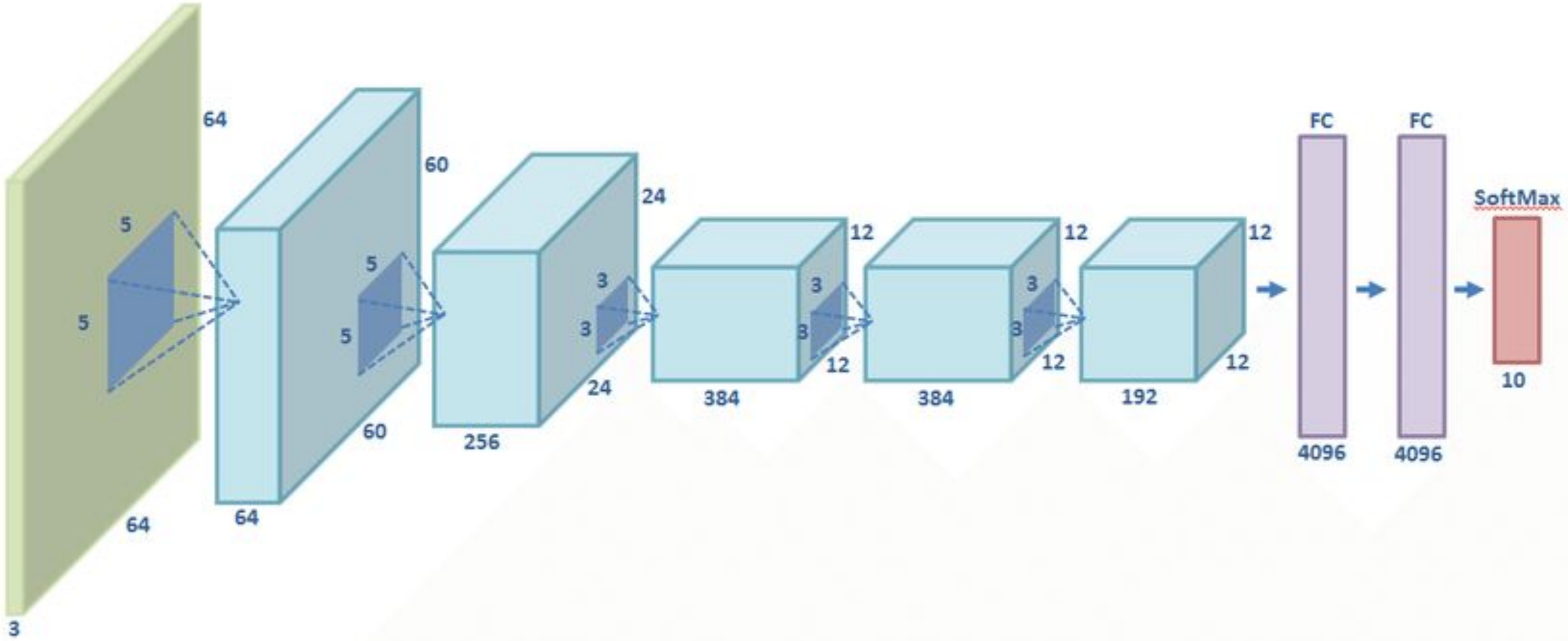
-1	0	1
-2	0	2
-1	0	1

Features



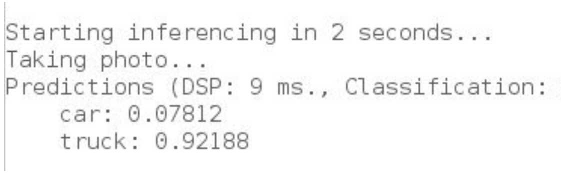
AlexNet Paper

Convolutional Neural Networks

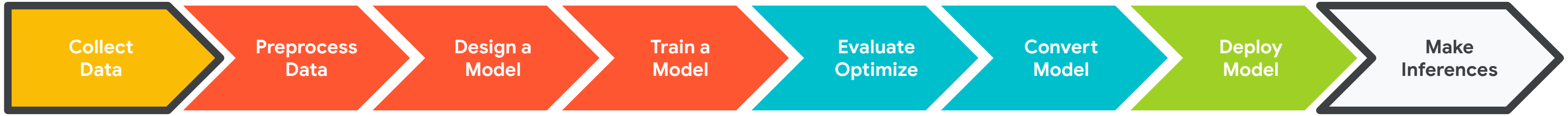


The TinyML Workflow

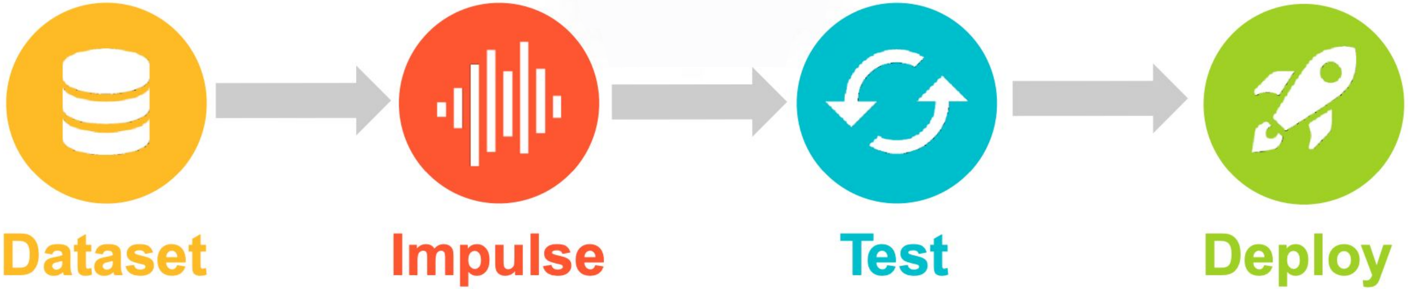
Camera feed



```
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 9 ms., Classification:
    car: 0.07812
    truck: 0.92188
```



Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Optimize → Convert Model → Deploy Model → Make Inferences



Dataset → Impulse → Test → Deploy

Feedback Link:

bit.ly/SciTinyML-22A-CV

African Regional Workshop on SciTinyML: Scientific Use of Machine Learning on Low-Power Devices

25-29 April 2022
Online

Further information:
http://indico.ictp.it/event/9792/
smr3708@ictp.it

Convolutions for Hands-on Computer Vision

Brian Plancher
Harvard John A. Paulson School of Engineering and Applied Sciences
Barnard College, Columbia University
brianplancher.com

https://arduino.github.io/arduino-cli/0.21/installation/

Edge Impulse CLI Notes:

1. Install the Arduino CLI
   a. On linux:

      ```
      curl -fsSL https://raw.githubusercontent.com/arduino/arduino-cli/master/install.sh | sh
      ```

   b. On mac:

      ```
      brew update
      brew install arduino-cli
      ```

   c. Or view the link for binaries
2. Add to your `.bashrc`:

   ```
   # Arduino (CLI)
   export PATH="ARDUINO_INSTALL_LOCATION/bin:$PATH"
   ```

   Where `ARDUINO_INSTALL_LOCATION` is e.g.,: `$HOME/Documents/arduino-1.8.19` https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation

Edge Impulse CLI Notes:

1. Install the Edge Impulse CLI
   a. Install Node.js by following the link or on Linux:

   ```
   curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
   sudo apt-get install -y nodejs
   ```

   b. Run: `npm install -g edge-impulse-cli --force`

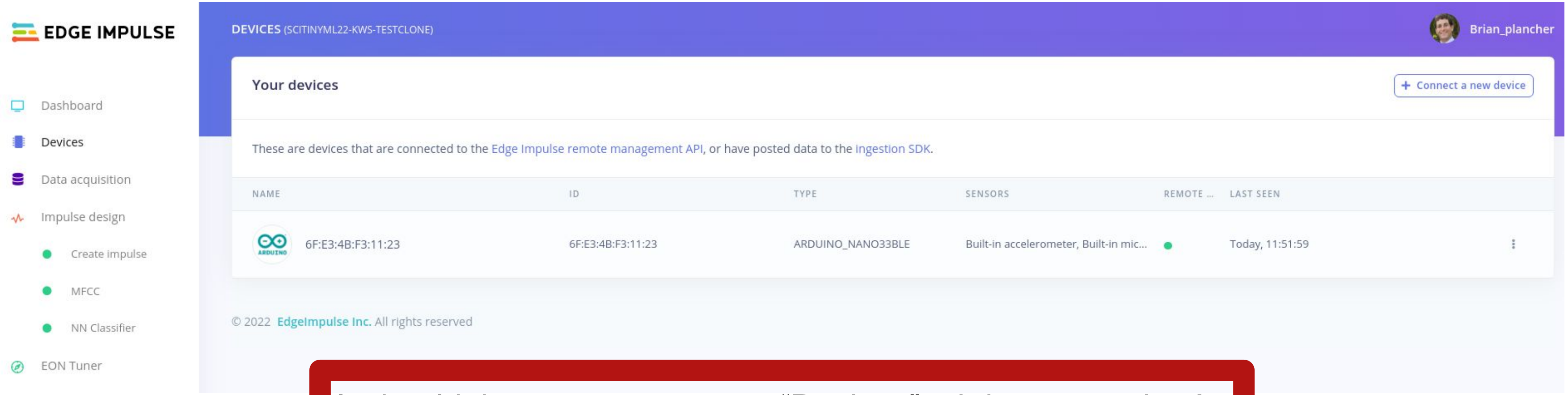
   c. Add to your `.bashrc`:

   ```
   # EI (CLI)
   export PATH="$HOME/.npm-global/bin:$PATH"
   ```

2. Run `edge-impulse-daemon --clean` to start the daemon and then follow the instructions in the terminal to add it to your current project using your edge impulse account!

https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation

Edge Impulse CLI Notes:



Your devices

These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

NAME	ID	TYPE	SENSORS	REMOTE ...	LAST SEEN
6F:E3:4B:F3:11:23	6F:E3:4B:F3:11:23	ARDUINO_NANO33BLE	Built-in accelerometer, Built-in mic...	●	Today, 11:51:59

© 2022 EdgeImpulse Inc. All rights reserved

It should then appear on your “Devices” tab in your project!

And then if you go to “Data Acquisition” you should be able to proceed as you would with the standard instructions!